

HTML : HyperText Markup Language



Présentation

- Les pages web sont écrites en HTML (.htm ou .html) :
HyperText Markup Language
- Ce qu'est HTML :
 - Un langage descriptif : contenu de la page, description des paragraphes, images, liens, tableaux...
 - Pas de compilation : le navigateur reçoit le HTML, l'interprète, et produit la page telle qu'affichée au client. Toujours possible pour le client de consulter le code HTML !
- Ce qu'HTML n'est pas :
 - Un langage de programmation :
 - Pas de structures conditionnelles («if ... then ... else»)
 - Pas de boucles (for, while, ...)
 - Pour programmer, il faudra utiliser un langage de programmation côté client (javascript, flash...) ou côté serveur (PHP, perl...).

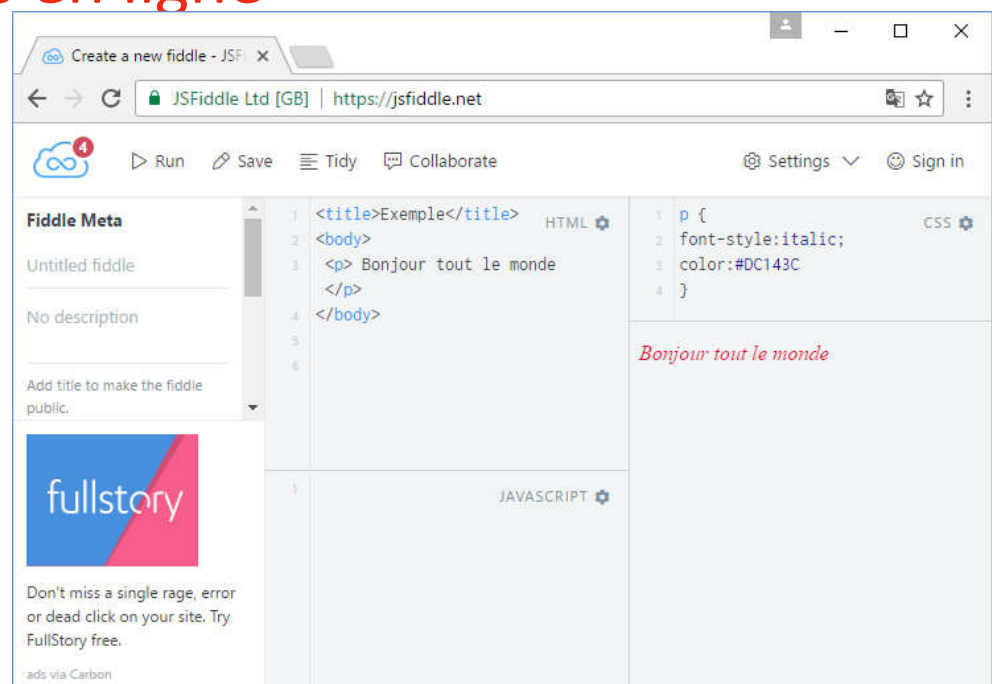
Outils pour créer du contenu web

- Outils de base
 - Editeur de texte (bloc-notes, notepad++, sublime, vi, ...)
 - Environnements de développement (Visual Studio, Eclipse, ...)
- Outils spécifiques
 - Outils Adobe (Dreamweaver, Brackets, ...)
 - ...
- Outils en ligne
 - Éditeurs
 - JS Fiddle, Code Pen, JS Bin, CSS Deck...
 - IDE
 - Shiftedit, Source Lair...

45

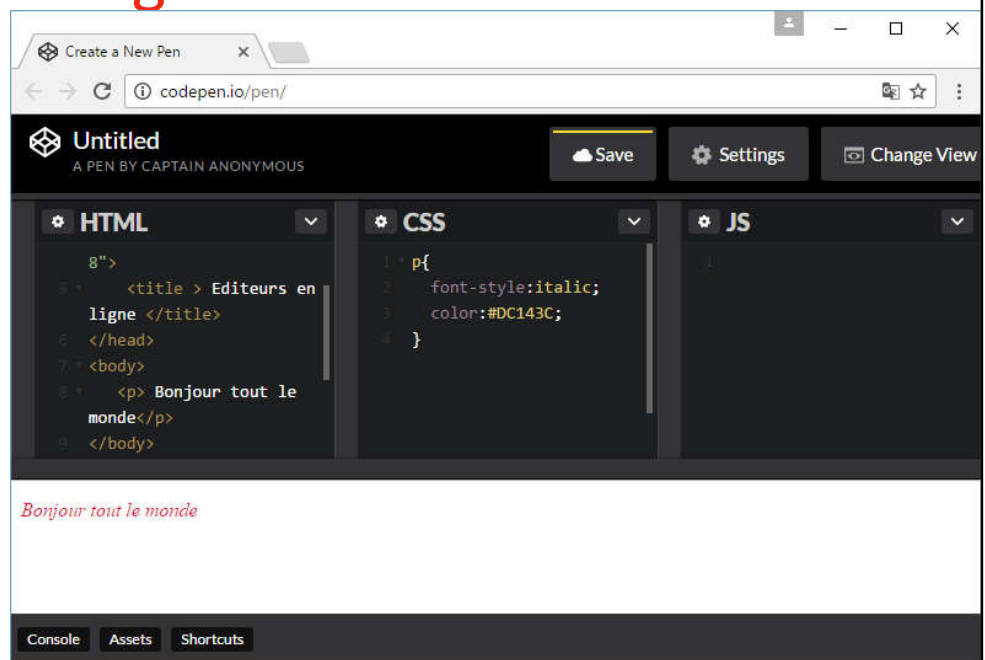
Éditeurs en ligne

■ JSFiddle



Éditeurs en ligne

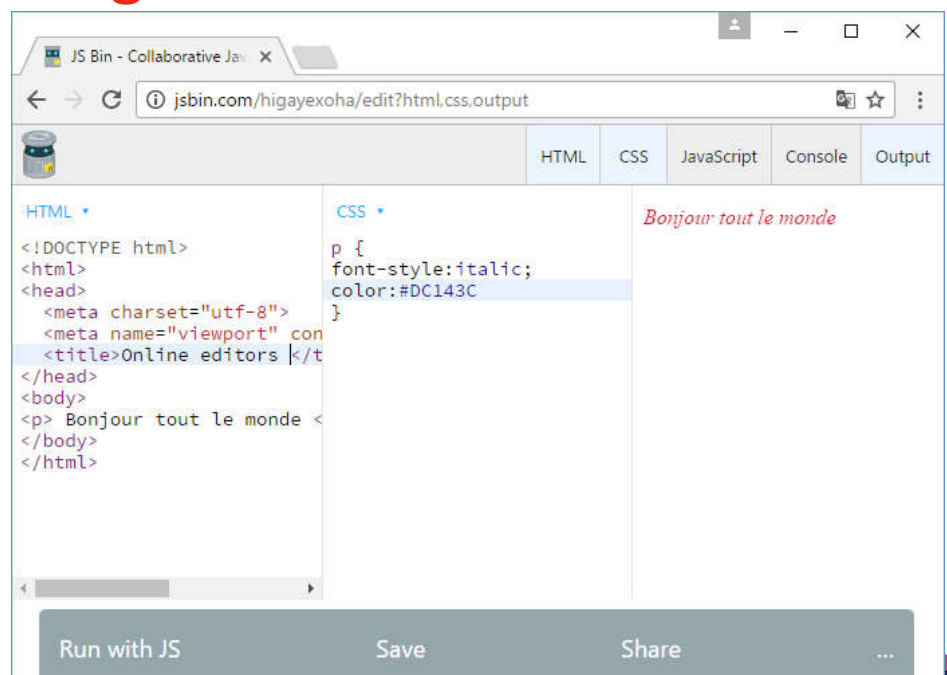
■ Codepen



47

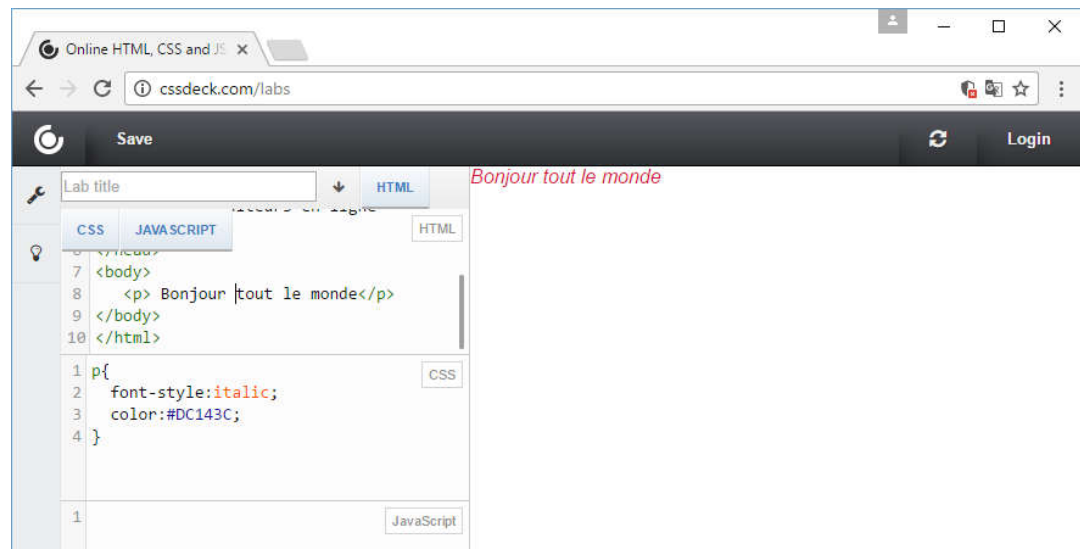
Éditeurs en ligne

■ jsbin



Éditeurs en ligne

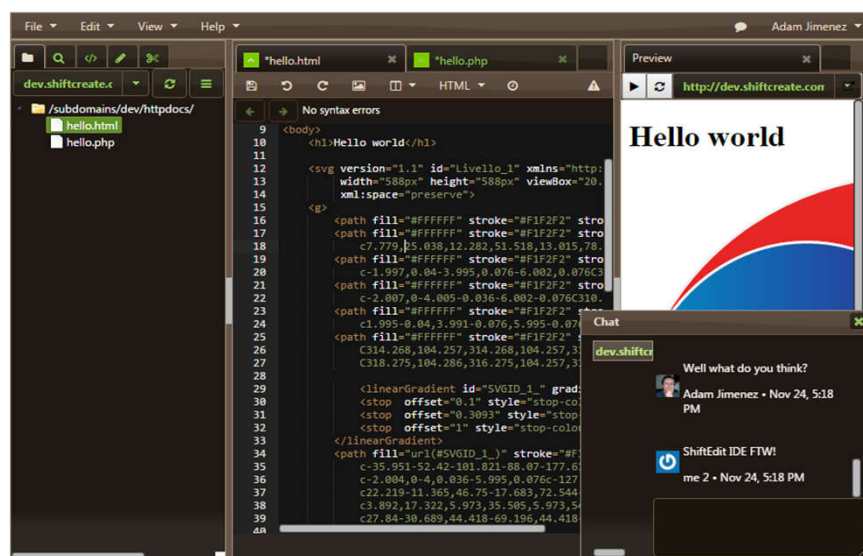
■ Cssdeck



49

IDE en ligne

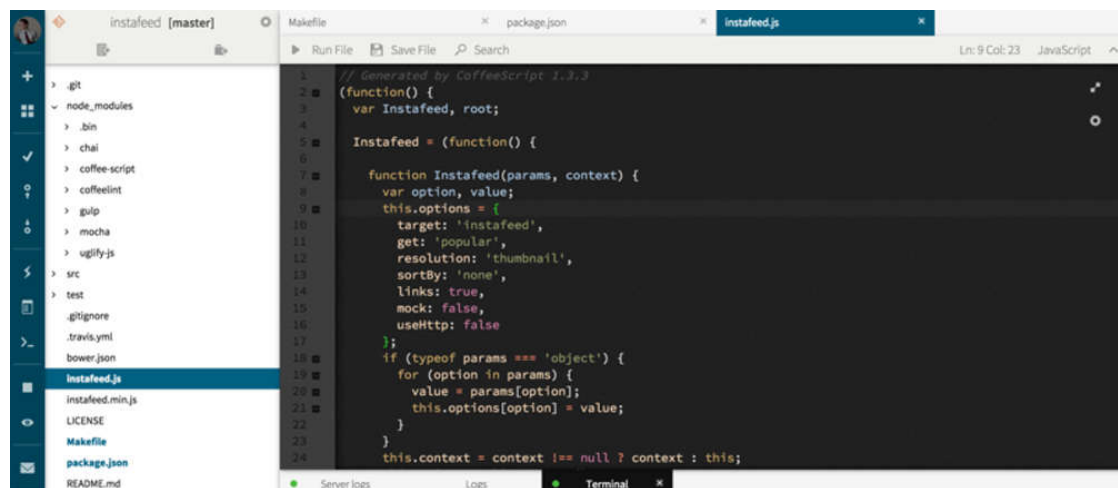
■ Shiftedit



50

IDE en ligne

■ Source Lair



51

Outils pour débbugger

- Debuggers intégrés dans les navigateurs (F12)
 - Chrome Dev Tools
 - Mozilla Firefox Developer Tools
 - Microsoft Developer Tools
 - Safari Developer tools
- Quel type de debugging ?
 - Web Inspector: HTML, CSS
 - Debuggage JavaScript
 - Analyse réseau
 - Analyse des performances

52

Historique

- Quelques dates
 - 1991 : première utilisation du terme HTML par Tim Berners-Lee
 - 1993 : HTML première version publiée par IETF
 - 1995 : HTML 2 normalisé par le W3C
 - 1997 : HTML 3.2
 - 1999 : sortie d'HTML 4.01
 - 2001 : XHTML
 - 2011 : HTML5

53

HTML5 : nouveautés

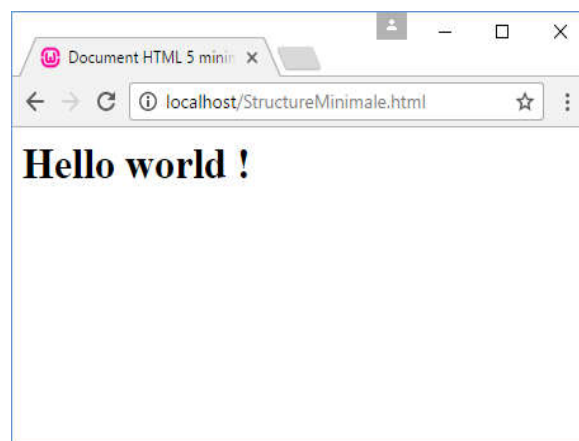
- À la fois une évolution de HTML 4 et un ensemble de technologies développées par le **WHATWG** et le **W3C**.
 - Nouveaux éléments sémantiques
 - <article>, <section>, <aside>, <header>, <footer>...
 - Nouveaux éléments multimédia
 - <canvas>, <audio>, <video>
 - Nouvelles API JavaScript:
 - Web worker, géolocalisation, web-storage, drag&drop...
 - Souvent associé à CSS 3
 - HTML5 vs XHTML

54

Page HTML

- Exemple d'une page HTML

```
<!DOCTYPE html>
<html>
<head lang="fr">
  <meta charset="UTF-8">
  <title> Document HTML 5 minimal </title>
</head>
<body>
  <h1>Hello world !</h1>
</body>
</html>
```



55

Le doctype

- Un document HTML doit toujours débiter par un doctype
 - Il s'agit d'une déclaration permettant de renseigner le navigateur sur le type de document HTML délivré.
 - Il ne s'agit pas d'un tag HTML à part entière, mais bien d'un préambule.
- Jusqu'à HTML5
 - le **doctype** mentionnait l'utilisation d'un document de référence, la **DTD** (Document Type Definition), grammaire résumant la syntaxe HTML et ses attributs.
 - Exemple HTML 4.01 Strict :


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
```

56

Le doctype

- Depuis HTML5
 - le **doctype** a été considérablement simplifié
 - il n'est pas sensible à la casse
 - aucune DTD n'est précisée.
 - Exemple de doctype HTML5 :
 - `<!DOCTYPE html>`

57

Structure d'une page

- Structure 1/2
 - un document HTML est en réalité un document XML avec des balises prédéfinies
 - `<!DOCTYPE html>`
 - la balise **html** marque le début du document
 - `<html>`
 - la balise **head** contient des informations relatives à la page : le type d'encodage du texte, le titre, . . .
 - `<head lang="fr">`
 - `<meta charset="UTF-8">`
 - `<title> Document HTML5 minimal </title>`
 - `</head>`

58

Structure d'une page

■ Structure 2/2

- la balise **body** contient le corps de la page (les éléments visuels de la page ; dans l'exemple, un texte mis en style h1)

```
<body>
```

```
<h1>Hello world !</h1>
```

- le document se termine par la fermeture des balises body et html

```
</body>
```

```
</html>
```

59

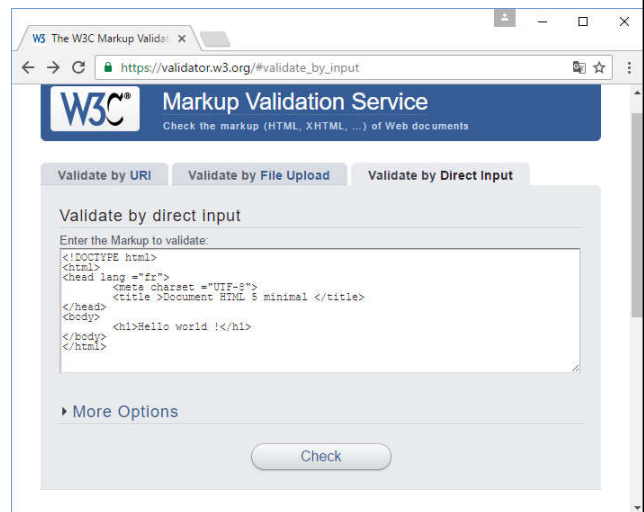
Validation du code HTML

- À la différence de langages de programmation, un non-respect de la syntaxe de HTML n'empêchera pas l'affichage sur le navigateur.
- Plus généralement, le navigateur affichera quelque chose (de non spécifié) ou n'affichera rien, mais n'affichera jamais d'erreur.
- La validation permet de vérifier la correction syntaxique.
 - Il faut **impérativement valider** son code HTML avant de le publier.

60

Validation du code HTML

- Un outil indispensable
 - le validateur du W3C
 - <http://validator.w3.org>

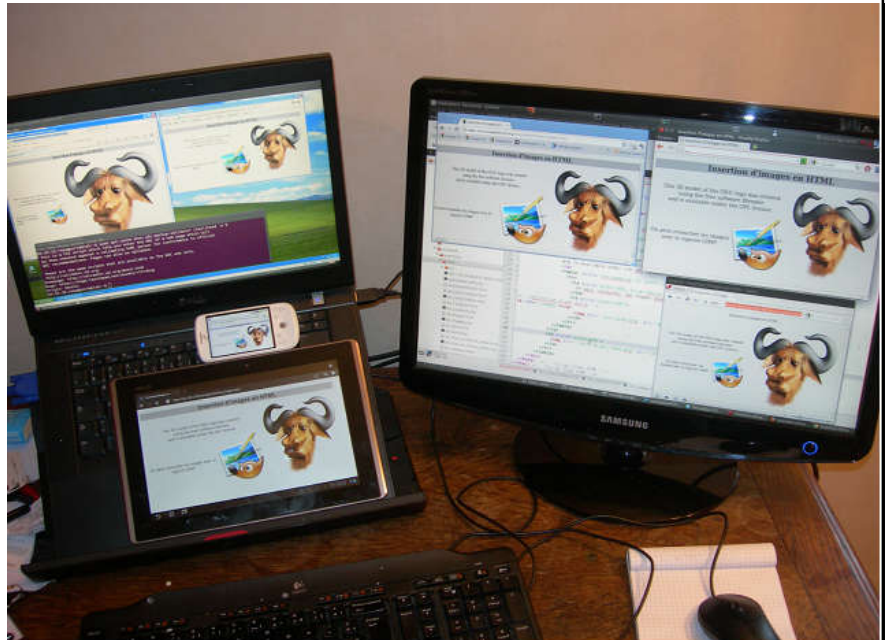


Tests de portabilité

- Même pour un document validé, les navigateurs n'ont pas tous la même implémentation du moteur d'analyse et de rendu du document
 - notamment en ce qui concerne les styles par défaut.
- Pour cette raison, il est indispensable avant de publier un site web, de le tester sur le plus possible de plate-formes et de navigateurs.

Tests de portabilité

l'affichage de la page est testé simultanément avec Internet Explorer et Safari sous Windows, Firefox, Chrome et Opera sous linux et Chrome sous tablette et smartphone Android.



63

Un langage à base de balises

- HTML repose sur la notion de balises
- Deux types de balises :
 - Les balises qui sont ouvertes puis fermées, et encadrent du contenu
 - ex : `<p> contenu du paragraphe </p>`
 - Les balises qui s'ouvrent et se ferment en même temps
 - ex : `
`

64

Un langage à base de balises

- Exemple :

```

<h1>Un exemple de titre</h1>
<hr />
<div>
  <p>Du texte <b>gras</b>,
  puis<em>italique</em>,
  puis <b><em>gras et italique</em></b>.
</p>
  
</div>

```

65

Règles d'écriture: Imbrication/Emboitement

- Le contenu d'un élément peut contenir d'autres éléments « imbriqués ».

```

<p>
  début
  <code>
    emboité1
    <strong>
      emboité2
    </strong>
    emboité1
  </code>
  suite
</p>

```

66

Règles d'écriture: Imbrication/Emboitement

- Les emboitements obéissent à des règles, tout n'est pas possible.
- Exemples :
 - un élément `<p>` ne peut pas contenir un élément `<h1>`
 - un élément `` contient nécessairement au moins un élément ``
 - un élément `` est nécessairement emboité dans un élément ``
 - etc.

67

Règles d'écriture: Imbrication/Emboitement

- Les éléments ne doivent pas se chevaucher : premier ouvert, dernier fermé.
- Ceci est **interdit** :

`<p>` début `<code>` emboité ? `</p>` suite `</code>`

68

Balises et hiérarchie

- Les balises sont structurées de façon hiérarchique
- Chaque balise hérite des propriétés de la balise de niveau supérieur à moins qu'elles ne les redéfinissent.
 - `<p>Ce texte est seulement en gras.``</p>`
 - `<p>Ce texte est seulement en gras et cette`
partie est en gras et en italique`</i>.``</p>`

69

Balises et attributs

- Certaines balises peuvent posséder des attributs.
 - Exemples :
 - `<meta charset="utf-8" />`
 - ``
 - `<p style="text-align : right"> Du texte aligné à droite``</p>`

70

Attributs

- définit une propriété pour un élément
- se note dans la balise ouvrante de l'élément, en minuscules et sa valeur est entre guillemets " "
- certains sont communs à tous les éléments
 - **title** information textuelle sur le contenu de l'élément
 - **class** associe une ou plusieurs classes à l'élément, une classe peut être partagée par plusieurs éléments
 - **id** nomme un élément de manière unique dans le document
- d'autres sont spécifiques à un élément
 - **src** pour l'élément `` désigne la source de l'image

```

```

71

Commentaires

- Les commentaires sont du code HTML qui n'est pas exécuté (affiché)
- Syntaxe : `<!-- Commentaires -->`

```
<h1>Un titre (qui va s'afficher)</h1>
<p>Cette phrase va s'afficher sur le navigateur.
<!-- En revanche, cette phrase ne s'affichera pas --></p>
```
- NB : Les commentaires restent visibles dans la source de la page Web, qui est accessible depuis le client.

72

Balises pour le texte

- Paragraphes

- Balise `<p>` : paragraphe
- Balise `
` : saut de ligne

- Exemple :

```
<body>  
  <p>Un premier paragraphe.</p>  
  <p>Un second paragraphe.</p>  
</body>
```

- Bonne pratique

- Il est d'usage de créer un second paragraphe (`<p>`) plutôt qu'un saut de ligne (`
`).

73

Balises pour le texte

- Mise en forme

- Balise `` (bold) : gras
- Balise `<i>` (italic) : italique
- Balise `<u>` (underline) : souligné
- Balise `` (emphasis) : emphase (s'affiche en général comme italique)
- Balise `` : mise en évidence plus forte (en général gras)

- Bonne pratique

- Le soulignage ne devrait jamais être utilisé en informatique, à l'exception (éventuelle) des liens hypertextes (qui tend elle-même à disparaître).

74

Balises pour le texte

- Espaces
 - Les espaces affichées en HTML s'affichent également sur le navigateur.
 - Attention donc à ne pas multiplier les espaces !
 - En revanche, les sauts de ligne en HTML deviennent des espaces sur le navigateur.
 - Pour afficher un saut de ligne, utiliser `
`
`<p>`Cette phrase va s'afficher
sur une seule ligne.`
`Cette phrase sera dessous .`</p>`

75

Balises pour le texte

- Les caractères spéciaux
 - En HTML, tous les caractères spéciaux peuvent être remplacés par un code commençant par **&**

| | | | | | |
|------------|----------|----------|---|---------|----------|
| © | © | © | « | « | « |
| ™ | ™ | ™ | » | » | » |
| ® | ® | ® | " | " | " |
| Esp. inséc | | | & | & | & |
| < | < | < | ¬ | ¬ | ¬ |
| > | > | > | º | ⩝ | º |
| ¿ | ¿ | ¿ | ª | &orda; | ª |
| \$ | - | $ | ¢ | ¢ | ¢ |
| ¥ | ¥ | ¥ | £ | £ | £ |
| ¡ | ¡ | ¡ | € | € | € |

76

Balises pour le texte

- Les titres
 - les titres sont définis par la balise `<hx>`
 - où `x` désigne un niveau
 - il existe 6 niveaux : 1 étant associé à la taille de caractères la plus grande et 6 la plus petite

77

Balises pour le texte

- Les titres
 - Exemple :
 - `<h1>Premier chapitre</h1>`
 - `<h2>Première partie</h2>`
 - `<h3>Premier paragraphe</h3>`



78

Les listes

- il existe plusieurs types de listes : ordonnées ou non ou de définitions
 - les listes ordonnées contiennent une séquence d'éléments dont la position est importante
 - les listes non ordonnées contiennent des éléments dont l'ordre n'a pas d'importance
 - une liste de définition contient des termes et leurs définitions
- une liste peut contenir d'autres listes

79

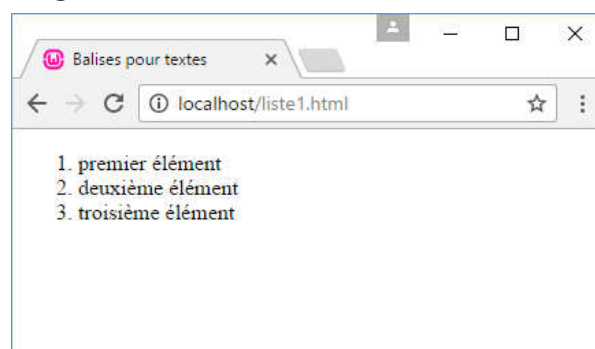
Les listes

- Exemple de liste ordonnée

premier élément

deuxième élément

troisième élément



80

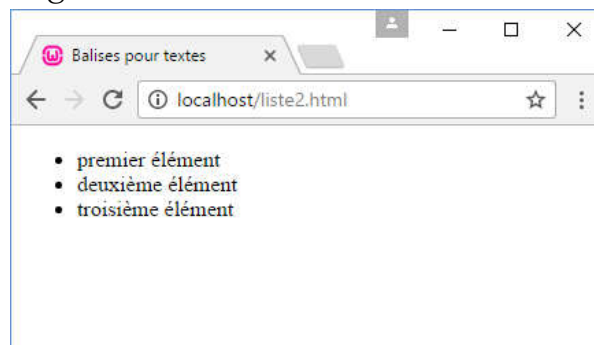
Les listes

■ Exemple de liste non ordonnée

premier élément

deuxième élément

troisième élément



81

Les listes

■ Exemple de liste de définition

<dl>

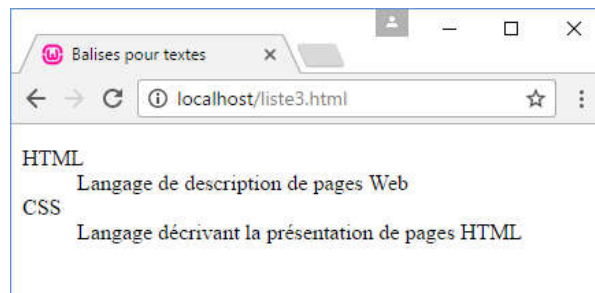
<dt>HTML</dt>

<dd>Langage de description de pages Web</dd>

<dt>CSS</dt>

<dd>Langage décrivant la présentation de pages HTML</dd>

</dl>



82

Les tableaux

■ Définition

- les tableaux permettent d'organiser les éléments d'une page sous forme tabulaire
- un tableau est un ensemble de lignes
- une ligne est un ensemble de cellules

■ Exemple:



| Article | Prix (DH) |
|---------|-----------|
| Ecran | 800 |
| Clavier | 70 |
| Total | 870 |

83

Les tableaux

■ Les balises

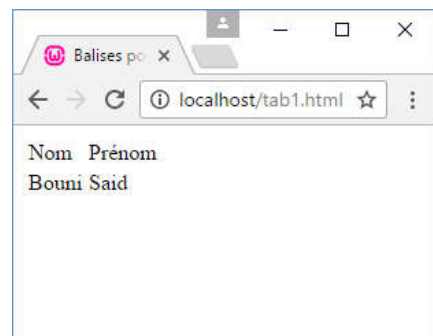
- la balise `<table>` pour délimiter le tableau
 - `<tr>` (*table row*) pour définir une ligne
 - `<td>` (*table data*) pour une cellule
- le tableau s'adapte aux éléments placés dans les cellules
- par défaut, aucun élément graphique indique les séparations entre les cellules

84

Les tableaux

■ Exemple <table> minimaliste

```
<table>
  <tr>
    <td>Nom</td>
    <td>Prénom</td>
  </tr>
  <tr>
    <td>Bouni</td>
    <td>Said</td>
  </tr>
</table>
```



85

Les tableaux

■ Entête, corps et pied

- il est possible de " typer " les lignes :
 - <thead> pour l'entête
 - <tbody> pour le corps
 - <tfoot> pour le pied
- de plus, les cellules peuvent être typées "entête " avec la balise <th> (*table header*) et l'attribut **scope**
 - <th> doit être imbriquer dans une balise <tr>
- **Attention** : <tfoot> doit apparaitre avant <tbody> dans le code afin de fournir au navigateur les renseignements qu'il détient, avant de charger le reste du contenu.

86

Les tableaux : Exemple

```

<table>
  <thead>
    <tr>
      <th scope="col">
        Article</th>
      <th scope="col">
        Prix (DH)</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Ecran</td>
      <td>800</td>
    </tr>
    <tr>
      <td>Clavier</td>
      <td>70</td>
    </tr>
    <tr>
      <td>Total</td>
      <td>870</td>
    </tr>
  </tbody>
</table>

```



| Article | Prix (DH) |
|---------|-----------|
| Ecran | 800 |
| Clavier | 70 |
| Total | 870 |

87

Les tableaux

■ Fusion

- afin d'améliorer la présentation, on peut fusionner des colonnes ou des lignes
- deux attributs :
 - **colspan** : n colonnes d'une ligne donnée sont fusionnées
 - **rowspan** : n lignes d'une colonne donnée sont fusionnées

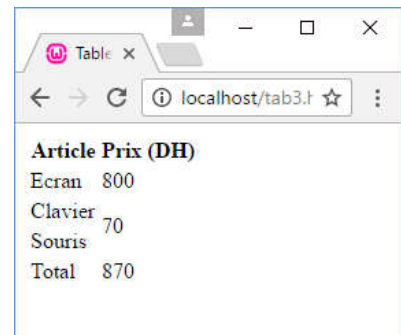
88

Les tableaux : Exemple

```

<table>
  <thead>
    <tr>
      <th>Article</th>
      <th>Prix (DH)</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td>Total</td>
      <td>870</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Ecran</td>
      <td>800</td>
    </tr>
    <tr>
      <td>Clavier</td>
      <td rowspan="2">70</td>
    </tr>
    <tr>
      <td>Souris</td>
    </tr>
  </tbody>
</table>

```



| | |
|---------|-----|
| Ecran | 800 |
| Clavier | 70 |
| Souris | 70 |
| Total | 870 |

89

Les tableaux

- Espacement et remplissage
 - les attributs **cellspacing** et **cellpadding** définissent l'espacement entre les cellules et le remplissage d'une cellule
 - **Attention**: non supporté par HTML5 et donc utiliser les CSS

90

Les tableaux : Exemple

```

<table cellpadding="10">
  <thead>
    <tr>
      <th>Article</th>
      <th>Prix (DH)</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Ecran</td>
      <td>800</td>
    </tr>
    <tr>
      <td>Clavier</td>
      <td>70</td>
    </tr>
    <tr>
      <td>Total</td>
      <td>870</td>
    </tr>
  </tbody>
</table>

```



| Article | Prix (DH) |
|---------|-----------|
| Ecran | 800 |
| Clavier | 70 |
| Total | 870 |

91

Les tableaux

■ Légende ou titre

— La balise <caption>

- permet d'attribuer une légende ou un titre à un tableau de données.
- doit être insérée immédiatement après la balise ouvrante
- ne doit être utilisée qu'une seule fois au sein de cet élément.

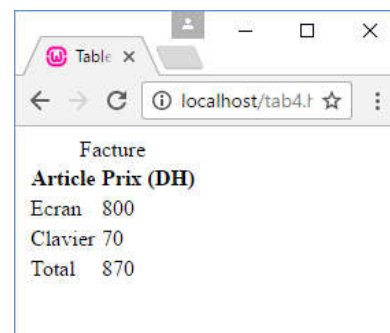
92

Les tableaux : Exemple

```

<table>
  <caption> Facture
</caption>
  <thead>
    <tr>
      <th>Article</th>
      <th>Prix (DH)</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Ecran</td>
      <td>800</td>
    </tr>
    <tr>
      <td>Clavier</td>
      <td>70</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Total</td>
      <td>870</td>
    </tr>
  </tfoot>
</table>

```



| Article | Prix (DH) |
|---------|-----------|
| Ecran | 800 |
| Clavier | 70 |
| Total | 870 |

93

Les Liens

- Définition
 - la notion de lien est au cœur du Web : référencer au sein d'une page d'autres pages ou ressources
 - la balise utilisée est **<a>**
 - l'adresse cible (ou URL) est précisée par l'attribut **href**
 - un lien est représenté par l'élément placé entre **<a>** et ****
 - cet élément peut être un texte mais aussi une image
- Exemple
 - `Mon site web `

94

Les Liens

■ URL

- toute ressource du Web possède un identifiant unique : l'URL (*Uniform Resource Locator*)
- il existe 2 types d'URL : **absolue** et **relative**
- une URL absolue commence par le nom de domaine du serveur contenant la ressource
- pour désigner une ressource appartenant au même serveur que la ressource qui la référence : utilisation des URL relatives
 - ne contient ni protocole ni nom de domaine
- on peut aussi faire référence à une partie d'une page au sein de la même page

95

Les Liens

■ Exemples : Un lien

- sur une page du même répertoire
 - `Contact `
- sur une page contenue dans un sous-répertoire
 - `Contact `
- sur une page contenue dans le répertoire parent
 - `Contact `

96

Les Liens

- Liens pour les mails
 - une URL débutant par **mailto** demande au navigateur de lancer un client mail
 - l'adresse du destinataire est précisée après mailto
- Exemple :
 - `Écrivez-moi`
- Cible
 - par défaut, le clic sur un lien provoque le remplacement de la page précédente
 - l'attribut **target** permet de préciser où sera affiché la nouvelle page :
 - `target="_blank"` : une nouvelle fenêtre s'ouvre

97

Les Liens

- Liens vers une partie de la page
 - un lien peut pointer sur un élément particulier dans une page ou d'une autre page
 - cliquer sur ce type de lien provoque sa visualisation
 - très pratique pour des pages très longues
 - syntaxe :
 - `Haut de la page`
 - `Vers la référence`
 - ou ` Vers la référence `
 - ou ` Vers la référence `
 - la cible doit être identifiée par une balise contenant l'attribut **id**

98

Les images

■ Organisation

- un site utilise beaucoup d'images de tout type et de toute taille
- il est important de structurer leur stockage : dans un répertoire img, par ex
- les formats reconnus sont : jpeg, png, gif et svg

■ Critères de choix du format

- jpeg : idéal pour les photos et illustrations complexes contenant des millions de couleurs ; attention à la compression !
- gif : pratique pour des petites images (icône, par exemple) ; 256 couleurs max ; gestion de la transparence et de l'animation
- png : format sans perte ; 8 bits, 24 bits, 32 bits... ; possibilité de transparence
- svg : format vectoriel ; ensemble d'objets graphiques ; pas de perte de qualité ; plus léger et plus rapide

99

Les images

■ Comparatif pour une image transparente



100

Les images

- La balise `img`
 - l'insertion d'une image dans une page est possible à l'aide de la balise `img`
 - deux attributs :
 - `src` : URL de l'image (absolue ou relative)
 - `alt` : un texte alternatif en cas de non chargement de l'image
- Exemple
 - ``

101

Les images

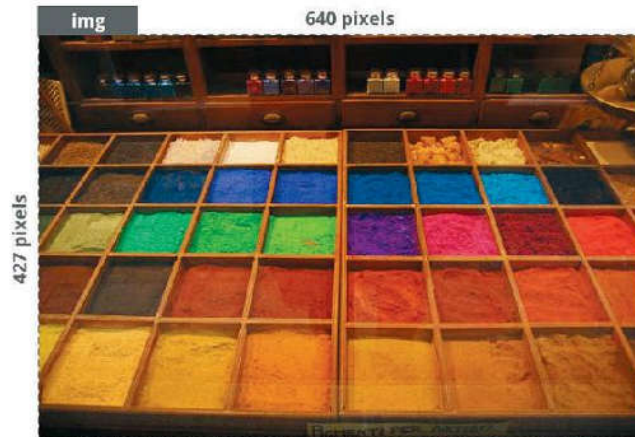
- Poids des images
 - faire attention au poids d'une image ! énorme impact sur le temps de chargement d'une page
 - jouer sur la résolution (72dpi, par exemple), la taille et le taux de compression (pour le format jpeg)
- Taille des images
 - par défaut, une image s'affiche avec sa taille (largeur / hauteur)
 - la largeur et la hauteur peuvent être modifiées via les attributs *width* et *height*
 - attention aux proportions !

102

Les images

- Exemple :

- ``



103

Les images

- Placement des images

- en HTML4 et inférieur, la balise *align* permet de spécifier la position relative d'une image vis à vis du texte
- cette balise est maintenant obsolète
- il faut utiliser les CSS

104

Les images

■ SVG : Inclusion HTML

– Depuis les débuts de SVG sur le Web

- La balise `<object>`

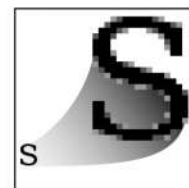
- Exemple: `<object type="image/svg+xml" data="image.svg" width="800" height="600"> <p> Texte alternatif : Pas de support SVC</p> </object>`

- La balise ``

- Exemple: ``

– En HTML5: La balise `<svg>`

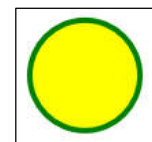
- Exemple : `<svg width="100" height="100">
<circle cx="50" cy="50" r="40" stroke="green"
stroke-width="4" fill="yellow" /> </svg>`



Raster
.jpeg .gif .png



Vector
.svg



105

Groupement

■ Définition

– les balises de groupement sont destinées à en grouper d'autres.

– deux balises :

- `div` : conteneur générique par excellence

- `<div>` représente une division de document.

- d'un point de vue sémantique, il n'a aucune valeur et n'est voué qu'à regrouper des éléments ayant des propriétés communes

- `Span` : conteneur spécifiquement dédié au contenu phrasé.

- il ne possède pas de sens particulier d'un point de vue sémantique.

- Il peut être utilisé également pour grouper des éléments ou une portion de texte à des fins d'affectation de styles ou de partage d'attributs.

106

Groupement

- Exemple d'utilisation de `<div>`:

```
<div>  
  <div>  
    <!-- Une première partie -->  
  </div>  
  <div>  
    <!-- Une seconde partie -->  
  </div>  
</div>
```

- Exemple d'utilisation de ``:

```
<p>HTML signifie <span lang="en"> Hypertext Markup  
Language</span> en anglais.</p>
```

107

Groupement

- Bonne pratique :
 - `<div>` et `` doivent en théorie n'être utilisés qu'en dernier recours lorsqu'il n'existe aucun autre élément plus approprié.

108

Structure des éléments en HTML5

- Éléments HTML sémantiques
 - HTML5 introduit de nouveaux éléments dits *sémantiques*.
 - Ces éléments permettent de renseigner sur leur fonction au sein du document Web afin d'en faciliter l'analyse par les robots des moteurs de recherche et, par conséquent, d'en améliorer le référencement.

109

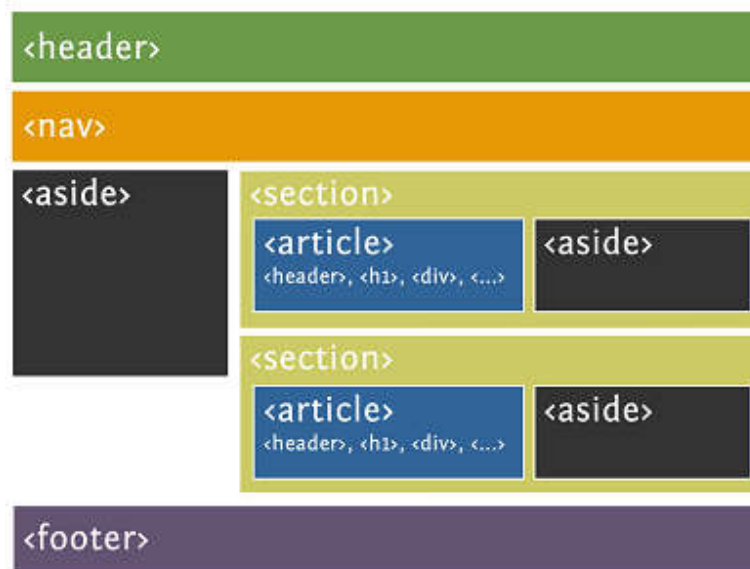
Structure des éléments en HTML5

- <header>
 - Entête du document Web où l'on retrouve en général un logo ainsi que le titre du document. .
- <nav>
 - Menu de navigation.
- <footer>
 - Pied de page du document Web.
- <section>
 - Section générique, regroupant une thématique de contenu, de préférence avec un en-tête.
- <article>
 - Section de contenu dans un document web, dont la composition peut être indépendante du reste de la page et extraite individuellement.
- <aside>
 - Section dont le contenu est tangentiellement lié à ce qui l'entoure, et qui peut être considérée comme séparé de ce contenu.

110

Structure des éléments en HTML5

■ Exemple



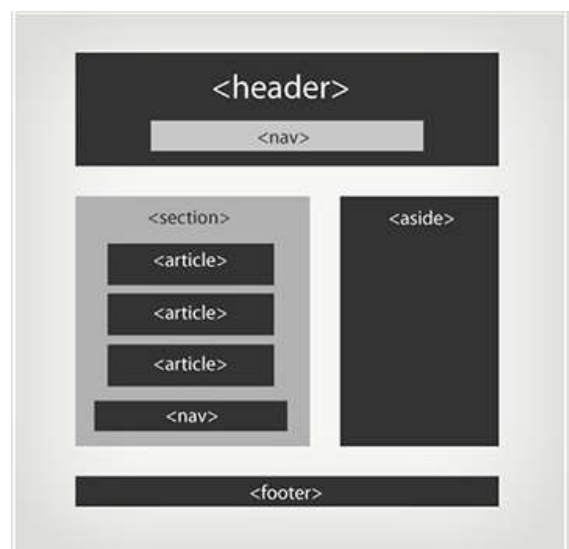
111

Exemple :

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title> Ma structure complète en HTML5</title>
  </head>
  <body>
    <!-- Début -->
    <header>
      <nav></nav>
    </header>
    <section>
      <article></article>
      <article></article>
      <article></article>
      <nav></nav>
    </section>
    <aside></aside>
    <footer></footer>
    <!-- Fin -->
  </body>
</html>

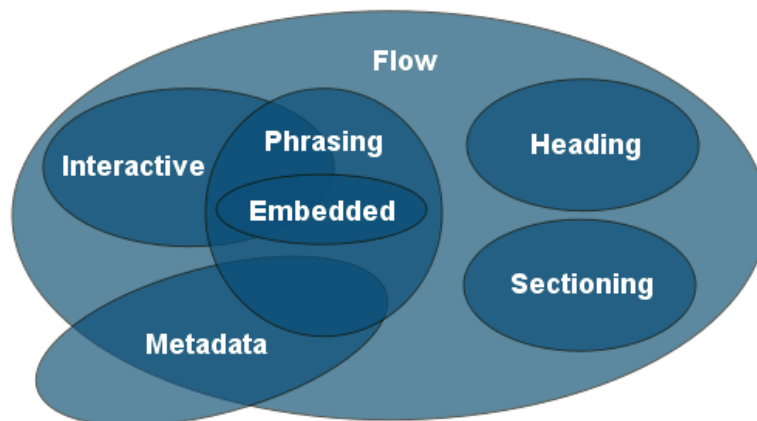
```



112

Différentes catégories d'éléments

Tous les éléments HTML5 sont regroupés dans des modèles de contenu (*Content models*) qui se déclinent dans 7 catégories, dont certaines se recouvrent mutuellement.



113

Différentes catégories d'éléments

- Flux (flow) :
 - regroupe la plupart des éléments courants (texte simple + éléments)
- *Heading* :
 - comprend tous les titres hiérarchiques (titres et sous-titres <h1> à <h6> et <hgroup>)
- Contenu sectionnant (*sectioning*)
 - définit les grandes zones du document HTML
 - Exemples : <article>, <aside>, <nav>, <section>
- Contenu de phrasé (*phrasing content*)
 - correspond à la plupart des éléments pouvant apparaître dans un flux de texte, et qui sont pour la plupart d'entre eux affichés en-ligne
 - Ex: span, strong, small, img

114

Différentes catégories d'éléments

- Contenu embarqué (*embedded*)
 - Ressources externes à inclure dans le document
 - Ex: audio, video, canvas, img, iframe, svg
- Contenu interactif
 - destiné à l'interaction avec l'utilisateur
 - Ex : a, input, textarea, label, menu, button
- Métadonnées (*metadata*) :
 - apportent des renseignements complémentaires au contenu ou agissent sur la présentation.
 - Ex: titre, style, script, link