

Bienvenue ! Alsacréations est une communauté dédiée à l'apprentissage des standards web (HTML, CSS, JavaScript), du design et de l'accessibilité numérique. Vous y trouverez tutoriels, forum, annonces d'emploi, quiz, concours et bien d'autres choses. [Créez votre profil](#) en quelques secondes pour participer ou consultez notre [politique de protection des données](#).

[Masquer](#)

Les Media Queries CSS3

Niveau

ARTICLE cssPublié par [Rodolphe](#) le 07 Janvier 2011, mis à jour le 08 Février 2019 (1340821 lectures)[css](#) [html](#) [css3](#) [media queries](#) [media](#) [@media query](#).

La [spécification CSS3 Media Queries](#) définit les techniques pour l'application de feuilles de styles en fonction des périphériques de consultation utilisés pour du HTML. On nomme également cette pratique **Responsive Web Design**, pour dénoter qu'il s'agit d'adapter dynamiquement le design à l'aide de CSS.

Ces bonnes pratiques permettent d'exploiter encore plus les avantages de la séparation du contenu et de la présentation : l'intérêt est de pouvoir satisfaire des contraintes de dimensions, de résolutions et d'autres critères variés pour améliorer l'apparence graphique et la lisibilité (voire l'utilisabilité) d'un site web. Les plateformes exotiques sont concernées en premier lieu : navigateurs mobiles et tablettes, écrans à faibles résolutions, impression, tv, synthèses vocales, plages braille, etc.



CSS3

Media Queries

Approche historique

Avec CSS2 et HTML4, il était déjà possible de spécifier un média de destination pour l'application d'une ou plusieurs feuilles de style. C'est ainsi que l'on a pu associer des règles CSS complémentaires pour l'impression, modifiant la mise en page, favorisant tel élément ou faisant disparaître un autre inutile à la sortie sur papier, par exemple un menu de navigation. La balise `<link>` est alors dupliquée pour autant de feuilles de style que nécessaire, et comporte un attribut `media` précisant le contexte dans lequel les styles doivent être appliqués :

```
<!doctype html>
<head>
<meta charset="utf-8">
<title>Media Queries !</title>
<link rel="stylesheet" media="screen" href="screen.css" type="text/css" />
<link rel="stylesheet" media="print" href="print.css" type="text/css" />
</head>
<body>
...
</body>
```

HTML

L'attribut `media` peut prendre (depuis CSS2) les valeurs suivantes :

screen

Écrans

handheld

Périphériques mobiles ou de petite taille

print

Impression

aural (CSS 2.0) / speech (CSS 2.1)

Synthèses vocales

braille

Plages braille

embossed

Imprimantes braille

projection

Projecteurs (ou présentations avec slides)

tty

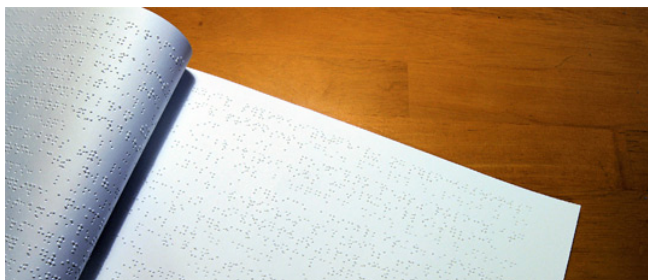
Terminal/police à pas fixe

tv

Téléviseur

all

Tous les précédents



Une impression braille (embossed)

Ces directives peuvent parfaitement être intégrées au sein même d'une feuille de style grâce à une règle `@media` suivie directement du type. La syntaxe sera alors légèrement différente :

```
@media print {  
  #menu, #footer, aside {  
    display:none;  
  }  
  body {  
    font-size:120%;  
    color:black;  
  }  
}
```

CSS

Note : d'autres variantes peuvent exister avec la règle `@import`, cependant celle-ci pose quelques problèmes de performance sur d'anciens navigateurs et se voit ignorée lorsqu'elle est utilisée en combinaison avec les *media queries* sur Internet Explorer versions inférieures à 8. Ce point ne sera donc pas approfondi ici.

Or, les médias peuvent être très variés, surtout en ce qui concerne les écrans. De surcroît, tout dépend du support de l'interprétation de ces définitions. Rien n'oblige un périphérique ou un navigateur à appliquer ce qui semblerait le plus indiqué. Par exemple, la règle media `handheld` est ignorée par la grande majorité des navigateurs mobiles, y compris par Safari Mobile sur iOS qui se considère comme un média `screen`.

Syntaxe des Media Queries CSS3

La philosophie des Media Queries (ou requêtes de media) en CSS3 est d'offrir un panel de critères plus vaste et plus précis, à l'aide de propriétés et de valeurs numériques, ainsi que de combinaisons multiples de ces mêmes critères. Le but est de cibler plus finement les périphériques de destination en fonction de leurs capacités intrinsèques.

L'écriture de ces requêtes est relativement explicite (en anglais) : une *media query* est une **expression dont la valeur est toujours vraie ou fausse**. Il suffit d'associer les différentes déclarations possibles avec un opérateur logique pour définir l'ensemble des conditions à réunir pour l'application des styles compris dans le bloc adjacent.

Les opérateurs logiques peuvent être :

- `and` "et",
- `only` "uniquement"
- `not` "non".

Pour obtenir l'équivalent du "ou", il suffit d'énumérer différentes *media queries* à la suite, séparées par des **virgules** : si l'une d'entre elles est valable, alors l'ensemble de la règle sera appliquée.

En général, on combine ensemble un type de média (screen, all...) et une expression grâce à `and`, bien qu'une expression seule puisse être utilisée. L'expression est toujours écrite entre parenthèses.

Les deux exemples suivants ciblent les écrans de largeur inférieure à 640 pixels grâce à la règle `max-width` associée à la valeur `640px` :

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

HTML

```
@media screen and (max-width: 640px) {  
  .bloc {  
    display:block;  
    clear:both;  
  }  
}
```

CSS

Les combinaisons peuvent être multiples. Ici l'on s'adresse à un écran dont la résolution en largeur est comprise entre 200 et 640 pixels :

```
@media screen and (min-width: 200px) and (max-width: 640px) {  
  .bloc {  
    display:block;  
    clear:both;  
  }  
}
```

CSS

Fonctionnalités

La plupart des critères (ou fonctionnalités) peuvent être préfixés par `min-` et `max-` lorsqu'elles acceptent des valeurs numériques pour définir des valeurs minimales ou maximales à respecter.

color

support de la couleur (bits/pixel)

color-index

périphérique utilisant une table de couleurs indexées

device-aspect-ratio

ratio du périphérique de sortie (par exemple 16/9)

aspect-ratio

ratio de la zone d'affichage

device-height

dimension en hauteur du périphérique

device-width

dimension en largeur du périphérique

grid

périphérique bitmap ou grille (ex : lcd)

height

dimension en hauteur de la zone d'affichage

monochrome

périphérique monochrome ou niveaux de gris (bits/pixel)

orientation

orientation du périphérique (portrait ou landscape)

resolution

résolution du périphérique (en dpi, dppx, ou dpcm)

scan

type de balayage des téléviseurs (progressive ou interlace)

width

dimension en largeur de la zone d'affichage

Les dimensions pourront être évaluées avec des unités (px, em). Les ratio avec des fractions (entier/entier). Une résolution sera définie en dpi (points par pouce) ou en dpcm (points par centimètres).

Certaines de ces propriétés peuvent être testées d'une façon raccourcie sans valeur, par exemple `(color)` qui sera équivalent à `(min-color: 1)` ou considérée comme vraie pour une valeur différente de 0. La fonctionnalité `monochrome` n'est pas uniquement booléenne avec la syntaxe raccourcie (monochrome), on peut aussi considérer un nombre de niveaux de gris, par exemple `(min-monochrome: 2)` pour 2 bits par pixel.

Démonstrations

Avec les navigateurs actuels il est tout à fait possible de tester les *media queries* "en direct" simplement en redimensionnant la fenêtre. Vous verrez les feuilles de style s'appliquer dynamiquement :



👁 Démonstration CSS media queries

Voici un exemple de disposition changeante en 3 étapes avec `min-width` et `max-width`.



👁 Démonstration CSS media queries avec mise en page

Au service des mobiles

Dans la majorité des cas, on utilise les *media queries* pour produire des améliorations spécifiques à l'affichage sur les mobiles, qui sont directement concernés par des critères sur les dimensions de l'écran (en terme de résolution et d'espace disponible) et sur l'utilisation tactile. Les périphériques sous iOS, Android ou d'autres systèmes mobiles supportent tous les *media queries*.



Ainsi on retrouvera le plus fréquemment des règles pour :

- agrandir la taille du texte
- agrandir la taille des contrôles et zones cliquables (pour une utilisation au doigt)
- faire passer le contenu sur une seule colonne
- masquer ou afficher des éléments spécifiques
- ajuster les dimensions et marges

La fonctionnalité `orientation` a été introduite pour des périphériques pouvant être orientés, par exemple l'iPad. On peut alors déclarer une feuille de style spécifique pour ajuster l'affichage :

```
<link rel="stylesheet" media="(orientation:portrait)" href="portrait.css">
<link rel="stylesheet" media="(orientation:landscape)" href="paysage.css">
```

HTML

Certains navigateurs classiques l'implémentent également en version bureau et calculent le ratio hauteur/largeur pour simuler un affichage "orienté".

Autres exemples

Impression sur un support plus large que 5 pouces :

```
print and (min-width: 5in)
```

CSS

Ecrans possédant un ratio 16/9 ou 16/10e :

```
tv, (device-aspect-ratio: 16/9), (device-aspect-ratio: 16/10)
```

CSS

🌐 Les sites web alsacreations.fr (agence web) et alsacreations.com bénéficient eux aussi de ce mécanisme pour affiner l'affichage sur petits écrans.