

EP2IC3IV - ECUE Introduction au web

Tableau de bord / Mes cours / EP2IC3IV - ECUE Introduction au web / Concepts et notions de CSS

[Syllabus](#)

[Installation VMLinux](#)

HTML & CSS

[Le Réseau et le Protocole HTTP/1.1](#)

[Les Services Web](#)

Section 11

Bases d'HTML

Concepts et notions de CSS

[Applications en HTML/CSS](#)

[Responsive Web Design](#)

Introduction au CSS

Les feuilles de style en cascade (*Cascading Style Sheets* ou CSS) est un langage informatique permettant de mettre en forme des pages web (HTML ou XML), notamment les polices, les couleurs, les marges, hauteur/largeur de la page ou des images et le positionnement des différents éléments HTML. On peut ainsi facilement séparer le fond de la page web (le code HTML et texte important vis-à-vis des utilisateurs) de sa forme (le code CSS). C'est uniquement dans la feuille de style que l'on modifiera l'apparence de ces éléments importants (gras, italique, mais aussi couleur, soulignement, clignotement ...).

Que se passe-t'il quand la page web est chargée dans le navigateur ?

Différents contenus doivent être pris en charge : le contenu HTML, CSS et multimédias (image, son, vidéo, etc).

1. Le navigateur charge la page HTML reçue de la part du serveur web.
2. Il convertit le document HTML en DOM.
3. Le navigateur envoie une requête pour accéder et récupérer les différentes ressources liées à la page HTML (CSS, JS, multimédias, etc.). Il va lire les informations contenues dans les balises `<head></head>` pour afficher correctement la page. **C'est donc entre ces balises que se place le lien de la feuille de style attachée au document HTML** : `<link href="style.css" rel="stylesheet" type="text/css"/>`.
4. Le navigateur parse le contenu CSS et trie les règles CSS selon les types de sélecteurs et modifie le DOM en fonction de ces règles. Si le navigateur rencontre un sélecteur qu'il ne connaît pas alors il l'ignore et passe à la règle suivante.

Comment intégrer du code CSS dans une page HTML ?

Une première manière consiste à écrire les règles de style dans un fichier séparé de la page HTML. Le lien vers le fichier CSS est intégré dans la balise `<head></head>` en utilisant la syntaxe mentionnée plus haut : `<link href="style.css" rel="stylesheet" type="text/css"/>`.

Il est également possible d'intégrer du code CSS dans une page web via la balise `<style></style>` où l'attribut style peut être ajouté dans n'importe quelle balise. Ces méthodes sont cependant déconseillées car elle ne permettent pas de séparer le fond de la forme dans une page web et ne facilite donc pas la maintenance du site web.

Comment se structure une feuille de style ?

Un fichier CSS est une collection de règles décrivant la façon dont les éléments HTML d'une page web doivent s'afficher. Chaque règle contient :

Un **sélecteur** qui définit sur quel élément HTML s'applique la déclaration CSS.

Une **déclaration** qui définit les propriétés (couleur, police, taille du texte, etc.) et les valeurs (rouge, arial, 12px, etc.) à appliquer au sélecteur.

Une règle CSS respecte la syntaxe suivante :



```

selecteur_1, ... , selecteur_n{
    propriété_1 : valeur_1;
    ...
    propriété_n : valeur_n;
}

```

Le CSS suit le principe suivant : les propriétés attribuées à un élément de la page web seront transmises aux éléments qu'il imbrique. Si des règles communes s'appliquent à plusieurs d'entre eux (couleur du texte, marges ...) alors elles seront définies globalement. Si plusieurs règles affectent la même propriété d'un même élément, c'est la règle la plus précise qui l'emporte (celle qui possède le plus de sélecteurs). Si les deux règles possèdent la même précision, c'est la règle décrite en dernier qui écrase la règle antérieure.

Exemple :

```

h1 {
    font-size: 12px;
}

p, h1{
    color: red;
    font-style: italic;
    font-size: 10px;
}

```

Comment seront mis en forme les titres de grande importance h1 ?

Ils auront une taille de 10px et une couleur rouge.

La première règle permet de sélectionner tous les titres de grande importance (h1) et de modifier la taille de leur police (font-size) à 12 pixels de hauteur (12px). La seconde règle permet de sélectionner tous les paragraphes et les titres de grand importance (sélecteurs *p*, *h1*), de modifier leur couleur (propriété *color*) en rouge (propriété *red*), de mettre leur style (propriété font-style) en italique (propriété *italic*) et d'imposer une taille de police de 10px.

Les sélecteurs CSS

Il existe 4 catégories de sélecteurs :

Les sélecteurs **simples** : Parmi eux figurent les :

sélecteurs de **type** qui correspondent aux noms des éléments HTML à sélectionner,

sélecteurs de **classe** (.) qui regroupent et ciblent les éléments en fonction de leur attribut *class* (e.g. <div class="content"> a pour sélecteur *.content*),

sélecteurs d'**identifiant** qui ciblent les éléments en fonction de leur attribut *id* (e.g. <div id="unique_content"> a pour sélecteur *#unique_content*),

ou encore les sélecteurs d'**attributs** qui ciblent les éléments en fonction de la valeur d'un de leurs attributs. Plusieurs éléments HTML peuvent appartenir à une même classe mais un identifiant n'est valable que pour un seul élément.

Les **combineurs** : Permettent de combiner plusieurs sélecteurs pour former de nouveaux sélecteurs plus spécifiques. Parmi eux figurent les :

sélecteurs de voisin direct (+) qui ciblent les éléments HTML qui se succèdent immédiatement dans le DOM (ex. "*div + h1*" sélectionne n'importe quel élément <h1> qui suit immédiatement un élément *div*.),

sélecteurs de voisins (~) qui ciblent les éléments HTML qui ont le même parent mais qui ne se suivent pas forcément (ex. "*p ~ ul*" sélectionne tous les éléments qui suivent --immédiatement ou non-- un élément <p>),

sélecteurs d'éléments fils (>) qui sélectionnent les fils directs d'un élément dans le DOM,

les sélecteurs de descendants (espace) ciblent les éléments descendants c'est à dire pas forcément les fils directs de l'élément (ex. "*p a*" sélectionne tous les éléments <a> se trouvant à l'intérieur d'un élément <p>, que <a> soit un fils direct ou indirect de <p>.)

Les **pseudo-classes** : Permettent de sélectionner des éléments selon l'état dans lequel ils sont (ex. *a:visited* représente les liens qui ont été consultés, *a:hover* sélectionne les liens survolés).

Il existe bien entendu un bon nombre d'autres sélecteurs qu'il est possible d'utiliser pour plus d'efficacité que l'on appelle des **pseudo-éléments**. Par exemple :

::first-line (*::first-letter*) permet de ne sélectionner que la première ligne (lettre) d'une balise. *p::first-line* sélectionnera donc la première ligne de tous les paragraphes de la page.

:first-child (*:last-child*) permet de sélectionner un élément s'il est le premier (dernier) enfant d'un élément. *p:first-child* sélectionnera tous les paragraphes qui sont les premiers enfants d'une autre balise.



Une liste plus conséquente de sélecteurs est disponible sur le site de W3Schools :

http://www.w3schools.com/cssref/css_selectors.asp.

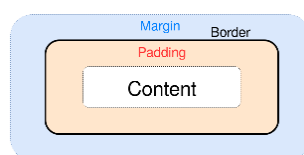
Les propriétés CSS

Tout comme pour les éléments html, une liste non exhaustive des propriétés les plus utilisées est disponible en téléchargement à la fin du cours. Elle est conçue comme un aide-mémoire où les balises sont triées par domaine puis par ordre alphabétique. Là encore, les sites précédemment conseillés (MDN, W3Schools, etc.) peuvent être utiles pour consulter la liste complète des propriétés CSS.

Remarque pratique : en CSS, les commentaires sont encadrés par les symboles `/*` et `*/`.

Le modèle de boîtes

Les boîtes constituent la structure de base de l'affichage d'une page html. Les balises HTML sont en effet représentées par une boîte dans le navigateur. Une balise imbriquée dans une autre dans le code html sera ainsi représentée par une boîte imbriquée dans une autre. On appelle cette organisation le **modèle de boîtes**.



Une boîte est caractérisée par quatre éléments :

La **marge extérieure** (*margin*), l'espace entre cette boîte et les boîtes qui l'entourent.

La **bordure** (*border*), par défaut, cette dernière a une épaisseur de 0 pixel et n'est donc pas visible.

La **marge intérieure** (*padding*), l'espace entre la bordure et le contenu de la boîte.

Le **contenu** (*content*), ne dépendant pas du CSS mais du code HTML.

Si la balise correspondant à la boîte possède un arrière-plan, celui-ci ne sera affiché qu'à l'intérieur de la bordure, c'est à dire derrière le contenu et la marge intérieure. De plus, les marges intérieures et extérieures sont toujours transparentes.

Les propriétés CSS *margin*, *border* et *padding* permettent de modifier la taille des marges et de la bordure (0 correspondant à l'absence de marge/bordure). La taille peut-être exprimée en différentes unités (pixels -px-, centimètres -cm-, pouces -in-, etc.) ou en pourcentage de la taille de la boîte (%).

Il est également possible de modifier les marges et la bordure uniquement sur un côté du bloc en suffixant la propriété CSS correspondante par *-top*, *-left*, *-bottom*, *-right*. Par exemple, *margin-top* ne modifie que la marge extérieure située au dessus de l'élément et *padding-right* ne modifie que la marge intérieure du côté droit de l'élément.

Astuce : les propriétés *padding* et *margin* agissent également comme des raccourcis selon le nombre de valeurs qui leur est affecté :

avec une seule valeur, la propriété affecte tous les côtés de l'élément,

avec deux valeurs, la première valeur affecte le haut et le bas et la seconde les côtés de l'élément,

avec trois valeurs, la première valeur affecte le haut, la seconde les côtés et la troisième le bas de l'élément.

avec quatre valeurs, les valeurs sont affectées dans l'ordre suivant : haut, droite, bas, gauche (comme le nommage des points d'un carré en géométrie).

Quelques exemples :

Haut et bas à 1 pouce (*in* pour inches) et les côtés à 2 pouces.

```
p{
  padding: 1in 2in;
}
```

Haut à 2 pixels, côtés à 20% de la longueur de la boîte et bas à 1 centimètre.

```
p{
  margin: 2px 20% 1cm;
}
```

Les deux codes suivants sont équivalents :

```
p{
    margin: 1em 2em 3em 4em;
}

p{
    margin-top: 1em;
    margin-right: 2em;
    margin-bottom: 3em;
    margin-left: 4em;
}
```

Pour plus de précisions sur les unités utilisées ici, il est possible de se rendre sur le site de W3Schools à l'adresse suivante : <https://www.w3.org/Style/Examples/007/units.fr.html>.

Éléments *Inline* et *Block*

Chaque élément de la page web va se voir associé une boîte CSS en vue de son affichage. La boîte sera gérée en *inline* ou en *block*, selon sa nature par défaut ou en se fiant aux règles CSS. Le navigateur calcule alors à quel endroit l'élément sera placé dans la fenêtre, en fonction de la taille de la fenêtre et des boîtes précédentes déjà placées. Le placement naturel, appelé le **flux normal**, se fait logiquement de haut en bas et de gauche à droite.

D'après les définitions issues de la W3School https://www.w3schools.com/html/html_blocks.asp :

Une **boîte *block*** commence toujours sur une nouvelle ligne et prendra toute la largeur du navigateur.

Une **boîte *inline*** ne commencera pas une nouvelle ligne et prendra uniquement la place nécessaire pour afficher son contenu.

Un troisième type de boîte existe : la **boîte *inline-block***. Ces dernières mélangent les caractéristiques de boîtes *block* et *inline*.

En règle générale, les boîtes *inline* se placent dans des boîtes *block* afin d'avoir un maximum de contrôle sur l'apparence d'une page web.

Pour changer le mode d'affichage par défaut d'une boîte il suffit de modifier la propriété `display` de l'élément. Par exemple,

```
h1 {
    display: inline-block;
}
```

Voici maintenant quelques caractéristiques à prendre en compte pour comprendre la mise en page des boîtes et obtenir une visualisation correcte d'une page web :

Une boîte *inline* :

respecte les options de padding et de marges à droite et à gauche mais pas du haut ni du bas.

ne respecte pas les définitions de largeur (*width*) ni de hauteur (*height*).

permet la mise en place des éléments à droite et à gauche.

Une boîte *block* :

respecte toutes les options de padding, marges, largeur et hauteur.

force la création d'une nouvelle ligne après son placement.

par défaut, la largeur est définie à 100% de la page.

Une boîte *inline-block* :

respecte toutes les options de padding, marges, largeur et hauteur.

permet la mise en place des éléments à droite et à gauche.

Centrer des boîtes en CSS

Centrer horizontalement

Pour centrer du texte dans un élément HTML, il est possible d'utiliser la propriété `text-align` :

```
p.texteaucentre {
    text-align: center;
}
```

On utiliserait alors le code HTML suivant :

```
<p class="texteaucentre"> Texte centré </p>.
```

Pour centrer un bloc qui ne prend qu'une certaine partie de la page (par ex. 80% de la page), on fait appel à d'autres propriétés comme suit :

```
.centre {  
margin: auto;  
width: 80%;  
}
```

Certaines balises HTML ne possèdent pas la propriété *text-align*. Il est alors possible de combiner ces balises avec la balise `<p>`. Par exemple avec la balise `` :

```
<p class="texteaucentre"></p>.
```

Centrer verticalement

Pour aligner verticalement un élément, on a la propriété *vertical-align*. Pour centrer verticalement un élément, il faudrait donc appliquer la règle *"vertical-align: middle;"*.

Le problème est que cette propriété est applicable uniquement aux boîtes *inline* et aux cellules des tableaux. Comment fait-on donc pour centrer un bloc ?

Solution 1 : placer un tableau invisible et y mettre le texte.

Solution 2 : plus légère et simple, combiner une boîte `<div>` avec la boîte `<p>` et "convertir" la boîte `<div>` en cellule de tableau à l'aide de la déclaration CSS *"display = table-cell"*.

Solution 3 : Utiliser un *padding*.

Par exemple :

Solution 2 et 3:

```
div.milieu {  
height: 10cm;  
display: table-cell;  
vertical-align: middle;  
}
```

```
div.milieu {  
padding: 50px 0;  
}
```

Et en HTML :

```
<div class="milieu"><p>texte</p></div>
```

Exercices

à l'exception du fichier « 1_exo.html », les fichiers à compléter, éditer se trouvent sous forme de ressources extras pour ce TP (voir fin de page).

Exercice 1 : Un peu d'HTML

Éditez le fichier `1_exo.html` pour reproduire la page suivante :



Exercice 1 - Un peu d'HTML

Le fond et la forme

Un document XHTML sert avant tout à décrire la structure d'une page web.

Les balises **** et ** servent ainsi à signifier au navigateur l'importance de leur contenu, et c'est le navigateur qui, par convention, mettra en forme le contenu en gras ou en italique.

Les listes

Il est possible de créer des énumérations et des listes que l'on peut imbriquer à l'envie.

1. Premier élément
 - Sous-élément 1.1
 - Sous-élément 1.2
2. Second élément
 - Sous-élément 2.1
 - Sous-élément 2.2

Les liens

Il est possible de placer des liens absolus vers d'autres sites web : [lien 1](#).

Il est également possible d'utiliser des liens relatifs vers une autre page ([lien 2](#)) présente sur le site. Ce lien devra alors prendre en compte l'arborescence à parcourir depuis la page actuelle vers la page cible.

Les images



Tux, la mascotte Linux

Précisions :

le titre de l'exercice est en h1 et les sous-titres en h2,

le texte "****" se trouve entre les balises HTML "****",

le texte "**" se trouve entre les balises HTML "**",

les caractères "<" et ">" sont de caractères spéciaux dans un document HTML (utilisés pour l'écriture des balises).

pour visualiser le caractère "<" utilisez la commande "<" dans le code HTML

pour visualiser le caractère ">" utilisez la commande ">" dans le code HTML

le lien 1 doit rediriger vers un site extérieur,

le lien 2 doit rediriger vers la page 2_exo.html,

l'image

correspondant au fichier img_tux.png doit posséder une description alternative

est accompagnée d'une étiquette (caption).

Validez votre page ici <https://validator.w3.org/>. Vous devrez corriger les erreurs signalées par le validateur.

Exercice 2 : Premiers pas avec une feuille de style

Créez un dossier "sites" et créez dans ce dossier un sous-dossier "html" où vous placerez votre fichier 2_exo.html

. Modifiez le code HTML pour

1. Associer et créer une feuille de style CSS "2_feuille_de_style.css" au fichier 2_exo.html. Pour cela vous devez créer un dossier "css" dans le répertoire du site web (i.e. dans le dossier "sites").
2. Sans modifier plus le code HTML, mais en modifiant votre feuille de style locale, faite le nécessaire pour atteindre le look suivant pour votre page :



Exercice 2 - utiliser une feuille de style

Habiller une page web

Une feuille de style **CSS** permet d'ajouter des éléments décoratifs au document **XHTML** afin d'en améliorer la présentation.

On peut ainsi modifier le style de certaines balises existantes : **** et **** ou tout simplement se servir des balises neutres pour délimiter une zone de texte précise à l'aide des classes et des **identifiants**. Contrairement à l'identifiant, qui doit être unique, une classe peut-être attribuée à plusieurs balises.

Les listes

Pour une coloration plus précise, on peut utiliser différents sélecteurs afin de ne colorer que **certaines éléments** et non d'autres.

1. Premier élément
 - Sous-élément 1.1
 - Sous-élément 1.2
2. Second élément
 - Sous-élément 2.1
 - Sous-élément 2.2

Les liens

Il est également possible de colorer les liens selon qu'ils aient été visités ou non ([lien vers l'exercice 1](#)).

Précisions :

la première lettre de chaque paragraphe est colorée en rouge, majuscule et en gras.

pour souligner un texte, utilisez la propriété "text-decoration: underline" (ou "text-decoration: none" pour la supprimer),

le lien doit devenir rouge et ne plus être souligné lorsque la souris le survole,

les couleurs utilisées dans ce TD sont *black*, *red*, *blue*, *orange*, *magenta* et *purple*,

"font-weight" permet de changer le texte en format "bold" ou "normal",

"font-style" permet de déclarer un texte en italique ("italic") ou "normal",

du texte en minuscule peut être transformé en majuscule facilement avec la propriété "text-transform: capitalize;",

la propriété "font-variant: small-caps;" capitalise un mot en utilisant une taille réduite pour les caractères initialement en minuscule,

le texte "" dans la figure est en italique.

Rappel : Il est également possible d'utiliser un code hexadécimal à la place du nom de la couleur (par exemple le code du rouge est #FF0000).

Exercice 3 : Le modèle de boîtes

Ouvrez la page `3_exo.html` dans votre navigateur et la feuille de style `3_feuille_de_style.css` dans un éditeur. Certaines règles ont déjà été ajoutées, vous les connaissez toutes sauf *border*. Cette propriété permet de régler l'apparence de la bordure d'un élément. La valeur affectée dans cette règle (*solid 2px black*) désigne successivement le style de la bordure (simple trait, double trait, pointillés, etc.), son épaisseur et sa couleur. Il est possible de préciser *border -xxxx* où *xxxx* désigne la partie de la bordure à modifier (top, bottom, left, right).

N'hésitez pas à tester les propriétés énoncées sur chaque question afin de confirmer vos réponses. Lisez les règles CSS déjà présentes pour correctement répondre aux questions.

Question 1 : Quelle est la distance totale entre les bords droits et gauche, et entre les bords supérieur et inférieur, du bloc 0 (c'est à dire du paragraphe qui est à l'intérieur du div d'id part1) si la propriété "padding: 10px 20px 30px 40px;" lui est affectée ?

Question 2 : Que se passe-t-il (quels changement visuel prennent effet) lorsque la propriété "margin: 0px 0px 100px 0px;" est ajoutée au bloc 0 ? Supposez que la propriété padding de la question précédente est toujours en place.

Question 3 : Que se passe-t-il si les blocs 1, 2 et 3 sont tous des boîtes

block ?

inline ?

inline-block ?

Question 4 (Optionnel) : Testez les propriétés suivantes sur le bloc 0 et prenez le temps de bien identifier les différentes zones de la boîte (contenu, padding, bordure, margin) :

margin à 1em, 100px, 10%, 2cm.

padding à 1em, 100px, 10%, 2cm.

Essayer de ne changer que le padding ou margin du dessus, dessous ou des côtés (en utilisant *margin-top* par exemple).

Essayer de donner quatre valeurs différentes au margin (ou le padding) du dessus, dessous ou des côtés.

Question 5 (Optionnel) : Explorez ce qu'il se passe si les boîtes 1, 2 et 3 deviennent des boîtes

block, inline et inline-block respectivement ?



inline-block, inline et block respectivement ?
 inline, block et inline-block respectivement ?
 testez d'autres combinaisons possibles.

Exercice 4 : Listes d'éléments *inline* et *block*

À l'aide des pages https://developer.mozilla.org/fr/docs/Web/HTML/Block-level_elements et https://developer.mozilla.org/fr/docs/Web/HTML/Éléments_en_ligne, listez quelques boîtes (i.e. balises) de type *block* et *inline* que vous considérez les plus communs ou importantes.

Exercice 5: Positionnement au centre de boîtes CSS

Sans modifier le fichier `5_exo.html`, donnez à la page Web l'agencement suivant

Exercice 1 - Les colonnes et leur positionnement

- Entrée 1
- Entrée 2
 - sous-menu 1
 - sous-menu 2
 - sous-menu 3
- Entrée 3

A key trick to the manipulation of HTML elements is understanding that there's nothing at all special about how most of them work. Most pages could be made up from just a few tags that can be styled any which way you choose. The browser's default visual representation of most HTML elements consist of varying font styles, margins, padding and, essentially, display types.

Inline

inline does just what it says - boxes that are displayed inline follow the flow of a line. Anchor (links) and emphasis are examples of elements that are displayed inline by default.

The following code, for example, will cause all list items in a list to appear next to each other in one continuous line rather than each one having its own line...

Block

block makes a box standalone, fitting the entire width of its containing box, with an effective line break before and after it. Unlike inline boxes, block boxes allow greater manipulation of height, margins, and padding. Heading and paragraph elements are examples of elements that are displayed this way by default in browsers.

The next example will make all links in "nav" large clickable blocks:

Inline-block


display:inline-block will keep a box inline but lend the greater formatting flexibility of block boxes, allowing margin to the right and left of the box, for example.

The most fundamental types of display are inline, block and none and they can be manipulated with the display property and the shockingly surprising values inline, block and none.

Tweets

de @Laboratoire_I3S

Laboratoire I3S a retweeté



Fabien Gandon
@fabien_gandon

Intégrer

Voir sur Twitter

Précisions :

les largeurs maximales et minimales de la page sont de 1024px et 800px respectivement (si on agrandit la fenêtre du navigateur, le contenu de la page n'occupe pas plus de 1024px de large, et si on rétrécit la fenêtre, on ne peut pas la rétrécir autant que l'on veut, parce qu'on impose de voir le contenu de la page (son body) avec une largeur d'au moins 800px)

il y a une distance de 10px entre les bords droit et gauche de la page et son contenu. Le bord est un trait de 2 pixels noir.

la page (i.e. bords droit et gauche + contenu) est centrée. Notez bien que vous devez donc centrer l'élément "<body>" (lire en haut comment centrer un bloc avec quelques règles CSS).

le titre principal à l'intérieur de la page est centré horizontalement. Il y a un trait (couleur lightblue et 2px d'épaisseur) placé à une distance de 20px en haut et en bas du titre (donc centré aussi verticalement entre ces 2 traits).

le texte en italique (le texte en dessous du menu)

ne prend que 60% de la largeur de la page (par exemple, 600 pixels si la largeur de la page est de 1000 pixels)

chaque ligne du texte est centrée horizontalement

Votre progression ?

 [Ressources TD1](#)



 [Rendus TD1](#)



[HTML & CSS](#)

[Applications en HTML/CSS](#)



Connecté sous le nom « anjou Raphael » (Déconnexion)

Accueil

Université Côte d'Azur

 <https://univ-cotedazur.fr>



Tableau de bord

Cours

Tous les cours

Recherche de cours

Accompagnement

Pédagothèque

Documentation

Français (fr)

English (en)

Français (fr)

Résumé de conservation de données

