* 1. TP n°2 : PROGRAMMATION CSS

.

|  |
| --- |
| Tout langage de programmation possède un lexique que l’on appelle « Reference ».Vous trouverez ci-joint le Reference pour HTML 5 .   * <https://www.w3schools.com/css/default.asp> |

|  |  |
| --- | --- |
| Exercice préliminaire : Modifiez le code sur le lien ci-dessous pour avoir.   * **le titre :**   + Couleur rouge.   + Taille de police 80pix.   + Style de caractère italique. * **les paragraphes :**   + Couleur bleu.   + En gras.   **le fond:**   * + Couleur grise.   Accédez à l’url : <https://www.w3schools.com/css/default.asp> |  |

**Q1.** Expliquer ce que signifie le mot sélecteur et citer les différents sélecteurs utilisés dans le code précédent.

Un **sélecteur** CSS est un motif qui cible les éléments HTML pour leur appliquer des styles.

**Sélecteurs utilisés dans le code :**

1. **body** - Cible l'élément <body> (fond bleu clair)
2. **h1** - Cible les titres <h1> (couleur blanche, centré)
3. **p** - Cible les paragraphes <p> (police Verdana, taille 80px)

|  |  |
| --- | --- |
|  | **Q2.** Utiliser votre vocabulaire en anglais et visualiser les différents exemples fournis sur le site afin de tester les différentes propriétés.  *Indice:Font,Text,Background,color* |

|  |  |
| --- | --- |
| **Indice** | **Explication** |
| Font | **Propriétés qui contrôlent l'apparence du texte : font-family (type de police), font-size (taille), font-weight (gras), font-style (italique)** |
| Text | **Propriétés qui gèrent l'affichage du texte : text-align (alignement), text-decoration (soulignement), text-transform (majuscules/minuscules), line-height (hauteur de ligne)** |
| color | **Propriété qui définit la couleur du texte : color (couleur du texte), background-color (couleur de fond)** |
| Background | **Propriétés qui contrôlent l'arrière-plan : background-color (couleur), background-image (image), background-size (taille), background-position (position)** |

## /!\ Important /!\

Une bonne pratique dans l’organisation de son travail est essentielle. Créer dans votre espace personnel un dossier organisé comme suivant.

TP2

* DATA
* HTML
* CSS

Pour la suite du tp, créer dans le dossier CSS un fichier style.css dans ce dossier

* 1. **Q3.** Nous allons travailler avec un fichier css séparé pour cela il faut
  2. Chercher comment faire du code css dans un fichier séparé.
  3. mots clés: link, stylesheet
  4. Exercice 1 :

Grâce à votre fichier nommer exercice 1.html générer une page CSS et rédiger le contenu afin d’obtenir le résultat suivant.

**Exemple :**

|  |
| --- |
|  |

**Q3.** Une autre méthode pour optimiser notre code, nous pouvons regrouper les sélecteurs en les séparant d’une virgule (on peut utiliser 1,2,3,4...n sélecteur).

exemple: <https://www.w3schools.com/css/tryit.asp?filename=trycss_grouping>

Modifier et commenter votre code (en css les commentaires se place avec « /*\* » et « \*/ »).*

Le cas où l’on ne met pas de virgule, nous permettra d’imposer un style sur un élément imbriqué. Dans le cas suivant tous les éléments « a » dans les éléments « p »

**Exemple**

|  |  |
| --- | --- |
| HTML | CSS |
| <!DOCTYPE html>  <html>  <head>  <link rel="stylesheet" href="./css/styles.css">  <title>Page Title</title>  </head>  <body>  <p>Pour plus d'infos : <a href="mailto:san@antonio.net"> Contactez moi </a></p>  </body>  </html> | p{  color:red;  }  p a{  color:orange;  background-color: black  } |

Tester ce code.

Afin de comprendre les différents jumelages de sélecteur, ajouter cette partie de code au précédent et tester les différentes assemblage et en déduire leur signification :

|  |
| --- |
| Html |
| <h2>Un titre de niveau 2</h2>  <!--Note : Utiliser href="#" permet de créer un lien HTML valide mais  qui ne mène nulle part (qui est "vide"), ce qui est parfois utile-->  <a href="#">Un lien</a>  <p>Un premier paragraphe</p>  <a href="#">Un autre lien</a>  <p>Un autre paragraphe avec un <a href="#">lien</a></p>  <ul>  <li><a href="#">Elément de liste (lien) 1</a></li>  <li>Elément de liste 2</li>  </ul>  <a href="#">Un dernier lien</a> |

**Q4.** Remplir le tableau ci-dessous.

ressources : <https://code.tutsplus.com/fr/tutorials/the-30-css-selectors-you-must-memorize--net-16048>

|  |  |
| --- | --- |
| Sélecteur CSS | Signification |
| E | sélectionne tous les éléments de type E |
| E,F | sélectionne tous les éléments E et tous les éléments F (liste groupée) |
| E F | électionne tous les éléments F descendants de E (à n’importe quelle profondeur) |
| E+F | sélectionne l’élément F qui suit immédiatement E (frère adjacent) |
| E>F | sélectionne les éléments F enfants directs de E |
| E~F | sélectionne tous les éléments F frères de E qui le suivent (frères généraux) |

1. Exercice 2 :

Class & id sont des attributs que vous pouvez ajouter aux éléments HTML afin de les sélectionner, cibler bloc par bloc les éléments à mettre en forme.

Textez le code suivant dans un fichier « Exercice2.html »:

|  |  |
| --- | --- |
| HTML | CSS |
| <!DOCTYPE html>  <html>  <head>  <link rel="stylesheet" href="./css/styles.css">  <title>Page Title</title>  </head>  <body>  <h1>Un titre de niveau 1</h1>  <h2 id="titre2">Un titre de niveau 2</h1>  <h3 class="titre3" >Un titre de niveau 3</h3>    <p>Un premier paragraphe</p>  <p>Un autre paragraphe avec un <a href="#">lien</a></p>  <ul>  <li>Elément de liste 1</li>  <li class="vert">Elément de liste 2</li>        </ul>  <p>Un troisième paragraphe</p>      </body>  </html> | p{  color:red;  }  p a{  color:orange;  background-color: black    }  .vert{  color:green;  }  #titre2{  color: orange;  font-size: 24px;  }  .titre3{  color: blue;  font-size: 24px;  } |

Pour donner un style aux éléments marqués par un attribut « id » on établira un appel par le sélecteur #id exemple :

|  |  |
| --- | --- |
| HTML | CSS |
| <h2 id="titre2">Un titre de niveau 2</h1> | #titre2{  color: orange;  font-size: 24px;  } |

Pour donner un style aux éléments marqués par un attribut « class» on établira un appel par le sélecteur .class exemple :

|  |  |
| --- | --- |
| HTML | CSS |
| <h3 class="titre3" >Un titre de niveau 3</h3> | .titre3{  color: blue;  font-size: 24px;  } |

**Q5.** Que se passe-t-il si nous insérons comme attribut une class et un id ?

tester sur votre code.

|  |
| --- |
| Si nous insérons à la fois une classe et un ID sur un élément, le style appliqué dépendra de la spécificité CSS. |

**Q6.**Sur un même élément ,donnez deux classes, une pour le texte et l’autre pour la couleur.Que se passe-t-il ?

|  |
| --- |
| Le paragraphe aura une taille de police de 20px (grâce à la classe .texte), une couleur rouge et un texte en gras (grâce à la classe .gras). **Tous les styles des deux classes sont appliqués** à l'élément, car les classes sont cumulatives. |

Ajouter une information de couleur sur l’une des deux class et associer là à un id. Que ce passe-t-il ?

|  |
| --- |
| Le paragraphe aura :   * une taille de police de 20px (classe .texte), * un texte en gras (classe .gras), * **une couleur bleue** (id #mon-id).   **La couleur bleue est appliquée** car l'ID a une **spécificité plus élevée** que les classes en CSS. La règle de l'ID écrase donc la propriété color définie dans la classe .gras. |

Qu’est-ce que cela signifie ?

|  |
| --- |
| Cela signifie que **la spécificité des sélecteurs** détermine quelle règle CSS est appliquée en cas de conflit. |

Important : un « id » est unique.

1. Exercice 3:

La notion d’héritage (copier le fichier html Exercice2.html et renommer le en Exercice3.html)

Pour comprendre les mécanismes fondamentaux de cascade et d’héritage en CSS, il faut avant tout comprendre ce qu’est un conflit CSS.

Parfois, plusieurs sélecteurs différents vont nous permettre d’appliquer des styles CSS à un même élément.

Imaginons par exemple un élément  p auquel on attribuerait un attribut « class » et un attribut « id ». Nous allons pouvoir appliquer des styles CSS à cet élément de trois façons évidentes différentes :

* en utilisant un sélecteur élément ;
* en le ciblant via son attribut class.
* en le ciblant via son attribut id.

Appeler l’enseignant.

Déterminer les niveaux importances en CSS dans le tableau suivant :

|  |  |
| --- | --- |
|  | Niveau d’importance |
| Sélecteur (exemple body) | 1 (le plus faible) |
| Class | 2 |
| id | 3 |
| Style sur HTML | 4 (le plus important) |

* 1. Exercice 4 :

Les belle boîte.

Le meilleur moyen de visualiser les boîtes CSS est de faire dessiner leur bord par le navigateur. Reliez dans un premier temps les fichier Ma\_StartUp.html et style.css. Chargez Ma\_StartUp.html dans chrome

**lien pour télécharger le document.**

[**https://drive.google.com/drive/folders/1Dau8jZHiZB7LeJWSv0aWM6mNOXEXxRi0?usp=sharing**](https://drive.google.com/drive/folders/1Dau8jZHiZB7LeJWSv0aWM6mNOXEXxRi0?usp=sharing)

### Le sélecteur universel

C'est facile en utilisant le sélecteur \*, qui permet de sélectionner *tous* les éléments d'une page html. Les propriétés structurelles des boîtes seront vues en détail dans la page suivante, pour le moment on va simplement utiliser cette règle :   
\* { border: 1px red dotted; }

Rechargez Ma\_StartUp.html dans chrome, maintenant vous voyez les boîtes CSS avec une bordure rouge (red), en pointillés (dotted), et de 1px de large.

Faites varier avec la souris la taille de la fenêtre que constatez-vous comme changement au niveau des boîtes ?

|  |
| --- |
| * Quand la fenêtre rétrécit ou s’agrandit, les blocs en pourcentage s’adaptent (plus étroits/plus larges). * Le texte se remet à la ligne, donc la hauteur des boîtes change. * La barre de gauche reste fixe, visible au même endroit. * Le diaporama garde ses slides mais son cadre change de taille. * Les bordures rouges montrent clairement l’espace occupé par chaque élément (même les liens). |

### En ligne, en bloc

Vous constatez avec Ma\_StartUp.html dans chrome qu'il existe plusieurs comportements différents pour ces boîtes CSS. Certaines prennent toute la largeur de la fenêtre, on les appelle boîtes de type *block*. Si vous diminuez la largeur de la fenêtre du navigateur, vous constatez que la hauteur de ces boîtes block s'adapte à leur contenu.

C'est le cas pour les éléments de type p, h1, ou encore ul et li. En traitement de texte pour impression papier, ces éléments seraient de la famille des paragraphes.

Les boîtes des éléments de type a ou strong ont une largeur adaptée à leur contenu, et leur hauteur ne change pas si on diminue la largeur de la fenêtre : elles reviennent à la ligne si nécessaire. Par conséquent, elles ne sont pas toujours rectangulaires!   
On les appelle boîtes de type *inline*. En traitement de texte pour impression papier, ces éléments seraient plutôt de la famille des chaînes de caractères (morceaux de texte).

Enfin, on serait tenté de ranger les boîtes des éléments de type img dans la catégorie inline, mais on voit vite qu'elles n'ont pas le même comportement lorsqu'on diminue la largeur de la fenêtre: l'image contenue ne revient évidemment pas à la ligne, et une partie disparaît.   
Ces boîtes, qui servent à contenir des objets non textuels insérés dans une page web, forment une troisième catégorie (sans nom).

### Balises div et span

Ces balises n'ont pas de signification particulière en html, c'est pourquoi on ne s'en est jamais servi à un niveau avancé. Elles peuvent être utiles en css, par contre. Les éléments de type div donnent des boîtes de type block, et ceux de type span des boîtes de type inline.   
On se servira par exemple de balises div pour structurer le contenu d'une page html, un peu comme avec des sections. On pourra ainsi agir plus facilement sur sa mise en page par css.

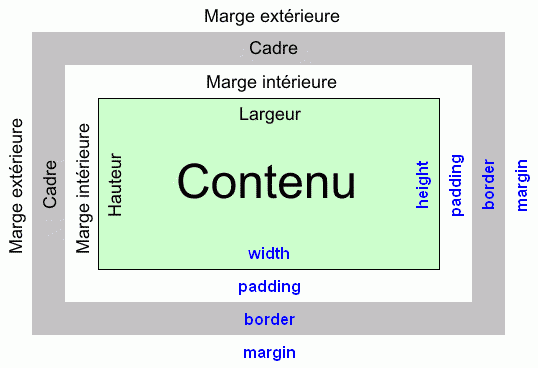
## Structure d'une boîte CSS <https://www.w3schools.com/css/css_boxmodel.asp>

Combien une boîte a-t'elle de propriétés ? Beaucoup, donc nous étudierons les principales petit à petit. Commençons par la structure, c'est-à-dire ce de quoi est constituée une boîte.

### Propriétés

|  |  |
| --- | --- |
| Propriété | Valeur |
| margin | longueur relative ou absolue |
| border-width | longueur relative ou absolue, 3px par défaut |
| border-color | black par défaut, red, blue, ... |
| border-style | none par défaut, solid, dotted, ... |
| border | voir ci-dessous |
| padding | longueur relative ou absolue |

Chaque boîte possède une zone de contenu délimitée par un espacement (ou marge intérieure : padding) , une bordure (border) et une marge (margin). C'est dans la zone de contenu qu'est positionné le contenu de l'élément.   
Le schéma ci-dessous illustre les relations entre ces zones et la terminologie (anglophone) employée pour les désigner.



Les propriétés width et height servent à définir les dimensions du contenu de la boîte, on en parlera page suivante.

### Bordures : <https://www.w3schools.com/css/css_border.asp>

Dans les navigateurs, en général, la valeur par défaut de la propriété border-style est none (pas de matérialisation). Par conséquent, vous ne verrez pas de bordure tant que vous n'aurez pas fixé une autre valeur pour cette propriété.

On a utilisé dans la page précédente la règle:

\* { border: 1px red dotted; }

La propriété border permet de résumer sur une seule ligne les 3 propriétés :

* border-width,
* border-color,
* border-style,

L'ordre des 3 valeurs est en fait sans importance, car il n'y a pas de confusion possible par le navigateur entre un nombre, une couleur et une forme de trait.

### Marges: <https://www.w3schools.com/css/css_margin.asp>

Les marges intérieures et extérieures permettent de régler les marges du document comme en traitement de texte pour impression papier: il suffit de fixer les marges extérieures de l'élément « body ».

Elles permettent aussi de modifier des attributs usuels de paragraphes comme le retrait (gauche ou droit) et l'espacement (du haut ou du bas).

Lorsque vous ne souhaitez agir que sur une seule de ces quatre directions, vous pouvez utiliser une propriété plus précise obtenue en ajoutant -left, -right, -top ou -bottom au nom de la propriété à modifier, par exemple: margin-left, padding-top, border-bottom, ou encore border-left-style.

*En faisant afficher les bordures comme dans la page précédente, on voit l'existence de valeurs par défaut pour les marges: très souvent, les bordures des boites ne se touchent pas. Il n'y a pas de normalisation sur ces valeurs par défaut, elles peuvent être différentes d'un navigateur à l'autre. Si vous avez besoin de les contrôler,* \* { border: 1px red dotted; }\* { border: 1px red dotted; }

Une remarque importante:

Lorsque deux boîtes sont côte à côte, leurs marges *ne s'additionnent pas*, c'est la plus grande des deux qui est utilisée. Vérifiez-le avec la boîte de l'élément h1.   
Une autre façon de le voir: mettre margin: 5px; dans style.css.

et rechargez la page Ma\_StartUp.html. Tous les espacements entre boîtes block sont de 5px, aucun ne fait 10px.

## Dimensions du contenu <https://www.w3schools.com/css/css_dimension.asp>

### Propriétés

Les propriétés width et height (largeur et hauteur, en anglais) servent à modifier les dimensions par défaut du contenu d'une boîte de type block. Si la boîte devient trop petite pour son contenu, la propriété overflow permet de gérer le débordement.

|  |  |
| --- | --- |
| Propriété | Valeur |
| width | longueur relative ou absolue |
| height | largeur relative ou absolue |
| overflow | visible par défaut, hidden, auto |

### Modifications

Reprenez le fichier Ma\_StartUp.html et fixez la largeur des éléments de type « h1 » à 200px.

Que constatez-vous pour leur hauteur dans le navigateur?

|  |
| --- |
| La hauteur s'adapte automatiquement au contenu, et le contenu reste aligné à gauche |

Elle s'adapte en fonction du contenu, comme on s'y attendait.   
Par ailleurs, ce contenu reste affiché du côté gauche de la fenêtre.

A présent, fixez aussi la hauteur des éléments de type p à 30px.

Certaines boîtes sont maintenant trop grandes pour leur contenu, mais d'autres trop petites: observez si celui-ci déborde.

Ces propriétés ne sont pas supposées s'appliquer aux boîtes de type inline, qui sont sensées servir uniquement à des mises en forme locale du texte et non à de la mise en page.

Essayez avec les éléments de type strong (ou a): fixez leur largeur et leur hauteur à 200px, vous ne devriez pas voir de modification.

Et les images? Ajoutez une image et une règle pour le sélecteur img et la propriété width (ou height), et voyez ce qui se passe pour l'image contenue dans la boîte.

|  |
| --- |
| Les images respectent les propriétés de dimension |

### Centrage horizontal d'une boîte

Une boîte de type block occupe par défaut la largeur de la fenêtre. Grâce à la propriété width, on peut faire en sorte que sa largeur soit plus petite, et on a vu que la boîte s'aligne alors à gauche de la fenêtre.   
Comment centrer une telle boîte dans la fenêtre? Il est facile de le faire approximativement en modifiant la marge de gauche, mais le résultat ne tient plus si on change la largeur de la fenêtre...

La solution consiste à utiliser la valeur auto pour les marges extérieures gauche et droite: le navigateur calcule alors ces deux marges par soustraction (largeur de la fenêtre moins largeur totale de la boîte) et division par 2.   
Faites l'essai avec les éléments de type h2.

h2{

height:200px;

width:50%; margin:0px auto;

}

Essai suivant: en modifiant simplement la valeur de la marge droite, comment aligner ces boîtes à droite de la fenêtre au lieu de les centrer?

Évidemment, le navigateur refait le calcul si vous modifiez la largeur de la fenêtre: c'est son travail normal d'affichage!

### Débordement (overflow, en anglais)

Dans notre exemple en cours, on constate que le texte déborde de certaines boîtes: la valeur par défaut de la propriété overflow est visible.   
Si vous demandez que cette propriété ait la valeur hidden, le débordement sera invisible (caché). Faites-le pour les boîtes concernées (sélecteur p à nouveau).   
Mais alors, du texte est perdu. La valeur auto permet d'obtenir un ascenseur pour l'accès au texte caché, essayez!

overflow: hidden;