Web Mobile & PhoneGap

Julien Roche



Sensibilisation à la mobilité



La mobilité en quelques chiffres

- A l'heure actuelle, nous vendons plus de smartphones / tablettes que tous les autres supports réunis
- La moitié du débit Internet d'ici 2014 se fera uniquement sur les supports mobiles
- 30% des tweets se font depuis les smarpthones
- 70% des entreprises françaises ont une application mobile, ou en cours de réalisation



Un constat très simple





Mais je fais comment?



Natif





Web Mobile hybride





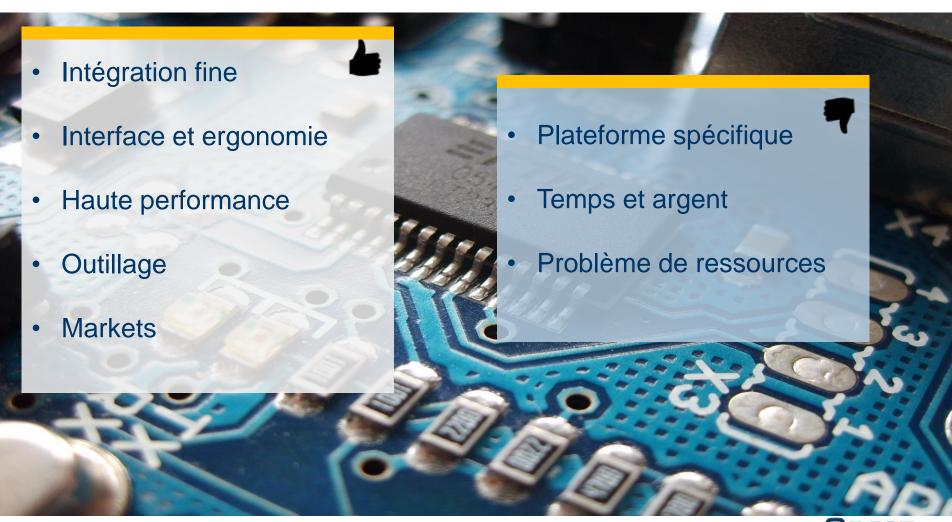
Web Mobile



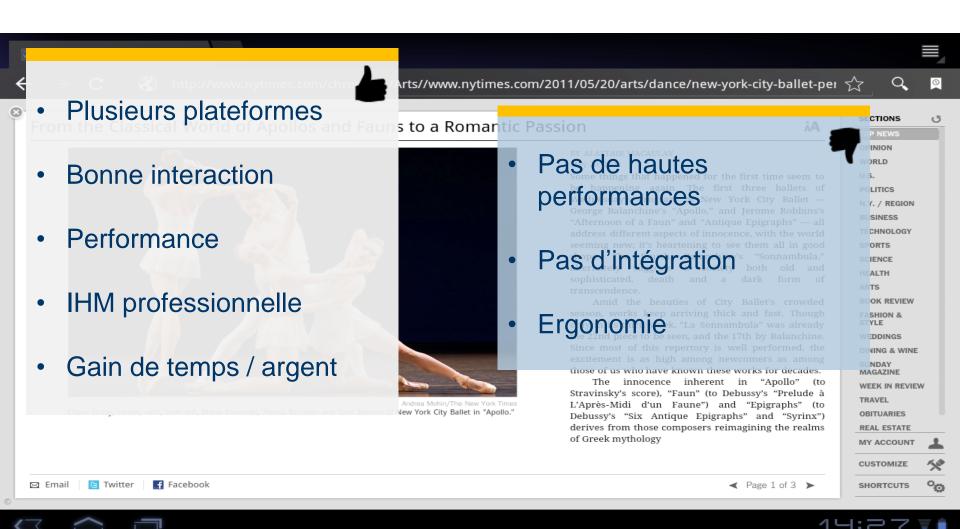




Et si on partait sur du natif?



Bon ok, et le Web Mobile?





En somme

Natif



- IHM
- Hautes performances
- Outillages
- Markets

Web Mobile



- Cross-platform
- Gain de temps / argents
- Ressources
- Nombreuses solutions



Et le Web Mobile hybride!

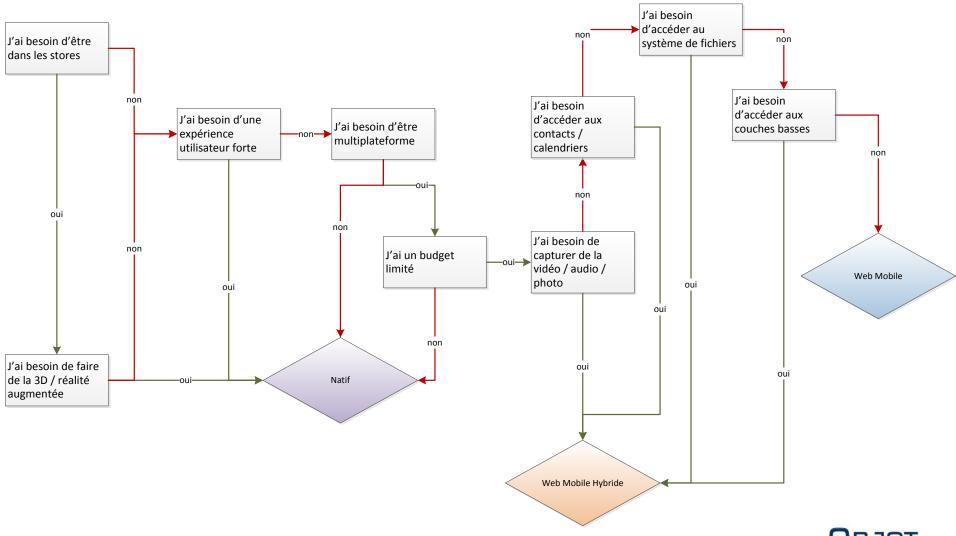


Permet de:

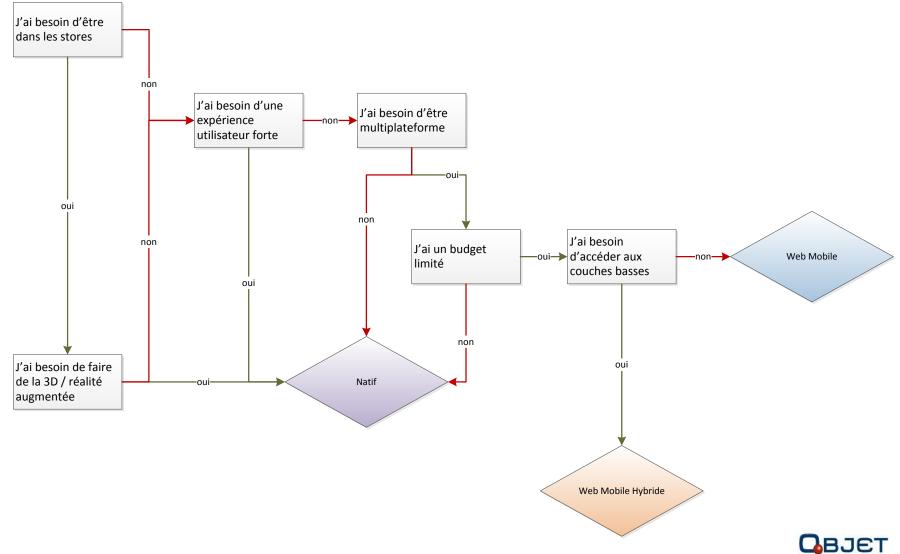
- Embarquer son application Web Mobile au sein d'une application native
- Bénéficier d'un pont entre le système et notre application Web Mobile
- Accéder aux markets



D'ici 2014



A partir de 2014



Quelles sont les usages courants ?

Jeux vidéos



App. Grand publique



App. professionnelle





Quelles sont les usages courants?

Jeux vidéos





App. Grand publique































App. professionnelle



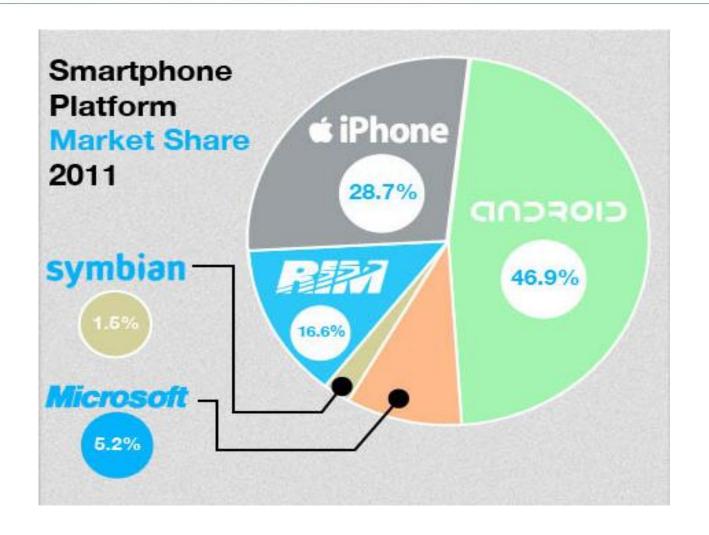


Forger son expérience mobile

D'un point de vue matériel et logiciel

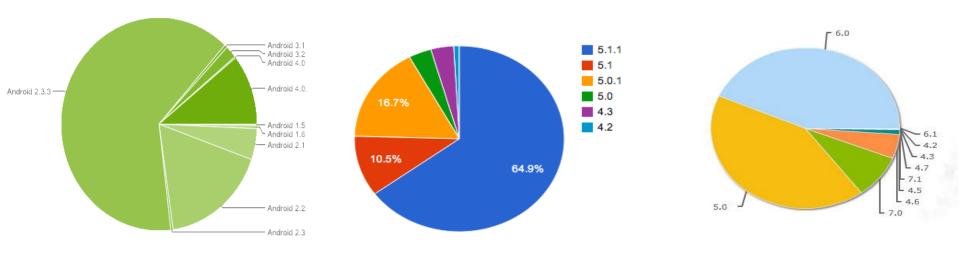


Quelles plateformes visées?





Quelles versions?



iOS



Blackberry

Android

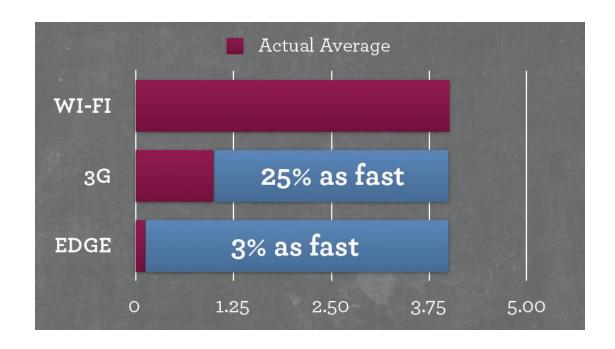
Attention aux contraintes!

Quelque soit votre solution, nous devez vous sensibiliser aux contraintes liées à la mobilité!





Tout d'abord, le débit!





Ensuite, les déconnexions!

Aujourd'hui, nous sommes mobiles:

- Car nous avons des supports pour cela
- Car nous accédons à l'Internet n'importe où

Du coup:

- Prendre en compte les déconnexions
- Prendre en compte le faible débit
- Bref, gestion du cache et des données!!





Pour gérer le cache

- Eviter de faire de trop nombreux appels aux serveurs
- Utiliser des formats de données léger comme JSON
- Utiliser des Web Services simples à consommer, comme REST
- Stocker vos données en locales
 - Données brutes
 - Images
 - Fichiers
 - ...





Les différents tailles des supports ...





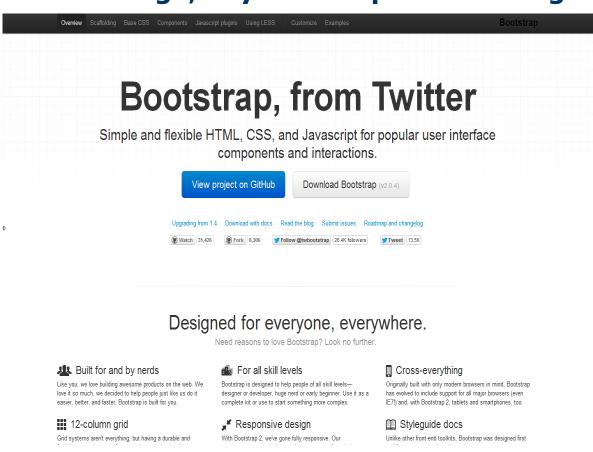
... et les prochaines!

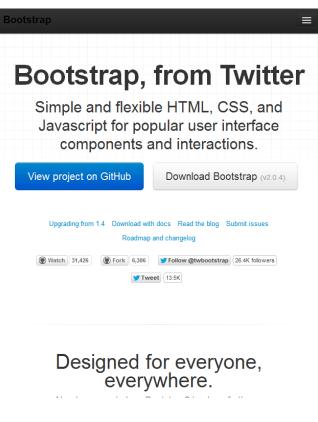




Réadapter votre contenu!

En fonction de la taille de ce support, n'offrez pas le même affichage, soyez « responsive design »!





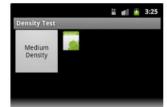


Attention à la densité des écrans!

- Un iPhone 4 a un largeur physique de 640px mais son viewport n'est plus que de 320px!
 - Autrement dit, 1 pixel affiché ne correspond pas forcement à un pixel physique
- Un élément d'interface utilisateur semble physiquement plus grand sur un écran de faible densité et inversement.
 - En fonction de la taille de l'écran, affiché une image correcte en conséquence !

Gestion de la densité mauvaise













Gestion de la densité correcte



Les autres contraintes que nous oublions souvent

La batterie

- Qui se décharge très rapidement si trop de services lancés et / ou de traitements lourd
 - Polling
 - GPS
 - Camera
 - ...



■ La mémoire

Ne lancer pas trop de services en tâche de fonds





Forger son expérience mobile

D'un point de vue humain



Les interactions utilisateurs (1)

- Plus de claviers, plus de souris, tout avec les doigts!
- Et un doigt fait environ 45 à 60 pixels de hauteur / largeur

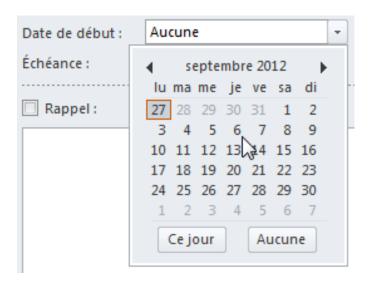


Pensez donc à adapter vos éléments en conséquence



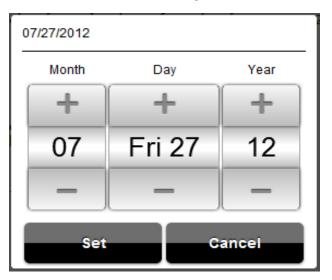
Les interactions utilisateurs (2)

Exemple avec un datepicker:



Sur un desktop

Sur un smartphone



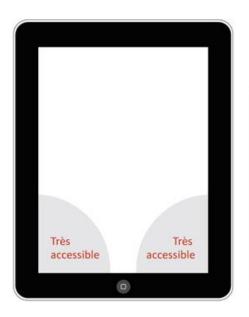
Beaucoup plus simple d'affichage et à utiliser!

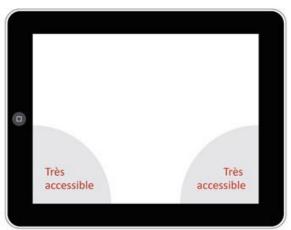


La disposition des éléments (1)

- Pensez à disposer vos éléments en fonction de la position des doigts de l'utilisateur
 - Sachant que ce dernier prend son smartphone à la main et fait la plupart des choses avec son pouce.









La disposition des éléments (2)

Cela indique où placer des éléments en fonction de leur criticité.

Photo Post

Cancel

What is this?

 Par exemple, un bouton qui doit confirmer la suppression de tous les contacts se mettra plutôt en haut pour ne pas être facilement

Publish

accessible.





Respecter les conventions d'interactions

Tap



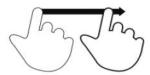
Briefly touch surface with fingertip

Double tap



Rapidly touch surface twice with fingertip

Drag



Move fingertip over surface without losing contact

Flick



Quickly brush surface with fingertip

Pinch



Touch surface with two fingers and bring them closer together

Spread



Touch surface with two fingers and move them apart

Press



Touch surface for extended period of time

Press and tap



Press surface with one finger and briefly touch surface with second finger

Press and drag



Press surface with one finger and move second finger over surface without losing contact

Rotate



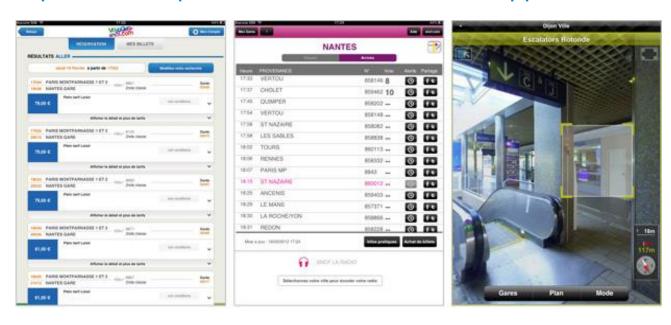
Touch surface with two fingers and move them in a clockwise or counterclockwise direction





Rester simple

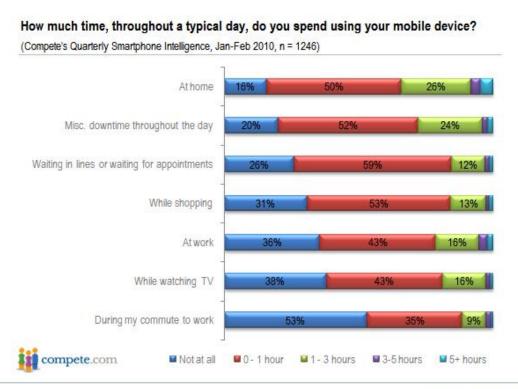
- Il faut mieux avoir peu de fonctionnalités, mais avoir une application extrêmement efficace
 - car nous n'avons pas autant d'aisance qu'avec un ordinateur
- Créer des applications spécifiques
 - Exemple: SNCF qui a créé une multitude d'applications concises





Demandez-vous où est votre client?

- En règle générale, les gens utilisent leurs smartphones principalement ... chez eux ! Et pendant les temps « d'arrêts »
 - http://blog.compete.com/2010/03/12/smartphone-owners-aready-and-willing-audience/





Web Mobile



I've seen the FUTURE It's in my BROWSER

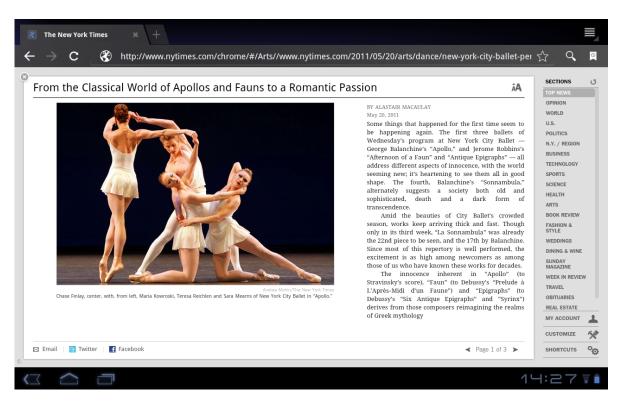


CSS3





Démo - New York Times



http://www.nytimes.com/chrome/

Problèmes du New York Times

- Mise en page équivalente à la version papier
- Accessible sur desktop et tablettes

Solution : CSS3

- Interface dynamique et professionnelle
- Interface s'adaptant à la résolution
- Réduction significative du code JavaScript



Définir une propriété sur les navigateurs

- Définition standardisée par le W3C
- Mais chaque navigateur a son équivalent en attendant la validation officielle:
 - Firefox: -moz-
 - Konqueror: -khtml-
 - Internet Explorer: -ms-
 - Opera: -o-
 - Webkit: -webkit-

- Il est recommandé de définir la propriété du W3C et la propriété des navigateurs
 - En fonction de la version du navigateur, celui-ci prendra soit sa propriété, soit la propriété du W3C



Les couleurs

Opacité

- Possibilité de rendre opaque / transparent nos éléments
- Propriété: opacity
- Valeurs: de 0 à 1

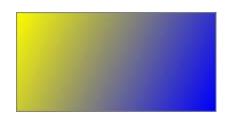
Définir une couleur

- RGB: définir une couleur rouge-vert-bleu
 - color: rgb(255, 255, 0)
- RGBA: définir une couleur RGB avec un alpha
 - color: rgba(255, 255, 0, 0.5)
- HSL: définir une couleur teinte-saturation-lumière
 - color: hsl(0, 100%, 100%) /* red color */
- HSLA: définir une couleur HSL avec un alpha
 - color: hsla(0, 100%, 100%, 0.5)

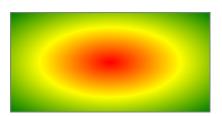


Les gradients

- Nous pouvons définir un gradient de couleur
- Nouvelle valeur pour les propriétés color et backgroundcolor:
 - linear-gradient: gradient linéaire
 - linear-gradient(top left, yellow, blue);



- radial-gradient: gradient par rapport à une zone et à une « opacité »
 - radial-gradient(red, yellow, green);
 - radial-gradient(bottom left, farthest-side, red, yellow 50px, green);







Les gradients

Parfois, Webkit préfère sa propre syntaxe

• Exemple:



-webkit-gradient(linear, left center, right center, from(#003567), to(#66A1C9))

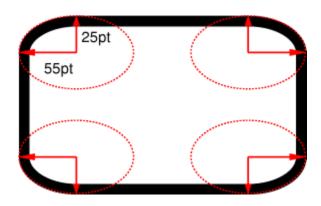
versus

linear-gradient(left center, right center, #003567, #66A1C9)



Les bordures arrondies

- Il est désormais possible de définir des bords arrondis pour nos éléments
- Propriétés:
 - border-radius:
 - valeur pour tous les coins ou une valeur par coin (en commençant par le haut gauche, puis haut droit, bas droit et bas gauche)
 - La valeur peut être soit uniforme (et donc représente le rayon d'un cercle) soit défini par une paire de valeur (représentant les rayons d'une ellipse)
 - border-top-left-radius, border-top-right-radius, border-bottom-right-radius, border-bottom-left-radius: définition par coin



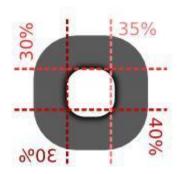


Les images en tant que bordure

Nous pouvons utiliser une image qui représentera la bordure de notre élément

Propriétés:

- border-image: lien sur l'image, avec la position du « découpage » sur l'image et définir si les portions découpées qui ne correspondent pas au coin devront être étirées (stretch) ou répétées (repeat)
- border-top-left-image, border-top-right-image, border-bottom-right-image, border-bottom-left-image



border-image: url(border.png) 30% 35% 40% 30% stretch stretch;

Fonds multiples

- Il est désormais possible de définir plusieurs images de fond pour un élément
- Utilisation de background-image: div { background-image: url(tl.png), url(tr.png) }
- Avec l'utilisation conjointe de background-position et background-repeat, en fonction de la taille de l'image, ces dernières peuvent toutes s'afficher:

```
div {
    background-image: url(tl.png), url(tr.png),
    background-repeat: repeat-y, repeat-y;
    background-position: top left, top right;
}
```



Media queries

Objectif: mettre des conditions sur un ensemble de règles CSS en fonction des propriétés des média

Comment l'appliquer

- Lors de l'import dans le fichier
 - rel="stylesheet" media="screen and (color)" href="example.css" />
- Lors de l'import CSS3
 - @import url(color.css) screen and (color);
- En tant que bloc
 - @media (min-width:500px) { ... }



Media queries - Propriétés

Propriétés des médias

- height, width
- device-height, device-width
- orientation
- aspect-ratio
- device-aspect-ratio
- color
- color-index
- monochrome
- resolution
- scan
- grid



Transformations 2D

- Nous pouvons manipuler graphiquement nos éléments
- Nous pouvons:
 - Faire une translation
 - Faire une rotation
 - Faire une homothétie
 - Pencher / obliquer
 - Combiner les points précédents
- Utilisation de la propriété
 - transform

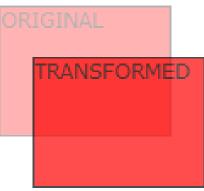


Transformation 2D - translation

■ Valeur de la propriété

- translate: nous devons préciser la translation sur l'axe X et sur l'axe Y
- translateX, translateY

```
div {
  transform: translate(50px, 100px);
}
```





Transformation 2D - rotation

■ Valeur de la propriété

• rotate: nous devons préciser l'angle de rotation

```
div {
   transform: rotate(30deg);
}
```





Transformation 2D - zoomer

■ Valeur de la propriété

- scale: nous devons définir l'échelle pour l'axe X et l'axe Y. Une valeur à 1 correspond à la taille d'origine, 2 une taille multipliée par deux.
- scaleX, scaleY

```
div {
  transform: scale(2, 4);
}
```



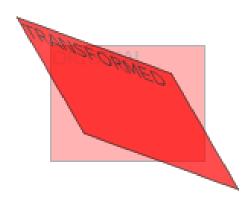


Transformation 2D - obliquer

■ Valeur de la propriété

• skew: nous devons préciser l'angle pour l'axe X et l'axe Y

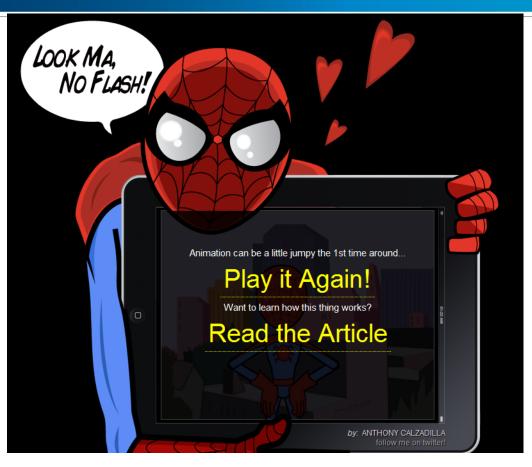
```
div {
  transform: skew(30deg, 20deg);
}
```





Animations





- http://www.optimum7.com/css3man/animation.html
- http://www.w3.org/TR/css3-animations/



HTML5



HTML5 et Doctype

Déclaration pour l'XHTML5

```
Avant
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"</pre>
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<a href="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
   <head>
       <meta http-equiv="Content-Type" content="text/html; charset=utf-8"</pre>
  />
  </head>
</html>
    Après
<!DOCTYPE html>
<html>
   <head>
       <meta charset="utf-8" />
   </head>
</html>
```



Les sélecteurs

- Avant, nous ne pouvions sélectionner un élément que par son identifiant, son nom ou par rapport à sa balise
- Maintenant, nous pouvons également les rechercher en utilisant les sélecteurs CSS

```
Exemples:
// Premier élément de type élément qui contient une classe CSS `new'
var firstElement = document.querySelector("li.new");
```

// Tous les éléments de type élément qui contiennent une classe CSS 'new'

var allElementsAsArray = document.querySelectorAll("li.new");



Formulaires

- De nouveaux types de champs de saisie sont disponibles afin d'enrichir les formulaires Web
- Certains sont plus dédiés aux mobiles comme le type « tel »
- Permet d'éviter d'utiliser des APIs Javascript pour faire des vérifications ou des traitements sur les champs qui bien souvent alourdissent la page Web.
- De nouveaux attributs !! Comme required !!
 - Liste des attributs: http://www.w3schools.com/html5/tag_input.asp



Formulaires

ABC 🌐

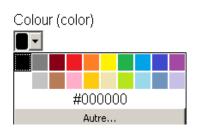
- Liste des nouveaux types à placer dans l'attribut « type » de la balise HTML « input »:
 - color
 - email
 - date
 - datetime
 - datetime-local
 - month
 - number
 - range
 - search
 - tel
 - time
 - url
 - week





Go

space







Multimédia

- Avant HTML5, pour les vidéos : plugin Flash nécessaire
- Avec HTML5, support natif par le navigateur
- 2 nouveaux tags :
 - <audio>
 - <video>
- Une API Javascript pour piloter le « player »
- Supporté par YouTube, DailyMotion,
- Par contre, pas de codec standard !
 - Le codec est l'algorithme utilisé pour décoder / encoder un flux vidéo ou audio



Le choix du codec

	Conteneur MP4 (AAC+H.264)	Conteneur Ogg (Vorbis, Theora)	Conteneur WebM (Vorbis+VP8)
IE	9.0+	-	9.0+ (requiert l'installation du codec VP8)
Firefox	-	3.5+	4.0+
Safari	3.0+	Requiert l'installation d'un plugin QuickTime	Requiert l'installation d'un plugin QuickTime
Chrome	5.0+	5.0+	6.0+
Opera	-	10.5+	10.6+
iPhone	3.0+ Accélération matérielle	-	-
Android	2.0+	-	2.3.3+ 4.0+ (for streaming)

Sources:

http://diveintohtml5.org/video.html

http://developer.android.com/guide/appendix/media-formats.html



Tag audio - Exemple

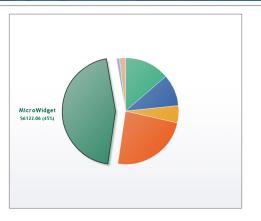
```
<audio id="music" controls autoplay>
 <source src="exemple.mp3" type="audio/mpeg" />
 <source src="exemple.ogg" type="audio/ogg" />
 <!-- Support de IE < 9 -->
 <object data="player_mp3_maxi.swf"</pre>
      type="application/x-shockwave-flash">
      <param name="movie" value="player_mp3_maxi.swf"</pre>
 />
      <param name="FlashVars" value="mp3=exemple.mp3"</pre>
 />
 </object>
</audio>
```



Tag vidéo – Exemple Boutons de contrôle du navigateur

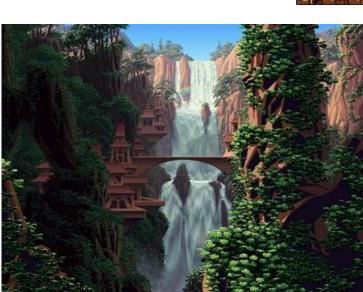
```
<video id="movie" controls autoplay</pre>
        width="400px" height="300px">
 <source src="lapins_cretins.mp4"</pre>
      type='video/mp4; codecs="avc1.42E01E, mp4a.40.2"'
 />
 <source src="lapins cretins.webm"</pre>
      type='video/webm; codecs="vp8, vorbis"' />
 <source src="lapins cretins.ogv"</pre>
      type='video/ogg; codecs="theora, vorbis"' />
 <!-- Support de IE < 9 -->
 <object data="videoplayer.swf"
      type="application/x-shockwave-flash">
       <param name="movie" value="lapins_cretins.swf" />
 </object>
</video>
```

Canvas vs SVG



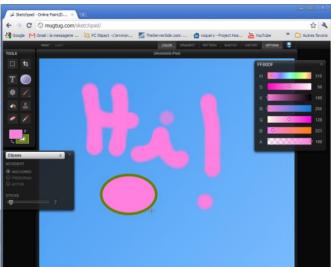
WIDGET	SALES (\$)	
SuperWidget	1862.12	
MegaWidget	1316.00	
HyperWidget	712.49	
WonderWidget	3236.27	
MicroWidget	6122.06	
NanoWidget	128.11	
LovelyWidget	245.55	

SVG





Canvas





Canvas vs SVG

HTML5 propose 2 APIs pour dessiner en 2D :

Canvas, SVG

Canvas : bitmap

- Vous dessinez dans un bitmap
- Avantages
 - Emprunte mémoire très faible (pas d'objets graphiques)
- Inconvénients
 - Une modification nécessite de tout redessiner
 - API bas niveau

SVG : vectoriel

- API DOM : chaque objet graphique est représenté par un objet DOM
- Avantages
 - Un objet graphique peut être manipulé indépendamment des autres
 - Le navigateur prend en charge la mise à jour de l'affichage lorsqu'un objet graphique est modifié
 - API plus haut niveau que Canvas

Inconvénients

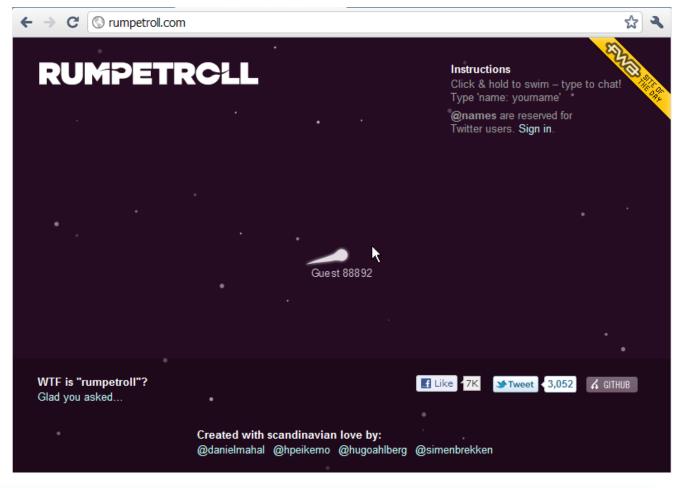
- La mémoire nécessaire est proportionnelle au nombre d'objets graphiques
- Vous ne contrôlez pas les rafraîchissements : problème de performance si nombreuses modifications sur un graphique vectoriel contenant beaucoup d'objets





Actualisation des données en temps réel (push)

http://rumpetroll.com/





WebSocket API

Réception de données

```
// Create new WebSocket
var mySocket = new
WebSocket("ws://websockets.org/example");
// Attach listeners
mySocket.onopen = function(evt) {
       console.log("Connection open...");
};
mySocket.onmessage = function(evt) {
       console.log("Received message: " + evt.data);
};
mySocket.onclose = function(evt) {
       console.log("Connection closed...");
```



WebSocket API (suite)

Envoi de données

```
// Sending data
mySocket.send("HTML5 WebSocket Rocks!");

//Close WebSocket
mySocket.close();
```

Support de WebSocket par le navigateur

```
//Checking for browser support
if (window.WebSocket) {
   document.getElementById("support").innerHTML =
        "HTML5 WebSocket is supported";
} else {
   document.getElementById("support").innerHTML =
        "HTML5 WebSocket is not supported";
}
```



Stockage des données

- HTML5 propose plusieurs approches
 - Local Storage
 - Session Storage
 - Web SQL
 - IndexedDB
- Web SQL n'est plus supporté par le W3C et est remplacé par IndexedDB



Local Storage

- Stocke sous la forme de couples clé/valeur une valeur textuelle
- Taille plus grande qu'un cookie (jusqu'à 200ko)
- Sous Firefox :
 - < 3.5, utiliser : globalStorage[location.hostname]</p>
 - ≥ 3.5, utiliser : localStorage
- Exemple d'utilisation: http://people.w3.org/mike/localstorage.html



Local Storage

Exemples:

```
// Add a value
localStorage.setItem('key', value');

// Get a value
var value = localStorage.getItem('key');

// Remove an entry
localStorage.removeItem('key');

// Remove all entries
localStorage.clear();
```



Cache HTML5

- Informer le navigateur comment cacher les ressources d'un site Web
- Possibilité ainsi de créer des applications hors-ligne
- Pré-requis: faire reconnaître par le navigateur le format d'un fichier de type « text/cache-manifest »
- Déclarer dans la page Web le manifest:

```
<html manifest="example.manifest">
...
</html>
```



Cache HTML5

Lister les ressources à cacher
CACHE MANIFEST
V2
The previous line is really usefull for the refresh
Explicitly cached entries
CACHE:
stylesheet.css
images/logo.png
scripts/main.js

Resources that require the user to be online.

static.html will be served if main.py is inaccessible
offline.jpg will be served in place of all images in images/large/
FALLBACK:
/main.py /static.html
images/large/ images/offline.jpg



NETWORK:

http://api.twitter.com

login.php



Des interfaces optimisées pour les mobiles avec HTML5





- S'adapter à la taille réduite de l'écran
 - ViewPort
- Adapter automatiquement le contenu lors du pivotement du mobile (paysage / portrait)
 - CSS3 Media Queries
 - Evénement 'orientationChange'
- Exploiter l'accéléromètre et le compas
 - API DeviceOrientation
- Géolocaliser l'utilisateur, traceur GPS
- Exploiter les interfaces tactiles multi-points
 - API Touch, API Gesture

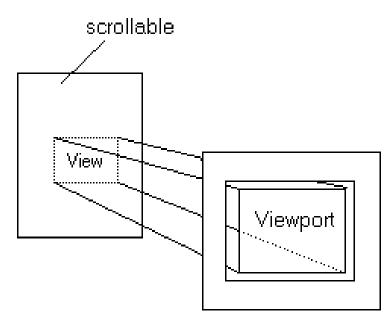


Viewport - Principe

- Adapte la page à la zone affichée
- **Exemples:**
 - Ajustement automatique de la page lors d'une rotation (paysage <-> portrait)
 - Affichage des pages HTML inadaptées à la taille de l'écran du mobile (en dézoomant)









Viewport

■ Déclaration dans le header d'une page Web:

<meta name = "viewport" content = "width = 320 " >

■ Possible de déclarer:

- width: soit un chiffre, soit device-width
- height: soit un chiffre, soit device-height
- initial-scale: niveau de zoom initiale. Par défaut 1.0
- minimum-scale: niveau de zoom minimum
- maximum-scale: niveau de zoom maximum
- user-scalable: booléen autorise ou non à l'utilisateur de zoomer manuellement



Géolocalisation (1)

Permet d'obtenir des informations géographiques comme:

- La latitude de l'utilisateur
- La longitude de l'utilisateur
- La précision de la latitude et de la longitude
- L'altitude de l'utilisateur
- La précision de l'altitude
- L'angle par rapport au Nord
- La vitesse par rapport à la dernière position en mètre par seconde
- La date et l'heure de la dernière position

Utilisable en deux modes:

- Ponctuellement
- Régulièrement
- Lien: http://www.html5-css3.fr/demos/exemple-geolocalisation-html5.html



Géolocalisation (2)

Contenu de l'objet javascript position:

- timestamp
- coords
 - latitude
 - Longitude
 - accuracy
 - Altitude
 - altitudeAccuracy
 - heading
 - speed

Contenu de l'objet javascript error

- code: Code de l'erreur (voir les constantes ci-dessous)
- message: message de l'erreur
- PERMISSION_DENIED
- POSITION_UNAVAILABLE
- TIMEOUT



Géolocalisation (3)

Obtenir la position ponctuellement

```
if(navigator.geolocation){
 navigator.geolocation.getCurrentPosition(
       function(position){
              alert("Lat:" + position.coords.lat);
       function(error){
              if(error.code == error.TIMEOUT){
                     alert("Timeout !! ");
```



Géolocalisation (4)

S'abonner pour obtenir la position régulièrement (comme un GPS)

```
var watchId = navigator.geolocation.watchPosition(
 function(position) {
       alert("Long:" + position.coords.long);
 },
 function(error){
      if(error.code == error.TIMEOUT){
             alert("Timeout !! ");
 },
 { enableHighAccuracy:true, timeout:10000,
 maximumAge:0 }
);
```



Géolocalisation (5)

■ Se désabonner du service de positionnement

navigator.geolocation.clearWatch(watchId);





Orientation

Appliquer des styles CSS différents en fonction de la compara de la c l'orientation (media queries)

```
@media all and (orientation:
          portrait) {
            body div { width: 10%; }
          @media all and (orientation:
          landscape) {
            body div { width: 15%; }
•iOS 4.2, Android 2.3, Blackberry 6, Palm 2.0, Bada
```

Possibilité de détecter le basculement sur l'événément JavaScript « orientationchange »



Support

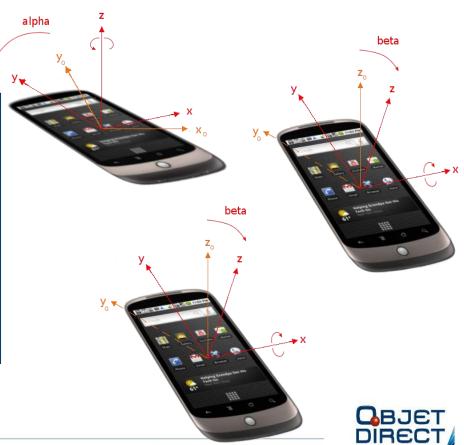
Accéléromètre, compas

■ API du W3C : DeviceOrientation Event Specification

Pour être notifié des changements d'orientation

• alpha, beta, gamma: double

```
window.addEventListener(
  "deviceorientation",
   function(event) {
      // traiter event.alpha,
      // event.beta et
   event.gamma
      },
      true
);
```



Accéléromètre, compas

Pour être notifié des mouvements

- Acceleration: 1 objet avec 3 attributs x,y et z de type double
- RotationRate: 1 objet avec 3 attributs alpha, beta, gamma de type double

```
window.addEventListener(
  "devicemotion",
  function(event) {
    // Traiter event.acceleration,
    //
  event.accelerationIncludingGravit
  y,
    // event.rotationRate et
  event.interval
    },
    true);
```

Support

• iOS 4.2, Android 2.3



Touch

Pour être notifié des mouvements des doigts sur l'écran

Evénements physiques (bas niveau)

```
<div id="square"
  ontouchstart="touchStart(event)"
  ontouchmove="touchMove(event)"
  ontouchend="touchEnd(event)" />
```

ontouchstart	Un doigt commence à toucher l'écran du mobile. Si plusieurs doigts sont positionnés, plusieurs événements sont notifiés.
ontouchmove	Un ou plusieurs doigts bougent en glissant sur l'écran.
ontouchend	Un ou plusieurs doigts ont été retirés de l'écran.
ontouchcancel	Interruption (inattendue) du mouvement en cours.

Support

• iOS 4.2, Android 2.3, Blackberry 6



Gesture

Pour être notifié des gestes de rotation et de zoom

Evénements logiques (API haut niveau)

```
<div id="square"
  ongesturestart="gestureStart(event)"
  ongesturechange="gestureChange(event)"
  ongestureend="gestureEnd(event)" />
```

```
function
gestureChange(event) {
  var angle = event.rotation;
  var scale = event.scale;
}
```

ongesturestart	Début d'un geste : plusieurs doigts ont été posés sur l'écran
ongesturechange	 Geste de rotation ou geste de zoom : L'angle est exprimé en degrés. Un valeur positive indique une rotation dans le sens des aiguilles d'une montre, une valeur négative, une rotation dans le sens inverse. Pour le zoom, ratio < 1 : rapetissement ; ratio = 1 : taille originale, ratio > 1 : grossissement.
ongestureend	Fin du geste : l'utilisateur a retiré ses doigts de l'écran

Support

• iOS 4.2, Android 2.3, Blackberry 6



Et des nouvelles briques à venir!

- Accéder aux contacts
- Accéder aux calendriers
- Vibrer le téléphone, connaître l'état du réseau, de la batterie ...
- Capture du son et de la vidéo
- Notifications
- WebIntents



■ Prévu d'ici courant 2014



PhoneGap



Tout d'abord ...



... une mise au point

Si vous vous êtes penchés sur la question, vous avez dû entendre parler de:





Quelle est la différence ?



Si nous devions faire une analogie





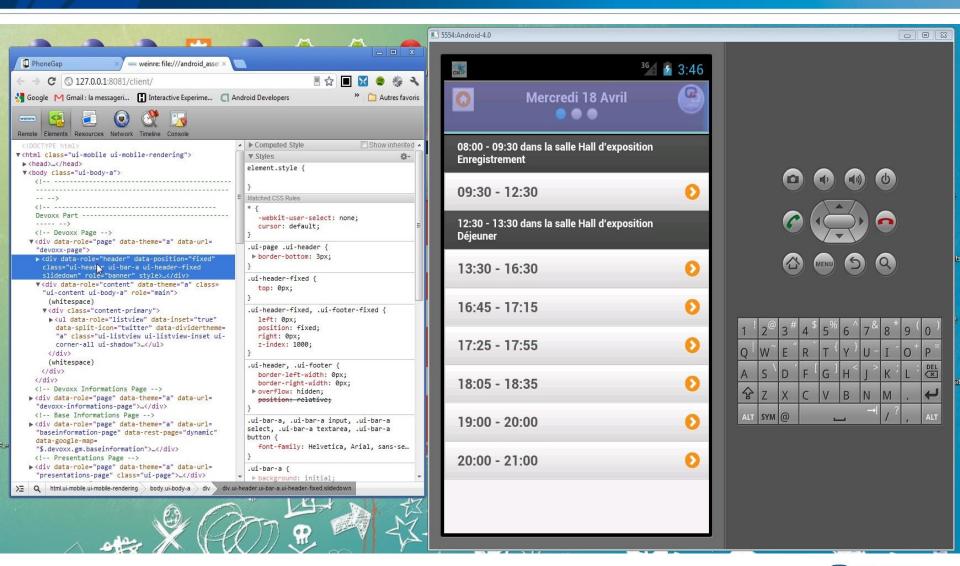
Cordova est pour PhoneGap ce que Webkit est pour Chrome Autrement dit, PhoneGap se repose sur Cordova, mais offre des services supplémentaires







Analyser à distance votre application: Weinre



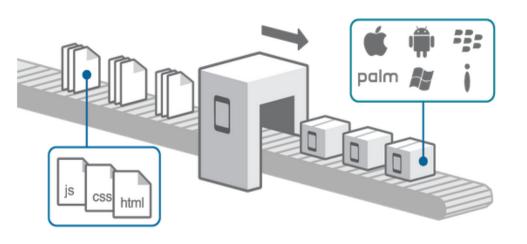


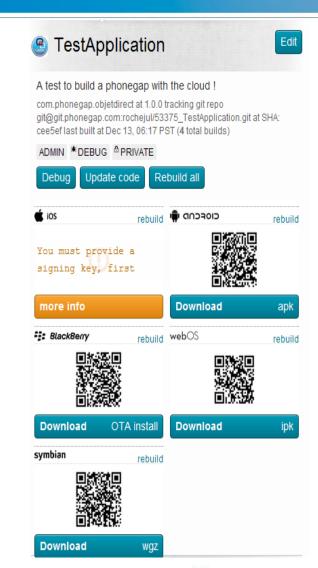
PhoneGap dans le Cloud



Take the pain out of compiling mobile apps for multiple platforms

Create an account







Et maintenant ...



... qu'est-ce que PhoneGap?

■ PhoneGap est une solution innovante pour rendre ses solutions multi-plateformes



Son approche:

- Ecrire son application avec les standards du Web, comme HTML5, CSS3, JavaScripts
- Utiliser l'enveloppe native PhoneGap qui nous intéresse
- Déployer l'application native sur le market dédié
- Ainsi, nous avons une application réalisée par les technologies du Web mais qui s'installent de façon native sur les mobiles



Mais c'est plus que ça!

■ PhoneGap offre également une API JavaScript pour nous faciliter l'accès aux couches basses!



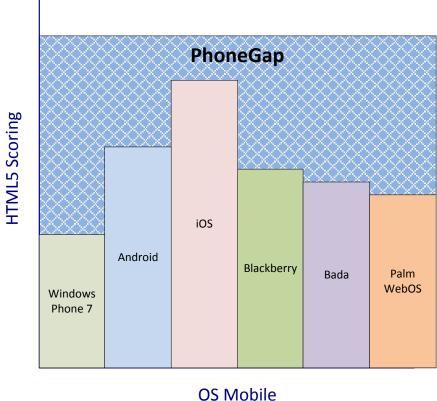
Nous pouvons par exemple:

- Accéder aux contacts
- Accéder aux notifications
- Accéder à l'état du réseau



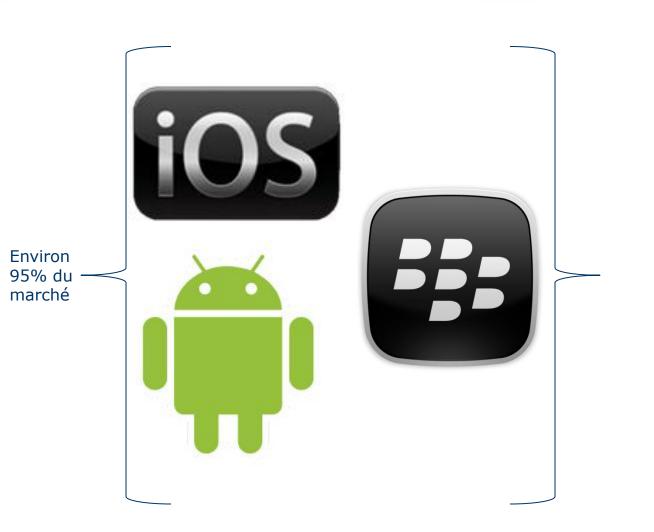
En somme...

... combler le manque des différents navigateurs Web mobile





Les plateformes mobiles ciblées











Les plugins

- PhoneGap a été conçu également pour pouvoir concevoir des plugins
- Ces derniers ont pour objectif de pallier à certains manques de l'API de base
- Ainsi, si nous voulons accéder à des couches basses spécifiques de notre mobile, cela est tout à fait possible et surtout normalisée





Codons un peu!



Les pièges à éviter

- Le fichier cordova.js n'est pas commun pour toutes les plateformes.
 - Il faut donc utiliser celui qui est présent dans l'enveloppe native qui vous intéresse



- Téléchargez toujours la dernière version en cours
- Suivez bien les instructions pour initialiser votre projet
 - Il y a parfois des petites modifications à faire que nous ne pensons pas à faire et qui empêche le bon fonctionnement de PhoneGap



Les pièges à éviter

- Même si vous mettez des permissions pour aller sur Internet, par défaut PhoneGap filtre les liens
 - Exemple sous Android, dans le fichier« res/xml/cordova.xml »

- Sous ios, ce sera « Cordova.plist »
 - Propiété « ExternalHosts » qui contient un tableau
- Du coup, pensez bien à configurer ce paramètre !!!!





Astuce pour la gestion de projets

- Nous allons avoir à créer plusieurs projets pour les différentes plateformes
- De ce fait, pour éviter la duplication de code, nous allons la propriété svn « svn:externals »
 - Pour cela, lancer la commande « svn propedit svn:externals . »
 - Et dans l'éditeur, ajouter pour chaque ligne:
 - <DirectoryName> <SVNURL>
 - Puis « svn:update »
 - Les répertoires devraient apparaître





Mon premier code

- Pour pouvoir utiliser des fonctionnalités de Phonegap, ou les plugins, nous devons nous assurer que l'application a fini de se charger
- Pour se faire, assurer sous Android, que la permission suivante soit déclarée:

```
<uses-permission android:name="android.permission.ACCESS NETWORK STATE" />
```

■ Ensuite, écrivez le code JavaScript suivant:

■ Attention! Utilisez addEventListener! Et non bind de jQuery. Sinon vous ne capturez jamais l'événement



Mon second code

■ Faisons une action simple sur Android: quitter l'application si nécessaire sur le bouton back:

```
document.addEventListener("deviceready", function() {
        // Listen the back button (only on Android)
        document.addEventListener("backbutton", function(event){
            if(aConditionToDefine)){
                event.preventDefault();
                navigator.notification.alert(
                    "Souhaitez quitter l'application ?",
                    function() { navigator.app.exitApp(); },
                    "Quitter l'application",
                    "Quitter",
                    "Annuler"
                );
            else {
                navigator.app.backHistory();
        }, false);
 }, false);
```



Mon troisième code

- Parfois, nous avons des « services » qui tournent en tâche de fonds.
 - Nous pouvons les suspendre quand l'utilisateur switche d'applications
 - Nous pouvons les réactiver quand l'utilisateur revient sur l'application

```
// When the application is paused
document.addEventListener("pause", function(){
    // We stop our "thread" here
}, false);

// When the application is resumed
document.addEventListener("resume", function(){
    // We restart our "thread" here
}, false);
```



Mon premier plugin pour Android (1)

Un plugin qui va faire une somme

Commençons par déclarer le plugin côté JavaScript:

■ Et importons le ! (après le fichier PhoneGap bien sûr)

```
<script src="javascripts/phonegap-1.4.1.js"></script>
<script src="javascripts/phonegap-plugins/sum.js"></script>
```



Mon premier plugin pour Android (2)

Maintenant, déclarons le dans notre projet Android:

Cela se passe dans res/xml/plugins.xml

```
res

drawable-hdpi
drawable-ldpi
drawable-mdpi
layout
layout
makes
drawable-mdpi
layout
layout
layout
drawable-mdpi
layout
drawable-mdpi
layout
layout
drawable-mdpi
layout
layou
```

```
🗖 plugins.xml 🔀
    <?xml version="1.0" encoding="utf-8"?>
    <plugins>
  3
        <plugin name="App" value="com.phonegap.App"/>
        <plugin name="Geolocation" value="com.phonegap.GeoBroker"/>
  5
        <plugin name="Device" value="com.phonegap.Device"/>
  6
        <plugin name="Accelerometer" value="com.phonegap.AccelListener"/>
  7
        <plugin name="Compass" value="com.phonegap.CompassListener"/>
  8
        <plugin name="Media" value="com.phonegap.AudioHandler"/>
  9
        <plugin name="Camera" value="com.phonegap.CameraLauncher"/>
        <plugin name="Contacts" value="com.phonegap.ContactManager"/>
 10
        <plugin name="Crypto" value="com.phonegap.CryptoHandler"/>
 11
        <plugin name="File" value="com.phonegap.FileUtils"/>
 12
        <plugin name="Network Status" value="com.phonegap.NetworkManager"/>
 13
        <plugin name="Notification" value="com.phonegap.Notification"/>
 14
        <plugin name="Storage" value="com.phonegap.Storage"/>
 15
        <plugin name="Temperature" value="com.phonegap.TempListener"/>
 16
        <plugin name="FileTransfer" value="com.phonegap.FileTransfer"/>
 17
        <plugin name="Capture" value="com.phonegap.Capture"/>
 18
         <plugin name="MySumPlugin" value="com.phonegap.plugins.MySumPlugin"/>
 19
 20
```



Mon premier plugin pour Android (3)

Maintenant, déclarons notre fichier MySumPlugin.java

```
package com.phonegap.plugins;
import org.json.JSONArray;
import org.json.JSONException;
import com.phonegap.api.Plugin;
import com.phonegap.api.PluginResult;
public class MySumPlugin extends Plugin {
      private static final String METHOD SUM = "sum";
      @Override
      public PluginResult execute(String action, JSONArray args, String callbackId) {
             if(METHOD SUM.equals(action)) {
                   if(args.length() < 2){ return new PluginResult(PluginResult.Status.ERROR, "We need two parameters"); }</pre>
                    Double value1 = null, value2 = null;
                    try {
                          value1 = args.getDouble(0);
                          value2 = args.getDouble(1);
                    } catch (JSONException e) {
                          return new PluginResult(PluginResult.Status.ERROR, "We need numeric parameters");
                    }
                   // The core of our method :)
                   return new PluginResult(PluginResult.Status.OK, (float) (value1 + value2));
             }
             return new PluginResult(PluginResult.Status.ERROR, "This method is not available");
}
```



Mon premier plugin pour Android (4)

Il ne reste plus qu'à l'utiliser dans notre JavaScript!

```
window.plugins && window.plugins.sumPlugin && window.plugins.sumPlugin.sum(
    1,
    2.53,
    function(sum){
        alert("The sum is:" + sum);
    },
    function(errorMsg) {
        alert("Gasp, an error occured:" + errorMsg);
    }
);
```

Et de manière plus simple:

```
sumPlugin.sum(
    1,
    2.53,
    function(sum){
        alert("The sum is:" + sum);
    },
    function(errorMsg) {
        alert("Gasp, an error occured:" + errorMsg);
    }
);
```

