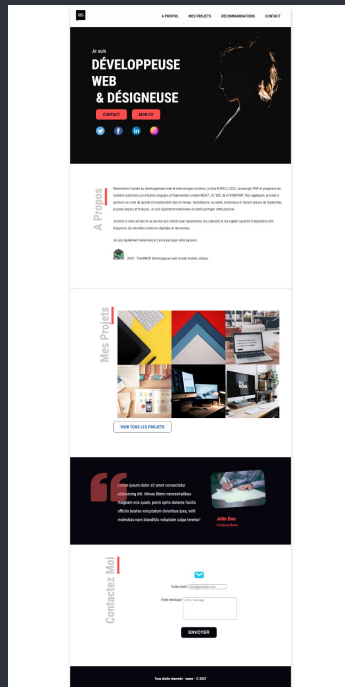


```
1  
2  
3 .Atelier d'initiation à la  
4 'programmation'{  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14
```

[Quelques pas de théorie
avec la pratique ...]

< Création d'un portfolio HTML CSS >

}



'Qui suis-je? Qui êtes-vous?';

Développeuse web depuis 2ans.

- * Reconversion professionnelle via **Webforce3/O'Clock**.
- * Passage du titre professionnel RNCP "Développeur web et web mobile" niveau Bac +2.
- * Spécialisation PHP / Symfony - certification [Opquast](#)
- * Plusieurs stages et missions freelance
- * Actuellement en mission longue, transformation en CDI fin 2021.

→ Quel est votre parcours jusqu'à votre inscription à cet atelier?

→ [Les femmes dans la tech?](#) En connaissez-vous?

→ Quels sont [les métiers](#) que vous connaissez dans l'univers du digital/numérique?

→ Pourquoi pas vous?

Les bases du 'web' {

- Introduction

Web, langage, page internet?

< Le langage de balisage > HTML

< Mettre en forme par les feuilles de style > CSS

- Atelier pratique : le Portfolio

< Du code au déploiement >

}

Introduction; {

Nous découvrirons les fondements et l'utilisation du **HTML**, qui constitue la base de tous les sites internet.

Nous verrons en même temps la partie **CSS**, qui vous permettra de mettre en forme vos contenus HTML de façon simple.

Nous aborderons aussi la notion de « **Responsive Design** », à l'heure où une grande part de notre utilisation du web provient des terminaux nomades (portables et tablettes).

Aucun prérequis mais nous allons tout de même, brièvement, rappeler le fonctionnement des sites web d'abords ⇒

}

Introduction; {

Comment consultez-vous vos sites web préférés?

Pour consultez vos sites web, vous utilisez un programme appelé le navigateur web.

Il en existent plusieurs (préinstallés ou à installer) sur votre système d'exploitation, parmi lesquels on retrouve ?

Chrome (de Google),
Internet Explorer (de Microsoft),
Safari (d'Apple),
Firefox etc...

Ils ont tous le même rôle: vous permettre d'accéder et lire vos pages web.

}

Introduction; {

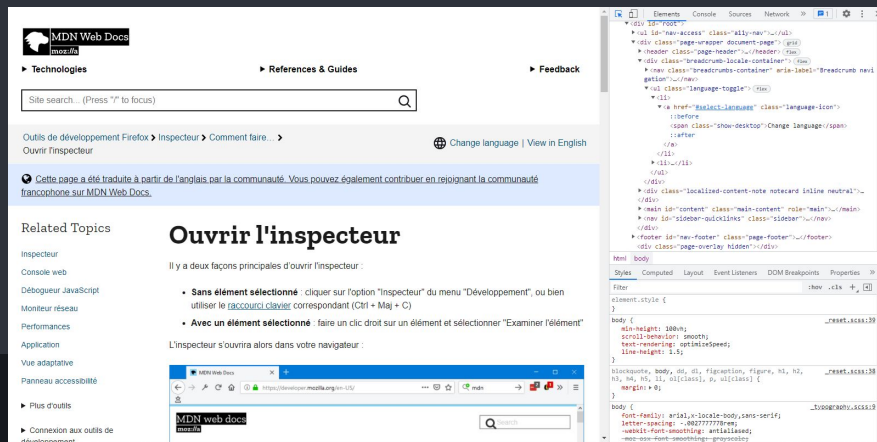
⇒ Qu'est-ce que vient faire **HTML** et **CSS** dans cette histoire ?

Les pages web sont écrites en HTML et CSS, voir du javascript et le navigateur est capable d'interpréter ces langages pour vous les retranscrire. Vous pouvez par ailleurs voir le code source des pages web visitées.

(via l'inspecteur -
option du navigateur).

Ctrl + maj + C
F12

Clic droit - inspecter
Votre meilleur allié!



Introduction; {

1 ⇒ Comment le navigateur fait pour accéder aux pages web ?

2 Votre navigateur vous permet, grâce aux URLs notamment d'accéder à des pages stockées sur des
3 machines que l'on appelle des serveurs.

4 Tout cela fonctionne selon une architecture appelée : client-serveur.

5 Client = votre terminal (PC, portable, tablette...) avec son navigateur.

6 Serveur = la machine distante où sont hébergés les pages web.

7 1/Le client demande une page (Requête)

8 2/Le serveur lui envoie une Réponse (la page demandée si elle a été trouvée).

9 3/Le navigateur interprète la page pour vous l'afficher correctement.

10 ⇒ Le protocole HTTP (HyperText Transfer Protocol).

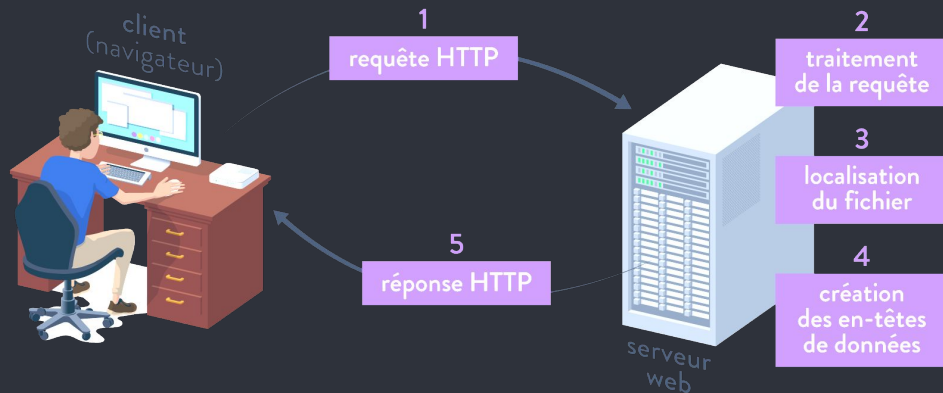
11 Un protocole est un contrat (des règles à respecter) entre le client et le serveur pour garantir
12 le transfert des fichiers HTML

13 via internet. C'est le navigateur qui gère lorsque vous tapez http:// (ou https:// pour le HTTP
14 version sécurisé), il comprend alors qu'il s'agit d'un échange HTTP.

}

Introduction; {

Le protocole HTTP



Lors de la Requête, ce protocole prévoit certaines méthodes (appelées aussi des verbes) qui spécifie l'action que l'on souhaite réaliser sur le serveur (par exemple POST, pour envoyer un message via un formulaire...
Lors de la Réponse, le protocole prévoit aussi que le serveur nous renvoie un statut de l'opération demandée. Ces statuts sont des nombres:

- les 200 : tout est OK
- les 300 : c'est une redirection de page
- les 400 : erreur côté client
- les 500 : erreur côté serveur

html



HTML signifie HyperText Markup Language, en français Langage de balisage hypertexte.

Ce n'est pas un langage de programmation à proprement dit, mais plutôt un langage de balisage (description et mise en forme de contenu via des balises) interprété par les navigateurs comme Firefox, Chrome, IE etc.

Son histoire : HTML \Rightarrow

il a fait son apparition dès 1991 lors du lancement du Web. Son rôle est de gérer et organiser le contenu. C'est donc en HTML que vous écrirez ce qui doit être affiché sur la page : du texte, des liens, des images... Vous direz par exemple : «Ceci est mon titre, ceci est mon menu, voici le texte principal de la page, voici une image à afficher, etc. ».

HTML 1 : c'est la toute première version créée par Tim Berners-Lee en 1991. Il n'était même pas possible d'afficher des images à ce temps là.

HTML 2 : de 1994 à 1996

HTML 3 : apparue en 1996, pose les bases de la W3C. Apparition de nombreuses fonctionnalités, comme les tableaux par exemple.

html



1 **HTML 4** : apparue en 1998. Avec des tableaux améliorés, des formulaires
2 améliorés, les frames et surtout l'intégration et l'exploitation du CSS.

3 **HTML 5** : c'est la dernière version. Plus complète et plus adaptée au web
4 d'aujourd'hui.

5 ⇒ Voilà pour la théorie, mais maintenant de quoi avons nous besoin pour créer
6 notre site ?

7 Commencer avec HTML:

- 8 • Un simple éditeur de texte suffit, pour écrire une page HTML.
- 9 • Ne pas confondre un éditeur de texte avec un logiciel de traitement de texte !
- 10 • Les éditeurs de texte : Bloc-Notes, Notepad++, SublimeText, VS Code...
- 11 • Ne pas oublier de sauvegarder son document avec l'extension .html (ou .htm)

12 [VScode studio](#) <<



Examinons les éléments de base HTML

Rendez-vous dans votre éditeur de texte!





Syntaxe: balises et attributs

Commencer avec HTML:

- Une balise classique se déclare avec un chevron ouvert «<», suivi du nom de la balise à utiliser, puis d'un chevron fermé «>».
- On ferme le contenu de la balise avec un chevron ouvert «<», suivi d'un slash «/», puis du nom de la balise, et enfin d'un chevron fermé «>».
- La balise peut contenir des attributs.

Raccourci VScode : ! puis → touche tabulation

Les balises de type auto-fermantes sont des balises qui sont ouvrantes et fermantes en même temps.

On ferme donc la balise en ajoutant un espace et un slash / à la fin de la balise avant le symbole >.

Il est important de préciser que le slash (/) et l'espace de fin ne sont pas obligatoires. En effet, on peut écrire une balise auto-fermante sans ces derniers, ça marche également. Cependant, la bonne pratique recommande d'ajouter un slash pour éviter de confondre avec une balise ouvrante

Quelques balises auto-fermantes :<link /> <meta /> <input />



Syntaxe: les commentaires HTML

- Les commentaires ne seront pas interprétés par les navigateurs.
- Ils servent uniquement aux développeurs pour faire une description dans le code.

Raccourci VScode : ctrl + k + c // décommenter ctrl + k + u

Attention à ne pas mettre de commentaires confidentiels ou sensibles (comme les mots de passes etc.), car tout le monde peut accéder au code source de la page et donc voir les commentaires.

Règles et bonnes pratiques:

- Les balises (nom, attribut) doivent être écrites en minuscule
- Les valeurs des attributs doivent être entre guillemets : "valeur"
- Une balise ouverte, doit être fermée.
- Une balise peut contenir du texte et/ou des balises (balise parent - balise enfant)
- On ne peut pas fermer une balise parent sans avoir fermé la balise enfant.
- Il est important d'indenter votre code
- N'hésitez pas à commenter votre code
- Sauvegarder vos fichiers HTML au format .html (ou .htm)



Quelques éléments sémantiques:

```
<header></header>
<nav></nav>
<main></main>
<footer></footer>
<section></section>
<article></article>
```

- Les éléments sémantiques donnent un sens concret à vos pages.
- Il en existe plusieurs et sont à privilégier par rapport aux éléments non sémantiques.



Les éléments textuels:

`<h1>Mon titre H1</h1>`

`<h2>Mon titre H2</h2>`

`<h3>Mon titre H3</h3>`

`<h4>Mon titre H4</h4>`

`<h5>Mon titre H5</h5>`

`<h6>Mon titre H6</h6>`

- Les éléments `<h1>` à `<h6>` représentent six niveaux de titres dans un document
- H1 est le titre principal, les autres sont successivement des sous-titres.
- Un élément de titre décrit brièvement le sujet de la section qu'il introduit.



Des balises de type bloc

```
1
2 <p>Ceci est un paragraphe</p>
3 <p>Ceci est un autre paragraphe</p>
4 <div>Ceci est une div</div>
```

- Les balises de type bloc placent le contenu dans un bloc avec retour à la ligne.
- Exemple <p> pour décrire un paragraphe.
- Exemple <div> pour division, est un conteneur autre que sémantique.

Saut de ligne et ligne horizontale

```
11
12 <p>Ceci est un paragraphe<br> long avec un retour à la ligne</p>
13 <hr>
14 <p>Deuxième partie<p>
```




Les listes ordonnées

```
1
2 <ol>
3 <li>La ligne 1</li>
4 <li>La ligne 2</li>
5 <li>La ligne 3</li>
6 </ol>
```

Les listes non ordonnées

```
7
8
9 <ul>
10 <li>La ligne 1</li>
11 <li>La ligne 2</li>
12 <li>La ligne 3</li>
13 </ul>
```

- Liste non ordonnée = liste à puces
- On peut imbriquer les listes ordonnées et non ordonnées..



Les éléments non sémantiques

```
1 <div>
2   <p>La ligne 1</p>
3 </div>
4 <div>
5   <p>Autre contenu !</p>
6 </div>
7 • Balise de type bloc, non sémantique.
8 • Regroupe des contenus dans un conteneur, sur lequel on peut
9   appliquer du CSS.
10 <span>Mon contenu en ligne.</span>
11 <span>... Suite de mon contenu en ligne.</span>
12 • Balise de type inline, non sémantique.
13 • Regroupe des contenus dans un conteneur, sur lequel on peut
14   appliquer du CSS.
```



L'ATELIER !!!!

Rendez-vous sur <https://github.com/jessicakuier/atelier-portfolio>
Et ensuite dans votre éditeur de texte!





Les Attributs universels

<https://developer.mozilla.org/fr/docs/Web/HTML/Element>

Retenir:

L'**ID** est très important c'est celui là qui nous permettra de l'identifier un élément de créer des liens où d'utiliser des événements sur celui-ci.

L'attribut universel **class** indique une liste de classes associées à l'élément courant.

- On peut donc référencer plusieurs classes sur un même élément.
- Une classe n'étant pas unique, on peut retrouver d'autres éléments l'utilisant.
- Les classes permettent de manipuler les éléments, via CSS en utilisant les sélecteurs de classe

L'attribut universel **style** permettra d'intégrer une règle CSS directement sur la balise à laquelle elle est rattachée.



Lien externe

```
1 <a href="https://google.fr">Lien vers google</a>
```

```
2 <a href="https://google.fr" target="_blank">
```

```
3 Lien vers google dans un nouvel onglet
```

```
4 </a>
```

5 • Un lien vers un site externe se fait en indiquant l'URL (chemin absolu) de celui ci.

6 • L'attribut target avec la valeur _blank permet d'ouvrir le lien dans un nouvel onglet

Lien vers une page interne

```
9 <a href="mapage.html">Lien vers mapage.html</a>
```

```
10 <a href="dossier/page.html">Lien vers page dans sous-dossier</a>
```

```
11 <a href=" ../page.html">Lien vers page dans dossier parent</a>
```

12 • La balise <a> pour anchor (ancree) définit un lien.

13 • L'attribut href détermine le chemin du fichier cible (vers lequel aller).

14 • On peut utiliser le chemin relatif ou absolu pour accéder à la page cible.

• ../ permet de remonter vers le dossier parent, si la page cible s'y trouve.



Lien vers un élément interne externe

```
1 <a href="#monid">Lien vers la balise d'id monid</a>
```

```
2 ...
```

```
3 <p id="monid">
```

```
4 Mon paragraphe !!
```

```
5 </p>
```

- Un lien vers une section interne en indiquant l'id de la section ciblée.
- Utiliser # pour référencer l'id

Lien vers l'élément d'une autre page

```
8 <a href="mapage.html#monid">Lien vers la balise d'id monid</a>
```

```
9 ...
```

```
10 <!-- page mapage.html -->
```

```
11 <p id="monid">
```

```
12 Mon paragraphe de la page mapage.html !!
```

```
13 </p>
```

```
14
```



Les éléments multimédia

La balise

- Comme pour les liens, l'attribut src détermine le chemin de l'image à insérer.
- Ce chemin peut être externe, relatif ou absolu.
- L'attribut alt permet d'afficher le message en cas de problème chargement de l'image. Cet attribut a aussi un intérêt pour l'accessibilité : les non-voyants auront la lecture de cet attribut.

```

```

```

```

```

```

```

```

[CSS leçon théorique]

< CSS (Cascading Style Sheets), c'est cet autre langage qui vient compléter le HTML.

CSS ? C'est lui qui vous permet de choisir la couleur de votre texte.
Lui qui vous permet de sélectionner la police utilisée sur votre site.
Lui encore qui permet de définir la taille du texte, les bordures, le fond...

Et aussi, c'est lui qui permet de faire la mise en page (positionnements) de votre site.>

}



Intégrer le CSS

Mettez en place le CSS

Methode 1: `<link rel="stylesheet" href="style.css">`

Methode 2: `<style>...</style>`

Methode 3: `<p style="propriete:valeur;">`

- Méthode 1 (recommandé): Définir dans un fichier .css a part.
- Méthode 2: Définir ses règles css dans une balise `<style>`.
- Méthode 3: Définir ses règles directement sur la balise concernée a l'aide de l'attribut universel `style`.
- Il est possible d'intégrer plusieurs feuilles de style (css) à votre fichier HTML



Intégrer le CSS

selecteur

```
{  
  propriete1: valeur1;  
  propriete2: valeur2;  
  propriete3: valeur3;  
}
```

- **sélecteur** : C'est l'endroit HTML où l'on souhaite appliquer la règle css.
- **propriété** : C'est le nom de l'effet de style que l'on souhaite donner a notre sélecteur.
- **valeur** : C'est la valeur souhaitée pour la propriété correspondante.

Dans un code CSS, on trouve trois éléments différents :

Le sélecteur : on écrit les noms des balises, id ou classes, dont on veut modifier l'apparence.

Des propriétés CSS : les « effets de style » de la page sont rangés dans des propriétés. exemple: propriété color, qui permet d'indiquer la couleur du texte, font-size qui permet d'indiquer la taille du texte, etc.

Les valeurs : pour chaque propriété CSS, on doit indiquer une valeur. Par exemple, pour la propriété color, il faut indiquer le nom de la couleur. Pour font-size, il faut indiquer quelle taille on veut, etc.



Appliquer du style à plusieurs balises

```
h1, em {
```

```
    color: blue;
```

```
}
```

- Ici, ceci aura pour effet d'appliquer la couleur bleu a toutes les balises h1 et em de notre page HTML

- Cela évite d'écrire 2 règles CSS qui font la même chose.

Le CSS et l'héritage

- Toutes les balises incluses dans la balise parente hériteront par défaut des propriétés de celui-ci.

- Son nom « Feuilles de style en cascade » vient de ce concept.

```
<div id="monid">
```

```
    <p>
```

```
        <span>...</span>
```

```
    </p>
```

```
</div>
```

```
#monid {
```

```
    background-color: green;
```

```
    color: white;
```

```
}
```



```
1 h3 em {  
2 }
```

- Ici, sélectionne toutes les balises situées à l'intérieur d'une balise <h3>
- Séparation par un espace.

```
4  
5 h3 + p  
6 {  
7 }  
8
```

- Ici, sélectionne la première balise <p> située après un titre <h3>
- Séparation par un +.

9 Les autres sélecteurs :

10 https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs_CSS

11 OU

12 <https://www.w3.org/Style/css3-selectors-updates/WD-css3-selectors-20010126.fr.html#selectors>

13

14

CSS



1 Pour modifier la taille du texte, on utilise la propriété CSS font-size.

2 Taille absolue:

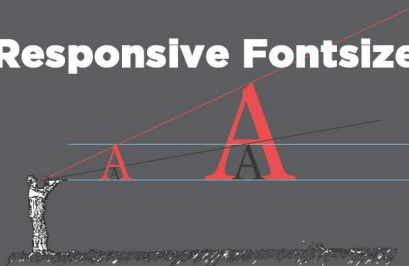
3 Pour indiquer une taille absolue, on utilise généralement les pixels. Pour avoir un
4 texte de 16 pixels de hauteur, vous devez
5 donc écrire : font-size: 16px;

6 **Valeur relative**: elle s'adapte plus facilement au contexte de vos visiteurs.

7 Les mots-clés possibles pour les valeurs relatives: xx-small, x-small, small, medium,
8 large, x-large, xx-large

9 Utiliser em pour les valeurs relatives: 1em, le texte a une taille normale. Inscrire
10 une valeur supérieure a 1 pour grossir (ex:
11 1.3em). Inscrire une valeur inférieure à 1 (ex: 0.8em)

Responsive Fontsize





1 La police personnalisée avec @font-face

```
2 @font-face {  
3   font-family: 'Mapolice';  
4   src: url('Masuperpolice.ttf');  
5 }
```

- 6 • @font-face est permet au css d'indiquer au navigateur quelle police télécharger si
7 le navigateur de votre visiteur ne l'a possède pas.
8 • Attention a bien télécharger les bonnes extensions en fonction du navigateur (ttf,
9 eot, otf, svg, woff).
10 • Idéalement, placer ses polices dans un dossier font de vos sources.

11 Pour le portfolio,
12 Nous utiliserons Google Fonts et
13 le lien externe @import →
14



Google Fonts



1 **Font-style** : Italique, oblique, normal

- 2 • font-style : est la propriété CSS pour mettre en italique.
- 3 • Il prend 3 valeurs : normal, oblique et italic.

4 **Font-weight** : mettre en gras

- 5 • Font-weight : est la propriété CSS pour mettre en gras.
- 6 • Il peut prendre comme valeurs : bold, normal.

7 **Text-décoration** : soulignement et autres décorations

- 8 • Text-décoration : est la propriété CSS pour souligner et autres.
- 9 • Il peut prendre comme valeurs : underline, overline, blink, line-through, none.

10 **L'alignement**

- 11 • **Text-align** : est la propriété CSS pour les alignements connus.
- 12 • Il peut prendre comme valeurs : left, center, right, justify.
- 13 • Attention : ne fonctionne que sur les balises de type block (et non inline).(cf.slide 18)

CSS



Couleur du texte

- `color` : est la propriété CSS pour mettre un texte en couleur.
- Il existe plusieurs façon d'indiquer la couleur (son nom, en hexadécimale, en RGB).

Balise {

```
    color: red;  
}
```

Indiquer une couleur:

⇒ Il existe **16 couleurs** standards en tapant leur nom en anglais (white, silver, gray, black, red, maroon, lime, green, yellow, olive, blue, navy, fuchsia, purple, aqua, teal) + d'autres dans certains navigateurs.

⇒ **Hexadécimal** : (#000000) c'est une notation sur en base 16 qui va retranscrire le code en couleur. On doit toujours commencer par écrire un dièse (#), suivi de six lettres ou chiffres allant de 0 à 9 et de A à F.

⇒ **RGB** (Red-Green-Blue): On défini ici une quantité de Rouge, de Vert et de Blue pour en résulter une couleur. `rgb(255,0,0);`

Il existe une version **RGBA** RGB (Red-Green-Blue-Alpha), qui gère aussi la transparence `rgba(255,0,0,0.8);`

Opacité: La propriété CSS `opacity` donne l'opacité a appliquer a un élément de texte ou de couleur - gère la transparence entre 0 et 1 (0 étant transparent et 1 totalement opaque). ex: `opacity: 0.8;`



Appliquer une image de fond

- **Background-image** : est la propriété CSS pour mettre une image de fond.
- On peut appliquer une image de fond sur la page, les titres, paragraphes, etc.

Options pour l'image de fond

- background-attachment : permet de fixer le fond.
- valeurs : **fixed**, scroll (par défaut)
- background-repeat : permet de définir la répétition de l'image de fond.
- valeurs : no-repeat, repeat-x, repeat-y, repeat (par défaut)
- background-position : positionne l'image de fond.
- valeurs : en pixel, top, bottom, left, center, right

Combiner les propriétés:

```
background: url("soleil.png") fixed no-repeat top right;
```



La propriété **border** peut utiliser jusqu'à 3 valeurs :

La **largeur** : indiquez la largeur de votre bordure. Mettez une valeur en pixels (comme 2px).

La **couleur** : c'est la couleur de votre bordure. Utilisez, comme on l'a appris, soit un nom de couleur (black, red, ...), soit une valeur hexadécimale (#FF0000), soit une valeur RGB (rgb(198, 212, 37)).

Le type de bordure : là, vous avez le choix. Votre bordure peut être un simple trait, ou des pointillés, ou encore des tirets, etc.
Voici les différentes valeurs disponibles :

none: pas de bordure (par défaut) ;

solid: un trait simple ;

dotted: pointillés ;

dashed: tirets ;

double: bordure double ;

groove: en relief ;

ridge: autre effet relief ;

inset: effet 3D global enfoncé ;

outset: effet 3D global surélevé.

Ex: border: 3px blue dashed;



Bordures arrondies

- `border-radius` : arrondir les angles.
- valeurs : en pixel, pourcentage
- On peut arrondir de façon globale : `border-radius: 10px;`
- On peut arrondir chaque coin : `border-radius: 10px 5px 10px 5px;`

Les valeurs correspondent aux angles suivants dans cet ordre :

- 1/en haut à gauche ;
- 2/en haut à droite ;
- 3/en bas à droite ;
- 4/en bas à gauche.

Les ombres des boîtes ex: `box-shadow: 6px 6px 0px black;`

- `box-shadow` : est la propriété CSS pour ombrer les boîtes.
- S'applique à tout le bloc et prend 4 valeurs dans l'ordre suivant:
 - 1/ Le décalage horizontale de l'ombre
 - 2/ Le décalage vertical de l'ombre
 - 3/ L'adoucissement du dégradé
 - 4/ La couleur de l'ombre



Les positions de boîtes

Pause grenouilles

FLEXBOX FROGGY
APPRENDRE LES FLEXBOX PAR LE JEU



CSS



Hauteur et Largeur des boîtes

Les dimensions

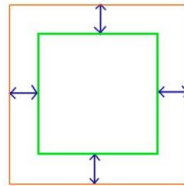
- Un bloc a des dimensions précises.
- Il possède une hauteur (**height**) et une largeur (**width**).
- Par défaut un bloc prend 100% de la largeur disponible

Si on applique une règle CSS (`width: 50%;`), cette boîte prendra 50% de la largeur disponible.

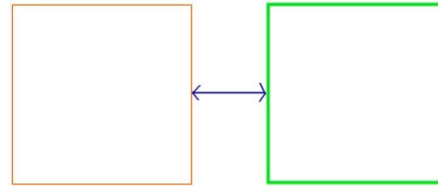
Marges intérieures - Marges extérieures

- Marge intérieure = padding.
- Marge extérieure = margin.

En CSS le padding et le margin s'exprime en général en px ou %



Padding



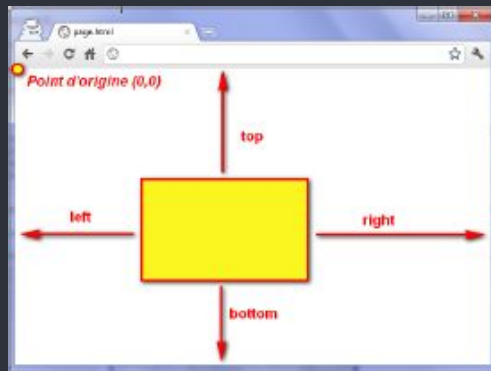
Margin



Les Positions :

Le positionnement absolu : il nous permet de placer un élément n'importe où sur la page (en haut à gauche, en bas à droite, tout au centre, etc.).

top : position haut de page
left : position gauche de page
right : position droite de page
bottom : position bas de page
• Valeurs en px ou %



Le positionnement fixe : identique au positionnement absolu mais, cette fois, l'élément reste toujours visible, même si on descend plus bas dans la page.

Le positionnement relatif : permet de décaler l'élément par rapport à sa position normale.

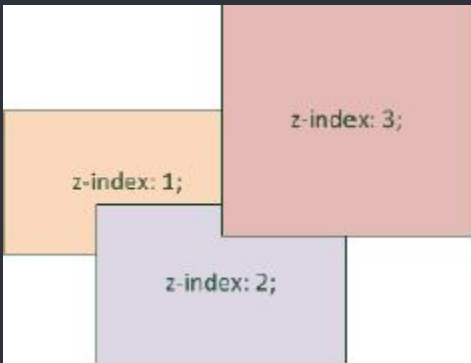


Les Positions :

Le positionnement absolu: **z-index**

- z-index dans le cas de superposition d'éléments en absolu, qui auraient des coordonnées proches, il est possible d'indiquer l'ordre de superposition.

```
position: absolute;  
z-index: 3;
```



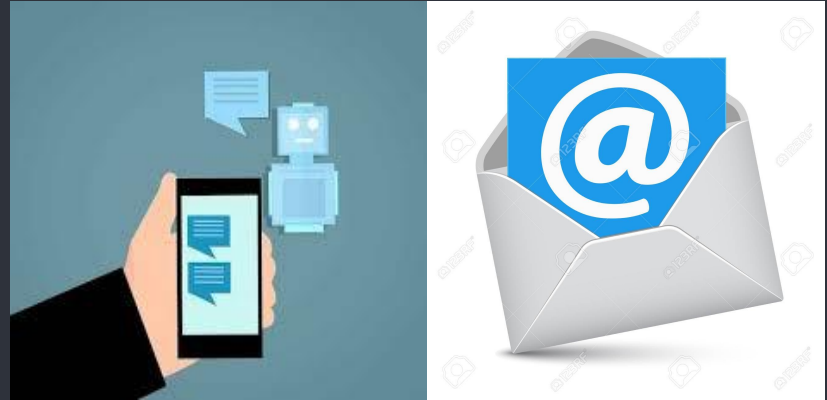
Les éléments positionnés en absolu sont placés par-dessus le reste des éléments de la page ! Si plusieurs éléments en absolu ont des positions proches, ils se chevauchent et la propriété z-index permet alors d'indiquer quel élément placer au dessus
⇒ z-index: 1; z-index: 2; etc.



1 Pour le formulaire de contact, nous utiliserons la solution <https://formspree.io/>
2 Afin de pouvoir faire transiter les données d'envoi vers un serveur qui
3 nous retransmet les informations reçues sur notre boîte email.

4
5 Cela évite d'être spammée, attention à ne jamais laisser trainer son
6 adresse email publiquement.

7
8 Autre solution envisageable : chatbot.





Les formulaires

La balise `<form>`

`<form>` `</form>`

- `<form>` : c'est la 1ère balise. Elle indique le début et la fin du formulaire.
- C'est une balise de type bloc.

Les zones de saisie

`<input />`

- `<input>` :
- Les zones de texte permettent de saisir du texte dans un formulaire.
- C'est une balise « orpheline ».
- Il conviendra de définir un attribut type. Ex: type=«text»
- Par défaut le type est «text»

les libellés `<label>`

`<label for="pseudo">Votre pseudo</label>`

`<input type="text" id="pseudo" name="pseudo" />`

- `<label>` : pour donner un libellé à son champ de saisie.
- `for` : permet de lier le label avec le champ de saisie d'id portant la même valeur.



Les formulaires

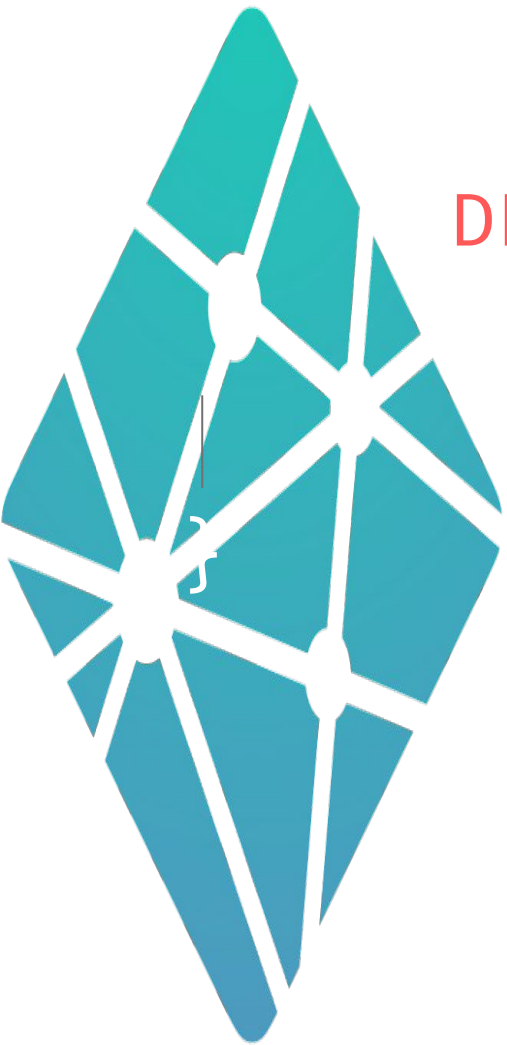
placeholder: donner une indication sur le contenu du champ

Pour créer une zone de texte multiligne, on change de balise : nous allons utiliser `<textarea>` `</textarea>`.

Les boutons d'envoi

```
<input type="submit" value="Envoyer" />
```

- submit: le bouton d'envoi de formulaire
- reset: remise à zéro du formulaire.



DEPLOIEMENT

netlify

déploiement



Après s'être inscrite,
un simple
glisser/déposer suffit
au bas de la page,
onglet "Site".

Démo en live!

The screenshot shows the Netlify dashboard for a team named 'jessicakuijer's team'. The top navigation bar includes links for Team overview, Sites, Builds, Plugins, Domains, Members, Audit log, Billing, and Team settings. A 'New site from Git' button is visible in the top right. The main content area displays a list of sites with a search bar at the top. Each site entry includes a thumbnail, the site name, the deployment source, and the last published date. At the bottom of the dashboard, there is a dashed box with the text: 'Want to deploy a new site without connecting to Git? Drag and drop your site output folder here'.

Site Name	Deployment Source	Last Published
portfolio-tutoriel	Manual deploys	Nov 13 (a day ago)
jessicakuijer	Deploys from GitHub	Oct 8 (a month ago)
discovery-medalmadness	Deploys from GitHub	Jul 21 (4 months ago)
foodlocal-apidoc	Manual deploys	Mar 3 (8 months ago)
strapi-blog-jessicakuijer	Deploys from GitHub	Oct 21, 2020 (a year ago)
sanity-gatsby-blog-studio-kfgrpc4s	Deploys from GitHub	Aug 18, 2020 (a year ago)
sanity-gatsby-blog-web-49v1thna	Deploys from GitHub	Aug 18, 2020 (a year ago)
sanity-nextjs-landing-pages-web-5h6whntz	Deploys from GitHub	Aug 18, 2020 (a year ago)
sanity-nextjs-landing-pages-studio-5jtncjji	Deploys from GitHub	Aug 18, 2020 (a year ago)

Want to deploy a new site without connecting to Git?
Drag and drop your site output folder here

Balises OPEN GRAPH



Open graph



Vous pouvez contrôler ce qui sera partagé dans les réseaux sociaux grâce aux métadonnées open graph.



```
<!-- (Final) Ajouter balises OG (Open Graph) après déploiement sur netlify -->  
<meta property="og:url" content="">  
<meta property="og:title" content="">  
<meta property="og:description" content="">  
<meta property="og:image" content="">
```

DESKTOP FIRST MOBILE FIRST?

|
}



Media Queries? {

Ce ne sont pas de nouvelles propriétés mais des règles que l'on peut appliquer dans certaines conditions..

- Ces conditions se basent sur les résolutions d'écran, le type d'écran, le nombre de couleurs, l'orientation de l'écran etc.

Méthode 1: `<link rel="stylesheet" media="xxx" href="resolution.css"/>`

} Méthode 2: `@media screen and (max-width: 1280px)`



Place aux questions de manière
générale :)

Questionnaire mesure d'impact (lien à venir)

