

# MEMO RESPONSIVE WEB DESIGN

## Mémo Conception Internet Réactive

- 1. Qu'est que le *Responsive Design* ?
  - 1.1 Les différents types de design
    - Design statique
    - Design liquide (ou fluide)
    - Design adaptatif (évolutif)
    - Design *Responsive* (réactif)
- 2. Gérer les différentes zones de visualisation
  - 2.1 Les requêtes média (*Media queries*)
  - 2.2 Les points de rupture (*breakpoints*)
  - 2.3 Les unités de mesure et la police *responsive*
- 3. Diverses techniques du fluide au *Responsive*.
  - 3.1 *Mobile First*

# 1. Qu'est-ce que le *Responsive Design* ?

Le *Responsive design* a été traduit en français par **Conception réactive**. Ce design permet de modifier la mise en page d'un site afin que le contenu s'adapte à l'écran quel que soit le terminal utilisé (smartphone, tablette, ordinateur de bureau, TV...). Il s'agit d'adapter le site à toutes les résolutions. Autrement dit, les blocs de contenus (textes et images) se redimensionnent et se réorganisent en fonction de l'espace disponible sur l'écran.



Stéphanie Walter : <https://stephaniewalter.design/blog/download-illustration-content-is-like-water/>



Les fans de Bruce Lee se souviendront du légendaire « Be water my friend », <https://www.youtube.com/watch?v=cJMwBwFj5nQ>

On rencontre aussi le terme **Responsive Web Design (RWD)** qui a été inventé par Ethan Marcotte en Mai 2010 sur le webzine « A List Apart » : <https://alistapart.com/article/responsive-web-design/> Il y aborde pour la première fois le concept de *Responsive Design*, qu'il développe dans un livre (« *Responsive Web Design* ») un an plus tard. Comparé à un design statique, un site web *responsive* présente différents avantages, notamment :

- Centraliser les contenus en gérant les CSS selon le terminal
- Création et maintenance du site plus rapide et plus facile
- Design et contenus identiques, donc familiers pour l'utilisateur

## 1.1 Les différents type de design Web

C'est un domaine toujours en ébullition et vous en trouverez pas toujours les mêmes définitions pour les différents type de design. En voici quatre définitions qui devrait faire l'unanimité.

### 1. Design statique

La **mise en page statique** existe depuis longtemps, c'est ce qu'il y a de plus traditionnel. Le site a une largeur définie à la conception, 960 pixels de large par exemple. Si la fenêtre du navigateur est plus large que le site, il y aura du blanc ou un arrière plan sur les côtés. Par contre, si la fenêtre du navigateur est moins large que le site, il y aura une barre de défilement horizontal qui apparaîtra en bas de la fenêtre pour pouvoir accéder à la partie qui déborde, ce qui n'est pas très ergonomique.

### 2. Design liquide (ou fluide)

Dans une **mise en page liquide (ou fluide)**, les largeurs des différents éléments, principalement les colonnes, sont exprimées en pourcentage, ce qui fait que le site s'adapte de façon automatique à la taille de l'écran ou de la fenêtre du navigateur. la largeur du site et des colonnes qui le composent sera toujours proportionnelle à la largeur de la fenêtre du navigateur. Ce design n'est pas très adapté aux très grandes tailles d'écrans ni aux très petites tailles.

### 3. Design adaptative (évolutif)

Un **design adaptative** utilise les *Media Queries CSS3* pour prévoir plusieurs mises en page différentes pour différentes tailles d'écrans. Lorsque la fenêtre du navigateur est redimensionnée, à chaque franchissement d'une largeur d'écran définie par les *Media Queries*, la mise en page correspondante est utilisée pour afficher le site.

### 4. Design responsive (réactif)

Un **design responsive (ou RWD pour responsive web design)**, tout comme un design *adaptative*, prévoit plusieurs mises en page différentes pour différentes tailles d'écrans, toujours à l'aide des *Media Queries CSS3*, mais chacune de ces mises en page est fluide (ou liquide). cela permet de toujours utiliser de façon optimale la largeur d'affichage disponible, quelle que soit la taille de l'écran ou de la fenêtre du navigateur.

#### Références

Usabilis : Qu'est-ce que le Responsive Webdesign ou adaptative Design ?

<https://www.usabilis.com/responsive-web-design-site-web-adaptatif/>

Statique, liquide, adaptative ou responsive ?

<https://www.hatihati.fr/blog/statique-liquide-adaptative-ou-responsive/>

Le guide du débutant pour un design web responsive (exemples de code et de mise en page)

<https://kinsta.com/fr/blog/design-web-responsive/>

## 2. Gérer les différentes zones de visualisation

### 2.1 Les requêtes média (*Media queries*)

Les requêtes média sont basées sur les règles CSS @media (voir W3C Media Queries Level 4 : <https://www.w3.org/TR/mediaqueries-4/> ) que l'on peut décomposer ainsi :

@media	screen	and	(min-width:320px)	and	(max-width:768px)
Règle média précédé d'un @	Le type de média	opérateur	Fonctionnalité 1	opérateur	Fonctionnalité 2

Les requêtes média permettent de faire correspondre la taille de l'écran à des plages de valeurs. Ainsi, comme on peut le voir dans l'exemple ci-après les règles CSS contenu dans la règle média ne sera appliqué que si la taille (et non la résolution) de l'écran correspond à cette plage de valeur. Si la taille de l'écran est comprise entre 320px et 768px les règles CSS media s'appliqueront écrasant celles écrit plus haut dans le code.

```
body {  
    /* La règle CSS par défaut */  
    background-color: cyan;  
}  
  
@media screen and (min-width:320px) and (max-width:768px){  
    /* La règle CSS ne s'appliquant que lorsque la taille de l'écran est entre 320 < 768 pixels */  
    body {background-color: magenta};  
}
```

Voir le fichier : test01mq.html

Attention aux espaces dans une règle @media ! Il ne faut pas d'espace entre l'arobase et media, et certains navigateurs n'apprécient pas plus d'un espace entre les paramètres de la règle media (entre les opérateurs et les fonctionnalités).

#### Références

Une présentation des *Media Queries* par Pierre Giraud :

<https://www.pierre-giraud.com/html-css-apprendre-coder-cours/media-queries-responsive-design/>  
avec un exemple en ligne : <https://codepen.io/pierregiraud/pen/ZVXRgp>

Je vous recommande l'excellent Guide de Chris Coyer sur CSS-Tricks :

*A Complete Guide to CSS Media Queries*

<https://css-tricks.com/a-complete-guide-to-css-media-queries/>

## 2.2 Les points de rupture (*breakpoints*)

Quelles taille d'écran devons nous concevoir ? C'est un thème récurrent dans le développement Web. Les sites *Web Responsive* bien construit s'appuie sur un petit nombre de taille. Mais quelles tailles utilisées ? Il y a plusieurs approche sur le sujet :

- On peut se baser sur un groupe d'appareils : mobile, tablette, PC portable et de bureau.
- Reprendre les points de rupture des framework(s) populaires comme *Bootstrap*.
- Utiliser les tailles d'écran de vos visiteurs (statistiques).
- Repartir sur les points de rupture les plus utilisés de l'année précédente.

Pour les tailles d'appareils courant :

*Media Queries for Standard Devices*

<https://css-tricks.com/snippets/css/media-queries-for-standard-devices/>

Breakpoints from Bootstrap 5.0

<https://getbootstrap.com/docs/5.0/layout/breakpoints/>

```
// x-small devices (phone)
@media only screen and (max-width: 575px) {

}
// Small devices (phone, landscape phones)
@media only screen and (min-width: 576px) and (max-width: 767px){

}
// Medium devices (tablets)
@media only screen and (min-width: 768px) and (max-width: 991px){

}
// Large devices (Laptop)
@media only screen and (min-width: 992px) and (max-width: 1199px){

}
// x-large devices (desktop)
@media only screen and (min-width: 1200px) and (max-width: 1399px){

}
// xx-large devices (larger desktop and up)
@media only screen and (min-width: 1400px){

}
```

*The breakpoints we tested in 2021, and the ones to test in 2022*

<https://polypane.app/blog/the-breakpoints-we-tested-in-2021-and-the-ones-to-test-in-2022/>

La référence des statistiques sur la résolution des écrans

<https://gs.statcounter.com/>

Ce site est indispensable pour la SEO !

## 2.3 Les unités de mesure et la police *responsive*

Comment créer un site Web réactif avec deux unités de mesure CSS : EM et REM ?

em et rem sont des unités flexibles et évolutives qui sont traduites par le navigateur en valeurs de pixels, en fonction des paramètres de taille de police de votre conception.

Comme je l'ai mentionné ci-dessus, les unités rem et em sont traduites en pixels. Pixel est l'unité principale et l'unité fixe pour mesurer chaque taille d'écran.

### ***Comment fonctionne l'unité REM ?***

Les unités REM sont basées sur l'élément racine. Lors de l'utilisation d'unités rem, la taille de pixel qu'elles traduisent dépend de la taille de la police de l'élément racine de la page, c'est-à-dire l'élément HTML. Cette taille de police racine est multipliée par le nombre que vous utilisez avec votre unité rem.

Remarque : 1 rem = 16 px

L'unité REM offre une flexibilité dans notre conception. REM Unit offre une excellente occasion de modifier la typographie et l'espacement de l'ensemble du site en modifiant simplement la taille de la police à un endroit dans un élément racine.

Nous concluons donc que rem est utilisé lorsque nous voulons apporter des modifications à tout un site car il nous permet de ne modifier la taille de la police qu'à un seul endroit.

### ***Comment fonctionne l'unité EM ?***

L'unité EM est basée sur la taille de police du même élément où elle est définie, si le même élément n'a pas de taille de police définie, **il héritera de son parent ou d'un autre élément ancêtre**, en remontant éventuellement jusqu'au racine du document. Lors de l'utilisation d'unités em, la valeur du pixel est multipliée par la taille de la police de l'élément donné.

par exemple, vous pouvez définir le rembourrage, la marge et la hauteur de ligne autour de la navigation, élément de menu pour utiliser les valeurs em. De cette façon, si vous modifiez la taille de la police du menu, l'espacement autour des éléments du menu sera mis à l'échelle proportionnellement, indépendamment du reste de la mise en page.

Comme dans notre exemple ci-dessus, les composants de conception tels que les éléments de menu, les boutons et les en-têtes peuvent avoir leurs propres tailles de police explicitement indiquées. Si vous modifiez ces tailles de police, vous souhaitez que l'ensemble du composant soit mis à l'échelle proportionnellement.

Les propriétés communes auxquelles cette directive s'appliquera sont les paramètres de marge, de remplissage, de largeur, de hauteur et de hauteur de ligne lorsqu'ils sont utilisés sur des éléments avec une taille de police autre que celle par défaut.

## Conception Web réactive à l'aide des unités REM et EM

Maintenant que nous avons tout compris sur les unités REM et EM et sur leur fonctionnement. Il est temps de voir comment nous utilisons cette flexibilité pour créer un site Web réactif.

La plupart de l'espacement et de la taille sont ajustés par l'unité REM en définissant la police dans l'élément racine.

Une chose importante à garder à l'esprit est que vous utilisez toujours des valeurs de pourcentage lors de la définition de la valeur *font-size* dans l'élément racine, par exemple, HTML. L'avantage d'utiliser le pourcentage sur les pixels est que notre élément racine n'a plus de valeur fixe codée en dur.

```
html{  
  font-size: 62.5%; // 10px  
}
```

La règle des 62.5%, déclarer dans la balise html, permet d'harmoniser les tailles px / rem.

Pour obtenir la taille Rem depuis le pixel : multiplier par 10

Pour obtenir la taille px depuis Rem : diviser par 10

Ainsi 200 px = 20 rem et 48 rem = 480 px

- Les unités rem et em sont calculées en valeurs de pixel par le navigateur, en fonction des tailles de police de votre conception. Les unités em sont basées sur la taille de police de l'élément sur lequel elles sont utilisées.
- Les unités rem sont basées sur la taille de la police de l'élément HTML. les unités em peuvent être influencées par l'héritage de la taille de police de n'importe quel élément parent
- Les unités rem peuvent être influencées par l'héritage de la taille de police des paramètres de police du navigateur.
- Utilisez les unités **em** pour un dimensionnement qui doit être mis à l'échelle en fonction de la **taille de la police** d'un élément autre que la racine.
- Utilisez les unités **rem** pour un dimensionnement qui n'a pas besoin d'unités em, et qui doit être mis à l'échelle en fonction des paramètres de taille de police du navigateur.
- Utilisez les unités rem sauf si vous êtes sûr d'avoir besoin d'unités em, y compris les tailles de police.
- Utiliser les unités **rem sur les requêtes media**
- **N'utilisez pas em ou rem dans les largeurs de mise en page multi-colonnes - utilisez % à la place.**
- N'utilisez pas em ou rem si la mise à l'échelle entraînerait inévitablement la rupture d'un élément de mise en page.

### Références

Pourquoi j'utilise l'unité Rem et non l'unité Pixel ? :

<https://blog.lesieur.name/pourquoi-j-utilise-l-unite-rem-et-non-l-unite-pixel/>

Unités de mesures : EM vs REM

<https://medium.com/codshake/unit%C3%A9s-de-mesures-em-vs-rem-eac03dbcb9c7>

Convertisseur Px, Em, Rem, %, Pt.

<https://outils-css.aliasdmc.fr/convertir-unites-de-longueurs-css.php>

### 3. Diverses techniques du fluide au *responsive*

#### Mobile First

IONOS : <https://www.ionos.fr/digitalguide/sites-internet/web-design/mobile-first-la-nouvelle-approche-du-web-design/>

anthedesign.fr : <https://www.anthedesign.fr/webdesign-2/mobile-first/>

Luke WROBLEWSKI, *Mobile First*, Eyrolles, A book apart N°6, 2012.