

DÉVELOPPEMENT WEB

HTML / CSS

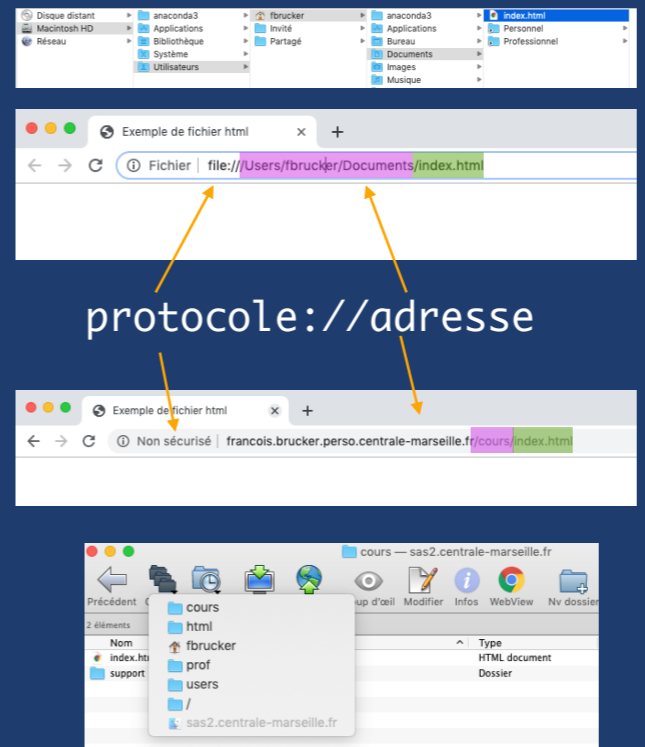
base du html et css

et utilisation de framework web

le texte souligné et très souvent en bleu est un lien

LIRE UN FICHER HTML

- nom : `index.html`
- interprété par le navigateur grâce à une `uri` :
 - localement : `file://chemin_absolu/index.html`
 - via le réseau : `http://serveur/chemin_relatif/index.html`



chemin absolu = sur la machine

chemin relatif = par rapport à la racine du serveur : ici `/users/prof/fbrucker/html/`

uri : https://fr.wikipedia.org/wiki/Uniform_Resource_Identifier

chrome cache souvent le protocole. Non sécurisé signifie `http://` La norme est actuellement `https` (s= « sécurisé »)

COMMENT APPRENDRE ?

- il faut en faire.
- quelques sites :
 - il y a tout, sans tuto : <http://www.w3schools.com>
 - un long tuto : https://developer.mozilla.org/fr/docs/Apprendre/Commencer_avec_le_web
 - des tutos super bien (avec un accent anglais rigolo) : [the net ninja](#)
- attention : le web est plein de vieux trucs. Assurez vous d'être bien en html5 et css3

FICHER HTML MINIMAL

- `squelette.html`
- constitué de balises et forme un arbre (l'arbre DOM)
 - type de balise (tag) : `html`, `head`, `body`, ...
 - le contenu est entre les balises ouvrante `<tag>` et fermante `</tag>`
 - des attributs `nom="valeur"`
- cas particuliers
 - balise constitué uniquement de paramètres : s'ouvre et se ferme dans la même balise, comme les balises `meta`
 - **DOCTYPE** : type d'html utilisé, par défaut l'html 5. Il ne faut pas utiliser autre chose.
 - commentaire `<!--mon commentaire-->`

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemple de fichier html</title>
  <meta charset="utf-8" />
</head>

<body>
  <!-- Ici votre page web -->
</body>
</html>
```

entête html

corps

squelette.html à montrer dans chrome

balises : <https://jaetheme.com/balises-html5/>

entête :

- contient au minimum le titre de la page (obligatoire) et l'encodage (fortement conseillé)

- on mettra plus tard le chargement de fichiers de style, du javascript, les balises meta : <https://www.alsacreations.com/article/lire/628-balises-meta.html>

corps :

- le contenu proprement dit

ÉCRIRE DU TEXTE

- `du_texte.html`
- `<p>`un paragraphe`</p>`
- les titres : de `<h1>`titre principal`</h1>` à `<h6>`titre de niveau 6`</h6>`
- texte ``renforcé`` (en gras) ou mis en ``emphase`` (en italique)
- les listes ordonnées (``) et non-ordonnées (``).
Chaque liste est composée d'item (``).
- du [faux texte \(lorem ipsum\)](#) pour le remplissage

du_texte.html à montrer dans chrome

lorem ipsum : cela ressemble à du texte. Alors que smkdjpkjkjfv jdclnscl pas du tout. C'est pour préparer le design et pas le contenu.

BALISES SÉMANTIQUES

- les [balises sémantiques](#) permettent de structurer le texte :
 - `nav`, `header`, `footer`, `section`, `article`, ...
 - structures complexes : [table](#) ou [form](#)
- ils permettent de rationaliser le développement du site
- il faut gérer leurs apparences avec un css

LE STYLE

- [permet de modifier](#) l'apparence de chaque élément en changeant ses [propriétés](#)
- s'applique :
 - dans le fichier html avec la balise `<style></style>`
 - dans un fichier séparé avec la balise `<link rel="stylesheet" type="text/css" href="nom_de_fichier" />`
- On préférera séparer forme et fond et ainsi faire un fichier de style séparé lorsque le fichier html devient grand.
- Si plusieurs propriétés s'appliquent pour des blocs imbriqués (body et p par exemple), c'est la propriété du bloc le plus imbriqué qui est appliquée
- Si plusieurs modifications s'appliquent à la même propriété, c'est le dernier qui a parlé qui a raison.

```
sélecteur {  
    propriété: valeur;  
    ...  
    propriété: valeur;  
}
```

les éléments **p**

```
p {  
    font-size: 40px;  
    color: #FF0000;  
    background-color: skyblue;  
}
```

format [rgb](#)

couleur de fond

[noms](#)

rgb(a) : https://www.w3schools.com/css/css3_colors.asp

POLICES DE CARACTÈRES

- choix de la police
 - un police présente sur tous les navigateurs : [web safe fonts](#)
 - charger une police explicitement, par exemple avec [google-fonts](#)
- la taille ([une propriété css](#))
 - unité **em** : par rapport à la balise père. **font-size: 2em** signifie une taille deux fois plus grande que la balise père. Chaque navigateur donne une taille à la balise **html** (la balise racine)
 - unité **px** : par rapport à la résolution de l'écran (attention aux différentes résolutions, un bon design doit s'adapter aux différentes tailles d'écran)
 - noms : **small, medium, xx-large, ...**

on ne peut pas pré supposer qu'un utilisateur a une police de caractère que vous possédez

EST-CE DU BON ?

- Il est indispensable d'écrire du html/css correct
 - outils de validation : <https://validator.w3.org> (validez [html_a_valider.html](#))
 - coding style : <https://google.github.io/styleguide/htmlcssguide.html>
- S'il n'est pas correct, les navigateurs vont essayer de l'interpréter pour le rendre présentable.
 - Il existe [des différences d'interprétation](#) entre navigateurs, mais la plupart du temps les cela est du au rattrapage des erreurs du html ([quirks mode](#))
 - très compliqué à déboguer car difficile de savoir ce qui cause l'erreur.
 - code impossible à faire évoluer
 - autant tout de suite bien faire les chose.

faire le text avec [index_a_valider.html](#)

SITE PHASE 1

BASES

phase_1/index.html



MODÈLE DE BOITES

- chaque balise est représentée sous la **forme d'une boîte** et s'intègre dans le flux du rendu
- Il y a deux grands types de boîtes (propriété `display` en css) :
 - **box** : les uns en dessous des autres.
 - `p, h1, ...`
 - elles prennent la largeur de la fenêtre par défaut. La hauteur est déterminée par le contenu
 - **inline** : les uns à côté des autres
 - `strong, em, ...`
 - pas de largeur ni de hauteur
- le positionnement précis des balises peut être assez complexe. Il est recommandé de rester simple ou d'utiliser des frameworks pour gérer le positionnement dans le flux.



Pour voir les bord de tous les blocs, on peut ajouter la propriété css : `* { border: 1px solid black; }`

par défaut les élément box prennent toute la largeur de la page et la hauteur est déterminée par le contenu

on peut changer la propriété en inline pour un type par défaut box (et réciproquement). L'essayer avec les outils de dév (on peut rajouter une propriété css).

Cas particulier du inline-block

CONFIGURER SES BOITES

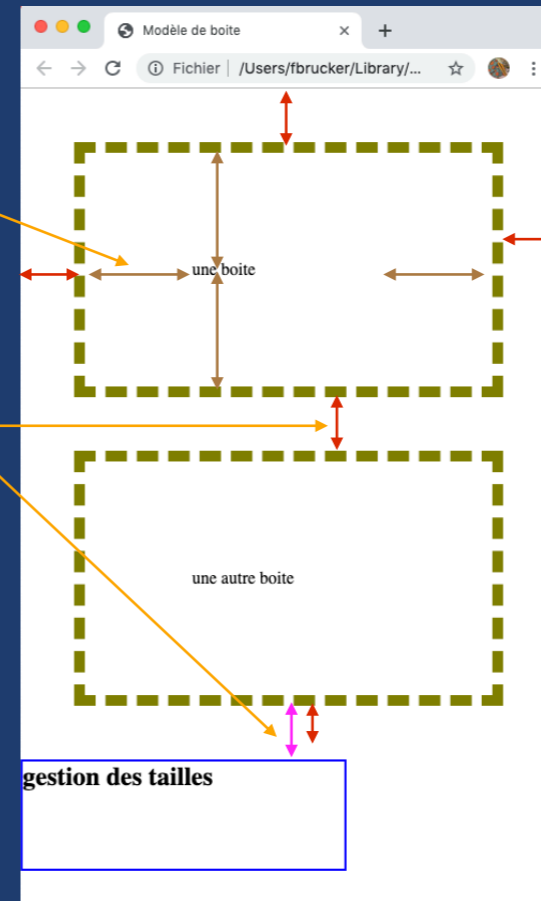
par défaut, toute la largeur est prise

- [modèle_de_boite.html](#)

```
• html, body {  
  padding: 0;  
  margin: 0;  
}  
p {  
  padding: 100px; ←→  
  border: 10px dashed olive;  
  margin: 50px; ←→  
}  
h2 {  
  width: 300px;  
  height: 100px;  
  border: 2px solid blue;  
  margin: 0;  
  margin-top: 60px; ←→  
}
```

fusion des marges

- [centrer les objets](#)



Par défaut on supprime les padding et marge du body et de html.

padding à la place des marges et pas de border permet d'éviter la fusion des marges
centrer des éléments : plein de cas différent. Un peu embêtant sans framework.

CHOISIR SES ÉLÉMENTS

- [ajout d'attributs](#) aux balises :
 - **id** : identifiant **unique** pour une balise, s'utilise avec un **#** dans les css
 - **class** : identifiant d'une **classe** d'éléments. Peut regrouper des blocs de natures diverses. S'utilise avec un **.** dans les css.
 - balises anonymes **div** et **span**: sélection uniquement via son id et ou ses classes. Permet de contrôler parfaitement le bloc.
 - **div** pour les blocs
 - **span** pour les inline
- **class** et **div** sont très utilisé dans les frameworks.

```
#premier_paragraphe { font-size: 2em; }  
#dernier_paragraphe { font-size: .5em; }  
.vert { color: #00ff00; }  
.font_rouge { background-color: #ff0000; }
```

```
<p id="premier_paragraphe">  
je suis 2x plus grand</p>  
<p class="vert">  
je suis vert sur fond blanc  
</p>  
<p class="vert rouge">  
je suis vert sur fond rouge  
</p>  
<div id="dernier_paragraphe" class=  
"rouge">  
je suis vert et 2x plus petit  
</div>
```



SELECTEUR CSS

- la hiérarchie des balises s'appelle l'arbre DOM. On parle de racine (`html`), de père, de fils et de descendants.
- toute modification s'applique par défaut à l'élément et ses descendants
- On peut affiner la sélection. Par exemple :
 - `h1, h2`: les titres `h1` et `h2`
 - `.cl`: éléments de classe `cl`
 - `p.cl`: paragraphes de classe `cl`
 - `.cl1.cl2`: éléments ayant les classes `cl1` et `cl2`
 - `section h2`: les `h2` descendants d'une section (peut-être aussi profond que l'on veut)
 - `ul > li`: les `li` fils d'un `ul` (et pas d'un `ol` par exemple)

SITE PHASE 2

SÉLECTEURS

phase_2/index.html



LIENS

- `texte où cliquer`
- deux types de liens :
 - relatifs à la position du fichier html : `./index.html` ou `../répertoire/autre_index.html`)
 - externe : une adresse web (doit commencer par `https://` ou `http://`)

expliquer . et ..

PROPRIÉTÉS CSS DES LIENS

- [des_liens.html](#)
- ajout de [pseudo-classes](#)
- **a** : sélecteur du lien. Mais il y a plusieurs sous cas :
 - **a:visited** : apparence d'un lien après qu'il ait été visité
 - **a:hover** : lorsque la souris passe dessus
 - **a:active** : lorsque l'on clique dessus

```
a, a:visited {
  text-decoration: none;
  color: #D7661C;
  border-bottom: 1px dotted #ccc;
}

a:hover {
  color: #94ab36;
  border-bottom: 1px solid #ccc;
}

a:active {
  color: #94ab36;
  border-bottom: 3px solid #ccc;
}

a.pdf {
  color: #360;
  background: url(pdf.gif) no-repeat;
  min-height: 10px;
  padding-left: 12px;
}
```

un lien doit être visible en css. On ne doit pas se demander si on peut cliquer dessus ou pas.

propriété du lien seul (sans pseudo-classes) : s'applique à tout par défaut. Exemple les lien de classe pdf.

comme ici border-bottom non défini dans a.pdf c'est les normaux qui s'appliquent. Il faut le redéfinir si on en veut pas (ex : mettre border-bottom: hidden)

couleur #360 est un « raccourci » css. Pas vraiment à utiliser mais c'est pur montrer. C'est équivalent à #336600

IMAGES

- ``
- Pour éviter les problèmes de chargement on mettra toujours les paramètres `width` et `height` à la taille réelle de l'image
- par défaut une image est en display `inline`
- pour centrer une image on utilisera donc les paramètres css:
 - `display: block;`
 - `margin-left: auto;`
 - `margin-right: auto;`

ne pas changer la taille des images car l'image sera adaptée à la taille demandée :

- si on la réduit : on use de la bande passante pour rien puisque toute l'image sera téléchargée avant d'être réduite
- si on l'agrandit : on verra les pixel

Attention également aux photos de vos téléphones : elles sont énormes ! L'affichage du site risque d'être lente.

Il vaut toujours redimensionner l'image pour qu'elle soit facile à afficher et faire un lien vers l'image (grosse) à télécharger.

IMAGES DE FOND

- [plusieurs format de couleurs](#)
- images de fond plusieurs propriétés :
 - background-color, background-image, ...
 - on peut les assembler en une seule : [background](#) qui est une propriété raccourcie.
- [plein de possibilités](#)

Les propriétés raccourcis ne sont pas forcément à utiliser. C'est juste s'ils le voient dans un tuto. Il vaut mieux être explicite si on peut.

SITE PHASE 3

LIENS ET
IMAGES

phase_3/index.html



le css est devenu gros, on l'a split dans un autre fichier. Chemin relatif. Idem pour l'image.
dans le css on a mis des marges au main pour séparer du bord (fond d'écran en plusieurs parties).

POSITIONNEMENT

- Il est possible de [sortir les éléments du flux](#) avec la propriété position
 - **absolute** : par rapport à la page
 - **fixed** : par rapport au navigateur
 - **relative** : décalage
- centrer verticalement du texte
- **attention** : on préfère actuellement utiliser un framework pour le positionnement. C'est bien plus simple et maintenable que le faire à la main. Exemple : [centrer verticalement du texte en css](#)

ODDS & ENDS

- Différents techniques plus ou moins avancées en html/css pour briller en soirée mais pas indispensable (souvent il existe l'équivalent en plus simple via un framework)
- [flottants](#)
- [animations](#)
- [encore plus de moyens de selections](#)

SITE PHASE 4

POSITION ET ANIMATIONS

phase_4/index.html



En mettant le titre en position fixed, on a du ajouter une marge au body pour que tout en haut le titre ne recouvre pas le texte.