

Comment faire une page ayant la même hauteur que le navigateur ?

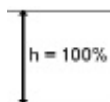
Par Anonyme



www.openclassrooms.com

Sommaire

Sommaire	2
Comment faire une page ayant la même hauteur que le navigateur ?	3
La technique CSS	3
La technique Javascript	4
Annexe	6
Partager	7

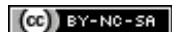


Comment faire une page ayant la même hauteur que le navigateur ?

Par Anonyme

Mise à jour : 03/03/2010

Difficulté : Facile

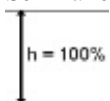


Vous avez certainement cherché une fois à réaliser une page reprenant la hauteur exacte du navigateur du visiteur. Dans ce tutoriel, je vous expliquerai comment réaliser cette page et vous montrerai le cas d'une utilisation de celle-ci.



Ces techniques ne sont pas compatibles avec tous les navigateurs, surtout Internet Explorer 5.x. Des connaissances en Javascript vous seront aussi utiles.

Sommaire du tutoriel :



- [La technique CSS](#)
- [La technique Javascript](#)
- [Annexe](#)

La technique CSS

Cette première technique utilise seulement les CSS pour fonctionner.

Tout d'abord, commençons par faire le squelette de notre page.

Nous allons faire une page simple, avec un bloc (div, p, ou autre) que nous allons essayer d'étendre en hauteur pour qu'il fasse la même hauteur que la zone d'affichage du navigateur du visiteur.

Voici donc notre page:

Code : HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
  <title>Document sans titre</title>
  <link href="design/design.css" rel="stylesheet" type="text/css" />
</head>

<body>
  <div id="page">
    Ceci est un essai de page de hauteur 100%
  </div>
</body>
</html>
```

Nous allons maintenant nous pencher sur la CSS à utiliser.

Vous devez sûrement connaître la propriété **height** en CSS, qui permet de donner une hauteur à un bloc. C'est cette propriété que nous allons utiliser pour notre bloc.

Voici donc notre CSS:

Code : CSS

```
body{
  margin: 0px;
  padding: 0px;
  outline: 0px;
  background-color: yellow;
}

#page{
  background-color: red;
  height: 100%;
  width: 100%;
}
```

[Voir ce code en action](#)

Comme vous pouvez le voir, le bloc **page** fait bien toute la largeur du navigateur, mais il ne fait pas toute sa hauteur. C'est un des problèmes de **Height: 100%**, la plupart des navigateurs ne savent pas comment l'interpréter. Pourtant **Width: 100%** est bien interprété par nos navigateurs, sans erreur.



Comment veux-tu réaliser ton bloc alors? 🤔

Eh bien, en utilisant une solution très simple:

Code : CSS

```
html, body{
  margin: 0px;
  padding: 0px;
  outline: 0px;
  height: 100%;
  background-color: yellow;
}

#page{
  background-color: red;
  height: 100%;
  width: 100%;
}
```

[Voir ce code en action](#)

Comme vous pouvez le voir, le bloc **page**, avec son fond rouge, recouvre maintenant tout le fond jaune de **Body**. Cette technique force la balise **Body** et **Html** à faire exactement la hauteur du navigateur.

Mais comme je vous le disais plus haut, cette technique n'est pas compatible avec tous les navigateurs.

Nous allons donc maintenant voir une autre technique compatible avec la plupart des navigateurs.

La technique Javascript

Nous allons donc maintenant voir une autre technique, utilisant le Javascript.



Cette technique n'est utilisable qu'à partir de IE6!

Nous allons reprendre le même squelette que dans la partie précédente, tout en y insérant notre code Javascript

Code : HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-
8859-1" />
<link href="test3.css" rel="stylesheet" type="text/css" />
<title>Document sans titre</title>
</head>

<body>
<div id="page">
  Ceci est un essai de page de hauteur 100%
</div>
</body>
</html>
```

Définissons, dans la foulée, la feuille de style associée:

Code : CSS

```
body{
  margin: 0px;
  padding: 0px;
  outline: 0px;
  background-color: yellow;
}

#page{
  background-color: red;
  width: 100%;
}
```

Maintenant, réfléchissons...

Nous voulons que le bloc **page** fasse exactement la même hauteur que la zone d'affichage du navigateur du visiteur.

Il faut donc pouvoir reprendre cette valeur 😊 !

C'est ici que se pose un problème:



Tous les navigateurs ne fonctionnent pas de la même manière, ils ont chacun leur méthode de lecture du HTML et du Javascript, qui peuvent donc différer d'un navigateur à l'autre.

Revenons à nos moutons!

Nous voulons donc prendre la hauteur de la zone d'affichage du navigateur.

Deux propriétés se proposent alors à nous : la propriété **window.innerHeight** et la propriété **document.documentElement.clientHeight**.

À présent, nous devons vérifier le navigateur du visiteur, et donc utiliser la propriété appropriée à son navigateur.

Une simple condition vérifiant que les propriétés marchent fera l'affaire:

Code : JavaScript

```
var hauteur;
```

```
if( typeof( window.innerHeight ) == 'number' )
    hauteur = window.innerHeight;
else if( document.documentElement &&
document.documentElement.clientHeight )
    hauteur = document.documentElement.clientHeight;
```

Maintenant que nous avons la hauteur de notre zone d'affichage, il ne nous reste plus qu'à l'appliquer.

Nous allons en profiter pour faire de ce code une fonction, et appliquer cette fonction à l'événement **onload()** de notre page :

Code : JavaScript

```
function height(bloc){
    var hauteur;

    if( typeof( window.innerWidth ) == 'number' )
        hauteur = window.innerHeight;
    else if( document.documentElement &&
document.documentElement.clientHeight )
        hauteur = document.documentElement.clientHeight;

    document.getElementById(bloc).style.height = hauteur+"px";
}

window.onload = function(){ height("page") };
```

Il ne nous reste plus qu'un détail à régler !

Je ne sais pas si vous l'avez remarqué, mais la fonction est seulement appelée au chargement de notre page. Si le visiteur redimensionne son navigateur, le bloc ne sera pas redimensionné, il aura la même hauteur qu'au moment du chargement de la page. Pour remédier à ce problème, il suffit d'appeler notre fonction au moment où la page est redimensionnée, avec l'événement **onresize()**



L'événement **resize()** a un comportement différent selon le navigateur, par exemple, Firefox fait appel à cette fonction pendant toute la durée du redimensionnement tandis que Safari l'appelle seulement une fois le redimensionnement terminé.

Code : JavaScript

```
function height(bloc){
    var hauteur;

    if( typeof( window.innerWidth ) == 'number' )
        hauteur = window.innerHeight;
    else if( document.documentElement &&
document.documentElement.clientHeight )
        hauteur = document.documentElement.clientHeight;

    document.getElementById(bloc).style.height = hauteur+"px";
}

window.onload = function(){ height("page") };
window.onresize = function(){ height("page") };
```

[Voir ce code en action](#)

Annexe

Le tableau ci-dessous recense une liste non exhaustive de navigateurs ainsi que leur compatibilité avec les différentes

techniques étudiés durant ce tutoriel:

Nom du navigateur	CSS	JavaScript
Internet Explorer 3.1	Non compatible	Non compatible
Internet Explorer 4.1	Non compatible	Non compatible
Internet Explorer 5.1	Non compatible	Non compatible
Internet Explorer 5.55	Compatible	Non compatible
Internet Explorer 6.1	Compatible	Compatible
Internet Explorer 7	Compatible	Compatible
Internet Explorer 8	Compatible	Compatible
FireFox 1	Compatible	Inconnu
FireFox 1.5	Compatible	Compatible
FireFox 2	Compatible	Compatible
FireFox 3.5.1	Compatible	Compatible
Opera 9	Compatible	Compatible
Opera 10	Compatible	Compatible
Safari 4.0	Compatible	Compatible

Ce tutoriel est maintenant terminé, j'espère que celui-ci vous a été utile 😊



Vous pouvez bien sûr appliquer ces techniques à plusieurs blocs dans une même page!

Si vous avez des questions ou autres, envoyez-moi un MP 😊

Partager

