Landing Page de A à Z

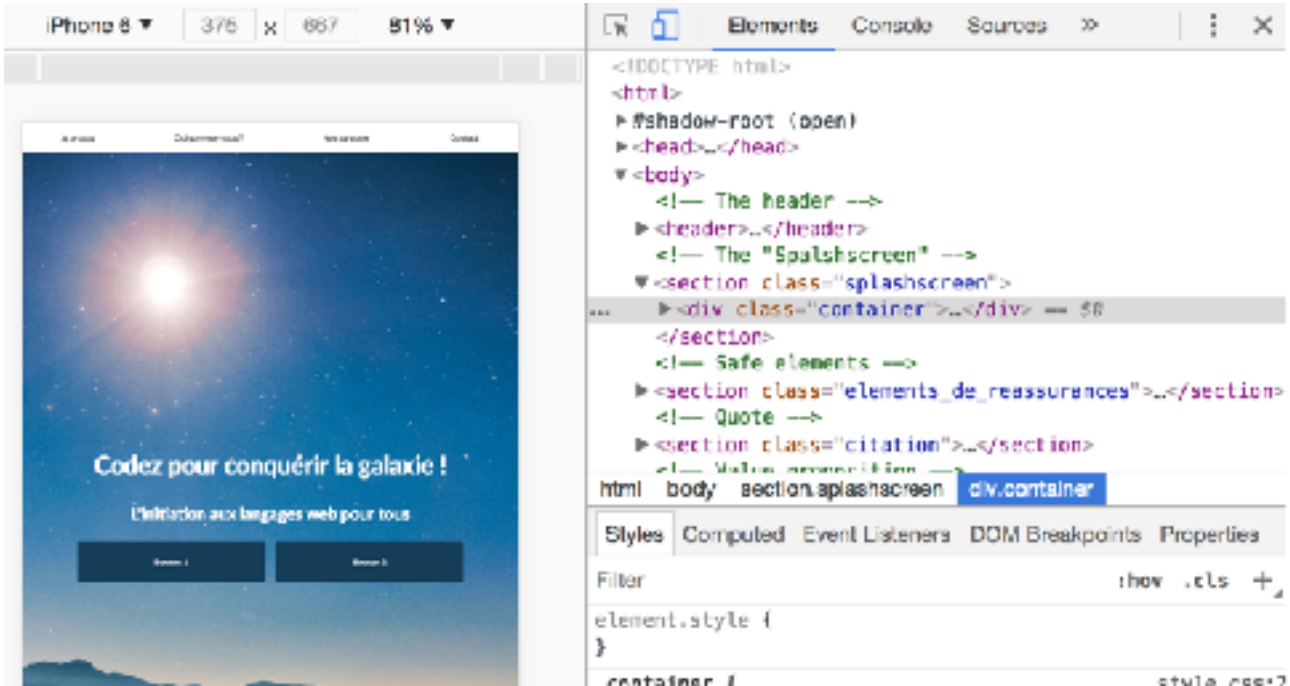
Cours 3

Faisons un petit test..

- ➔ Allez sur votre site
- ➔ Redimensionnez votre fenêtre pour la rendre petite
- ➔ Observez.. Qui voit un problème ?

Démo

➔ Astuce : comment observer un site web sur différents devices ?



Quelqu'un voit un problème ?

Sur iPhone 6, le texte est petit, difficilement lisible, et l'affichage est loin d'être optimisé pour mobile !

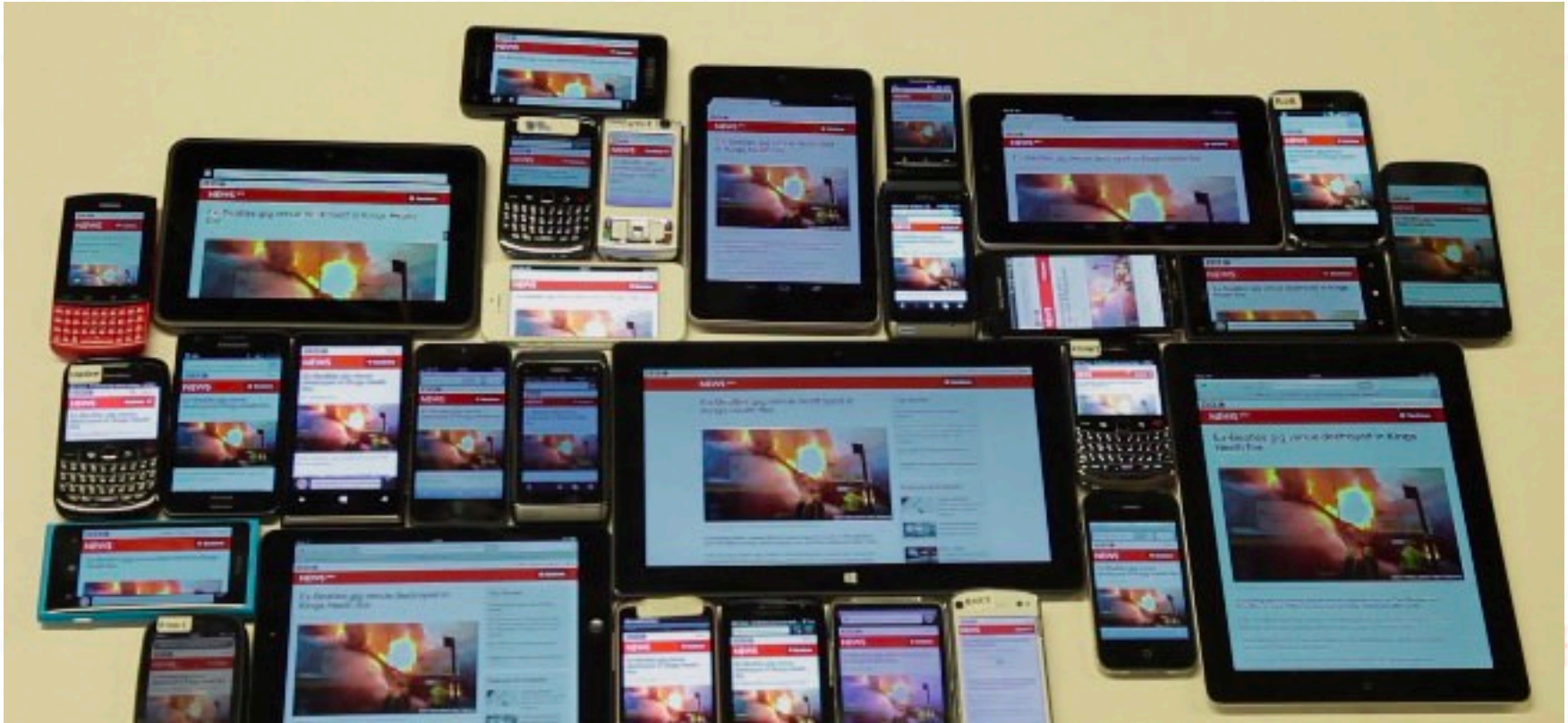
Responsive Web Design

En anglais RWD
(Responsiveness = réactivité)

Le responsive concerne l'adaptation des sites web aux différents terminaux. Cette évolution est relative aux nouvelles fonctionnalités des navigateurs.

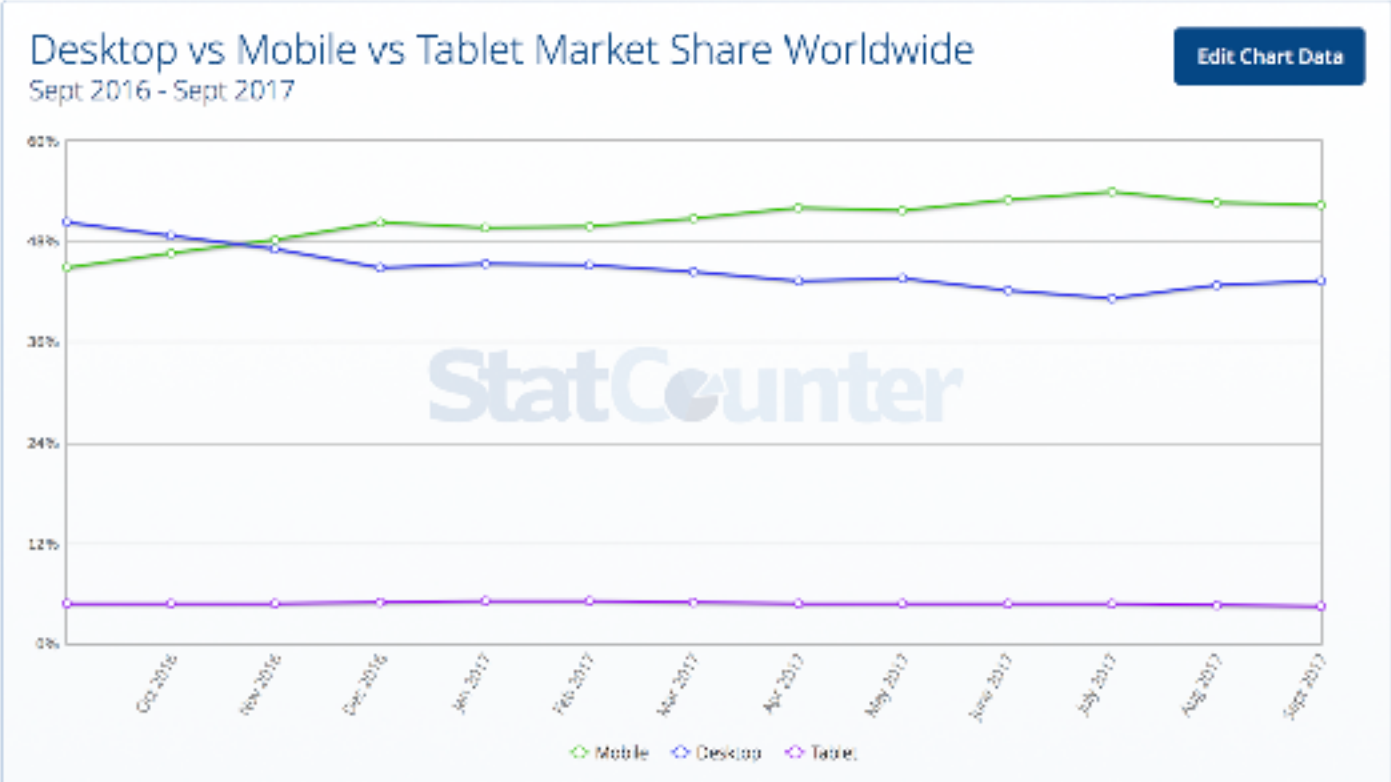
Le responsive est incontournable aujourd'hui et il rentre dans les critères de référencement de votre site web par google

Pourquoi le responsive ?



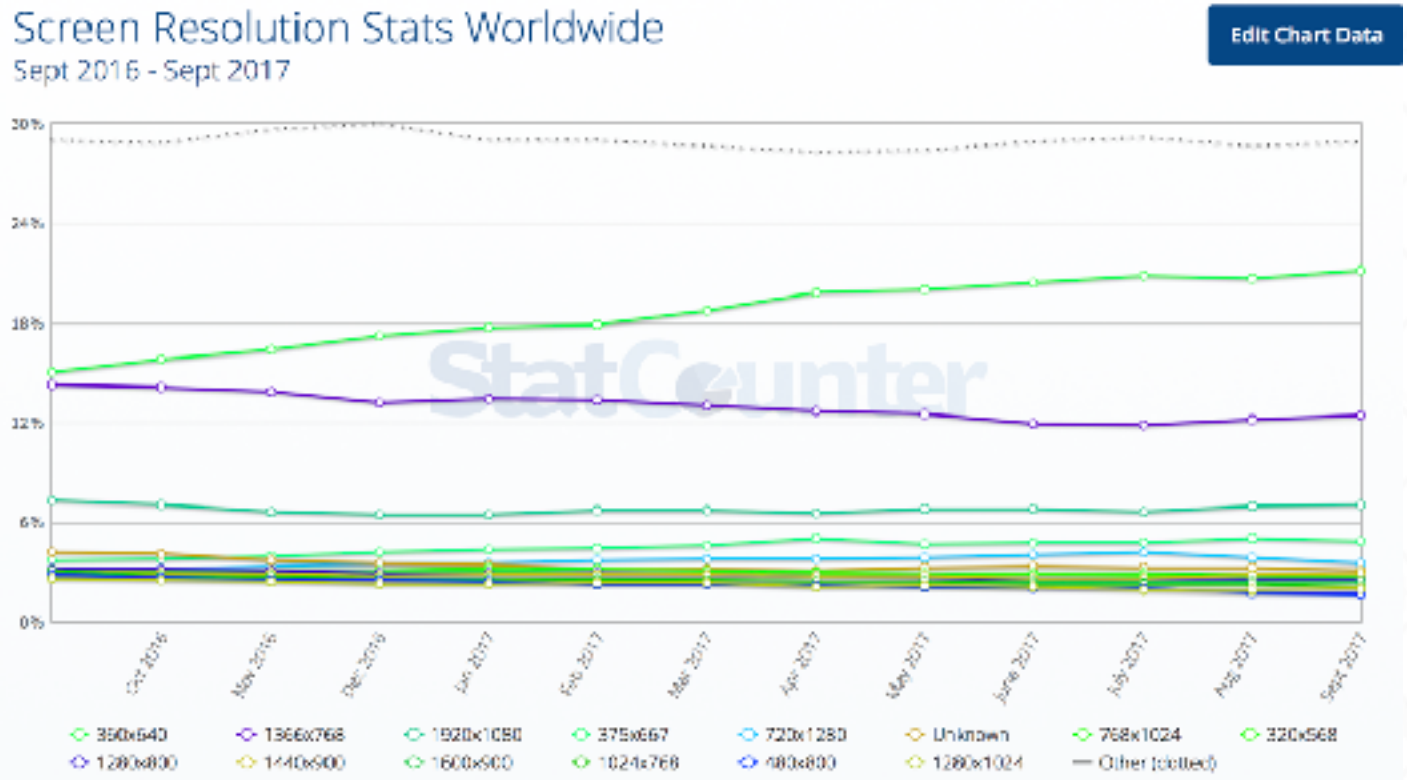
Pourquoi le Responsive ?

source : statcounter.com

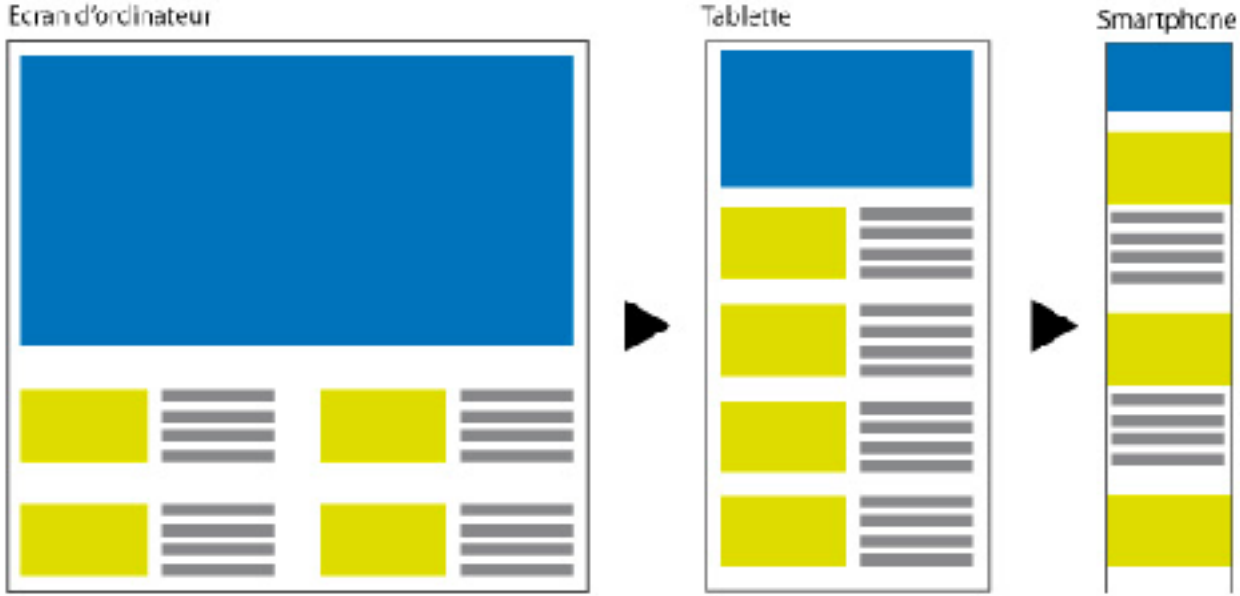


Pourquoi le Responsive ?

source : statcounter.com



Le Responsive



Un (bel) exemple

https://www.thinkwithgoogle.com/

L’apparition de multi-devices

L’apparition de multi-devices oblige à rendre **les sites web** adaptables à n’importe quelle taille d’écran : smartphone, tablette, laptop, PC, télévision ...

Il existe donc 3 solutions :

- La création d’une nouvelle version du site
  - Exemple : je refais une version de mon site et j’utilise un sous-domaine m.monsiteweb.com
- La création d’une application dédiée (smartphone, PC, TV …)
  - Dans ce cas, on ne passe plus par le navigateur. L’utilisateur doit télécharger l’appli
- **Le responsive web design : un incontournable**

Le Responsive Web Design : qu’est-ce que c’est ?

➔ Test : réduire la fenêtre de votre navigateur

Le Responsive Web Design consiste à adapter tous les éléments d’un site web aux différentes tailles d’écrans et aux différents devices

Plusieurs objectifs que l’on retrouve systématiquement pour le mobile:

- Rendre les textes et images lisibles
- Adapter les éléments cliquables au terminal mobile
- Cacher certaines informations inutiles
- Passer les blocs dans une seule colonne centrale

=> Travailler l’expérience utilisateur (UX) pour une meilleure navigation

Le responsive : historique

- Il était déjà possible en CSS2 et HTML4 de définir des styles pour différentes tailles d'écran… (Début des années 2000)
- L'arrivée des normes du CSS3 se généralisent à partir de 2008
- A partir du 21 Avril 2015, Google commence à mettre en avant les sites responsives

Les navigateurs actuels reconnaissent tous ces spécifications depuis :

- Mozilla Firefox : 3.5+
- Internet Explorer : 9+
- Google Chrome : 5+
- Opera : 10.5+, Opera Mobile : 10+, Opera Mini : 5+
- Apple Safari : 4+ et iOS (mobile) 3.2+
- Android : 2.1+

Comment fait-on ?

Images flexibles

Pour les images par exemple, on peut utiliser des % pour la taille.

Exemple : https://codepen.io/anon/pen/zEPqPx

Viewport

Par défaut, les navigateurs mobiles / tablettes souhaitent afficher les site web en entier, de manière à ce que tout le contenu rentre dans l'écran. Le problème, c'est que un écran de mobile, le contenu devient vite illisible !

Il va donc falloir désactiver ce zoom réalisé par défaut avec une balise viewport, dans le head ! :

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Grilles fluides

« Fluid grid » : adapter la taille des blocs d'une grille et le nombre de blocs par ligne, selon la résolution.

Dans le modèle flexbox, on peut aussi utiliser flex-wrap et jouer avec les pourcentage de taille de blocs.

Exemple : https://codepen.io/anon/pen/rGYeJN

Les Media Queries (CSS3)

On va écrire des styles (CSS) différents pour différentes tailles d'écran

- **à la fin de son fichier style.css**

```
@media only screen {
html, body {
        font-family: Arial;
   font-size: 1.0em;
   margin: 0px;
   padding: 0px;
}

/* Mobile styles is set under 768px*/
@media screen and (max-width: 768px) {
 body {
    font-size: 2.0em;
 }
}
```

- **dans un autre fichier mobile.css**

```
<link rel="stylesheet" media="screen and (max-width: 640px)" href="smallscreen.css" type="text/css" />
```

➔ **Quelle est la meilleure pratique ? Pourquoi ?**

Les Media Queries: Syntaxe

Les media queries permettent de spécifier plusieurs paramètres aux styles (CSS)

```
@media  screen and (max-width: 1072px) and (min-width: 768px) {   /*  Mes styles  */  }
```

- **La déclaration** d'ouverture des media queries, avec @media
- **Le type de média** : screen, print, speech…

/!\ Les autres médias sont maintenant déconseillés (tels que tv, projection, braille…)

- **Les paramètres de l'écran** d'affichage pour lesquels vont s'appliquer les styles vont s'écrire entre parenthèse: width, height, resolution, orientation, color-index, aspect-ratio …
- **Les opérateurs** : and, not et only

Les Media Queries

➔ Pleins de manières d’écrire ses média queries (chercher les différences)

```
@media only screen {
h1 {
        font-size: 1.6em;
                font-weight: bold;
}

/* Tablet styles is set between 768px and 1072px */
@media (max-width: 1072px) and (min-width: 768px) {
                h1 {
                                font-size: 2.2em;
                                font-weight: normal;
                }
}

/* Mobile styles is set under 768px*/
@media screen and (max-width: 768px) {
                h1 {
                                font-size: 2.0em;
                }
}
```

```
h1 {
        font-size: 1.6em;
                font-weight: bold;
                margin-top: 10px;
}

/* Tablet styles is set between 768px and 1072px */
@media  screen and (min-width: 768px) {
                h1 {
                                font-size: 2.2em;
                                font-weight: normal;
                }
}

/* other styles are set above 1072px*/
@media screen and (min-width: 1072px) {
                h1 {
                                font-size: 2.0em;
                }
}
```

Quels gabarits choisir ?

Avec les média Queries, on met en place des **breakpoints**, c'est à dire des largeurs minimales / maximales. on choisira généralement de 2 à 5 gabarits

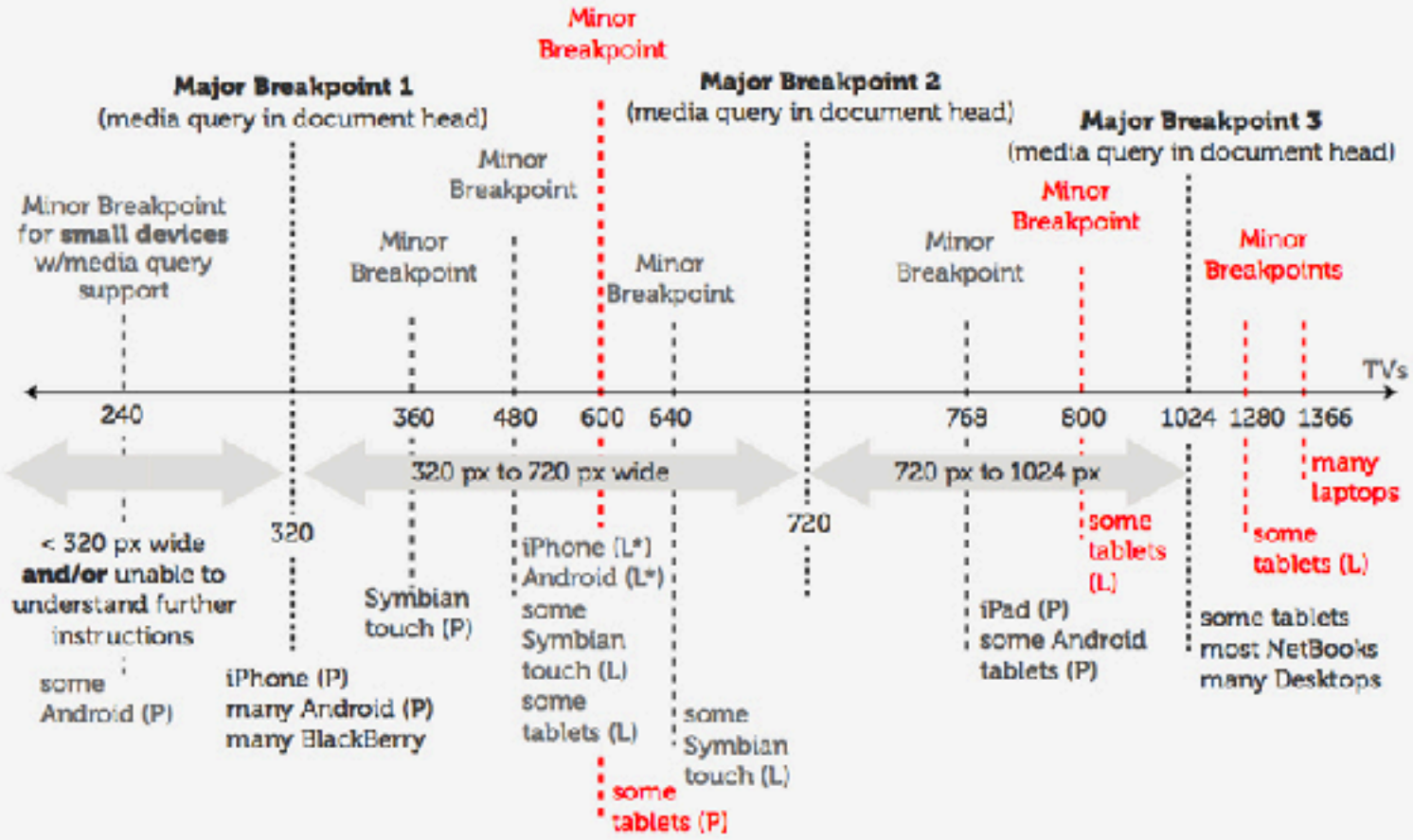
```
@media only screen { /*  Mes styles  */ }
@media screen and (max-width: 1072px) {   /*  Adaptation mobile  */  }
```

```
@media only screen { /*  Mes styles  */ }
@media screen and (max-width: 1024px) {   /*  Adaptation tablette  */  }
@media screen and (max-width: 768px) {   /*  Adaptation tablette  */  }
```

```
@media only screen { /*  Big Screens  */ }
@media screen and (max-width: 1450px)
@media screen and (max-width: 1024px)  and (min-width: 768px) {  /*  tablet  */  }
@media screen and (max-width: 768px) {   /*  mobile  */  }
@media screen and (max-width: 768px) {   /*  Small screens */  }
```

Simple

Multiple / Complexe


Minor Breakpoint
Major Breakpoint 1
(media query in document head)
Major Breakpoint 2
(media query in document head)
Major Breakpoint 3
(media query in document head)
Minor Breakpoint for small devices w/media query support
Minor Breakpoint
Minor Breakpoint
Minor Breakpoint
Minor Breakpoint
Minor Breakpoint
Minor Breakpoints
TVs
240
320
360
480
600
640
720
768
800
1024
1280
1366
320 px to 720 px wide
720 px to 1024 px
many laptops
< 320 px wide and/or unable to understand further instructions
some Android (P)
iPhone (P)
many Android (P)
many BlackBerry
Symbian touch (P)
iPhone (L*)
Android (L*)
some Symbian touch (L)
some tablets (L)
some tablets (P)
some Symbian touch (L)
some tablets (L)
iPad (P)
some Android tablets (P)
some tablets (L)
some tablets
most NetBooks
many Desktops

Repérez les breakpoints (mineurs et majeurs) sur ce site

https://www.thinkwithgoogle.com/

Les recommandations de gabarits

Avec les média Queries, le choix des breakpoints est libre. On va les choisir en fonction du projet. Quelques petites règles :

- Il est inutile de choisir ses breakpoints en fonction de devices populaires (iPhone, Samsung …).
- De plus en plus, les développeurs commencent à écrire le style pour mobile, puis utilisent les media queries pour adapter les versions tablette / desktop
- Il est conseillé d'utiliser des unités de valeur relatives (em ou rem) plutôt qu'absolues (px)
- De nouvelles propriétés en CSS3 permettent de mieux gérer les hauteurs / largeurs, tel que Viewport Height (vh), Viewport Width (vw) ...

Challenge - Améliorer le responsive de son site

Arriverez-vous à améliorer la lecture de votre site pour une largeur d’écran sur mobile ?

- Choisir les gabarits sur lesquels on va travailler
- Passer tous les éléments en une seule colonne
- Redimensionner les éléments trop gros / petits
- Rendre les liens cliquables sur mobile
- Cacher certains éléments avec un display:none;

Le responsive dans la gestion de projets

- **Le graphiste** en charge du projet doit connaître les bases du HTML / CSS. Aujourd'hui, le site doit être pensé en connaissant ces normes.
- Dans la majorité des cas, les versions mobiles et tablettes font l'objet de maquettes **par le graphiste**, avec l'avis **du développeur**.
- Dans d'autres cas, la version mobile est réalisée a postériori (projets plus petits).

De plus en plus souvent, **les UX / UI designers** commencent à penser la version mobile avant les autres versions tablettes et desktops.

Bootstrap et les frameworks CSS

Les frameworks CSS sont des fichiers CSS très complets et responsive. On va alors utiliser les classes CSS pour afficher 3 colonnes en desktop, avec une adaptation mobile.

=> gain de temps pour le développeur

Les frameworks CSS les plus connus :

- Bootstrap
- Zurb foundation
- Pleins d’autres existent avec différentes caractéristiques