

Développement Web

Les bases HTML et CSS

LI328

UPMC

Production partie visuelle du site

Langages
HTML et CSS



Traduction par
l'ordinateur

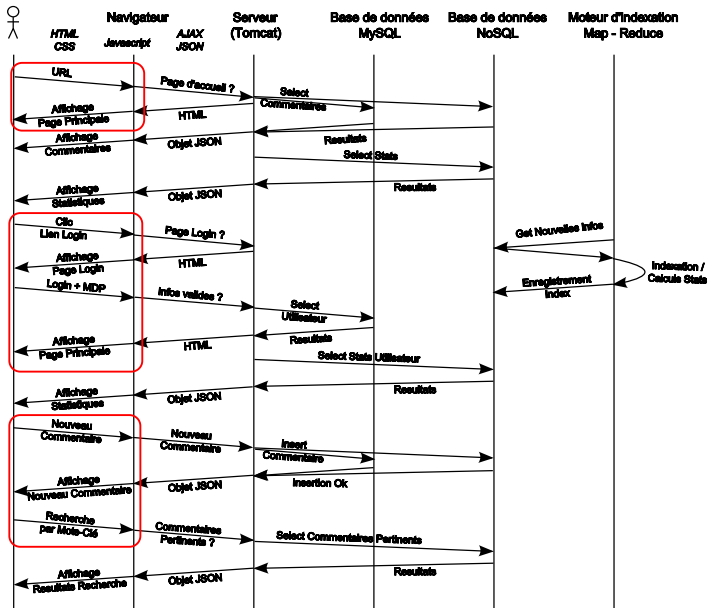


Résultat visible
à l'écran



- Interprétation côté client
- Pris en charge par le navigateur Web
 - Firefox
 - Internet Explorer
 - Google Chrome
 - etc...
- Correspond à la vue du modèle MVC

La Vue : Interface utilisateur



Deux langages complémentaires

- HTML (Hypertext Markup Language : langage de balisage d'hypertexte)
 - Définition des éléments graphiques de la page
 - Création d'hyperliens (liens entre pages Web)
 - Ensemble de balises <HTML> ... </HTML>
- CSS (Cascading Style Sheets : feuilles de style en cascade)
 - Propriétés visuelles des éléments de la page (agencement, positionnement, décoration, couleur, taille du texte...)
 - Ensemble de règles d'association propriété - valeur

Les langages HTML et CSS

HTML
(pas de CSS)



HTML + CSS



- Deux langages pour :
 - Séparer la structure de la page de son style de présentation
 - Décliner les styles selon les clients
 - Permettre des combinaisons (cascades) de styles

⇒ Développement en deux temps

- HTML pour créer les éléments à afficher...
- ... Puis CSS pour améliorer le rendu visuel

HTML

- Une des trois inventions à l'origine du World Wide Web :
 - Hypertext Markup Language (HTML)
 - Hypertext Transfer Protocol (HTTP)
 - Adressage web (URL)
- Origine : 1990
 - Simple structuration du texte en titres, sous-titres, listes ou texte brut
- Aujourd'hui : HTML 5
 - Prise en charge de nombreux medias / technologies riches
- Régi par le World Wide Web Consortium (W3C)

HTML : Ensemble de balises

- Une page HTML est un ensemble de balises
 - ⇒ Déclarer les éléments constitutants de la page
 - ⇒ En définir la structure
- Une balise est de la forme :
 $\langle \text{nom} \text{ attribut}_1 = \text{"valeur"} \text{ attribut}_2 = \text{"valeur"} \dots \rangle$
- Exemple : $\langle a \ href = \text{"www.lip6.fr"} \rangle$
- Certaines balises, $\langle \text{balise} \rangle \dots \langle / \text{balise} \rangle$, peuvent englober des éléments... D'autres, $\langle \text{balise} / \rangle$, non.

Code HTML

```
<HTML>
  <HEAD>
    <TITLE>Titre de la page</TITLE>
  </HEAD>

  <BODY>
    Contenu de la page
  </BODY>
</HTML>
```

- Différents types de balises
 - ⇒ Chaque balise à un rôle / comportement qui lui est propre
 - ⇒ Définies dans la DTD (Document Type Definition) utilisée
- Exemple de DTD :
<http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd>
- Déclaration DTD utilisée
 - ⇒ Pour validation W3C
 - ⇒ Par navigateur pour distinguer le type de page HTML

Déclaration DTD

- Différentes DTD selon la version HTML

HTML4

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

HTML5

```
<!DOCTYPE html>
```

XHTML1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

⇒ Pour bien choisir sa DTD :

<http://www.alsacreations.com/article/lire/560-dtd-doctype-html-xhtml-comment-choisir.html>

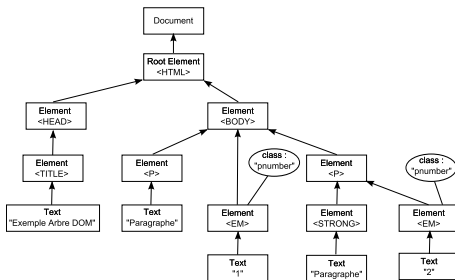
Visualisation de code HTML

The screenshot shows the homepage of 'le Site du Zéro', a job portal. The browser address bar displays 'www.siteduzero.com/emploi'. The site features a navigation bar with links like 'Informatique', 'Cours', 'Forums', 'Participez', 'Études', 'Emploi', and 'Boutique'. A sidebar on the left lists 'Top recruteurs' (SimpleIT, Flagtory, CommOnEc...) and 'Dernières annonces' (VIE/A, Stage, CDI, CDI). The main content area is titled 'Emploi' and includes a search bar with the text 'Recherche' and 'Entrez des mots clés séparés par des virgules pour affiner'. Below the search bar is a section 'Annonces à la une' (Featured Announcements) displaying a job listing for 'Développeur web Symfony senior (H/F)' by 'Simple IT' in Paris. A context menu is open over this listing, showing options like 'Back', 'Forward', 'Reload', 'Bookmark This Page', 'View Page Source', and 'Inspect Element (Q)'. The 'View Page Source' option is highlighted. The job listing itself shows 'Simple IT', 'Paris', 'CDI', and 'Mise en ligne : 06/01/2012'. A 'Consulter' button is visible. Below the featured announcement, it says '35 annonces'.

Exemple de code source

Arbre DOM

- L'ensemble de balises d'un fichier HTML peut se représenter sous forme d'arbre
- ⇒ L'arbre DOM (Document Object Model)



Code HTML

```
<HTML>
<HEAD>
  <TITLE>Exemple Arbre DOM</TITLE>
</HEAD>
<BODY>
  <p> Paragraphe <em class="pnumber">1 </em> </p>
  <p> <strong> Paragraphe </strong> <em class="pnumber">2 </em> </p>
</BODY>
</HTML>
```

- Différents types de balises
 - Balises de premier niveau (html, head, body)
 - Balises d'en-tête (title, meta, link, etc...)
 - Balises de contenu (p, a, br, img, etc...)
 - Balises de liste (ul, li, ol, etc...)
 - Balises de tableau (table, td, tr, etc...)
 - Balises de formulaire (form, input, button, etc...)
 - Balises de structure (frameset, div, span, header, etc...)

- Balises de premier niveau
 - `< html > ... < /html >` : Balise principale. Englobe l'ensemble du code (sauf déclaration doctype).
 - `< head > ... < /head >` : En-tête de page. Regroupe les balises d'entête (déclaration titre, mots-clé, fichiers de style, etc...).
 - `< body > ... < /body >` : Corps de la page. Contient l'ensemble du contenu à afficher.

- Balises d'en-tête

- `< title > ... < /title >` : Titre de la page
- `< style > ... < /style >` : Permet d'inclure du code CSS
- `< link / >` : Permet d'indiquer certaines informations
 - ⇒ Inclure une page de style :
`< link rel = "stylesheet" href = "style.css" / >`
 - ⇒ Déclarer un icône pour le site :
`< link rel = "shortcut icon" type = "image/x-icon" href = "icon.ico" / >`
 - ⇒ Déclaration de fils RSS, page accueil site, fichier d'aide, etc...
- `< script > ... < /script >` : Permet d'inclure un script
 - ⇒ Du code :
`< script type = " text /javascript" > ... < /script >`
 - ⇒ Un fichier script :
`< script type = " text /javascript" src = "script.js" >< /script >`

HTML : Balises d'en-tête

- `< meta / >` permet de :

⇒ Définir différentes propriétés

- Auteur de la page :

`< meta name = "author" content = "Sylvain Lamprier" / >`

- Description de la page :

`< meta name = "description" content = "Cours Technos Web" / >`

- Mots-clé de la page :

`< meta name = "keywords" content = "Web, HTML, CSS, Javascript, AJAX, Tomcat" / >`

- Adresse de contact :

`< meta name = "reply - to" content = "Sylvain.Lamprier@lip6.fr" / >`

⇒ Spécifier l'encodage des caractères :

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

⇒ Empêcher la mise en cache de la page :

`< meta http - equiv = "pragma" content = "no - cache" / >`

⇒ Forcer le rafraîchissement régulier de la page :

`< meta http - equiv = "refresh" content = "10; URL = http : // www.lip6.fr" / >`

- Balises de contenu

- `<p> ... </p>` : Paragraphe
- `<pre> ... </pre>` : Paragraphe affiché tel qu'il a été tapé dans le code
- `<h$> ... </h$>` : Titre de niveau \$ (avec $\$ \in \{1..6\}$)
- ` ... ` : Hyperlien vers une page située à l'adresse \$
- `
` : Passage à la ligne
- `<hr />` : Ligne de séparation horizontale
- `` : Insertion d'une image
``
- `^{...}` : Mise en exposant
- `_{...}` : Mise en indice
- Nombreuses autres balises de mise en forme ``, ``, `<cite>`, `<blockquote>`, etc...

Code HTML

```
<!doctype html>
<html>
  <head>
    <title > Exemple de page HTML </title >
  </head>
  <body>
    <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>
    Ici les      multiples espaces et

    retours à la ligne ne sont pas pris en compte. <br /> Pour passer
    à la ligne , utiliser &lt; br /&gt;;
    <p> ou placer dans un paragraphe. </p>
    <pre> Ici par contre,      les
    espaces et retours à la ligne
    sont      pris en compte car on est dans
    un bloc &lt; pre &gt;;. </pre>
    <hr />
    Enfin , voici un <a href="Exemple_balises_contenu2.html"> hyperlien </a>

  </body>
</html>
```

Titre de niveau 1

Titre de niveau 2

Titre de niveau 3

Titre de niveau 4

Titre de niveau 5

Titre de niveau 6

Ici les multiples espaces et retours à la ligne ne sont pas pris en compte.
Pour passer à la ligne, utiliser `< br />`

ou placer dans un paragraphe.

```
Ici par contre,          les
    espaces et retours à la ligne
    sont          pris en compte car on est dans
    un bloc < pre >.
```

Enfin, voici un [hyperlien](#)

- Balises de listes

- ` ... ` : Liste à puces non numérotée
- ` ... ` : Liste à puces numérotée
- ` ... ` : Item de liste à puces
- `<dl> ... </dl>` : Liste de définitions
- `<dt> ... </dt>` : Terme à définir
- `<dd> ... </dd>` : Définition

Code HTML

```
<!doctype html>
<html>
  <head>
    <title> Exemple de page HTML </title>
  </head>
  <body>
    <h3> Liste Non Numerotée </h3>
    <ul>
      <li> Element 1 </li>
      <li> Element 2 </li>
      <li> Element 3 </li>
    </ul>

    <h3> Liste Numerotée </h3>
    <ol>
      <li> Element 1 </li>
      <li> Element 2 </li>
      <li> Element 3 </li>
    </ol>

    <h3> Liste Définitions </h3>
    <dl>
      <dt> Terme 1 </dt>
      <dd> Definition terme 1 </dd>
      <dt> Terme 2 </dt>
      <dd> Definition terme 2 </dd>
    </dl>
  </body>
</html>
```

Liste Non Numerotée

- Element 1
- Element 2
- Element 3

Liste Numerotée

1. Element 1
2. Element 2
3. Element 3

Liste Définitions

- Terme 1
Definition terme 1
- Terme 2
Definition terme 2

- Balises de tableau
 - `< table > ... < /table >` : Déclare un tableau
 - `< caption > ... < /caption >` : Titre du tableau
 - `< tr > ... < /tr >` : Ligne du tableau
 - `< td > ... < /td >` : Case du tableau
 - Attribut `colspan=n` : la case occupe n colonnes
 - Attribut `rowspan=n` : la case occupe n lignes
 - `< th > ... < /th >` : Case en-tête

Code HTML

```
<!doctype html>
<html>
  <head>
    <title> Exemple de page HTML </title>
  </head>
  <body>
    <table>
      <caption>Tableau </caption>
      <tr>
        <th>Colonne 1</th>
        <th>Colonne 2</th>
        <th>Colonne 3</th>
      </tr>
      <tr>
        <td> Case 1,1 </td>
        <td> Case 1,2 </td>
        <td> Case 1,3 </td>
      </tr>
      <tr>
        <td> Case 2,1 </td>
        <td> Case 2,2 </td>
        <td> Case 2,3 </td>
      </tr>
    </table>
  </body>
</html>
```

| Tableau | | |
|-----------|-----------|-----------|
| Colonne 1 | Colonne 2 | Colonne 3 |
| Case 1,1 | Case 1,2 | Case 1,3 |
| Case 2,1 | Case 2,2 | Case 2,3 |

- Balises de formulaire

- `< form > ... < /form >` : Déclare un tableau
- `< label > ... < /label >` : Titre d'un élément de formulaire
- `< input / >` : Champ de formulaire
 - ⇒ Zone de texte d'une ligne `< input type = "text" / >`
 - ⇒ Mot de passe `< input type = "password" / >`
 - ⇒ Case à cocher `< input type = "checkbox" / >`
 - ⇒ Envoi de fichier `< input type = "file" / >`
 - ⇒ Bouton d'option `< input type = "radio" / >`
 - ⇒ Bouton d'envoi `< input type = "submit" / >`
 - ⇒ Bouton de remise à zéro `< input type = "reset" / >`
 - ⇒ Champ caché `< input type = "hidden" / >`
- `< textarea > ... < /textarea >` : Zone de saisie multiligne
- `< select > ... < /select >` : Liste déroulante
- `< option > ... < /option >` : Element de liste déroulante

HTML : Balises de formulaire

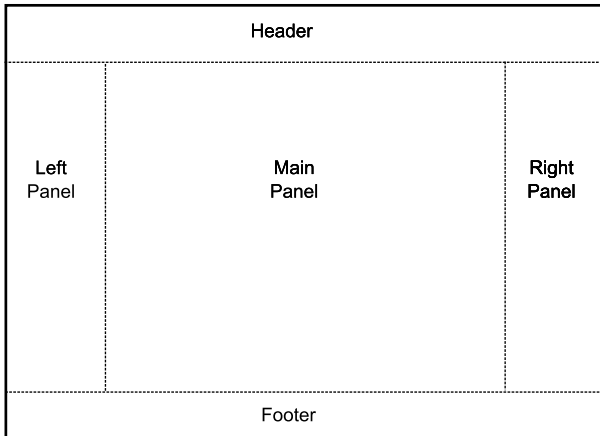
Code HTML

```
<html>
<body>
<FORM method=post action="cgi-bin / Enregistrement . pl">
  <TABLE BORDER=0>
    <TR><TD>Nom</TD><TD><INPUT type=text name="nom"></TD></TR>
    <TR><TD>Mot de Passe</TD>
      <TD><INPUT type="password" name="pass"></TD></TR>
    <TR><TD>Sexe</TD><TD>
      Homme : <INPUT type=radio name="sexe" checked value="M"> <br/>
      Femme : <INPUT type=radio name="sexe" value="F"></TD></TR>
    <TR><TD>Fonction</TD>
      <TD><SELECT name="fonction">
        <OPTION VALUE="enseignant">Enseignant</OPTION>
        <OPTION VALUE="etudiant">Etudiant</OPTION>
      </SELECT></TD></TR>
    <TR><TD>Commentaires</TD><TD>
      <TEXTAREA rows="3" name="commentaires">Tapez ici vos commentaires
      </TEXTAREA></TD></TR>
    <TR><TD COLSPAN=2>
      <INPUT type="submit" value="Envoyer">
      <INPUT type="reset" value="Rétablir ">
    </TD></TR>
  </TABLE>
  <INPUT type="checkbox" name="rester" value="true"/> Rester connecté
</FORM>
</body>
</html>
```

| | |
|--|---|
| Nom | <input type="text" value="Lamprier"/> |
| Mot de Passe | <input type="password" value="•••••"/> |
| Sexe | Homme : <input checked="" type="radio"/> Femme : <input type="radio"/> |
| Fonction | <input type="text" value="Enseignant"/> |
| Commentaires | <div>Tapez ici vos commentaires</div> |
| <input type="button" value="Envoyer"/> <input type="button" value="Rétablir"/> | |
| <input type="checkbox"/> Rester connecté | |

- Balises de structure
 - Servent à structurer les pages
 - Disposition des éléments les uns par rapport aux autres
- Différentes balises
 - `< frameset >`, `< frame>` pour une structure par cadres
 - `< div >`, `< span >` : boîtes génériques permettant de regrouper des éléments
 - Balises structurantes `< header >`, `< footer >`, `< aside >`, etc...

Structuration de pages Web



- Différentes possibilités
 - Structuration par tableau
 - Utilisation de cadres (frames)
 - Utilisation du "flux" naturel
 - Usage de balises structurantes
 - Positionnement / Structuration CSS

Structuration par tableau

Code HTML

```
<!doctype html>
<html>
  <head>
    <title> Structuration par tableaux </title>
  </head>
  <body>
    <TABLE BORDER=1 style="width:100%;height:500px">
      <TR ALIGN="center" style="height:20%"><TD colspan="3"> Header </TD> </TR>
      <TR ALIGN="center" style="height:60%">
        <TD style="width:20%"> Left <br /> Panel </TD>
        <TD style="width:60%">Main Panel</TD>
        <TD style="width:20%"> Right <br /> Panel </TD></TR>
      <TR ALIGN="center" style="height:20%"><TD colspan="3"> Footer </TD> </TR>
    </TABLE>
  </body>
</html>
```

| Header | | |
|------------|------------|-------------|
| Left Panel | Main Panel | Right Panel |
| Footer | | |

- Possibilité d'imbriquer des tableaux pour structures plus complexes
- Structuration par tableau à éviter
 - ⇒ Structure figée (difficilement modifiable, problèmes de portabilité)
 - ⇒ Coûteux en ressources machine
 - ⇒ Code complexe

(Autre exemple sur slide "HTML: Balises de formulaire")

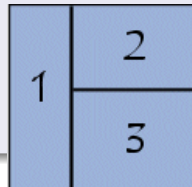
Structuration par Frames

- Code page distribué sur différents fichiers
 - Un fichier principal décrivant la structure
 - Des fichiers à afficher dans les différents cadres
- Deux balises :
 - `< frameset > ... < /frameset >` : définit la manière de disposer les cadres
 - Attribut `rows="x1, x2, ..., xn"` : Dispose les cadres les uns en dessous des autres (les valeurs x correspondent aux hauteurs des différents cadres)
 - Attribut `cols="x1, x2, ..., xn"` : Dispose les cadres les uns à côté des autres (les valeurs x correspondent aux largeurs des différents cadres)
 - `< frame/ >` : Correspond à un cadre L'attribut `src` permet de spécifier la fichier html à charger dans le cadre
L'attribut `name` permet de donner un nom au cadre
- Si un cadre correspond à un menu ses liens doivent spécifier le cadre cible : `< a href = " accueil.html" target = " main" >`, où `main` correspond au cadre principal

Structuration par Frames

Exemple de structuration par Frames

```
<frameset cols="30%,*">
  <frame src="menu.html">
  <frameset rows="40%,*">
    <frame src="top.html">
    <frame src="main.html">
  </frameset>
</frameset>
```



- Structuration par frames à éviter
 - Rendu peu esthétique / démodé
 - Pages mal indexées par moteurs de recherche
 - Problèmes d'impression, de navigation, d'accessibilité
- Exemples de sites avec Frames :
 - www.angelfire.com/super/badwebs/
 - www4.tripnet.se/~slarti/FrameEx/Bad/Uk/Top/frameset.htm

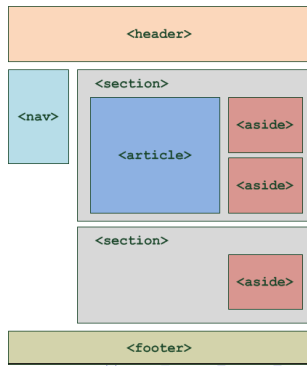
Structuration selon le flux naturel

- Flux naturel => Comportement par défaut des navigateurs
- En flux naturel, deux facteurs sont considérés pour positionner les éléments :
 - L'ordre d'apparition des balises
 - ⇒ Détermine l'ordre d'apparition des éléments
 - Le type des balises
 - ⇒ Détermine la manière dont s'enchainent les éléments (horizontalement ou verticalement)
- Quatre grands types de balises :
 - inline
 - block
 - inline-block
 - none
- Possibilité de changer le type d'une balise par la propriété CSS "display"

35 / 73

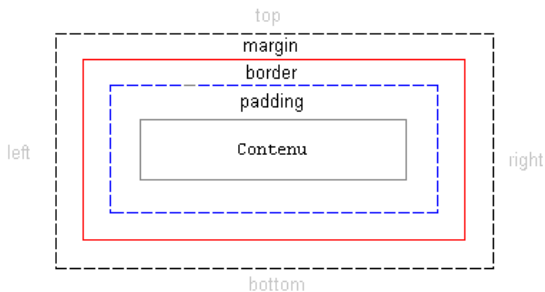
Balises structurantes

- Depuis HTML 5.0, nouvelles balises de type block :
 - `<header> ... </header>` pour définir l'en-tête de la page
 - `<footer> ... </footer>` pour définir le pied de page
 - `<nav> ... </nav>` visant à contenir des liens de navigation
 - `<section> ... </section>` pour regrouper du contenu (selon une thématique par exemple)
 - `<article> ... </article>` pour englober une portion autonome de la page (article de journal par exemple)
 - `<aside> ... </aside>` pour apporter des informations complémentaires
- Structure envisageable avec ces balises :
- **Attention !** Rien n'est défini par défaut, Tout reste à positionner avec CSS
- Equivalent à utiliser des `<div>` partout...
- ⇒ Mais meilleure lisibilité du code
- ⇒ Évolutions possibles des navigateurs
- ⇒ Structure plus riche



- CSS offre de nombreuses possibilités pour
 - Personnaliser les éléments (couleurs, taille, formes...)
 - Réorganiser les éléments (sortir du flux normal pour positionnements spéciaux)
- Différentes propriétés...
 - couleur texte, couleur fond
 - police du texte, taille des caractères
 - taille bordure
 - espacements entre éléments
 - espacements entre bordure et contenu
 - ...
- ... Mais aussi différents types de positionnement :
 - static
 - absolute
 - relative
 - fixe
 - float

Le modèle de boîtes



- bord de la marge
- bord de la bordure
- - - bord de l'espacement
- bord du contenu

CSS : Appliquer un style aux éléments de la page

Plusieurs possibilités

CSS dans balise

```
<p style="color: blue;font-size:10px"> Texte bleu de taille 10px </p>
```

CSS dans head du fichier HTML

```
<html>  
  <head>  
    <style>  
      Code CSS  
    </style>  
  </head>  
</html>
```

CSS dans un ou plusieurs fichiers séparés (recommandé)

```
<html>  
  <head>  
    <link rel="stylesheet" href="style1.css" />  
  </head>  
</html>
```

CSS : Ensemble de règles d'application de propriétés

- Un fichier .css est composé de blocs contenant :
 - Un sélecteur
 - Un ensemble de couples propriété-valeur

Bloc CSS générique

```
selecteur
{
    propriété1 : valeur1 ;
    propriété2 : valeur2 ;
}
```

Bloc CSS exemple

```
p
{
    color : blue ;
    font-size : 10px ;
}
```


- Sélecteurs

- Sélecteur = Motif

⇒ Repérage des éléments auxquels appliquer les propriétés correspondantes

- Différents types

- Sélecteur universel `*{...}`. S'applique à tous les éléments.

- Noms de balise

Exemples : `p{...}` `h1{...}` `h1,h2{...}`

- Identifiants d'éléments (spécifiés par attribut "id" dans la balise de l'élément). Sélecteur = `#id`

Exemple : `< p id = "intro" >` ⇒ `#intro{...}`

- Classes d'éléments (spécifiés par attribut "class" dans la balise des éléments). Sélecteur = `.class`

Exemple : `< p class = "script" >` ⇒ `.script{...}`

- Sélecteurs de base
 - * : tout élément
 - E : tout élément de type E (correspondant au motif E)
 - E.x : tout élément de type E appartenant à la classe x
 - E#x : tout élément de type E d'identifiant x
 - E, F : tout élément de type E ou F

● Sélecteurs avancés

- $E F$: tout élément F qui est un descendant d'un élément E
- $E > F$: tout élément F qui est un enfant direct d'un élément E
- $E:first-child$: tout élément E premier enfant d'un autre élément
- $E + F$: tout élément F directement précédé d'un élément E
- $E[x]$: tout élément E possédant un attribut x
- $E[x="y"]$: tout élément E possédant un attribut x de valeur "y"
- $E[x\sim="y"]$: tout élément E possédant un attribut x contenant la sous chaîne "y"
- $E[x|= "y"]$: tout élément E possédant un attribut x commençant par la sous chaîne "y"
- $E:link$, $E:visited$: lien E n'ayant pas encore été visité ($:link$) / ayant déjà été visité ($:visited$)
- $E:active$, $E:hover$, $E:focus$: élément E activé ($:active$), pointé ($:hover$), recevant l'attention ($:focus$)

- Différents types de propriétés
 - Propriétés des conteneurs (boîtes)
 - Propriétés textuelles
 - Propriétés des listes
 - Propriétés de tableaux
 - Propriétés de positionnement

- Dimensions du cadre de contenu (uniquement pour block ou inline-block)
 - width, height : largeur, hauteur (en % de la taille du parent ou en px). Par défaut :
 - La hauteur correspond à la taille des éléments contenus
 - La largeur est maximale pour les blocks
 - La largeur dépend du contenu pour les inlines-block
 - min-width,max-width : largeurs minimales et maximales
 - min-height, max-height : hauteurs minimales et maximales (en cas de redimensionnement)
- ⇒ Attention aux dépassements de boîte !

- Marges Extérieures (entre bordure et extrémité de la boîte)
 - margin-left, margin-right : marges extérieures horizontales
 - ⇒ Décalage horizontal selon adjacents
 - margin-top, margin-bottom : marges extérieures verticales
 - ⇒ Dans le cas des block et inline-block : décalage vertical selon adjacents
 - ⇒ Dans le cas des inline : aucun effet
- Marges Intérieures (entre contenu et bordure)
 - padding-left, padding-right : marges intérieures horizontales
 - ⇒ Agrandissement horizontal du cadre et décalage des adjacents
 - padding-top, padding-bottom : marges intérieures verticales
 - ⇒ Dans le cas des inline : agrandissement vertical du cadre sans décalage des adjacents
 - ⇒ Dans le cas des block et inline-block : agrandissement vertical du cadre avec décalage des adjacents

- Bordures

- border-width : Épaisseur de la bordure
- border-style : Style de bordure
 - none : pas de bordure (par défaut)
 - solid : trait plein
 - double : ligne double
 - dashed : tirets
 - dotted : pointillés
 - inset, outset, ridge : styles 3D
- border-color : couleur de bordure (soit nom de couleur, soit Hexa, soit code RGB)
- border-top, border-right, border-bottom, border-left :
épaisseur - style - color pour les 4 différentes bordures.
Exemple : *border-left : 2px solid blue;*
- border-radius : 4 valeurs d'arrondi pour chacun des coins,
en partant de celui en haut à gauche

- Affichage

- visibility : “hidden” (caché) ou “visible”
- clip : permet (seulement pour les boîtes positionnées en absolu) de rogner la boîte selon le rectangle défini par rect(<haut>,<droite>, <bas>,<gauche>) où :
 - < haut > = décalage par rapport au bord haut de la boîte
 - < gauche > = décalage par rapport au bord gauche de la boîte
 - < bas > – < haut > = taille verticale du rectangle
 - < droite > – < gauche > = taille horizontale du rectangle
- box-shadow : Permet de donner une ombre à la boîte.
Valeur = x y z c, avec :
 - x = Décalage vertical en px
 - y = Décalage horizontal en px
 - z = Adoucissement en px
 - c = Couleur

Exemple : *box-shadow: 2px 2px 4px black*

- Affichage

- opacity : valeur de 0 (boîte transparente) à 1 (boîte opaque)
- background-color : couleur de fond
- background-image : image de fond
Exemple : *background-image: url("images/fond.png");*
- background-attachment : fixed (le fond reste fixe) ou scroll (le fond défile avec la page)
- background-repeat : répétition du fond
 - repeat (par défaut) : le fond se répète
 - no-repeat : pas de répétition
 - repeat-x : répétition sur une ligne uniquement
 - repeat-y : répétition sur une colonne uniquement
- background-position : positionnement du fond
 - En px ou en % par rapport au bord gauche puis haut
Exemple : *background-position: 50px 20px;*
 - Alignement horizontal (left, center, right) puis vertical (top, middle, bottom)
Exemple : *background-position: left top;*

- Comportement

- overflow : permet de spécifier un comportement si le contenu dépasse les dimensions de la boîte
 - visible : tout l'élément sera affiché (par défaut)
 - hidden : élément coupé s'il dépasse les limites de la boîte
 - scroll : élément coupé mais barres de défilement pour voir la partie cachée
- cursor: permet de donner le style de la souris lorsque l'on passe sur la boîte :
 - "default" (curseur standard)
 - "pointer" (curseur en forme de main)
 - "text" (curseur texte)
 - "wait" (sablier)
 - "progress" (curseur avec sablier)
 - "help" (point d'interrogation)
 - "move" (croix de déplacement)
 - ...

- Propriétés textuelles

- ⇒ Définir des propriétés pour le texte contenu
- ⇒ Propriétés héritées (une propriété textuelle se propage jusqu'aux feuilles)

- Propriétés de police

- font-family: police1, police2, police3, ...
 - ⇒ Permet de spécifier la police à utiliser
 - ⇒ Si police1 non disponible, police2
 - ⇒ Si police2 non disponible, police3
 - ⇒ Exemple : *font-family: "Arial Black", Arial, Verdana, serif;*
- font-size : taille de police
 - ⇒ En pixels (px) : taille fixe
 - ⇒ En cadratins (em) : taille relative à la taille de police du navigateur
 - ⇒ En pourcentages (%) : taille relative à un standard (100% = normal)
- color : couleur du texte (soit nom de couleur, soit Hexa, soit code RGB)

- Propriétés de police
 - font-weight :
 - bold (en gras), bolder (plus gras), normal ou lighter (plus fin)
 - font-style :
 - italic (italique), oblique (autre italique) ou normal
 - text-decoration :
 - underline : souligné
 - overline : ligne au dessus
 - line-through : barré
 - blink : clignotant
 - none : normal
 - text-transform :
 - uppercase : tout mettre en majuscules
 - lowercase : tout mettre en minuscules
 - capitalize : début des mots en majuscules
 - none : normal

- Mise en forme

- text-indent : indentation du paragraphe en pixels
- white-space : passage à la ligne
 - normal : passage à la ligne automatique en fin de boîte block ou inline-block
⇒ La propriété additionnelle "word-wrap:break-word;" autorise la césure de mots
 - nowrap : pas de passage à la ligne automatique
 - pre : passage à la ligne selon texte source (comme <pre>)
 - L'ajout de la propriété "word-wrap:break-word;" permet de forcer la césure en cas de dépassement de boîte (autorise le coupage de mots)
- text-shadow : Permet de donner une ombre au texte. Valeur = x y z c, avec :
 - x = Décalage vertical en px
 - y = Décalage horizontal en px
 - z = Adoucissement en px
 - c = Couleur

Exemple : *text-shadow: 2px 2px 4px black*

- **line-height** : Permet de spécifier la hauteur des lignes de texte contenues
 - ⇒ Pas d'effet sur les propriétés d'une boîte inline
 - Augmente uniquement l'espacement des lignes
 - Peut permettre de donner des marges entre boîtes inline
 - ⇒ Mais joue sur la hauteur des block et inline-block qui contiennent du texte (si height:auto)
- **text-align** : Permet de donner un alignement horizontal au texte contenu dans un block ou inline-block
 - Valeurs = left, right, center ou justify
 - ⇒ Attention : affecte la position des boîtes inline-block et inline contenues

CSS : Propriétés des listes

- `list-style-type` : type de puces
 - Pour listes non numérotées (``) :
 - `disc` : un disque plein (défaut)
 - `circle` : un cercle
 - `square` : un carré
 - `none` : aucune puce
 - Pour listes numérotées (``) :
 - `decimal` : des nombres 1,2,3,... (défaut)
 - `decimal-leading-zero` : des nombres 01,02,03,...
 - `upper-roman` : numérotation romaine en majuscules (I,II,...)
 - `lower-roman` : numérotation romaine en minuscules (i,ii,...)
 - `upper-alpha` : numérotation alphabétique en majuscules
 - `lower-alpha` : numérotation alphabétique en minuscules
 - `lower-greek` : numérotation grecque
- `list-style-image` : image pour puces de listes non numérotées.
Exemple : `list-style-image:url("images/puce.png");`
- `list-style-position` : appliquer un retrait
 - `outside` : retrait (défaut)
 - `inside` : pas retrait

- `caption-side` : position du titre (balise `<caption>`)
 - `top` : en haut du tableau
 - `bottom` : en bas du tableau
 - `left` : à gauche du tableau
 - `right` : à droite du tableau
- `empty-cells` :
 - `collapse` : les bordures des cellules vides sont masquées (défaut)
 - `show` : les bordures des cellules vides sont affichées
- `border-collapse` :
 - `separate` : les bordures du tableau et des cellules sont séparées (défaut)
 - `collapse` : les bordures du tableau et des cellules sont mélangées

- `position:static;`
 - Positionnement normal (par défaut)
 - A préférer quand c'est possible
- Autres positionnements
 - `position:absolute;`
 - `position:relative;`
 - `position:fixe;`
 - + Positionnement flottant

CSS : Propriétés de positionnement

- Positionnement absolu
 - position:absolute;
 - ⇒ Par rapport à boîte parent positionnée en absolu ou en relatif si existe
 - ⇒ Par rapport à la fenêtre du navigateur
- Distances aux bords en px (fixe) ou en % (étirable)
 - top : distance au bord haut du parent
 - bottom : distance au bord bas du parent
 - left : distance au bord gauche du parent
 - right : distance au bord droit du parent
- **Attention !** En absolu :
 - Element hors du flux (ne prend pas de place dans le flux naturel)
 - Par défaut, pas de hauteur ni de largeur
 - Définies par la conjonction d'alignements opposés
 - ... ou par height et width (inline dimensionnables)
 - ... ou par le contenu si non positionné en absolu ou fixe

- Positionnement relatif
 - `position:relative;`
 - ⇒ Par rapport à la position normale
- Distances à la position en px (fixe)
 - `top` : décalage vers le bas
 - `bottom` : décalage vers le haut
 - `left` : décalage vers la droite
 - `right` : décalage vers la gauche
- **Attention !** En relatif :
 - Element garde la place initiale occupée dans le flux
 - Hauteurs et largeurs comme en statique
- Peu utile sauf pour
 - Corrections de positionnement mineures
 - Servir de référent à un enfant absolu

- Positionnement fixe
 - `position:fixed;`
 - ⇒ Par rapport à la fenêtre du navigateur
 - ⇒ Le défilement de la page n'a aucun effet sur un contenu en position fixe
- Distances à la position en px (fixe) ou en % (étirable)
 - `top` : distance au bord haut du navigateur
 - `bottom` : distance au bord bas du navigateur
 - `left` : distance au bord gauche du navigateur
 - `right` : distance au bord droit du navigateur
- **Attention !** En fixe :
 - Element hors du flux (ne prend pas de place dans le flux naturel)
 - Par défaut, pas de hauteur ni de largeur
 - Définies par la conjonction d'alignements opposés
 - ... ou par `height` et `width` (inline dimensionnables)
 - ... ou par le contenu si non positionné en absolu ou fixe

CSS : Propriétés de positionnement

- Positionnement flottant

- Pas défini avec propriété position mais avec float
- Deux valeurs possibles :
 - left: Element flottant à gauche de son conteneur
 - right: Element flottant à droite de son conteneur
- Un élément flottant prend le comportement d'un inline-block...
- Sauf que les éléments qui le suivent peuvent s'enrouler autour

- Pour éviter de s'enrouler autour d'un flottant :

- Propriété clear
 - left : Contre un effet flottant à gauche
 - right : Contre un effet flottant à droite
 - both : Contre tout effet flottant

⇒ Avant tout pour répondre à un besoin typographique
précis: la création d'habillages

