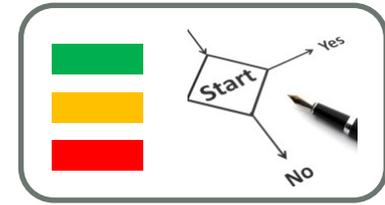


VÉRIFICATION DE VOTRE INSTALLATION VISUAL STUDIO POUR DEV MOBILE

Test Projet Cordova

Test Projet Xamarin crossplatform, android,

Les installation pour le cours Xamarin



- Sélectionnez l'ensemble des composants du package « Développement d'applications Windows et web » - « Microsoft Web Developer Tools »
- Sélectionnez l'ensemble des composants du package « Développement multi plateforme pour appareils mobiles »

ENVIRONNEMENTS LOGICIELS POUR L'INFORMATIQUE MOBILE

CROSS DÉVELOPPEMENT



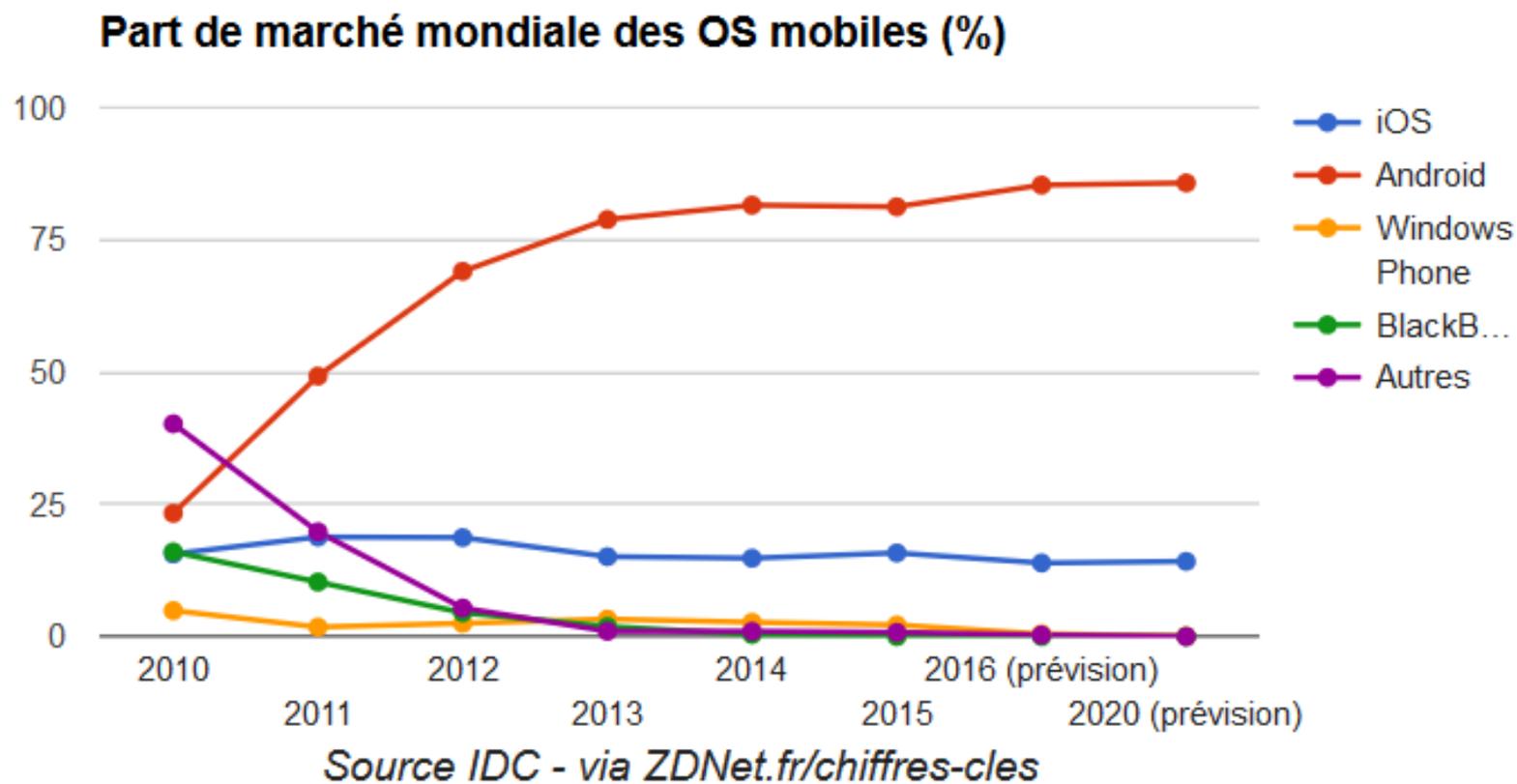
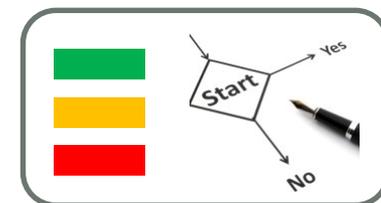
Jean-Yves Tigli, <http://www.tigli.fr>

Email : tigli@unice.fr

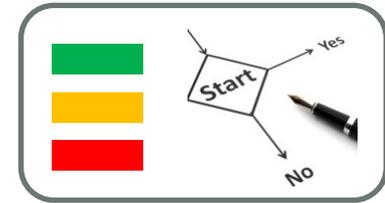
Page du cours :

https://www.tigli.fr/doku.php?id=cours:plim:plim_2017_2018

Rappel sur la diversité des plateformes cibles

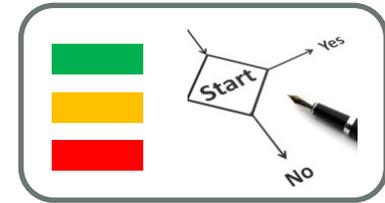


LE PARADOXE DU DÉVELOPPEMENT POUR MOBILE



- Les terminaux mobiles sont des cibles logicielles très **hétérogènes** de par leurs OS, leurs frameworks liés à leurs spécificités matérielles.
- **Pourquoi ?**
- Alors que les constructeurs matériels cherchent toujours plus d'originalité pour rendre leurs produits attractifs, les constructeurs logicielles quant à eux essayent de factoriser au plus les développements.

LE PARADOXE DU DÉVELOPPEMENT POUR MOBILE

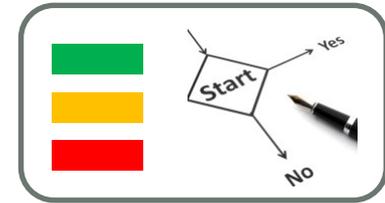


Différentes approches de développement

1. le développement très générique type HTML5 (Web)
2. le développement hybride (WebApp)
3. le cross développement
4. le développement natif, très spécifique lié à chaque OS et API

Tous ces choix dépendent des contraintes de l'application développée et des accès plus ou moins spécifiques aux fonctionnalités du smart phone.

DANS CE COURS



Différentes approches de développement

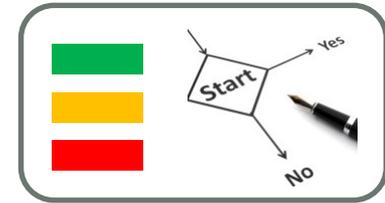
1. Exemple de responsive sur CSS3 / HTML5 (Web)
2. Exemple de développement hybride (WebApp) avec Cordova
3. Exemple de cross développement avec Xamarin
4. Exemple de développement en code natif,
 1. Sur Android
 2. Sur iOS

Soit dans tous les cas 99% des pratiques pour le développement logiciel sur Smart Phone

DÉVELOPPEMENT D'APPLICATION WEB

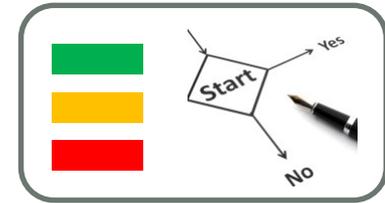
Ex. CSS3 - HTML5

Développement d'applications Web



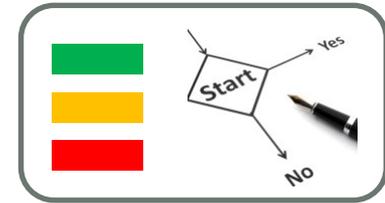
- Une web application est une application mobile développée en HTML accessible et exécutable par le biais d'un navigateur Internet pour téléphone mobile.
- Elle utilise le navigateur du smartphone et ne nécessite pas forcément de télécharger l'application.
- Elle est accessible par tous les smartphones quelques soient leur marque et système d'exploitation.
- La web application « complète » les applications natives qui doivent être téléchargées et installées par les mobinautes.

Développement d'applications Web



- Cependant une web application ne prend pas en considération les différents modèles de smartphones et les différents systèmes d'exploitation.
- Elle n'est donc pas toujours ergonomique (résolution plus mauvaise sur certains smartphones que sur d'autres, interface non adaptée, etc.).
- Elle n'utilise pas la mémoire embarquée du smartphone et est du coup beaucoup moins performante qu'une application native (Temps de chargement beaucoup plus long lors de l'activation des widgets, beaucoup moins de fluidité, etc.).

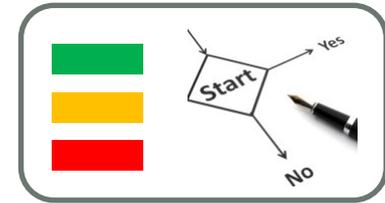
Browsers Web sur Smart Phone



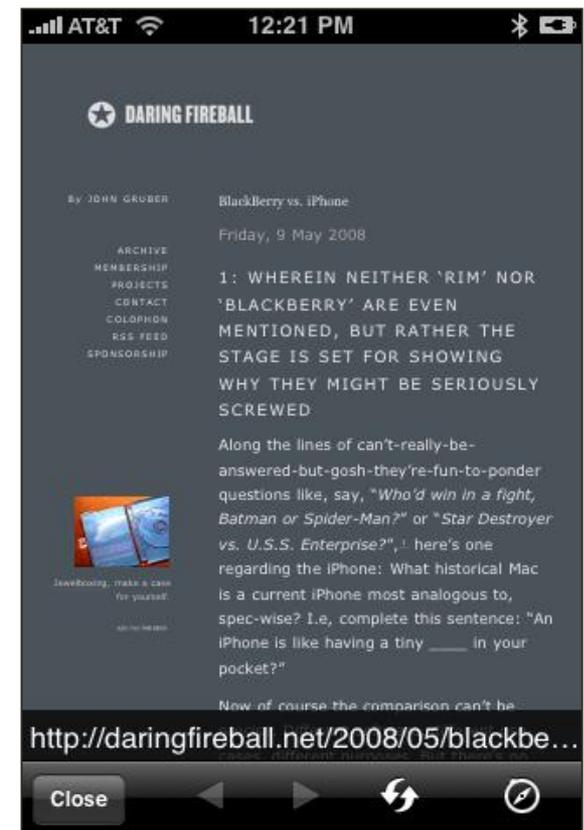
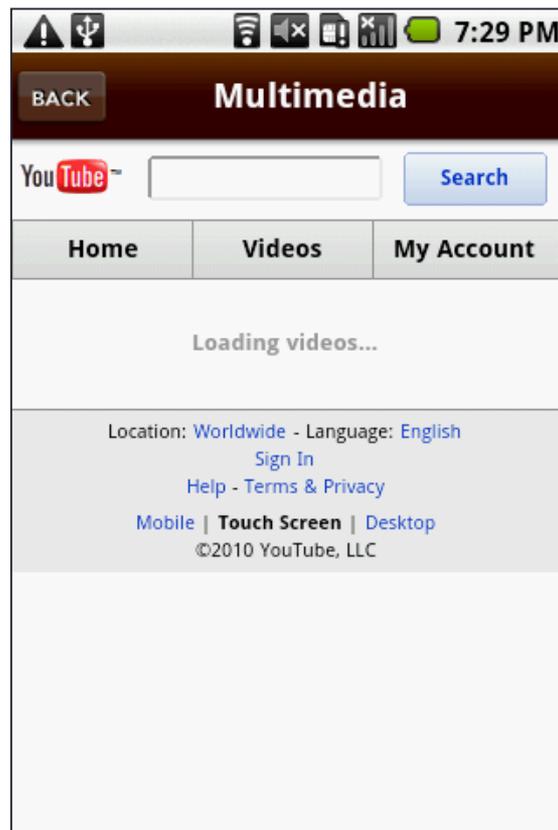
- Tous les Smart Phones disposent d'un Browser Web qui supporte de mieux en mieux les spécification HTML5/CSS3

Mobile OS	Browser
Android	Webkit based
iPhone	Webkit based
BlackBerry 6.0 +	Webkit based
Window Phone 7	IE 7 based *
WebOS	Webkit based
Nokia	Webkit based

WebViews



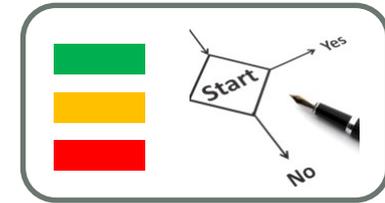
- La plupart des Smart Phones permettent d'utiliser ces browsers Web comme des vues embarquées ou WebViews.



WEB RESPONSIVE DESIGN

CSS3 HTML5

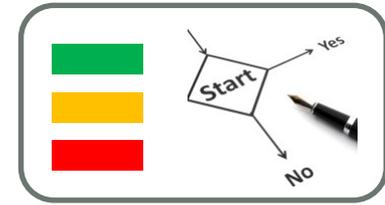
Web Responsive Design



- Au-delà des limitations quant à l'accès au matériel spécifique
- Problème des différentes capacités et caractéristiques des terminaux pour l'affichage de page Web

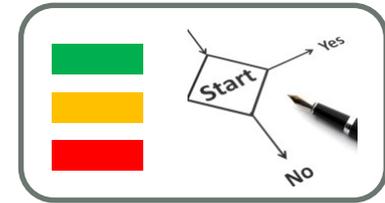


Responsive Design et caractéristiques des Ecrans



- Smart Phones : <http://screensiz.es/phone>
- Tablets : <http://screensiz.es/tablet>
- Moniteurs : <http://screensiz.es/monitor>

Responsive Web Design

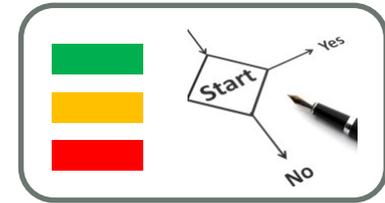


- Le terme de "Responsive Web design" a été introduit par Ethan Marcotte dans un article de [A List Apart](#) publié en mai 2010.
- Il décrira par la suite sa théorie et pratique du responsive dans son ouvrage ["Responsive Web Design"](#) publié en 2011.
- Celle-ci se limite à des adaptations côté client (grilles flexibles en pourcentages, images fluides et CSS3 Media Queries).

EXEMPLE CSS3 / HTML5

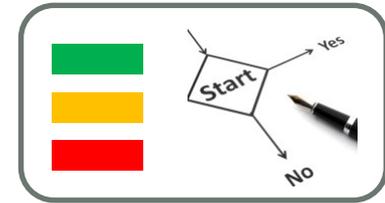
Ca marche aussi pour les Smart Phone si le Browser Web est le bon ...

Web Responsive Design



- Techniquement, le responsive web design se base sur les Media Queries.
- C'est une fonctionnalité offerte par le langage CSS, qui permet de mettre en page les sites web.
- **Concepts de base**
- Trois principales caractéristiques techniques sont au cœur de la conception de Web sensibles:
 - Media Queries et Media Queries listeners
 - Une mise en page à base de grille flexible qui utilise un dimensionnement relatif
 - images et médias flexibles, par le biais de redimensionnement dynamique ou CSS

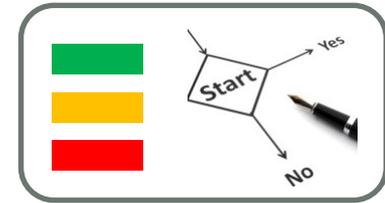
Media Queries



- Depuis CSS 2.1, les types media étaient utilisés pour appliquer des styles CSS différents pour screen et print.
Exemple :

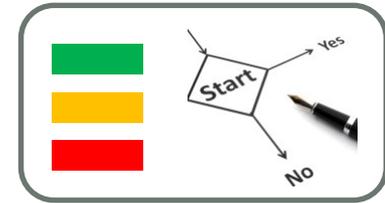
```
<link rel="stylesheet" type="text/css" href="style.css" media="screen" />  
<link rel="stylesheet" type="text/css" href="printfriendly.css" media="print" />
```
- Depuis CSS3, vous pouvez utiliser des requêtes des médias à des styles de portée des capacités spécifiques, en appliquant des styles différents en fonction des capacités qui correspondent à votre requête. Vous pouvez même combiner les requêtes qui testent plusieurs fonctionnalités en utilisant des opérateurs sémantiques tels que AND et NOT). Les caractéristiques comprennent la largeur, la hauteur, max-width, max-hauteur, dispositif-hauteur, orientation, rapport d'aspect, la résolution et plus.

Rappel CSS



- Les styles en cascade
- Il est possible de définir plusieurs styles en utilisant les différents moyens qu'offrent les CSS.
- Ainsi, lorsque plusieurs feuilles de style externes sont appelées, on obtient ce que l'on appelle une cascade de styles, c'est-à-dire une combinaison de styles pour divers éléments HTML.
- Si plusieurs styles concernent le même élément, seul le dernier style sera conservé.
- `<LINK rel=stylesheet type="text/css" href="style1.css">`
- `<LINK rel=stylesheet type="text/css" href="style2.css">`
- `<LINK rel=stylesheet type="text/css" href="style3.css">`

Style externe

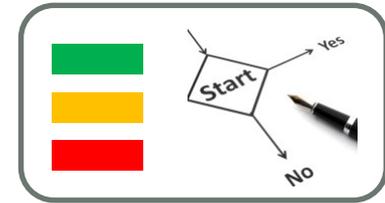


Le fait de pouvoir stocker la définition des feuilles de style à l'extérieur du document est un "plus" car il est ainsi possible, en modifiant le fichier contenant les feuilles de style, de changer l'allure de toutes les pages web s'y référant

Il s'agit dans un premier temps de créer un fichier texte contenant les feuilles de style et dont l'extension est .css, par exemple style.css :

```
<!--
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
body {background-image: home.gif;}
LI {font: 13px Verdana;}
B {font: 14px Verdana; font-weight: bold;}
A {
  font:12px Verdana;
  font-weight: bold;
  color=black;
  text-decoration: none;
}
H1 {font: 16px Arial;font-weight: bold;color=black;}
H2 {font: 14px Arial;font-weight: bold;color=black;}
-->
```

Style externe



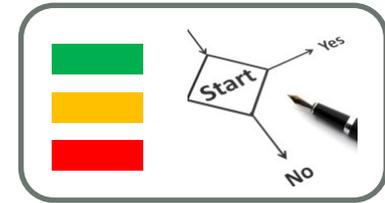
Dans un second temps il suffit de créer dans chaque page HTML le raccourci vers cette page de définition de style :

```
<HTML>  
<HEAD>  
  <LINK rel="stylesheet" type="text/css" href="style.css">  
</HEAD>
```

...

- La balise `<LINK>` avertit le navigateur qu'il doit chercher un document situé à l'extérieur de la page HTML.
- L'attribut `rel="stylesheet"` précise que le document en question est une feuille de style externe.
- L'attribut `type="text/css"` précise le type de feuille de style.
- L'attribut `href=" URL "` donne l'URL de la feuille de style, c'est-à-dire son emplacement sur Internet.

Style importé



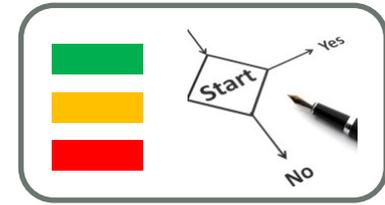
- Les recommandations du W3C offrent une dernière façon d'inclure des feuilles de style dans un document: en important des feuilles de style.
- Il est en effet possible d'importer des feuilles de style externes au niveau de la déclaration du style de document, en insérant la commande `@IMPORT` immédiatement après la balise `style`

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">  
<HTML>  
<HEAD>  
<STYLE type="text/css">  
<!--  
@IMPORT URL(url de la feuille à importer);
```

Définition des styles du document;

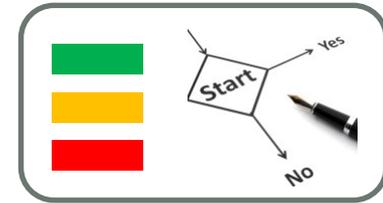
```
-->  
</STYLE>  
</HEAD>  
<BODY></BODY>  
</HTML>
```

Style externe : Appliquer une Media Query



- Les media queries sont donc des règles qui indiquent quand on doit appliquer des propriétés CSS. Il y a deux façons de les utiliser :
 - en chargeant une feuille de style .css différente en fonction de la règle (ex : « Si la résolution est inférieure à 1280px de large, charge le fichier petite_resolution.css »)
 - en écrivant la règle directement dans le fichier .css habituel (ex : « Si la résolution est inférieure à 1280px de large, charge les propriétés CSS ci-dessous »).

Style externe : Appliquer une Media Query



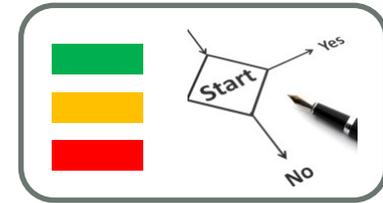
- Chargement d'une feuille de style différente
- Vous vous souvenez de la balise `<link />` qui permet, dans notre code HTML, de charger un fichier .css ?

```
<link rel="stylesheet" href="style.css" />
```

- On peut lui ajouter un attribut `media`, dans lequel on va écrire la règle qui doit s'appliquer pour que le fichier soit chargé. On dit qu'on fait une « requête de media » (media query en anglais). Voici un exemple :

```
<link rel="stylesheet" media="screen and (max-width: 1280px)"  
      href="petite_resolution.css" />
```

Style externe : Appliquer une Media Query



- Au final, votre code HTML pourrait proposer plusieurs fichiers CSS : un par défaut (qui est chargé dans tous les cas) et un ou deux autres qui seront chargés en supplément uniquement si la règle correspondante s'applique.

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<link rel="stylesheet" href="style.css" /> <!-- Pour tout le monde -->
```

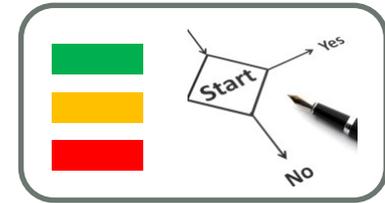
```
<link rel="stylesheet" media="screen and (max-width: 1280px)" href="petite_resolution.css" /> <!-- Pour ceux qui ont une résolution inférieure à 1280px -->
```

```
<title>Media queries</title>
```

```
</head>
```

```
...
```

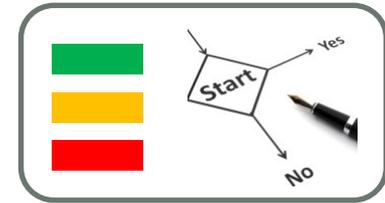
Style externe : Appliquer une Media Query



- Chargement des règles directement dans la feuille de style
- Une autre technique, que je préfère personnellement pour des raisons pratiques, consiste à écrire ces règles dans le même fichier CSS que d'habitude.
- Dans ce cas, on écrit la règle dans le fichier .css comme ceci :

```
@media screen and (max-width: 1280px)
{
  /* Rédigez vos propriétés CSS ici */
}
```

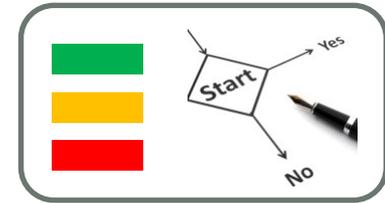
Des exemples : Media Queries



- http://www.w3schools.com/css/tryit.asp?filename=tryresponsive_mediaquery
- **Example**
- Si la fenêtre du browser est plus petite que 500px, la couleur de fond change pour lightblue:

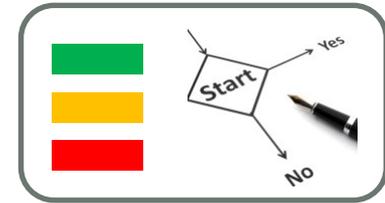
```
@media only screen and (max-width: 500px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

Méthodologie pour le Web Responsive Design



- A chaque fois que vous réorganisez les blocs de votre design est appelé un point de rupture (breakpoint).
- Cela ne correspond pas forcément à une largeur dite "tablette" ou "mobile" (car il existe plusieurs largeurs type pour des tablettes et des mobiles).
- Ces points de rupture dépendent complètement de votre site. C'est votre webdesigner qui doit vous indiquer à quel moment ces points de rupture doivent être créés.

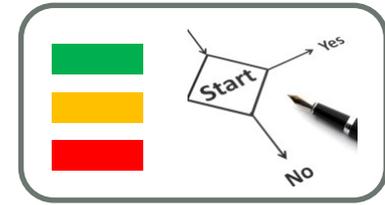
Des exemples : points de rupture



- Exemples :
 - Sans Media Queries :
 - http://www.w3schools.com/css/tryit.asp?filename=tryresponsive_styles
 - Avec Media Queries :
 - http://www.w3schools.com/css/tryit.asp?filename=tryresponsive_breakpoints
- Quand la fenêtre du Browser est plus petite que 768px, chaque colonne prend une taille de 100% :

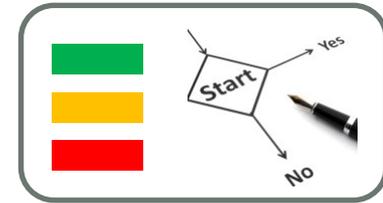
```
@media only screen and (max-width: 768px) {  
  /* For mobile phones: */  
  [class*="col-"] {  
    width: 100%;  
  }  
}
```

Site de test pour le Responsive Design et Manipulations



- Responsinator
- <http://www.responsinator.com/>
- Faite glisser le lien « Responsinator » de la page <http://www.responsinator.com/about/> vers votre barre de signets
- Dés lors vous pourrez tester n'importe quelle url chargée en cliquant sur ce lien ...
- `javascript:location.href='http://www.responsinator.com/?url='+window.location.href`
- Exemple de Responsive Design avec :
- http://www.htmlgoodies.com/imagesvr_ce/3028/mq_demo.html

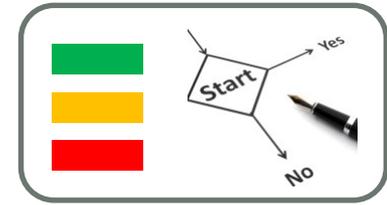
Site de test pour le Responsive Design et Manipulations pour Apple



- Et pour Iphone, Ipad ..etc.
- <http://lab.maltewassermann.com/viewport-resizer/>
- Faite glisser le lien « Click or Bookmark » de la page vers votre barre de signets
- Dés lors vous pourrez tester n'importe quelle url chargée en cliquant sur ce lien ...
- Exemple de Responsive Design avec :
- http://www.htmlgoodies.com/imagesvr_ce/3028/mq_demo.html

DÉVELOPPEMENT D'APPLICATIONS HYBRIDES

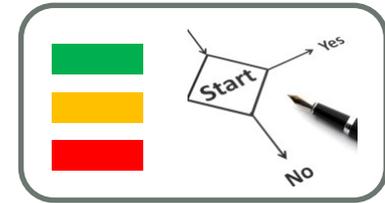
Développement d'applications Hybrides



Les applications hybrides

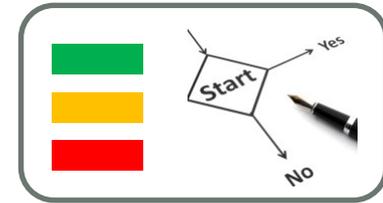
- Implémentées en HTML, CSS et Javascript
- Pas de réécriture de code à chaque changement de plate-forme
- Plusieurs framework permettent leur création et leur déploiement
- Cordova (...Phonegap...), Titanium, Sencha et JQuery mobile

Développement d'applications hybrides



- Une application hybride combine des éléments HTML5 sous forme de web application et des éléments d'une application native
- Ceux-ci permettent d'utiliser les fonctionnalités natives des smartphones.
- De plus elle pourra être distribuée en tant qu'application sur les plateformes d'applications (App Store, Android Market, etc.).
- Le principe de l'application hybride permet de réduire les coûts et délais de développement.

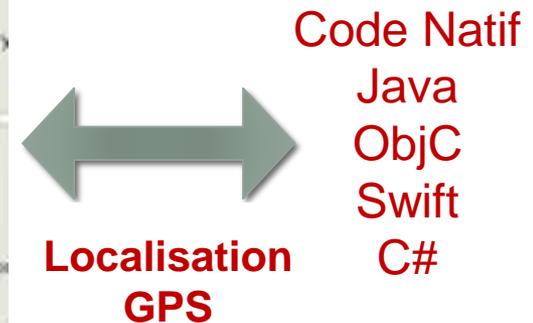
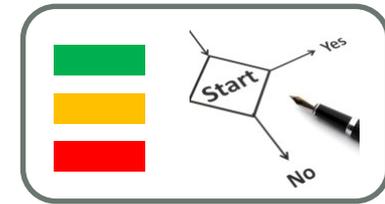
Développement d'applications hybrides



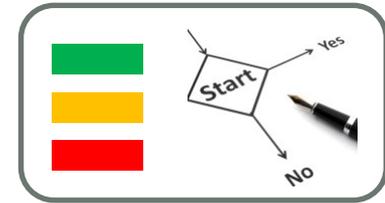
- Cependant la qualité, la performance, et la résolution de ces applications sont nettement inférieures à celles des applications natives.
- En effet l'application hybride peut ne pas bien s'adapter au système d'exploitation utilisé par le smartphone du mobinaute (interface polluée par des widgets inutiles, mauvaise résolution etc.).
- En plus les applications hybrides ne sont accessibles que sur iPhone et Android, et sont parfois refusées sur certaines plateformes d'applications.

Applications Hybrides

- Tous les browser web (principalement webkit) supportent Javascript pour appeler du code natif
- Les applications HTML/Javascript charge Google Maps et utilise du code natif pour accéder à la position GPS du Smart Phone

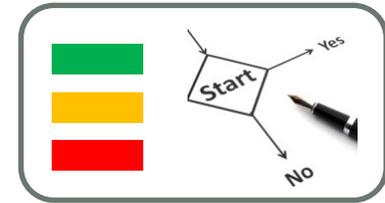


Framework Apache Cordova



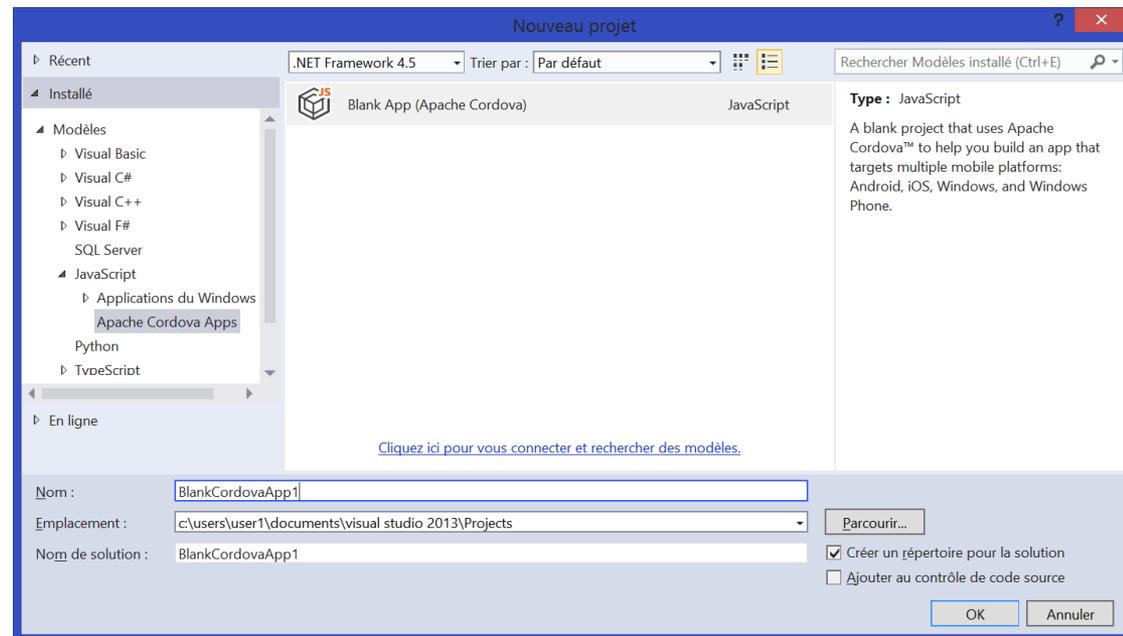
- Apache Cordova est un framework de développement mobile open-source.
- Il permet d'exploiter les technologies Web courantes telles que HTML5, CSS3 et JavaScript pour développer des applications multi-plateformes
- Il évite ainsi l'utilisation des langages natifs propres aux différentes plates-formes mobiles.
- Les applications s'exécutent dans des wrappers ciblés pour chaque plate-forme, elles s'appuient sur des API conformes aux standards permettant l'accès aux capteurs de chaque appareil, aux données ainsi qu'à l'état du réseau.

Apache Cordova sous Visual Studio (Android)

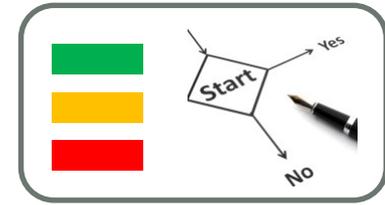


- Créer votre première application à l'aide de Visual Studio Tools for Apache Cordova
- <https://msdn.microsoft.com/fr-fr/library/dn757057.aspx#BuildHello>

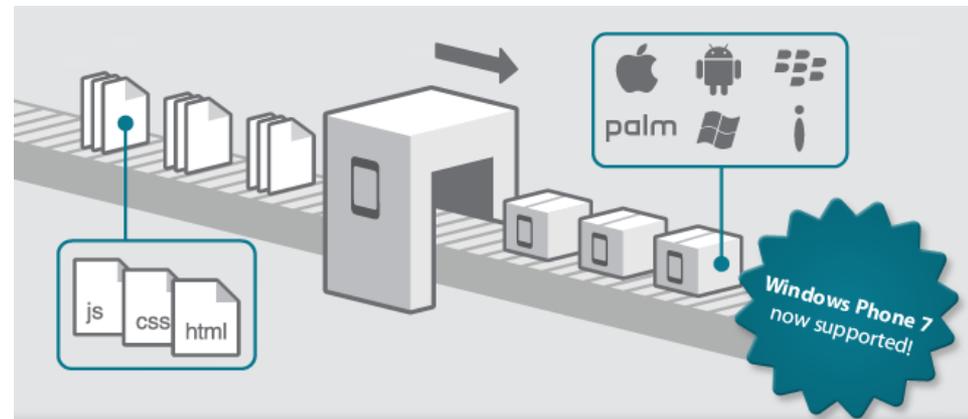
New Project
Javascript /
Apache Cordova Apps



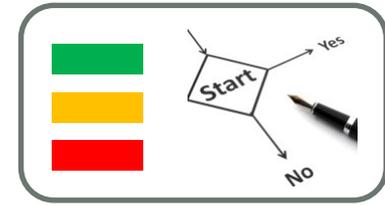
Autre distribution Cordova : Phonegap



- Framework open source de développement mobile
- Développé par la société Netiobi software, racheté par Adobe
- Utilise les langages javascript, Html5 et css3.
- Supporte plusieurs plateformes (IOS, Android, Symbian, Windows Phone, Blackberry OS)
- Bonne documentation
- Lenteur en comparaison aux applications natives
- Très difficile à débbugger

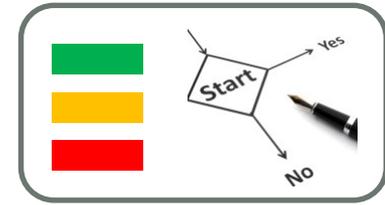


« Hello World » en Phonegap



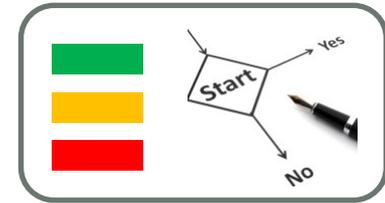
```
<title>Home</title>  
<script charset="utf-8" src="phonegap-1.2.0.js"  
type="text/javascript">  
</script>  
<form id="form1">  
<h1>  
Hello PhoneGap..!</h1>  
</form>
```

« Hello World » en Phonegap



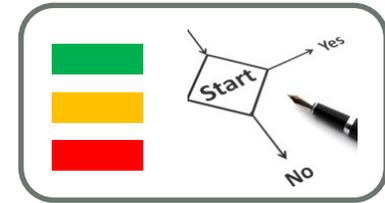
```
<title>Home</title>  
<script charset="utf-8" src="phonegap-1.2.0.js"  
type="text/javascript">  
</script>  
<form id="form1">  
<h1>  
Hello PhoneGap..!</h1>  
</form>
```

« Hello World » en Phonegap



```
<!DOCTYPE html>
<html>
<head>
<title>Mobilecore OW on Exit Example </title>
<script type="text/javascript" charset="utf-8" src="phonegap-1.2.0.js"></script>
<script type="text/javascript" charset="utf-8">
// Call onDeviceReady when PhoneGap is loaded.
// At this point, the document has loaded but phonegap-1.2.0.js has not.
// When PhoneGap is loaded and talking with the native device,
// it will call the event 'deviceready'.
function onLoad() {
    document.addEventListener("deviceready", onDeviceReady, false); }
// PhoneGap is loaded and it is now safe to call PhoneGap functions
function onDeviceReady() {
// Register the event listener
    document.addEventListener("backbutton", onBackKeyDown, false); }
// Handle the back button
function onBackKeyDown() {
    window.plugins.MobilecorePlugin.showOfferwall( function(r){navigator.app.exitApp();},
    function(e){console.log('error OW');
    })
}
</script>
</head>
<body onload="onLoad()">
</body>
</html>
```

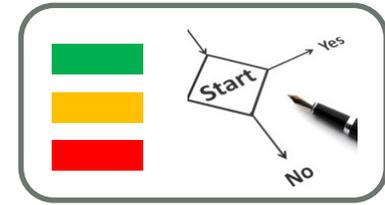
Phonegap : Compilation Online



- <https://build.phonegap.com/>

The screenshot shows the Adobe PhoneGap Build website. At the top, there is a navigation bar with the 'Bd' logo and links for 'Apps', 'Plugins', 'Docs', 'Blog', 'FAQ', and 'Support'. On the right side of the navigation bar, there is a user profile icon and the Adobe logo. The main content area features a large heading: 'Welcome to Adobe® PhoneGap™ Build!' followed by the sub-heading 'Let's get you started building an app.'. Below this, there is a 'Feedback' button and a 'our apps' link. A horizontal menu contains 'open-source', 'private', and 'Close' options. The main form area includes a text input field with the placeholder 'paste .git repo', an 'or' separator, and a button labeled 'Upload a .zip file'. Below the input field, there is a link to 'Connect your Github account'. At the bottom, a promotional message reads: 'Create more apps! Upgrade now to build up to 25 private apps.'

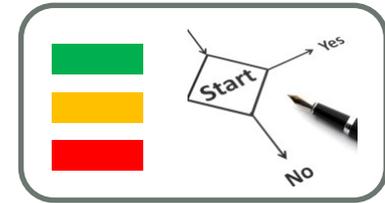
Paramétrage de son appli Phonegap



This screenshot shows the Adobe Developer Console interface for a PhoneGap application named "MonAppliPhoneGap". The top navigation bar includes "Bd", "Apps", "Plugins", "Docs", "Blog", "FAQ", "Support", and the Adobe logo. The main area displays the application name and a "Feedback" button. Below the name, there is a section for "Essai avec PhoneGaP" and a "Tip: Make your life easier by including config.xml in your project.more info". At the bottom, there are checkboxes for "Enable debugging" and "Enable hydration", along with "delete" and "Ready to build" buttons.

This screenshot shows the Adobe Developer Console interface for a PhoneGap application named "MonAppliPhoneGap". The top navigation bar includes "Bd", "Apps", "Plugins", "Docs", "Blog", "FAQ", "Support", and the Adobe logo. The main area displays the application name and a "Feedback" button. Below the name, there is a section for "Essai avec PhoneGaP" and a QR code. At the bottom, there are platform selection options (iOS, Android, Windows, BlackBerry, HP, and Info) and "Update code" and "Rebuild all" buttons.

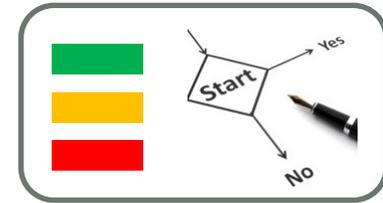
Tutorial Apache Cordova sous Visual Studio



- <https://msdn.microsoft.com/fr-fr/library/dn771545.aspx>
- On en profitera pour tester l'installation de VS 2015 et les add on Cordova et Xamarin

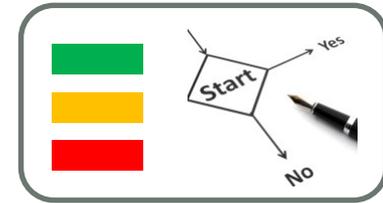
DÉVELOPPEMENT DES APPLICATIONS NATIVES

Développement d'applications natives



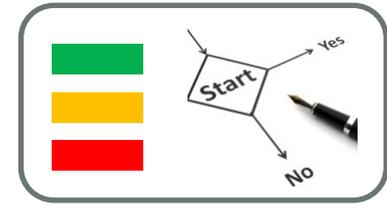
- développée avec un langage spécifique à son système d'exploitation
- distribuée uniquement par l'intermédiaire des plateformes d'applications qui contrôlent sa nature et ses contenus
- permet d'utiliser la mémoire du smartphone ainsi que toutes les fonctionnalités liées au système d'exploitation visé (GPS, accéléromètre, appareil photo, etc.).

Développement d'applications natives



- permet également de proposer des applications plus riches en terme de fonctionnalité, de qualité, de performance et de résolution que les web applications en HTML5 ou les applications hybrides.
- Une application native ne peut cependant être utilisée que par les mobinautes utilisant le système d'exploitation mobile visé
- Il convient donc de multiplier les applications natives si on souhaite toucher le plus grand nombre.

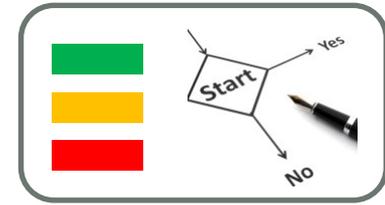
Développement d'applications natives



Les applications natives

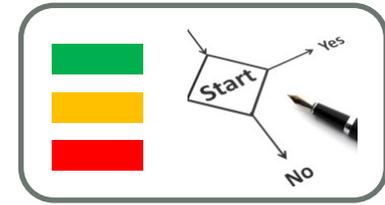
- Une approche classique pour implémenter une application
- java pour Android, Objective C et swift pour iOS, C# pour Windows phone 7
- Accès direct aux fonctionnalités de l'appareil : Téléphone, GPS, Camera, Carnet d'adresses, Agenda
- Nécessite de réécrire le code pour chaque plateforme

Les outils de développement



- Xcode pour IOS
- dispose de deux API de programmation :
 - Carbon qui permet de programmer avec les langages C et C++.
 - Cocoa qui permet de développer avec les langages Objective-C, AppleScript et Java. Fourni avec toute une suite logicielle (graphiques, audio, etc.).
- .Net Compact framework pour Windows phone
 - framework fonctionnant avec Visual studio en programmant avec C# ou Visual Basic.Net.
 - sous-ensemble de la bibliothèque .NET Framework
 - contient également des classes spécialement conçues
 - hérite la totalité de l'architecture de Common Language Runtime et d'exécution de code managé du .NET Framework.

Les outils de développement

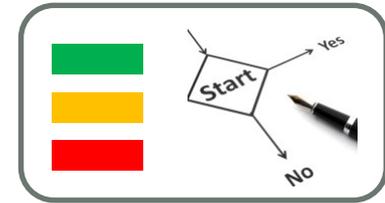


- Java pour Android
 - classes java spécifiques avec l'import du paquetage android.
 - Une application android contient six principaux éléments
 - Activity : couche visuelle de l'application.
 - Views : composants IHM de l'application.
 - Service : Permet l'exécution de l'algorithme sur un temps indéfini et terminé en fonction de la tâche.
 - Content Provider : fournit des données aux applications via un fournisseur de contenu capable de les partager avec d'autres applications (agenda, photos, contacts).
 - Intents : une application peut appeler un service ou une activité (explicite) ou un service du système android (implicite).
 - Broadcast Receiver : Reçoit les messages systèmes et les Intents implicites.



Exemple de plateforme de Cross
développement

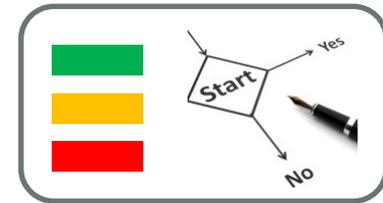
Historique



- Xamarin est une société américaine fondée en mai 2011
- S'appuie sur le développement de la technologie mono
- Mono est créé en tant que projet open source, en vue de proposer une implémentation de la plate-forme .Net sous UNIX

- En février 2016, Xamarin est rachetée par Microsoft
- Mise à disposition gratuite de Xamarin dans Visual Studio Community Edition.

Les cibles de Xamarin



iOS
watchOS
macOS
tvOS

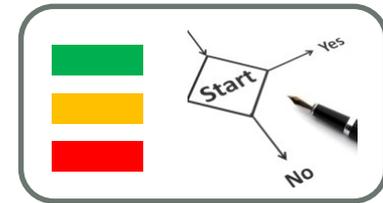


Desktop
Mobile
Xbox
HoloLens
IoT

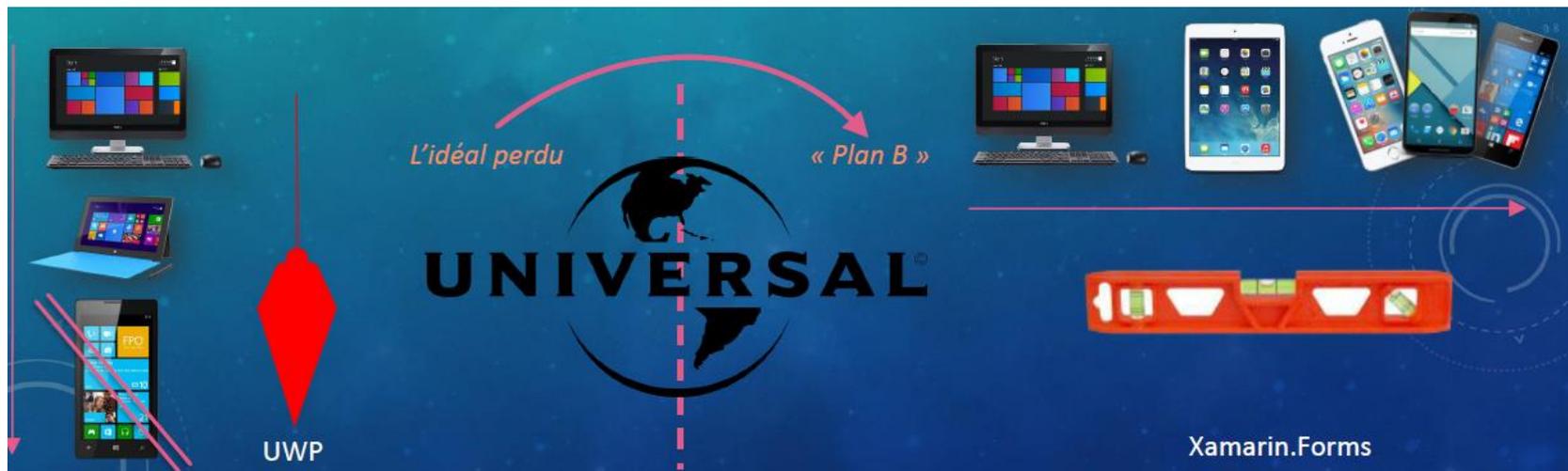


Android phone
Android tablet
Android wear

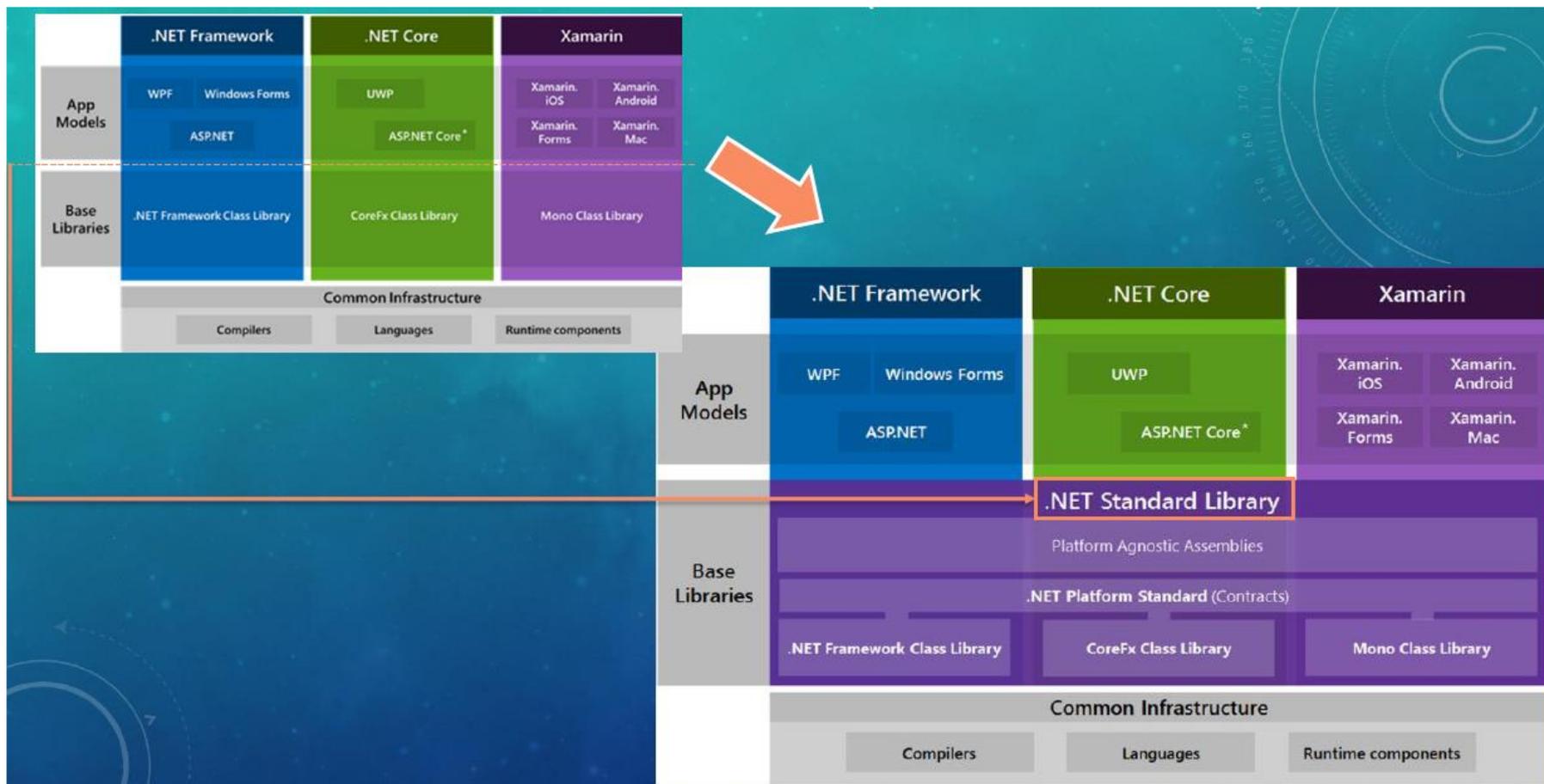
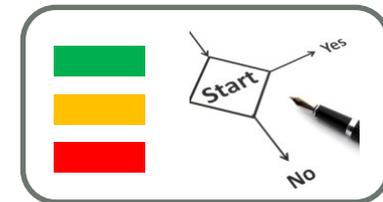
XAMARIN & STRATÉGIE MICROSOFT



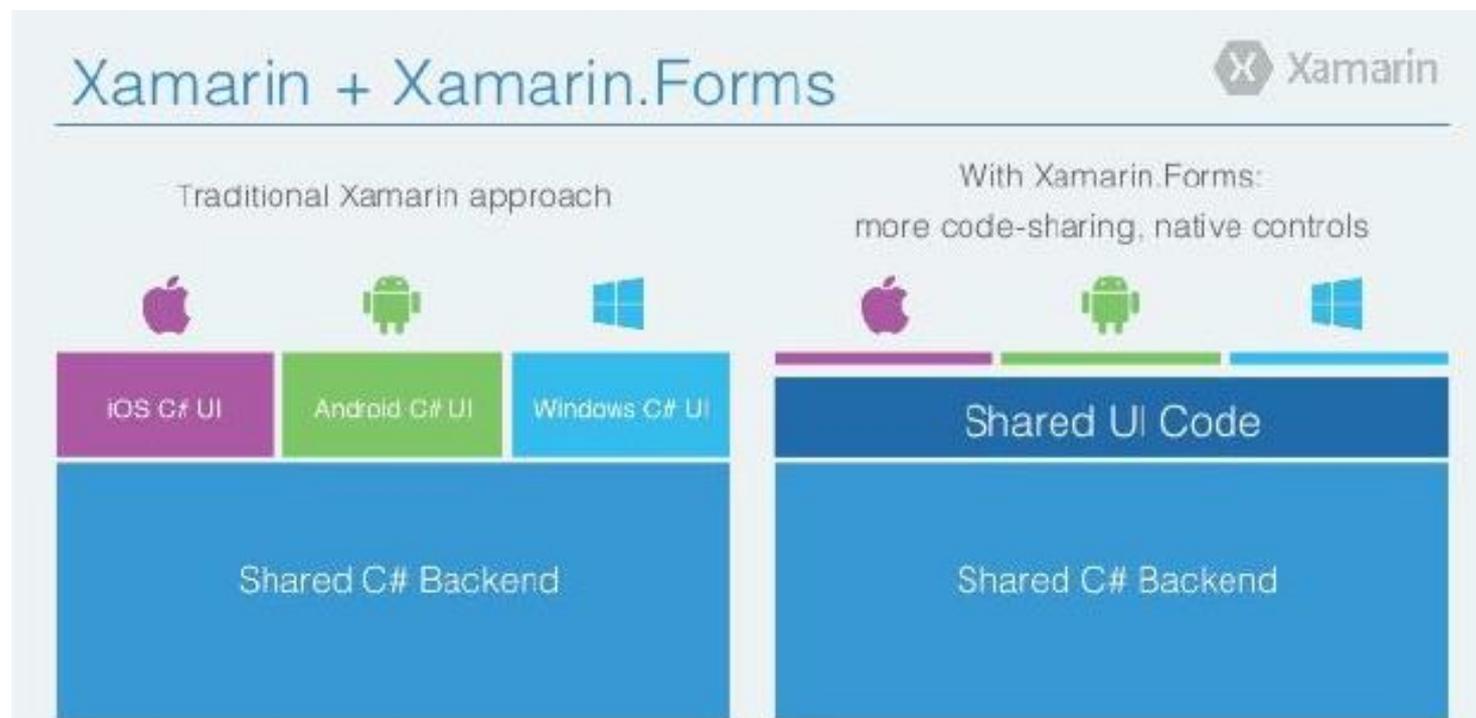
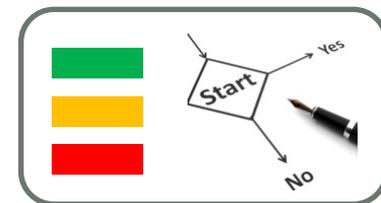
- Intégré à Visual Studio 2015-3
- Xamarin acquis par Microsoft cette année
- Plus qu'un ajout c'est la pierre angulaire d'une nouvelle stratégie Microsoft Cross-plateforme (comme .NET Core)



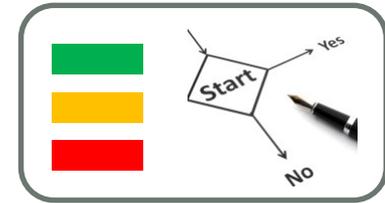
LA GRANDE UNIFICATION DES APIS (.NET STANDARD)



UNIFICATION DES UI : XAMARIN FORMS...



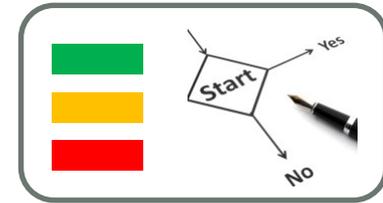
POURQUOI CHOISIR LES XFORMS ?



- UN SEUL CODE & UNE SEULE UI
- UNE SEULE COMPETENCE C# / .NET / XAML
- REUTILISATION DES COMPETENCES
- REUTILISATION DES OUTILS (VS, Resharper, Unit Testing...)
- REUTILISATION DES METHODES (MVVM...)

= GAIN DE TEMPS, d'ARGENT et ... PLUS DE PLAISIR !

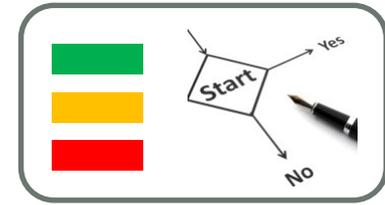
SI VOUS VENEZ DE WINDOWS FORMS !



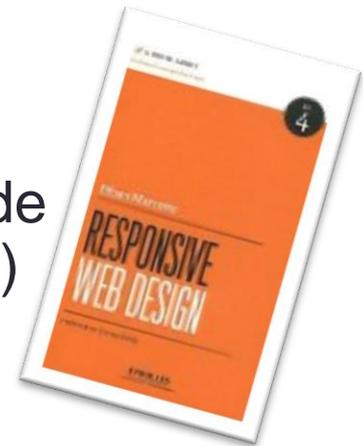
- Vous savez déjà 70% de ce qu'il faut connaître !
- Même EDI (VS ou Xamarin Studio)
- Même plateforme (.NET)
- Même langage (C#)
- Mêmes méthodes de travail (testing...)

- Si vous connaissez Silverlight, WPF ou WinRT / UWP vous êtes même à 90% !

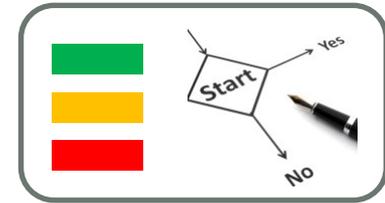
Références



- Responsive Web Design by Ethan Marcotte May 25, 2010 Published in CSS, Layout & Grids, Mobile/Multidevice, Responsive Design, Interaction Design, <http://alistapart.com/article/responsive-web-design>
- Responsive Web design, N°4 – 15 septembre 2011 de Ethan Marcotte (Auteur), Charles Robert (Traduction)
- Comparison of Cross-Platform Mobile Development Tools, Manuel Palmieri, Inderjeet Singh, Antonio Cicchetti, 2012, in 16th International Conference on Intelligence in Next Generation Networks. <http://www-sop.inria.fr/members/Inderjeet.Singh/06376023.pdf>



Références



- Creating Mobile Apps with Xamarin.Forms Book First Edition, Charles Petzold, Microsoft Press, <https://developer.xamarin.com/guides/xamarin-forms/creating-mobile-apps-xamarin-forms/>

