



Developpez

Le Mag

Édition de juin – juillet 2015

Numéro 58

Magazine en ligne gratuit

Diffusion de copies conformes à l'original autorisée

Réalisation : Alexandre Pottiez

Rédaction : la rédaction de Developpez

Contact : magazine@redaction-developpez.com

Sommaire

Java	Page	2
Club	Page	4
Excel	Page	6
Access	Page	16
JavaScript	Page	23
Windows	Page	33

Éditorial

Pour vos vacances d'été, n'oubliez pas votre magazine préféré pour flâner tranquillement sur la plage tout en restant connecté aux nouvelles technologies. Profitez-en bien !

La rédaction

Article Excel



Excel et l'enregistreur de macro - Enregistrer les actions répétitives dans un fichier Excel

Si vous répétez régulièrement les mêmes manipulations lors du traitement d'un fichier Excel, ce tutoriel sur l'enregistreur de macro pourrait vous intéresser.

par [Pierre Fauconnier](#)

Page 6



Article Windows

À la découverte de Windows 10 Insider Preview - télécharger, installer et découvrir

Découvrez la nouvelle version de Windows en prévisualisation.

par [Michel Martin](#)

Page 33



Java

Les dernières news Java

Java souffle sa 20e bougie

que pensez-vous de l'évolution du langage et de son avenir sous l'égide d'Oracle ?

Voilà déjà 20 ans que le langage de programmation Java existe dans l'univers du développement informatique, et il n'est pas prêt à prendre sa retraite.

Les travaux sur Java débutent en 1991, sous le nom du projet Green, dans les laboratoires de Sun Microsystems. Le projet est dirigé par Patrick Naughton, Mike Sheridan et James Gosling. Il donne naissance finalement au langage de programmation « Oak ».

Basé sur une syntaxe très proche de celle du C++, le langage est destiné à la conception des applications pour les petits appareils. Il reprend le concept de machine virtuelle déjà exploité dans le Pascal UCSD.

En 1992, Sun développe avec Oak un assistant numérique, qui sera un échec. Mais, les éléments de base de Oak ont été repris pour créer un nouveau langage appelé Java, qui fut présenté pour la première fois le 23 mai 1995, lors de la conférence SunWorld à San Francisco.

Sun dévoile alors le navigateur Web HotJava écrit en Java, et capable d'exécuter des applets Java. Le concept a été repris par les autres navigateurs, dont Netscape.

L'explosion de la bulle d'Internet et le fait que Java ne dépende pas d'une plateforme particulière ont contribué au succès du langage de programmation.

Robuste, performant, multiplateforme et disposant d'un vaste ensemble de bibliothèques intégrées, Java s'est positionné à cette époque comme le langage de référence pour le développement d'applications d'entreprises.

Bénéficiant du support de certains géants de l'IT, dont IBM, Java devient l'un des langages les plus utilisés dans le monde. Au fil des versions, il gagne en maturité et s'enrichit des nouvelles fonctionnalités importantes, tout en concevant ses aspects fondamentaux.



Java constitue aujourd'hui la colonne vertébrale d'une multitude de logiciels présents aussi bien dans notre vie personnelle que dans notre vie profession-

nelle. Il est le langage de programmation utilisé par 9 millions de développeurs et embarqué aujourd'hui dans 7 milliards d'appareils et objets, selon Oracle.

Pour Mark Reinhold, architecte en chef du projet Java, l'autre atout de Java est sa lisibilité et sa simplicité.

Les développeurs d'applications d'entreprise peuvent choisir librement dans un écosystème de 30 implémentations compatibles des standards Java EE 6 et Java EE 7, proposées par 12 éditeurs différents. Plus de 125 millions d'appareils multimédias basés sur Java ont déjà été déployés, et plus de 10 milliards de cartes JavaCard ont été déployées depuis l'arrivée de Java.

« Java s'est développé pour devenir aujourd'hui l'une des technologies les plus importantes et les plus incontournables de notre industrie. Ceux qui ont choisi Java ont été maintes fois récompensés par l'amélioration constante de ses performances, de son évolutivité, de sa fiabilité, de sa compatibilité et de ses fonctionnalités, » déclare Georges Saab, Vice President of Development, Java Platform Group chez Oracle. « L'écosystème Java rassemble des quantités impressionnantes de bibliothèques, de frameworks et de ressources pour aider les programmeurs les plus novices aussi bien que les experts. Le développement de Java lui-même s'effectue de façon transparente dans le cadre de la communauté Open-JDK. Avec les investissements considérables réalisés par Oracle et d'autres acteurs de cette communauté, nous sommes impatients d'assister aux 20 prochaines années d'évolution et de croissance de Java. »

Suite au rachat de Sun, Java est passé sous l'égide d'Oracle. Sous la supervision de la firme, deux versions majeures de la plateforme ont vu le jour, Java 7 et Java 8.

Java 8 représente l'une des versions les plus importantes de la plateforme. Java 8 apporte la plus importante évolution du modèle de programmation de Java depuis son lancement. Comme nouveautés phares, on note les expressions lambdas, les méthodes par défaut, les interfaces fonctionnelles, Profiles ou encore Streams.

Actuellement, Oracle et la communauté travaillent activement sur la prochaine version de Java. La nouveauté la plus attendue de Java 9 est la modularité avec le projet Jigsaw, qui rendra sa taille

encore plus pertinente et évolutive sur une large gamme d'appareils. Jigsaw a entraîné de gros changements au JDK. Elle permettra de découper la bibliothèque d'exécution de base de Java en différents modules. Ainsi, une machine virtuelle Java (JVM) pourra fonctionner sans le support de certains packages de base.

Java 9 apportera également la console en ligne de commande Java Shell, un outil interactif permettant d'évaluer les bouts de code Java ; une nouvelle API HTTP-Client pour supporter HTTP/2 et les WebSockets ; un portage pour l'architecture ARM AArch64 sur Linux ; et de différentes mises à jour des API existantes ainsi que quelques améliorations importantes des performances.

D'après la feuille de route publiée par Oracle, Java 9 sortira en version stable le 22 septembre 2016. Le gel des fonctionnalités est prévu pour le 10 décembre 2015.

Pour commémorer les 20 ans de Java, Oracle Certification offre une remise de 20 % sur tous les examens de certification Java. Cette offre est disponible dans le monde entier jusqu'au 31 décembre 2015. Les candidats doivent fournir le code promotionnel Java20 lors de leur inscription.

Oracle ne s'est pas contenté de faire évoluer le langage. Il a également créé des sueurs froides dans la communauté, en poursuivant en justice Google pour violation de ses brevets Java dans Android et sa machine virtuelle Davilk. Oracle soutient que « Les API Java peuvent être protégées par le droit d'auteur ». Cette affaire représente une menace importante pour le secteur de la technologie.

Néanmoins, Java demeure le langage de programmation de choix de nombreux développeurs et compte évoluer désormais plus rapidement, à une fréquence d'une nouvelle version majeure tous les deux ans.

En tant que communauté de développeurs, souhaitons un heureux anniversaire à Java et célébrons les 20 ans de la plateforme.

Rétrospective des événements qui ont marqué les 20 années de Java : [lien 1](#)

Commentez la news de **Hinault Romaric** en ligne : [lien 2](#)



Club

Les dernières news du Club

Nouvelle mise à jour de la page cours C++, plus de 25 nouveaux tutoriels publiés et bien d'autres à venir

Chers membres du club,

Il y a eu pas mal de nouvelles ressources qui ont été publiées dernièrement sur la rubrique C++. Il s'agit notamment d'articles, tutoriels, billets de blogs, news, etc. C'est avec un grand plaisir que nous vous annonçons plus de 25 nouvelles ressources C++ publiées sur nos pages. Ces ressources concernent plusieurs domaines et outils du C++.

Ce résultat n'aurait pas été possible sans les efforts des rédacteurs de Developpez.com et des auteurs externes qui ont autorisé à publier et à traduire leurs travaux sur la rubrique C++ que nous remercions.

Les meilleurs cours et tutoriels C++ : [lien 3](#)

Ci-dessous je vous présente un récapitulatif des articles et tutoriels qui ont été publiés sur le portail C++ :

Général

Comment écrire un algorithme « standard » : [lien 4](#)

Combattre les fuites de mémoire et les erreurs C++11, un article de Kacper Kolodziej : [lien 5](#)

Introduction à la programmation d'applications modulaires en C++ : première partie, un article de Kacper Kolodziej : [lien 6](#)

Programmation d'applications modulaires en C++ : première partie, un article de Kacper Kolodziej : [lien 7](#)

Programmation d'applications modulaires en C++ : seconde partie, un article de Kacper Kolodziej : [lien 8](#)

Cinq mythes courants sur le C++ par Joly Loïc : [lien 9](#)

Le débogage d'une application : méthodes et exercices, un tutoriel de Alexandre Laurent : [lien 10](#)

Tutoriel : Les règles d'or de la programmation, par mogwai162 et Alexandre Laurent : [lien 11](#)

Programmation fonctionnelle en C++, un article de John Carmack traduit par Arzar : [lien 12](#)

Analyse statique de code, un article de John Carmack traduit par Arzar : [lien 13](#)

Vérification et envoi d'e-mail en C++ avec boost, par Neckara : [lien 14](#)

Utilisation conjointe de lambdas et fonction, un billet de blog de germinolegrand : [lien 15](#)

Développer en C ou C++ avec l'environnement Eclipse, un tutoriel de Laurent Barbareau : [lien 16](#)

Présentation de libclang / libtooling, une API pour manipuler du code source par David Come : [lien 17](#)

Compte rendu événement Windows App spécial 3D, Minko, Project Anarchy, Babylon.js et Oculus Rift à l'honneur : [lien 18](#)

OpenGL

Tutoriels sur la programmation OpenGL moderne par l'équipe de la rubrique Jeux : [lien 19](#)

Tutoriels sur la programmation OpenGL moderne (ogldev en Français) par l'équipe de la rubrique Jeux : [lien 20](#)

DirectX 11

Tutoriels sur la programmation avec DirectX 11 (rastertek en Français) par l'équipe de la rubrique Jeux : [lien 21](#)

Un avant goût de DirectX 11 : tout savoir sur la onzième version de l'API de Microsoft, un article de Grégory Massal : [lien 22](#)

Arduino

L'univers Arduino - partie II : réalisation d'un prototype à base d'Arduino, un tutoriel de f-leb : [lien 23](#)

L'univers Arduino - partie I : aperçu de la plateforme Arduino et ses cartes d'interface (shields), un tutoriel de f-leb : [lien 24](#)

Visual C++

Visual C++ 2015 : voyage à la découverte d'un nouveau monde : [lien 25](#)

GCC

Nouvelles optimisations du x86 dans les prochains GCC 5.0 - Partie 1 : Vectorisation des sections de load/store : [lien 26](#)

Nouvelles optimisations du x86 dans les prochains GCC 5.0 - Partie 2 : Amélioration du PIC (Position independent code) en mode 32 bits : [lien 27](#)

Avancés

Métaprogrammation et métafonctions en C++11 : [lien 28](#)

Plusieurs d'autres ressources sont en cours de rédaction au sein de notre équipe de même que plu-

sieurs projets de traductions. Vous aussi vous pouvez faire partie de notre équipe soit en proposant des tutoriels/articles, soit en rejoignant notre équipe de traduction ou bien d'autres types de contribution. N'hésitez pas à nous contacter.

*Commentez la news de **Francis Walter** en ligne : [lien 29](#)*

Page cours Android : nouvelle mise à jour

Bonjour à tous,

Une nouvelle mise à jour a été faite sur les pages cours d'Android : [lien 30](#).

Détail des nouvelles entrées dont les principaux thèmes des tutoriels ajoutés sont explicités entre parenthèses.

- Page cours Android (nouvelles entrées)
 - Développement d'application Android sous l'architecture x86
 - Les Bases du développement d'application sous Android

- Les animations sous Android
- Les interfaces graphiques
- etc.

La page cours Android compte désormais 124 articles, cours complets et tutoriels.

N'hésitez pas à proposer vos contributions ou à poster vos idées de tutoriels. Par ailleurs, n'hésitez pas à nous faire part de ressources sur le Net qui pourraient être passées sous le format de DVP.

L'équipe Android

*Commentez la news de **Feanorin** en ligne : [lien 31](#)*

Excel



Les derniers tutoriels et articles

Excel et l'enregistreur de macro

Enregistrer les actions répétitives dans un fichier Excel

Si vous répétez régulièrement les mêmes manipulations lors du traitement d'un fichier Excel, ce tutoriel sur l'enregistreur de macro pourrait vous intéresser.

1 Introduction

1.1 Utilité de l'enregistreur de macro

En utilisant Excel, il n'est pas rare de devoir réaliser régulièrement, parfois quotidiennement, des opérations répétitives laborieuses et porteuses de peu, voire de pas du tout, de plus-value. Qui n'a jamais pesté de devoir de façon récurrente sélectionner les x premières lignes d'une plage pour les transférer dans une autre feuille, en les mettant en italique sur fond rouge avec bordure violette à pois jaunes ?

La machine pourrait parfaitement remplacer l'humain pour ces tâches abrutissantes. Excel vous permet d'enregistrer les opérations récurrentes et répétitives, de manière à vous permettre de les exécuter de façon automatique. Cet outil, c'est **l'enregistreur de macro**.

Cet enregistreur va mémoriser vos actions dans Excel et va les transcrire en code informatique, exécutable à l'envi.

Si le moins que l'on puisse dire est que cet enregistreur ne remplace pas un bon programmeur, il permet à l'utilisateur peu formé aux langages informatiques, voire pas du tout, de créer des « macros » simples et réutilisables.

Il permet également au programmeur VBA chevronné de « se souvenir » rapidement d'un code dont il ne connaît plus bien la syntaxe.

1.2 Objectifs de ce tutoriel

Ce tutoriel a pour objectif de vous initier à l'utilisation de cet enregistreur, au travers d'un petit exemple simple ne demandant pas trop de connaissances en Excel et surtout, **ne demandant AUCUNE connaissance en langage de programmation**.

1.3 Version utilisée

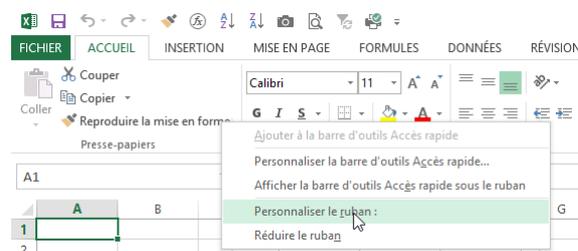
La version utilisée pour ce tutoriel est Excel 2013. Les adaptations de l'application sont identiques pour les versions 2007 et suivantes. Pour ce que nous verrons dans ce tutoriel, les illustrations sont similaires ou très approchantes. Les utilisateurs d'une version antérieure à X12007 pourront adapter le contenu de ce cours avec assez peu d'efforts.

Les forums sont bien évidemment là pour vous permettre de compléter votre apprentissage.

1.4 Préparation de l'application

Enregistrer une macro, c'est réaliser un acte de **développement d'application**. Certes, vous n'allez pas vous lancer dans la création d'un logiciel grand public, l'enregistreur en tant que tel ne vous en donne pas les moyens. Mais comme vous allez *développer*, il vous faut des outils de développeur.

Vous trouverez ces outils sur l'onglet [Développeur] du ruban, invisible lors de l'installation d'Excel. Pour le rendre accessible, cliquez droit sur le ruban puis Personnaliser le ruban. Dans la fenêtre des options qui s'ouvre, cochez **Onglet Développeur** dans le panneau de droite.





Le ruban [Développeur] apparaît, normalement à droite de l'onglet [Affichage].



2 Prise en main

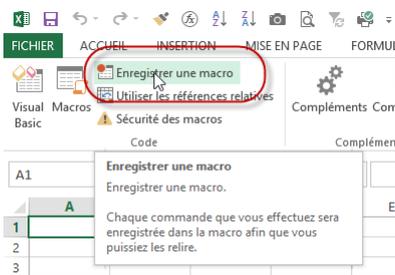
2.1 Enregistrer une première macro

Votre première macro va simplement consister à :

- colorer le fond d'une sélection en rouge ;
- placer un cadre +/- épais autour de la sélection ;
- placer des lignes en pointillé sur les bords intérieurs de la sélection.

Créez un nouveau classeur et sélectionnez une plage sur la feuille active.

Lorsque cette mise en place est effectuée, vous allez démarrer l'enregistreur via l'onglet [DEVELOPPEUR]>Code>Enregistrer une macro.



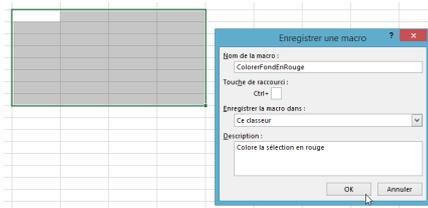
Avant d'enregistrer une nouvelle macro, vous devez :

- lui donner un nom ;
- détailler brièvement ce qu'elle fait (facultatif) ;
- lui attribuer un raccourci (facultatif) ;
- spécifier le classeur dans lequel elle sera enregistrée. Dans le cas qui vous occupe, choisissez **Ce classeur**.

L'emplacement de l'onglet peut différer selon votre personnalisation du ruban. Vous pouvez également afficher l'onglet [DEVELOPPEUR] via le menu Backstage [FICHIER]>Options>Personnalisation du ruban. Le ruban illustré peut différer chez vous, mais ce n'est pas très important. Dans le cadre de ce tutoriel, nous n'utiliserons que le premier groupe d'actions nommé **Code**.



Notez qu'il n'est pas nécessaire que le ruban soit actif pour pouvoir démarrer et stopper l'enregistreur (voir plus bas dans le tutoriel). Cependant, je conseille d'afficher le ruban, car il permet notamment de placer un bouton sur la feuille pour activer facilement une macro.



Ne cliquez pas sur OK trop vite. Tout ce que vous ferez dans Excel (enfin, presque tout) sera enregistré dans la macro. Il faut donc faire TOUT ce que vous voulez enregistrer, mais UNIQUEMENT ce que vous voulez enregistrer. Réfléchissez bien aux données que vous saisissez dans cette boîte de dialogue, car il sera malaisé, du moins lors de vos premiers pas, de modifier ces informations lorsque la macro sera enregistrée.



Observez rapidement les zones de cette boîte de dialogue :

- Nom de la macro : donnez un nom explicite à votre macro, surtout si vous en créez plusieurs au sein d'un classeur. Respectez les règles suivantes. Le nom :
 - commence toujours par une lettre,
 - ne contient pas d'espaces (utilisez le _ pour matérialiser l'espace),
 - ne contient que des lettres et des chiffres, si possible sans caractères accentués (ils sont admis, mais déconseillés) ;

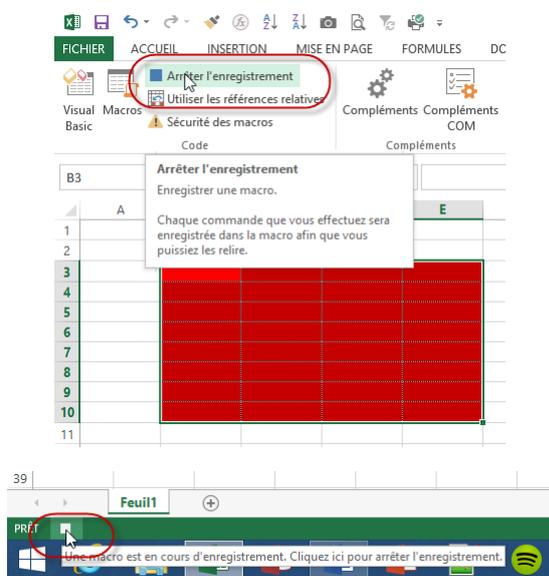
- Touche de raccourci : à utiliser si vous souhaitez pouvoir lancer la macro *via* un raccourci. N'utilisez pas un raccourci générique d'Excel !
- Enregistrer la macro dans : vous avez plusieurs choix possibles :
 - Ce classeur : la macro fait partie intégrante du classeur. Si vous transmettez ce classeur à un tiers, il pourra utiliser la macro,
 - Nouveau classeur : la macro sera enregistrée dans un nouveau classeur. Dans les faits, je n'ai jamais utilisé cette option,
 - Classeur de macros personnelles : la macro sera enregistrée dans un classeur spécial qui sera systématiquement ouvert en invisible à chaque démarrage d'Excel. Les macros seront donc disponibles dans tous vos classeurs. Le corollaire est que pour qu'un tiers puisse utiliser la macro, vous devrez lui transmettre votre classeur de macros personnelles ;
- Description : description succincte de ce que la macro est censée réaliser.

Lorsque nous sommes prêts, nous cliquons sur **OK** et c'est parti. Il suffit alors de :

- colorier en rouge la sélection des cellules ;
- choisir les bordures appropriées.

 Ne « bégayez » pas durant ces opérations. L'enregistreur de macro est actif et enregistre donc (presque) toutes les actions réalisées sur votre classeur.

Après avoir réalisé les étapes citées plus haut, vous stoppez la macro, soit par le ruban, soit *via* la barre d'état.



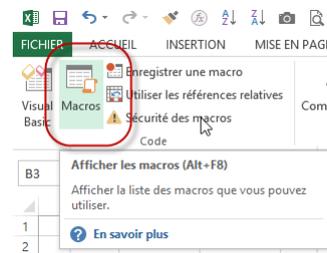
La macro est enregistrée et l'enregistreur s'arrête.

 Ces deux boutons sont des boutons bascule (ON/OFF). Vous pouvez donc démarrer ET stopper la macro *via* ces boutons. Notez également qu'il n'est pas nécessaire d'activer le ruban pour que ces boutons spécifiques soient disponibles. *Via* l'onglet d'affichage, vous avez également accès aux boutons de démarrage et d'arrêt de la macro, toujours sans qu'il soit nécessaire que le ruban **Développeur** soit activé.

 Toutes les actions ne sont pas enregistrées dans la macro. Seules celles qui altèrent le classeur le sont. C'est ainsi que l'action de cliquer sur des onglets du ruban n'est pas enregistrée dans la macro.

2.2 Exécuter la macro enregistrée

Pour exécuter une macro disponible, vous pouvez aller sur l'onglet [DEVELOPPEUR]>Code>Macros.



Dans la fenêtre, il suffit de sélectionner la macro souhaitée, puis de cliquer sur **Exécuter**.

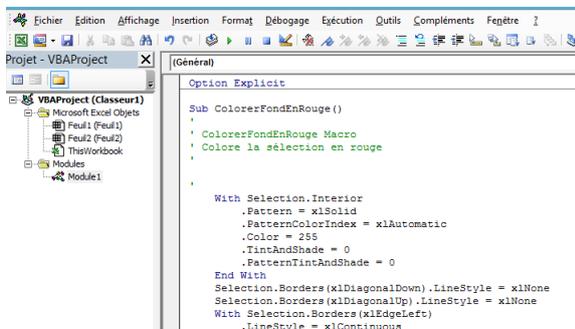
La nouvelle sélection est alors colorée en rouge et quadrillée selon les choix opérés durant l'enregistrement de la macro.



Vous remarquerez donc que la macro travaille avec la sélection active, puisque nous n'avons pas sélectionné une plage durant l'enregistrement de la macro.

2.3 Quel est le code qui a été enregistré

Pour visualiser le code qui a été enregistré, il suffit d'ouvrir la fenêtre des macros comme pour en exécuter une, et de cliquer sur **Modifier**. L'éditeur de code (VBE pour Visual Basic Editor) s'ouvre et vous pouvez admirer le code que « vous » avez enregistré.



Plus loin dans le tutoriel, nous reviendrons sur le code produit par l'enregistreur.

2.4 Enregistrement du classeur

Pour conserver le code enregistré et donc pouvoir jouer à nouveau la macro, il faut enregistrer le classeur. Le code associé à celui-ci sera enregistré en même temps, puisqu'il fait partie intégrante du classeur.

Jusqu'à la version 2003 comprise, un classeur contenait ou non des macros, et ce n'est qu'à l'ouverture dudit classeur, et pour autant que les sécurités soient mises en place, que l'on savait s'il contenait des macros incorporées ou non.

Depuis la version 2007, Microsoft a différencié les classeurs selon qu'ils contiennent ou non des macros. Les classeurs XLSX **ne contiennent jamais de macros**. Les classeurs XLSM **peuvent contenir des macros**. Les icônes sont différentes, ainsi

3 Un exemple concret

3.1 Introduction

Après cette prise en main rapide de l'enregistreur, vous allez maintenant comprendre comment Excel sélectionne les cellules et les feuilles durant l'enregistrement de la macro. Vous pourrez alors créer des macros réellement utiles, dont la complexité augmentera de quelques crans par rapport à la macro de coloriage.

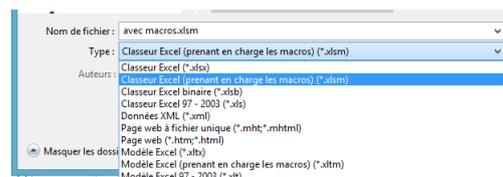
3.2 Cahier des charges

Vous devez saisir les informations de personnes participant à des formations. Il vous faut donc enregistrer les données personnelles des participants,

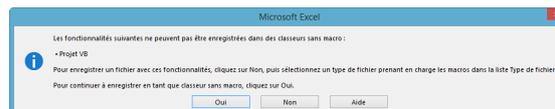
que les infobulles s'affichant dans l'explorateur de fichiers de Windows.



Vous devez choisir un format qui supporte les macros pour que les macros soient enregistrées avec le fichier.



Si vous choisissez un format n'acceptant pas les macros pour enregistrer un fichier qui en contient, Excel supprimera simplement les macros. Vous avez bien lu « supprimera », ce qui signifie que vous les perdrez **définitivement**. Au vu des conséquences, Excel vous affichera un message explicite avant l'enregistrement.



2.5 Conclusions de cette prise en main

Vous venez de découvrir l'enregistreur de macro d'Excel. Vous savez maintenant créer vos propres macros simples à l'aide de cet outil. Vous savez également où enregistrer vos macros, et vous avez découvert l'envers du décor en allant voir du côté du VBE pour découvrir le code créé par l'enregistreur.

Vous savez également enregistrer votre classeur contenant des macros de manière à pouvoir les réutiliser par la suite.

ainsi que la formation à laquelle ils s'inscrivent et la date de la session.

Vous disposez pour cela d'un classeur contenant une feuille de saisie de données et d'une feuille dans laquelle stocker vos données. Pour faciliter la saisie, la zone de saisie est mise en forme. Après la saisie, vous souhaitez transférer les données du « formulaire » vers la plage de données. Les cellules à transférer sont plus ou moins nombreuses, la saisie de données est fréquente et la personne qui effectue le travail voudrait automatiser le transfert des données de la feuille formulaire vers la plage de données.

La bonne nouvelle est que l'enregistreur de macro va vous aider à automatiser ce travail. Vous al-

lez également vous apercevoir qu'une bonne maîtrise des outils Excel facilitera la création de vos macros et simplifiera celles-ci, parfois de manière substantielle.



Je sais qu'Excel n'est pas l'outil le plus approprié pour réaliser ces opérations, et qu'il serait professionnellement plus judicieux d'utiliser des outils tels qu'Access pour réaliser la saisie et le stockage de données. Mais dans la mesure où tout le monde ne dispose pas d'Access, Excel peut être une solution intéressante.

3.3 Mise en place du classeur

L'enregistreur de macro retenant tout ce qui est réalisé durant l'enregistrement, il convient que le classeur soit préparé et que la séquence à enregistrer soit bien répétée, de façon à n'enregistrer que ce qui doit l'être.

3.3.a Création des feuilles

3.3.a.a Feuille de saisie

La feuille de saisie est préparée de manière à recevoir les données saisies. Pour ce tutoriel, vous vous baserez sur l'illustration suivante, dans laquelle la zone de saisie est colorée en vert.

	A	B	C	D	E
1	Formulaire de saisie				
2					
3	Nom				
4	Prénom				
5	Adresse				
6	CP		Localité		
7	Email				
8					
9	Formation				
10	Date				
11					



Évitez comme la peste les cellules fusionnées dans ce genre de travail. Si vous voulez centrer sur plusieurs colonnes, utilisez la mise en forme adéquate. La fusion de cellules ne peut être utilisée que pour la mise en forme esthétique, mais doit être évitée à tout prix pour les cellules « de travail ».

3.3.a.b Feuille de la table des données

Naturellement, la plage de données reprendra les mêmes données, mais en colonnes.

3.4 Scénarisation de la séquence

Lors de l'utilisation de l'enregistreur de macro, il est impératif de bien scénariser ce que vous allez devoir réaliser durant l'enregistrement. Pour rappel, l'enregistreur enregistre tout ce que vous faites. Si vous vous trompez durant l'enregistrement, vous devrez soit aller trifouiller dans le code pour nettoyer les actions inutiles, soit enregistrer à nouveau la macro.

Il importe donc de rédiger un « road-book » de ce que vous allez effectuer, et de penser à tous les cas de figure.

Par exemple, si la cellule B3 du formulaire est active au moment de démarrer l'enregistreur, Excel n'enregistrera pas qu'il faut la sélectionner. Dès lors, si vous démarrez l'exécution de la macro alors qu'une autre cellule est active, vous transfèrerez de mauvaises données dans la table.



Une bonne habitude est donc de se mettre « hors contexte » avant de démarrer l'enregistreur. Dès lors, on notera dans le road-book d'activer une autre cellule de la feuille, puis on sélectionnera éventuellement une autre feuille du classeur, dans le cas où la macro pourrait démarrer avec une mauvaise feuille active, de manière à ce que l'enregistreur mémorise la sélection de la bonne feuille et de la bonne cellule.



Afin de faciliter l'apprentissage de l'utilisation de l'enregistreur et de faciliter la démarche, nous insérerons les nouvelles données **en début de tableau**. Dans la mesure où normalement, l'ordre des données dans la table ne peut influencer le traitement de celles-ci, ce n'est pas un problème en soi.

3.4.a Étapes du traitement

1. Avant de lancer l'enregistrement
 - (a) Remplissage de la fiche ;
 - (b) Sélection d'une cellule hors zone de saisie pour obliger Excel à se placer sur la bonne cellule en début de macro (dans ce cas précis, il n'est pas nécessaire de sélectionner une autre feuille).
2. Lancer l'enregistrement.

3. Sélectionner la feuille du tableau.
4. Se placer en A2 :H2.
5. Par clic droit, insérer et décaler vers le bas.
6. Sélectionner la feuille du formulaire.
7. Se placer sur la cellule B3.
8. Copier la cellule.
9. Sélectionner la feuille **Données**.
10. Se positionner en A2.
11. Coller la valeur en collage spécial valeur.
12. Sélectionner la feuille **Saisie**.
13. Sélectionner la cellule B4.
14. Répéter les opérations 4 à 9 pour chaque cellule du formulaire, en sélectionnant chaque fois la bonne cellule du formulaire et la bonne cellule de la feuille de données.
15. Sélectionner la feuille du formulaire.
16. Vider le formulaire
 - (a) Sélectionner les cellules du formulaire ;
 - (b) Vider les cellules du formulaire de saisie.
17. Sélectionner la cellule B3.
18. Stopper la macro ! (Ne **JAMAIS** oublier cette dernière ligne. Mettez-la en dernière position dès la création de votre road-book.)

C'est en répétant le scénario sur votre classeur que vous verrez si vous n'avez rien oublié. Par exemple :

- en pensant que la macro va être déclenchée au départ du formulaire, il faut que ce soit cette feuille qui soit sélectionnée avant de démarrer l'enregistrement ;
- si vous loupez les étapes 1 à 6, vous allez arriver avec le nom copié sur la feuille du tableau puis vous allez devoir insérer une nouvelle ligne, et vous « perdrez » donc votre nom copié.



Ce ne sont que les multiples répétitions qui vous permettront de bien reprendre **TOUTES** les étapes et **DANS LE BON ORDRE**.

3.5 Répétitions avant l'enregistrement

Après avoir rédigé votre road-book, vous allez répéter votre macro plusieurs fois, de manière à être certain que vous n'avez rien oublié et que vous êtes bien rôdé. Je le répète encore une fois pour que cela rentre bien, l'enregistreur mémoriserait tous vos faits et gestes dans Excel (enfin, presque tous). Il convient dès lors d'être certain de la pièce à jouer, sous peine de devoir recommencer l'enregistrement ou d'aller patauger dans le code pour le « nettoyer ».

3.6 Enregistrement

Lorsque vous êtes fin prêt, vous créez une nouvelle macro comme expliqué plus haut dans ce tutoriel, et vous vous lancez, en respectant scrupuleusement votre road-book. Normalement, suite à vos répétitions, il est nickel et reflète bien l'ensemble des actions que vous devez mener.

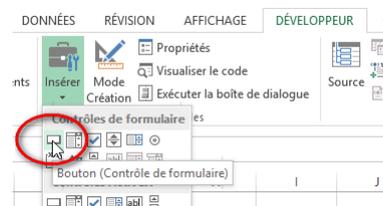
3.7 Utilisation de la macro

Si tout s'est bien passé, que vous n'avez pas bégayé durant l'enregistrement, vous disposez maintenant d'une macro qui transfère les données du formulaire vers une « table de données », puis qui nettoie le formulaire et se positionne pour une nouvelle saisie. Vous remarquerez que les saisies repoussent les lignes existantes vers le bas, puisque nous insérons toujours une ligne en première ligne de la plage de données. Dans la mesure où l'ordre de saisie n'a normalement que peu d'importance, ce n'est pas bien dramatique.

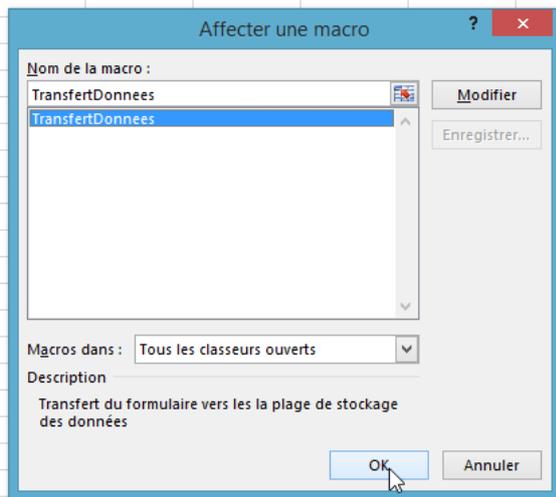
	A	B	C	D	E	F	G	H
1	Nom	Prénom	Adresse	CP	Localité	Email	Formation	Date
2	Tartempion	Jean-Yves	Avenue des	5000	Namur	probleme@Débogage VB		13-06-15
3	Fauconnier	Pierre	Rue du progr	1000	Bruxelles	pierre@exc VBA pour le p		23-06-15
4	Dupont	Jean	Rue de l'Info	75000	Paris	jean@dupor XL Enregistrei		16-04-15

3.8 Bouton de lancement de la macro

Ce serait un peu plus « fun » si vous disposiez d'un bouton de lancement sur la feuille de formulaire, de manière à pouvoir cliquer dessus pour lancer la macro. L'onglet [Développeur] expose plusieurs contrôles à placer sur une feuille, dont un bouton de commande.

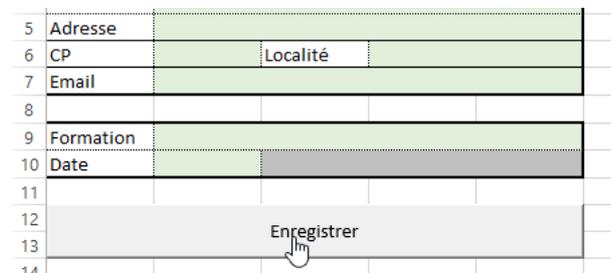


Sélectionnez **Bouton (Contrôle de formulaire)** et placez-le sur la feuille du formulaire. Ne vous tracassez pas trop de sa taille ni de sa position, vous modifieriez cela par la suite. Dès le bouton placé, une fenêtre s'ouvre pour vous permettre d'affecter une macro au clic du bouton. Sélectionnez votre macro et validez.



Par clic droit sur le bouton, modifiez son texte, déplacez-le et donnez-lui les dimensions souhaitées.

Lorsque tout est prêt, cliquez sur une cellule de la feuille. En survolant le bouton avec la souris, vous verrez une main indiquant que vous pouvez cliquer. Il suffit de remplir le formulaire, puis de cliquer sur le bouton pour que la macro s'exécute.

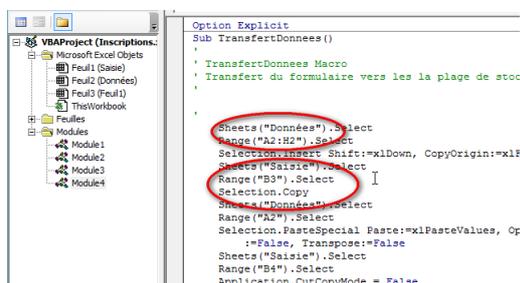


4 Limites de l'enregistreur de macro

4.1 Structure du classeur figée

Après avoir réalisé cette macro déjà plus volumineuse - elle fait un « vrai » travail réutilisable -, il est nécessaire de bien baliser ce que fait l'enregistreur de macro.

Un simple regard sur le code vous permettra de comprendre une des limites de l'enregistreur :



Les noms de feuille sont inscrits « en dur » dans la macro. Normalement, cela ne doit jamais être ainsi dans un code informatique professionnel. Cette technique s'appelle le *hard coding* et est normalement à proscrire. En effet, la simple modification du nom d'une des feuilles rendra la macro **inutilisable**. Toutefois, dans le cadre de ce tutoriel, nous devrons nous en contenter.

Le *hard coding* continue avec les adresses des cellules en dur. Si vous insérez des lignes avant la ligne 3 de la feuille du formulaire, ou si vous insérez des colonnes avant la dernière colonne utilisée de la feuille des données, votre macro fonctionnera, mais ne récupérera plus les bonnes données et/ou ne les copiera pas dans les bonnes colonnes. En effet, un déplacement de cellules en Excel n'adapte pas le code VBA.



En clair, cela veut dire que toute modification structurelle de votre classeur vous obligera à modifier, voire réécrire entièrement la macro.

4.2 Code inutile

L'enregistreur enregistre tout ce que vous réalisez, il copie donc les actions de sélection tant pour les feuilles que pour les cellules. Une programmation professionnelle réaliserait les mêmes actions en beaucoup moins de lignes.

4.3 Code redondant

Si vous souhaitez créer une macro qui réalise x fois la même opération, le code doit être écrit x fois dans la macro. Si vous voulez l'exécuter par la suite x-y fois ou x+y fois, vous devrez adapter la macro.

En effet, l'enregistreur de macro ne permet pas de manipuler les variables. Si l'on veut qu'une partie de la macro, voire tout le code, s'exécute un certain nombre de fois, il faudra adapter le code « à la main », ce qui dépasse le cadre de ce tutoriel.

4.4 Annulation impossible



Une macro exécutée **NE PEUT PAS ÊTRE ANNULÉE!!**

Si vous exécutez une macro, vous ne pourrez pas en annuler les effets avec la touche d'annulation (Ctrl+Z) et la liste des actions annulables sera remise à zéro. Ne perdez **jamais** cela de vue.

Excel ne saurait évidemment pas mémoriser ce que vous réalisez avec la macro. Il ne saurait donc pas le détricoter. Et puisque votre macro aura placé votre classeur dans un état qu'il n'a pas mémorisé, il ne saurait pas plus vous proposer d'annuler des actions réalisées avant l'exécution de la macro.

Ce point est trop souvent oublié par les programmeurs, c'est pourquoi j'insiste ici sur cet état de fait.

5 Déplacements absolus et relatifs

Malgré ces limites, Excel offre un peu plus de souplesse dans l'enregistrement du code. Vous avez vu plus haut que les adresses sont inscrites en dur dans le code. Par exemple, si je décale mon formulaire d'une ligne vers le bas, je vais devoir adapter toutes les lignes qui concernent la sélection des cellules de mon formulaire. De même, si je décale la zone de données de la feuille de données, la macro ne s'exécutera plus comme souhaité.

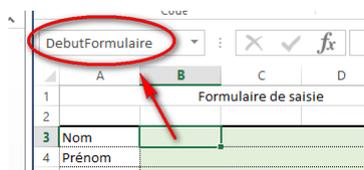
Dans ce qui va suivre, je vous propose d'abord d'utiliser quelques astuces pour contourner ce problème tout en ne modifiant pas le code « à la main ». Cela ajoutera un peu de souplesse à la macro.

5.1 Utilisation de cellules nommées

Si l'on nomme une plage (mono ou multicellulaire) et qu'on la déplace, Excel adapte évidemment l'adresse de la plage. Vous pourrez utiliser cette astuce en nommant la première cellule du formulaire et en vous arrangeant pour qu'Excel mémorise que vous la sélectionnez par son nom plutôt que par son adresse.

5.1.a Nommer la cellule

Pour nommer une cellule, il suffit de la sélectionner puis, dans la zone des noms (à gauche de la barre de formules), vous saisissez le nom souhaité et vous validez par ENTER (c'est souligné, OK ?).



Vous allez pouvoir sélectionner cette cellule plutôt que sélectionner B3. En effet, Excel offre depuis toujours (?) la possibilité d'atteindre une cellule, soit par son adresse, soit par son nom. La façon la plus simple de procéder consiste à saisir l'adresse (ou le nom) de la cellule souhaitée dans la zone des noms. Excel activera directement la cellule ou la plage renseignée. Vous pouvez également dérouler la liste des noms et sélectionner le nom de la plage que vous souhaitez atteindre. On pourra donc enregistrer l'activation de la cellule nommée *DebutFormulaire* plutôt que l'activation de la cellule B3. Si l'on insère des lignes ou des colonnes en haut et à gauche du formulaire, le déplacement vers *DebutFormulaire* activera bien la bonne cellule.

Vous pourrez procéder de la même manière pour nommer la première cellule d'entête de la zone de données, par exemple *DebutDonnees*. De cette manière, si la plage de données était déplacée, la macro pourrait continuer à fonctionner normalement.

Et tant qu'à faire, vous pouvez également nommer toutes les cellules de données du formulaire, de manière à pouvoir les sélectionner d'un seul coup pour les vider. Nommez cette plage *Formulaire*, par exemple.

5.2 Déplacements relatifs.

Excel permet d'enregistrer les déplacements relatifs. Dans la macro d'enregistrement des données du formulaire, nous étions en B3 pour copier le nom, puis nous avons voulu nous déplacer en B4 pour copier le prénom.

```
selection.insert shift:=x
Sheets("Saisie").Select
Range("B3").Select
Selection.Copy
Sheets("Donnees").Select
Range("A2").Select
Selection.PasteSpecial Pa
:=False, Transpose:=F
Sheets("Saisie").Select
Range("B4").Select
Application.CutCopyMode =
```

La ligne de code enregistrée est bien un déplacement en B4, ce qui fait que où que nous soyons dans la feuille avant l'exécution de la ligne, nous serons en B4 juste après. Mais nous pouvons également nous déplacer une ligne en dessous de la cellule *DebutFormulaire*. Dès lors, même si nous déplaçons le formulaire, nous nous déplacerons toujours vers les bonnes cellules.

De même, pour prévenir le déplacement de la zone de données dans laquelle vous stockerez les informations, vous pouvez nommer la première cellule d'entête de la zone de données, par exemple *DebutDonnees*. Vous pourrez alors l'atteindre en sélectionnant dans la zone des noms, puis vous déplacer d'une cellule vers le bas avant d'insérer une nouvelle ligne.

Pour effectuer un déplacement relatif plutôt qu'absolu, il faut basculer l'enregistreur en mode relatif. Vous verrez comment dans les paragraphes suivants.

5.3 Nouveau road-book

Notre road-book doit être modifié en conséquence. En rouge, les lignes ajoutées ou modifiées.

1. Avant de lancer l'enregistrement
 - (a) Remplissage de la fiche ;
 - (b) Sélection d'une cellule hors zone de saisie pour obliger Excel à se placer sur la bonne cellule en début de macro (dans ce cas précis, il n'est pas nécessaire de sélectionner une autre feuille).
2. **PASSER EN MODE RELATIF.**
3. Lancer l'enregistrement.
4. **Sélectionner la cellule *DebutDonnees* (via la liste déroulante de la zone des noms).**

5. Se déplacer d'une cellule vers le bas.
6. Sélectionner toutes les cellules de la ligne de données.
7. Par clic droit, insérer et décaler vers le bas.
8. Sélectionner la première cellule de la ligne (nous sommes en relatif, nous sélectionnons donc la cellule active (B2 si nous n'avons pas déplacé le formulaire).
9. Se placer sur la cellule *DebutFormulaire* (via la zone des noms).
10. Copier la cellule.
11. Sélectionner la feuille **Données**.
12. Coller la valeur en collage spécial valeur.
13. Se déplacer d'une cellule vers la droite.
14. Sélectionner la feuille **Saisie**.
15. Sélectionner la cellule B4.
16. Répéter les opérations 4 à 9 pour chaque cellule du formulaire, en sélectionnant chaque fois la bonne cellule du formulaire et la bonne cellule de la feuille de données (comme nous sommes en déplacement relatif, Excel mémorise le déplacement d'une cellule à l'autre et non l'adresse absolue de la cellule.
17. Sélectionner la plage nommée *Formulaire*.
18. Vider la plage avec DELETE.
19. Sélectionner la cellule B3.
20. Stopper la macro! (Ne **JAMAIS** oublier cette dernière ligne. Mettez-la en dernière position dès la création de votre road-book.)
21. Repassez en mode déplacement absolu grâce au bouton de type de sélection.

Réalisez une ou deux fois ces opérations avant d'enregistrer une nouvelle macro. Si vous souhaitez que le bouton de commande active cette nouvelle macro, vous le faites par clic droit sur le bouton, puis **Affecter une macro** et vous choisissez votre nouvelle macro.

Notez que le code est modifié pour tenir compte des déplacements relatifs. Notez également que les déplacements ne sont pas relatifs lorsque vous sélectionnez des zones nommées. Pour bien comprendre la différence entre les déplacements absolus et relatifs, comparez les lignes surlignées avec celles du point V.B.

```
Sub Macro10 ()
    Macro10 Macro
    Application.Goto Reference:="DebutDonnees"
    ActiveCell.Offset(1, 0).Range("A1:H1").Select
    Selection.Insert Shift:=xlDown, CopyOrigin:=xlFormatFromLeftOrAbove
    ActiveCell.Select
    Application.Goto Reference:="DebutFormulaire"
    Selection.Copy
    Sheets("Donnees").Select
    Selection.PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
    :=False, Transpose:=False
    ActiveCell.Offset(0, 1).Range("A1").Select
    Sheets("Saisie").Select
    ActiveCell.Offset(1, 0).Range("A1").Select
    Application.CutCopyMode = False
    Selection.Copy
```

Vous pouvez alors tester que tout est opérationnel en déplaçant la zone du formulaire et/ou la zone de données.



Déplacez bien chaque bloc EN ENTIER, car les déplacements à l'intérieur des blocs ne permettent pas d'insérer des lignes et/ou des colonnes à l'intérieur des zones de formulaire ou de données.

5.4 Notes complémentaires sur les déplacements

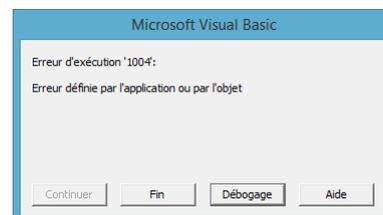
5.4.a Déplacements absolus et relatifs au sein d'une même macro

Au sein d'une même macro, vous pouvez alterner entre déplacement relatif et déplacement absolu. Avant de sélectionner une cellule/plage non nommée, basculez le mode en fonction de votre besoin.

5.4.b Ne sortez pas des limites

Vous pouvez générer une erreur d'exécution lorsqu'un déplacement relatif vous fait sortir des limites de la feuille.

Vous êtes en B10 et vous enregistrez un code qui vous fait remonter en relatif en A1 (vous remontez donc de 9 lignes et vous vous décalez d'une colonne à gauche). Si vous exécutez ce code alors que vous êtes en A5 par exemple, Excel va vouloir remonter de 9 lignes et se déplacer d'une colonne à gauche et va donc être hors limites, avec un beau plantage à la clé.



```
Sub Macro13 ()
    Macro13 Macro
    ActiveCell.Offset(-9, -1).Range("A1").Select
End Sub
```

6 Conclusions

Après avoir lu et mis en pratique ce tutoriel, il vous reste à pratiquer et encore pratiquer pour créer des automatisations grâce à l'enregistreur de macro. Gardez toutefois les points suivants en tête lorsque vous déciderez d'automatiser une tâche dans Excel grâce à l'enregistreur :

- voyez petit. L'enregistreur n'est pas là pour créer une application robuste et professionnelle ;
- rédigez et testez votre road-book ;
- utilisez les astuces Excel telles que les cellules/plages nommées. Elles simplifieront vos actions et vos macros. L'astuce d'une plage nommée permet parfois d'éviter de sélectionner une feuille (dont le nom est écrit en dur dans la macro) qui ne serait pas reconnue après renommage ;
- réfléchissez bien avant de modifier la structure des feuilles et plages de données concernées par les macros écrites à l'aide de l'enregistreur et partez du principe que les structures créées ne peuvent pas être modifiées (en ce compris le nom des feuilles sélectionnées).

Retrouvez l'article de **Pierre Fauconnier** en ligne : [lien 32](#)



Access

Les derniers tutoriels et articles

Simulation de process industriel

Simulation de process industriel.
Niveau requis : confirmé.

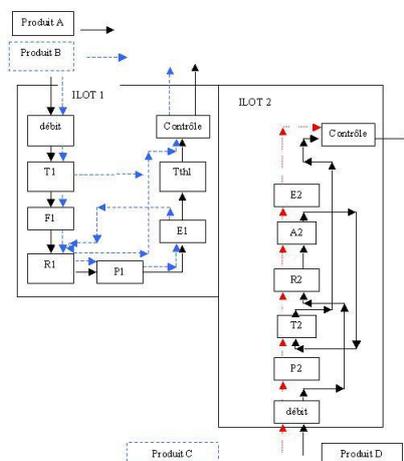
1 Introduction

Les process industriels impliquent souvent des calculs complexes et nécessitent donc un peu d'expérience pour appréhender le code mis en œuvre. Le but n'étant pas d'alourdir le propos avec des for-

mules trop compliquées, je vais donc essayer de décrire un exemple simple de simulation d'un process industriel dans *Ms Access*, suivant une gamme de fabrication.

2 Contexte

Dans l'industrie, les machines peuvent être regroupées en ensembles autonomes de production spécialisés par type de produits qui utilisent les mêmes machines : ces ensembles s'appellent des îlots. Les flux de produits (matières) peuvent utiliser les postes de travail de l'îlot dans un ordre différent suivant leur gamme de fabrication, alors que dans une ligne de fabrication l'ordre est impératif.



Description d'un îlot de machines



Pour simplifier, on va ici considérer une gamme de fabrication composée de machines placées en série et négliger les durées de transition entre machines.

3 Description du principe

L'objectif est d'obtenir, à partir de formules mathématiques, le nombre de pièces en attente sur une machine quelconque ainsi que le total des pièces déjà usinées à un instant donné. Nous injecterons ces fonctions traduites en VBA dans des requêtes pour

afficher et représenter la progression des pièces sur les différentes machines.

3.1 Formules mathématiques de base mises en oeuvre

```
1 duree_fabrication = (n-1)*duree_max + duree_total
```

- duree_max : durée de passage la plus longue entre les machines.
- duree_total : durée totale de passage sur les machines.

On en déduit :

```
1 nbre_fabrique = ((d - duree_total) \ duree_max) + 1
```



C'est cette formule mathématique qui va nous servir de base pour nos calculs dans nos requêtes.

3.2 Formules mathématiques généralisées

```
1 duree_passage(i) = (n-1)*duree_max(i) + duree_total(i)
```

- duree_max(i) : durée de passage la plus longue entre les machines d'indice 1 à i.
- duree_total(i) : durée totale de passage sur les i machines.

En inversant la formule on obtient :

```
1 nbre_passe(i) = ((d - duree_total(i)) \ duree_max(i)) + 1
```



Nous utilisons ici la division entière, notée « \ » en VBA, pour obtenir un nombre entier de pièces.

On en déduit enfin :

```
1 nbre_attente(i) = (((d - duree_total(i-1)) \ duree_max(i-1)) + 1) - (((d - duree_total(i)) \ duree_max(i)) + 1)
```

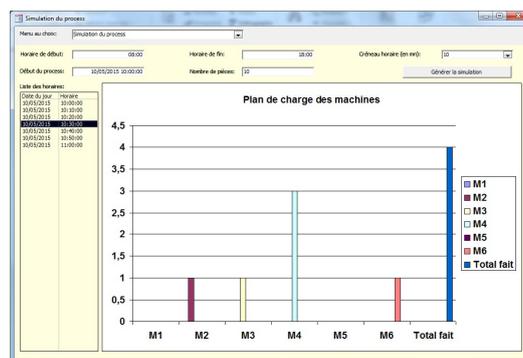
- duree_max(i-1) : durée de passage la plus longue entre les machines d'indice 1 à (i-1);
- duree_total(i-1) : durée totale de passage sur l'ensemble des (i-1) machines;
- duree_max(i) : durée de passage la plus longue entre les machines d'indice 1 à i;
- duree_total(i) : durée totale de passage sur les i machines.



Cette dernière formule va nous permettre de connaître, à un instant donné, le nombre de pièces en attente sur une machine quelconque et surtout, le nombre total de pièces déjà terminées en bout de gamme.

4 Rendu final de la simulation du process

Voici le rendu du formulaire qui présente, pour chaque machine, le nombre de pièces en attente et donc la progression des pièces usinées.



Aperçu du formulaire de simulation



On constate que le programme permet de visionner, pour chaque tranche horaire, le plan de charge de toutes les machines.

Voici les objets nécessaires pour aboutir à ce résultat.

5 Les tables nécessaires

Pour sauvegarder les durées d'usinage sur chaque machine ainsi que les jours et tranches horaires intervenant dans la simulation, nous aurons besoin de trois tables :

5.1 La table T_Creneau

Cette table contient la liste des créneaux horaires accompagnés de leur indice.

Nom du champ	Type de données	Description
Indice	Numérique	Indice du créneau horaire
Creneau	Texte	Créneau horaire

5.2 La table T_Jour

Elle enregistre les indices des jours durant lesquels se déroule le processus industriel.

Nom du champ	Type de données	Description
Jour	Entier long	Numéro d'ordre du jour

5.3 La table T_Machine

Cette table sauvegarde les noms des machines et les durées de passage des pièces qui peuvent être paramétrés.

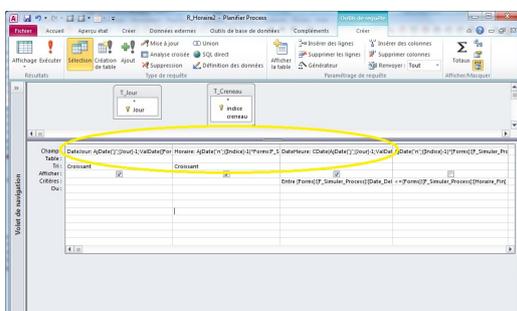
Nom du champ	Type de données	Description
Indice	Entier long	Numéro d'ordre de la machine
Machine	Texte	Nom de la machine
Duree	Entier long	Durée de passage sur la machine

6 Les requêtes pour la partie simulation

Afin de pouvoir afficher, pour chaque jour et chaque tranche horaire, le plan de charge simulé des machines, on a besoin de réaliser quelques requêtes.

6.1 La requête affichant les tranches horaires

Cette requête paramétrée réalise le produit cartésien entre la table **T_Creneau** et **T_Jour**, pour afficher toutes les tranches horaires du process :



Aperçu de la requête affichant les horaires



On constate que la requête comprend, dans la partie encadrée, des champs calculés que nous allons détailler.

6.1.a La formule donnant les jours

```
1 DateJour: AjDate("j";[Jour]-1;ValDate([Forms]![F_Simuler_Process]![Date_Debut]))
```

Forms ![F_Simuler_Process]![Date_Debut] : paramètre donnant la date et l'heure du début du process.

6.1.b La formule donnant les heures

```
1 Horaire: AjDate("n";([Indice]-1)*Forms!F_Simuler_Process!Creneau;Forms!F_Simuler_Process!Horaire_Debut)
```

Forms ![F_Simuler_Process]![Creneau] : paramètre donnant la durée des tranches horaires.

Forms ![F_Simuler_Process]![Horaire_Debut] : paramètre donnant l'heure du début du process.

6.1.c La formule donnant les dates et heures

Les dates et horaires pris en compte dans la simulation s'obtiennent en fonction du paramètre définissant le 1^{er} jour et le 1^{er} horaire :

```

1 DateHeure: CDate(AjDate("j";[Jour]-1;
  ValDate([Forms]![F_Simuler_Process
  ]![Date_Debut])) & " " & AjDate("n"
  ;([Indice]-1)*[Forms]![
  F_Simuler_Process]![Creneau];[Forms
  ]![F_Simuler_Process]![Horaire_Debut
  ]))

```

6.1.d Fonction évaluant la date et l'heure de fin du process

On évalue la date et l'heure de fin du processus à partir de la formule de base définie précédemment :

```

1 duree_fabrication = (n-1)*duree_max +
  duree_total

```

La fonction VBA mise en œuvre pour ce calcul va permettre de filtrer la requête sur le champ *date* et *heure* :

```

1 Public Function fin_process(ByVal nbre
  As Long, ByVal debut As Date, ByVal
  Horaire_Debut, ByVal Horaire_Fin,
  ByVal Creneau As Integer) As Date
2 ' date et heure de fin du process en
  fonction du nombre de pièces, de la
  date de début du process,
3 ' des heures de début et de fin dans la
  journée et du créneau horaire.
4 Dim rs As DAO.Recordset
5 Dim leSql As String
6 Dim d As Long, d1 As Long, d2 As Long
7 Dim dt As Date
8 Dim j As Integer
9 Dim m As Long
10
11 ' requête affichant la durée totale du
  process en se basant sur la formule
  (n-1)*duree_max + duree_total.
12 leSql = "SELECT " & (nbre - 1) & "*Max([
  Duree])+Sum([Duree]) AS durée " & _
13 "FROM T_Machine;"
14
15 ' le jeu d'enregistrement correspondant.
16 Set rs = CurrentDb.OpenRecordset(leSql,
  dbOpenForwardOnly)
17
18 ' la durée totale.
19 d = rs!durée
20
21 rs.Close
22 Set rs = Nothing
23
24 d1 = DateDiff("s", TimeValue(debut),
  Horaire_Fin)
25
26 If d <= d1 Then ' si compris dans l'
  intervalle.
27   fin_process = DateAdd("s", d,
  debut)
28
29 Else ' sinon on décompose la durée.
30
31   d = d - d1
32   dt = CDate(Format(debut + 1, "dd/
  mm/yyyy") & " " & Format(
  Horaire_Debut, "hh:nn"))
33   m = DateDiff("s", Horaire_Debut,
  Horaire_Fin)
34   j = d \ m
35   dt = DateAdd("d", j, dt)

```

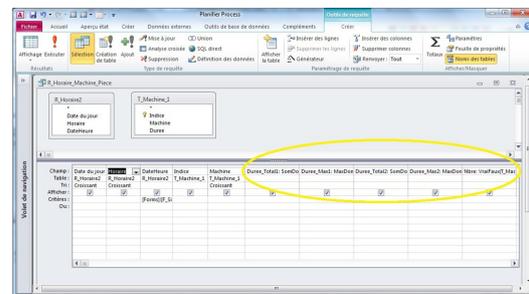
```

36 dt = DateAdd("s", d Mod m, dt)
37
38 fin_process = dt
39
40 End If
41
42 ' renvoie la date et l'heure de fin du
  process.
43 fin_process = DateAdd("s", (Creneau * 60
  ), fin_process)
44
45 End Function

```

6.2 La requête affichant la charge des machines

Cette requête paramétrée réalise le produit cartésien entre la requête précédente **R_Horaire2** et la table **T_Machine**, de manière à afficher toutes les tranches horaires du process et les quantités en attente sur chaque machine :



Aperçu de la requête finale



On remarque que la requête comprend dans la partie encadrée des champs calculés que nous allons détailler.

6.2.a Mise en œuvre de la formule généralisée

Nous avons en fait injecté dans la requête au niveau du champ **Nbre** la formule généralisée vue précédemment :

```

1 nbre_attente(i) = (((d - duree_total(i-1)
  )) \ duree_max(i-1)) + 1) - (((d -
  duree_total(i)) \ duree_max(i)) + 1)

```

À cet effet, nous avons évalué les différentes durées maximales et totales au moyen de fonctions de domaine.

6.2.a.a Durée totale (i-1)

```

1 Duree_Total1: SomDom("Duree";"T_Machine"
  ;"Indice <=" & ([T_Machine_1].[Indice
  ]-1) & " or (Indice=1)")

```



On choisit ici une fonction de domaine, dans la seconde version de l'exemple disponible en téléchargement nous utilisons une sous-requête à la place.

6.2.a.b Durée maximum (i-1)

```
1 Duree_Max1: MaxDom("Duree";"T_Machine";"
    Indice<=" & ([T_Machine_1].[Indice]-
    1) & " or ( Indice=1)")
```

6.2.a.c Durée totale (i)

```
1 Duree_Total2: SomDom("[Duree]";"
    T_Machine";"Indice<=" & [T_Machine_1
    ].[Indice])
```

6.2.a.d Durée maximum (i)

```
1 Duree_Max2: MaxDom("Duree";"T_Machine";
    "Indice<=" & [T_Machine_1].[Indice
    ])
```

Puis, on a inséré dans le champ *nbre* de la requête principale une fonction VBA qui est la traduction de la formule généralisée mentionnée plus haut.

6.2.a.e Nombre de pièces en attente à l'indice i

On a injecté la formule généralisée vue précédemment au niveau de la requête en utilisant une fonction VBA *fnc_nbre* :

```
1 Nbre: VraiFaux(T_Machine_1.Indice=12;
    fnc_Nbre(Forms!F_Simuler_Process!
    Date_Debut;[DateHeure];Forms!
    F_Simuler_Process!Horaire_Debut;
    Forms!F_Simuler_Process!Horaire_Fin
    ;[Duree_Total2];[Duree_Max2];Forms!
    F_Simuler_Process!Nbre_Pieces);(
    VraiFaux(T_Machine_1.Indice=1;Forms!
    F_Simuler_Process!Nbre_Pieces;
    fnc_Nbre(Forms!F_Simuler_Process!
    Date_Debut;[DateHeure];Forms!
    F_Simuler_Process!Horaire_Debut;
    Forms!F_Simuler_Process!Horaire_Fin
    ;[Duree_Total1];[Duree_Max1];Forms!
    F_Simuler_Process!Nbre_Pieces))) -
    fnc_Nbre(Forms!F_Simuler_Process!
    Date_Debut;[DateHeure];Forms!
    F_Simuler_Process!Horaire_Debut;
    Forms!F_Simuler_Process!Horaire_Fin
    ;[Duree_Total2];[Duree_Max2];Forms!
    F_Simuler_Process!Nbre_Pieces))
```



On utilise ici la fonction *VraiFaux* pour tester si on est en bout de gamme (l'indice égal à 12 est juste un indicateur de fin supérieur au nombre maximum de machines), dans ce cas, on calcule le nombre de pièces fabriquées. L'expression comprend également la fonction *fnc_Nbre* qui donne le nombre de pièces passées sur les i machines, après une durée d.

Les champs *durées* et *date/heure* définis précédemment :

- Duree_Max1 : durée maximum jusqu'à la machine d'indice (i-1);
- Duree_Total1 : durée totale jusqu'à la machine d'indice (i-1);
- Duree_Max2 : durée maximum jusqu'à la machine d'indice (i);
- Duree_Total2 : durée totale jusqu'à la machine d'indice (i);
- DateHeure : date et heure prises en compte pour le calcul.

Les paramètres de la simulation (début et fin du process) qui sont des zones de texte situées sur le formulaire simulateur du process :

- Forms!F_Simuler_Process!Date_Debut : paramètre indiquant la date et l'heure de début de la simulation;
- Forms!F_Simuler_Process!Horaire_Debut : paramètre indiquant l'heure de début du process;
- Forms!F_Simuler_Process!Horaire_Fin : paramètre indiquant l'heure de fin du process.

```
1 Public Function fnc_Nbre(ByVal debut As
    Date, ByVal fin As Date, ByVal
    Horaire_Debut As Date, ByVal
    Horaire_Fin As Date, ByVal
    duree_total As Integer, ByVal
    duree_max As Integer, ByVal nbre As
    Long) As Long
2 ' nombre de pièces passées sur la
    machine en fonction de la durée
    totale et de la durée max. jusqu'à
    cette machine.
3
4 Dim j As Integer
5 Dim d1 As Long, d2 As Long
6 Dim d As Long
7
8 j = DateDiff("d", debut, fin)
9
10 If j > 1 Then ' process sur plusieurs
    jours.
11     d = (j - 1) * DateDiff("s",
        Horaire_Debut, Horaire_Fin)
12     d1 = DateDiff("s", TimeValue(debut
        ), Horaire_Fin)
13     d2 = DateDiff("s", Horaire_Debut,
        TimeValue(fin))
14     d = d + d1 + d2
15 ElseIf (j = 1) Then ' simulation sur
    1 jour entier.
16     d1 = DateDiff("s", TimeValue(debut
        ), Horaire_Fin)
17     d2 = DateDiff("s", Horaire_Debut,
        TimeValue(fin))
18     d = d1 + d2
19 Else ' process se déroulant dans la
    journée.
20     d = DateDiff("s", TimeValue(debut)
        , TimeValue(fin))
21 End If
22
23 If (d >= duree_total) Then
```

```

24     fnc_Nbre = (d - duree_total) \
        duree_max + 1 ' formule mathé
        matique définie dans l'article
        .
25
26     If fnc_Nbre > nbre Then
27         fnc_Nbre = nbre
28     End If
29
30 Else
31     fnc_Nbre = 0
32 End If
33
34 End Function
    
```

6.2.a.f Date et heure de fin du process

On évalue la date et l'heure de fin du processus à partir de la formule de base définie précédemment :

```

1  duree_fabrication = (n-1)*duree_max +
    duree_total
    
```

La fonction VBA mise en œuvre pour ce calcul est la suivante :

```

1  Public Function fin_process(ByVal nbre
    As Long, ByVal debut As Date, ByVal
    Horaire_Debut, ByVal Horaire_Fin,
    ByVal Creneau As Integer) As Date
2  ' date et heure de fin du process en
    fonction du nombre de pièces, de la
    date de début du process,
3  ' des heures de début et de fin dans la
    journée et du créneau horaire.
4  Dim rs As DAO.Recordset
5  Dim leSql As String
6  Dim d As Long, d1 As Long, d2 As Long
7  Dim dt As Date
8  Dim j As Integer
9  Dim m As Long
10
11 ' requête affichant la durée totale du
    process en se basant sur la formule
    (n-1)*duree_max + duree_total.
    
```

```

12 leSql = "SELECT " & (nbre - 1) & "*Max([
    Duree])+Sum([Duree]) AS durée " & _
13 "FROM T_Machine;"
14
15 ' le jeu d'enregistrement correspondant.
16 Set rs = CurrentDb.OpenRecordset(leSql,
    dbOpenForwardOnly)
17
18 ' la durée totale.
19 d = rs!durée
20
21 rs.Close
22 Set rs = Nothing
23
24 d1 = DateDiff("s", TimeValue(debut),
    Horaire_Fin)
25
26 If d <= d1 Then ' si compris dans l'
    intervalle.
27     fin_process = DateAdd("s", d,
        debut)
28
29 Else ' sinon on décompose la durée.
30
31     d = d - d1
32     dt = CDate(Format(debut + 1, "dd/
        mm/yyyy") & " " & Format(
        Horaire_Debut, "hh:nn"))
33     m = DateDiff("s", Horaire_Debut,
        Horaire_Fin)
34     j = d \ m
35     dt = DateAdd("d", j, dt)
36     dt = DateAdd("s", d Mod m, dt)
37
38     fin_process = dt
39
40 End If
41
42 ' renvoie la date et l'heure de fin du
    process.
43 fin_process = DateAdd("s", (Creneau * 60
    ), fin_process)
44
45 End Function
    
```

7 Le formulaire de simulation

Nous allons décrire le formulaire qui présente la simulation du process sur les différentes machines de la gamme de fabrication.



Le programme permet de visionner pour chaque tranche horaire le plan de charge de toutes les machines.



Aperçu du formulaire de simulation

7.1 La zone de liste affichant les horaires

Sur la partie gauche du formulaire se trouve une zone de liste qui affiche l'ensemble des tranches horaires prises en compte dans la simulation du process. Elles sont sous la forme *Jour/Heure*.

7.2 Le graphique pour afficher la charge des machines

Sur la plus grande partie du formulaire se trouve le graphique qui affiche le plan de charge des machines dont voici le SQL :

```
1 PARAMETERS [Forms]![F_Simuler_Process]![  
  Nbre_Pieces] Value;  
2 TRANSFORM First(R_Horaire_Machine_Piece.  
  Nbre) AS First_Nbre  
3 SELECT R_Horaire_Machine_Piece.Machine  
  FROM R_Horaire_Machine_Piece  
4 GROUP BY R_Horaire_Machine_Piece.Machine  
5 PIVOT R_Horaire_Machine_Piece.Machine In  
  ("M1", "M2", "M3", "M4", "M5", "M6", "  
  Total fait");
```

8 Téléchargement



Pour faciliter les tests et assurer sa portabilité, la base exemple n'utilisera pas de composants externes à **Access** (**ActiveX** ou autres).

La base exemple est au format Access 2000 : [lien 33](#).
Retrouvez l'article de **Denis Hulo** en ligne : [lien 34](#)



JavaScript

Les blogs JavaScript

Comprendre la délégation d'événement en JavaScript

La délégation d'événement, qu'est-ce que c'est ?

La délégation d'événement est une technique assez courante en JavaScript qui consiste à poser des écouteurs d'événements non pas sur l'élément HTML ciblé, mais sur l'un de ses ancêtres dans le DOM.

Comment ça marche ?

Le concept essentiel pour comprendre cette technique est la notion de bouillonnement d'un événement. À de rares exceptions près, quasiment tous les événements bouillonnent. Si l'on se représente une page Web comme une feuille de papier sur laquelle on poserait d'autres feuilles de tailles différentes et placées à des endroits précis selon l'arborescence des balises et la mise en page CSS, on peut se représenter un événement (un clic sur une balise par exemple) comme étant intercepté par l'élément le plus au-dessus de l'empilement (et inversement, le plus profond dans l'arborescence du DOM). C'est donc cet élément qui va recevoir l'événement. Le bouillonnement est le mécanisme qui va faire « remonter » cet événement jusqu'au plus haut niveau de l'arborescence (la première feuille de papier) en passant par tous les éléments se trouvant à l'emplacement où l'événement a été déclenché, permettant ainsi à tous les gestionnaires d'événements associés de se déclencher.

En résumé, si l'on considère le code HTML suivant

```
1 <body>
2   <div>
3     <span>Cliquer ici</span>
4   </div>
5 </body>
```

et que vous placez un écouteur d'événement clic sur la div, cliquer sur le texte du span déclenchera l'événement de la div bien que ce span « cache » cette dernière. Ceci est possible grâce au bouillonnement.

Pour en revenir à notre délégation d'événement, nous pourrions savoir, avec l'objet Event associé à tout événement, quel élément HTML a réellement déclenché cet événement (Event.target) et si ce dernier est bien celui que nous souhaitons cibler.

À quoi cela peut-il servir ?

Il y a deux cas typiques d'utilisation de la délégation d'événement.

- Lorsque l'on souhaite associer des événements similaires à différents éléments. Par exemple, imaginons que l'on souhaite afficher un texte initialement masqué en cliquant sur des items d'une liste ordonnée. Plutôt que de définir autant d'événements que d'éléments dans la liste, on pourra n'en utiliser qu'un seul placé sur la balise

```
1 <ul id="maListe">
2   <li class="element" data-texte="foo">Item 1</li>
3   <li class="element" data-texte="bar">Item 2</li>
4   <li class="element" data-texte="baz">Item 3</li>
5   <li class="element" data-texte="toto">Item 4</li>
6   <li class="element" data-texte="titi">Item 5</li>
7   <li class="element" data-texte="tata">Item 6</li>
8   <li>Item 7</li>
9   <li>Item 8</li>
10  <li>Item 9</li>
11  <li class="element" data-texte="42">Item 10</li>
12 </ul>
```

```
1 document.getElementById('maListe').
  addEventListener('click', function(e)
2  ){
3    var initElem = e.target;
4    if(initElem.className != 'element'){
5      // Si l'élément n'est pas un de
6      // ceux à traiter
7      return;
8    }
9    alert(initElem.dataset.texte);
10  });
```

Voir l'exemple sur JSFiddle : [lien 35](#).

Dans ce code, on cherche à afficher le contenu de l'attribut data-texte pour tous les éléments ayant la classe element. On commence par récupérer l'élément ayant reçu l'événement. On vérifie ensuite si cet élément est bien du type que l'on cible, si ce n'est pas le cas, on stoppe l'exécution de la fonction, sinon, on affiche le message.

- Lorsque l'on veut prévoir des gestionnaires d'événements sur des éléments n'étant pas encore présents dans la page. Il est de plus en plus fréquent, avec les interfaces riches et AJAX, que le contenu d'une page évolue en fonction des actions de l'utilisateur. On peut donc avoir à gérer des événements sur des éléments qui n'existent pas au chargement de la page mais qui sont susceptibles d'y apparaître. Seulement, lorsque l'on déclare un gestionnaire, il ne peut s'appliquer qu'à des éléments effectivement présents au moment de la déclaration de l'écouteur : JavaScript n'est pas devin ni prédictif. Pour cela, on pourra facilement poser le gestionnaire sur un ancêtre connu (et existant) dans lequel seront insérés les futurs éléments.

```

1 <div id="parent"></div>
2 <button>Ajouter un élément</button>

1 document.getElementById('ajout').onclick
  = function(){
2   document.getElementById('parent').
    innerHTML = '<span id="enfant">É
    lément enfant ajouté
    dynamiquement</span>';
3 };
4 document.getElementById('parent').
  addEventListener('click', function(e
  ){
5   var initElem = e.target;
6   if(initElem.id == 'enfant'){
7     alert('Vous avez cliqué !');
8   }
9 });

```

Voir l'exemple sur JSFiddle : [lien 36](#).

Au moment de déclarer l'événement, l'élément enfant n'existe pas, pourtant, lorsqu'on l'insère dans la page et que l'on clique dessus, le message s'affiche.

Retrouvez le blog de *Didier Mouronval* en ligne : [lien 37](#)

Aller plus loin

Il est important, quand on utilise la délégation d'événement, de faire attention que si l'élément ciblé possède des éléments enfants, alors `Event.target` peut ne pas correspondre à celui ciblé, il faudra dans ce cas remonter l'arborescence jusqu'à l'élément sur lequel le gestionnaire est posé en testant à chaque palier si l'élément en cours est celui recherché.

```

1 var initElem = false, tmpElem = e.target
  ;
2 do{
3   if(tmpElem.id == 'id_recherche'){
4     initElem = tmpElem;
5   }
6 }
7 while(tmpElem = tmpElem.parentNode &&
  tmpElem != this);
8 if(initElem){
9   // tmpElem ne vaut pas false si on
  // entre dans cette condition
10  // donc l'élément recherché a été
  // trouvé
11  // on peut donc faire les traitement
  // voulus
12 }

```

Il est aussi à noter que la plupart des bibliothèques JavaScript permettent d'utiliser la délégation d'événement. Par exemple pour jQuery, la syntaxe

```

1 $('#elem1').on('click', '.elems2',
  callback);

```

permet de déléguer la gestion des événements clic sur les éléments ayant la classe CSS `elems2` sur l'élément dont l'identifiant est `elem1`. N'hésitez pas à indiquer dans les commentaires les différentes syntaxes pour les autres *frameworks*.

Les derniers tutoriels et articles

Onze idées fausses sur l'héritage en JavaScript

une traduction de l'article « Common Misconceptions About Inheritance in JavaScript » écrit par Eric Elliott

WAT? [wat] - interjection : le son que fait un programmeur quand quelque chose viole le principe du moindre étonnement en l'étonnant avec un comportement contre-intuitif.

```
> .1 + .2
0.30000000000000004
> WAT? OMG! STFU! STUPIDE JAVASCRIPT!!!
...
```

Par ailleurs, WAT? est également le son que je fais quand je discute avec beaucoup de développeurs JavaScript chevronnés qui ont négligé l'apprentissage des mécanismes de base de l'héritage par prototype : l'une des plus importantes innovations dans l'histoire de l'informatique et l'un des Deux Piliers du JavaScript.

D'après moi, ce serait comme un photographe professionnel qui aurait encore à apprendre le triangle d'exposition — la formule basique pour contrôler le style visuel d'une photographie. Simplement : *Si vous ne comprenez pas les prototypes, vous ne comprenez pas JavaScript.*

1 L'héritage par classe et l'héritage par prototype sont équivalents, c'est juste une préférence de style

Non.

L'héritage par classe et l'héritage par prototype sont **fondamentalement et sémantiquement distincts**.

Il y a des caractéristiques bien définies entre l'héritage par classe et l'héritage par prototype. Afin que la moindre phrase de cet article présente un sens, vous devez garder en tête ces points.

Avec l'héritage par classe, les instances héritent d'un schéma (la classe) et créent une relation entre classe et sous-classe. En d'autres termes, vous ne pouvez pas utiliser la classe de la même façon que vous utilisez l'instance. **Vous ne pouvez pas appeler les méthodes de l'instance sur la définition de la classe elle-même.** Vous devez d'abord créer une instance puis appeler les méthodes sur cette instance.

Avec l'héritage par prototype, les **instances héritent d'autres instances**. Utiliser des prototypes délégués (prendre un objet comme exemple pour être le prototype d'une instance) suit littéralement le

principe des objets liés à d'autres objets (**Objects Linking to Other Objects**, ou **OLOO**, comme l'appelle Kyle Simpson). Utiliser de l'**héritage par concaténation** revient à copier les propriétés du prototype vers la nouvelle instance.

C'est très important que vous compreniez ces différences. L'héritage par classe, en vertu de ces mécanismes, crée **une hiérarchie de classes** en plus de créer la sous-classe. De ces hiérarchies résulte un code ankylosé, difficile à modifier et **fragile** (facile à casser à cause des effets de bord lorsque vous modifiez les classes de base).

L'héritage par prototype ne crée pas nécessairement ce genre de hiérarchies. Je vous recommande de garder les chaînes de prototypes les moins longues possible. Il est facile de mettre à plat de nombreux prototypes ensemble pour former **un seul prototype délégué**.

En bref :

- une classe est un modèle, un schéma ;
- un prototype est une instance, un objet.

2 Les classes sont la bonne manière de créer des objets en JavaScript

Non.

Il y a plusieurs **bonnes façons** de créer des objets en JavaScript. La première et la plus commune est l'écriture littérale. Ça ressemble à ça (en ECMAScript 6, alias ES6) :

```

1 // ES6 / ES2015
2 let mouse = {
3   furColor: 'brown',
4   legs: 4,
5   tail: 'long, skinny',
6   describe () {
7     return 'A mouse with ${this.furColor}
8     } fur,
9     ${this.legs} legs, and a ${this.
10    tail} tail.';
11   }
12 };

```

Bien sûr, l'écriture littérale existe depuis bien plus longtemps que ES6, mais il manquait la syntaxe raccourcie pour décrire des méthodes, et vous deviez utiliser 'var' au lieu de 'let'. Oh, et la notation de template dans la chaîne de caractères de la fonction *describe* n'existe pas non plus en ECMAScript 5 (ES5).

Vous pouvez attacher des prototypes délégués avec `Object.create()` (une fonctionnalité ES5) :

```

1 let animal = {
2   animalType: 'animal',
3
4   describe () {
5     return "An ${this.animalType}, with
6     } ${this.furColor} fur,
7     ${this.legs} legs, and a ${this.
8     tail} tail.";
9   }
10 };
11
12 let mouse = Object.assign(Object.create(
13   animal), {
14   animalType: 'mouse',
15   furColor: 'brown',
16 });

```

3 On a besoin d'une fonction constructeur pour définir le comportement de l'instanciation et initialiser l'objet

Non.

N'importe quelle fonction peut créer et retourner des objets. Quand il ne s'agit pas d'une fonction constructeur, on l'appelle une fonction usine (**factory function**).

```

1 let animal = {
2   animalType: 'animal',
3
4   describe () {
5     return "An ${this.animalType} with $
6     } ${this.furColor} fur,
7     ${this.legs} legs, and a ${this.
8     tail} tail.";
9   }
10 };

```

```

13   legs: 4,
14   tail: 'long, skinny'
15 });

```

Décomposons un peu ce code. *animal* est un **prototype délégué**. *mouse* est une instance. Quand vous essayez d'accéder à une propriété de *mouse* qui est inexistante, le moteur JavaScript va chercher la propriété sur *animal* (le délégué).

Object.assign() est une fonctionnalité ES6 sous la tutelle de Rick Waldron qui était auparavant implémentée dans une petite douzaine de bibliothèques. Vous devez la connaître comme *\$.extend()* de jQuery ou *_.extend()* de Underscore. Lodash en a une version appelée *assign()*. Vous lui passez un objet de destination, et autant d'objets sources que vous voulez, séparés par des virgules. Cela copiera toutes les propriétés propres et énumérables en les assignant depuis les objets sources jusqu'à l'objet de destination. Si plusieurs propriétés sont en conflit, la version du dernier objet aura le dessus.

Object.create() est une fonctionnalité ES5 sous la tutelle de Douglas Crockford qui vous permet d'attacher des prototypes délégués sans devoir utiliser des constructeurs ou le mot-clé *new*.

Je vais passer outre l'exemple de la fonction constructeur parce que je ne peux pas le recommander. Je l'ai souvent vu utiliser abusivement, et causer beaucoup de dégâts : [lien 38](#). Il est à noter que beaucoup de gens intelligents sont en désaccord avec moi. Les gens intelligents feront ce qu'ils veulent.

Les gens sages écouteront le conseil de **Douglas Crockford** :

« Si une fonctionnalité est parfois dangereuse et qu'il y a une meilleure option, alors utilisez toujours la meilleure option. »

```

9
10 let mouseFactory = function mouseFactory
11   () {
12     return Object.assign(Object.create(
13     } animal), {
14     } animalType: 'mouse',
15     } furColor: 'brown',
16     } legs: 4,
17     } tail: 'long, skinny'
18   });
19 let mickey = mouseFactory ();

```

Je n'appelle généralement pas mes fonctions d'usine "factory" - c'est juste pour illustrer. Normalement, je l'aurais simplement appelée *mouse()*.

4 On a besoin des fonctions constructeur pour privatiser les variables

Non.

En JavaScript, chaque fois que vous exportez une fonction, cette fonction a accès aux variables au-delà de son corps de définition. Quand vous les utilisez, le moteur JS crée une fermeture (**closure**). Les fermetures sont un schéma courant en JavaScript, et elles sont couramment utilisées pour la privatisation de données.

Les fermetures ne sont pas réservées aux fonctions constructeur. N'importe quelle fonction peut créer une fermeture pour gérer des données privées :

```

1 let animal = {
2   animalType: 'animal',
3
4   describe () {
5     return "An ${this.animalType} with $
6       {this.furColor} fur,
7       ${this.legs} legs, and a ${this.
8         tail} tail.";
9   }
10 };
11
12 let mouseFactory = function mouseFactory
13   () {
14   let secret = 'secret agent';
15
16   return Object.assign(Object.create(
17     animal), {

```

```

14   animalType: 'mouse',
15   furColor: 'brown',
16   legs: 4,
17   tail: 'long, skinny',
18   profession () {
19     return secret;
20   }
21 });
22 };
23
24 let james = mouseFactory();

```



5 « new » signifie que le code utilise de l'héritage par classe

Non.

Le mot-clé *new* est utilisé pour invoquer une fonction constructeur. Ce qu'il fait concrètement est :

- créer une nouvelle instance ;
- attacher le mot-clé *this* dans la fonction constructeur à la nouvelle instance ;
- assigner par référence le prototype délégué du nouvel objet à l'objet référencé par la propriété prototype de la fonction constructeur ;
- nommer le type de l'objet comme le constructeur, ce que vous remarquerez le plus souvent dans la console de débogage. Vous verrez [Object Foo], par exemple, au lieu de [Object object] ;
- permettre à *instanceof* de vérifier ou non si la référence du prototype de l'objet est la même que celle pointant vers la propriété *.prototype* de la fonction constructeur.

5.1 « instanceof » est un mensonge

Faisons une pause ici un moment pour reconsidérer la valeur du mot-clé *instanceof*. Vous pourriez changer d'avis quant à son utilité.

Important : *instanceof* ne vérifie pas le type de la façon à laquelle vous pourriez vous attendre, comme pour les autres langages à typage fort. Au lieu de ça, il vérifie l'identité du prototype, et on peut facilement duper l'opérateur. Cela peut ne pas fonctionner entre plusieurs contextes d'exécution par exemple (ce qui est généralement une source de bogues, de frustrations et de limitations inutiles). Un exemple dans la nature, sur *bacon.js* : [lien 39](#).

On peut aussi facilement le manipuler en faux positifs, et (plus souvent) en faux négatifs à partir d'autres sources. Puisqu'il s'agit d'un test d'identité sur la propriété *.prototype* de l'objet ciblé, cela peut amener de drôles de choses :

```

1 > function foo() {}
2 > var bar = { a: 'a'};
3 > foo.prototype = bar; // Object {a: 'a'
4 > baz = Object.create(bar); // Object {a:
5 > baz instanceof foo // true. oops.

```

Le dernier résultat est en accord complet avec la spécification JavaScript. Rien n'est cassé - c'est juste que *instanceof* ne peut pas garantir la fiabilité du type. **On peut le tromper facilement avec à la fois des faux positifs et faux négatifs.**

Au-delà de ça, essayer de forcer votre code JS à se comporter comme un code fortement typé peut

empêcher vos fonctions d'être déclinées en des versions plus génériques, qui sont bien plus utiles et réutilisables.

***instanceof* limite la réutilisabilité de votre code, et introduit potentiellement des bogues dans les programmes qui utilisent votre code.**

« instanceof » est un mensonge. Optez pour le duck typing : [lien 40](#).

6 Il y a une grosse différence de performances entre l'héritage par classe et l'héritage par prototype

Non.

Vous avez peut-être entendu parler des **classes cachées**, et pensez que les constructeurs battent à plate couture en performances les objets instanciés avec ***Object.create()***. Ces différences de performances sont **largement surestimées**.

Une petite fraction du temps d'exécution de votre application est passée à exécuter le JavaScript, et une fraction minuscule de ce temps est passée à accéder aux propriétés des objets. En fait, même les ordinateurs les plus lents produits aujourd'hui peuvent accéder à *des millions de propriétés par seconde*.

Ce n'est pas le goulot d'étranglement de votre application. Faites-vous une faveur et mesurez la performance ([lien 41](#)) de votre application pour **identifier les réels goulots d'étranglement**. Je suis sûr qu'il y a un million de choses que vous devriez revoir avant de penser à ce genre de micro-optimisations.

Pas convaincu ? Pour qu'une micro-optimisation ait un impact appréciable sur votre application, vous devez avoir à répéter l'opération des **centaines de milliers de fois**, et les seules différences auxquelles vous devriez vous soucier sont encore un ordre de magnitude plus loin.

Règle d'or : analysez votre application et éliminez autant que possible les chargements, requêtes réseau, lectures/écritures de fichiers et goulots d'étranglement dans les fonctions de rendu, autant que vous pouvez en trouver. **Ensuite et seulement ensuite, vous pouvez commencer à penser aux micro-**

7 Il y a une grosse différence de consommation mémoire entre l'héritage par classe et par prototype

Non.

Les deux peuvent utiliser des prototypes délégués pour partager des méthodes entre de nombreuses instances d'objets. Les deux peuvent exploiter ou éviter les fermetures pour encapsuler un tas de variables d'état.

En fait, si vous commencez avec des fonctions usines, c'est plus facile de basculer ensuite sur des

5.2 « new » est bizarre

WAT ? *new* fait aussi de drôles de choses pour retourner des valeurs. Si vous essayez de retourner une primitive, cela ne fonctionnera pas. Si vous retournez un objet arbitrairement, cela fonctionnera, mais *this* sera jeté aux oubliettes, en perdant toute référence à lui (y compris avec *.call()* et *.apply()*), et cassant le lien avec la propriété *.prototype* du constructeur.

optimisations.

Pouvez-vous dire la différence entre *0,0000000001* seconde et *0,000000001* seconde ? Moi non plus, mais je peux dire pour sûr la différence entre charger dix petites icônes et charger une police de caractères iconiques.

Si vous avez analysé votre application et avez constaté que la création d'objets était un goulot d'étranglement pour vous, alors la méthode la plus performante n'est pas d'utiliser *new* et la programmation objet par classe. **La méthode la plus rapide est d'utiliser l'écriture littérale.** Vous pouvez l'utiliser à l'intérieur d'une boucle et ajouter les objets à une collection pour éviter qu'ils soient jetés par le collecteur de déchets (garbage collector). Si cela vaut la peine d'abandonner la programmation objet orientée prototype pour des questions de performances, alors cela vaut la peine d'abandonner toute chaîne d'héritage pour ne plus utiliser que des objets déclarés littéralement.

Mais Google a dit que les classes étaient rapides...

WAT ? Google conçoit un moteur JavaScript. Vous concevez une application. Évidemment vous vous souciez de choses bien différentes. Laissez Google gérer les micro-optimisations. Souciez-vous des **réels goulots d'étranglement** de votre application. Je vous le promets, vous aurez un bien meilleur retour sur investissement en vous concentrant sur autre chose.

collections d'objets afin que vous puissiez gérer la mémoire plus attentivement et éviter d'être périodiquement bloqué par le passage du garbage collector. Pour en savoir plus sur la raison pour laquelle faire la même chose avec les constructeurs est plus délicat, lisez la note **WAT ?** sur l'opérateur *new*.

En d'autres mots, si vous voulez plus de flexibilité pour la gestion de la mémoire, optez pour des

fonctions usines au lieu de constructeurs et d'héritage par classe.

« ... si vous voulez plus de flexibilité pour la gestion de la mémoire, optez pour des fonctions usines... »

8 Les API natives utilisent les constructeurs, donc ceux-ci sont plus couramment utilisés que les fonctions usines

Non.

Les fonctions usines sont extrêmement communes en JavaScript. Par exemple, la bibliothèque JavaScript la plus populaire de tous les temps, **jQuery** expose une fonction usine à ses utilisateurs. John Resig a écrit à propos de ce choix d'utiliser une fonction usine et l'extension de prototypes plutôt qu'une classe. Basiquement, cela se réduit au fait qu'il ne voulait pas que les appelants aient à taper *new* à chaque fois qu'ils voulaient sélectionner un élément. À quoi cela aurait-il ressemblé ?

```

1  /**
2  classy jQuery - une réalité alternative
   où jQuery craint et n'a jamais dé
   collé
3  OU
4  Pourquoi personne n'aurait aimé jQuery s
   'il avait exporté une classe plutôt
   qu'une fonction usine.
5  **/
6
7  /* Cela a juste l'air débile. Est-ce que
   l'on crée un nouvel élément avec l'
   id foo ? Non. On sélectionne un élé
   ment existant du DOM, et on le place
   dans une instance d'objet jQuery, *
   /
8  var $foo = new $('#foo');
9
10 // De plus, cela nous oblige à taper
   plus pour un gain NUL.
11 var $bar = new $('.bar');
12 var $baz = new $('.baz');
13
14 // Et ça c'est..ben, je sais pas quoi.
15 var $bif = new $('.foo').on('click',
   fonction () {
16     var $this = new $(this);
17     $this.html('clicked!');
18 });

```

Qui d'autre expose des fonctions usines ?

- **React** : `React.createClass()` est une usine.

- **Angular** utilise des classes et des usines, mais les enveloppe toutes avec une usine dans son conteneur d'injection de dépendances. Tous les services Angular sont du sucre syntaxique utilisant la fonction usine `.provider()`. Il y a même un service `.factory()`, et le service `.service()` enveloppe les constructeurs normaux et expose... vous l'avez deviné : une usine pour les utilisateurs de l'injection de dépendances.
- **Ember** : `Ember.Application.create()` est une usine produisant une application. Plutôt que de créer des constructeurs à appeler avec `new`, la méthode `.extend()` vient étendre l'application.
- **Node** : les services du cœur de Node comme `http.createServer()` et `net.createServer()` sont des fonctions usines.
- **Express** est une fonction usine pour créer une application express.

Comme vous le voyez, presque toutes les bibliothèques et tous les frameworks populaires en JavaScript utilisent largement les fonctions usines. **Le seul schéma d'instanciation d'objets plus commun que les fonctions usines est l'écriture littérale.**

Les API natives de JavaScript ont commencé en utilisant les constructeurs parce que l'on a demandé à Brendan Eich de le faire davantage ressembler à Java (voir l'histoire de JavaScript). JavaScript a continué à utiliser les constructeurs par cohérence avec lui-même. Cela serait délicat de tout changer pour des fonctions usines et de déprécier les constructeurs aujourd'hui.

Cela ne veut pas dire que vos API doivent être mal fichues.

9 L'héritage par classe est plus naturel en JavaScript que l'héritage par prototype

Non.

Chaque fois que j'entends cette idée fausse, je suis tenté de répondre "**do u even JavaScript ?**" et de m'en aller... mais je résiste à l'envie et remets plutôt les pendules à l'heure.

Ne vous sentez pas mal si c'est aussi votre question. **Ce n'est pas de votre faute.** Les ressources pour apprendre le JavaScript sont nulles : [lien 41](#) !

La réponse à cette question est un grand, gigantesque

Non... (mais)

Les prototypes sont le paradigme idiomatique de l'héritage en JavaScript, et les classes sont des espèces envahissantes, des mauvaises herbes.

9.1 L'histoire en bref des bibliothèques populaires en JavaScript

À l'origine, tout le monde écrivait ses propres bibliothèques, et le partage public était minime. Puis **Prototype** est arrivé. (Le nom est un bon indice ici.) Prototype faisait son tour de magie en **étendant les prototypes délégués inclus** dans les navigateurs en utilisant de **l'héritage par concaténation**.

Plus tard, nous avons tous réalisé que modifier les prototypes natifs était un antimodèle et que les alternatives natives en conflit avec les bibliothèques cassaient l'Internet. Mais c'est une autre histoire.

Le suivant sur la montagne russe de la popularité des biblis JS a été **jQuery**. L'appel au succès de jQuery a été les **plugins jQuery**. Ils fonctionnaient en étendant le **prototype délégué de jQuery** en utilisant de **l'héritage par concaténation**.

Est-ce que vous commencez à voir un schéma ici ?

jQuery reste la bibliothèque JavaScript la plus populaire jamais conçue. Avec une LARGE marge. LARGE.

C'est là où les choses sont devenues confuses et que l'extension de classes a commencé à se faufiler dans le langage... John Resig (auteur de jQuery) a écrit un billet *Héritage simple par classes en JavaScript*, et les gens ont effectivement commencé à l'utiliser, malgré que John Resig lui-même ne pensait pas que cela avait sa place dans jQuery (**parce que la programmation objet orientée prototype fait le même boulot en mieux**).

Des semi-populaires et Java-esques frameworks comme ExtJS sont apparus, inaugurant les prémices de l'usage moyennement ordinaire des classes en JavaScript. C'était en 2007. JavaScript avait alors **12 ans** avant qu'une quelque peu populaire bibliothèque commence à exposer aux utilisateurs de l'héritage par classes.

Trois ans plus tard, Backbone **a explosé** et avait une méthode `.extend()` qui imitait l'héritage par classe, en incluant toutes les vilaines fonctionnalités comme les hiérarchies fragiles d'objets. **Voilà où l'enfer se déchaîne.**

Une application d'environ 100 000 lignes de code a été réalisée avec Backbone. Quelques mois après le début du projet, j'ai débogué une hiérarchie à six niveaux en essayant de trouver un bug. Je suis passé sur chaque ligne de code du constructeur en remon-

tant peu à peu la chaîne d'héritage. J'ai finalement trouvé et corrigé le bug sur la classe de plus haut niveau. Ensuite, j'ai eu à corriger un tas de classes enfants parce qu'elles dépendaient du comportement bogué de la classe de base. Des heures de frustration qui auraient dû être une correction de cinq minutes.

Ce n'est pas JavaScript. Je me suis soudainement retrouvé à vivre de nouveau dans l'enfer Java. Cet endroit de solitude, sombre et effrayant, où chaque mouvement rapide peut faire frémir des hiérarchies entières qui s'effondrent avec fracas, nous laissant agonisant et pris de fortes convulsions.

Voilà de quoi sont faites les refontes complètes d'applications.

Mais, perdu dans la documentation de Backbone, un rayon de soleil d'or :

```

1 var object = {};
2
3 _.extend(object, Backbone.Events);
4
5 object.on("alert", function(msg) {
6   alert("Triggered " + msg);
7 });
8
9 object.trigger("alert", "an event");

```

Notre bon vieil ami, **l'héritage par concaténation**, nous égaye la journée avec un **mixin Backbone.Events**.

Il s'avère que, **si vous regardez n'importe quelle bibliothèque JavaScript non triviale d'assez près, vous y trouverez des exemples de concaténation et de délégation.** C'est tellement commun et automatique pour les développeurs JavaScript qu'ils le font sans même y penser comme à de l'héritage, même si cela accomplit le même objectif.

L'héritage en JavaScript est tellement facile qu'il rend confus les gens qui s'attendent à ce que cela demande des efforts. Pour le rendre plus difficile, nous avons ajouté « class ».

Et comment avons-nous ajouté class? Nous l'avons conçu **par-dessus l'héritage par prototype en utilisant des prototypes délégués et de la concaténation d'objets bien entendu!**

C'est comme rouler avec votre Tesla Model S jusqu'à un vendeur de voitures pour l'échanger contre une Ford Pinto rouillée de 1983.

10 Le choix entre l'héritage par classe et par prototype dépend du cas d'utilisation

Non.

La programmation objet orientée prototype est plus simple, plus flexible, et bien moins sujette aux erreurs de conception. J'ai tenu ce discours et encouragé

les gens à venir vers moi avec un cas d'usage montrant la supériorité des classes pendant de nombreuses années. Des centaines de milliers de personnes ont entendu cet appel. Le peu de réponses

que j'ai reçues dépendaient d'une ou plusieurs idées fausses adressées dans cet article.

J'ai été jadis un fan de l'héritage par classes. J'achetais l'idée complètement. Je construisais des hiérarchies d'objets partout. J'ai conçu des outils de développement d'applications rapides en orienté objet pour aider les architectes logiciels à concevoir des hiérarchies d'objets et des relations qui prenaient du sens. Il fallait un outil visuel pour vraiment pouvoir cartographier toutes les relations d'objets dans les applications pour entreprise ayant cette taxonomie d'héritage par classes.

Peu après ma transition de C++/Java à JavaScript, j'ai arrêté de faire tout ça. Non pas parce que je construisais des applications moins complexes (au contraire), mais parce que JavaScript était tellement plus simple que je n'avais plus besoin de tout cet outillage de conception OO.

Je faisais auparavant du conseil en conception d'applications et recommandait souvent des réécritures radicales. Pourquoi? Parce que **toutes les hiérarchies d'objets deviennent à la longue inadaptées aux nouveaux cas d'utilisation.**

Je n'étais pas le seul. En ce temps, les **réécritures complètes** étaient très courantes pour les nouvelles versions de logiciels. La plupart de ces réécritures étaient nécessaires à cause du verrouillage causé par des hiérarchies de classes fragiles et ankylosées : [lien 42](#). Des livres entiers ont été écrits sur les erreurs de conception OO et comment les éviter ou les rectifier. Il semblait que chaque développeur avait une copie de "Design Patterns" ([lien 43](#)) sur son bureau.

Je vous recommande de suivre le conseil du Gang of Four ([lien 44](#)) sur ce point :

« Favoriser la composition d'objets par rapport à l'héritage par classes. »

En Java, c'était plus difficile, car vous aviez toujours à utiliser des classes pour y parvenir.

En JavaScript, nous n'avons pas cette excuse. C'est en fait **bien plus simple** en JavaScript de simplement créer les objets dont vous avez besoin en assemblant divers **prototypes** plutôt que de gérer des hiérarchies d'objets.

WAT? Sérieusement. Vous voulez que l'objet jQuery puisse changer un input en *megaCalendarWidget*? Vous n'avez pas à *extend* une *class*. JavaScript a l'extension dynamique d'objets, et jQuery expose son propre prototype afin que vous puissiez l'étendre — sans un mot-clé *extend*! **WAT?** :

```

1 /*
2  Comment étendre le prototype de jQuery :
3  Tellement difficile.
4  Mon cerveau me fait mal.
5  ouch.
6  */
7
8  jQuery.fn.megaCalendarWidget =
9      megaCalendarWidget;
10 // omg je suis si content que ce soit
    fini
  
```

La prochaine fois que vous appellerez la fonction usine jQuery, vous obtiendrez une instance qui peut rendre vos inputs méga bien. (*Note du traducteur : c'est ici une plaisanterie de l'auteur, car les widgets datepicker sont connus pour rendre furieux les utilisateurs sur smartphone/tablettes; utilisez des polyfills sur <input type='date'>.*)

De manière similaire, vous pouvez utiliser **Object.assign()** pour composer un nombre quelconque d'objets ensemble :

```

1 import ninja from 'ninja'; // ES6
2   modules
3 import mouse from 'mouse';
4
5 let ninjamouse = Object.assign({}, mouse
6   , ninja);
  
```

Non, vraiment - **n'importe quel nombre d'objets** :

```

1 // Je ne suis pas sûr que Object.assign
2   () soit disponible (ES6)
3 // donc cette fois j'utilise Lodash. C'
4   est comme Underscore,
5 // mais en 200 % mieux. Vous pouvez
6   aussi utiliser
7 // jQuery.extend() ou Underscore's
8   .extend()
9 var assign = require('lodash/object/
10  assign');
11
12 var skydiving = require('skydiving');
13 var ninja = require('ninja');
14 var mouse = require('mouse');
15 var wingsuit = require('wingsuit');
16
17 // La quantité de génialitude qui suit
18   peut être trop importante
19 // pour les seniors à problèmes
20   cardiaques ou les jeunes enfants.
21 var skydivingNinjaMouseWithWingsuit =
22   assign({}, // crée un nouvel objet
23   skydiving, ninja, mouse, wingsuit); //
24   copier toute la coolitude dedans
  
```

Cette technique est appelée **héritage par concaténation**, et les prototypes dont vous héritez sont parfois appelés **prototypes exemplaires**, ce qui les différencie des prototypes délégués dans le sens où vous les **copiez**, plutôt que de leur déléguer le travail.

11 ES6 a le mot-clé « class » et nous devrions tous l'utiliser

Non.

Il y a des tas de raisons valables pour éviter le mot-clé ES6 class (lien 45), et non des moindres, ce qui en fait un ajustement gênant pour JavaScript.

Nous avons déjà un **incroyablement puissant et expressif système d'objets** en JavaScript. Le concept de classe tel qu'il est implémenté aujourd'hui en JavaScript est plus restrictif (dans le sens mauvais, pas le sens typage fort), et il obscurcit l'ex-

cellent système de prototypes objet (lien 46) qui a été conçu pour le langage il y a bien longtemps.

Vous savez ce qui serait vraiment bien pour JavaScript ? Une meilleure syntaxe et des abstractions conçues par-dessus les prototypes **de la perspective d'un programmeur familier à la programmation objet orientée prototype.**

Ce serait vraiment cool : lien 47.

Retrouvez l'article d'Eric Elliott traduit par Sylvain Pollet-Villard en ligne : lien 48



Windows

Les derniers tutoriels et articles

À la découverte de Windows 10 Insider Preview

télécharger, installer et découvrir

1 Téléchargement de Windows 10 Insider Preview

Windows 10 Insider Preview (c'est le nouveau nom de Windows 10 Technical Preview) est librement téléchargeable sur le site de Microsoft : lien 49.



Cette préversion est disponible en version 32 et 64 bits pour les langues suivantes : anglais (US et UK), chinois, portugais, allemand, français, français pour le Canada, russe, japonais, italien, néerlandais, espagnol, espagnol pour l'Amérique latine, polonais, turc, suédois, tchèque, arabe, coréen, finnois et thaï.

Je vous déconseille d'installer Windows 10 Insider Preview sur votre machine principale. Il ne s'agit en aucun cas d'une version définitive du système et le mieux est de la tester dans une machine virtuelle.

Vous allez voir comment l'installer dans une machine Hyper-V et VirtualBox dans les sections suivantes.

À terme, et durant l'année qui suivra le lancement de Windows 10, tous les utilisateurs de Windows 7 et de Windows 8.1 pourront faire une mise à jour gratuite de leur système vers Windows 10.

Avant d'installer Windows 10, vous allez télécharger une image ISO. Cliquez sur le lien **Télécharger les fichiers ISO**. Une nouvelle page s'affiche. Déplacez-vous vers le bas de la page, développez l'entrée **Français** et cliquez sur le lien **Télécharger** de la version 32 bits ou de la version 64 bits. Tous les ordinateurs récents sont équipés de processeurs 64 bits.



Alors que j'écris ces lignes, la dernière *release* accessible est la 10074.



Téléchargez :

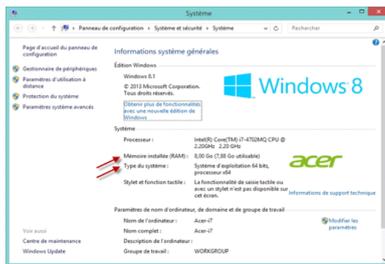
- la version 32 bits si vous voulez tester Windows 10 Insider Preview sur une machine ancienne équipée d'un processeur 32 bits ;
- la version 64 bits dans tous les autres cas, et en particulier si votre ordinateur est équipé de plus de 4 Go de mémoire.

Si vous n'avez aucune idée de ce dont je parle, appuyez simultanément sur les touches **Windows** et **Pause** du clavier pour afficher la fenêtre **Système**. Regardez ce qui est affiché à la suite de l'étiquette **Type du système**, et en particulier le type du processeur :

- x64 signifie 64 bits ;
- x86 signifie 32 bits.

Une autre information peut également vous aider : la quantité de mémoire installée.

Dans cet exemple, l'ordinateur est équipé d'un processeur 64 bits et de 8 Go de mémoire. C'est donc clairement vers une version 64 bits de l'image ISO que nous allons nous orienter :



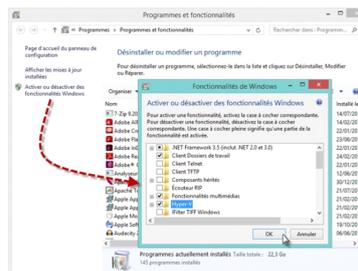
2 Installation de Windows 10 Insider Preview dans une machine virtuelle Hyper-V

Pour pouvoir utiliser Hyper-V, l'hyperviseur de Microsoft, deux conditions doivent être réunies. Votre ordinateur doit :

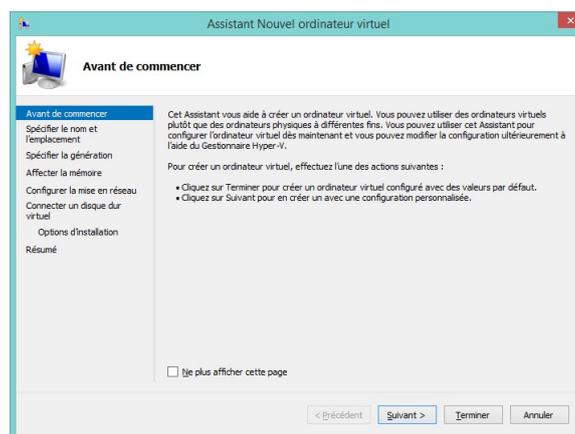
1. Fonctionner avec une version Professionnelle ou Entreprise de Windows 7 ou 8.1.
2. Être équipé d'un processeur qui supporte la virtualisation (c'est le cas de la plupart des processeurs actuels). En corollaire, la virtualisation doit avoir été activée dans le BIOS.

Dans ce tutoriel, nous allons installer une version française 64 bits dans une machine virtuelle Hyper-V. Le système hôte sera Windows 8.1 Update français 64 bits.

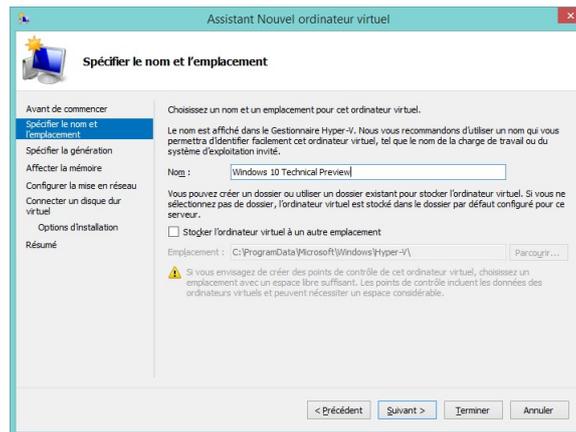
Commencez par exécuter le Gestionnaire Hyper-V. Si cette application n'est pas accessible, cliquez sur **Démarrer**, tapez **fonctionnalités** et cliquez sur **Programmes et fonctionnalités**. La boîte de dialogue Programmes et fonctionnalités s'affiche. Cliquez sur **Activer ou désactiver des fonctionnalités Windows** dans le volet gauche. Cochez la case qui précède l'entrée Hyper-V, cliquez sur **OK** et refermez la boîte de dialogue **Programmes et fonctionnalités** :



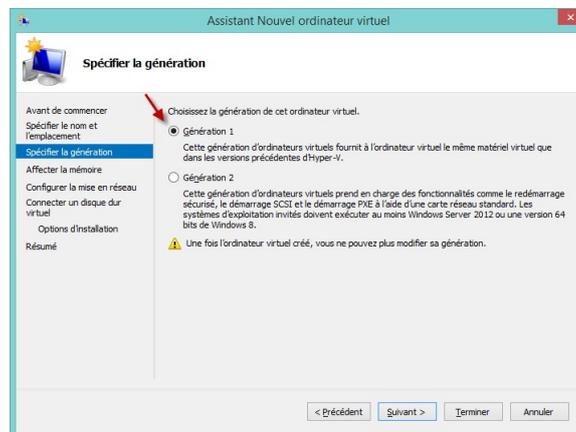
Lancez la commande **Nouveau/Ordinateur virtuel** dans le menu **Action** :



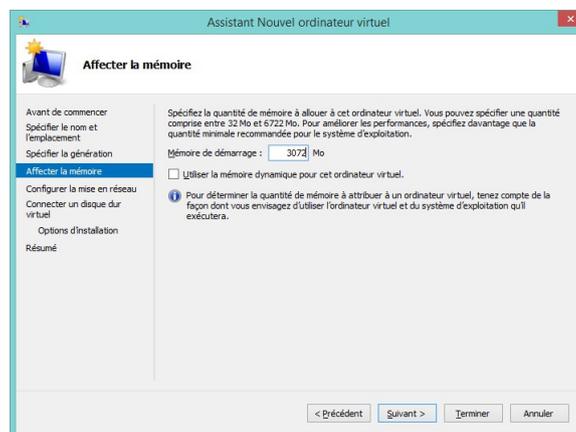
Cliquez sur **Suivant** et donnez un nom à l'ordinateur virtuel : Windows 10 Insider Preview.



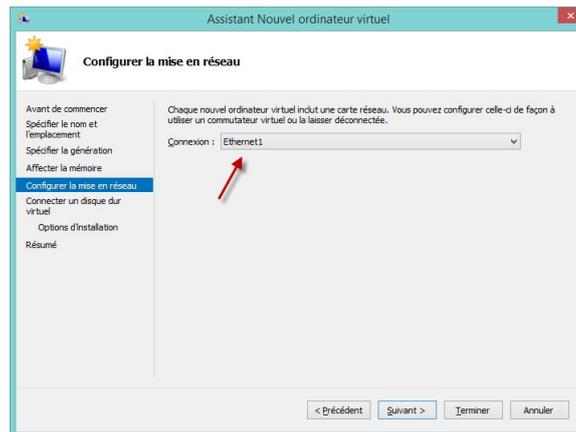
Cliquez sur **Suivant** et choisissez un ordinateur virtuel de génération 1.



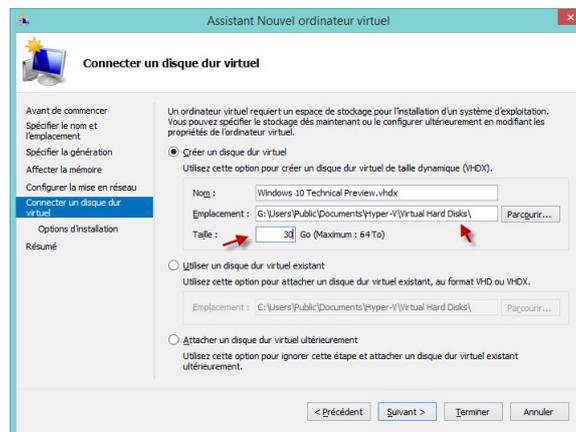
Cliquez sur **Suivant** et affectez 2 ou 3 Go de mémoire à cette machine virtuelle :



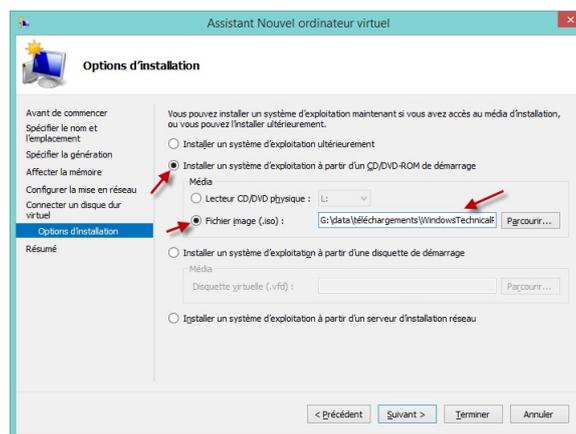
Cliquez sur **Suivant** et indiquez comment la machine virtuelle se connectera à Internet :



Cliquez sur **Suivant** et indiquez l'emplacement et la taille du disque virtuel :

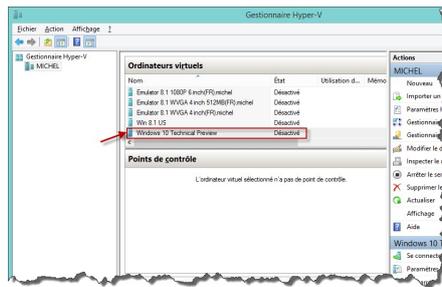


Cliquez sur **Suivant**. Choisissez l'option **Installer un système d'exploitation à partir d'un CD/DVD-ROM de démarrage**. Dans le groupe d'options Media, choisissez **Fichier image (.iso)** et désignez l'image ISO que vous avez téléchargée en cliquant sur **Parcourir** :

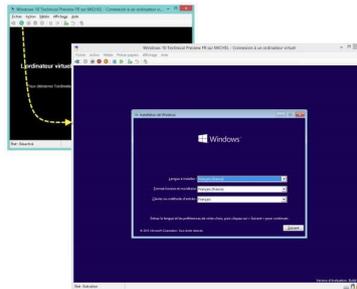


Cliquez sur **Suivant**, vérifiez que toutes les informations sont correctes puis cliquez sur **Terminer**.

Quelques instants plus tard, la machine virtuelle que vous venez de créer apparaît dans la zone **Ordinateurs virtuels** :



Double-cliquez dessus pour ouvrir l'ordinateur virtuel puis cliquez sur l'icône **Démarrer** pour déclencher l'installation de Windows 10 Insider Preview dans votre machine virtuelle :



Une fois l'installation terminée, vous aurez tout le loisir de tester cette nouvelle mouture de Windows sans nuire à votre système d'exploitation principal.

3 Installation de Windows 10 Insider Preview dans une machine virtuelle VirtualBox

Si votre ordinateur n'est pas en mesure d'exécuter HyperV, vous pouvez cependant installer une machine virtuelle en faisant appel à une application tierce nommée VirtualBox.

3.1 Première étape

Ouvrez votre navigateur Web et allez sur la page de téléchargement de VirtualBox : [lien 50](http://www.virtualbox.org/wiki/Downloads). Téléchargez la dernière version du logiciel en cliquant sur le lien **VirtualBox for Windows hosts**.



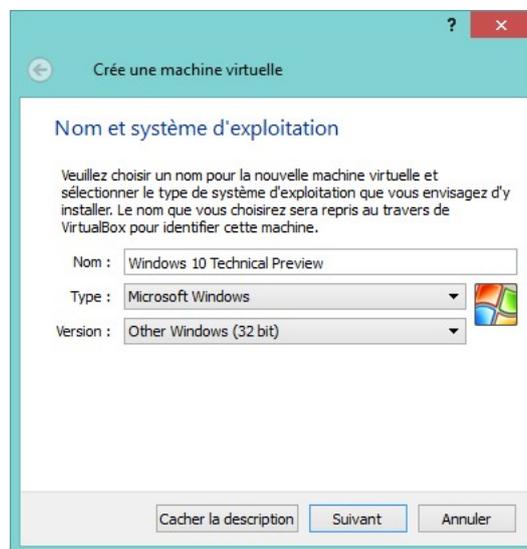
3.2 Deuxième étape

Lancez VirtualBox. Pour cela, cliquez sur **Démarrer**, tapez **virtual** dans la zone de texte Recherchez et cliquez sur **Oracle VM VirtualBox**.

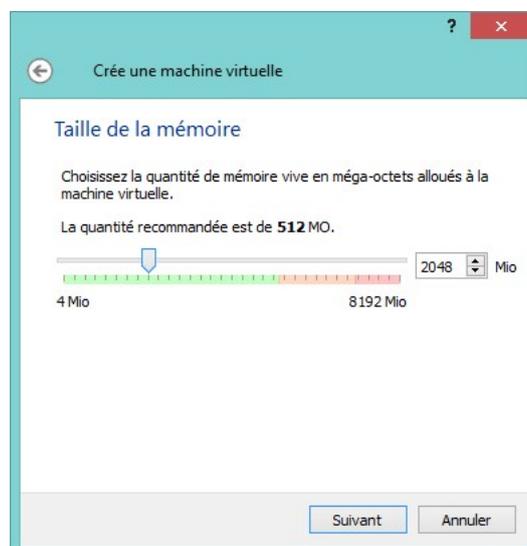


Cliquez sur l'icône **Nouvelle**, dans le coin supérieur gauche de la fenêtre. Donnez un nom à la machine virtuelle. Sélectionnez **Microsoft Windows** dans la première liste déroulante. Si ces options existent, sélectionnez **Windows 10 (32 bits)** ou **Windows 10 (64 bits)** dans la seconde liste déroulante selon que vous voulez installer une version 32 ou 64 bits de Windows 10. Si ces options n'existent pas, sélectionnez **Other Windows (32 bits)** ou **Other Windows (64bits)**.

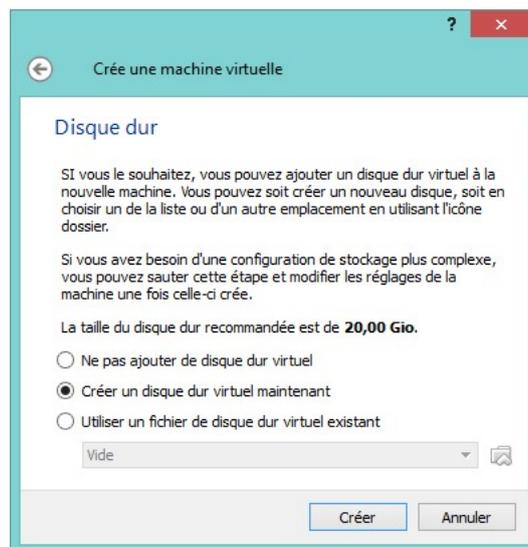
À titre d'exemple, nous allons installer Windows 10 Insider Preview Entreprise 32 bits. La boîte de dialogue **Créer une machine virtuelle** est donc configurée comme ceci :



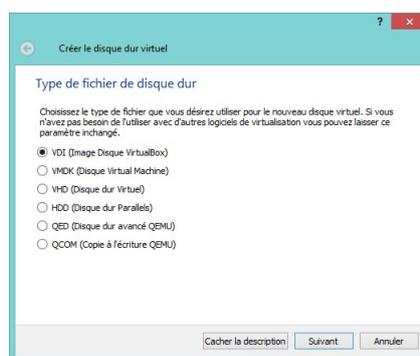
Cliquez sur **Suivant** et sélectionnez la quantité de mémoire que vous souhaitez allouer à la machine virtuelle. Pour information, la version 32 bits de Windows 10 a besoin d'au moins 1 Go de mémoire, et la version 64 bits d'au moins 2 Go. Ceci étant dit, plus vous affecterez de mémoire à la machine virtuelle, meilleur sera son comportement.



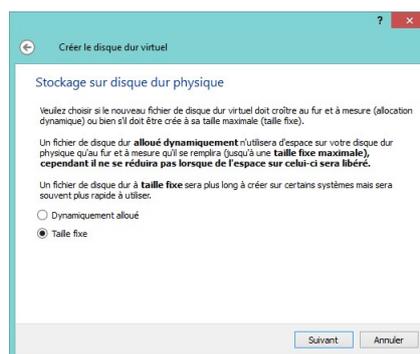
Cliquez sur **Suivant**. Sélectionnez l'option **Créer un disque dur virtuel maintenant** et cliquez sur **Créer** :



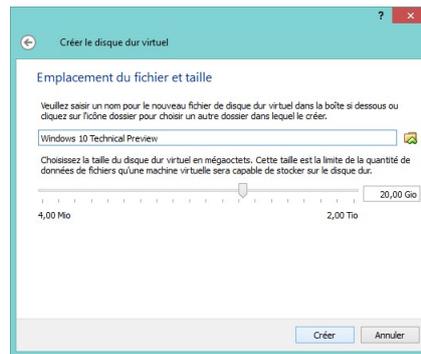
Sélectionnez l'option **VDI (Image Disque VirtualBox)** et cliquez sur **Suivant** :



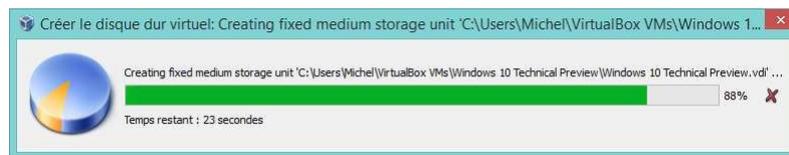
Dans l'étape suivante, sélectionnez **Taille fixe** et cliquez sur **Suivant** :



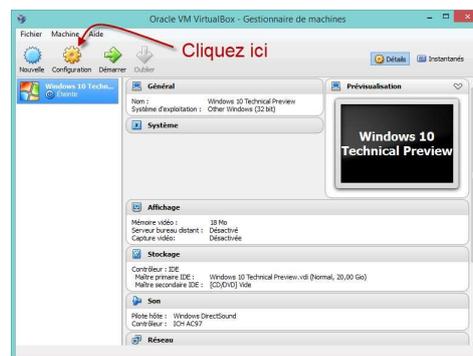
Dans l'étape suivante, allouez 20 Gio ou 30 Gio à votre machine virtuelle. C'est bien assez pour tester Windows 10 Insider Preview :



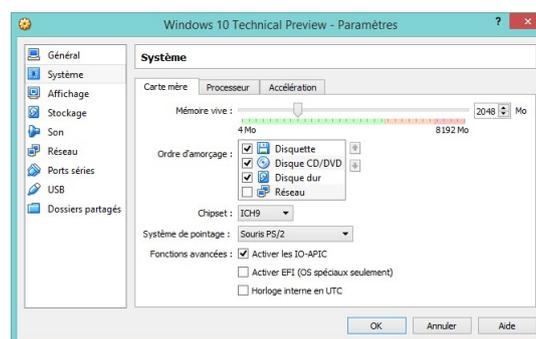
Cliquez sur **Créer** et patientez jusqu'à la fin de la création du disque VDI :



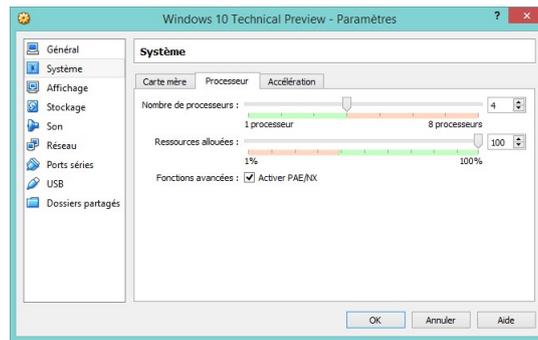
Lorsque le disque aura été créé, la fenêtre de VirtualBox se présentera comme ceci. Cliquez sur l'icône **Démarrer** dans la barre d'outils :



Cliquez sur l'icône **Configuration**. Cette action affiche la boîte de dialogue Paramètres. Sélectionnez l'onglet **Système**. Assurez-vous que la case **Activer les IO APIC** est cochée, puis sélectionnez **ICH9** dans la liste déroulante **Chipset**.



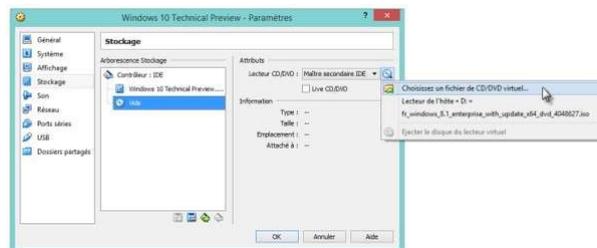
Basculez sur l'onglet **Processeur** et choisissez le nombre de processeurs à utiliser dans la machine virtuelle. Vérifiez aussi que la case **Activer PAE/NX** est cochée :



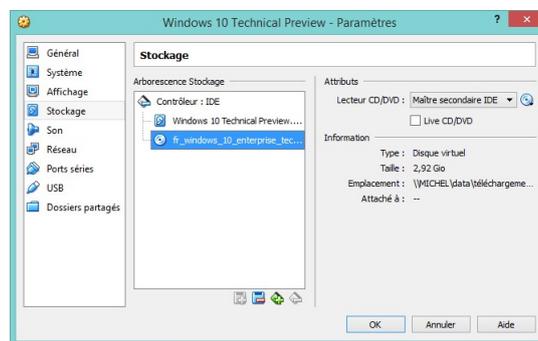
Basculez sur l'onglet **Accélération**. Cochez la case **Activer VT-x/AMD-V** pour utiliser la virtualisation matérielle **Intel VT-x** ou **AMD-V**.

Sélectionnez l'onglet **Stockage** dans le volet gauche. Cliquez sur **Contrôleur IDE** et choisissez **ICH6** dans la liste déroulante **Type**.

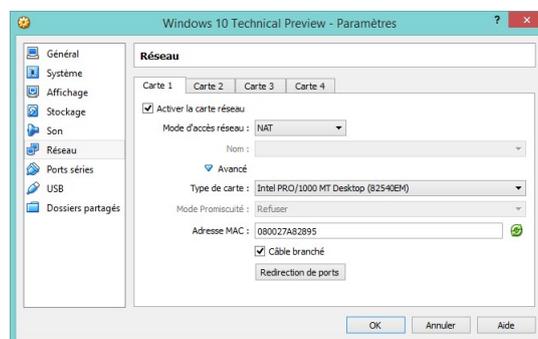
Cliquez sur l'icône **Vide**. Sous Attributs, cliquez sur l'icône **Configuration du lecteur de CD/DVD virtuel**, puis sur la commande **Choisissez un fichier de CD / DVD virtuel**.



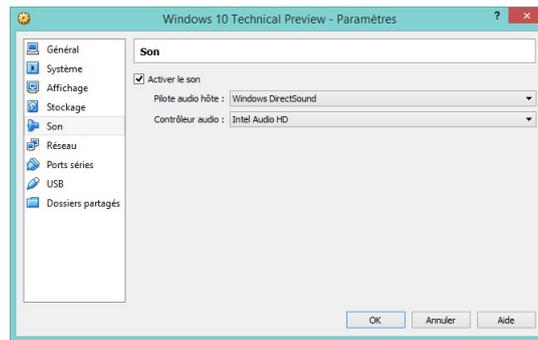
Désignez le fichier ISO à utiliser et cliquez sur **Ouvrir**.



Basculez sur l'onglet **Réseau** dans le volet gauche. Cochez la case **Activer la carte réseau**, sélectionnez **NAT** dans la liste déroulante **Mode d'accès réseau**, cliquez sur **Avancé**, puis sélectionnez **Intel PRO/1000 MT Desktop** dans la liste **Type de carte**.

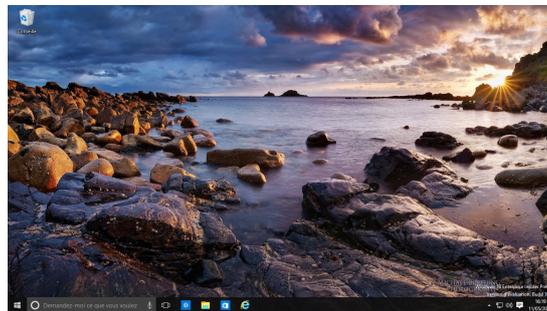


Basculez sur l'onglet **Son** dans le volet gauche. Cochez la case **Activer le son**. Sélectionnez **Windows DirectSound** dans la liste déroulante **Pilote audio hôte** et **Intel Audio HD** dans la liste **Contrôleur audio**.



Validez tous ces réglages en cliquant sur **OK**. La boîte de dialogue **Paramètres** se ferme : votre machine virtuelle est maintenant totalement configurée.

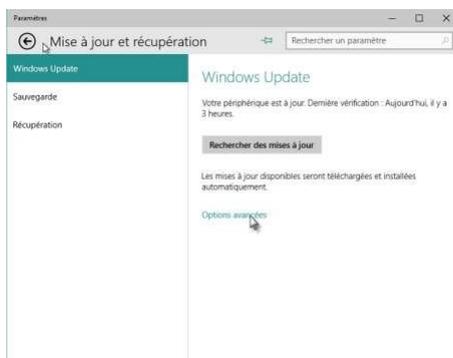
Cliquez sur l'icône **Démarrer** pour démarrer l'installation de Windows 10 Insider Preview dans la machine virtuelle. Suivez les directives de l'**Assistant d'installation**. Dans quelques minutes, vous devriez être prêt à utiliser votre nouveau système :



4 Accès aux dernières versions de Windows 10 Insider Preview

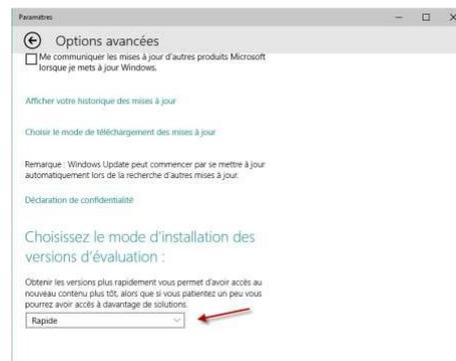
Pour accéder aux mises à jour mineures de Windows 10 Insider Preview, vous devez modifier le paramétrage de Windows Update. Cliquez sur **Démarrer**, tapez **update** et cliquez sur **Windows Update**.

La fenêtre **Mise à jour et récupération** s'affiche, onglet **Windows Update** sélectionné. Cliquez sur **Options avancées** dans le volet droit :



Déplacez-vous dans la partie inférieure de la fe-

nêtre **Options avancées** et choisissez **Rapide** dans la liste déroulante :



Cliquez sur la flèche **Précédent** dans l'angle supérieur gauche de la fenêtre. La fenêtre **Mise à jour et récupération** s'affiche à nouveau. Cliquez sur **Recherche des mises à jour** et patientez jusqu'à la fin de l'installation de la dernière version de Windows 10 Insider Preview.

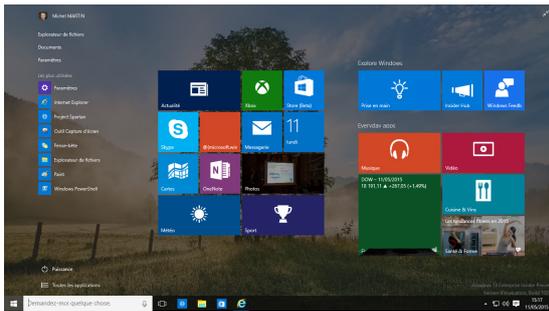
5 Découverte de Windows 10 Insider Preview

Nous allons rapidement passer en revue les principales nouveautés de cette préversion.

5.1 Premiers pas - Le menu Démarrer et les fenêtres des applications Modern UI

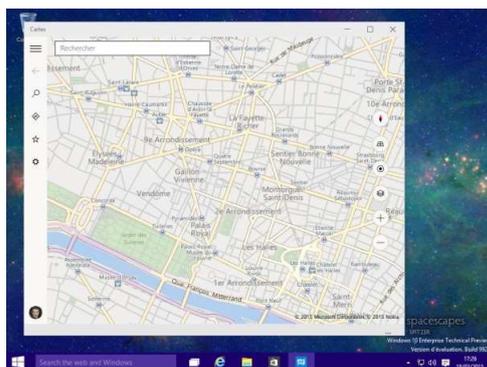
Dès le démarrage de Windows 10, vous vous retrouvez sur le Bureau. Est-ce que cela veut dire que l'interface Modern UI a disparu ? Pour en avoir le cœur net, appuyez sur la touche **Windows** du clavier.

Surprise ! Ce n'est pas l'écran d'accueil qui s'affiche, mais le menu **Démarrer** qui se déroule, comme si vous étiez sous Windows 7 :

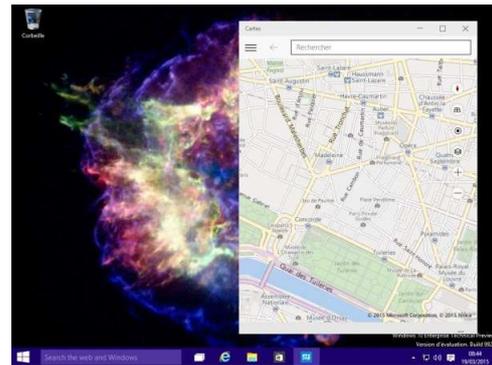


Regardez d'un peu plus près : la partie droite du menu **Démarrer** contient des vignettes d'applications Modern UI. Ces dernières n'ont donc pas disparu. Remarquez également le retour d'Aero Glass dans l'interface utilisateur.

Pour en savoir un peu plus sur le rendu des applications modernes, cliquez (par exemple) sur l'icône **Cartes**. L'application **Cartes** s'ouvre dans une fenêtre, comme s'il s'agissait d'une application de bureau :



Remarquez les icônes de minimisation, d'agrandissement et de fermeture dans l'angle supérieur droit de la fenêtre. Si les icônes de minimisation et de fermeture étaient disponibles dans Windows 8.1 Update, l'icône d'agrandissement est une nouveauté. Cliquez dessus. Comme vous pouvez le voir, la fenêtre de l'application Modern UI est redimensionnée, comme s'il s'agissait d'une application de Bureau. Vous pouvez d'ailleurs agir sur ses bordures pour la redimensionner et la déplacer sur l'écran en agissant sur sa barre de titre :



Cette constatation est valable pour toutes les applications modernes : vous pouvez désormais les redimensionner et les déplacer comme s'il s'agissait de fenêtres traditionnelles !

5.2 Étendre le menu Démarrer

Pour ajouter des icônes dans le menu **Démarrer**, vous pouvez :

- cliquer du bouton droit sur une des icônes affichées dans la partie gauche du menu et sélectionner **Épingler à l'écran d'accueil** dans le menu ;
- cliquer sur **Toutes les applications** pour afficher toutes les applications installées, cliquer du bouton droit sur l'une d'entre elles et sélectionner **Épingler à l'écran d'accueil** dans le menu.

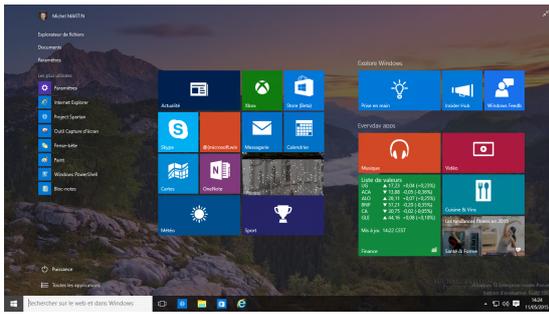
Vous pouvez également ouvrir l'**Explorateur de fichiers** et glisser-déposer un fichier quelconque sur le bouton **Démarrer**. Ce fichier sera ajouté aux éléments épinglés, dans la partie gauche du menu.

Remarquez également que vous pouvez réorganiser les icônes du menu **Démarrer** par un simple glisser-déposer, aussi bien dans la partie droite que dans la partie gauche du menu, ou encore redimensionner les icônes affichées dans la partie droite du menu **Démarrer** en cliquant du bouton droit et en sélectionnant une entrée sous **Redimensionner**.

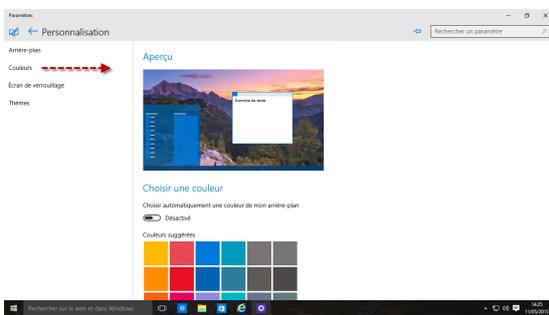
5.3 Menu Démarrer, écran d'accueil, barre des tâches et centre de notification

Le menu **Démarrer** a quelque peu changé. Notez :

- l'icône **Puissance** qui donne accès aux commandes **Déconnecter**, **Arrêter** et **Redémarrer** ;
- la possibilité d'agrandir ou de diminuer la taille du menu en utilisant l'icône en forme de double flèche en haut et à droite :



Notez également que le menu **Démarrer** et la barre des tâches sont désormais transparents. Cliquez droit sur le bureau et sélectionnez **Personnaliser** dans le menu. L'onglet **Couleurs** permet de personnaliser la couleur de la barre des tâches, des bordures et du menu **Démarrer** :

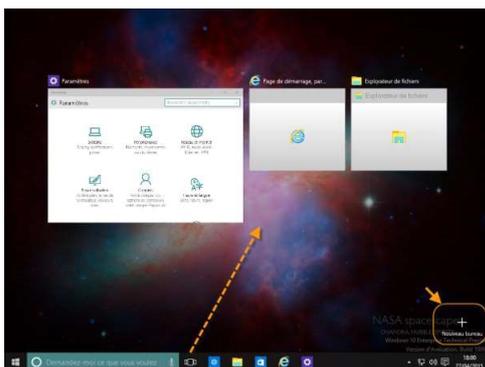


5.4 Accès aux applications et bureaux virtuels

L'icône **Applications actives**, à droite de **Cortana**, ou le raccourci clavier **Windows + Tab**, donne accès aux applications ouvertes et aux bureaux virtuels :



Cliquez sur l'icône **Applications actives**. Des miniatures donnent accès aux applications ouvertes et l'icône **Nouveau bureau**, dans l'angle inférieur droit de la fenêtre permet de créer un bureau virtuel :



Pour accéder à une des applications ouvertes, il suffit de cliquer sur sa miniature.

Pour créer un bureau virtuel, cliquez sur **Nouveau bureau**. Le nouveau bureau est immédiatement créé. Cliquez sur son icône (Bureau 2 dans la copie d'écran suivante). Vous pouvez ouvrir une ou plusieurs applications dans ce nouveau bureau :



Pour transférer une application d'un bureau à un autre, cliquez sur l'icône **Applications actives** ou appuyez sur **Windows + Tab**, puis glissez-déposez l'application sur la vignette du bureau destination :

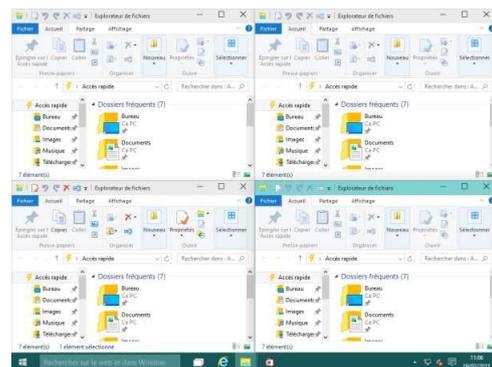


Vous pouvez également cliquer du bouton droit sur la vignette de l'application à déplacer, pointer **Déplacer vers** et choisir le bureau destination :



5.5 Snap multiple

Désormais, il est possible d'ancrer jusqu'à quatre applications sur l'écran en utilisant les raccourcis **Windows + Gauche**, **Windows + Droite**, **Windows + Haut** et **Windows + Bas**. Ici par exemple, quatre fenêtres de l'**Explorateur de fichiers** se partagent l'écran :

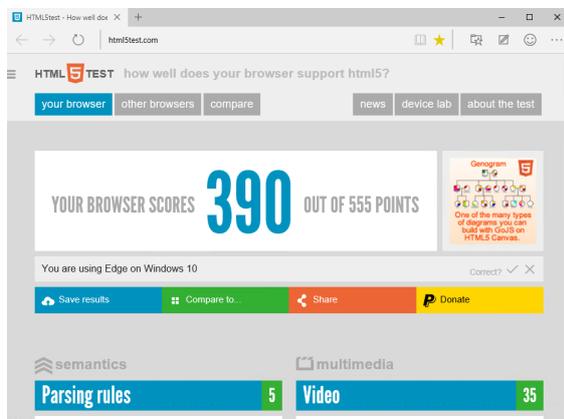


5.6 Microsoft Edge

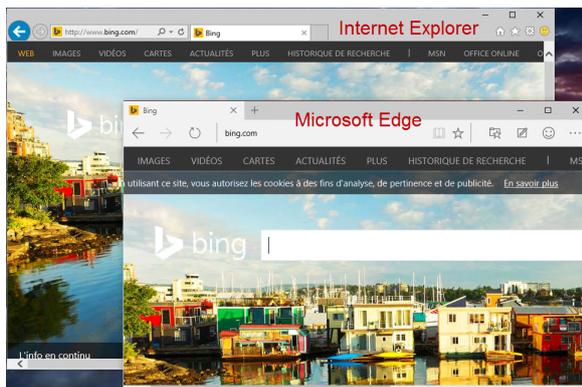
Microsoft Edge (anciennement connu sous le nom Project Spartan) est un navigateur moderne pour PC, tablettes et téléphones. Par « moderne », entendez qu'il s'agit d'une application Modern UI et non de bureau (ce qui finalement, ne change pas grand-chose, car ces deux interfaces ont de plus en plus tendance à se rejoindre).

Edge est là pour prêter main-forte au vieillissant Internet Explorer qui, normalement, ne devrait plus être maintenu à l'avenir. Rassurez-vous cependant, les mises à jour de sécurité d'Internet Explorer seront toujours proposées...

Edge supporte mieux HTML5 qu'Internet Explorer : le score obtenu est de 390 points sur 555 contre 348 pour Internet Explorer 11 :



Edge est doté d'une interface minimaliste, assez proche de celle d'Internet Explorer et des autres navigateurs Web :

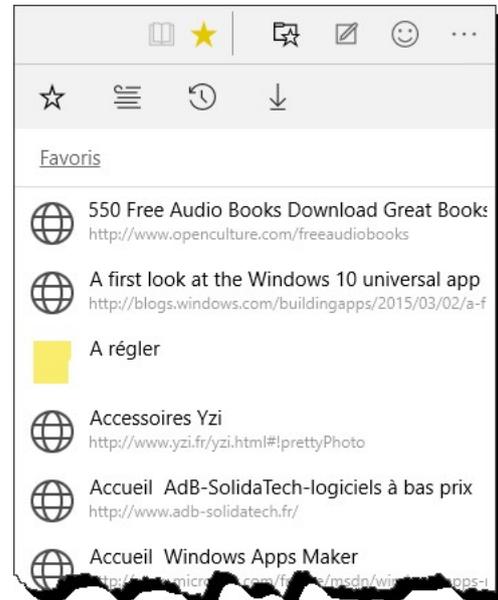


Notez cependant quelques nouvelles icônes dans la barre d'outils :

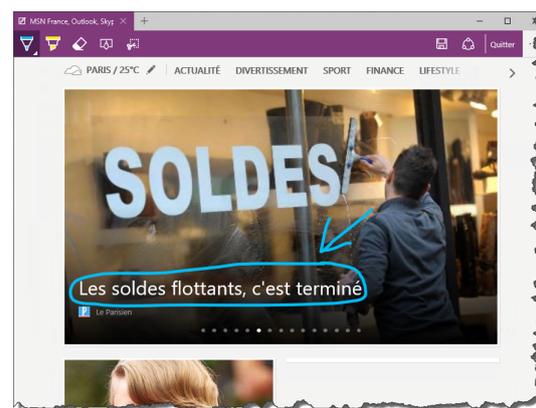


L'icône **Mode Lecture** améliore la lisibilité de la page en supprimant ses fioritures (barres latérales, effets graphiques, etc.) pour se concentrer sur le texte.

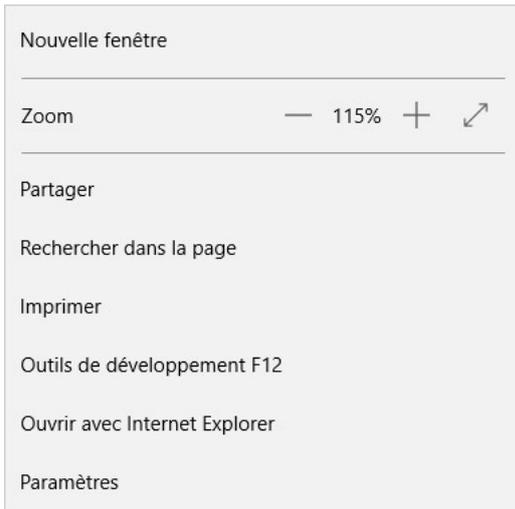
L'icône **Favoris**, liste de lectures, historique et téléchargements révèle un volet sur le côté droit de la fenêtre dans lequel vous avez accès à vos sites favoris, aux pages que vous voulez lire (liste de lectures), à l'historique de navigation et aux fichiers téléchargés :



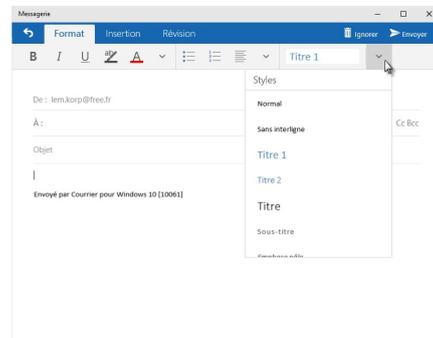
L'icône **Annotations** permet d'ajouter des marques et des annotations sur les pages Web visitées :



L'icône **Commentaires** permet d'envoyer des commentaires à Microsoft, et enfin, l'icône **Autres actions** donne accès à des commandes complémentaires (anciennement proposées dans l'icône **Outils**, en forme de roue crantée, d'Internet Explorer) :



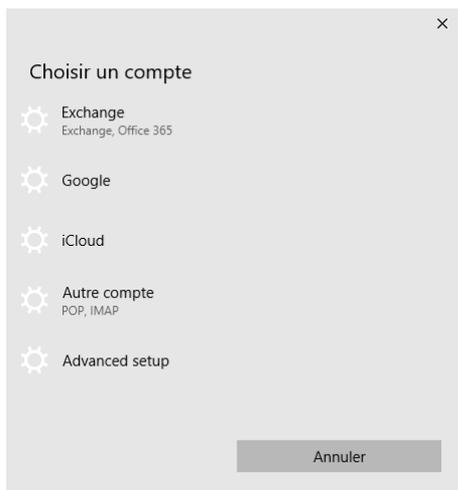
qui rappellent étrangement **Word**. Vous pouvez par exemple insérer des tableaux, des photos et des listes à puces dans vos mails :



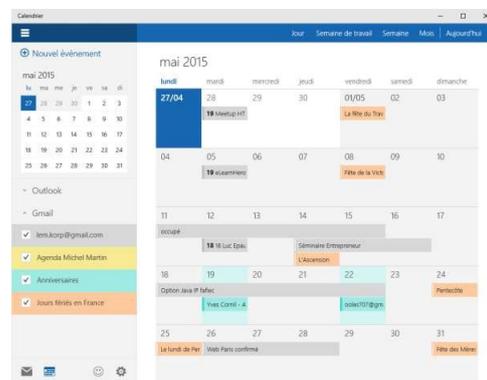
5.7 Applications Courrier et Calendrier

L'application moderne **Messagerie** remplace l'application **Courrier**.

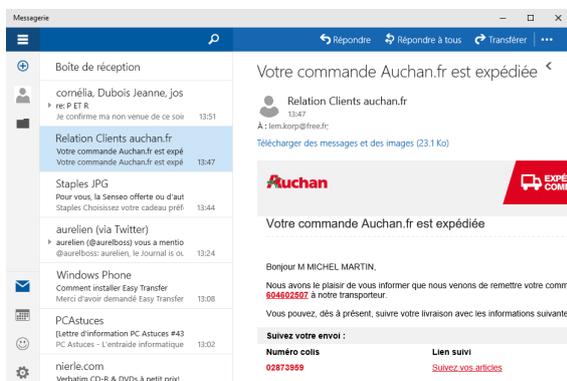
Après avoir défini un ou plusieurs comptes (Exchange, Outlook, Google, iCloud, POP ou IMAP) :



L'application **Calendrier** peut être liée à un compte **Outlook**, **Exchange**, **Google**, **iCloud**, etc. Voici un avant-goût de cette nouvelle mouture du calendrier :



L'application est immédiatement opérationnelle. Voici comment elle se présente :



5.8 Cortana

L'assistante vocale **Cortana** est disponible en français depuis la build 10041 de Windows 10 Insider Preview. Pour pouvoir l'utiliser, cliquez sur la zone **Rechercher sur le Web et dans Windows**, affichée dans la partie gauche de la barre des tâches. Un écran de bienvenue s'affiche. Cliquez sur **Auto-riser** :

La fenêtre de création de messages est très facile à utiliser. Divers contrôles sont accessibles à travers les onglets **Format**, **Insertion** et **Révision**



Cliquez sur **Autoriser**. Cortana vous informe qu'elle doit activer la personnalisation de la voix, de l'entrée manuscrite et de la frappe. Cliquez sur **Oui**.

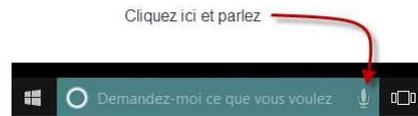
Vous devez alors entrer le **nom** ou le **pseudonyme** avec lequel vous voulez que Cortana s'adresse à vous.



Complétez la zone de texte et cliquez sur **Entrer**.

Cortana vous propose alors d'écouter la prononciation de votre nom ou pseudo. Cliquez sur **Ecouter la prononciation** puis sur **C'est parfait** ou sur **C'est faux** en fonction du résultat. Cliquez enfin sur **Terminer**. Cortana est entièrement configurée. Vous pouvez désormais lui demander ce que

vous voulez en cliquant sur le **micro** et en posant votre question :

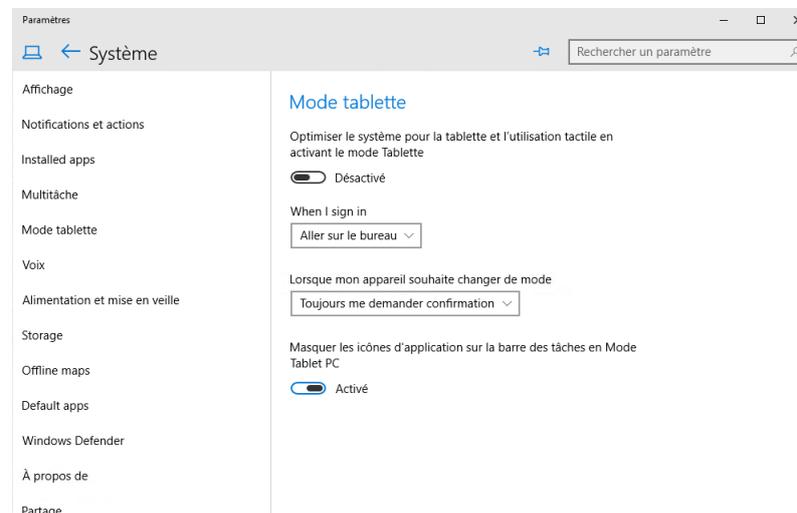


5.9 Mode continu

En détectant la présence ou l'absence d'un clavier, Windows 10 sait s'il s'exécute sur un ordinateur ou sur une tablette. L'affichage est alors optimisé pour obtenir la meilleure expérience utilisateur possible. La build 10074 améliore l'expérience utilisateur en mode tablette. Ainsi, la fermeture d'une application renvoie automatiquement sur l'écran **Démarrer**.

En utilisant la fenêtre **Paramètres**, vous pouvez :

- optimiser le rendu sur tablette ;
- choisir le mode d'affichage à l'ouverture de chaque session ;
- définir le comportement lors du passage du mode PC au mode tablette et inversement ;
- déterminer si les icônes des applications sont affichées en mode tablette :



5.10 Pour terminer

Le multitâche a été amélioré. Il en découle un plus grand sentiment de fluidité, y compris dans une machine virtuelle HyperV ou VirtualBox. De nombreuses améliorations d'ordre cosmétique ont également été apportées :

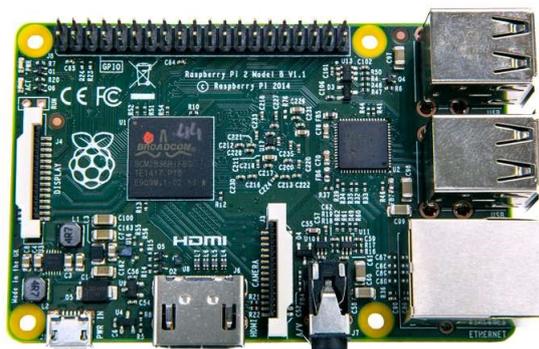
- animations des vignettes dans le menu Démarrer ;
- nouveau jeu de sons système ;
- Panneau de configuration ;

- évolution des apps Musique, Vidéo et Xbox et du Windows Store ;
- accès facilité à Windows Defender depuis la zone de notifications.

Pour les prochaines builds, Microsoft travaille à l'amélioration :

- du module de connexion réseau sans fil, qui permettra de paramétrer la connexion sans passer par un écran de paramétrage moderne ;
- de l'application moderne Photos, qui donne accès aux photos locales et à celles stockées dans OneDrive ;
- de la reconnaissance de l'écriture manuscrite ;
- de l'écran de verrouillage où des conseils d'utilisation pourraient être affichés.

Sachez également qu'à terme, Windows 10 sera multiplateforme : il pourra s'exécuter sur des PC, des tablettes, des smartphones, des objets connectés et des nano-ordinateurs tels que le Raspberry Pi 2 modèle B. Quel que soit votre ordinateur, vous ferez donc le plein d'applications sur un store unifié.



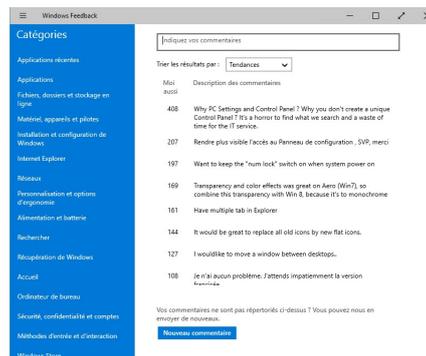
De nombreuses autres modifications verront le jour dans les mois à venir, mais vous savez d'ores et déjà grossièrement à quoi vous attendre...

5.11 Feedback, feedback, feedback

La commercialisation de Windows 10 est prévue pour la rentrée 2015 (en septembre ou un peu avant). D'ici là, amusez-vous bien avec Windows 10 Insider

Retrouvez l'article de **Michel Martin** en ligne : [lien 53](#)

Preview et n'hésitez pas à faire remonter vos observations en utilisant l'application Windows Feedback :



5.12 Avant de nous quitter

Depuis de nombreuses années, ma passion pour l'informatique et pour l'écriture m'a amené à écrire de nombreux ouvrages. Aujourd'hui, j'ai publié plus de 300 livres chez des éditeurs aussi prestigieux que Pearson, Sybex, Micro Application, Nathan, Bordas, Open Classrooms et PSI.

En 2009, je me suis lancé dans l'édition numérique en créant la société Mediaforma Learning. Si vous avez apprécié cet eBook, peut-être qu'une autre de mes publications saura vous intéresser. Vous pouvez accéder à tous mes ouvrages en vous rendant sur le site immatériel : [lien 51](#).



Vous pouvez également me retrouver sur le site Mediaforma Learning ([lien 52](#)), où je publie tous les jours des articles et des vidéos sur plusieurs sujets autour de l'informatique.

Liste des liens

Page 3

- lien 1 : ... <http://oracle.com.edgesuite.net/timeline/java/>
- lien 2 : ... <http://www.developpez.net/forums/d1522069/java/general-java/java-20-ans/>

Page 4

- lien 3 : ... <http://cpp.developpez.com/cours/>
- lien 4 : ... <http://cpp.developpez.com/tutoriels/Explicit-C++/ecrire-algorithme-standard/>
- lien 5 : ... <http://cpp.developpez.com/tutoriels/fuites-memoires-erreurs/>
- lien 6 : ... <http://cpp.developpez.com/tutoriels/programmation-modulaire/introduction/>
- lien 7 : ... <http://cpp.developpez.com/tutoriels/programmation-modulaire/ligne-de-commande/>
- lien 8 : ... <http://cpp.developpez.com/tutoriels/programmation-modulaire/scripts//>
- lien 9 : ... <http://loic-joly.developpez.com/articles/5Mythes/>
- lien 10 : ... <http://alexandre-laurent.developpez.com/articles/debogage-application/>
- lien 11 : ... <http://cpp.developpez.com/index/redirect/15574/Tutoriel-Les-regles-d-or-de-la-programmation-par-mogwai162-et-Alexandre-Laurent/>
- lien 12 : ... <http://cpp.developpez.com/index/redirect/15192/Programmation-fonctionnelle-en-Cplusplus-un-article-de-John-Carmack-traduit-pas-Arzar/>
- lien 13 : ... <http://cpp.developpez.com/index/redirect/14824/Analyse-statique-de-code-un-article-de-John-Carmack-traduit-par-Arzar/>
- lien 14 : ... <http://cpp.developpez.com/index/redirect/17154/Verification-et-envoi-d-e-mail-en-Cplusplus-avec-boost-par-Neckara/>
- lien 15 : ... <http://cpp.developpez.com/actu/55438/Utilisation-conjointe-de-lambdas-et-fonction-un-billet-de-blog-de-germinolegrand/>
- lien 16 : ... <http://cpp.developpez.com/index/redirect/16329/Developper-en-C-ou-Cplusplus-avec-l-environnement-Eclipse-un-tutoriel-de-Laurent-Barbareau/>
- lien 17 : ... <http://cpp.developpez.com/index/redirect/20357/Presentation-de-libclang-libtooling-une-API-pour-manipuler-du-code-source-par-David-Come/>
- lien 18 : ... <http://cpp.developpez.com/index/redirect/19312/Compte-rendu-evenement-Windows-App-special-3D-Minko-Project-Anarchy-Babylon-js-et-Oculus-Rift-a-l-honneur/>
- lien 19 : ... <http://jeux.developpez.com/tutoriels/OpenGL-moderne/>
- lien 20 : ... <http://jeux.developpez.com/tutoriels/OpenGL-ogldev>
- lien 21 : ... <http://jeux.developpez.com/tutoriels/directx-11-rastertek/>
- lien 22 : ... <http://cpp.developpez.com/index/redirect/14802/Un-avant-gout-de-DirectX-11-tout-savoir-sur-la-onzieme-version-de-l-API-de-Microsoft-un-article-de-Gregory-Massal/>
- lien 23 : ... <http://cpp.developpez.com/index/redirect/18517/L-univers-Arduino-moins-partie-II-realisation-d-un-prototype-a-base-d-Arduino-un-tutoriel-de-f-leb/>
- lien 24 : ... <http://cpp.developpez.com/index/redirect/18107/L-univers-Arduino-moins-partie-I-aperçu-de-la-plateforme-Arduino-et-ses-cartes-d-interface-shields-un-tutoriel-de-f-leb/>
- lien 25 : ... <http://loic-joly.developpez.com/articles/TechDays2015/>
- lien 26 : ... <http://intel.developpez.com/tutoriels/gcc/nouvelles-optimisations-x86-gcc-5-0/vectorisation-sections-load-store/>
- lien 27 : ... <http://intel.developpez.com/tutoriels/gcc/nouvelles-optimisations-x86-gcc-5-0/amelioration-pic-mode-32-bits/>

Page 5

- lien 28 : ... <http://akrzemi1.developpez.com/metaprogrammation-metafonctions-cpp-11/>
- lien 29 : ... <http://www.developpez.net/forums/d1519703/c-cpp/cpp/nouvelle-mise-jour-page-cours-cpp-plus-25-nouveaux-tutoriels-publies-bien-d-autres-venir/>
- lien 30 : ... <http://android.developpez.com/cours/>
- lien 31 : ... <http://www.developpez.net/forums/d1522538/java/general-java/java-mobiles/android/page-cours-android-nouvelle-mise-jour/>

Page 15

- lien 32 : ... <http://fauconnier.ftp.developpez.com/tutoriels/excel-enregistreur-macro/>

Page 22

lien 33 : ... http://denishulo.developpez.com/tutoriels/access/simulation-process-industriel/fichiers/simuler_process.zip

lien 34 : ... <http://denishulo.developpez.com/tutoriels/access/simulation-process-industriel/>

Page 23

lien 35 : ... <https://jsfiddle.net/fdg88cdy/>

Page 24

lien 36 : ... <https://jsfiddle.net/msp2no5s/>

lien 37 : ... <http://www.developpez.net/forums/blogs/243304-bovino/b530/comprendre-delegation-devenement-javascript/>

Page 26

lien 38 : ... <https://medium.com/javascript-scene/how-to-fix-the-es6-class-keyword-2d42bb3f4caf>

Page 27

lien 39 : ... <https://github.com/baconjs/bacon.js/issues/296>

Page 28

lien 40 : ... http://fr.wikipedia.org/wiki/Duck_typing

lien 41 : ... <http://www.paulirish.com/2015/advanced-performance-audits-with-devtools/>

Page 31

lien 42 : ... <https://medium.com/javascript-scene/the-two-pillars-of-javascript-ee6f3281e7f3>

lien 43 : ... <http://www.amazon.com/gp/product/0201633612?ie=UTF8&camp=213733&creative=393185&creativeASIN=0201633612&linkCode=shr&tag=eejs-20&linkId=QYF6ABRMZ406KML2>

lien 44 : ... <http://www.amazon.com/gp/product/0201633612?ie=UTF8&camp=213733&creative=393185&creativeASIN=0201633612&linkCode=shr&tag=eejs-20&linkId=QYF6ABRMZ406KML2>

Page 32

lien 45 : ... <https://medium.com/javascript-scene/how-to-fix-the-es6-class-keyword-2d42bb3f4caf>

lien 46 : ... <http://ericleads.com/2013/02/fluent-javascript-three-different-kinds-of-prototypal-oo/>

lien 47 : ... <https://github.com/ericelliott/stampit>

lien 48 : ... <http://sylvainpv.developpez.com/traductions/javascript/idees-fausses-heritage/>

Page 33

lien 49 : ... http://windows.microsoft.com/fr-fr/windows/preview-download?ocid=tp_site_downloadpage

Page 37

lien 50 : ... <https://www.virtualbox.org/wiki/Downloads>

Page 48

lien 51 : ... <https://www.7switch.com/fr/list/author-145722-michel-martin-mediaforma/page/1/date>

lien 52 : ... <http://www.mediaforma.com/>

lien 53 : ... <http://michel-martin.developpez.com/tutoriels/windows-10/telecharger-installer-apprendre-windows-10/>