

CSS media queries

Mathilde Vigouroux & Chloé Dubois

1. À quoi servent les media queries ?

À l'heure actuelle, l'un des grands enjeux du développement web est de créer des sites web qui fonctionnent, qui restent lisibles et agréables à lire, quelle que soit la taille du viewport (zone d'affichage) et sans égard du type de *devices* (ou "appareils") sur lesquels ils sont consultés (smartphone, ordinateur, tablette, télévision, voire montre connectée). Un site web doit aujourd'hui être capable de répondre à ce besoin d'adaptation à ces variations. On parle de web *responsive*, notion que vous avez sûrement déjà entendue.

Bien qu'il existe désormais des solutions pour créer un layout adaptatif ou *responsive* (par exemple flexbox ou css grid + "auto-fit" ou "auto-fill"), les **media queries** restent la 'pierre angulaire' d'un design web *responsive*.

En très bref, les *media queries* sont une fonctionnalité CSS3 qui permet de spécifier quand ou sous quelles conditions certaines instructions css doivent être appliquées.

Elles nous permettent de cibler les instructions css en fonction de différents paramètres, parmi lesquels le **type de média** sur lequel le site est visionné et certaines **caractéristiques** telles que la dimension du viewport, l'orientation (portrait vs paysage), le ratio (ex. 16/9), la couleur (color vs monochrome).

1.1. Les types de médias

all	Valable pour <u>tous</u> les appareils
braille *	Valable pour les <u>appareils tactiles</u> en braille
embossed *	Valable pour les <u>imprimantes braille</u>
handheld *	Valable pour les <u>appareils portables</u> Attention: les smartphones et tablettes ne répondent pas à handheld (utilisez screen)
print	Valable pour les <u>imprimantes</u> et l'aperçu d'impression des navigateurs.
projection *	Valable pour les présentations projetées, les <u>projecteurs</u> .
screen	Valable pour les <u>écrans d'ordinateur</u> en couleur (incluant desktop, laptop, tablettes <u>et</u> smartphones)
speech	Valable pour les <u>synthétiseurs vocaux</u>
tty *	Valable pour les media qui utilisent Used for media using a fixed-pitch character grid (such as teletypes, terminals, or portable devices with limited display capabilities).
tv *	Valable pour les appareils du type <u>télévision</u> (couleur, son, mais pas de scroll)

* types obsolètes selon <https://www.w3.org/TR/mediaqueries-4/> Le site recommande aux auteurs de plutôt utiliser les caractéristiques media appropriées.

1.2. Les caractéristiques media

❖ Dimensions du viewport ou du device * :

width, height, min-width, min-height, max-width, max-height, device-height, device-width, min-device-height, etc.

```
@media screen and (device-width: 800px) { ... }
```

❖ Orientation portrait vs paysage

portrait, landscape

```
@media all and (orientation:portrait) { ... }
```

```
@media all and (orientation:landscape) { ... }
```

❖ Ratio (rapport hauteur/largeur)

aspect-ratio, device-aspect-ratio, min-aspect-ratio, max-aspect-ratio, etc.

```
@media screen and (device-aspect-ratio: 16/9) { ... }
```

❖ Couleur :

color, monochrome, min-color, max-color (+nbr de bits par composant de couleur)

❖ Résolution :

Resolution, min-resolution, max-resolution, etc.

```
@media print and (min-resolution: 300dpi) { ... }
```

❖ ...

* Remarque sur le fonctionnement de l'affichage d'un site sur mobile :

Les navigateurs mobiles affichent le site en “dézoomant” la page pour avoir une vue d'ensemble. La zone d'affichage simulée est appelée le **viewport** : c'est la largeur de la fenêtre du navigateur sur le mobile. Il faut savoir que le viewport change selon le navigateur utilisé (Exemple : Safari sur iPhone a un viewport par défaut de 980px, alors qu'un Android va afficher un viewport par défaut de 800px).

2. Comment intégrer les media queries dans votre projet ?

Il y a plusieurs manières d'intégrer les media queries dans un projet :

2.1. Plusieurs feuilles de style css

On a la possibilité de créer plusieurs feuilles de style (css stylesheet) pour chaque version en spécifiant l'attribut `media=' '` dans le lien vers les stylesheets du document html.

Ex. : on crée des feuilles de style différentes pour print pour les pages couleur ou monochrome.

```
<link rel="stylesheet" media="all" href="normal.css" />
```

```
<link rel="stylesheet" media="print and (color)" href="color-print.css" />
```

```
<link rel="stylesheet" media="print and (monochrome)" href="mono-print.css" />
```

[à ce sujet : <https://css-tricks.com/resolution-specific-stylesheets/>]

2.2. À l'intérieur de la feuille de style css

On intègre les media queries dans notre feuille de style en utilisant l'attribut `@media`

Ex. Ici on change la couleur de la font

```
.someContent {  
    color: green;  
    background-color: red;  
}  
  
@media screen and (min-width: 400px) {  
    .someContent {  
        color: yellow;  
    }  
}
```

Effet en “cascade” : Seules les propriétés spécifiées dans la règle `@media` sont modifiées, les autres propriétés (spécifiées en amont de la feuille de style) restent valables.

3. Syntaxe d'une CSS media query

3.1. Deux types de conditions : *media type* & *media features*

Dans la règle @media, on place **deux sortes de conditions (facultatives)**:

1. Une condition sur le **type de media** utilisé pour afficher la page

all est la valeur par défaut (les règles s'appliquent à tous les appareils). Il existe d'autres valeurs (voir tableau) comme screen, printer ou speech, etc.

```
@media screen { }
```

2. Une condition sur les **caractéristiques du media**.

```
@media (max-width: 699px) { }
```

Le type de média est toujours précisé en premier (avant les caractéristiques). Les **parenthèses** sont utilisées pour encadrer les **caractéristiques media** spécifiées.

3.2. La virgule et les opérateurs logiques *and*, *not* et *only*

Une media query fonctionne selon une **logique booléenne**. Elle est évaluée comme **TRUE** si la ou les conditions précisées sont **TRUE**, c'est-à-dire quand le type de média (si précisé) correspond au device sur lequel le document est consulté et/ou quand les conditions en termes de caractéristiques média (s'il y en a) sont remplies.

❖ **La virgule :**

On peut lister plusieurs queries en les séparant par une **virgule**. La virgule correspond à l'expression logique OR ("ou"). Dans ce cas, chaque requête est analysée individuellement et, si l'une d'entre elle est TRUE, les instructions seront appliquées.

❖ L'opérateur **and** :

L'opérateur **and** est utilisé lorsqu'on souhaite que plusieurs conditions soient remplies pour que la requête soit évaluée comme valide (TRUE).

```
@media all and (max-width: 699px) and (min-width: 520px) { }
```

❖ L'opérateur **only** :

L'opérateur **only** ("uniquement") indique que la règle media est TRUE uniquement pour le type de media choisi. Cela permet éventuellement de cacher la media queries aux anciens navigateurs qui peuvent rencontrer des problèmes de lecture et ainsi éviter des bugs.

```
@media only screen and (max-device-width: 480px) { }
```

❖ L'opérateur **not** :

L'opérateur **not** a pour effet d'inverser les conditions. Cela signifie que, si la requête est TRUE, on n'interprète pas ce qui est entre accolades. Il a le même effet que **only** sur les anciens navigateurs.

```
@media not screen and (max-device-width: 480px) { }
```

Pour résumer :

@media + **only** + type de media + **and** (caractéristiques des media) , (autres caractéristiques)

4. Démo

<https://codepen.io/s10wen/pen/ZvRbyX> → cas concret simple

Le texte de base est vert.

Le texte devient jaune quand on lit le contenu sur tous les screen (écrans) et les écrans avec une (min-width: 400px)

Le texte devient rouge quand on lit le contenu sur tous les écrans et les media avec une (min-width: 600px)

<https://codepen.io/SitePoint/pen/pJjjKj/> → lien vers un autre exemple avec une sidebar

On peut modifier le positionnement de la sidebar.

Elle passe au dessus quand on lit le contenu quand le media a une max-width de 40em.

Au dessus elle passe sur le côté gauche de l'écran.

<https://codepen.io/chris0stein/pen/DwBHr> → autre exemple avec des box

Pour seulement les écrans et les media ayant une (max-width: 20em) les box flottent en dessous du titre et prennent la largeur de l'écran.

Pour seulement les écrans et les media ayant une (min-width: 30.063em) et les medias ayant une (max-width: 50em) les box passent à gauche de l'écran et le titre du côté droit.

Conclusion:

On peut ainsi régler précisément les break-point en fonction des devices sur lesquels le site sera visible et gérer le positionnement du contenu.

5. Remarques finales

5.1. Cibler un device particulier

Ces pages montrent comment cibler un device particulier et en expliquent les limites :

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

<http://cssmediaqueries.com/target/>

Attention : Cibler un device particulier est considéré comme une mauvaise pratique, car on ne peut pas savoir quels devices vont répondre à la query. Il est préférable de spécifier les breakpoints (“points de rupture”) en se basant sur le design du site plutôt que sur les caractéristiques d’un device particulier.

5.2. Browser support

Les media queries sont compatibles avec la majorité des navigateurs, mais vous pouvez consulter le site “Can I Use” pour vérifier (si vous utilisez un type de media différent de screen par exemple).

<https://caniuse.com/#search=%40media>

7. Petite sitographie

Explications générales sur comment utiliser les media queries :

<http://cssmediaqueries.com/what-are-css-media-queries.html>

<https://css-tricks.com/css-media-queries/>

https://developer.mozilla.org/en-US/docs/Web/CSS/Media_Queries/Using_media_queries

<https://www.w3.org/TR/mediaqueries-4/>

<https://www.freecodecamp.org/news/css-media-queries-breakpoints-media-types-standard-resolutions-and-more/>

<https://www.chiny.me/les-media-queries-responsive-design-5-14.php>

<https://openclassrooms.com/fr/courses/1603881-apprenez-a-creez-votre-site-web-avec-html5-et-css3/1607616-utilisez-le-responsive-design-avec-les-media-queries>

Pages qui listent différents devices et les medias queries qui les ciblent plus particulièrement :

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

<http://cssmediaqueries.com/target/>