

Introduction

HTML | CSS



Objectifs pédagogiques

- Comprendre internet, son évolution et apprendre à créer un site internet avec une structure HTML, une mise en style en CSS et le dynamiser avec Javascript.



C'est quoi un internet ?

- (contraction de *Inter Network*) est un réseau informatique qui relie des ordinateurs entre eux à l'échelle du monde.
- Gigantesque réseau composé de millions de réseaux publics et privés plus petits, par exemple des réseaux universitaires, gouvernementaux ou commerciaux.

Et le Web ?

- Abréviation de **World Wide Web**, est un service d'internet parmi tant d'autres
- Système mettant à disposition des documents (**hypertexte**)
 - Texte
 - Images
 - Videos
 - Etc...



URI (Uniform Resource Identifier)

URI : Chaîne de caractères utilisées pour identifier une ressource sur internet

Protocole du serveur

Protocole

Nom de domaine

Site

Nom de la ressource

Fichier

http://www.website.com/Accueil/index.html

Hôte

Chemin

Chemin d'accès à une ressource

Uniform Resource Identifier

URI : URL + URN

Uniform Resource Identifier

URI

<http://www.website.com/index.html>

URL

Uniform Resource Locator
Localisateur uniforme de ressource

URN

Uniform Resource Name
Nom uniforme de ressource

HTML (HyperText Markup Language)

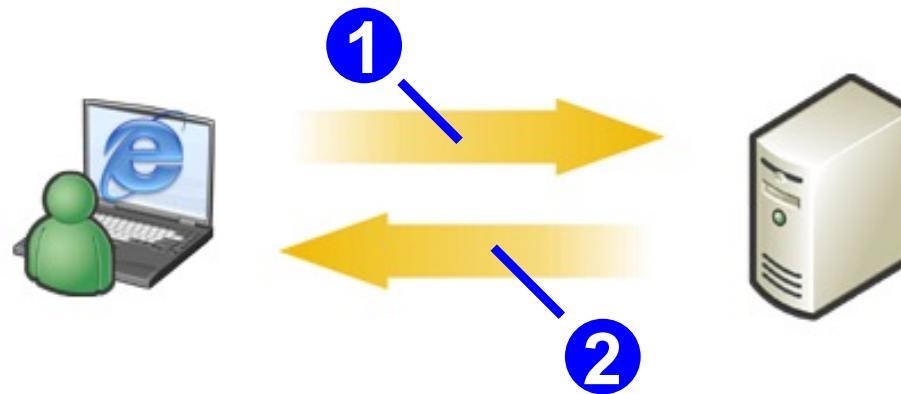
- Langage descriptif
- Utilisé pour formater un document
- Version actuelle : **HTML5.2**
- Concept hypertexte :
 - Facile à utiliser
 - Format léger

HTML



Modèle client - serveur

- 1 Le client transmet **une requête HTTP** au serveur Web pour atteindre une ressource
- 2 Le serveur Web répond en renvoyant une page HTML ou une autre ressource.



Le protocol HTTP

- HyperText Transfer Protocol
- Protocole de communications développé pour le Web
- Protocole de Requête/Réponse
 - Client (Navigateur)
 - Serveur (Serveur Web)
- Transfert de données entre navigateur et serveur Web
- Autres protocoles: HTTPS, FTP, POP, SMTP...

Message HTTP

- Ligne de requête composé de :
 - La méthode de requête utilisée
 - L'URI de la ressource
 - Le protocole et la version utilisée
- Plusieurs en-têtes (Headers)
- Une ligne vide
- Un corps de message optionnel (message body)

Exemple d'un message d'une requête HTTP:

```
POST /en/html/index.html HTTP/1.1
Host: www.website.com
User-Agent: Mozilla/5.0 (Windows; en-GB; rv:1.8.0.11)
Accept:
text/xml, text/html; q=0.9, text/plain; q=0.8, image/png,
 */*; q=0.5
Accept-Language: en-gb, en; q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1, utf-8; q=0.7, *; q=0.7
Keep-Alive: 300
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 39

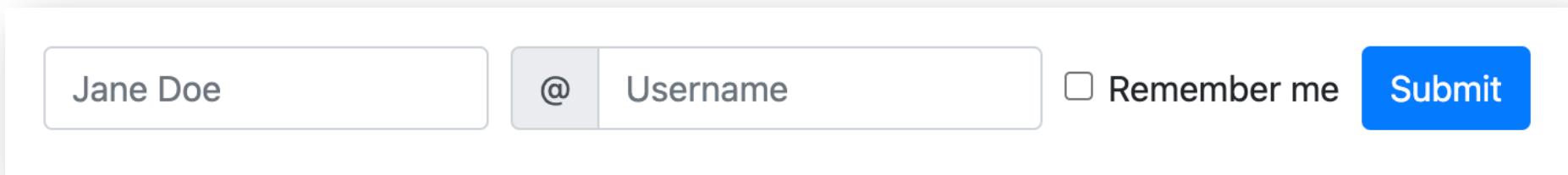
name=MyName&male=yes
```

Les méthodes HTTP

- HTTP défini 9 méthodes (verbes) :
 - **GET**: demande une representation de la ressource
 - **POST**: Soumet les données à traiter à la resource identifiée
 - Les données sont incluses dans le corps de la demande
 - **HEAD**: Identique à **GET** mais renvoie uniquement les en-têtes
 - **PUT**: *Upload* une representation de la resource spécifiée
 - **DELETE**: Supprime la resource spécifiée
 - ...

Quizz 1/3

Avec le navigateur, je complète le formulaire :



The image shows a light gray rectangular area representing a web page. Inside, there's a horizontal form. On the left is a text input field containing "Jane Doe". Next to it is another input field with a small gray box containing "@" followed by a larger box labeled "Username". To the right of these is a small square checkbox with the text "Remember me" next to it. On the far right of the form is a blue rectangular button with the word "Submit" in white.

- Quelle serait la méthode HTTP utilisée dans le message HTTP de requête au serveur ?

Quizz 2/3

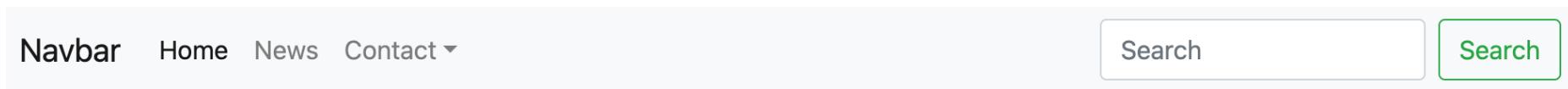
Avec le navigateur, je recherche un appartement dans la barre de recherche :



- Quelle serait la méthode HTTP utilisée dans le message HTTP de requête au serveur ?

Quizz 3/3

Je navigue sur le site internet, en cliquant sur News :



- Quelle serait la méthode HTTP utilisée dans le message HTTP de requête au serveur ?

Les codes de statut de réponse HTTP

- HTTP défini 5 familles de réponses :
 - **Informatives** : (100 à 199)
 - **Succès** : (200 à 299)
 - **Redirection** : (300 à 399)
 - **Erreur client** : (400 à 499)
 - **Erreur Serveur** : (500 à 599)

Quizz 1/3

- Le client (navigateur) navigue sur une url, le serveur détecte que cette url n'existe pas. (non défini côté serveur)
- Quel devrait-être le code de statut HTTP renvoyé ?

Quizz 2/3

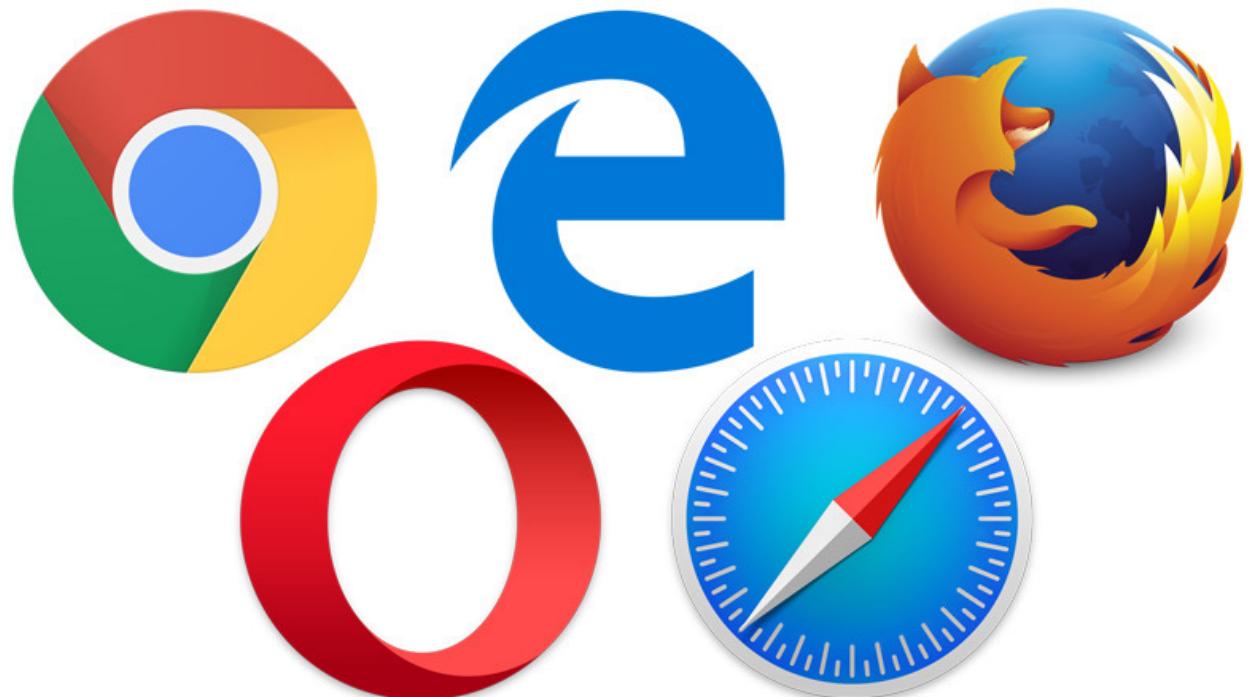
- Le client (navigateur) navigue sur une url interdite.
- Quel devrait-être le code de statut HTTP renvoyé par le serveur ?

Quizz 3/3

- Une erreur d'accès à la base de données empêche le serveur de renvoyer les informations demandées par le client (navigateur).
- Quel devrait-être le code de statut HTTP renvoyé ?

Les navigateurs internet

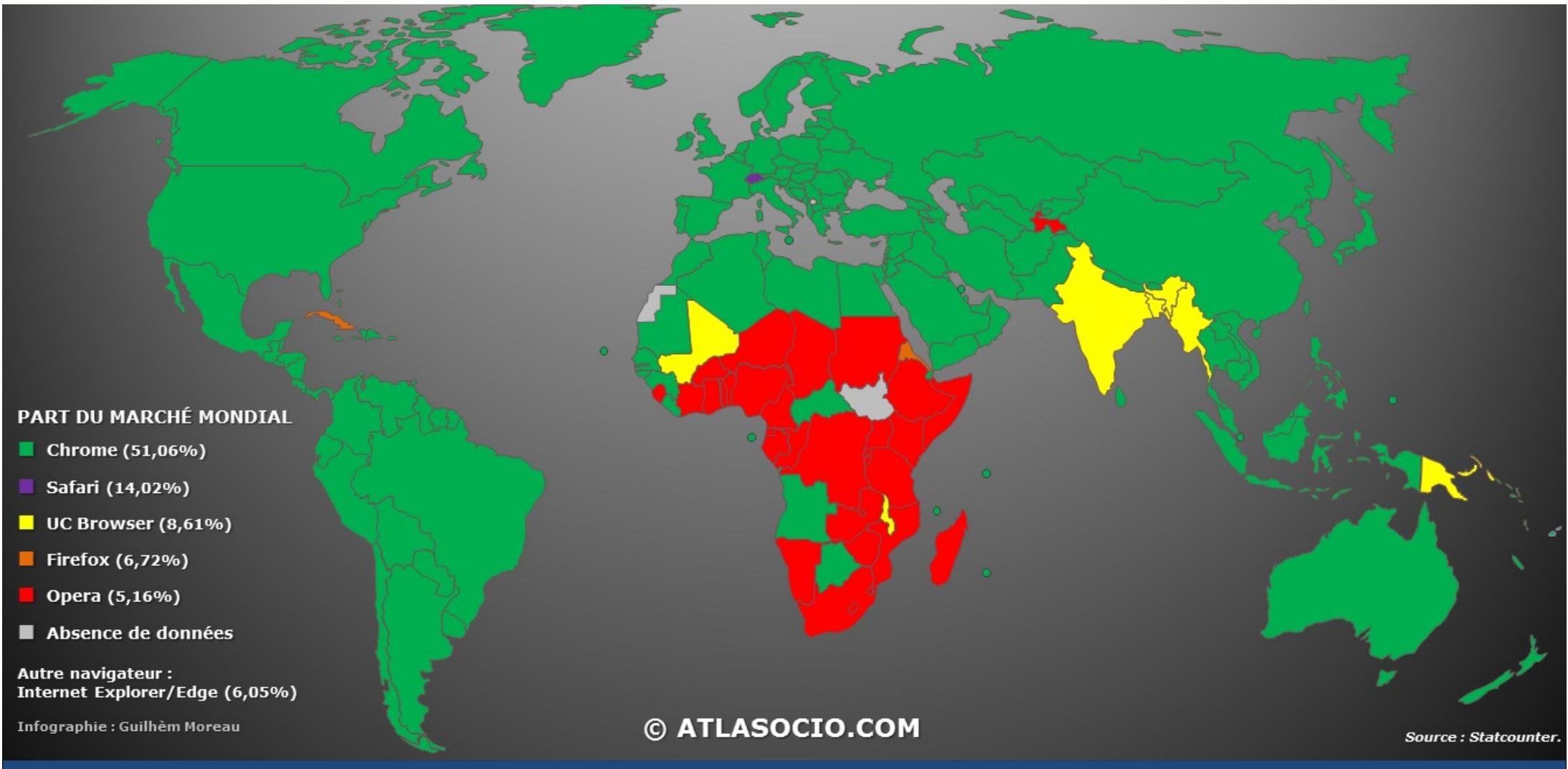
- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari
- Opéra



2008 : Marché des navigateurs



2016 : Marché des navigateurs



Exercice (1/2)

- Essayer d'envoyer une requête HTTP et observer la réponse
 - Sur Google Chrome:
 - Installer l'extension Postman
 - Sur Mozilla Firefox:
 - Installer l'extension Postman
- Si vous utilisez autre chose, il est temps de changer ☺

Exercice (2/2)

- Essayer différentes requêtes sur différents site
 - <http://www.google.fr>
 - <http://www.facebook.com/>
- Essayer d'envoyer une requête POST
 - Sur <https://login.facebook.com/login.php> par exemple
 - Avec vos accès dans le corps de la requête
- Essayer avec d'autres site

W3C (World Wide Web Consortium)

- W3C pour World Wide Web Consortium
- Crée en 1994
- organisme international qui développe des standards pour le Web afin que les gens puissent communiquer efficacement à travers Internet
- <http://www.w3.org/>

Exercice (1/2)

- Créez votre premier fichier HTML
 - Télécharger un éditeur de texte pour le web tels que :
 - Sublime Text
 - Visual Studio
 - Web Storm
 - Notepad++
 - Créez un fichier avec l'extension .html

Exercice (2/2)

- Écrire dans le fichier :

```
<html>
  <head>
    <title>Ma superbe première page</title>
  </head>
  <body>
    <h1>Ma page internet</h1>
    <p>HTML, je vais enfin te découvrir !</p>
  </body>
</html>
```

- Ouvrir le fichier avec votre navigateur

Introduction

HTML5



Structure de base

- **DOCTYPE** indique le format aux anciens navigateur
- **html** est l'élément racine du document
 - Contient l'attribut **lang** (langue du contenu)
 - Contient **head** et **body**
- **head** regroupe les métadonnées du site
 - Données non visible sur le site
 - Liens CSS
 - Chargements de scripts Javascript
- **body** contient le reste du document
 - Visible dans le navigateur

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Titre de la page</title>
  <link rel="stylesheet" href="style.css">
  <script src="script.js"></script>
</head>
<body>
...
<!-- Le reste du contenu -->
...
</body>
</html>
```

Les balises meta

- Meta-charset indique le jeu de caractères utilisés par le document
- Meta-author indique l'auteur
- Etc

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <title>Mon document </title>
  <!-- Métadonnées -->
  <meta charset="UTF-8">
  <meta name="author" content="Boris Sauvage">
  <meta name="description" content="Ma première HTML">
  <meta name="keywords" content="html, ma page">
</head>
```

- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/meta>

Les textes

- **<h1>** à **<h6>** représente 6 niveaux de titres
- **<h1>** doit être unique dans la page
- **<p>** est un paragraphe
- **** est une accentuation forte
- **** est une accentuation

```
<!DOCTYPE html>
<html>
  <head>
    <title>Les textes</title>
    <meta charset="UTF-8">
  </head>

  <body>
    <h1>Titre 1</h1>
    <h2>Titre 2</h2>
    ...
    <h6>Titre 6</h6>

    <p>Ceci est un <strong>texte fort</strong> et
      ceci est un <em>texte en      emphase</em>. </p>

  </body>
</html>
```

Les médias

```
<!-- Affichage d'une image stockée dans le dossier assets/images -->


<!-- Affichage d'une vidéo stockée dans le dossier assets/videos -->
<video controls>
    <source src=".assets/videos/flower.mp4" type="video/mp4" />
</video>

<!-- Affichage d'une vidéo via une url -->
<video controls>
    <source src="https://sample-videos.com/video321/mp4/720/big_buck_bunny_720p_2mb.mp4" type="video/mp4" />
</video>

<!-- Vidéo youtube intégrée -->
<iframe width="560" height="315" src="https://www.youtube.com/embed/9xwazD5SyVg?si=NjyMyT_egwS85t-3"
title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media;
gyroscope; picture-in-picture; web-share" allowfullscreen></iframe>
</body>
```

Les formulaires 1/2

- <**form**> représente un formulaire
 - **action** est l'url qui traite l'envoi du formulaire
 - **method** définit la méthode HTTP utilisée pour envoyer les données
- <**input**> est un contrôle interactif
 - **Type** influence le comportement
- <**label**> est la légende de l'input
- <https://developer.mozilla.org/fr/docs/Web/HTML/Element/form>

```
<form action="http://votreserveur.fr/" method="POST">
</form>
```

```
<div>
  <label for="email">Email</label>
  <input id="email" type="email" pattern="[a-zA-Z0-9._%+-]+@[a-zA-Z.-]+\.[a-zA-Z]{2,}" required>
</div>
```

```
<input type="submit" value="Register">
```

Les formulaires 2/2

```
<form action="http://votreserveur.fr/" method="POST">
    <div>
        <input id="radio_single" type="radio">
        <label for="radio_single">Single</label>
    </div>
    <div>
        <input id="radio_couple" type="radio">
        <label for="radio_couple">Couple</label>
    </div>
    <div>
        <label for="lastName">Lastname</label>
        <input id="lastName" type="text" minlength="2">
    </div>
    <div>
        <label for="email">Email</label>
        <input id="email" type="email" required>
    </div>
    <div>
        <label for="password">Password</label>
        <input id="password" type="password" required>
    </div>
</form>
```

```
<div>
    <label for="birthDate">Birth Date</label>
    <input id="birthDate" type="date" required>
</div>
<div>
    <label for="pictureProfil">Profil Picture</label>
    <input id="pictureProfil" accept="image/*" type="file" required>
</div>
<div>
    <input type="checkbox" id="tennis" checked />
    <label for="tennis">Tennis</label>
</div>
<div>
    <input type="checkbox" id="scubadiving" />
    <label for="scubadiving">Scuba Diving</label>
</div>
<div>
    <input type="checkbox" id="painting" />
    <label for="painting">Painting</label>
</div>
<input type="submit" value="Register">
</form>
```

Les entités HTML

- chaîne de texte qui commence par **&** et se termine avec **;**
- Utilisées pour afficher des caractères réservés et des caractères invisibles
 - Comme des espaces insécables
 - Comme d'autres caractères difficiles à taper avec un clavier standard

Caractère	Entité	Remarque
&	&	Début d'une référence d'entité ou de caractère.
<	<	Début d'une balise
>	>	Fin d'une balise
"	"	Début et la fin d'une valeur d'attributs
©	©	Copyright

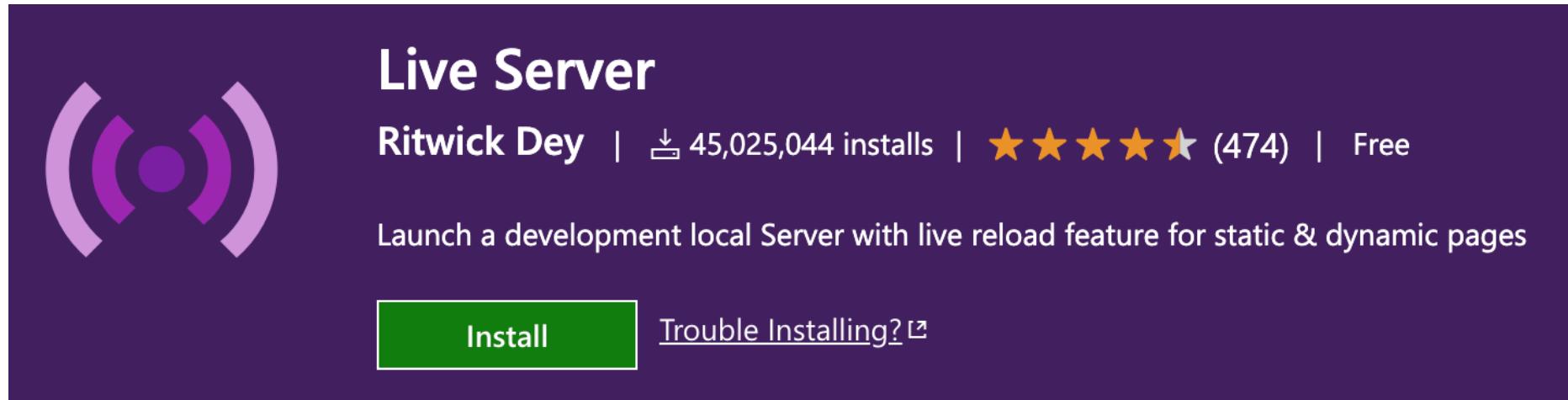
- <https://developer.mozilla.org/fr/docs/Glossary/Entity>

W3C Markup Validation

- Outil en ligne fourni par le World Wide Web Consortium (W3C)
- Permet de vérifier la conformité d'une page web aux standards du HTML
- Pour vérifier l'HTML : <https://validator.w3.org/>
- Pour vérifier le CSS : <https://jigsaw.w3.org/css-validator/>

Extension LiveServer

- Permet de lancer un serveur local avec recharge automatique du rendu
 - <https://marketplace.visualstudio.com/items?itemName=ritwickdey.LiveServer>



Structure HTML5

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="utf-8">
  <title>Titre de la page</title>
</head>
<body>
<header>
<nav> ...</nav>
</header>
<footer></footer>
</body>
</html>
```

<header>, <nav>, <section>, <article>, <footer>

... sont des nouvelles balises HTML5

Introduction

CSS3



Le CSS 1/2

- **CSS** (*Cascading Style Sheets*) permet de créer des pages web à l'apparence soignée
- Langage basé sur des règles de styles destinées à des éléments ou des groupes d'éléments dans la page

Exemple : « Je veux que le titre principal de ma page s'affiche en rouge en gros caractères. »

```
h1 {  
    color: red;  
    font-size: 5em;  
}
```

Le CSS 2/2

- Il est possible de déclarer du CSS dans :
 - Un fichier CSS
 - Une balise `<style></style>`
 - Un attribut style sur l'élément html

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style> ... </style>
</head>
```

```
<nav style="color: blue">
  ...
</nav>
```

Liaison du HTML et du CSS

- Il est recommandé de lier le fichier HTML avec le CSS

```
<!doctype html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <link rel="stylesheet" href=".style.css">
</head>
```



L'inspecteur du navigateur

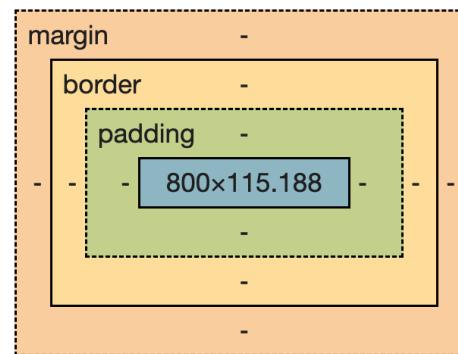
The screenshot shows the Firefox Developer Tools DevTools panel. The top bar has tabs for Éléments, Console, Enregistreur, Informations sur les performances, Sources, Réseau, and Performances. The Styles tab is selected in the bottom right corner.

The left pane displays the DOM tree:

```
<div class="homepage mdn-ui-body-m"> (flex)
  <div class="homepage-hero dark"> (grid)
    <section> (flex)
      <h1> ... </h1> == $0
      <p>Documenting web technologies, including CSS, HTML, and JavaScript, since 2005.</p>
      <div class="homepage-hero-search">...</div> (flex)
    </section>
    <div class="mandala-container homepage-hero-bg" aria-hidden="true">...
      </div> (flex)
    </div>
    <section class="place hp-main"></section> (flex)
    <div class="featured-articles">...</div> (flex)
    <section class="latest-news">...</section> (flex)
    <section class="recent-contributions">...</section> (flex)
```

The right pane shows the Styles tab with the following CSS rules:

```
element.style {
}
.homepage-hero section h1, .homepage-hero section p {
  text-shadow: var(--contrast-boost-shadow);
}
@media screen and (min-width: 769px) {
  .homepage-hero section h1 {
    font-size: 3rem;
  }
}
@media screen and (min-width: 426px) {
  .homepage-hero section h1 {
    font-size: 2.5rem;
  }
}
```



Démonstration de L'inspecteur du navigateur

L'attribut id

- Définit un identifiant unique dans le document hypertext
- Est utilisable pour :
 - Les labels de formulaires
 - Les liens ancre
 - Les sélecteurs CSS

```
<div id="introduction">  
...  
</div>
```

Les sélecteurs

- Définissent les éléments sur lesquelles s'applique un ensemble de règles CSS
- Il existe de nombreux sélecteurs :
 - Sélecteur d'ID
 - Sélecteur de type
 - Sélecteur de classe
 - Sélecteur d'identifiant
 - Sélecteur universel
 - Les combinateurs
 - Les pseudo-classes
 - Les pseudo-éléments

Sélecteur d'ID

- Ciblent un élément grâce à la valeur de son id

```
#introduce {  
    color: blue;  
}
```

```
<div id="introduction">  
    ...  
</div>
```

- https://developer.mozilla.org/fr/docs/Web/CSS/ID_selectors

Sélecteur de type

- Ciblent des éléments en fonction du nom de leur type
 - cible tous les nœuds avec ce nom contenus dans le document

```
a {  
    color: blue;  
}
```

```
<nav>  
    <a href="#">Lien 1</a>  
    <a href="#">Lien 2</a>  
    <a href="#">Lien 3</a>  
    <a href="#">Lien 4</a>  
</nav>
```

- https://developer.mozilla.org/fr/docs/Web/CSS/Type_selectors

Sélecteur de classe

- Ciblent des éléments en fonction du contenu de l'attribut class de chaque élément

```
/* Cible tous les éléments ayant la classe "spacious" */
.spacious {
    margin: 2em;
}

/* Cible tous les éléments <li> ayant la classe "spacious" */
li.spacious {
    margin: 2em;
}

/* Cible tous les éléments <li> ayant une classe qui "spacious" et "elegant" */
li.spacious.elegant {
    margin: 2em;
}
```

```
<ul>
    <li class="spacious elegant">Element 1</li>
    <li class="spacious elegant">Element 2</li>
    <li class="spacious elegant">Element 3</li>
</ul>
```

- https://developer.mozilla.org/fr/docs/Web/CSS/Class_selectors

Sélecteur universel

- L'astérisque * est le sélecteur universel en CSS
 - correspond à un élément de n'importe quel type

```
* {  
  color: green;  
}
```

- https://developer.mozilla.org/fr/docs/Web/CSS/Universal_selectors

Sélecteur d'attribut

- Ciblent un élément selon la présence d'un attribut ou selon la valeur donnée d'un attribut

```
/* Les éléments <a> avec un attribut title */
a[title] {
    color: purple;
}

/* Les éléments <a> avec un href qui correspond à "https://example.org" */
a[href="https://example.org"]
{
    color: green;
}

/* Les éléments <a> dont l'attribut class contient le mot logo */
a[class~="logo"] {
    padding: 2px;
}
```

- https://developer.mozilla.org/fr/docs/Web/CSS/Universal_selectors

Sélecteur enfant

- Ciblent seulement les éléments correspondant au second sélecteur qui sont des enfants **directs** des éléments ciblés par le premier sélecteur

```
/* Les éléments <li> qui sont des enfant d'un <ul class="my-element"> */
ul.my-element > li {
    margin: 2em;
}
```

- https://developer.mozilla.org/fr/docs/Web/CSS/Child_combinator

Sélecteur descendant

- Combine 2 sélecteurs afin de cibler les éléments qui correspondent au second sélecteur uniquement si ceux-ci ont un élément ancêtre qui correspond au premier sélecteur
 - Les sélecteurs qui utilisent ce combinateur sont souvent appelés des sélecteurs de descendants.

```
/* Les éléments <li> qui sont des descendants d'un <ul class="my-element">
*/
ul.my-element li {
    margin: 2em;
}
```

- https://developer.mozilla.org/fr/docs/Web/CSS/Descendant_combinator

Combinateur de voisins directs

- Sépare 2 sélecteurs et ne fait correspondre le second élément que s'il suit *immédiatement* le premier élément, et si les 2 sont des enfants du même élément parent

```
/* Ne cible que les paragraphes situés directement après une image */
img + p {
    font-style: bold;
}
```

- https://developer.mozilla.org/fr/docs/Web/CSS/Next-sibling_combinator

Combinateur de voisins suivants

- Permet de séparer 2 sélecteurs et de cibler *toutes les instances* du deuxième élément qui suivent le premier (qui ne sont pas nécessairement voisins immédiats) et qui partagent le même élément parent

```
/* Cible tous éléments <p> qui arrivent après un élément <img>. */
img ~ p {
  color: red;
}
```

- https://developer.mozilla.org/fr/docs/Web/CSS/Subsequent-sibling_combinator

Sélecteur pseudo-classes

- Mot-clé ajouté à un sélecteur afin d'indiquer l'état spécifique ciblé

```
div:hover {  
    background-color: #f89b4d;  
}
```

- **:hover** permet d'appliquer le CSS lorsque l'utilisateur survole l'élément ciblé
- Il existe de nombreuses pseudo-classes
- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-classes>

Sélecteur pseudo-éléments

- Mot-clé ajouté à un sélecteur qui permet de mettre en forme certaines parties de l'élément ciblé par la règle

```
/* La première ligne de chaque élément <p>. */
p::first-line {
    color: blue;
    text-transform: uppercase;
}
```

- Il existe de nombreux pseudo-éléments
- <https://developer.mozilla.org/fr/docs/Web/CSS/Pseudo-elements>

Héritage et réinitialisation 1/2

- Les navigateurs web appliquent des styles par défaut
 - (User Agent Stylesheet)
 - Sinon le document afficherait une page blanche
- Chaque navigateur applique ses propres styles par défaut
- 2 méthodes permettent de contourner ces différences d'affichages
 - **Reset CSS**
 - **Normalize CSS (recommandée)**

```
ul {                                     feuille de style user-agent
    display: block;
    list-style-type: disc;
    margin-block-start: 1em;
    margin-block-end: 1em;
    margin-inline-start: 0px;
    margin-inline-end: 0px;
    padding-inline-start: 40px;
}
```

Héritage et réinitialisation 2/2

- **Le Reset CSS** réinitialise toutes les valeurs des propriétés CSS à leur état initial
 - Permet de repartir de zéro et d'une base vierge avant d'appliquer notre CSS

```
/* Reset CSS */  
* {  
    padding: 0;  
    margin: 0;  
}
```

- **Normalize.css** normalise l'apparence des éléments dans les différents navigateurs
 - <https://necolas.github.io/normalize.css/>
 - Il existe également **sanitize.css** et **reboot.css (Bootstrap)**
 - <https://csstools.github.io/sanitize.css/>
 - <https://getbootstrap.com/docs/5.3/content/reboot/#approach>

Les couleurs

Il existe de nombreuses façons d'utiliser des couleurs en CSS

- Les valeurs RGB hexadécimales

#02798b

- Les valeurs RGB et RGBA

- Red, Green, Blue, Alpha (opacité)

rgb(197 93 161)

- Les valeurs HSL et HSLA

- Teinte (Hue), Saturation, Luminosité

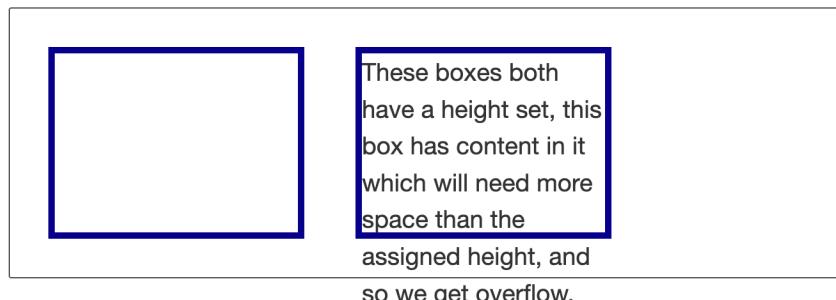
hsl(174 77% 31%)

Dimensionner

- Les éléments HTML ont une taille « naturelle » avant de faire du CSS

Exemple :

- Une image `` a une largeur et une hauteur définies par l'image
 - C'est la **taille intrinsèque** (taille naturelle)
- En revanche, une `<div>` n'a pas de taille, sans contenu à l'intérieur
 - Il est possible de fixer une taille à la **div**, nous parlerons de **taille extrinséque**



- Il n'est pas recommandé de fixer les tailles des éléments

Les unités de longueur absolue

Unité	Nom	Équivalent à
cm	Centimètres	1cm = 38px = 25/64in
mm	Millimètres	1mm = 1/10th of 1cm
Q	Quarts de millimètre	1Q = 1/40th of 1cm
in	Pouces (<i>inches</i>)	1in = 2.54cm = 96px
pc	Picas	1pc = 1/6e de 1in
pt	Points	1pt = 1/72e de 1in
px	Pixels	1px = 1/96th de 1in

- Ces unités sont utiles pour l'impression plutôt que pour l'affichage écran
 - La seule unité ici que vous rencontrerez fréquemment est px (pixels)

Les unités de longueur relative 1/4

Unité	Relative à
em	Pour les typographiques comme <u>font-size</u> (relative à la taille de la police de l'élément parent) <u>width</u> (relative à la taille de la police de l'élément)
ex	La hauteur d'x de la police de l'élément.
ch	La <u>chasse/avance</u> du glyphe « 0 » pour la police de l'élément.
rem	La taille de la police pour l'élément racine.
lh	La hauteur de ligne pour l'élément.
vw	1% de la largeur du <i>viewport</i> (la zone d'affichage).
vh	1% de la hauteur du <i>viewport</i> (la zone d'affichage).
vmin	1% de la plus petite dimension du <i>viewport</i> (la zone d'affichage).
vmax	1% de la plus grande dimension du <i>viewport</i> (la zone d'affichage).

Les unités de longueur relative 2/4

- Exemple :

```
<div class="wrapper">  
  <div class="box px">I am 200px wide</div>  
  <div class="box vw">I am 10vw wide</div>  
  <div class="box em">I am 10em wide</div>  
</div>
```

```
.wrapper {  
  font-size: 1em;  
}  
  
.px {  
  width: 200px;  
}  
  
.vw {  
  width: 10vw;  
}  
  
.em {  
  width: 10em;  
}
```

I am 200px wide

I am
10vw
wide

I am 10em wide

Ems et Rems 3/4

em et rem sont 2 unités que vous rencontrerez fréquemment

- **em** signifie « la taille de police de l'**élément parent** »
- **rem** signifie « la taille de police de l'**élément racine** »
 - Root em

Pourcentages 4/4

- Dans la plupart des cas, un pourcentage est traité comme une longueur
- Un pourcentage est toujours relatif à une autre valeur
- Exemple :
 - font-size en pourcentage est relatif à la valeur de font-size du parent
 - width, la valeur obtenue sera le pourcentage du width parent

Les variables

- Permet de déclarer des valeurs réutilisables

```
:root {  
  --main-color: #0065FC;  
  --main-bg-color: #F2F2F2;  
  --filter-bg-color: #DEEBFF;  
}  
  
section {  
  background-color: var(--main-color);  
}
```

Les polices d'écriture

- Il est possible d'intégrer des polices d'écritures avec CSS

Exemple : Utilisation de Google Font

- <https://fonts.googleapis.com/>

```
html {  
  font-family: "Raleway", sans-serif;  
  font-optical-sizing: auto;  
  font-weight: 800;  
  font-style: normal;  
}
```

```
<link rel="preconnect" href="https://fonts.googleapis.com">  
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>  
<link href="https://fonts.googleapis.com/css2?family=Raleway:ital,wght@0,100..900;1,100..900&display=swap" rel="stylesheet">
```

Disposition CSS: flux normal

- Lorsque le navigateur présente les pages HTML par défaut, sans CSS
 - C'est ce que l'on appelle **flux normal**

```
<p>Exemple de paragraphe.</p>
<ul>
  <li>Elément 1</li>
  <li>Elément 2</li>
  <li>Elément 3</li>
</ul>
<p>La fin&nbsp;!</p>
```

Exemple de paragraphe.

- Elément 1
- Elément 2
- Elément 3

La fin !

- Le HTML est affiché dans l'ordre exact où il est dans le code source
 - Les éléments disposés en empilement sont de type **block**
 - Les éléments disposés qui se succèdent comme des mots sont de type **inline**

Le positionnement : ! flux normal

- Permet de sortir les éléments du flux normal en se comportant différemment

Exemple :

- Placement d'un élément sur un autre
- Occuper toujours la même place dans la zone d'affichage du navigateur
- Il y a différents types de positionnement que vous pouvez appliquer à des éléments :
 - Positionnent static
 - Positionnent relatif
 - Positionnent absolu
 - Positionnent fixe
 - Positionnent sticky

Positionnement static

- Permet de positionner dans le flux normal de la page HTML
 - Comportement normal

Exemple :

- Placement d'un élément dans le flux normal
- Rien ne change

```
section {  
    position: static;  
    background: yellow;  
}
```

Positionnement relatif

- Similaire au positionnement static
- Une fois positionné, il est possible d'influencer sa position finale
- Utilise les propriétés **top, right, bottom, left** pour déterminer l'emplacement

Exemple :

```
section {  
    position: relative;  
    background: yellow;  
    top: 30px;  
    left: 30px;  
}
```

Positionnement absolu

- Positionnement par rapport au parent le plus proche
- Si aucun parent trouvé, il utilise le body
- Retiré du flux normal
- Position définie par les coordonnées (x,y)

Exemple :

```
section {  
    position: absolute;  
    background: yellow;  
    top: 30px;  
    left: 30px;  
}
```

Disposition CSS : display & float

- Chaque élément a une valeur **display** par défaut

Exemple :

- un paragraphe **<p>** est de type **display:block**
 - Les éléments s'affichent les uns sous les autres
- Un lien **<a>** est de type **display:inline**
 - Les éléments s'affichent les uns à côté des autres
- Il existe également **display:flex** et **display:grid**
 - faciliter une disposition alignée sur une dimension ligne ou colonne

Marge externe : margin

- Définit la taille des marges sur les 4 côtés de l'élément
- Propriété raccourcie qui permet de manipuler les marges par côtés
 - margin-top, margin-right, margin-left, margin-bottom

```
/* La propriété s'applique aux quatre côtés */
margin: 1em;

/* vertical | horizontal */
margin: 5% auto;

/* haut | horizontal | bas */
margin: 1em auto 2em;

/* haut | droit | bas | gauche */
margin: 2px 1em 0 auto;
```

- La valeur **auto** permet en répartissant la valeur de chaque côtés

Marge interne : padding

- Définit les différents écarts de remplissage sur les quatre côtés d'un élément
- Synthétise padding-top, padding-right, padding-bottom, padding-left

```
/* On applique la même valeur aux quatre côtés */
padding: 1em;

/* vertical | horizontal */
padding: 5% 10%;

/* haut | horizontal | bas */
padding: 1em 2em 2em;

/* haut | droit | bas | gauche */
padding: 5px 1em 0 2em;
```

Les boîtes flottantes

- Conçue pour faire flotter des images à l'intérieur d'un bloc de texte

```
<h1>Exemple simple de boîte flottante</h1>
<div class="box">Boîte flottante</div>
<p>Lorem50 ..</p>
<p>Lorem50 ...</p>
<p>Lorem50 ...</p>
```

```
.box {
  float: left;
  margin-right: 15px;
  width: 150px;
  height: 150px;
  border-radius: 5px;
  background-color: rgb(207, 232, 220);
  padding: 1em;
}
```

Exemple simple de boîte flottante

Boîte flottante

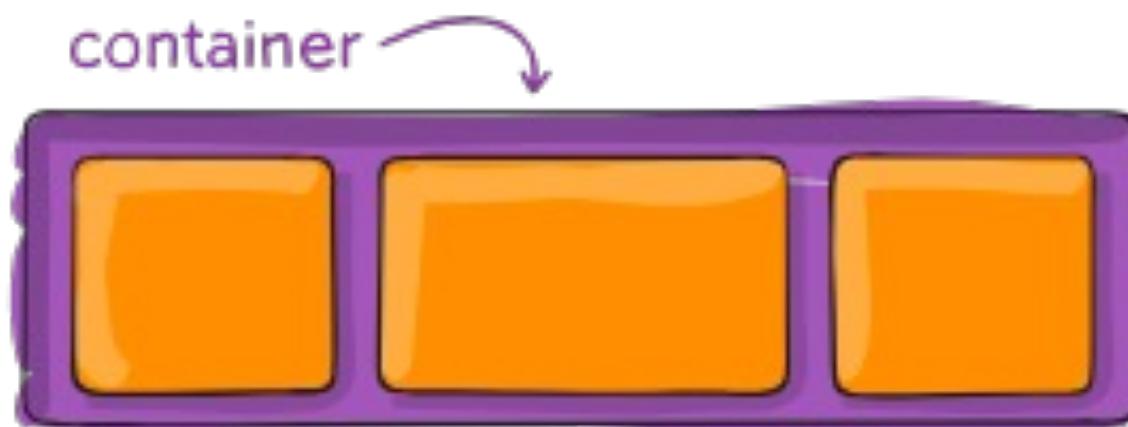
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla luctus aliquam dolor, eu lacinia lorem placerat vulputate. Duis felis orci, pulvinar id metus ut, rutrum luctus orci. Cras porttitor imperdiet nunc, at ultricies tellus laoreet sit amet.

Sed auctor cursus massa at porta. Integer ligula ipsum, tristique sit amet orci vel, viverra egestas ligula. Curabitur vehicula tellus neque, ac ornare ex malesuada et.

Nam vulputate diam nec tempor bibendum. Donec luctus augue eget malesuada ultrices. Phasellus turpis est, posuere sit amet dapibus ut, facilisis sed est. Nam id risus quis ante semper consectetur eget aliquam lorem. Vivamus tristique elit dolor, sed pretium metus suscipit vel. Mauris ultricies lectus sed lobortis finibus. Vivamus eu urna eget velit cursus viverra quis vestibulum sem. Aliquam tincidunt eget purus in interdum. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.

Les flexbox

- Pendant longtemps, les seuls outils de mise en page CSS fiables et compatibles avec les navigateurs, étaient les propriétés **float** et **position**
- Flexbox facilite considérablement les tâches de mise en page
- <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>



Aperçu pour tester 1/2

Sample flexbox example

First article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit totam voluptate? Culpa deleniti doloribus dolorum enim libero.

Second article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit totam voluptate? Culpa deleniti doloribus dolorum enim libero.

Third article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit totam voluptate? Culpa deleniti doloribus dolorum enim libero.

Aperçu pour tester 2/2

```
<header>
  <h1>Sample flexbox example</h1>
</header>

<section>
  <article>
    <h2>First article</h2>
    <p>Lorem50</p>
  </article>

  <article>
    <h2>Second article</h2>
    <p>Lorem50</p>
  </article>

  <article>
    <h2>Third article</h2>
    <p>Lorem50</p>
  </article>
</section>
```

```
html {
  font-family: sans-serif;
}

body {
  margin: 0;
}

header {
  background: #fff;
  height: 100px;
}

h1 {
  text-align: center;
  color: #1F2039;
  line-height: 100px;
  margin: 0;
}

article {
  color: #fff;
  padding: 10px;
  margin: 10px;
  background: #1F2039;
}
```

Les flexbox : display

- Mise en place du **display:flex** sur la section

Sample flexbox example

```
section {  
  display: flex;  
}
```

First article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similiqe totam ullam voluptate? Laudantium magni nisi, quos repellat sit totam voluptate? Culpa deleniti doloribus dolorum enim libero.

Second article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similiqe totam ullam voluptate? Laudantium magni nisi, quos repellat sit totam voluptate? Culpa deleniti doloribus dolorum enim libero.

Third article

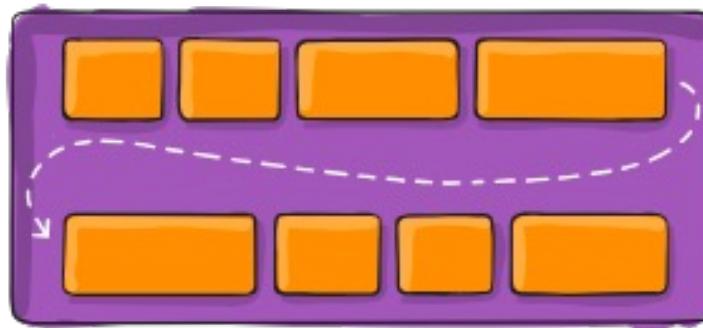
Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similiqe totam ullam voluptate? Laudantium magni nisi, quos repellat sit totam voluptate? Culpa deleniti doloribus dolorum enim libero.

Les flexbox : flex-wrap 1/2

- Par défaut, les éléments flexibles essayent de tenir sur une seule ligne
 - Vous pouvez autoriser les éléments à s'enrouler selon vos besoins avec cette propriété.

```
section {  
    flex-wrap: nowrap | wrap | wrap-reverse;  
}
```

- nowrap (**par défaut**) : les éléments flexibles sont disposés sur une seule ligne
- wrap: les éléments flexibles sont sur plusieurs lignes, de haut en bas
- wrap-reverse: les éléments flexibles sont sur plusieurs lignes, de bas en haut



Les flexbox : flex-wrap 2/2

- Aperçu :

Sample flexbox example

```
section {  
    display: flex;  
    flex-wrap: nowrap;  
}
```

First article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit.

Second article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit.

Third article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit.

X article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit.

X article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit.

X article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit.

X article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit.

X article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit.

X art

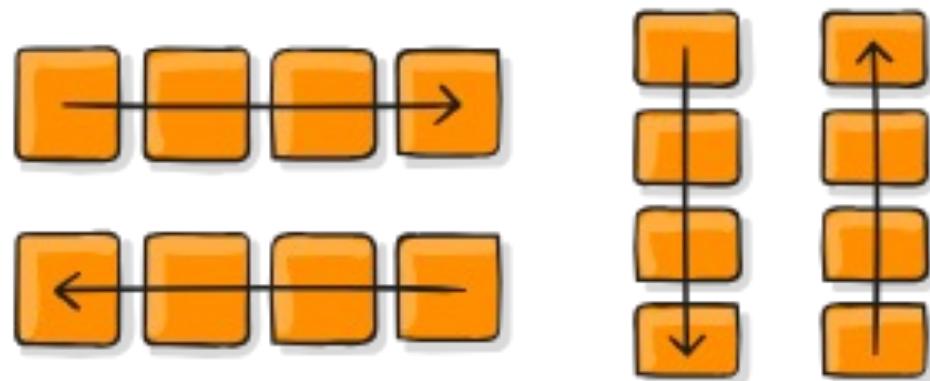
Lore ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit.

Les flexbox : flex-direction 1/2

- Mise en place de flex-direction sur la section

```
section {  
    flex-direction: row | row-reverse | column | column-reverse;  
}
```

- row (**par défaut**) : de gauche à droite
- row-reverse: de droite à gauche
- column: de haut en bas
- column-reverse: de bas en haut



Les flexbox : flex-direction 2/2

- Aperçu :

```
section {  
  display: flex;  
  flex-direction: column;  
}
```

Sample flexbox example

First article

Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit totam voluptate? Culpa deleniti doloribus dolorum enim libero.

Second article

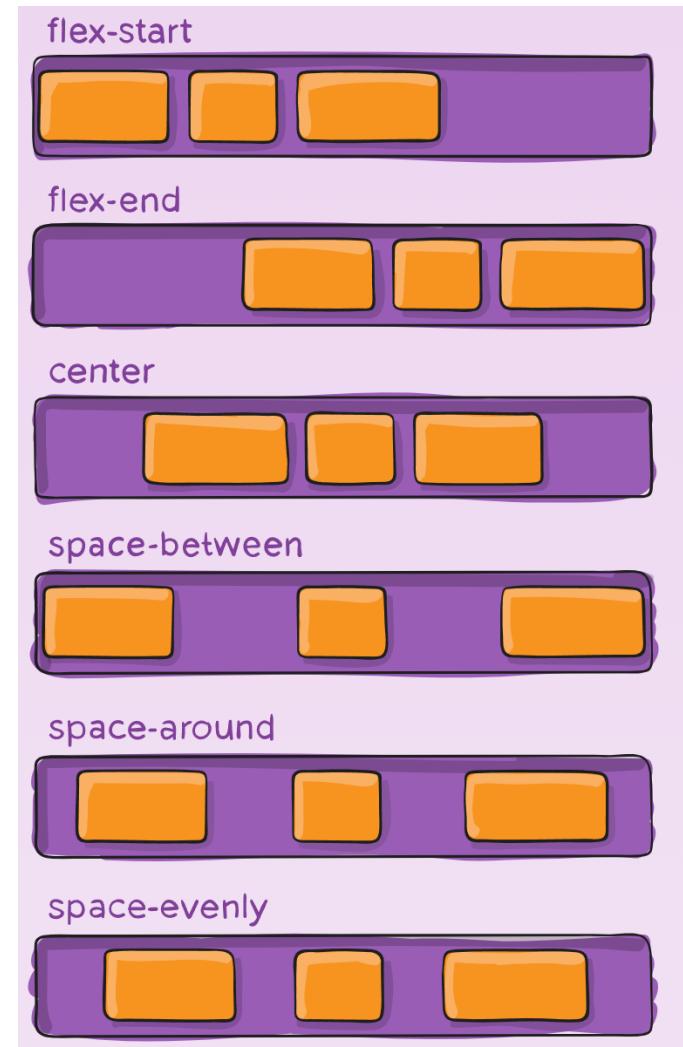
Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit totam voluptate? Culpa deleniti doloribus dolorum enim libero.

Third article

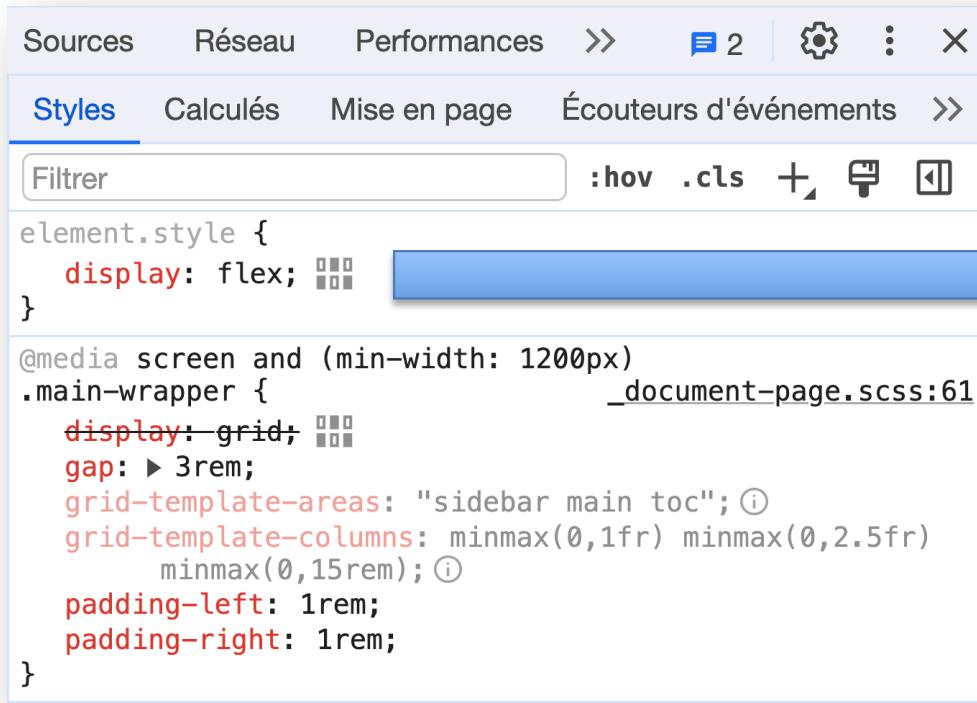
Lorem ipsum dolor sit amet, consectetur adipisicing elit. A atque autem, debitis dicta dolor earum enim facilis fugiat fugit impedit inventore itaque iusto libero modi, natus officia omnis pariatur quas repellat saepe similique totam ullam voluptate? Laudantium magni nisi, quos repellat sit totam voluptate? Culpa deleniti doloribus dolorum enim libero.

Les flexbox : justify-content

- Indique la façon dont l'espace doit être réparti entre et autour des éléments



Les flexbox : l'inspecteur



Sources Réseau Performances >> 2 : 2 : X

Styles Calculés Mise en page Écouteurs d'événements >>

Filtrer :hover .cls + ↕ ↖

```
element.style {
  display: flex; ⚡
}

@media screen and (min-width: 1200px)
.main-wrapper { _document-page.scss:61
  display: grid; ⚡
  gap: ▶ 3rem;
  grid-template-areas: "sidebar main toc"; ⓘ
  grid-template-columns: minmax(0,1fr) minmax(0,2.5fr)
                        minmax(0,15rem); ⓘ
  padding-left: 1rem;
  padding-right: 1rem;
}
```



Introduction

ASTUCES



Astuces

Sur VS Code :

- Installer et utiliser l'extension Emmet
 - Une fois installée, utilisez html:5



- Générer des paragraphes Lorem ipsum avec lorem50, lorem100, lorem150..
 - <https://fr.lipsum.com/>

Sur Google Chrome :

- Installer et utiliser l'extension simple-debug.css

Caniuse.com

- Service en ligne qui fournit des informations sur la compatibilité des fonctionnalités web avec différents navigateurs
- Permet de vérifier si les fonctionnalités HTML, CSS, JavaScript sont prises en charge par les principaux navigateurs
- <https://caniuse.com/>

- FIN -