

# Comprendre le méta tag viewport

**Afin d'avoir un « responsive design » pour votre site Web**

Par Paul Underwood - [Sébastien Germez](#) (traducteur)

Date de publication : 27 novembre 2012

TOUT PUBLIC

Dans cet article nous allons apprendre à utiliser le méta tag viewport afin d'adapter le design de votre site web.

**Commentez**

I - Traduction.....	3
II - Introduction.....	3
III - Qu'est-ce que le « responsive design » ?.....	3
IV - Définir le « viewport ».....	4
V - La largeur du « viewport ».....	4
VI - Les échelles du « viewport ».....	5
VII - Exemples de « viewport ».....	5
VIII - Conclusion.....	6
IX - Remerciements.....	6

## I - Traduction

Ce tutoriel est la traduction la plus fidèle possible du tutoriel original de **Paul Underwood**, **Understanding The Viewport Meta Tag**.

## II - Introduction

Quand vous travaillez sur un nouveau design de site Web, une des choses auxquelles il faut penser est le « responsive design ». Est-ce que le site que vous allez développer a besoin d'un « responsive design » ?

Tous les sites Web aujourd'hui devraient prendre en considération l'aspect « responsive design ». Il y a encore certains développeurs qui pensent qu'un visiteur sur mobile peut toujours zoomer pour rendre une page lisible, mais le sentiment général est que de nos jours, chaque développeur devrait rendre son site facile d'accès et d'utilisation.

## III - Qu'est-ce que le « responsive design » ?

Le « responsive design » c'est lorsque le design de votre site s'adapte à la résolution de l'écran de votre visiteur. Ce concept se met en œuvre avec l'utilisation des « media queries » CSS3. Ces derniers vous permettent de modifier la largeur de vos éléments HTML en fonction de paramètres que vous définissez dans vos feuilles de style. En ajoutant simplement width: 100 % en CSS sur vos éléments HTML peut suffire à les rendre « responsive » pour des appareils mobiles.

Les différents paramètres définissent chacun une fourchette de largeur qui utilisera un style particulier. C'est typiquement des largeurs d'affichage différentes en fonction de l'appareil utilisé pour visiter le site : un smartphone, une tablette, un PC de bureau, etc.

Vous pouvez définir ces paramètres directement avec une largeur en pixels ou bien avec un ratio minimum en pixels.

Le code suivant peut vous aider à démarrer avec les « media queries ».

```
/* Smartphones (portrait and landscape) ----- */
@media only screen
and (min-width : 320px)
and (max-width : 480px) {
/* Styles */
}
/* Smartphones (landscape) ----- */
@media only screen
and (min-width : 321px) {
/* Styles */
}
/* Smartphones (portrait) ----- */
@media only screen
and (max-width : 320px) {
/* Styles */
}
/* iPads (portrait and landscape) ----- */
@media only screen
and (min-width : 768px)
and (max-width : 1024px) {
/* Styles */
}
/* iPads (landscape) ----- */
@media only screen
and (min-width : 768px)
and (max-width : 1024px)
and (orientation : landscape) {
/* Styles */
}
```

```

/* iPads (portrait) ----- */
@media only screen
and (min-width : 768px)
and (max-width : 1024px)
and (orientation : portrait) {
/* Styles */
}
/* Desktops and laptops ----- */
@media only screen
and (min-width : 1224px) {
/* Styles */
}
/* Large screens ----- */
@media only screen
and (min-width : 1824px) {
/* Styles */
}
/* iPhone 4 ----- */
@media
only screen and (-webkit-min-device-pixel-ratio : 1.5),
only screen and (min-device-pixel-ratio : 1.5) {
/* Styles */
}
    
```

## IV - Définir le « viewport »

Le méta tag « viewport » dicte au navigateur quel comportement il doit adopter pour afficher une page ; vous pouvez spécifier la largeur qu'adoptera le « viewport ».

Le « viewport » correspond à la partie de la page affichée à l'écran. Par exemple si vous visitez un site Web depuis un appareil mobile, et qu'à l'écran s'affiche le coin haut gauche de la page (comme s'il y avait un zoom) c'est que le « viewport » a été défini à une certaine largeur. En revanche, si la page entière du site s'affiche et qu'elle occupe tout votre écran, c'est que le « viewport » a été défini pour afficher toute la page.

Ci-dessous un exemple d'utilisation du méta tag « viewport ».

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

Cette ligne définit la largeur du « viewport » pour être la même que la taille de l'écran de l'appareil utilisé pour afficher le site. L'échelle d'affichage du site sera 100 % et l'échelle maximum également de 100 %.

Chaque navigateur mobile dispose de son propre « viewport » par défaut, la plupart de ces navigateurs utilisant un « viewport » par défaut de 980px ce qui signifie que la page sera affichée dans une section d'affichage de 980 pixels et qu'un zoom arrière sera effectué si nécessaire pour que la page s'affiche en plein écran.

Plus l'écran de votre appareil sera petit plus le zoom arrière sera important. Si vous comparez l'affichage d'une même page Web sur un iPhone et sur un iPad, le zoom sera plus important sur l'iPhone que sur l'iPad.

Changer la valeur du « viewport » vous permet de personnaliser la façon dont le navigateur va afficher votre site.

## V - La largeur du « viewport »

La largeur que vous définissez dans votre « viewport » revient à dire que c'est la largeur qui permet l'affichage le plus optimisé de votre site. Si vous avez développé une page optimisée pour un affichage sur iPhone, alors la largeur de votre « viewport » sera de 320 pixels.

```
<meta name="viewport" content="width=320">
```

Cependant, cette façon de procéder ne permet pas de faire du « responsive design ». En effet, si vous visualisez cette page sur une tablette, vous aurez un zoom arrière sur la page pour qu'elle s'affiche dans une section de 320 pixels seulement, laissant le reste de l'écran vide.

La meilleure solution pour que votre site ait un « responsive design » est de définir comme largeur de « viewport » la largeur du matériel utilisé pour visualiser la page Web.

```
<meta name="viewport" content="width=device-width">
```

## VI - Les échelles du « viewport »

Sur les appareils mobiles vous pouvez zoomer (en avant ou en arrière) grâce à un pincement de doigts, mais si vous définissez la largeur du « viewport » pour être celle de l'appareil utilisé pour visualiser la page, vous n'aurez pas besoin d'utiliser de zoom. Pour vous assurer que votre page s'affichera sans zoom, vous pouvez utiliser la propriété initial-scale pour définir le zoom par défaut.

En mettant l'échelle à 1, on s'assure qu'aucun zoom ne sera fait à l'affichage de la page.

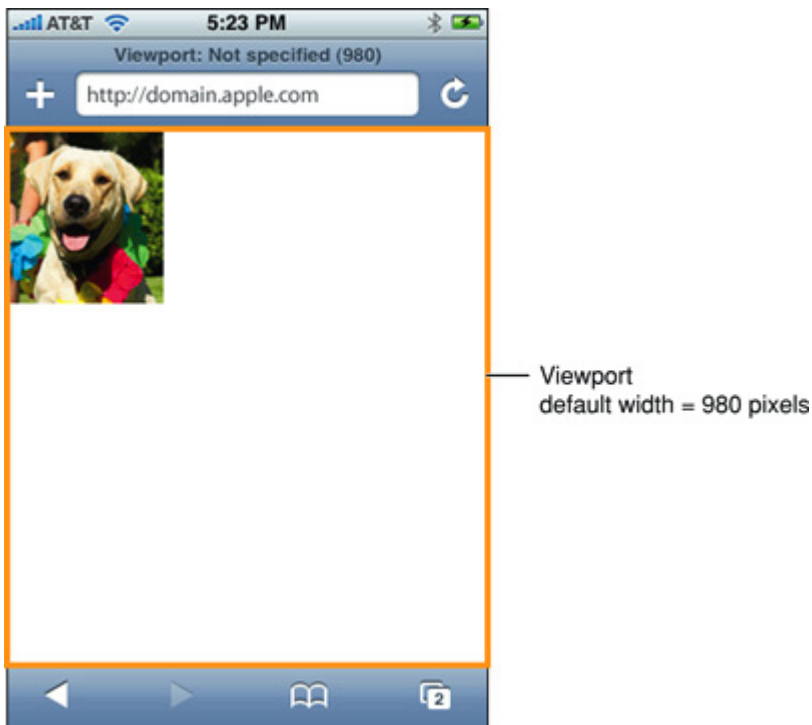
```
<meta name="viewport" content="initial-scale=1">
```

Si vous souhaitez empêcher tout zoom de la part de l'utilisateur, vous pouvez définir la propriété maximum-scale à 1 afin d'empêcher tout redimensionnement.

```
<meta name="viewport" content="maximum-scale=1">
```

## VII - Exemples de « viewport »

Les images suivantes montrent l'affichage de la même page avec deux « viewport » différents. La première image affiche la page dans un « viewport » de 980 pixels et la seconde dans un « viewport » de 320 pixels.





Viewport  
width = 320 pixels  
scale = 1.0

## VIII - Conclusion

Nous avons vu dans ce tutoriel comment utiliser le méta tag viewport afin d'avoir un « responsive design » sur notre site Web. Ce méta tag permet d'avoir un affichage de page dynamique en fonction de l'appareil utilisé pour afficher cette dernière (smartphone, tablette, laptop, etc.).

## IX - Remerciements

Je tiens à remercier **Paul Underwood** de m'avoir autorisé à traduire son tutoriel. Je remercie également **ClaudeLELOUP** pour sa relecture orthographique.