



ADRAR / **PÔLE NUMERIQUE**

- > **INFRASTRUCTURES** SYSTÈMES & RÉSEAUX
- > **CYBERSÉCURITÉ** INFRASTRUCTURES & APPLICATIONS
- > **DEVOPS / SCRIPTING** & AUTOMATISATION
- > **DEVELOPPEMENT** WEB & MOBILE
- > **TRANSFORMATION NUMERIQUE DES ENTREPRISES**

www.adrar-numerique.com

De quoi est composée une page web sur notre navigateur ?



- **HTML** => *HyperText Markup Language*
 - Structure les pages à l'aide de balise, c'est le squelette d'une page.
 - Est interprété par le navigateur



- **CSS** => *Cascading Style Sheets*
 - Donne l'apparence d'une page HTML (positionnement, couleurs, taille des textes, ...)
 - Le CSS n'existe pas seul, il a besoin du HTML



- **JavaScript** => Langage de script léger orienté objet
 - Permet de donner du dynamisme et de l'intelligence à une page web

Avec quoi faire des pages Web ?

Avec un simple bloc note ! Aucun logiciel spécialisé n'est requis

C'est quoi un Editeur de texte alors ?

C'est un logiciel qui va apporter des outils pour faciliter la rédaction de notre code, comme la colorisation syntaxique ou l'autocomplétion.

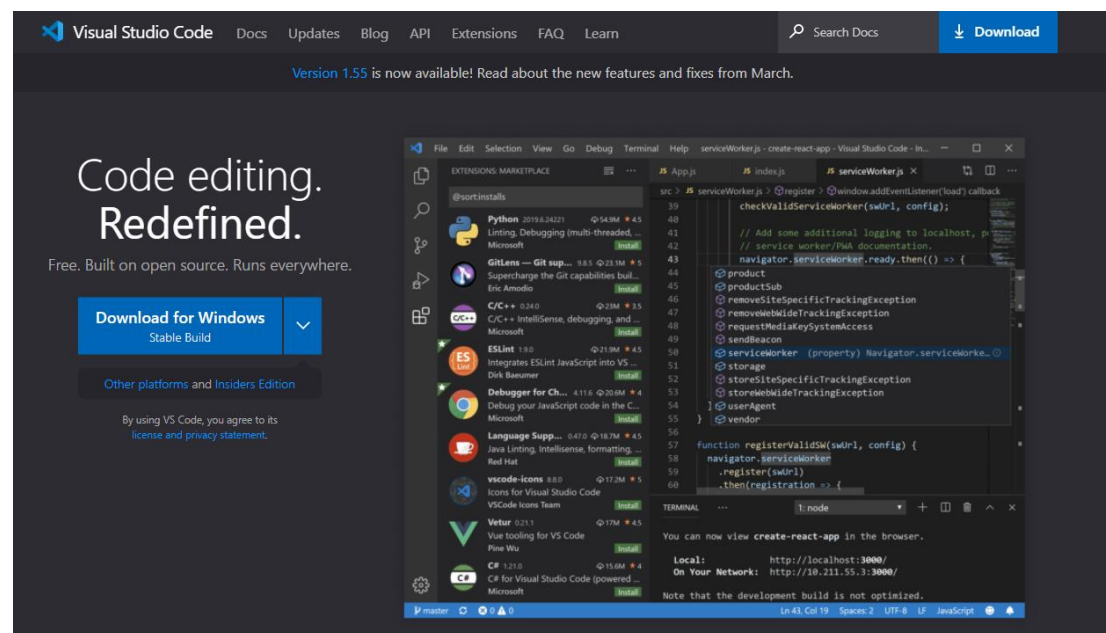
Exemple d'éditeur de texte : Notepad++, Sublime Text, Brackets, Visual Studio Code, et beaucoup d'autres !

Introduction au CSS: Visual Studio Code

Lien de téléchargement : <https://code.visualstudio.com/>

Pro tip !

Allez **TOUJOURS** sur les sites officiels pour télécharger vos Logiciels !



Introduction au CSS: Navigateurs

Lien de téléchargement Chrome : https://www.google.com/intl/fr_fr/chrome/

Lien de téléchargement FireFox : <https://www.mozilla.org/fr/firefox/new/>

Différence entre les navigateurs :

<https://caniuse.com/>

Pro tip !

Il est important d'avoir plusieurs navigateurs sur son ordinateur afin de TESTER toujours énormément son travail.

Le CSS est un langage interprété par le navigateur qui sert à mettre en forme une page web.

Le navigateur va « lire » les instructions et appliquer notre « style », une esthétique, à notre page.

Avec le CSS, nous pouvons agir par exemple sur :

- Les couleurs
- Les tailles
- Le placement des éléments sur la page
- Les polices de caractères (font)
- Etc...

Introduction au CSS: Comparatif

Que peut-on faire concrètement avec le CSS ?

CSS Zen Garden

The Beauty of CSS Design

A demonstration of what can be accomplished through CSS-based design. Select any style sheet from the list to load it into this page.

Download the example [html file](#) and [css file](#)

The Road to Enlightenment

Littering a dark and dreary road lay the past relics of browser-specific tags, incompatible DOMs, broken CSS support, and abandoned browsers.

We must clear the mind of the past. Web enlightenment has been achieved thanks to the tireless efforts of folk like the W3C, WaSP, and the major browser creators.

The CSS Zen Garden invites you to relax and meditate on the important lessons of the masters. Begin to see with clarity. Learn to use the time-honored techniques in new and invigorating fashion. Become one with the web.

So What is This About?

There is a continuing need to show the power of CSS. The Zen Garden aims to excite, inspire, and encourage participation. To begin, view some of the existing designs in the list. Clicking on any one will load the style sheet into this very page. The HTML remains the same, the only thing that has changed is the external CSS file. Yes, really.

CSS allows complete and total control over the style of a hypertext document. The only way this can be illustrated in a way that gets people excited is by demonstrating what it can truly be, once the reins are placed in the hands of those able to create beauty from structure. Designers and coders alike have contributed to the beauty of the web; we can always push it further.

Participation

Strong visual design has always been our focus. You are modifying this page, so strong CSS skills are necessary too, but the example files are commented well enough that even CSS novices can use them as starting points. Please see the [CSS Resource Guide](#) for advanced tutorials and tips on working with CSS.

You may modify the style sheet in any way you wish, but not the HTML. This may seem daunting at first if you've never worked this way before, but follow the listed links to learn more, and use the sample files as a guide.

Download the sample [HTML](#) and [CSS](#) to work on a copy locally. Once you have completed your masterpiece (and please, don't submit half-finished work) upload your CSS file to a web server under your control. [Send us a link](#) to an archive of that file and all associated assets, and if we choose to use it we will download it and place it on our server.

Benefits

Why participate? For recognition, inspiration, and a resource we can all refer to showing people how amazing CSS really can be. This site serves as equal parts inspiration for those working on the web today, learning tool for those who will be tomorrow, and gallery of future techniques we can all look forward to.

Requirements

Where possible, we would like to see mostly CSS 1 & 2 usage. CSS 3 & 4 should be limited to widely-supported elements only, or strong fallbacks should be provided. The CSS Zen Garden is about functional, practical CSS and not the latest bleeding-edge tricks viewable by 2% of the browsing public. The only real requirement we have is that your CSS validates.

Luckily, designing this way shows how well various browsers have implemented CSS by now. When sticking to the guidelines you should see fairly consistent results across most modern browsers. Due to the sheer number of user agents on the web these days — especially when you factor in mobile — pixel-perfect layouts may not be possible across every platform. That's okay, but do test in as many as you can. Your design should work in at least IE9+ and the latest Chrome, Firefox, iOS and Android browsers (run by over 90% of the population).

We ask that you submit original artwork. Please respect copyright laws. Please keep objectionable material to a minimum, and try to incorporate unique and interesting visual themes to your work. We're well past the point of needing another garden-related design.

Source: [CSS Zen Garden](#)



Introduction au CSS: Bonnes pratiques

Avant toute chose, où place t'on notre code CSS ?

Dans une balise directement grâce à l'attribut « style » :

```
<p style="color: red;">
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Mauris commodo lacus placerat, ultrices quam vitae, luctus odio.
</p>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Mauris commodo lacus placerat, ultrices quam vitae, luctus odio.

Dans le <head> grâce à une balise <style> :

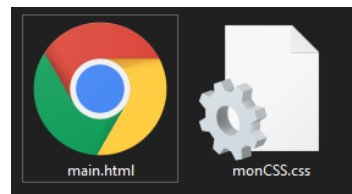
```
<head>
  <style>
    p{
      color: blue;
    }
  </style>
</head>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Mauris commodo lacus placerat, ultrices quam vitae, luctus odio.

Dans un fichier « .css » à part:

```
<head>
  <link rel="stylesheet" href="monCSS.css" >
</head>
```

```
p{
  color: magenta;
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 Mauris commodo lacus placerat, ultrices quam vitae, luctus odio.

Que choisir ?

Dans un fichier à part ! Car cela a de nombreux avantages :

- + Permet une maintenance du code plus simple
- + Permet d'utiliser le même fichier sur plusieurs pages HTML

Comment ça marche ?

Il suffit de lier notre page CSS avec notre page HTML grâce à la balise <link> dans le <head> :

```
<head>  
    <link rel="stylesheet" href="monCSS.css" >  
</head>
```

Introduction au CSS: En détail

Regardons notre code en détail :

```
p{  
    color: magenta;  
}
```

Nous avons :

- **En rouge** : Le sélecteur => Permet de choisir sur qui s'applique le style
- **En vert** : Les accolades => Délimite les propriétés pour un élément
- **En bleu** : La propriété => Les effets graphiques qui vont être appliqués
- **En jaune** : La valeur => La valeur (un nombre, une couleur, ...)
- **En rose** : Le séparateur => Le point virgule « ; » sert à séparer les propriétés entre elles

Ici, nous appliquons donc la couleur « magenta » au texte de nos balises <p>.

Comme pour le HTML, nous pouvons mettre des commentaires dans notre code qui ne seront pas interprétés par le navigateur, et qui n'apparaîtront pas sur la page.

Attention, tout ce que vous écrivez en commentaire pourra aussi être lu par n'importe qui !

Pour écrire un commentaire, il faut le mettre entre « /* » ouvrante et « */ » fermante comme ceci :

```
p{  
    /*J'applique la couleur magenta*/  
    color: magenta;  
}
```

Mais pourquoi « cascade » dans **C**ascading **S**tyle **S**heets ?

- Les feuilles de style sont exécutées dans l'ordre de déclaration :

```
<head>
    <link rel="stylesheet" href="monCSS.css">
    <link rel="stylesheet" href="monCSS2.css">
</head>
/* Les propriétés de monCSS2 écraseront celle de monCSS si elles sont
déclarées */
```

- Les déclarations sont aussi exécutées dans l'ordre d'écriture :

```
p{
    color: blue;
}

p {
    color: magenta;
}
/* La couleur du paragraphe sera donc magenta */
```

Mais pourquoi « cascade » dans **C**ascading **S**tyle **S**heets ?

- L'instruction la plus précise sera celle qui sera prioritaire :

```
<p id="monId">
Lorem ipsum dolor sit amet, consectetur adi
piscing elit.
</p>
```

```
#monId{
    color: red;
}
p{
    color: blue;
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

- Certaines propriétés héritent du parent vers l'enfant :

```
div{
    color: magenta;
}
```

```
<div>
    <p>
    Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    </p>
</div>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Introduction au CSS: Vocabulaire & Résumé

- On déclare de préférence notre CSS dans un fichier séparé que l'on lie à notre page grâce à la balise <link> qui se situe dans le <head> de la page.
- Notre fichier CSS contient des règles CSS.
- Chaque règle CSS contient une ou plusieurs déclarations qui sont composées d'une propriété et de sa valeur.
- On utilise le « : » pour séparer la propriété de sa valeur.
- On utilise le « ; » pour séparer les déclarations entre elles.
- On utilise les accolades « { } » pour séparer les règles entre elles.

```
h1{  
  color: blue;  
  font-size: large;  
}  
/*Je suis un commentaire*/  
p {  
  color: magenta;  
  background-color: black;  
}
```