

Licence 1



D'après Achref EL AMRI

S2 : 2025

Accès

- Indispensable

- Cours
- TD
- TP
- Quizz
- Dépôts
- Etc..

Mdp = tomczak

Vous vous inscrivez ?



Présentation Module HTML CSS JS

- Cours moodle :
- <https://moodle.uphf.fr/course/view.php?id=7633>

- 9 CM de 1h30 -> 12h
- 9 TD de 1h30 -> 12h
- 4 TP de 3h-> 12 h

- Notation : moyenne de
 - ▣ DS1 : intermédiaire (à la fin des TD)
 - ▣ TP : sous forme de projet
 - ▣ DS2 : fin de semestre

Plan du cours

5

- Chapitre 1: HTML
- Chapitre 2: CSS
- Chapitre 3: Javascript
- Chapitre 4: PhP

Chapitre 1

HTML



Plan

7

- HTML
- Structure d'un document HTML
- Syntaxe
 - ▣ Texte
 - ▣ Listes
 - ▣ Tableaux
 - ▣ Formulaires
 - ▣ etc.

Documentations

8

- L'officiel Documentation : <https://developer.mozilla.org/fr/docs/Web> (en Français)
- <https://www.w3schools.com/> (en Anglais)

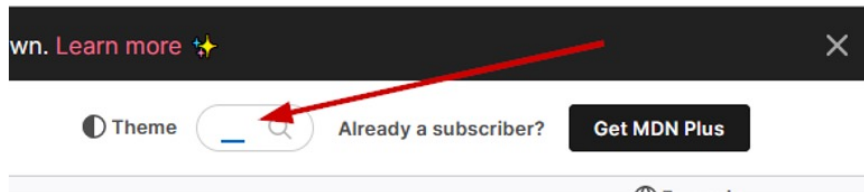
□ HTML

- ▣ <https://developer.mozilla.org/fr/docs/Web/HTML>
- ▣ <https://www.w3schools.com/tags/default.asp>

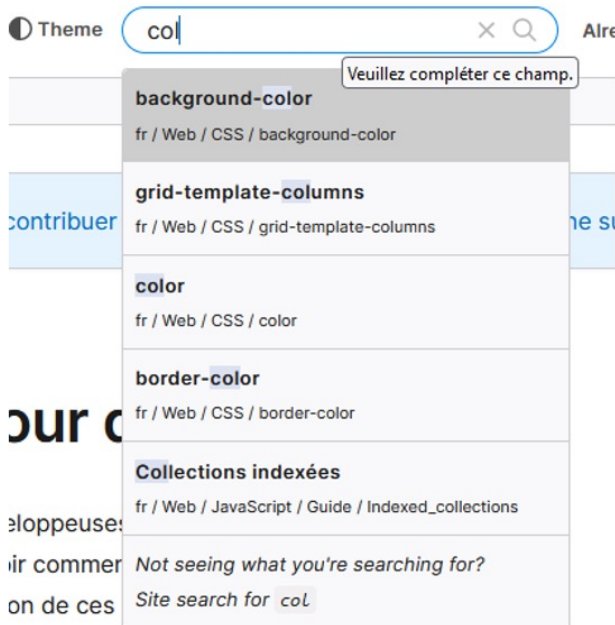
Comment recharger une information ?

9

Avec Documentation : <https://developer.mozilla.org/fr/docs/Web> :
la recherche se trouve là :



Puis la fenêtre s'agrandit :



Avec www.w3schools.com/ en haut à droite  ou en navigant sur la gauche.

Comment tester votre code en ligne ?

10

- Quelques fois, il est fastidieux de lancer tous les logiciels nécessaires pour tester un morceau de code ou vous êtes sur un ordinateur à la BU.
- www.w3schools.com (et bien d'autres) permet de tester votre code dans un navigateur :
 - Pur HTML : https://www.w3schools.com/html/tryit.asp?filename=tryhtml_default
 - CSS avec HTML : https://www.w3schools.com/css/tryit.asp?filename=trycss_default
 - Javascript : https://www.w3schools.com/js/tryit.asp?filename=tryjs_myfirst
- L'inconvénient, c'est que l'HTML, le CSS et le Javascript se trouve dans le même fichier

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-color: lightblue;
}

h1 {
    color: white;
    text-align: center;
}

p {
    font-family: verdana;
    font-size: 20px;
}
</style>
</head>
<body>

<h1>My First CSS Example</h1>
<p>This is a paragraph.</p>

</body>
</html>
```


- Il y a d'autres sites (peut-être en connaissez-vous ?), qui proposent de travailler avec trois fichiers différents :
- <https://jsfiddle.net/>
- <https://liveweave.com/>
- et mes deux préférés :
- 1. <https://html-css-js.com/>
- 2. [Source : https://codepen.io/pen/](https://codepen.io/pen/)

HTMLCSSJS

EditorTagsCheatSheetCharactersGeneratorsTemplatesBlogTutorialLinks

SettingsIncludeRun JS

1<!DOCTYPE html>
2<html>
3<head>
4<title>HTML CSS JS</title>
5</head>
6<body>
7<h1 id="welcome">HTML CSS JS</h1>
8<p>Welcome to HTML-CSS-JS.com</p>
9<p>online HTML, CSS and JavaScript editor
10with instant preview.</p>
11</body>
12</html>

HTML

1// JavaScript
2document.getElementById('welcome').innerText +=
3" Editors";
4

JavaScript

1/* CSS styles */
2h1 {
3font-family: Impact, sans-serif;
4color: #E5937;
5}
6

CSS

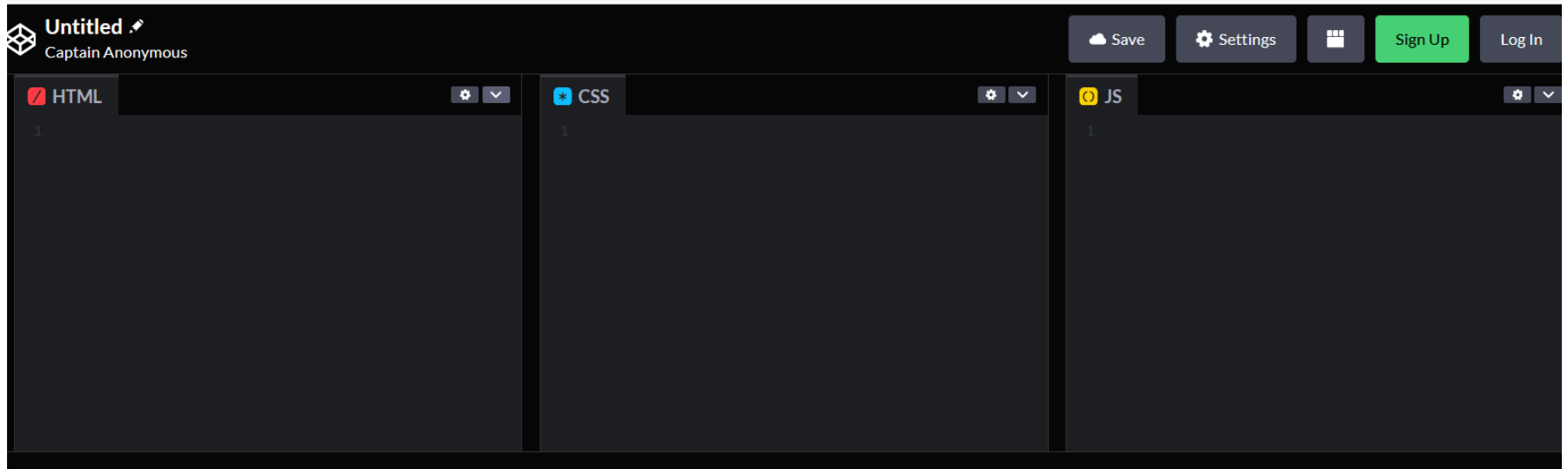
HTML CSS JS

Welcome to HTML-CSS-JS.com

Online HTML, CSS and JavaScript editor with instant preview.

Privacy

14



Liste des éditeurs pour HTML, CSS, et JavaScript

16

- **Éditeurs locaux pour débutants et professionnels**

- 1. Visual Studio Code (VS Code)
- 2. Sublime Text
- 3. Atom
- 4. JetBrains WebStorm

-

- **Plateformes en ligne**

- 5. CodePen
- 6. JSFiddle
- 7. JSBin

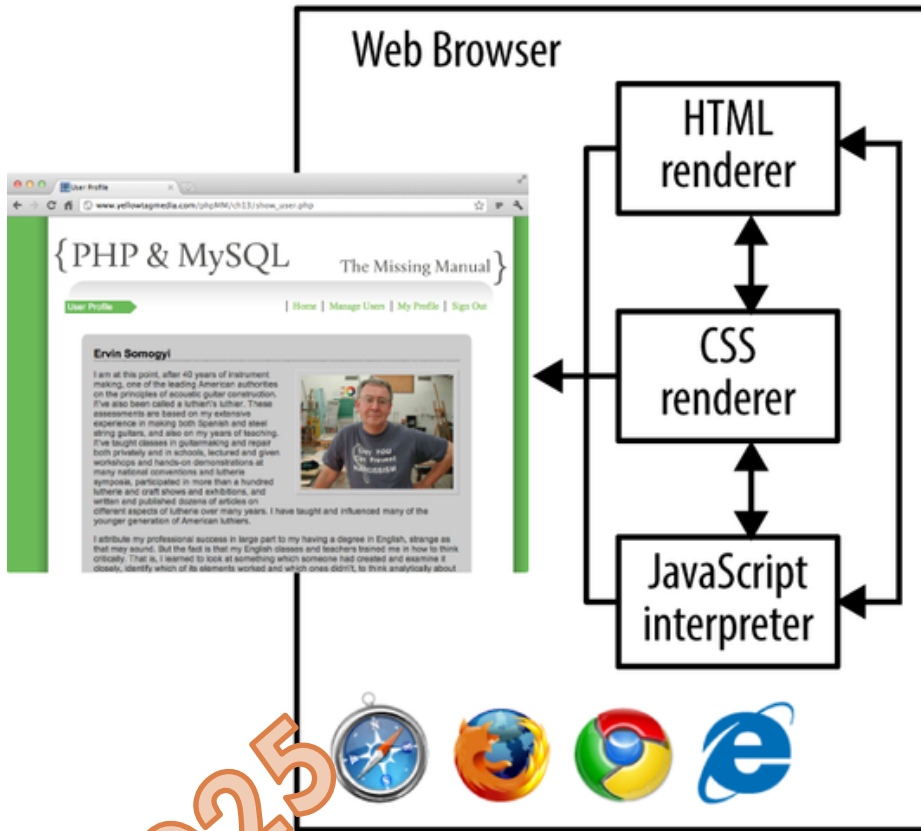
-

- **Alternatives légères et minimalistes**

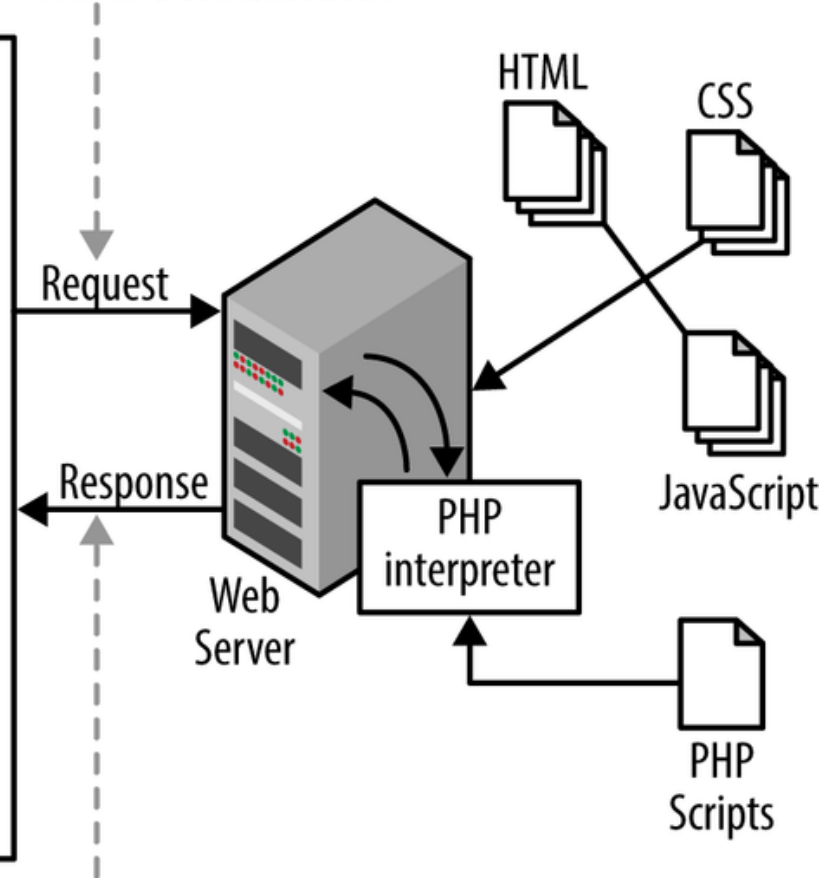
- 8. Brackets
- 9. Notepad++

1. Comment fonctionne un site web



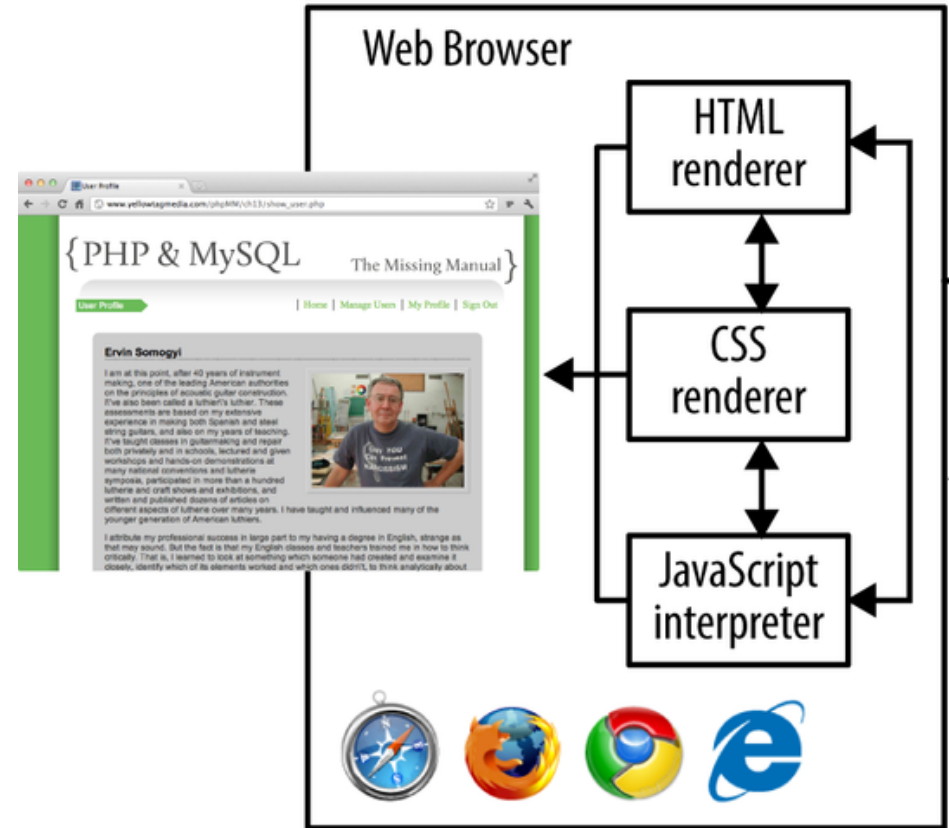


Could be for HTML, CSS, PHP or a combination.

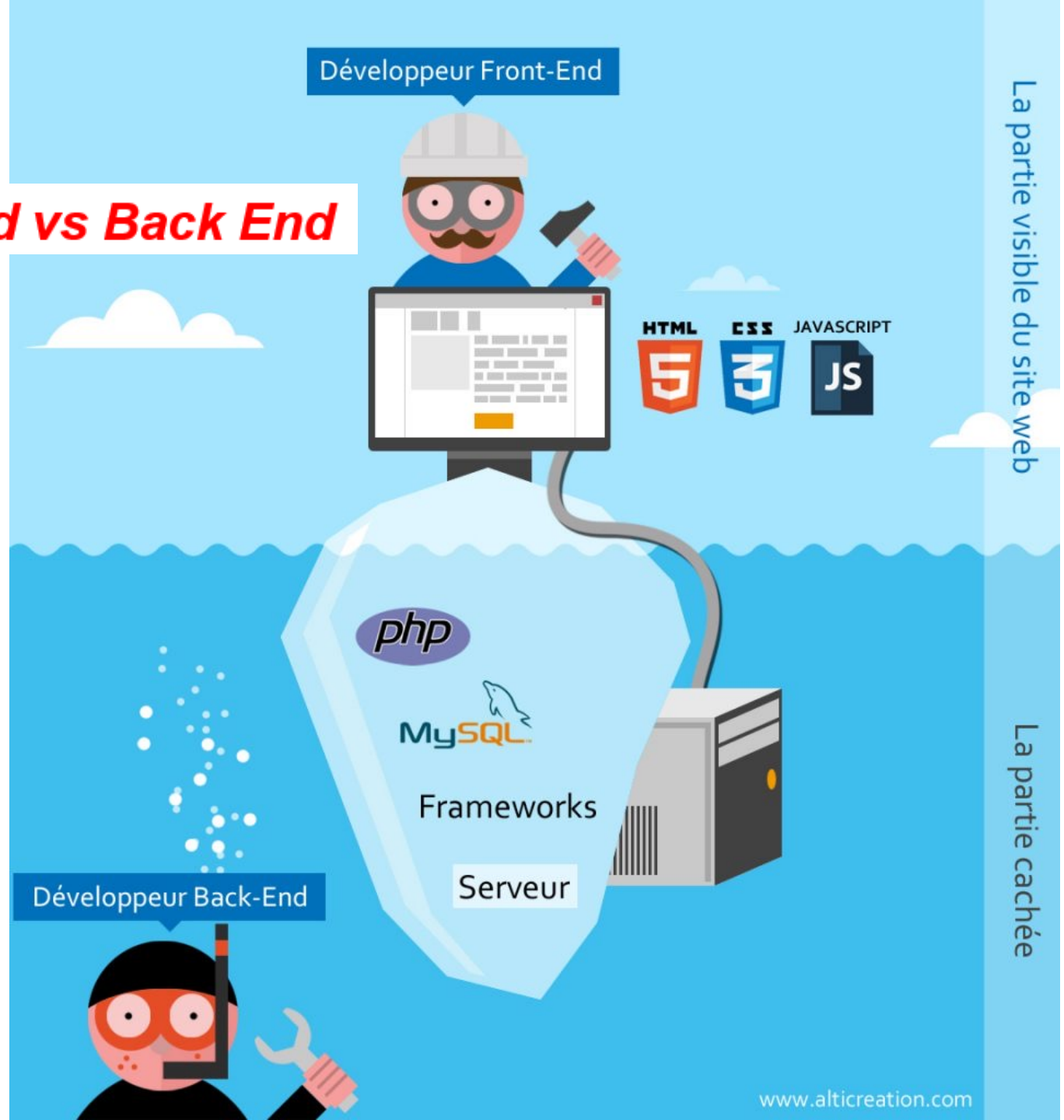


Response is not PHP, but the result of interpreting PHP, usually more HTML and CSS.

2025



2. *Front End vs Back End*



HTML

Pourquoi apprendre encore HTML en 2025

22

- ❑ Base fondamentale : Le socle de tout site web, utilisé par tous les frameworks modernes.
- ❑ Compatibilité universelle : Standard du web, supporté par tous les navigateurs.
- ❑ Accessibilité : Essentiel pour créer des sites inclusifs et accessibles à tous.
- ❑ Évolutions constantes : HTML5 offre des fonctionnalités modernes (balises sémantiques, multimédia).
- ❑ En bref : Une compétence intemporelle et essentielle pour des sites modernes, performants et inclusifs.

□ HTML (HyperText Markup Language)

- Hypertexte: texte + liens vers d'autres pages ou objets
- C'est un langage de balisage (<> </>) simple à utiliser
- Un document HTML définit:
 - La structure du texte (titres, paragraphes, sous titres, etc.)
 - Le contenu (images, vidéos, liens, etc.)
 - La mise en forme du texte (couleur, taille, alignement, etc.)

□ **HTML (HyperText Markup Language)**

- Le navigateur Web est chargé d'interpréter le code HTML et le présenter aux visiteurs du site
 - La présentation d'un document HTML diffère d'un navigateur à un autre
 - Elle diffère aussi d'une version à une autre du même navigateur
 - Il faut tester les pages Web sur plusieurs navigateurs

□ **HTML (HyperText Markup Language)**

- ▣ Ce n'est pas un langage proprement dit car on n'a pas de programmes créés
- ▣ C'est un langage multi-systèmes (Windows, Linux, Mac, etc.) et multi-plateformes (Explorer, Chrome, Firefox, etc.)
- ▣ Il permet de créer une page Web et assurer sa mise en forme
- ▣ À partir de sa version 4 (1997), la mise en forme est gérée par des feuilles de styles (fichiers CSS)

Structure d'un document HTML

Structure d'un document HTML

27

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
  <HEAD>
```

```
    <meta charset="utf-8"/>
```

```
    <meta name="author" content="auteur"/>
```

```
    <TITLE>Hello world!!</TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    Un premier exemple tout simple!!
```

```
  </BODY>
```

```
</HTML>
```

Structure d'un document HTML

28

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
  <HEAD>
```

```
    <meta charset="utf-8"/>
```

```
    <meta name="author" content="auteur"/>
```

```
    <TITLE>Hello world!!</TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    Un premier exemple tout simple!!
```

```
  </BODY>
```

```
</HTML>
```

Structure d'un document HTML

29

- Le code HTML est écrit sous forme de balises (tags).
- Une balise doit être ouverte et fermée.
- La fermeture d'une balise est identique à son ouverture sauf qu'elle comprend un slash « / » avant le nom de la balise.
 - ▣ **<balise> texte affecté par la balise </balise>**
 - ▣ Exemple:
 - `<title>Cours Introduction au développement WEB</title>`

Structure d'un document HTML

30

- Une balise peut être écrite en minuscule ou majuscule.
 - ▣ `<HTML>`, `<html>` et `<HtmL>` sont équivalentes
- Une balise peut comporter des attributs ayant des valeurs mises entre guillemets.
 - ▣ Exemple: `<meta name="author" content="auteur"/>`
- Les sauts de lignes et espaces ne sont pas interprétés.

Structure d'un document HTML

31

Commentaires

- Pour écrire un commentaire dans le code source, il faut le placer entre « <!-- » et « --> ».
- Le commentaire n'est visible que dans le code source (invisible dans le navigateur).
 - ▣ Exemple: <!-- ceci est un commentaire -->

Structure d'un document HTML

32

`<!DOCTYPE HTML>`

`<HTML>`

`<HEAD>`

`<meta charset="utf-8"/>`

`<meta name="author" content="auteur"/>`

`<TITLE>Hello world!!</TITLE>`

`</HEAD>`

`<BODY>`

`Un premier exemple tout simple!!`

`</BODY>`

`</HTML>`

Structure d'un document HTML

33

Balise DOCTYPE

- C'est une balise indispensable écrite dans la première ligne du document.
 - ▣ Elle commence par un point d'exclamation
 - ▣ Elle informe sur le type de document (doctype) traité par le navigateur Web
- La version 5 du HTML a simplifié l'écriture de cette balise.
 - ▣ Exemple: `<!DOCTYPE HTML>`

Structure d'un document HTML

34

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
<HEAD>
```

```
<meta charset="utf-8"/>
```

```
<meta name="author" content="auteur"/>
```

```
<TITLE>Hello world!!</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Un premier exemple tout simple!!
```

```
</BODY>
```

```
</HTML>
```

Structure d'un document HTML

35

Balise HTML

- C'est l'élément racine du code d'une page Web.
- C'est la balise principale regroupant tout le code écrit.

- ▣ Elle doit être fermée à la fin du code

- ▣ Exemple:

```
<HTML>
```

il faut mettre dedans tout le code

...

```
</HTML>
```

Structure d'un document HTML

36

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
<HEAD>
```

```
<meta charset="utf-8"/>
```

```
<meta name="author" content="auteur"/>
```

```
<TITLE>Hello world!!</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Un premier exemple tout simple!!
```

```
</BODY>
```

```
</HTML>
```

Structure d'un document HTML

37

Balise HEAD

- Il s'agit de l'entête du document.
- La balise comprend des informations générales du document (titre, auteur, encodage, etc.).
 - ▣ Ces informations ne seront pas affichées dans la page Web
 - ▣ Exemple:
<HEAD>
définir le titre, auteur, etc.
</HEAD>

Structure d'un document HTML

38

Balise HEAD

- Elle fait appel aux scripts et feuilles de style.
 - ▣ Exemples: `<script>`, `<style>`, `<link>`

- Elle comprend aussi les métadonnées du document (informations descriptives sur l'auteur, l'encodage, etc.)
 - ▣ Exemple: `<meta>`

Structure d'un document HTML

39

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
  <HEAD>
```

```
    <meta charset="utf-8"/>
```

```
    <meta name="author" content="auteur"/>
```

```
    <TITLE>Hello world!!</TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    Un premier exemple tout simple!!
```

```
  </BODY>
```

```
</HTML>
```

Structure d'un document HTML

40

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
  <HEAD>
```

```
    <meta charset="utf-8" />
```

```
    <meta name="auteur" content="auteur" />
```

```
    <TITLE>Hello world!!</TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    Un premier exemple tout simple!!
```

```
  </BODY>
```

```
</HTML>
```


Structure d'un document HTML

41

Balise TITLE

- C'est l'élément le plus important de l'entête du document.
- Il définit le titre de la fenêtre ou de l'onglet dans le navigateur Web.
- Il n'apparaît pas dans le contenu de la page Web.
- Il apparaît dans les résultats de recherche.
 - ▣ Exemple: `<TITLE> Cours Web </TITLE>`

Structure d'un document HTML

42

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
  <HEAD>
```

```
    <meta charset="utf-8"/>
```

```
    <meta name="author" content="auteur"/>
```

```
    <TITLE>Hello world!!</TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    Un premier exemple tout simple!!
```

```
  </BODY>
```

```
</HTML>
```

Structure d'un document HTML

43

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
<HEAD>
```

```
<meta charset="utf-8"/>
```

```
<meta name="author" content="auteur"/>
```

```
<TITLE>Hello world!!</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
Un premier exemple tout simple!!
```

```
</BODY>
```

```
</HTML>
```

Structure d'un document HTML

44

Balise META

- Elle déclare les métadonnées du document.
- Ces informations ne seront pas affichées dans la page Web.
- Elles sont utiles pour:
 - ▣ Les navigateurs (encodage des caractères)
 - ▣ Les moteurs de recherche (mots clés)
- C'est une balise avec attributs mais sans corps.
 - ▣ Balise orpheline (<meta/>)

Structure d'un document HTML

45

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
  <HEAD>
```

```
    <meta charset="utf-8"/>
```

```
    <meta name="author" content="auteur"/>
```

```
    <TITLE>Hello world!!</TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

```
    Un premier exemple tout simple!!
```

```
  </BODY>
```

```
</HTML>
```

Structure d'un document HTML

46

```
<!DOCTYPE HTML>
```

```
<HTML>
```

```
  <HEAD>
```

```
    <meta charset="utf-8"/>
```

```
    <meta name="author" content="auteur"/>
```

```
    <TITLE>Hello world!!</TITLE>
```

```
  </HEAD>
```

```
  <BODY>
```

Un premier exemple tout simple!!

```
  </BODY>
```

```
</HTML>
```

Structure d'un document HTML

47

Balise BODY

- C'est le corps du document.
- Il est placé après l'entête.
- Il regroupe tout le code du contenu qui sera affiché dans la page.
- La balise est unique.
 - ▣ Exemple:
<BODY> tout le contenu à afficher </BODY>

- Test
- <https://codepen.io/rtomczak/pen/VwxPWPP?editors=1000>

Syntaxe - Texte

Texte préformaté

- La balise **<PRE>** est utilisée pour afficher un texte préformaté c'est-à-dire afficher un texte comme c'est écrit dans le document HTML.
 - ▣ Il doit respecter les espaces, tabulations, etc.
 - ▣ La balise n'est pas appliquée aux titres, paragraphes, etc.
 - ▣ Exemple: `<PRE> je voudrais bien que ce texte soit
séparé des espaces et affiché dans la page </PRE>`

Texte

51

Texte préformaté

- Avec/sans **<PRE>**

```
<body>
  <p>
    Les inscrits sont:
    Nom          Prénom    Groupe
    -----
    Dupont       Michel    1
    Mathieu      Henri     2
  </p>
  <pre>
    Les inscrits sont:
    Nom          Prénom    Groupe
    -----
    Dupont       Michel    1
    Mathieu      Henri     2
  </pre>
</body>
```

Texte x +

← → ↻ ⓘ Fichier | E:/enseignement 6202/1

Les inscrits sont: Nom Prénom Groupe ----- Dupont Michel 1 Mathieu Henri 2

| Nom | Prénom | Groupe |
|---------|--------|--------|
| Dupont | Michel | 1 |
| Mathieu | Henri | 2 |

Texte

52

Mise en forme

```
<p>
  <b> b: affiche le texte en gras </b> <br>
  <i> i: affiche le texte en italique </i>
</p>
<p>
  <u> u: souligne le texte </u>
</p>
<p>
  <s> s: barre le texte </s>
</p>
<p>
  <small> small: police plus petite </small>
</p>
<p>
  <em> em: indique une mise en exergue (italique) </em> <br>
  <strong> strong: faire ressortir du texte (gras) </strong> <br>
  <var> var: variable </var> <br>
  <dfn> dfn: définition </dfn>
</p>
```

b: affiche le texte en gras

i: affiche le texte en italique

u: souligne le texte

~~s: barre le texte~~

small: police plus petite

em: indique une mise en exergue (italique)

strong: faire ressortir du texte (gras)

var: variable

dfn: définition

Paragraphe

- Le contenu d'un paragraphe est délimité par les deux balises `<p>` et `</p>`.
- On ne peut pas imbriquer les balises `<p>`.
- Une ligne blanche est ajoutée par le navigateur avant et après le paragraphe.
 - ▣ Exemple: `<p>` contenu du paragraphe `</p>`

Paragraphe

- L'affichage du contenu d'un paragraphe dépend de la taille de la fenêtre du navigateur.
- ▣ Le retour à la ligne est géré automatiquement

```
<p>  
    La programmation web est la programmation  
    informatique qui permet d'éditer des sites web.  
    Elle permet la création d'applications destinées  
    à être déployées sur Internet ou Intranet.  
</p>
```

La programmation web est la programmation informatique qui permet d'éditer des sites web. Elle permet la création d'applications destinées à être déployées sur Internet ou Intranet.

Texte

55

Retour à la ligne

- La balise `
` (break) permet d'effectuer un retour à la ligne dans un paragraphe.
- ▣ C'est une balise ouvrante seulement

```
<p>  
    La programmation web est la programmation <br>  
    informatique qui permet d'éditer des sites web. <br>  
    Elle permet la création d'applications destinées <br>  
    à être déployées sur Internet ou Intranet.  
</p>
```

La programmation web est la programmation
informatique qui permet d'éditer des sites web.
Elle permet la création d'applications destinées
à être déployées sur Internet ou Intranet.

Texte

56

Titre

- Il existe 6 niveaux de titres (écriture en gras).
 - **<H1>**: le premier niveau ayant la plus grande taille
 - ...
 - **<H6>**: le dernier niveau ayant la plus petite taille

```
<h1> Titre de niveau 1 </h1>  
<h2> Titre de niveau 2 </h2>  
<h3> Titre de niveau 3 </h3>  
<h4> Titre de niveau 4 </h4>  
<h5> Titre de niveau 5 </h5>  
<h6> Titre de niveau 6 </h6>
```

Titre de niveau 1

Titre de niveau 2

Titre de niveau 3

Titre de niveau 4

Titre de niveau 5

Titre de niveau 6

Citation

- Il existe trois balises dédiées aux citations:
 - ▣ **<q>**: les citations courtes
 - ▣ **<blockquote>**: les citations longues
 - ▣ **<cite>**: source de citation

```
<p>  
  Un exemple de citation: <br>  
  <q> Ce qui se conçoit bien s'énonce clairement </q>  
  <cite> Boileau </cite>  
</p>
```

Un exemple de citation:
“ Ce qui se conçoit bien s'énonce clairement ” *Boileau*

Texte

58

Citation

- La balise **<blockquote>** ajoute un retrait.

```
<p>  
  Exemple sans blockquote: <br>  
  La programmation web est la programmation <br>  
  informatique qui permet d'éditer des sites web. <br>  
  Elle permet la création d'applications destinées <br>  
  à être déployées sur Internet ou Intranet. <br>  
  Exemple avec blockquote: <br>  
  <blockquote>  
  La programmation web est la programmation <br>  
  informatique qui permet d'éditer des sites web. <br>  
  Elle permet la création d'applications destinées <br>  
  à être déployées sur Internet ou Intranet.  
  </blockquote>  
</p>
```

Texte

59

Citation

- La balise **<blockquote>** ajoute un retrait.

Exemple sans blockquote:

La programmation web est la programmation informatique qui permet d'éditer des sites web. Elle permet la création d'applications destinées à être déployées sur Internet ou Intranet.

Exemple avec blockquote:

La programmation web est la programmation informatique qui permet d'éditer des sites web. Elle permet la création d'applications destinées à être déployées sur Internet ou Intranet.

Indices et exposants

- Les deux balises **<sub>** et **<sup>** permettent d'afficher un caractère respectivement comme indice et exposant.

```
<p>  
Afficher a comme indice de X: X<sub>a</sub> <br>  
Afficher b comme exposant de X: X<sup>b</sup> <br>  
Afficher a comme indice et b comme exposant de X: X<sub>a</sub><sup>b</sup>  
</p>
```

Afficher a comme indice de X: X_a

Afficher b comme exposant de X: X^b

Afficher a comme indice et b comme exposant de X: X_a^b

Texte

61

Caractères accentués

| Caractère | Code HTML | Caractère | Code HTML |
|-----------|-----------|-----------|-----------|
| Á | Á | à | à |
| À | À | ä | ä |
| Ã | Ã | é | é |
| É | É | ê | ê |
| È | È | ú | ú |
| Î | Î | ü | ü |
| Ö | Ö | ï | ï |
| Â | Â | ç | ç |
| Å | Å | â | â |
| Ä | Ä | æ | æ |
| Ê | Ê | è | è |
| Ë | Ë | ë | ë |
| Ô | Ô | û | û |
| Æ | Æ | ö | ö |
| Ç | Ç | î | î |
| | | ù | ù |

Texte

62

Caractères spéciaux

| Caractère | Code HTML |
|--|-----------|
| > | > |
| < | < |
| Espace insécable (non-breaking space) | |
| ® | ® |
| ™ | ™ |
| » | » |
| « | « |
| º | ° |
| ² | ² |
| § | § |
| ½ | ½ |

| Caractère | Code HTML |
|-----------|-----------|
| & | & |
| ¶ | ¶ |
| œ | œ |
| " | " |
| © | © |
| ± | ± |
| ¹ | ¹ |
| ³ | ³ |
| ¼ | ¼ |
| ¾ | ¾ |
| · | · |
| ‰ | ‰ |

Syntaxe - Listes

Listes

64

Liste ordonnée

- La balise **** (**ordered list**) permet de créer une liste ordonnée (numérotée).
 - ▣ La balise **** (**list item**) permet de créer un élément de la liste
 - ▣ Exemple: ****
**** item 1 ****
**** item 2 ****
....

Listes

65

Liste ordonnée

- L'attribut **type** permet de définir le type de numérotation utilisée (numérique ou alphabétique).
 - ▣ **type = "1"**: par défaut, utiliser des nombres (1, 2, ...)
 - ▣ **type = "A"**: utiliser des lettres en majuscule (A, B, C, ...)
 - ▣ **type = "a"**: utiliser des lettres en minuscule (a, b, c, ...)
 - ▣ **type = "I"**: utiliser des chiffres romains en majuscule (I, II, III, ...)
 - ▣ **type = "i"**: utiliser des chiffres romains en minuscule (i, ii, iii, ...)

Listes

66

Liste ordonnée

□ Exemples:

```
<ol>
  <li> Rouge </li>
  <li> Vert </li>
  <li> Bleu </li>
</ol>
```

1. Rouge
2. Vert
3. Bleu

```
<ol type="I">
  <li> Rouge </li>
  <li> Vert </li>
  <li> Bleu </li>
</ol>
```

I. Rouge
II. Vert
III. Bleu

```
<ol type="A">
  <li> Rouge </li>
  <li> Vert </li>
  <li> Bleu </li>
</ol>
```

A. Rouge
B. Vert
C. Bleu

```
<ol type="i">
  <li> Rouge </li>
  <li> Vert </li>
  <li> Bleu </li>
</ol>
```

i. Rouge
ii. Vert
iii. Bleu

```
<ol type="a">
  <li> Rouge </li>
  <li> Vert </li>
  <li> Bleu </li>
</ol>
```

a. Rouge
b. Vert
c. Bleu

Listes

67

Liste ordonnée

- L'attribut **start** permet de spécifier la valeur de départ d'une liste.
- Exemples:

```
<ol start="3">  
  <li> Rouge </li>  
  <li> Vert </li>  
  <li> Bleu </li>  
</ol>
```

```
3. Rouge  
4. Vert  
5. Bleu
```

```
<ol start="3" type="i">  
  <li> Rouge </li>  
  <li> Vert </li>  
  <li> Bleu </li>  
</ol>
```

```
iii. Rouge  
iv. Vert  
v. Bleu
```

Liste non ordonnée

- La balise **** (**unordered list**) permet de créer une liste non ordonnée (non numérotée utilisant des puces).
- L'attribut **type** permet de définir le type de puce utilisée.
 - ▣ **type = "disc"**: par défaut, un disque plein
 - ▣ **type = "circle"**: un cercle
 - ▣ **type = "square"**: un carré

Listes

69

Liste non ordonnée

□ Exemples:

```
<ul>
  <li> Rouge </li>
  <li> Vert </li>
  <li> Bleu </li>
</ul>
```

- Rouge
- Vert
- Bleu

```
<ul type="circle">
  <li> Rouge </li>
  <li> Vert </li>
  <li> Bleu </li>
</ul>
```

- Rouge
- Vert
- Bleu

```
<ul type="square">
  <li> Rouge </li>
  <li> Vert </li>
  <li> Bleu </li>
</ul>
```

- Rouge
- Vert
- Bleu

Liste de définitions

- La balise **<DL>** (**Description/Definition List**) permet de créer une liste de termes avec leurs définitions.
 - ▣ **<DT>** (**Definition Term**): l'élément à définir
 - ▣ **<DD>** (**Definition Description**): définition de l'élément
 - ▣ Exemple: **<DL>**
<DT> Terme 1 **</DT>**
<DD> Définition 1 **</DD>**
<DD> Définition 2 **</DD>** ... **</DL>**

Listes

71

Liste de définitions

□ Exemple:

```
<dl>
  <dt> Étoile </dt>
  <dd> Objet céleste étudié par les astrophysiciens.</dd>
  <dd> Danseur confirmé d'un ballet. </dd>
  <dt> Liste </dt>
  <dd> Ensemble d'éléments. </dd>
</dl>
```

Étoile

Objet céleste étudié par les astrophysiciens.
Danseur confirmé d'un ballet.

Liste

Ensemble d'éléments.

Listes

72

Listes imbriquées

- On peut imbriquer les listes ordonnées et non ordonnées.
- Exemple:

```
<ol type="A">  
  <li> Premier élément </li>  
  <ul type="circle">  
    <li> Début du premier élément </li>  
    <li> Suite du premier élément </li>  
  </ul>  
  <li> Deuxième élément </li>  
</ol>
```

- A. Premier élément
 - Début du premier élément
 - Suite du premier élément
- B. Deuxième élément

Syntaxe - Liens

Liens

74

- La balise `<a>` permet de créer un lien hypertexte.
- Un lien est un outil de navigation entre pages ou objets situés dans le même dossier ou dans des dossiers différents.
 - ▣ Exemple:

```
<a href = "pageCible.html"> Si on clique sur ce lien,  
| on passera à la page pageCible </a>
```

Si on clique sur ce lien, on passera à la page pageCible

Liens

75

- Un lien non visité est souligné et coloré en bleu.
- Un lien visité est souligné et coloré en violet.
- Un lien actif est souligné et coloré en rouge.
- Les attributs de la balise **<a>** sont principalement:
 - **href**: spécifier la source du document cible
 - **id**: attribuer un identifiant unique à une ancre dans un document
 - **download**: télécharger un document
 - **target**: spécifier le type d'affichage du document
 - **title**: décrire un lien en affichant une infobulle

Liens

76

Lien externe

- Il permet d'accéder à un autre site Web.
- L'attribut **href** comprend l'adresse complète du site ou du fichier ainsi que le protocole utilisé.
 - ▣ Exemples:

```
<a href = "https://www.google.com"> Accéder à Google </a>  
<a href = "ftp://ftp.loria.fr"> Accéder au serveur ftp </a>
```

[Accéder à Google](https://www.google.com)
[Accéder au serveur ftp](ftp://ftp.loria.fr)

Lien interne au site

- Il pointe vers un autre document (page Web) situé dans le même répertoire ou dans un répertoire différent.
- ▣ Exemples:
 - `` Accéder à la page (page2) du même répertoire courant ``
 - `` Accéder à la page (page3) du répertoire (dossier) ``
 - `` Accéder à la page (page4) du répertoire parent ``

Lien interne à la page

- Il permet de pointer vers une partie d'un document (titre, section, définition, etc.).
 - ▣ Il s'agit d'un lien vers une ancre
- Deux étapes sont nécessaires:
 - ▣ Attribuer un identifiant à une ancre avec l'attribut **id**
 - Exemple: ` texte ciblé `
 - ▣ Créer un lien vers l'ancre avec l'attribut **href**
 - Exemple: ` texte du lien `

Liens

79

Lien interne à la page

```
<html>
  <body>
    <h1><a id="top">Titre de niveau 1</a></h1>
    <p>Paragraphe 1</p>
    <p>Paragraphe 2</p>
    <p>Paragraphe 3</p>
    ...
    <a href="#top">Aller vers le haut</a>
  </body>
</html>
```

Lien interne à la page

- Un lien peut aussi pointer vers une ancre dans une autre page du même répertoire ou d'un répertoire différent.

- ▣ Exemples:

- `` Pointer vers une ancre dans la page (page2) du même répertoire ``
- `` Pointer vers une ancre dans la page (page3) du répertoire dossier ``

Liens

81

Lien de téléchargement

- L'attribut **download** permet de télécharger un fichier (compressé, image, document pdf, vidéo, etc.)
 - ▣ Exemples:

```
<a href = "archive.rar" download> Cliquer pour télécharger  
le fichier compressé </a>
```

```
<a href = "test.docx" download> Cliquer pour télécharger  
le document Word </a>
```

```
<a href = "classeur.xlsx" download> Cliquer pour télécharger  
le fichier Excel </a>
```

Liens

82

Lien de téléchargement

▣ Exemples:

```
<a href = "document.pdf" download> Cliquer pour télécharger  
| le fichier pdf </a>  
<!--le document sera affiché dans le navigateur et peut  
| être ensuite téléchargé-->  
<a href = "son.mp3" download> Cliquer pour télécharger  
| le fichier mp3 </a>  
<!--le fichier sera joué dans le navigateur et peut  
| être ensuite téléchargé-->  
<a href = "video.mp4" download> Cliquer pour télécharger  
| le fichier mp4 </a>  
<!--le fichier sera joué dans le navigateur et peut être  
| ensuite téléchargé-->
```

Liens

83

Lien de messagerie

- Il est créé en utilisant une URI dont le schéma est **mailto** et l'information est l'adresse mail dans l'attribut **href** de la balise `<a>`.

- ▣ Exemples:

```
<a href = "mailto: x.y@gmail.com"> cliquer pour envoyer  
| un e-mail </a>  
<!--lancer une application de messagerie installée avec  
| un nouveau message vide-->  
  
<a href = "mailto: x.y@gmail.com?subject=message">  
| cliquer pour envoyer un e-mail ayant l'objet (message) </a>
```

Liens

84

Lien de messagerie

▣ Exemples:

```
<a href = "mailto: x.y@gmail.com?subject=message&body=bonjour">  
    cliquer pour envoyer un e-mail ayant l'objet (message) et  
    le corps (bonjour) </a>
```

```
<a href = "mailto: x.y@gmail.com?subject=message&body=bonjour&cc=y.x@gmail.com">  
    cliquer pour envoyer un e-mail ayant l'objet (message) et le corps  
    (bonjour) à deux destinataires </a>
```

Liens

85

Lien ouvrant une nouvelle fenêtre

- L'attribut **target** permet d'ouvrir un lien dans une nouvelle fenêtre (onglet).

- ▣ Exemples:

```
<a href = "https://www.google.com" target = "_blank"> cliquer pour ouvrir  
| le lien dans une autre fenêtre </a>
```

```
<a href = "https://www.google.com" target = "_top"> cliquer pour ouvrir  
| le lien dans la totalité de la fenêtre courante </a>
```

Liens

86

Lien affichant une infobulle

- L'attribut **title** permettra d'afficher une bulle d'aide si on met le curseur sur un lien.
 - ▣ Exemple:

```
<a href = "https://www.google.com" title = "Accéder à Google">  
| cliquer pour passer à Google </a>
```

[cliquer pour passer à Google](https://www.google.com)

Accéder à Google

Syntaxe - Images

Images

88

- La balise `` permet d'insérer une image dans un document HTML.
 - ▣ Balise orpheline (avec attributs mais sans corps)
- La balise doit être mise dans un paragraphe.
- Les deux attributs indispensables sont:
 - ▣ **src**: spécifier la source du fichier au format (JPEG, PNG, GIF, etc.) avec des chemins absolus ou relatifs
 - ▣ **alt**: décrire l'image (un texte alternatif qui sera affiché si l'image n'a pas été chargée)

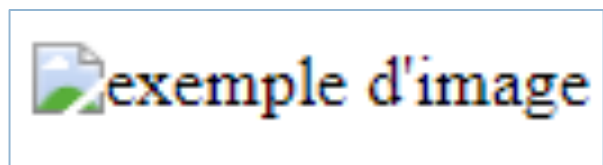
Images

89

▣ Exemple:

****:
ajouter l'image « image.png » dans la page Web.

- Si l'image n'a pas pu être chargée, le texte « exemple d'image » sera affiché à sa place



□ Il existe d'autres attributs de la balise :

- ▣ **width, height**: spécifier la taille (largeur et hauteur) de l'image en pixels
- ▣ **title**: afficher une infobulle (bulle d'aide)

Images

90

Attributs width et height

▣ Exemples:

```
<img src = "image.png" alt = "exemple d'image"  
width = "100px" height = "50px" />
```



```
<img src = "image.png" alt = "exemple d'image"  
width = "300px" height = "90px" />
```



Images

91

Attribut title

- C'est le même attribut que celui des liens.
 - ▣ Une bulle d'aide sera affichée si on met le curseur sur l'image
 - ▣ Exemple:

```
<img src = "image.png" alt = "exemple d'image"  
title = "image en format PNG" />
```



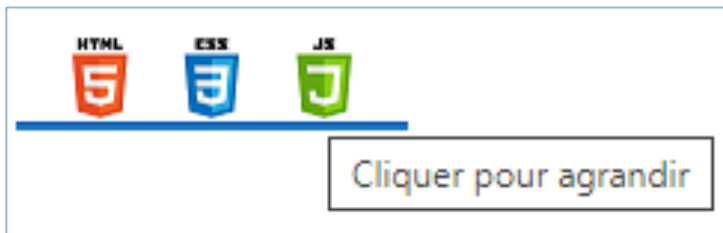
Images

92

Image miniaturisée cliquable

- Si les dimensions de l'image sont assez grandes, on peut mettre une image miniaturisée dans la page qui pointe vers l'image de taille réelle (originale).

```
<a href="image.png">  
    
</a>
```



Syntaxe – Audio

Fin S1

Audio

94

- La balise **<audio>** permet d'insérer un flux audio (mp3, wav et ogg) dans un document HTML.
- C'est une nouvelle balise de la version 5 du HTML.
 - ▣ Exemple:

```
<audio src = "son.mp3"> Votre navigateur ne peut pas  
| supporter ce format audio </audio>
```

- Ce message (texte alternatif) sera affiché si le format du flux audio n'est pas pris en charge par le navigateur

Audio

95

- Les attributs de la balise **<audio>** sont essentiellement:
 - ▣ **src**: spécifier le fichier audio à insérer
 - ▣ **controls**: afficher les boutons de contrôle (play, stop, volume, etc.)
 - ▣ **autoplay**: lire le fichier au chargement de la page
 - ▣ **loop**: répéter la lecture du fichier audio
 - ▣ **muted**: couper le son initialement

Attribut src

- Il définit la source du flux audio.
- On peut spécifier des chemins absolus ou relatifs pour la source.
 - ▣ Exemple:
 - **<audio src = "son.mp3"> Mettre à jour le navigateur </audio>**: aucun contenu ne sera affiché sur la page Web.
 - Le navigateur récupère les informations relatives au fichier audio et n'affiche rien

Audio

97

Attribut controls

- Il permet d'afficher le menu de lecture audio (temps, volume, bouton play/stop, etc.).
 - ▣ Exemple:

```
<audio src = "son.mp3" controls> Mettre  
| à jour le navigateur </audio>
```

- L'affichage diffère d'un navigateur à un autre



Audio

98

Attribut autoplay

- Il permet de lire le contenu du flux audio une fois la page Web chargée.
- Cette option est bloquée par Chrome (javascript comme solution), Firefox et Internet Explorer (activer l'autorisation de lecture automatique).
 - ▣ Exemple:

```
<audio src = "son.mp3" controls autoplay> Mettre  
| à jour le navigateur </audio>
```

Audio

99

Attributs loop et muted

- L'attribut **loop** assure une relecture automatique du flux audio.
- L'attribut **muted** permet de couper le son au chargement de la page Web.
 - ▣ Exemples:

```
<audio src = "son.mp3" controls loop>  
  Mettre à jour le navigateur </audio>  
  
<audio src = "son.mp3" controls muted>  
  Mettre à jour le navigateur </audio>
```



Audio

100

Formats audio multiples

- Afin d'éviter la non prise en charge du flux audio par un navigateur, on peut définir plusieurs formats du même fichier audio avec la balise **<source>**.
- Le navigateur choisira le format le mieux adapté.
 - ▣ Exemple:

```
<audio controls>  
  <source src = "son.mp3"/>  
  <source src = "son.wav"/>  
  Mettre à jour le navigateur </audio>
```

Syntaxe - Vidéo

Vidéo

102

- La balise **<video>** permet d'insérer une vidéo (mp4, webM, ogg, etc.) dans un document HTML.
- C'est une nouvelle balise de la version 5 du HTML.
 - ▣ Exemple:

```
<video src = "video.mp4"> Votre navigateur  
ne peut pas supporter ce format vidéo </video>
```

- Une image de la vidéo sera affichée sur la page Web
- Le corps de la balise sera affiché si la vidéo ne peut pas être chargée

Vidéo

103

- Les attributs de la balise **<video>** sont:
 - ▣ **src**: spécifier le fichier source de la vidéo
 - ▣ **poster**: spécifier une image à insérer si la vidéo n'est pas chargée
 - ▣ **controls**: afficher le menu de contrôle de lecture
 - ▣ **autoplay**: lire automatiquement la vidéo dès le chargement de la page
 - ▣ **loop**: relire la vidéo automatiquement
 - ▣ **width, height**: modifier les dimensions de la vidéo
 - ▣ **muted**: couper le son

Vidéo

104

Attribut poster

- Il permet d'afficher une image représentative de la vidéo.
 - ▣ Par défaut, c'est la première image de la vidéo qui sera affichée
 - ▣ Exemple:

```
<video src = "video.mp4" poster = "image.png">  
Votre navigateur ne peut pas supporter ce format vidéo </video>
```

- L'image « image.png » sera affichée sur la page Web

Vidéo

105

Attribut controls

- Il permet d'afficher le menu de lecture multimédia.
 - ▣ Exemple:

```
<video src = "video.mp4" poster = "image.png" controls>  
|   Votre navigateur ne peut pas supporter ce format vidéo  
</video>
```



Vidéo

106

Attribut autoplay

- Il permet de lire automatiquement le contenu d'une vidéo dès le chargement de la page Web.
 - ▣ Autorisation et javascript pour les navigateurs qui bloquent la lecture automatique
 - ▣ Exemple:

```
<video src = "video.mp4" poster = "image.png" controls autoplay>  
  Votre navigateur ne peut pas supporter ce format vidéo  
</video>
```

Vidéo

107

Attribut loop

- Il permet de relire le contenu d'une vidéo une fois la lecture en cours finie.
 - ▣ Exemple:

```
<video src = "video.mp4" poster = "image.png" controls loop>  
|   Votre navigateur ne peut pas supporter ce format vidéo  
</video>
```

Vidéo

108

Attribut muted

- Il permet de couper le son dès le chargement de la page Web.
- ▣ Exemple:

```
<video src = "video.mp4" poster = "image.png" controls muted>  
| Votre navigateur ne peut pas supporter ce format vidéo  
</video>
```



Vidéo

109

Attributs width et height

- Ils permettent de spécifier une taille fixe (largeur et hauteur) du lecteur vidéo en pixels.
 - ▣ Exemple:

```
<video src = "video.mp4" poster = "image.png" controls  
width = "500" height = "250" >  
| | Votre navigateur ne peut pas supporter ce format vidéo  
</video>
```

Vidéo

110

Formats vidéo multiples

- On peut proposer plusieurs formats du même fichier vidéo au navigateur qui utilisera le format qu'il peut prendre en charge.
 - ▣ Exemple:

```
<video controls>  
  <source src = "video.mp4"/>  
  <source src = "video.ogv"/>  
  Votre navigateur ne peut pas supporter ce format vidéo  
</video>
```

Syntaxe - Tableau

Tableau

112

- La balise **<table>** permet de créer un tableau dans un document HTML.
- Un tableau est constitué d'un ensemble de lignes et de cellules par lignes.
 - ▣ Les lignes sont ajoutées avec la balise **<tr>** (table row)
 - ▣ Les cellules sont ajoutées avec la balise **<td>** (table data)
- Le contenu des cellules peut être du texte, des images, sons, liens, etc.

Tableau

113

- Par défaut, les cellules sont affichées sans bordures.
 - ▣ L'attribut **border** permet d'ajouter des bordures au tableau avec ses cellules
 - ▣ Exemples:

```
<table>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
</table>
```

| | |
|-----------|-----------|
| Cellule 1 | Cellule 2 |
| Cellule 3 | Cellule 4 |

```
<table border="1">
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
</table>
```

| | |
|-----------|-----------|
| Cellule 1 | Cellule 2 |
| Cellule 3 | Cellule 4 |

Tableau

114

Titre d'un tableau

- ❑ La balise **<caption>** permet de définir le titre d'un tableau.
- ❑ Elle doit être insérée après la balise **<table>**.
- ❑ Par défaut, le titre est centré et écrit au dessus du tableau.
- ❑ On peut modifier sa position et son style à l'aide des fichiers de style CSS.

Tableau

115

Titre d'un tableau

▣ Exemple:

```
<table border>
  <caption>Exemple d'un tableau</caption>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
</table>
```

| | |
|-----------|-----------|
| Cellule 1 | Celulle 2 |
| Cellule 3 | Cellule 4 |

Tableau

116

Titre de cellules

- On distingue deux catégories de cellules:
 - ▣ Cellules de données comportant les informations du tableau
 - Écriture régulière et alignée à gauche par défaut
 - ▣ Cellules d'entête permettant de définir les titres de cellules
 - Écriture en gras et centrée par défaut

Tableau

117

Titre de cellules

- La balise `<th>` (table header) permet de spécifier le titre de cellules d'un tableau.

- Exemple:

| Titre 1 | Titre 2 |
|-----------|-----------|
| Cellule 1 | Celulle 2 |
| Cellule 3 | Cellule 4 |

```
<table border>
  <caption>Exemple d'un tableau</caption>
  <tr>
    <th>Titre 1</th>
    <th>Titre 2</th>
  </tr>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
</table>
```

Tableau

118

Fusion de cellules

- L'attribut **colspan** permet de fusionner des cellules d'une ligne d'un tableau.
- Il prend comme valeur le nombre de cellules à fusionner.
- Il peut être appliqué aux deux catégories de cellules:
 - ▣ Cellules d'entête (balise **<th>**)
 - ▣ Cellules de données (balise **<td>**)

Tableau

119

Fusion de cellules d'entête

■ Exemple:

```
<table border>
  <caption>Exemple d'un tableau</caption>
  <tr>
    <th colspan="2">Fusion du titre</th>
  </tr>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
</table>
```

| Fusion du Titre | |
|-----------------|-----------|
| Cellule 1 | Celulle 2 |
| Cellule 3 | Cellule 4 |

Tableau

120

Fusion de cellules de données

■ Exemple:

| Titre 1 | Titre 2 |
|-----------|-----------|
| Cellule 1 | Cellule 2 |
| Cellule 3 | Cellule 4 |
| Cellule 5 | |

```
<table border>
  <caption>Exemple d'un tableau</caption>
  <tr>
    <th>Titre 1</th>
    <th>Titre 2</th>
  </tr>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td>Cellule 4</td>
  </tr>
  <tr>
    <td colspan="2">Cellule 5</td>
  </tr>
</table>
```


Tableau

121

Fusion de lignes

- L'attribut **rowspan** permet de fusionner des lignes d'un tableau.
- Il prend comme valeur le nombre de lignes à fusionner.
- Il peut être appliqué aux deux catégories de cellules:
 - ▣ Cellules d'entête (balise **<th>**)
 - ▣ Cellules de données (balise **<td>**)

Tableau

122

Fusion de lignes

■ Exemple:

| Titre 1 | Titre 2 |
|-----------|-----------|
| Cellule 1 | Cellule 2 |
| Cellule 3 | Cellule 4 |
| Cellule 5 | |

```
<table border>
  <caption>Exemple d'un tableau</caption>
  <tr>
    <th>Titre 1</th>
    <th>Titre 2</th>
  </tr>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
  <tr>
    <td>Cellule 3</td>
    <td rowspan="2">Cellule 4</td>
  </tr>
  <tr>
    <td>Cellule 5</td>
  </tr>
</table>
```

Syntaxe - Formulaire

Formulaire

130

- ❑ Les formulaires sont utilisés pour saisir des informations, choisir une option, soumettre une demande, etc.
- ❑ La balise **<form>** permet de créer un formulaire dans un document HTML.
- ❑ Les éléments du formulaire doivent être ajoutés chacun dans un paragraphe pour inclure du texte.
- ❑ Les deux attributs indispensables de la balise **<form>** sont **method** et **action**.

Formulaire

131

- Un formulaire peut comporter des éléments comme:
 - ▣ Zones de texte
 - ▣ Cases à cocher
 - ▣ Boutons radios
 - ▣ Listes déroulantes
 - ▣ Etc.
- Les balises **<input>**, **<textarea>** et **<select>** permettent d'ajouter ces éléments dans une page HTML.

Balise `<input>` (orpheline)

- C'est la balise la plus utilisée permettant d'ajouter des éléments interactifs aux formulaires.
- Les attributs de cette balise sont essentiellement:
 - ▣ **name**: attribuer un nom à l'élément
 - ▣ **value**: attribuer une valeur par défaut à l'élément
 - ▣ **type**: déterminer le type de l'élément (champ de saisie, case à cocher, bouton radio, etc.)
 - ▣ **placeholder**: afficher un texte descriptif de l'élément

Formulaire

133

Balise `<input>` - attributs `name` et `value`

- L'attribut **name** est indispensable permettant d'identifier l'élément source de l'information (login et mot de passe saisis, par exemple).
- L'attribut **value** permet d'attribuer une valeur par défaut à l'élément (zone de texte, par exemple).
 - Exemple:
 - attribuer « login » et « user » respectivement comme nom et valeur par défaut à l'élément (zone de texte)

```
<input name="login" value="user"/>
```

Formulaire

134

Balise `<input>` - attribut `placeholder`

- L'attribut **placeholder** permet de spécifier une indication sur le contenu de la zone.
 - ▣ L'indication disparaît une fois l'utilisateur clique sur la zone
 - ▣ Exemple:

```
Votre nom: <input name="login" placeholder="Antoine"/>
```

Votre nom:

Balise `<input>` - attribut **type**

- L'attribut **type** détermine le type de l'élément et peut prendre comme valeurs:
 - ▣ **text**: pour les zones de texte
 - ▣ **checkbox**: pour les cases à cocher
 - ▣ **radio**: pour les boutons radios
 - ▣ **password**: pour les zones de mots de passe
 - ▣ **button**: pour les boutons
 - ▣ Etc.
- Il existe d'autres valeurs apparues avec HTML 5: **date**, **color**, **tel**, **email**, etc.

Formulaire

136

Balise `<input>` - attribut `type`

- La valeur **text** permet de créer une zone de texte.

- Exemple:

```
<p>  
  Votre nom: <input type="text" name="login" value="Antoine"/>  
</p>
```

Votre nom:

- Si aucun `type` n'est spécifié, une zone de texte sera créée par défaut

Formulaire

137

Balise `<input>` - attribut `type`

- La valeur **checkbox** permet de créer une case à cocher.
 - Aucune case n'est cochée par défaut
 - L'attribut **checked** permet de la cocher
 - Exemple:

```
Quels sont vos intérêts? <br>
<input type="checkbox" name="choix" value="c1"/> Sports <br>
<input type="checkbox" name="choix" value="c2"/> Peinture <br>
<input type="checkbox" name="choix" value="c3"/> Musique <br>
<input type="checkbox" name="choix" value="c4" checked="" /> Voyages
```

Quels sont vos intérêts?

☐ Sports

☐ Peinture

☐ Musique

☒ Voyages

Formulaire

138

Balise `<input>` - attribut `type`

- La valeur **radio** permet de créer un bouton radio.
 - ▣ Choix unique parmi plusieurs possibles
 - ▣ Tous les boutons doivent avoir la même valeur de l'attribut **name**
 - ▣ Aucun bouton radio n'est sélectionné par défaut
 - L'attribut **checked** permet de le sélectionner

Quel est votre statut?

☐ Maitre Assistant

☐ Maitre de conférence

☒ Professeur

```
Quel est votre statut?<br>
<input type="radio" name="statut" value="MA"/>Maitre Assitant <br>
<input type="radio" name="statut" value="MC"/>Maitre de conférence <br>
<input type="radio" name="statut" value="Pr" checked>Professeur </input>
```

Formulaire

139

Balise `<input>` - attribut `type`

- La valeur **password** permet de créer un champ de saisie dont les caractères sont masqués par des astérisques ou des points.
 - ▣ Exemple:

```
Mot de passe: <input type="password" name="passw" value="secret"/>
```

Mot de passe:

Formulaire

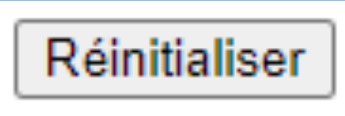
140

Balise `<input>` - attribut `type`

- La valeur **reset** permet d'ajouter un bouton qui réinitialise les champs du formulaire et les remet à leurs valeurs initiales.

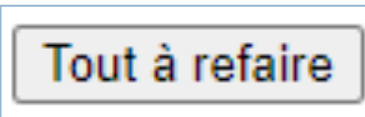
- ▣ Exemples:

```
<input type="reset" name="breset"/>
```



Réinitialiser

```
<input type="reset" name="breset" value="Tout à refaire"/>
```



Tout à refaire

Formulaire

141

Balise `<input>` - attribut `type`

- La valeur **submit** permet d'ajouter un bouton qui assure l'émission des données du formulaire à une autre page de traitement déclarée par l'attribut **action** de la balise `<form>`.

- Exemples:

```
<input type="submit" name="bsubmit"/>
```

Envoyer

```
<input type="submit" name="bsubmit" value="Transférer"/>
```

Transférer

Formulaire

142

Balise `<input>` - attribut `type`

- La valeur **button** permet d'ajouter un bouton qui appelle une fonction javascript par l'attribut **onclick**.

▣ Exemple:

```
<input type="button" name="bouton" value="Cliquez ici!" onclick="msg()"/>
```

```
<script>  
    function msg(){  
        alert("Bonjour");  
    }  
</script>
```

Cliquez ici!

Cette page indique
Bonjour

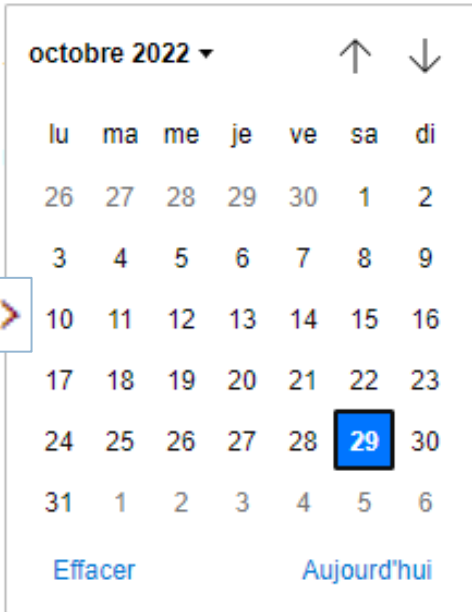
OK

Formulaire

143

Balise `<input>` - attribut `type`

- La valeur **date** permet d'ajouter un sélecteur de date.
 - ▣ La valeur renvoyée inclut le jour, le mois et l'année
 - ▣ Exemple:



The image shows a date picker interface. At the top, it displays 'octobre 2022' with a dropdown arrow and navigation arrows. Below is a calendar grid with days of the week (lu, ma, me, je, ve, sa, di) and dates. The date '29' is highlighted in a blue box. At the bottom, there are two buttons: 'Effacer' and 'Aujourd'hui'. Below the calendar, there is a text input field with the placeholder 'jj/mm/aaaa' and a calendar icon.

Date de naissance: `<input type="date" name="dn"/>`

Date de naissance:

Formulaire

144

Balise `<input>` - attribut `type`

- L'attribut **type** peut prendre d'autres valeurs comme:
 - **image**: pour ajouter une image comme bouton submit
 - **color**: pour ajouter un sélecteur de couleurs
 - **search**: pour effectuer une recherche sur Internet
 - **number**: pour saisir un nombre
 - **tel**: pour saisir un numéro de téléphone
 - **email**: pour saisir une adresse mail
 - Etc.

Formulaire

145

Balise <label>

- Elle définit une étiquette pour un élément du formulaire (zone de texte, liste déroulante, couleur, etc.).
 - ▣ L'attribut **for** de la balise **<label>** doit correspondre à l'attribut **id** de l'élément du formulaire

```
<p>  
  <label for="nom">Votre nom:</label>  
  <input type="text" name="login" value="Antoine" id="nom"/>  
</p>
```

Votre nom:

Formulaire

146

Sélection automatique d'un champ

- L'attribut **autofocus** permet de mettre le point sur un champ du formulaire une fois la page Web chargée.
 - ▣ Le curseur pointe sur le champ considéré
 - ▣ Exemple:

```
<label for="nom">Votre nom:</label>  
<input type="text" name="login" id="nom" autofocus/>
```

Votre nom:

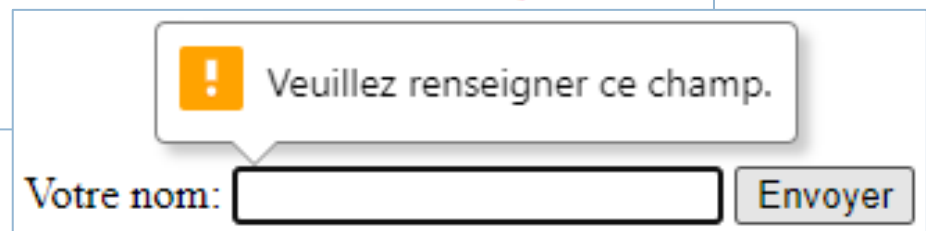
Formulaire

147

Champ obligatoire

- L'attribut **required** indique qu'un champ du formulaire est obligatoire.
 - ▣ Ce champ doit être rempli avant l'envoi des données (clic sur un bouton **submit**)
 - Sinon, une bulle d'information sera affichée
- ▣ Exemple:

```
<form>  
  <label for="username">Votre nom:</label>  
  <input type="text" name="username" id="username" required/>  
  <input type="submit"/>  
</form>
```



The screenshot shows a web form with the label "Votre nom:" followed by a text input field and a button labeled "Envoyer". A yellow speech bubble with an exclamation mark icon is positioned above the input field, containing the text "Veuillez renseigner ce champ." (Please fill in this field).

Formulaire

148

Balise <textarea>

- Elle permet d'ajouter une zone de texte multilignes.
 - ▣ Par défaut, le nombre de caractères est illimité
 - ▣ Comme une balise <pre>, les espaces et tabulations seront interprétés
 - ▣ Exemple:

```
<textarea name="multiline">Vous pouvez écrire votre  
commentaire ici</textarea>
```

Formulaire

149

Balise <select>

- Elle permet d'insérer une liste déroulante.
 - ▣ Les éléments de la liste sont ajoutés avec la balise **<option>**
 - ▣ Exemple:

```
<label for="liste">Choisir votre statut:</label><br>
<select name="drliste" id="liste">
  <option value="MA">Maitre Assistant</option>
  <option value="MC">Maitre de conférence</option>
  <option value="Pr">Professeur</option>
</select>
```

Choisir votre statut:

Maitre Assistant ▼

Choisir votre statut:

Maitre Assistant ▼

Maitre Assistant

Maitre de conférence

Professeur

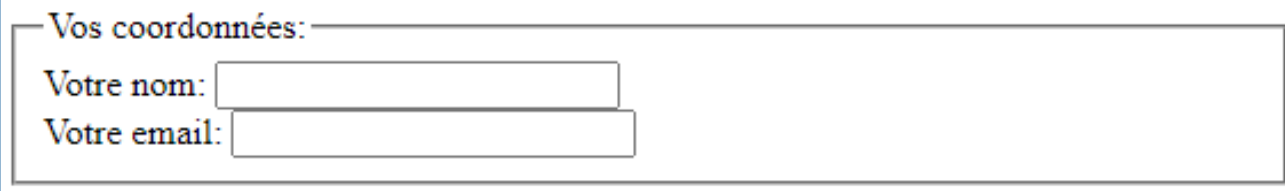
Formulaire

150

Balise <fieldset>

- Elle permet de regrouper des éléments associés dans un formulaire (balises **input**, **textarea**, **select**).
 - ▣ Un cadre sera ajouté autour du groupe créé
 - ▣ La balise <legend> définit le titre du groupe créé
 - ▣ Exemple:

```
<fieldset name="groupe">  
  <legend>Vos coordonnées:</legend>  
  <label for="nom">Votre nom:</label>  
  <input type="text" name="nom" id="nom"/><br>  
  <label for="mail">Votre email:</label>  
  <input type="email" name="mail" id="mail"/>  
</fieldset>
```



Pour finir...

154

- HTML est un langage de balisage permettant de définir la structure d'une page web.
 - ▣ Paragraphes
 - ▣ Listes
 - ▣ Liens
 - ▣ Tableaux
 - ▣ Multimédia
 - ▣ Etc.
- La mise en forme de la page n'est pas gérée par HTML plutôt par des feuilles de style CSS.

Référencement Naturel (SEO) et l'impact du HTML

Référencement Naturel (SEO)



Qu'est-ce que le SEO et pourquoi est-il important ?

156

Définition :

Le **SEO (Search Engine Optimization)** est l'ensemble des techniques permettant d'optimiser une page web pour qu'elle soit mieux classée dans les résultats des moteurs de recherche (Google, Bing, etc.).

Pourquoi est-ce important ?

- ✓ Augmente la visibilité d'un site web 
- ✓ Génère plus de trafic organique 
- ✓ Améliore l'expérience utilisateur 

Facteurs influençant le SEO :

Facteurs On-Page (HTML, contenu, liens internes, balises, etc.)

Facteurs Off-Page (backlinks, popularité, réseaux sociaux)

Facteurs Techniques (vitesse du site, mobile-friendly, HTTPS)



Le rôle du HTML dans le SEO

157

Le HTML joue un rôle clé dans l'optimisation SEO grâce à plusieurs éléments :

1 Balises essentielles pour le SEO :

`<title>` : Titre affiché dans l'onglet du navigateur et les résultats Google.

`<meta description>` : Résumé du contenu de la page affiché sous le titre dans les résultats de recherche.

`<h1>` à `<h6>` : Structure les titres et facilite la lecture pour Google.

`<alt>` pour les images : Décrit les images pour les moteurs de recherche et l'accessibilité.

2 Bonnes pratiques :

✓ **Titre optimisé** : 50-60 caractères, mots-clés importants au début.

✓ **Description percutante** : 150-160 caractères avec un call-to-action.

✓ **Structure claire** : Une seule `<h1>` par page, utilisation logique des sous-titres `<h2>`, `<h3>`.

Exemple d'une structure HTML optimisée pour le SEO :

158

Checklist SEO HTML :

- ✓ Un seul `<h1>` bien choisi
- ✓ Des `<h2>` et `<h3>` bien organisés
- ✓ Un `<title>` et `<meta description>` pertinents
- ✓ Des images avec des attributs `<alt>`
- ✓ Une structure lisible pour les moteurs de recherche et les utilisateurs

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Meilleures recettes de cuisine - Blog Culinaire</title>
  <meta name="description" content="Découvrez nos meilleures recettes de cui
</head>
<body>
  <header>
    <h1>Les Meilleures Recettes de Cuisine</h1>
  </header>
  <section>
    <h2>Recette du jour : Tarte aux pommes</h2>
    
  </section>
</body>
</html>
```

Checklist SEO HTML :

- ✓ Un seul <h1> bien choisi
- ✓ Des <h2> et <h3> bien organisés
- ✓ Un <title> et <meta description> pertinents
- ✓ Des images avec des attributs <alt>
- ✓ Une structure lisible pour les moteurs de recherche et les utilisateurs

160

DIV et SPAN



Introduction à `<div>` et ``

161

Que sont `<div>` et `` ?

✓ `<div>` (division) et `` sont des balises génériques utilisées pour regrouper du contenu sans donner de signification particulière.

✓ Elles sont souvent utilisées pour **appliquer du CSS, structurer une page, ou manipuler du contenu avec JavaScript.**

Différence entre `<div>` et ``

162

| | | |
|---------------------------|---------------------------|---|
| <code><div></code> | Bloc | Encapsule des sections de page (ex : paragraphes, menus, articles). |
| <code></code> | Ligne (inline) | Encapsule du texte ou une petite partie d'un élément. |



La balise `<div>` (division de contenu en blocs)

163

◆ Qu'est-ce que `<div>` ?

- ✓ `<div>` est une **balise de bloc** utilisée pour **structurer une page web**.
- ✓ Par défaut, elle prend **toute la largeur disponible** et **force un retour à la ligne**.

◆ Exemple d'utilisation de `<div>` :

```
<div>  
<h2>Section de contenu</h2>  
<p>Ceci est un exemple de division avec `<div>`.</p>  
</div>
```

<https://codepen.io/rtomczak/pen/bNbZgyx>



Utilisation principale :

- ▣ Encapsuler des **sections** de page (header, sidebar, footer).
- ▣ Appliquer un **style CSS** à une zone spécifique.
- ▣ Manipuler du contenu avec **JavaScript**.



La balise `` (mise en forme du texte inline)

164

◆ Qu'est-ce que `` ?

✓ `` est une **balise en ligne (inline)** utilisée pour **mettre en forme une partie spécifique du texte.**

✓ Contrairement à `<div>`, elle **ne force pas de retour à la ligne.**

◆ Exemple d'utilisation de `` :

<https://codepen.io/rtomczak/pen/vEBPgoJ>

✓ Utilisation principale :

Modifier **une partie d'un texte** sans affecter tout l'élément.

Appliquer un **style CSS** ciblé (couleur, police, arrière-plan).

Ajouter une **interaction JavaScript** sur une portion du texte.



Comparaison entre `<div>` et ``

165

| Critère | <code><div></code> | <code></code> |
|---------------------|--|--|
| Type | Bloc | Inline |
| Retour à la ligne ? | Oui | Non |
| Utilisation | Structure une page (sections, colonnes, menus, etc.) | Modifie une partie du texte |
| Peut contenir | Texte + autres balises (titres, paragraphes, images, etc.) | Seulement du texte ou d'autres balises inline |
| CSS typique | width, margin, padding, border, background | color, font-weight, text- decoratio |



Règle d'or :

- Utiliser `<div>` pour organiser la mise en page.
- Utiliser `` pour formater du texte à l'intérieur d'un paragraphe



Exemple avancé combinant <div> et

166

❑ ◆ Exemple : Carte de profil avec styles CSS

```
<div style="border: 1px solid black; padding: 15px; width: 250px;">
  <h2>Nom : <span style="color: blue;">Jean Dupont</span></h2>
  <p>Rôle : <span style="font-weight: bold;">Développeur Web</span></p>
</div>
```

❑ 🎯 Explication :



<div> structure la carte de profil.



 permet de styliser **le nom et le rôle** sans affecter tout le texte.



Conclusion : Quand utiliser `<div>` et `` ?

167

- ✓ **Utiliser `<div>`** pour structurer des blocs de contenu.
- ✓ **Utiliser ``** pour modifier une partie du texte.
- ✓ **Bien combiner les deux** pour une meilleure organisation et mise en page ! 🎨

168

Responsive



Pourquoi un site doit être responsive ?

169

1

Expérience utilisateur améliorée



Pas besoin de zoomer ni de faire défiler horizontalement.



Une navigation fluide et intuitive sur **tous les appareils**.

2

SEO et référencement Google



Google favorise les sites **mobile-friendly** dans les résultats de recherche.



Améliore le **temps passé sur la page**, ce qui renforce le référencement.

3

Adaptabilité universelle



Un seul code HTML/CSS qui s'adapte à **tous les écrans** (ordinateur, tablette, mobile).



Plus besoin de développer une **version mobile séparée**.



Définition `<meta>`

170

Explication de la balise `<meta name="viewport" content="width=device-width, initial-scale=1.0">`

Cette balise `<meta>` est utilisée pour **contrôler l'affichage et le comportement du site web sur les appareils mobiles et les écrans de différentes tailles.**



Comment rendre un site responsive ?

171

- **1** Utiliser la balise `<meta viewport>`
- **2** Utiliser des Media Queries en CSS
- **3** Utiliser un design "Flexbox" ou "Grid" en CSS

- Explication détaillée de la balise `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- La balise `<meta name="viewport">` est utilisée pour contrôler la manière dont une page web s'affiche sur différents appareils, en particulier sur **les mobiles et les tablettes**.
- Elle permet d'adapter automatiquement la mise en page pour **éviter d'avoir un zoom par défaut ou une mauvaise disposition du contenu sur les petits écrans**.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

173

| Attribut | Signification |
|---------------------------|--|
| name="viewport" | Indique que la balise concerne l'affichage et l'adaptation du site sur les différents écrans. |
| content="..." | Contient les différentes options de configuration du viewport. |
| width=device-width | Définit la largeur du site pour qu'elle corresponde à la largeur réelle de l'écran de l'appareil. |
| initial-scale=1.0 | Définit un zoom initial de 100% (aucun agrandissement ni réduction). |



Pourquoi cette balise est-elle essentielle ?

174

1 Sans la balise <meta viewport>

(mauvais affichage sur mobile)



Problèmes :


✗ Le site s'affiche comme sur un écran d'ordinateur.

✗ Le texte et les images sont **trop petits**, obligeant l'utilisateur à **zoomer** et **faire défiler horizontalement**.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Site sans viewport</title>
</head>
<body>
  <h1>Mon site web</h1>
  <p>Le texte est trop petit sur mobile et ne s'adapte pas.</p>
</body>
</html>
```

2 Avec la balise <meta viewport> (affichage optimisé sur mobile)

175

-  **Solutions :**
 - ✓ Le site s'ajuste **automatiquement** à l'écran de l'utilisateur.
 - ✓ Pas besoin de **zoomer** pour lire le texte ou voir les images correctement

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Site responsive</title>
</head>
<body>
  <h1>Mon site web</h1>
  <p>Le texte s'ajuste automatiquement à l'écran.</p>
</body>
</html>
```

2 Avec la balise <meta viewport> (affichage optimisé sur mobile)

176

Solutions :

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Site responsive</title>
</head>
<body>
  <h1>Mon site web</h1>
  <p>Le texte s'ajuste automatiquement à l'écran.</p>
</body>
</html>
```

correctement



Explication détaillée des options possibles dans `content="..."`

177

1 `width=device-width`

Explication :

Ajuste **la largeur du site web** pour qu'elle soit égale à la largeur de l'écran du périphérique.

Ex : Un smartphone de 375 pixels de large affichera le site sur **375px**, et non sur **1024px** comme sur un ordinateur.

2 `initial-scale=1.0`

Explication :

Définit le **niveau de zoom initial** de la page.

1.0 signifie **100%**, c'est-à-dire **aucun zoom appliqué**.

Ex : Si on met `initial-scale=2.0`, la page sera affichée **avec un zoom de 200%**.



Explication détaillée des options possibles dans `content="..."`

178

3 `maximum-scale=1.0`

Explication :

Empêche l'utilisateur de zoomer au-delà d'un certain seuil.

Ex : `maximum-scale=1.0` bloque le zoom au niveau 100%.



Attention : Ne pas **empêcher totalement le zoom**, car cela **réduit l'accessibilité** pour les utilisateurs malvoyants.

4 `minimum-scale=1.0`

Explication :

Définit le **zoom minimum autorisé** par l'utilisateur.



5 `user-scalable=no`

Explication :

Désactive complètement le zoom manuel (pinch-to-zoom sur mobile).



Mauvaise pratique pour l'accessibilité → Recommandé d'éviter cette option.

179

Les balises sémantiques



Pourquoi utiliser des balises sémantiques ?

180

Qu'est-ce qu'une balise sémantique ?

Une **balise sémantique** décrit clairement le contenu qu'elle contient, contrairement aux balises génériques comme `<div>` et ``.

Pourquoi c'est important ?

- ✓ **Améliore l'accessibilité** : Les lecteurs d'écran comprennent mieux le contenu.
- ✓ **Optimise le SEO** : Google et les moteurs de recherche analysent plus efficacement la structure du site.
- ✓ **Facilite la maintenance** : Un code mieux organisé est plus lisible et compréhensible.

Exemples de balises sémantiques :

`<header>` : En-tête de la page ou d'une section.

`<nav>` : Regroupe les liens de navigation.

`<section>` : Regroupe un contenu thématique.

`<article>` : Contenu indépendant (ex : un article de blog).

`<aside>` : Contenu secondaire ou complémentaire.

`<footer>` : Pied de page contenant des informations de fin (copyright, contact...).



Pourquoi utiliser des balises sémantiques ?

181

Qu'est-ce qu'une balise sémantique ?

Une **balise sémantique** décrit clairement le contenu qu'elle contient, contrairement aux balises génériques comme `<div>` et ``.

Pourquoi c'est important ?

- ✓ **Améliore l'accessibilité** : Les lecteurs d'écran comprennent mieux le contenu.
- ✓ **Optimise le SEO** : Google et les moteurs de recherche analysent plus efficacement la structure du site.
- ✓ **Facilite la maintenance** : Un code mieux organisé est plus lisible et compréhensible.

Exemples de balises sémantiques :

`<header>` : En-tête de la page ou d'une section.

`<nav>` : Regroupe les liens de navigation.

`<section>` : Regroupe un contenu thématique.

`<article>` : Contenu indépendant (ex : un article de blog).

`<aside>` : Contenu secondaire ou complémentaire.

`<footer>` : Pied de page contenant des informations de fin (copyright, contact...).



Comparaison entre balises sémantiques et génériques

182

```
<div id="header">En-tête du site</div>
<div id="menu">Menu de navigation</div>
<div id="content">
  <div class="post">
    <h1>Mon article</h1>
    <p>Contenu de l'article...</p>
  </div>
</div>
<div id="footer">Copyright 2025</div>
```

Problèmes :

✗ Manque de clarté pour les moteurs de recherche et lecteurs d'écran.

✗ Code plus difficile à maintenir.



Comparaison entre balises sémantiques et génériques

183

```
<header>En-tête du site</header>
<nav>Menu de navigation</nav>
<main>
  <article>
    <h1>Mon article</h1>
    <p>Contenu de l'article...</p>
  </article>
</main>
<footer>Copyright 2025</footer>
```

Avantages :

- ✓ Meilleure compréhension du contenu pour Google et les navigateurs.
- ✓ Code plus propre et structuré.

| Balise | Rôle |
|------------------------------|--|
| <code><header></code> | Contient le titre du site et la navigation principale. |
| <code><nav></code> | Regroupe les liens de navigation. |
| <code><main></code> | Contient le contenu principal de la page (un seul <code><main></code> par page). |
| <code><section></code> | Définit une section thématique (exemple : "À propos"). |
| <code><article></code> | Contenu autonome comme un article de blog ou une news. |
| <code><aside></code> | Contenu complémentaire (exemple : articles connexes). |
| <code><footer></code> | Contient le pied de page avec des infos légales, copyright, etc. |



Bonnes pratiques pour un HTML structuré

185

1 Toujours utiliser les bonnes balises

Utiliser `<header>` au lieu de `<div id="header">`.

Préférer `<article>` pour un contenu autonome plutôt que `<div>`.

Utiliser `<section>` pour organiser les différentes parties d'une page.

2 Respecter une hiérarchie logique

Un seul `<h1>` par page.

`<h2>`, `<h3>`, etc. doivent suivre un ordre logique.

3 Rendre le contenu accessible

Ajouter des **attributs alt** aux images pour les lecteurs d'écran.

Utiliser des **attributs ARIA** si nécessaire (`role="navigation"`, `aria-label`, etc.)



Bonnes pratiques pour un HTML structuré

186

4 Éviter les divs inutiles



`<div class="section">`



`<section>`

5 Tester son code

Vérifier la **validité** du

HTML : <https://validator.w3.org/>

Utiliser **Lighthouse** (Google Chrome) pour tester l'accessibilité et le SEO.

<header>

<nav>

<section>

<article>

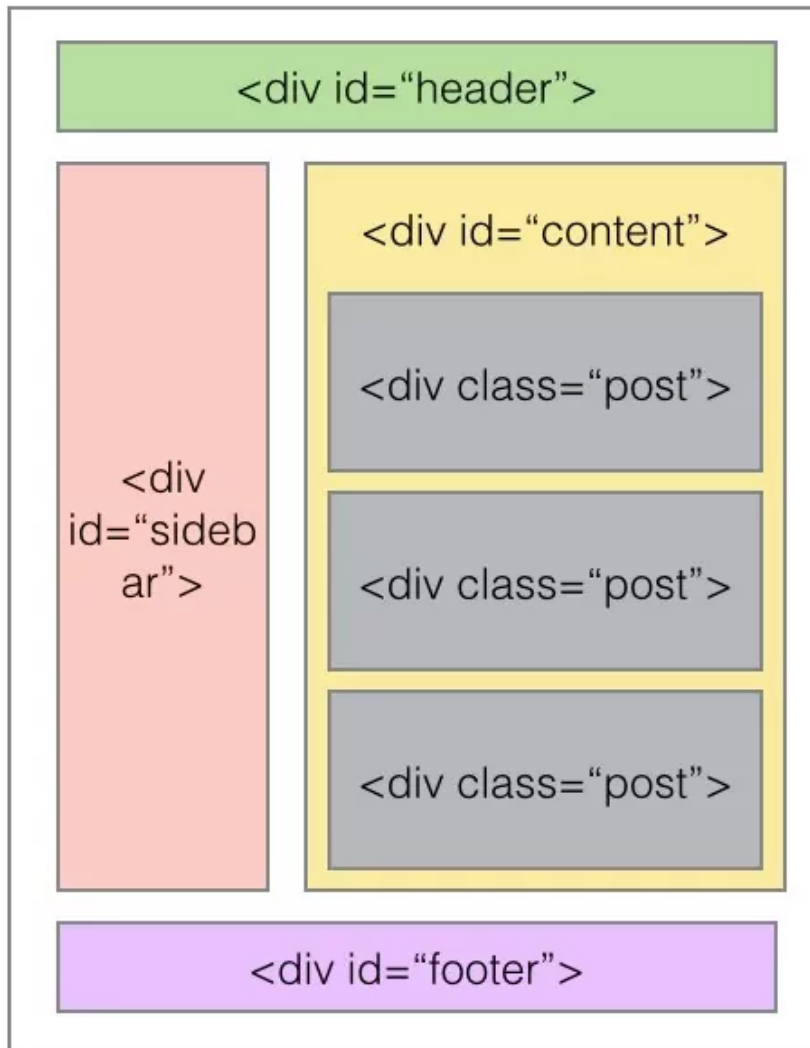
<article>

<article>

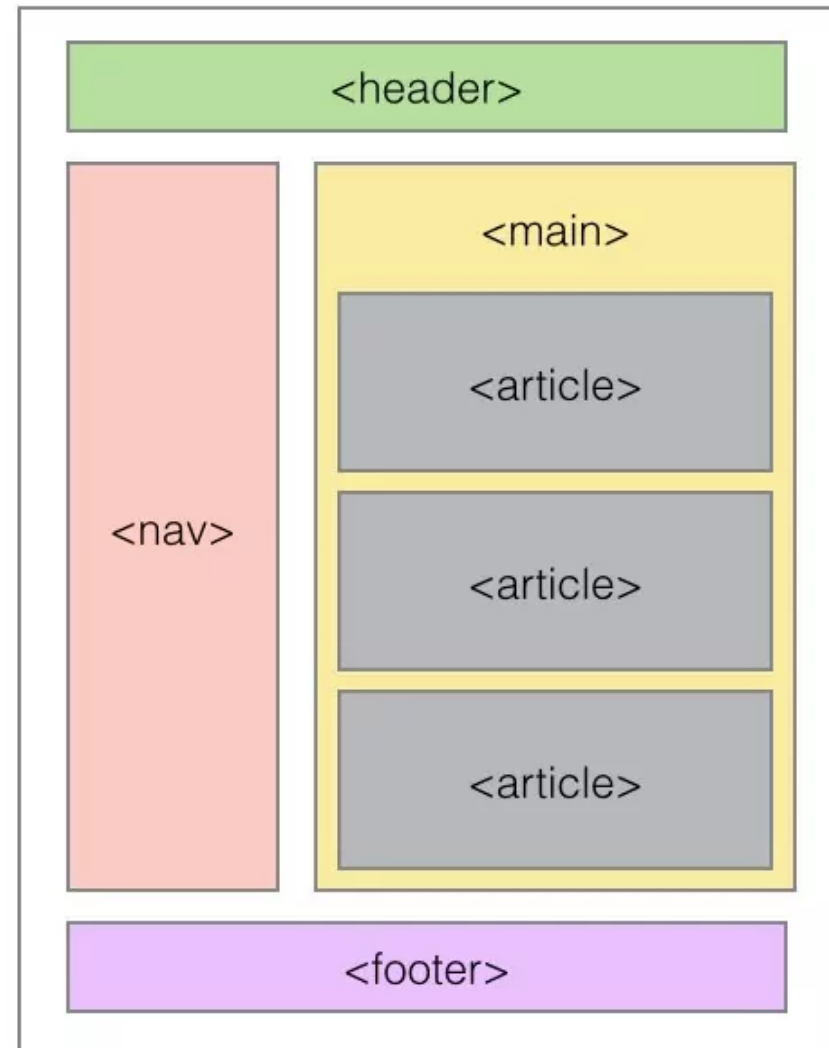
<aside>

<footer>

HTML4: Lots of Classes/IDs



HTML5: Semantic Tags/Sections



Exemple

189

- Lien
- <https://codepen.io/rtomczak/pen/gbYEgdG>

190

Erreurs communes

FIN (ENFIN ?)