

Principes du Responsive Web Design

BUT MMI 2^{ème} année 2024-2025

Plan

Le contexte

Les éléments techniques

Les 2 approches du RWD

Le viewport

Les unités en CSS

Usage d'internet : Part de marché des smartphones

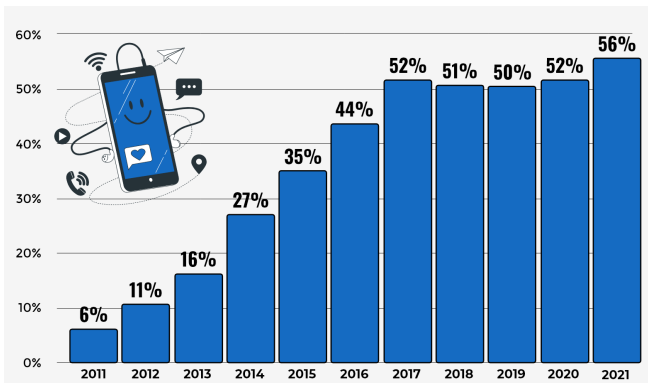
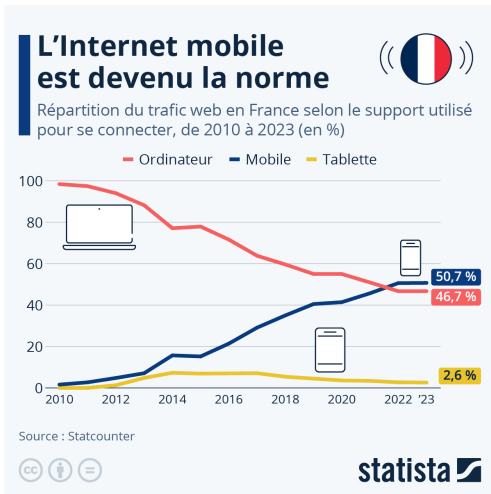


Figure – Source : www.broadbandsearch.net (2022)

Usage d'internet : Part de marché des smartphones

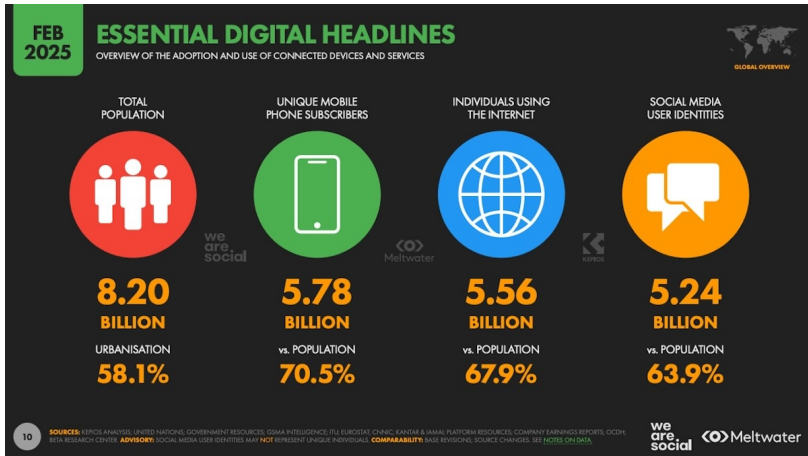


Usage des réseaux sociaux en 2022 dans le monde

D'après des sources `www.broadbandsearch.net` :

- ▶ 80% du trafic sur les réseaux sociaux s'effectue à partir d'un smartphone
- ▶ 95.1% des utilisateurs de Facebook utilisent un smartphone
- ▶ 86% de l'usage de twitter se fait sur mobile
- ▶ 60% de l'usage de linkedIn se fait sur mobile

Mobiles et réseaux sociaux en 2025



Conséquences pour le développeur Web

Impossible aujourd'hui de :

- ▶ Prévoir la taille de l'écran de l'utilisateur pour afficher le document (écran de smartphone ? écran d'ordinateur de bureau ?)
- ▶ Prévoir le support utilisé par l'utilisateur (écran ? tablette braille ? lecteur vocal ? télévision ?)
- ▶ Connaître les outils d'interaction de l'utilisateur avec le document (souris ? écran tactile ? voix ?)

Nécessité pour les documents Web de s'adapter à chaque utilisateur et de proposer une expérience de consultation et de navigation appropriée.

Responsive Web Design : Naissance

- ▶ Article fondateur d'Ethan Marcotte le 25 mai 2010 (<http://alistapart.com/article/responsive-web-design>)
- ▶ Le principe du RWD s'appuie sur 3 éléments techniques bien précis (appelée "*sacro-sainte trinité*" par Raphaël Goetter) :
 - Les **grilles fluides**
 - Les **Media Queries**
 - L'adaptabilité de la taille des images (**images flexibles**)
- ▶ L'utilisation de ces techniques est rendue possible grâce à l'évolution du langage CSS.

Plan

Le contexte

Les éléments techniques

Les 2 approches du RWD

Le viewport

Les unités en CSS

Les grilles fluides

- ▶ Ce sont des grilles où le contenu s'adapte à la largeur disponible de l'écran.
- ▶ On utilise souvent des largeurs exprimées en pourcentage (%).
- ▶ On divise souvent la largeur d'affichage disponible en plusieurs colonnes de même largeur et chaque élément en occupe un certain nombre. Par exemple, le framework CSS Bootstrap divise la zone d'affichage en 12 colonnes.
- ▶ On utilise maintenant les modules CSS **flexbox** et **Grid Layout** où les éléments adaptent automatiquement leur largeur à leur conteneur en respectant une certaine disposition.

Les Media Queries (1)

Les **Media Queries** permettent de modifier l'apparence d'un document web en fonction :

- ▶ Du type de média de sortie (écran ou imprimante par exemple)
- ▶ Des caractéristiques de ce media (viewport pour un écran par exemple)

Pour le RWD, on cible le plus souvent l'écran comme type de média et une largeur de zone d'affichage comme caractéristique :

```
1 @media screen and (max-width:1250px) { ... }
```

Les Media Queries (2)

```
1 @media screen and (max-width:1250px) { ... }
```

Dans l'exemple précédent, les règles CSS placées dans les accolades s'appliquent lorsque la zone d'affichage a une largeur inférieure à 1250px.

On peut alors modifier complètement le design du document lorsque la condition entre parenthèse est vraie. Cette condition est alors appelée un **point de rupture (breakpoint)**.

De manière générale :

```
1 @media media and (breakpoint) { règles CSS ... }
```

Les Media Queries (3)

C'est la largeur de la zone d'affichage ou la largeur du périphérique de sortie qui sont souvent ciblés par la condition du breakpoint. On utilise alors les propriétés CSS :

- ▶ `max-width`
- ▶ `min-width`
- ▶ `max-device-width`
- ▶ `min-device-width`

Les images flexibles

- ▶ Les images sont des éléments à taille fixe.
- ▶ Elles doivent être capables de se redimensionner si besoin.
- ▶ Utilisation de la règle CSS :

```
1 max-width : 100%;
```

lorsque la taille de l'image dépasse celle de son conteneur dans un design fluide.

- ▶ La hauteur de l'image s'adapte alors automatiquement
- ▶ Pour les images d'arrière plan si besoin :

```
1 background-size : cover;
```

Plan

Le contexte

Les éléments techniques

Les 2 approches du RWD

Le viewport

Les unités en CSS

Responsive dégradation

- ▶ Dans cette approche, le design est pensé pour un écran d'ordinateur de bureau.
- ▶ Lorsque la largeur disponible de l'écran diminue, les éléments sont réagencés ou tout simplement supprimés du document pour adapter le design à la plus faible résolution d'écran.
- ▶ Inconvénient : les terminaux mobiles chargent entièrement le document pour écran de bureau, y compris les éléments de style inutiles pour la version faible résolution. C'est un problème lorsque la qualité du réseau cellulaire n'est pas optimale.

Mobile First

- ▶ Dans cette approche, on pense l'ergonomie de la version mobile avant la version bureau.
- ▶ Doit être pensée pour des terminaux peu puissants et avec une connexion réseau non optimale.
- ▶ L'objectif est d'améliorer l'expérience utilisateur (UX)
- ▶ Approche de plus en plus utilisée étant donnée l'importance du mobile dans l'utilisation du web aujourd'hui.
- ▶ On adapte ensuite progressivement le design pour les écrans plus larges. C'est l'opposé de l'approche par dégradation.

Plan

Le contexte

Les éléments techniques

Les 2 approches du RWD

Le viewport

Les unités en CSS

La notion de viewport

- ▶ Attribut de la balise meta introduit par Apple en 2007 à la sortie de l'iPhone.
- ```
1 <meta name="viewport" content="VALEURS" />
```
- ▶ Permet de contrôler les dimensions de la zone d'affichage du document (certains devices font un zoom par défaut).
  - ▶ Le document peut s'afficher en fonction des dimensions du périphérique de sortie et non en fonction de dimensions fixées par le navigateur.
  - ▶ Permet de définir le niveau de zoom (ou dé-zoom) permis à l'utilisateur
  - ▶ Très utile pour l'affiche de documents web sur terminaux mobiles

# viewport : Exemple de l'Iphone 6 (1)

Pour l'Iphone 6

- ▶ La résolution physique de l'écran est de  $750 \times 1334$ . On parle de **pixels physiques**.
- ▶ Pour l'affichage de documents web, CSS dispose d'une résolution de  $375 \times 667$ . (valeurs des propriétés `device-width` et `device-height`). Ce ne sont pas des pixels physiques mais des **pixels CSS** ou encore **pixels indépendants** (DIP)
- ▶ La largeur par défaut du viewport est de 960px !

# viewport : Exemple de l'Iphone 6 (1)

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Fonctionnement du viewport</title>
5 <meta charset="utf-8" />
6 <style>
7 body {background:orange;}
8 div {width:300px; height:200px;
9 background-color:magenta; border-radius:10px;}
10 </style>
11 </head>
12 <body><div>Un block de 300px de large</div></body>
13 </html>
```

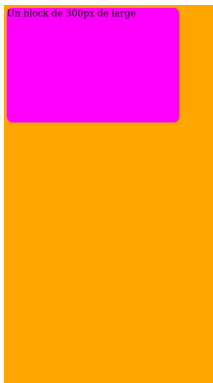
Affichage par défaut d'un bloc de  $300 \times 200$  dans l'iPhone 6.



Le bloc occupe  $\frac{300}{960}$  de la largeur de l'écran (375px)

On ajoute la balise meta suivante :

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



Le bloc occupe  $\frac{300}{375}$  de la largeur de l'écran.

# Plan

Le contexte

Les éléments techniques

Les 2 approches du RWD

Le viewport

Les unités en CSS



# Les unités absolues

Leur taille est fixée définitivement. Elles ne sont pas influencées par d'autres mesures présentes pour l'affichage du document.

- ▶ **px** : le pixel
- ▶ **pt** : le point
- ▶ **cm** : le centimètre
- ▶ **mm** : le millimètre
- ▶ **in** : le pouce
- ▶ **pc** : le pica

1in = 96px, 1in = 72pt, 1cm = 37.8px et 1pc = 12pt.

# Le pixel : px

- ▶ L'unité **px** varie en fait selon le type de matériel et de son usage.
- ▶ Elle est définie pour être petite mais visible. Une ligne horizontale de 1px de largeur peut être affichée de façon nette, sans 'arête' (ni anti-aliasing).
- ▶ Ce qui est net, petit et visible dépend évidemment du support et de la façon dont l'utilisateur le visionne.
- ▶ Les écrans récents à très haute résolution sont concernés par cette définition.
- ▶ Les unités absolues peuvent donc être dépendantes de la résolution et de la taille de l'écran et donc relatives par rapport à celui-ci.

# Les unités relatives

Elles sont **relatives** à d'autres dimensions utilisées dans le document.

Elles sont plus adaptées au Responsive Web Design.

- ▶ **%** : relative aux dimensions de l'élément parent.  
Indispensable pour obtenir des designs fluides.
- ▶ **em** : relative à la taille de la police actuellement utilisée.
- ▶ **rem** : relative à la taille de la police de l'élément racine.
- ▶ **ex** : relative à la hauteur des petites lettres de la police utilisée, comme le "a" ou le "e"

# Les unités **em** et **rem**

Elles sont relatives à la taille de la police (propriété CSS `font-size`).

- ▶ Par défaut, la taille de la police est héritée de l'élément parent
- ▶ L'unité **em** est donc relative à la taille de police de l'**élément parent**
- ▶ L'unité **rem** est en revanche relative à la taille de police de l'**élément racine**
- ▶ L'unité **rem** reste la même pour tout le document

# Les unités relatives au viewport

- ▶ **vw** : relative à la largeur du viewport
- ▶ **vh** : relative à la hauteur du viewport
- ▶ **vmin** : plus petite valeur entre **vw** et **vh**
- ▶ **vmax** : plus grande valeur entre **vw** et **vh**