

# Cours 7 : Intro aux applis mobiles



Applications web et mobiles



Christophe Gonzales

# Bref historique des mobiles


- ~1990 Norme GSM 2G
- 1992 1er smartphone : IBM Simon
- 1999 Nokia 3210, Blackberry OS (C++)
- 2002 BlackBerry 5810
- 2006 Norme GSM 3G
- 2007 iPhone 1, lancement d'iOS (XNU)
- 2007 Lancement d'Android (Linux)
- 2008 App store et Android Market
- 2009 Smartphone Samsung Galaxy i7500
- 2010 Norme GSM 4G, Windows Phone (C++/C#)
- 2012 Google Play : Android Market + Google Movies + Google Music






## Applications natives :

- ▶ **iOS** : Langages Objective-C, **swift** (depuis 2014)
- ▶ **Android** : Langages Java, C++, **Kotlin** (depuis 2017)
- ▶ **Windows Phone** : Langages C++, **C#**
- ▶ **Avantages** : accès aux fonctionnalités de l'OS (GPS, photo, *etc.*)
- ▶ **Inconvénients** : pas cross-plateforme, updates sur  

## Applications web :

- ▶ **WebView** : charger un contenu web dans une appli native
- ▶ **iOS** : UIWebView/WKWebView    **Android** : WebView
- ▶ **Avantages** : Cross-plateforme, développement simple
- ▶ **Inconvénients** : exécution plus lente, support , nécessite du réseau

## Applications hybrides :

- ▶ Mix entre applications web et applications natives :  
Code « web » + librairie accédant aux fonctionnalités de l'OS  
⇒ appli native
- ▶ **Apache Cordova** (2009) : code en JavaScript, HTML, CSS
- ▶ **Xamarin** (2011) : code en C#
- ▶ **React Native** (2015) : code en Javascript  
+ code natif Java , Objective-C ou Swift 
- ▶ **Ionic Capacitor** (2018) : code en JavaScript, HTML, CSS
- ▶ **Avantages** : Cross-plateforme, fonctionnalités de l'OS
- ▶ **Inconvénients** : 1 production par OS, installations  

## Progressive Web Applications **PWA** :

- ▶ Amélioration des applications hybrides :
    - ▶ **Fiabilité** : peut fonctionner sans réseau (cache)
    - ▶ **Sécurité** : utilisation exclusive de réseau sécurisé (https)
    - ▶ **Installable** : comme une appli native mais via le web
    - ▶ **Progressif** : Updates automatiques via des notifications
- 
- ▶ 3 caractéristiques pour être une **PWA** :
    - ▶ Utilisation de https
    - ▶ Avoir un manifeste (installations)
    - ▶ Utiliser un **Service Worker** (updates, cache, *etc.*)

# PWA et developer tool

