

# Comprendre l'emplacement des éléments HTML

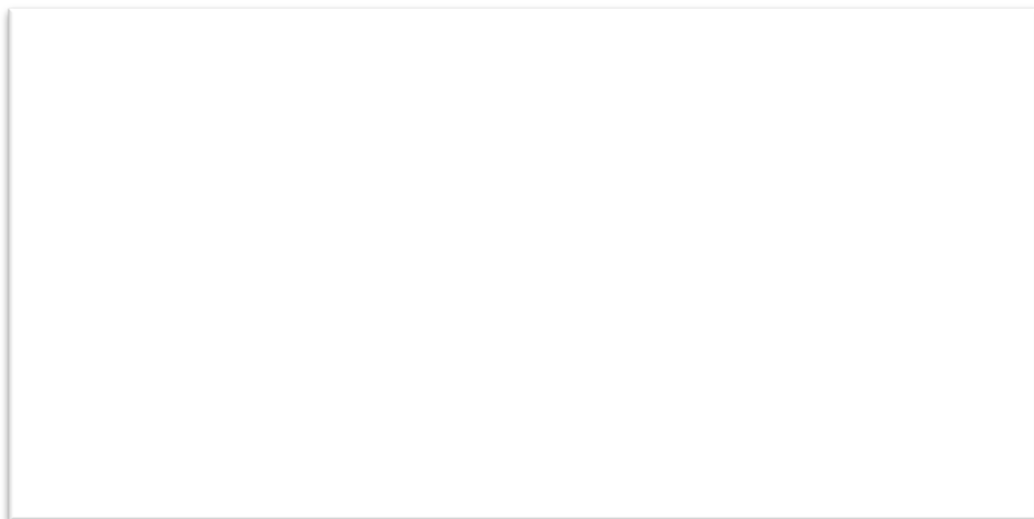
---

L'objectif de ce résumé est de vous expliquer ce qui se passe dans votre navigateur quand vous créez une balise HTML et le rôle de CSS.

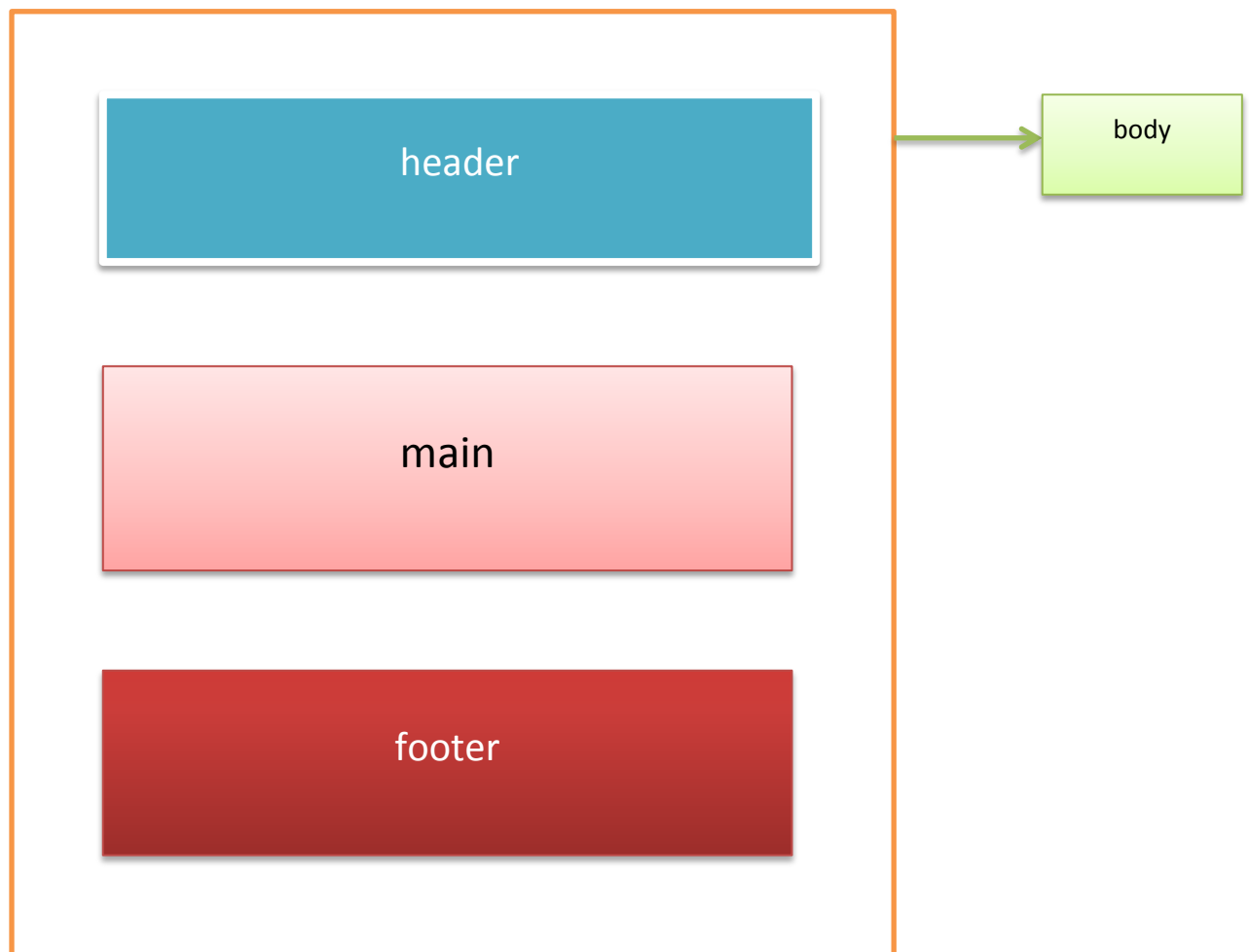
Commençons tout d'abord par créer notre fichier HTML :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>Document</title>
</head>
<body>
  <header></header>
  <main></main>
  <footer></footer>
</body>
</html>
```

Si vous ouvrez votre page HTML dans le navigateur, vous allez avoir une page blanche, n'est-ce pas ?



Mais en réalité, quand vous créez votre page HTML avec les balise : BODY, HEADER, MAIN et FOOTER, ce que votre ordinateur fait réellement, c'est qu'il va créer le schéma virtuel suivant :



Comme je l'ai mentionné ci-dessus, c'est un schéma virtuel, qui permettra de préciser l'emplacement de chaque élément de la page web, de cette manière ; le navigateur sait qu'un élément qui sera dans le header, se placera systématiquement au sommet de la page et qu'il ne va pas se placer au niveau du « main » et ainsi de suite.

Vous vous posez peut-être la question ; mais comment fait le navigateur pour savoir quelle largeur ou hauteur faut-il donner au header, main et footer ?

Très bonne question ! La réponse est simple, c'est vous qui avez ce pouvoir magique de définir la largeur et la hauteur d'un élément ce qui permet de définir la surface occupée par ce dernier grâce aux propriétés : width et height.

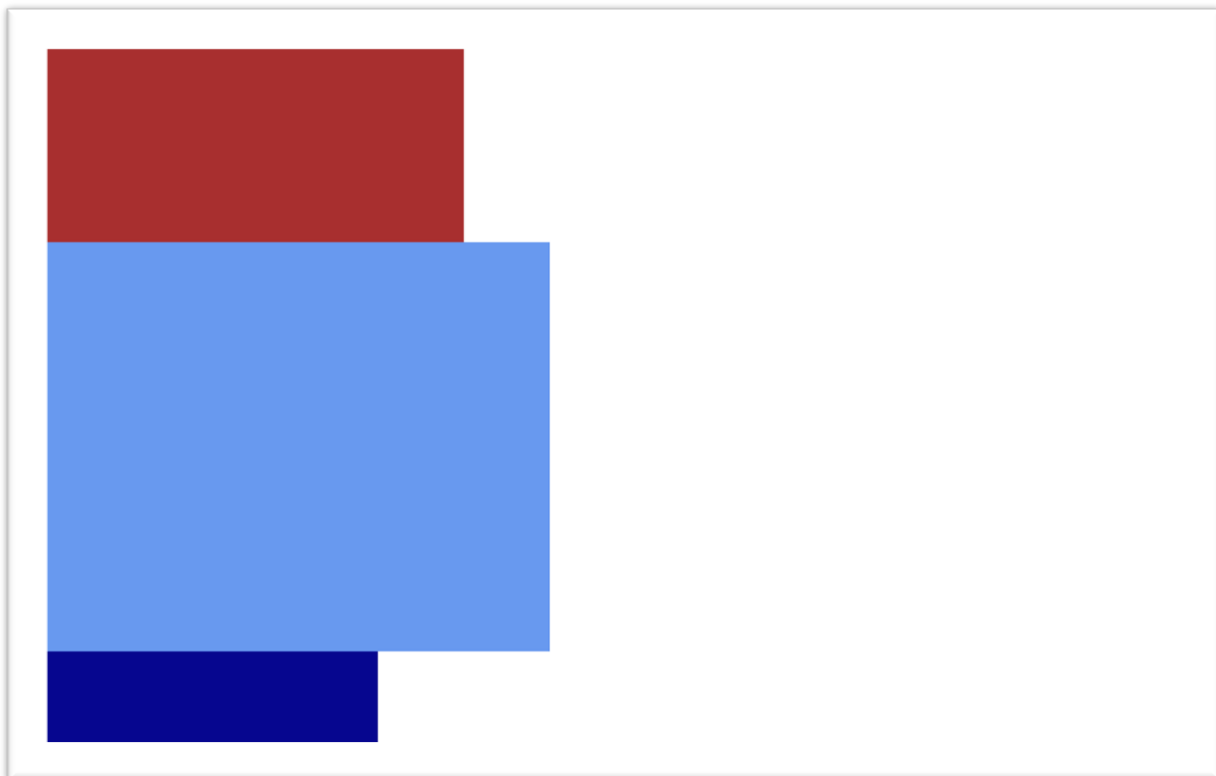
Dans notre fichier css on va rédiger ce code :

```
header{
  width: 500px;
  height: 200px;
  background-color: brown;
}

main{
  width:600px;
  height: 400px;
  background-color: cornflowerblue;
}

footer{
  width: 400px;
  height: 100px;
  background-color: darkblue;
}
```

Si vous ouvrez votre page html dans le navigateur vous obtenez le résultat suivant :



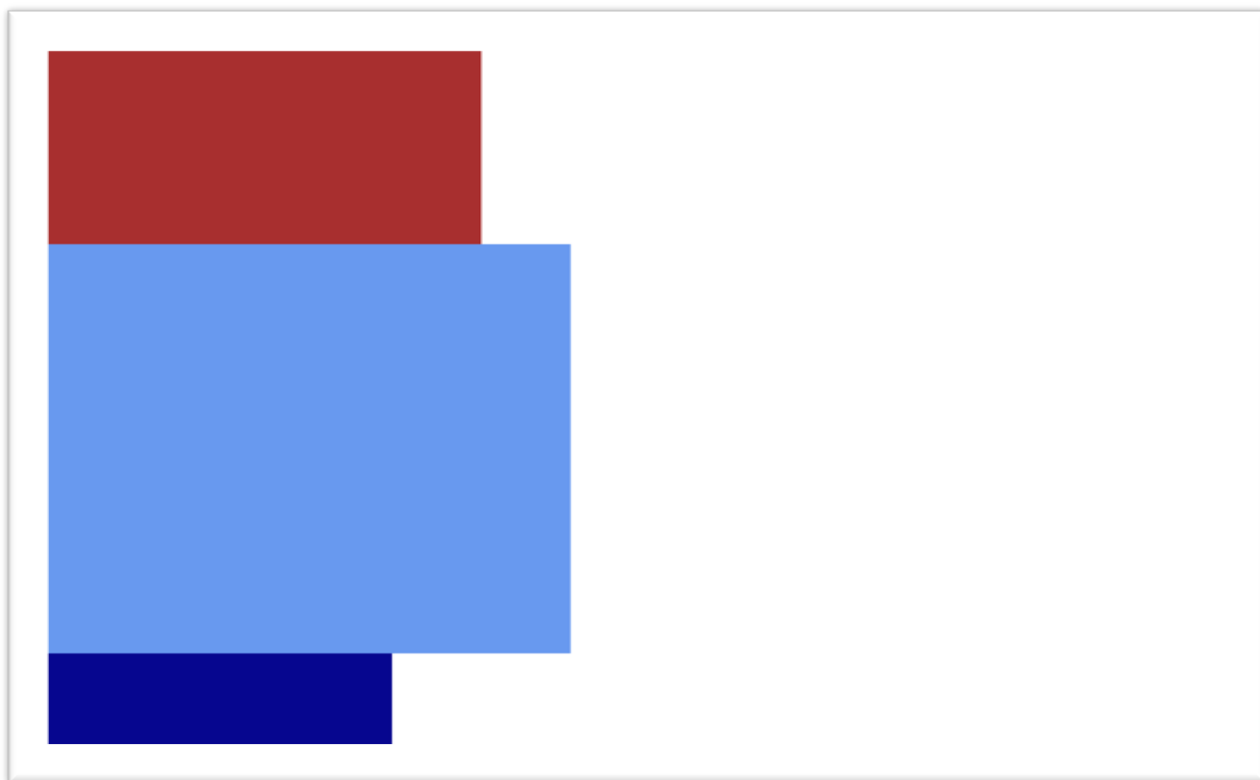
Vous pouvez vous amuser à changer les valeurs de width et de height pour voir la différence.

PS : le width ou height sont respectivement la largeur et la hauteur d'un élément par rapport à toute la page web.

On va essayer de rajouter des éléments dans « main » :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>Document</title>
</head>
<body>
  <header></header>
  <main>
    <div id="hi"></div>
    <div id="hello"></div>
  </main>
  <footer></footer>
</body>
</html>
```

Si vous actualisez votre page web ; vous allez avoir le même résultat :

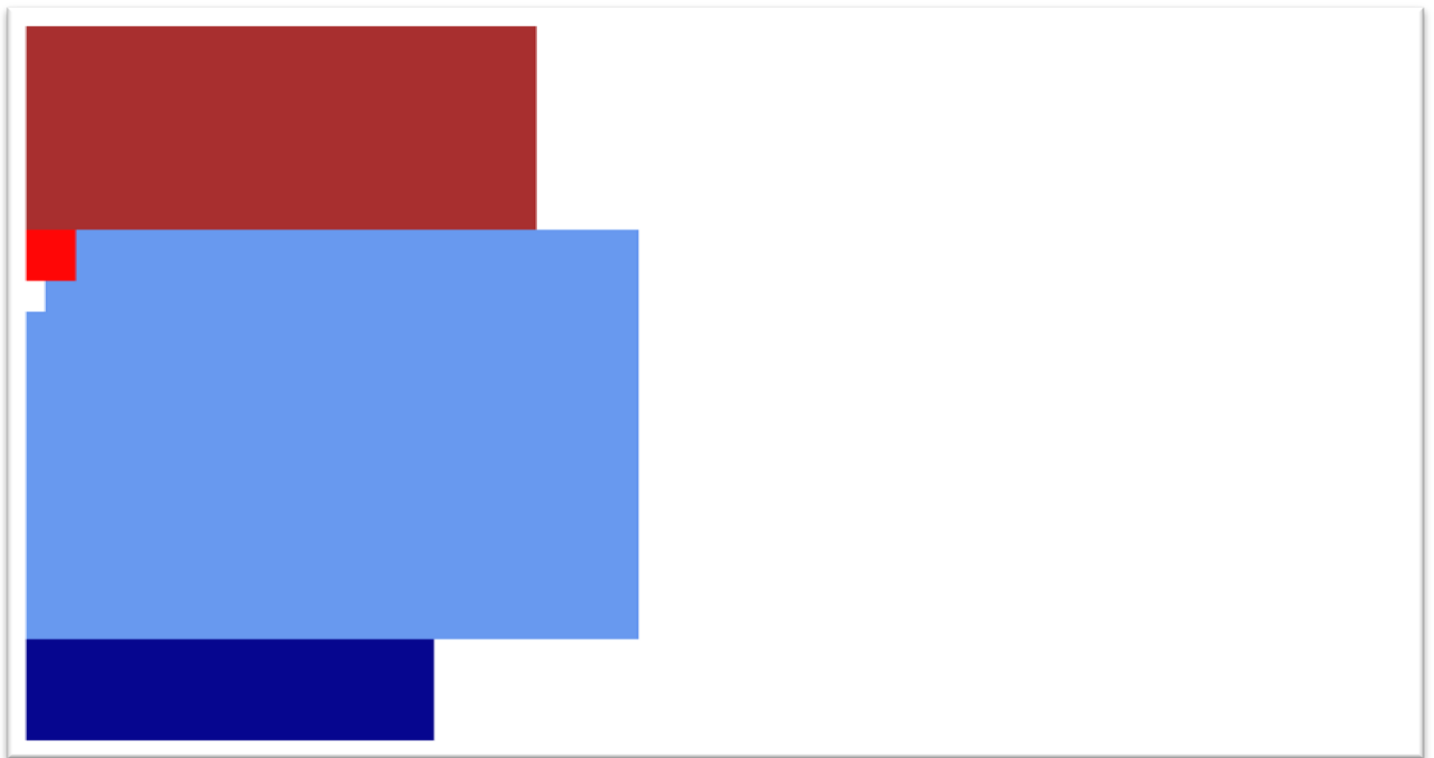


Mais si on rajoute ces lignes à notre CSS :

```
#hi{
  width: 50px;
  height:50px;
  background-color:red;
```

```
}  
  
#hello{  
  width:20px;  
  height:30px;  
  background-color: white;  
  
}
```

On aura ce résultat :



Vous voyez la différence ; nos DIV sont maintenant visibles (c'est le background-color qui nous permet de visualiser les div, vous pouvez essayer d'enlever le background-color pour voir la différence).

En pratique, on ne va pas utiliser height pour main et footer ; on pourra l'utiliser pour header pour définir la fin de ce dernier ou pour DIV en fonction de notre maquette.

Le but de cet exemple est de vous permettre de visualiser ce qui se passe dans votre navigateur quand vous déclarez une balise et que vous puissiez comprendre qu'une balise HTML est l'équivalent d'un marqueur qui définira l'emplacement d'un élément dans une page HTML ; comme vous voyez, les DIV déclarées dans le « main » se sont placées systématiquement dans le main ;)

Comme vous l'avez constaté j'imagine, les balises déclarées en HTML se sont placées l'une après l'autre, pas l'une à côté de l'autre. Par magie ? Non ! Certaines balises HTML ont ce pouvoir qui s'appelle **block** ; c'est-à-dire elles ne laissent aucun autre élément se placer à côté d'elles.

Regardez la première balise DIV, on peut imaginer qu'il reste de la place à côté de cette dernière et que la deuxième pourra se placer à côté d'elle et ben non, la balise DIV est une balise BLOCK elle n'accepte aucun élément à ses côtés;-)

Exemple de balise block : DIV, article, section, p ... à vous de découvrir le reste ;-)

Mais en pratique comment faire pour les mettre l'une à côté de l'autre ? Grâce à la propriété « display » qui nous sauvera la vie 😊

```
header{
  width: 500px;
  height: 200px;
  background-color: brown;
}

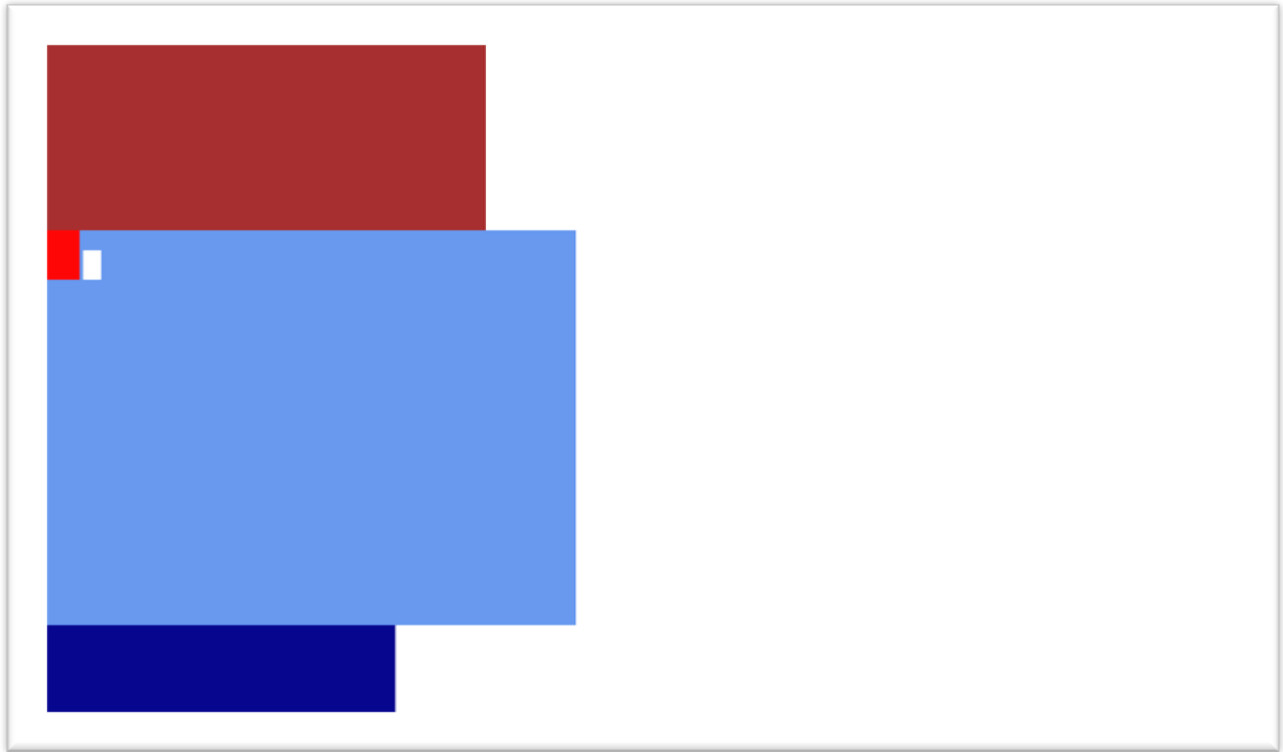
main{
  width:600px;
  height: 400px;
  background-color: cornflowerblue;
}

#hi{
  width: 50px;
  height:50px;
  background-color:red;
  display:inline-block;
}

#hello{
  width:20px;
  height:30px;
  background-color: white;
  display: inline-block;
}

footer{
  width: 400px;
  height: 100px;
  background-color: darkblue;
}
```

Comme vous le voyez, je rajoute cette line css : « display : inline-block »; qui veut dire que la div pourra accepter d'autres éléments à ses côtés, on aura donc ce résultat :



Comme vous le constatez les deux div sont placées l'une à côté de l'autre ;-)

Display a d'autres options : flex, block, none ..etc ; je vous invite à les découvrir !

On arrive à la fin de ce résumé, j'espère que c'était clair pour vous, n'hésitez pas à me poser des questions et je vous invite à me proposer un thème pour le prochain résumé, j'attends vos suggestions ;-)

Amicalement et happy coding day ☺

## Nada

---

30/04/2020