Introduction aux Technologies du Web

HTML + CSS +(Javascript)

Rachid EDJEKOUANE (edjek@hotmail.fr)

Prérequis

- Savoir utiliser un ordinateur
- Explorateur de fichiers
- Savoir créer des dossiers
- Se situer dans une arborescence de fichiers
- Ponctualité

Objectifs : comprendre le web

- Comprendre comment fonctionne le web
- Comprendre l’architecture d’un site web
- A quoi sert le HTML, la syntaxe, la sémantique, la structure d’une page, les liens…
- Apprendre à styliser avec du CSS (sélecteur { propriété : valeur; })
- Connaître les principales propriétés CSS
- Comprendre le positionnement des blocs HTML en CSS

Objectifs : comprendre le web

Introduction aux technologies web:

- HTML pour créer la structure du document et le contenu
- CSS pour contrôler l’aspect visuel
- Javascript pour l'interactivité

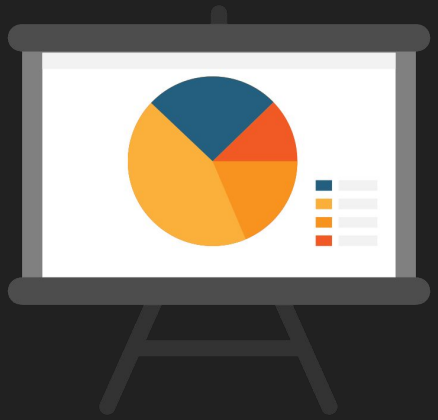
Pour créer une page Web

Ce dont vous avez besoin pour débuter :

- Un bon navigateur (Firefox ou Chrome)
- Un éditeur de texte : Visual Studio Code (cross platform), (Notepad++), ou Sublime Text (cross platform)...

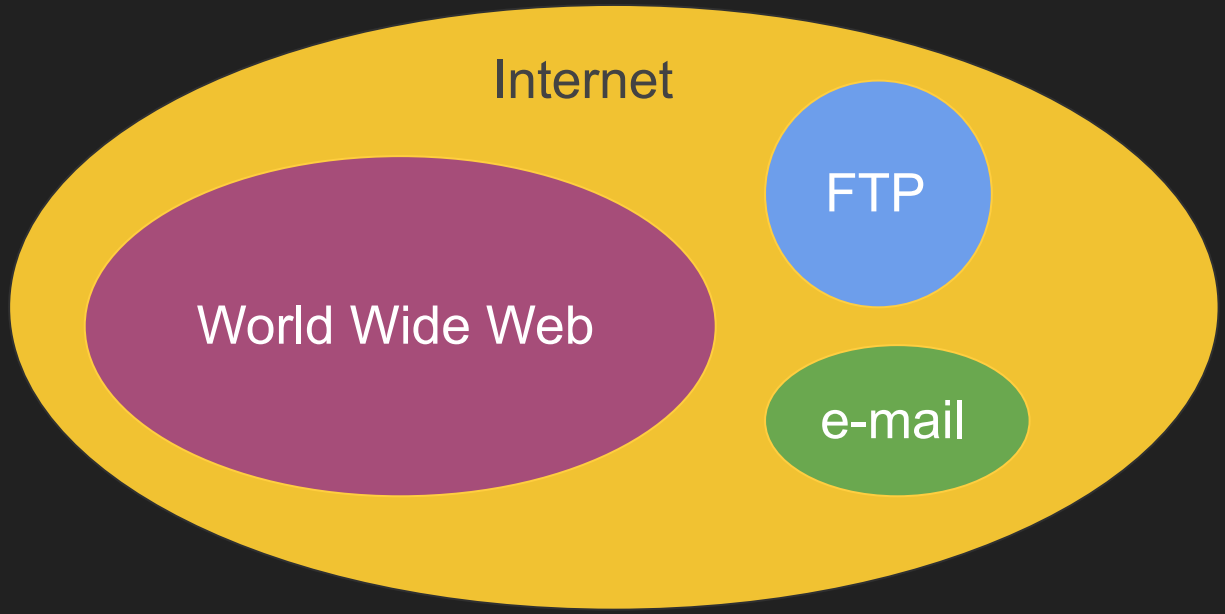



Introduction



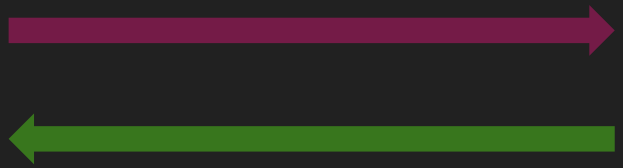
World Wide Web

Le web est une des applications d'internet



Model Client Server

Request



Response

Création du Web

En créant le Web (1989), Tim Berners-Lee invente ses trois technologies fondatrices :

- Le protocole de communication HTTP
- Les adresses Web sous forme d'URL
- Le langage informatique HTML.

W3C : World Wide Web Consortium (1994)

- Tim Berners-Lee, fondateur du W3C et inventeur du HTML
- Chargé de promouvoir la compatibilité des technologies entre les navigateurs
- 378 entreprises membres qui peuvent faire des propositions (Microsoft, Apple, Mozilla, Opera, Adobe, etc.)
- Propose un validateur http://validator.w3.org/

Un page Web c’est :

- Un fichier HTML est un format de fichier « texte » éditable dont les éléments ont du sens
- Au format .html
- Qui peut contenir du texte, des liens, des images, des médias, etc…
- Qui peut être liée à une autre page via des liens

Un site Web c’est ...

- Un ensemble de pages liées entre elles via des liens
- Accessible en ligne depuis n'importe où

SEO, qu’est-ce que c’est ?

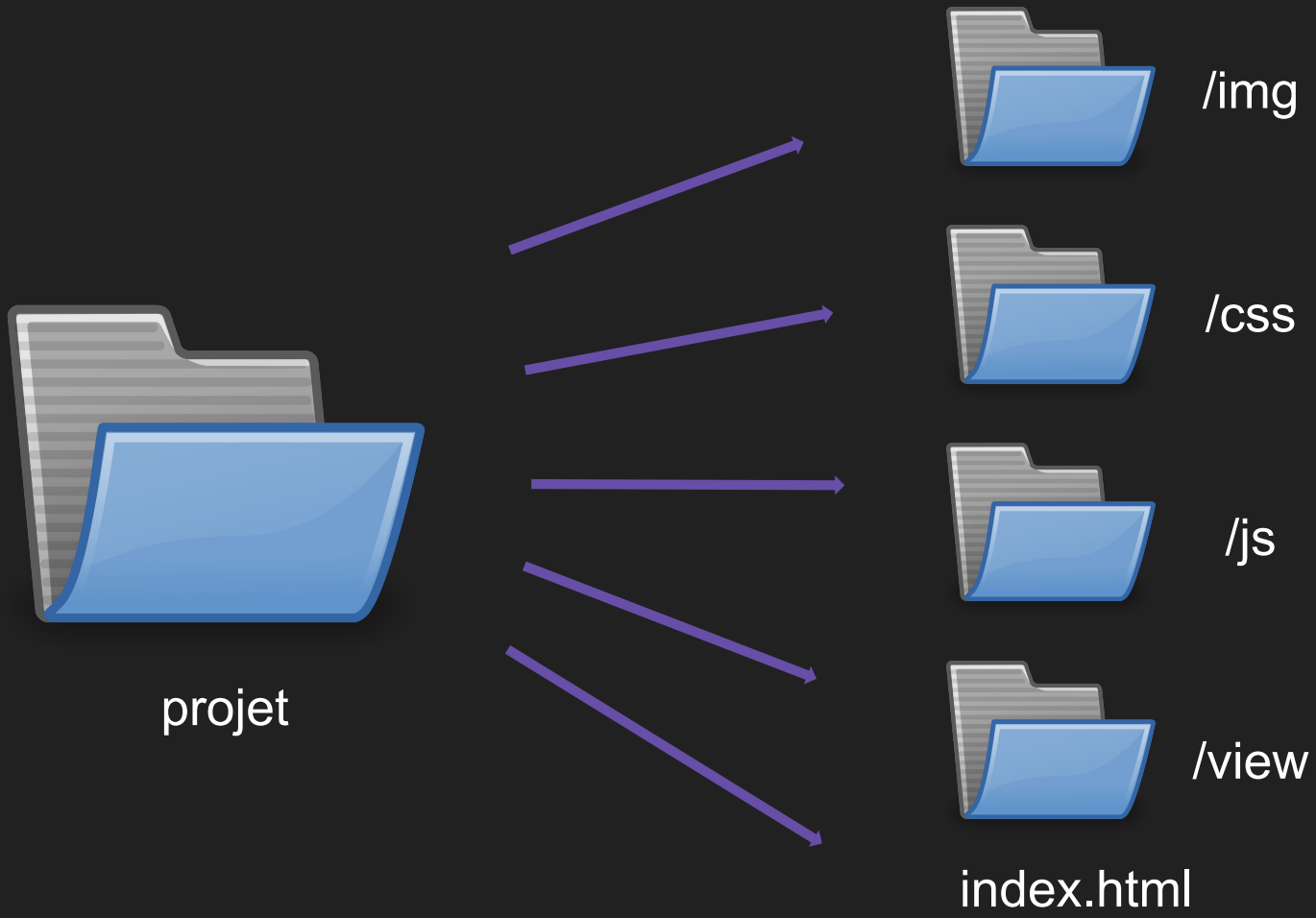
SEO (Search Engine Optimisation)

SEO en français : Optimisation pour les moteurs de recherche.

Ce terme définit l'ensemble des techniques mises en œuvre pour améliorer la position d'un site web sur les pages de résultats des moteurs de recherche. On l'appelle aussi référencement naturel.

On dit qu'un site est bien optimisé ou référencé s'il se trouve dans les premières positions d'un moteur de recherche sur les requêtes souhaitées.

Organisation


/img
/css
/js
/view
projet
index.html

Règles d’or

Tout fichier doit être enregistrer avec l’extension .html et non en .txt

Respecter une charte de nommage : pas de majuscule, pas d’espace, pas d’accent ou caractères spéciaux, séparation avec - dans les noms des fichiers et dossiers html, css, photo…

Chaque mot html doit être écrit en minuscule

Une page html commence par <html> et comporte deux parties : <head> et <body>

Pour ne pas que les mots html apparaissent comme du texte et s'affichent à l'écran je vais les enfermer entre CHEVRONS (< et >) pour être interpréter comme étant bien du html

Dans la partie <head>, chaque page web, doit avoir un titre via la balise <title></title>

Toute la partie destinée à apparaître à l'écran doit être comprise dans la partie <body>

Pour les problèmes d'accent, toujours préciser l'encodage avec la norme UTF-8

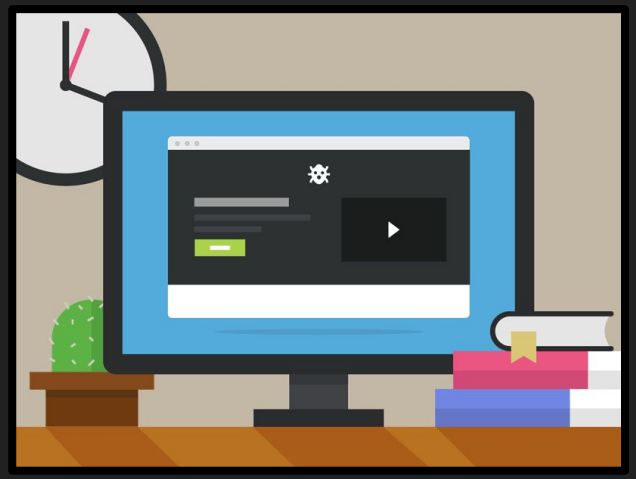
On va utiliser les balises : <meta> dans la partie <head>

A chaque fois que je place une balise html on duplique le même mot, précédé d'un slash

Certains mots HTML n'ont pas besoin d'être fermé, on parle de balises orphelines <hr>,
, <meta />

Technologies

- HTML
- CSS
- Javascript



HTML : HyperText Markup language

L’HTML nous permet de définir la structure du page d’un site web.

Il s'agit d'une série de balises imbriquées qui contiennent toutes les informations du site Web (comme les textes, les images et les vidéos).

Le HTML définit la structure de la page.
Un site Web peut avoir plusieurs liens HTML vers différentes pages.

HTML n’est PAS un langage de programmation, c'est un langage de balisage, ce qui signifie que son but est de structurer le contenu du site Web.

HTML : le minimum syndical


```
<!DOCTYPE html>
<html lang="fr">
    <head>
        <meta charset="utf-8">
        <title>Titre</title>
    </head>
    <body>

    </body>
</html>
```

HTML : règles de base

Il utilise la syntaxe XML (balises avec attributs, qui peut contenir d'autres balises).

```
<h1 attribut="valeur">contenu</h1>
```

Il stocke toutes les informations qui doivent être présentées à l'utilisateur.
Il existe différents éléments HTML pour différents types d'informations et de comportements.
Il donne au document une structure sémantique (par exemple, ceci est un titre, ceci est une section, ceci est un formulaire) qui est utile pour que les ordinateurs comprennent le contenu des sites Web.

Il ne doit pas contenir d'informations relatives à la façon dont il doit être affiché (ces informations appartiennent au CSS), donc aucune information sur la couleur, la taille de la police, la position, etc…

HTML : exemple de syntaxe


```
<div class="main">
    <!-- ceci est un commentaire -->

    <button class="btn">press me</button>
    <img src="picture.png" />
</div>
```

HTML : exemple de syntaxe

nom de balise

attributs

commentaire

```
<div class="main">
    <!-- ceci est un commentaire -->

    <button class="btn">press me</button>
    <img src="picture.png" />
</div>
```

balise auto-fermante

HTML : Les principaux éléments

5 éléments de type TEXTE

2 éléments de type MEDIAS

- LES TITRES
- LES PARAGRAPHES
- LES LISTES
- LES TABLEAUX
- LES FORMULAIRES
- LES LIENS
- LES IMAGES
- LES VIDEOS

HTML : les titres

```
<body>
    <h1>mon titre 1</h1>
    <h2>mon titre 2</h2>
    <h3>mon titre 3</h3>
    <h4>mon titre 4</h4>
    <h5>mon titre 5</h5>
    <h6>mon titre 6</h6>
</body>
```

mon titre 1

mon titre 2

mon titre 3

mon titre 4

mon titre 5

mon titre 6

HTML : les paragraphes

```
<p>
    Lorem ipsum dolor sit amet,
    consectetur adipiscing elit.
    Quisque faucibus eget lacus at
    ultricies. Phasellus augue neque,
    <br>
    laoreet vitae odio in, molestie
    interdum sem.
</p>
```

<p>

HTML : les listes non-ordonnées

```
<p>
Voici mes films préférés :
</p>
<ul>
    <li>Ben hur</li>
    <li>Fight club</li>
    <li>Batman</li>
</ul>
```

Voici mes films préférés :

- Ben hur
- Fight club
- Batman

HTML : les listes ordonnées

```
<p>
Voici mes films préférés :
</p>
<ol>
    <li>Ben hur</li>
    <li>Fight club</li>
    <li>Batman</li>
</ol>
```

Voici mes films préférés :

1. Ben hur
2. Fight club
3. Batman

HTML : les liens cliquables

```
<a href="index.html" target="_blank" title="accueil">
    Accueil
</a>
```

Attention :

l'attribut `target="_blank"` est facultatif, il permet de faire ouvrir la page dans un nouvel onglet

L'attribut title est facultatif, il permet de faire apparaître une info bulle.

HTML : les images

```
<img src="image.jpg" alt="description" title="hello">
```

Attention :

l’attribut alt est obligatoire !

HTML : les tableaux

```
<table>
    <tr>
        <td>Ben hur</td>
        <td>Fight club</td>
        <td>Batman</td>
    </tr>
    <tr>
        <td>Ben hur</td>
        <td>Fight club</td>
        <td>Batman</td>
    </tr>
</table>
```

Les attributs pour les tableaux :

rowspan : permet de fusionner des lignes

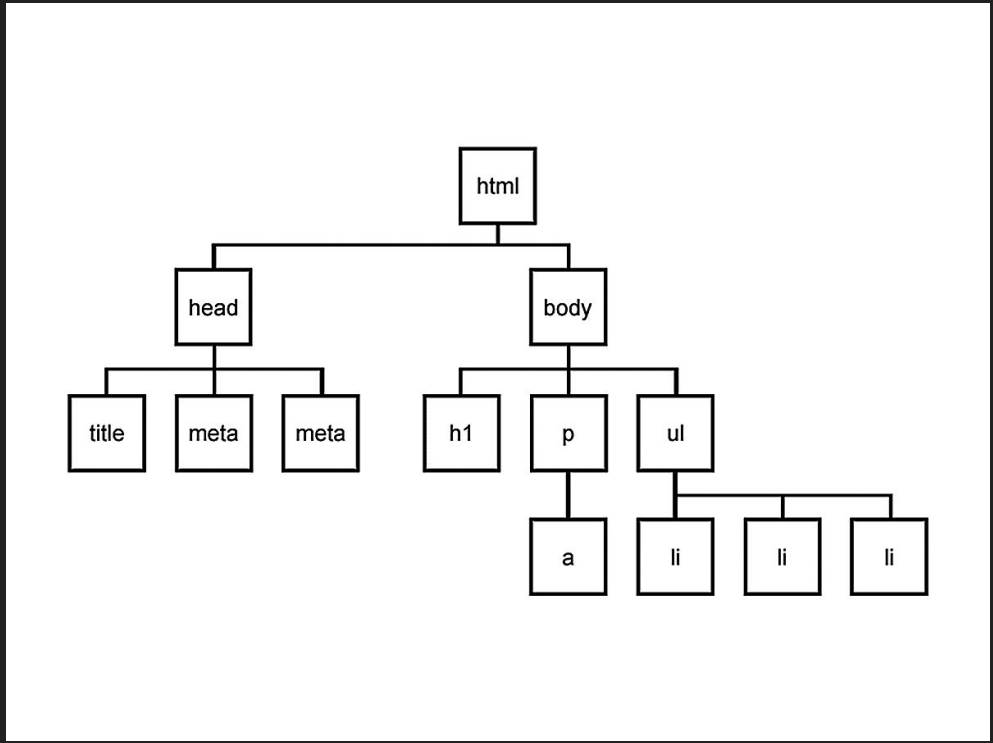
colspan : permet de fusionner des colonnes

HTML : les formulaires


```html
<form action="">
    <label for="name">Nom</label>
    <input type="text" id="name" required autofocus>

    <label for="pwd">Mot de passe</label>
    <input type="password" id="pwd" name="pwd">

    <input type="submit" value="Envoyer">
</form>
```

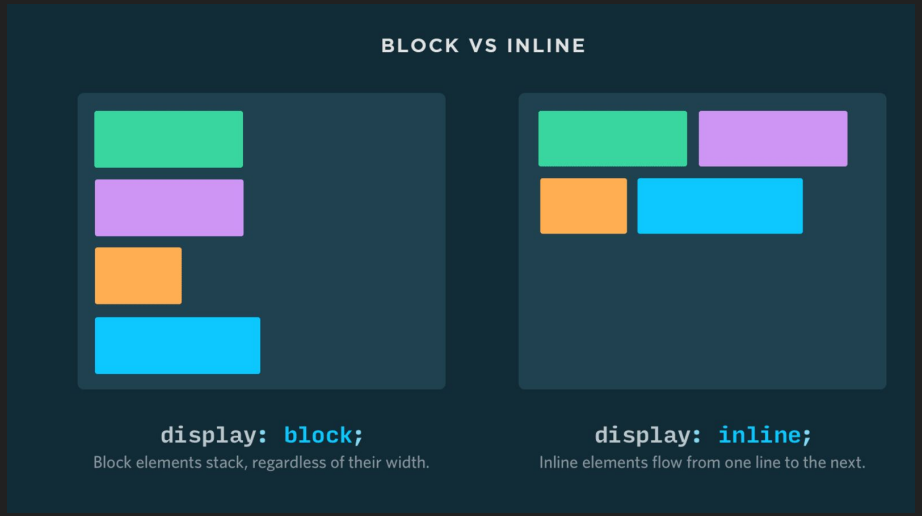
HTML : un arbre

Chaque nœud ne peut avoir qu'un seul parent et chaque nœud peut avoir plusieurs enfants, de sorte que la structure ressemble à un arbre.



HTML : les 2 grands types d'éléments

- Block
- Inline



HTML : les d'éléments inline


```
<span></span><em></em><strong></strong><a></a><img/>
```

Ils se mettent les uns à côté des autres.

Ils ne peuvent PAS accepter les propriétés CSS suivantes :

- largeur (width)
- hauteur (height)
- marge au-dessus et en-dessus (margin-top, margin-bottom)

HTML : les éléments block

```
<div></div>
<p></p>
<ul>
    <li></li>
    <li></li>
</ul>
```

Ils se mettent les uns en dessous des autres.

Ils peuvent accepter les trois propriétés CSS suivantes en même temps :

- largeur (width)
- hauteur (height)
- marge verticale(margin-top, margin-bottom)
- couleur de fond (background-color)

HTML : baliser correctement

Eviter de faire ceci :

```
<div>
Titre
C’est du contenu
Encore du contenu.
</div>
```

NE FAITES PAS CA!

Faites cela à la place

```
<div>
    <h1>Titre</h1>
    <p>C’est du contenu.</p>
    <p>Encore du contenu.</p>
</div>
```

HTML : layout HTML5

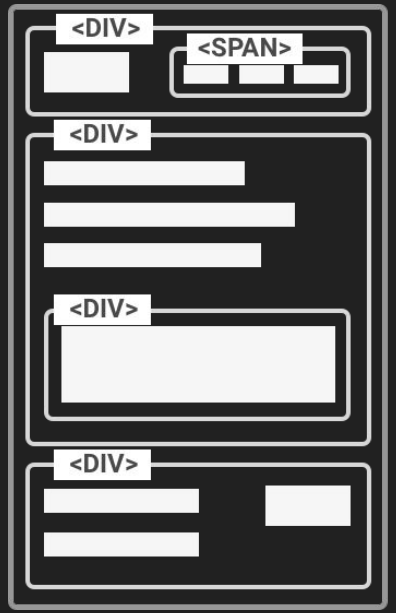
```
<body>
    <header>
        <nav></nav>
    </header>

    <main>
        <section></section>
        <section></section>
    </main>

    <footer></footer>
</body>
```

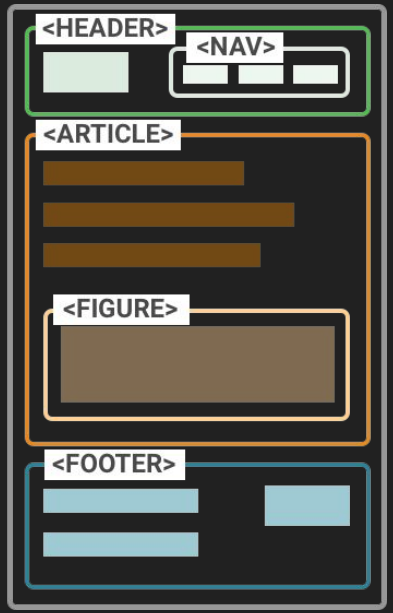
HTML : envelopper les informations

Nous utilisons des balises HTML pour envelopper différentes informations sur notre site.

Plus l'information est structurée, plus il sera facile d'y accéder et de la présenter.

Nous pouvons changer la façon dont les informations sont représentées à l'écran en fonction des balises où elles sont contenues, nous ne devrions donc pas nous inquiéter d'utiliser trop de balises.





HTML : les principales balises

Bien qu'il existe de nombreuses balises dans la spécification HTML, 99 % des sites Web utilisent un sous-ensemble de balises HTML avec moins de 10 balises, les plus importantes étant :

- <h1> : un titre (h2,h3,h4, h5, h6 sont des titres de moindre importance)
- <p> : un paragraphe de texte
- : une balise inline neutre
- <img/> : une image
- <a> : un lien cliquable pour aller vers une autre URL
- : un conteneur neutre, représente généralement une zone rectangulaire avec des informations à l'intérieur
- <input> : un widget permettant à l'utilisateur d'introduire des informations
- <style> : pour insérer des règles CSS
- <script> : pour exécuter Javascript

HTML : bonnes pratiques

Il est bon d'avoir toutes les informations correctement enveloppées dans des balises qui leur donnent une certaine sémantique.

Nous pouvons également étendre la sémantique du code en ajoutant des attributs supplémentaires aux balises :

- class: identifiant générique de balise
- id: identifiant unique de balise

```
<div id="profile-picture" class="hero-section">...</div>
```

HTML : références

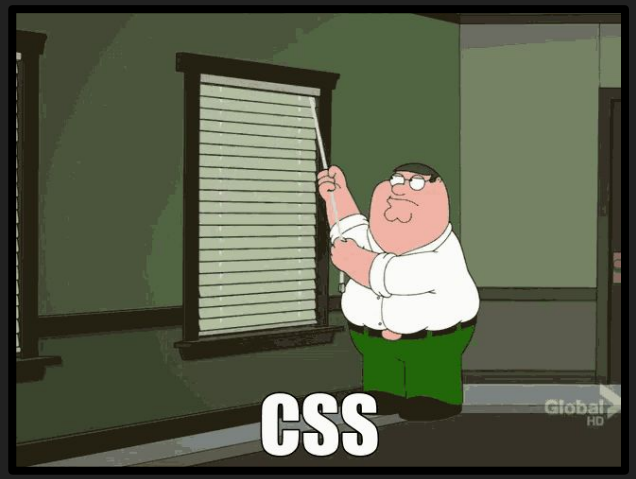
HTML reference : description de toutes les balises HTML.

The 25 Most used tags : une liste avec des informations sur les balises les plus courantes.

HTML5 Bonnes pratiques : quelques conseils pour les débutants

Technologies

- HTML
- CSS
- Javascript



CSS

CSS nous permet de spécifier comment mettre en forme (styliser) les informations stockées dans le HTML.

Grâce au CSS, nous pouvons contrôler tous les aspects de la visualisation et quelques autres fonctionnalités :

- color : couleur du texte
- width, height : largeur, hauteur
- margin : marge extérieure
- padding : marge intérieure
- position : où positionner l'élément
- hover : au passage de la souris

CSS : comment l’ajouter

Il existe quatre façons d'ajouter des règles CSS à votre site Web :

- Référencer un fichier CSS externe (la bonne manière)
  ```
  <link rel="stylesheet" href="style.css" />
  ```
- Insertion du code dans une balise de style dans l’HTML
  ```
  <style>
      p { color: blue; }
  </style>
  ```
- Utiliser le style d'attribut sur une balise
  ```
  <p style="color : blue; margin: 10px;"></p>
  ```
- Utilisation de Javascript

CSS : exemple


```css
p {
    margin: 0; /* un commentaire */
    padding: 0;
    box-sizing: border-box;
}
```

Ceci change tous les paragraphes de ma page en bleu avec la typo Tahoma en 14px, et laisse une marge de 10px autour.

CSS : sélecteur universel


```css
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
```

("*" est un sélecteur universel, il signifie tous les sélecteurs)

CSS : reset


```css
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}
```

Le reset CSS est une technique qui consiste à réinitialiser à 0 la valeur de certains éléments HTML afin d'éviter une partie des différences d'affichage sur les divers navigateurs.

CSS : propriétés

Voici une liste des champs CSS les plus courants :

- `color : #ff0099; | red; | rgb(255,00,100); // façons de spécifier les couleurs`
- `background-color : red;`
- `background-image : url('file.png');`
- `font : 18px 'Tahoma';`
- `border : 2px solid black;`
- `border-radius : 2px; //pour supprimer les coins et les rendre plus ronds`
- `margin : 10px auto; //distance de la bordure aux éléments extérieurs`
- `padding : 20px; //distance de la bordure aux éléments intérieurs`
- `width : 100%; | 300px; | 1.3em;  //de nombreuses façons de spécifier les dimensions`
- `height : 200px;`
- `text-align : center;`
- `box-shadow : 3px 3px 5px black;`
- `cursor : pointer;`
- `display : flex;`
- `overflow : hidden;`

CSS : sélecteurs (le nom de la balise HTML)

Commençons par changer la couleur de fond d'une balise de notre site :

```
div { background-color: red; }
```

Cette règle CSS signifie que chaque balise div va avoir une couleur de fond rouge. N'oubliez pas que les div sont principalement utilisées pour représenter des zones de notre site Web.

Nous pourrions également modifier l'ensemble de l'arrière-plan du site Web en affectant le corps de la balise :

```
body { background-color: red; }
```

CSS : sélecteurs (class)

Et si nous voulons changer une balise spécifique (pas toutes les balises du même type).

Nous pouvons spécifier des sélecteurs plus précis en plus du nom de la balise. Par exemple, par class ou id.

Pour spécifier une balise avec un nom de class donné, nous utilisons le point :

```
.intro { color: red; }
```

Cela n'affectera que les balises avec le nom de classe intro :

```
<p class="intro">
```

CSS : sélecteurs (id)

Nous pouvons spécifier un sélecteur UNIQUE, l'id.
Pour spécifier une balise avec un nom d' id donné, nous utilisons le #:

```
#intro {
    color: red;
}
```

Cela n'affectera que la balise p avec le nom d'id intro :

```
<p id="intro">
```

! On préfère réserver l'utilisation d'id pour le Javascript

CSS : sélecteurs multiples

Si vous souhaitez utiliser les mêmes actions CSS sur plusieurs sélecteurs, vous pouvez utiliser la virgule , caractère :

```
div, p { … }  ← cela s'appliquera à toutes les balises div et p
```

Si vous souhaitez sélectionner uniquement les éléments qui sont des enfants directs d'un élément, utilisez le caractère > :

```
ul.menu > li { ... }
```

CSS : sélecteurs descendant (chemin)

Vous pouvez également spécifier une balise par son contexte, par exemple : des balises qui se trouvent à l'intérieur de balises correspondant à un sélecteur. Séparez simplement les sélecteurs par un espace :

```
.main p { ... }
```

cela affecte toutes les balises p à l'intérieur de la div avec la class main

```
<div class="main">
    <p class="intro">....</p>  ← Affecte celle-ci
</div>

<p class="diff">....</p>  ← pas celle-là
```

CSS : sélecteurs

Et vous pouvez combiner des sélecteurs pour le réduire davantage.

```
#main .intro:hover { ... }
```

appliquera le CSS à n'importe quelle balise ayant la class intro dans une balise avec l'id main si la souris est en survole :hover.

Et si vous avez pas besoin de spécifier une balise, vous pouvez utiliser les sélecteurs de class ou d'id avec balise, cela signifie que cela affecte la balise avec la class main

```
p.main { ... }
```

CSS : sélecteurs (résumé)

Il existe plusieurs sélecteurs que nous pouvons utiliser pour limiter nos règles à des balises très spécifiques de notre site Web :

- nom de la balise : juste le nom de la balise
  - `p { ... }  //affecte toutes les balises <p>`
- (.) : affecte les balises avec cette class
  - `.highlight { ... } //affecte les balises avec la class="highlight"`
- (#) : spécifique à la balise qui a cet id
  - `#intro { ... } //affecte la balise avec l'id="intro"`
- (:) : comportement d'état (souris au survol)
  - `p:hover { ... } //affecte la balise <p> au survole de la souris`
- .main p : chemin ou se trouve l'élément ciblé
  - `.main p {...} // affecte la balise p dans la div avec la class="main"`

CSS : les couleurs

Il existe plusieurs façons de représenter une même couleur :

```
.color {
/* Valeurs avec un mot-clé pour une couleur nommée */
    color: orange;
/* Valeurs hexadécimales <hex-color> */
    color: #ffa500;
/* Valeurs utilisant la fonction <rgb()> */
    color: rgb(255, 165, 0);
/* Valeurs <hsl()> */
    color: hsl(39, 100%, 50%);
}
```

CSS : propriétés pour le texte


```html
<p class="message">hello world!</p>
```

```css
.message {
    color: #123299;
    font-size: 20px;
    font-family: Arial, sans-serif;
    font-style: italic;
    font-weight: bold;
    text-transform: uppercase;
    text-decoration: underline
    line-height: 22px;
}
```

HELLO WORLD!

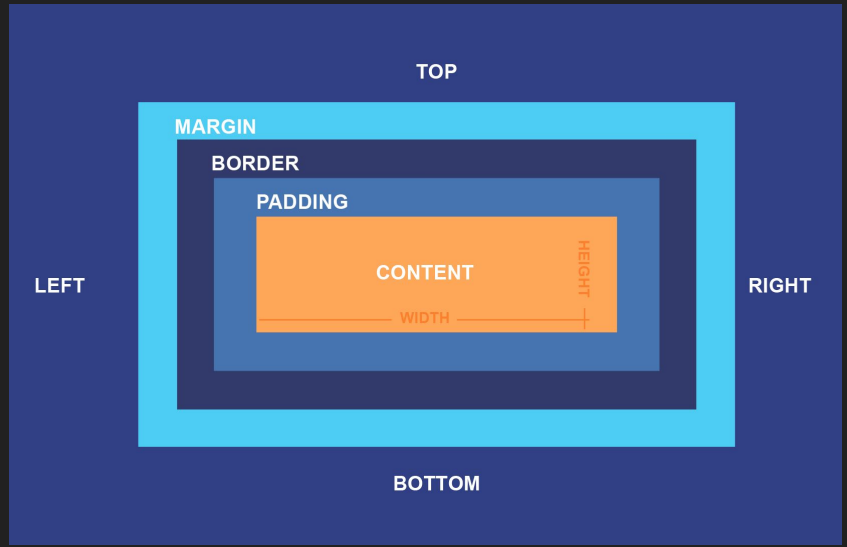
CSS : model de boite

Il est important de noter que par défaut, tout padding spécifiées pour un élément ne prend pas en compte sa marge, donc une div avec une largeur de 100px et un padding de 10px mesurera 120px à l'écran, et non 100px.

Cela pourrait être un problème pour casser votre mise en page.

Vous pouvez modifier ce comportement en modifiant le modèle de boîte de l'élément afin que la largeur utilise la bordure la plus à l'intérieur :

```
div { box-sizing: border-box; }
```



CSS : propriétés pour les block

```html
<div class="container"></div>
```

```css
.container {
    width: 50%;
    height: 200px;
    margin: 50px auto;
    padding: 20px;
    border: 1px solid #e8d718;
    border-radius: 5px;
    background-color: #73c553;
    box-shadow: 4px 4px 8px #999;
}
```

CSS : propriétés raccourcies

Il existe des formes raccourcies pour écrire sur 1 seule ligne plusieurs propriétés CSS.

```css
.container {
    margin-top: 10px;
    margin-right: 20px;
    margin-bottom: 25px;
    margin-left: 10px;
}
```

```css
.container {
    margin: 10px 20px 25px 10px;
}
```

CSS : propriétés raccourcies (border)

Il existe des formes raccourcies pour écrire sur 1 seule ligne plusieurs propriétés CSS.

```css
.container {
    border-color: #000;
    border-width: 1px;
    border-style: solid;
}
```

```css
.container {
    border: 1px solid #000;
}
```

CSS : background-image

Afin d’afficher une image en background nous pouvons spécifier des propriétés pour que celle-ci s’adapte correctement.

```
.section {
    background-image: linear-gradient(rgba(0, 0, 0, 0.8),
transparent), url(image.jpg);
    background-size: cover; (container)
    background-position: bottom;
    background-repeat: no-repeat;
}
```

CSS : les images <img>

Afin qu'une image s'adapte correctement à son container , on peut aussi spécifier certaines propriétés.

```css
.img {
    width: 100%;
    height: 100%;
    object-fit: cover;
    object-position: center;
}
```

CSS : display

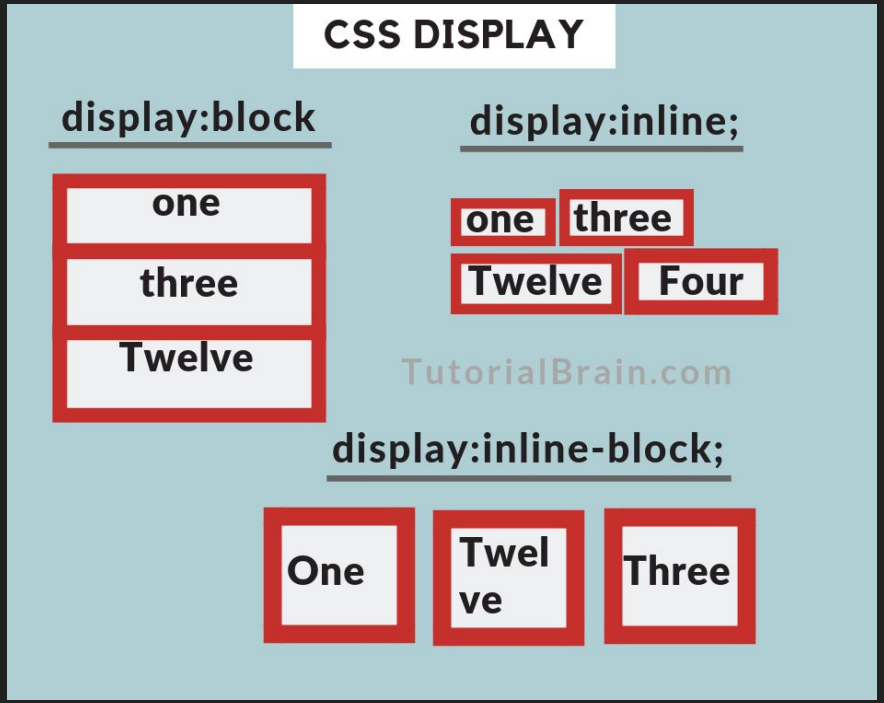
Il est important de comprendre comment le navigateur organise les éléments à l'écran.

Consultez ce tutoriel où il est expliqué les différentes manières dont un élément peut être disposé à l'écran.

Vous pouvez modifier la façon dont les éléments sont disposés à l'aide de la propriété display :

```
div { display: block ; }
```

Regardez aussi la propriété float.

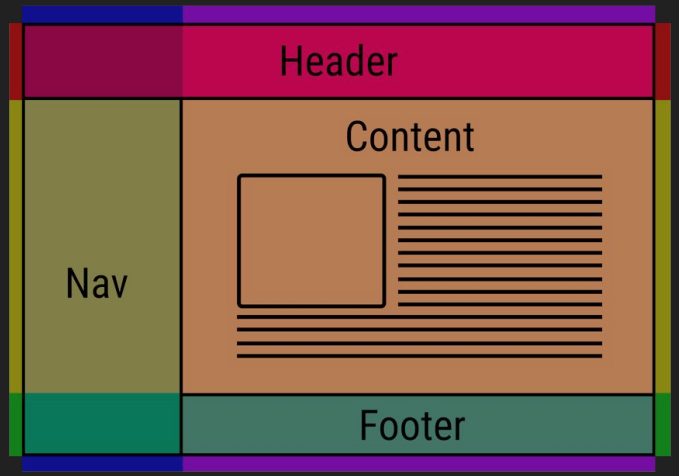


CSS : layout

L'une des parties les plus difficiles du CSS consiste à construire la mise en page de votre site Web (la structure à l'intérieur de la fenêtre).

Par défaut, HTML a tendance à tout mettre sur une seule colonne, ce qui n'est pas idéal.

Il y a eu de nombreuses possibilités en CSS pour résoudre ce problème (table, float, flex, grid, …).



CSS : positionner les blocs

On pouvait utiliser 5 méthodes pour positionner les blocs:

- Tableaux: <table>
- Inline-block: { display: inline-block; }
- Position: absolute: { position: absolute; }
- Float: { float: left; }
- Flex: { display: flex; } (bonne méthode)
- Grid: { display: grid; } (bonne méthode)

CSS : flexbox

Le module de mise en page de boîte flexible facilite la conception d'une structure de mise en page flexible et réactive sans utiliser de float ou de positionnement.

Le gros avantage de ce nouveau format c'est qu'il nous permet de piloter le comportement des blocs enfant directement depuis l'élément principal, permettant ainsi d'éviter d’avoir à mettre plein de propriétés sur les enfants.

CSS : { display: flex; }

```css
div {
    display: flex;
    flex-direction: row; /* column */
    justify-content: center; /* space-around|space-between */
    align-item: center; /* flex-start | flex-end */
    flex-wrap: wrap; /* no-wrap */
    gap: 10px;
}
```

CSS : astuce pour centrer

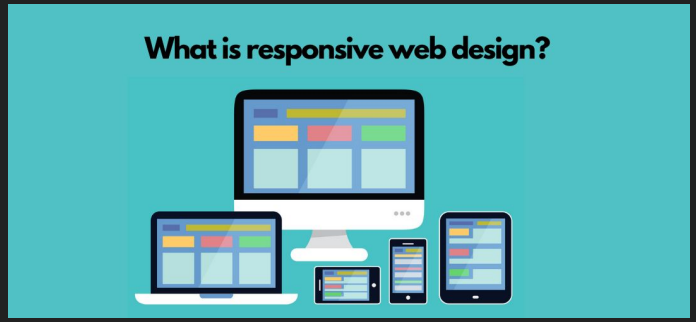
Centrer des div peut parfois être difficile, utilisez cette astuce :

```css
.horizontal-and-vertical-centering {
  display: flex;
  justify-content: center;
  align-items: center;
}
```

consultez ce tutoriel pour créer facilement la structure du site

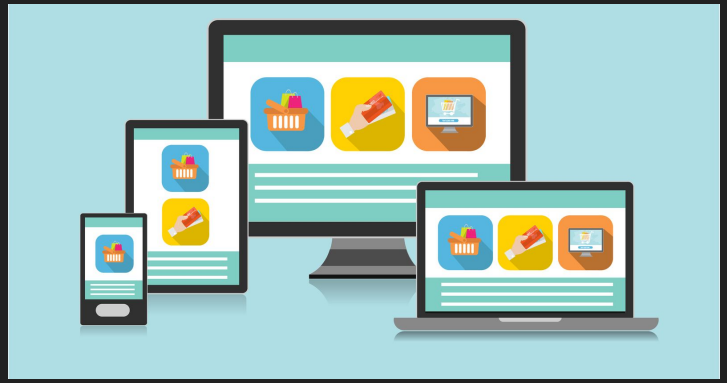
CSS : RWD (Responsive Web Design)

Un site web adaptatif (anglais RWD pour responsive web design, conception de sites web adaptatifs) est un site web dont la conception vise, grâce à différents principes et techniques, à offrir ***une consultation confortable même pour des supports différents.***



CSS : media query

```css
@media screen and (max-width: 768px) {
   .hero-section {
     padding: 15px 30px;
   }
}
```



CSS : adaptatif

Avec le responsive design, le contenu du site reste le même qu'importe le device et se réorganise dès qu'il atteint son seuil prédéfini ou point de rupture (breakpoint).

CSS : Création responsive

- Adapté à tous les écrans !
- Media queries
- Unités de mesure adaptées (%, px, rem, em, vh, vw)
- Images responsives (et qualité adaptée)
- Coder son site en desktop first ou mobile first
- Texte responsive

CSS : positionnement

La propriété position spécifie le type de méthode de positionnement utilisé pour un élément.
Les propriétés top, right, bottom et left déterminent l'emplacement final de l'élément positionné et z-index la superposition des éléments :

- { position : static; }
- { position : fixed; }
- { position : absolute; }
- { position : relative; }
- { position : sticky; }

CSS : { position: static; }

Comportement normal (par défaut). L'élément est alors positionné dans le flux avec sa position. Les propriétés top, right, bottom, left et z-index ne s'appliquent pas.

Tous les éléments ont la valeur static par défaut : Ils ne sont pas considérés comme des éléments positionnés.

CSS : { position: fixed; }

Un élément avec { position: fixed ; } est positionné par rapport à la fenêtre d'affichage, ce qui signifie qu'il reste toujours au même endroit même si la page défile.
Les propriétés top, right, bottom et left sont utilisées pour positionner l'élément.

Un élément fixe ne laisse pas de vide dans la page où il aurait normalement été situé.

CSS : { position: fixed; }

```css
.item {
    position: fixed;
    bottom: 50px;
    right: 50px;
    z-index: 1;
}
```

CSS : { position: relative; }

L'élément est positionné dans le flux normal du document puis décalé, par rapport à lui-même, selon les valeurs fournies par top, right, bottom et left.

Le décalage n'a pas d'impact sur la position des autres éléments. Aussi, l'espace fourni à l'élément sur la page est le même que celui fourni avec static.

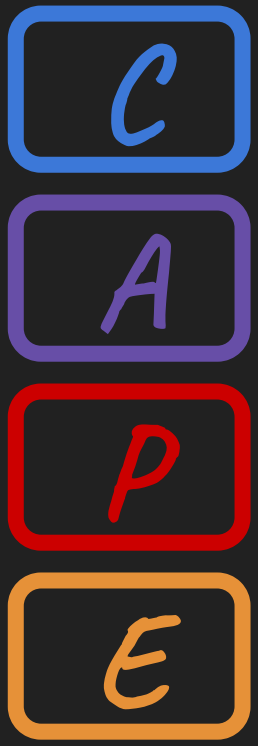
CSS : { position: absolute; }

L'élément est retiré du flux normal et aucun espace n'est créé pour l'élément sur la page.

Il est ensuite positionné par rapport à son ancêtre le plus proche qui est positionné s'il y en a un, ou par rapport au body.

La position finale de l'élément est déterminée par les valeurs de top, right, bottom et left.

CAPE : méthode d’apprentissage



Comprendre 30%

Apprendre 10%

Pratiquer 50%

Expliquer 10%

CSS : erreurs courantes

Accolade

on oublie de fermer l’accolade

Mauvaise organisation du CSS

aucune organisation n’a été mise en place et on est très vite perdu dans le fichier css

Commenter le code

absence de commentaire et le manque d’organisation vont rendre laborieux votre travail d’intégration

CSS : lectures complémentaires

Consultez certains des liens pour mieux les CSS :

Understanding the Box Model : Une bonne explication sur le modèle de boîte des éléments block en CSS.

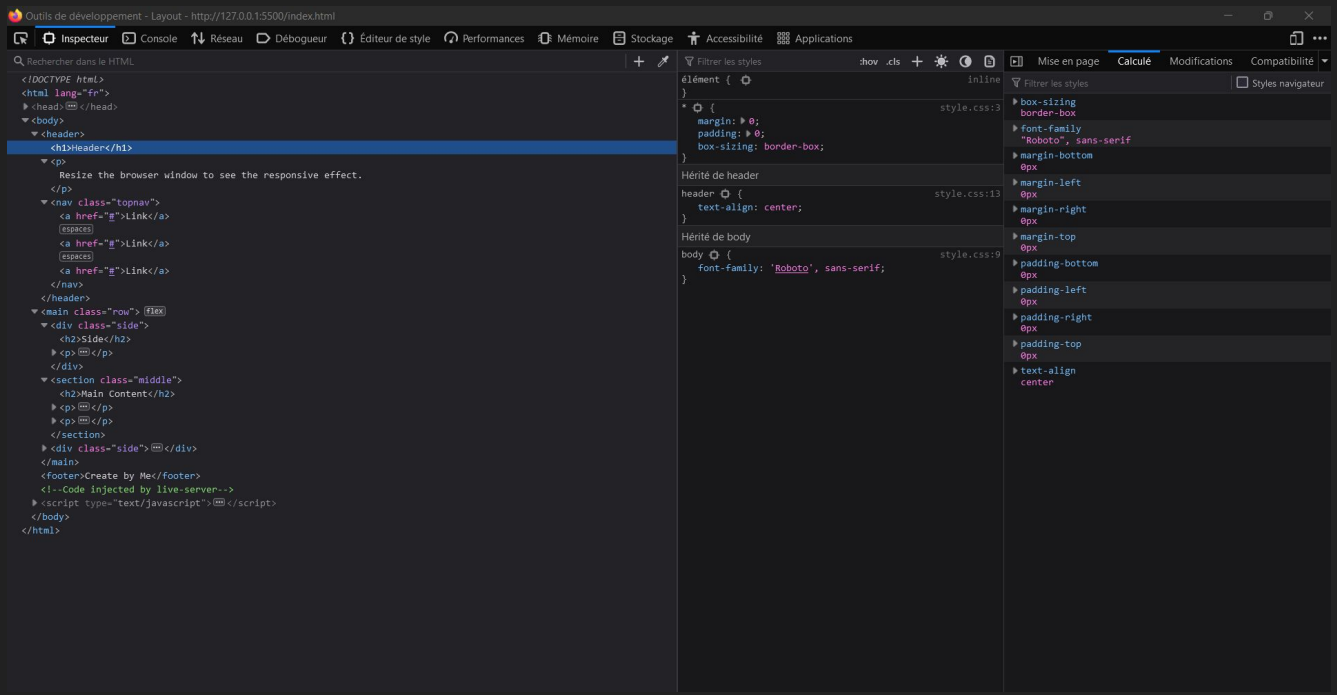
All CSS Selectors : 30 sélecteurs CSS à connaître.

CSS Transition : comment créer des animations uniquement en utilisant la propriété CSS { transition: ; }

One line layouts tutorials : Dispositions de sites web.

Utilisation des outils de développement

Press Control + Shift + i (ou F12) pour ouvrir l'inspecteur



< h1>MERCI<h1>

Technologies

- HTML
- CSS
- Javascript