

# MODULE HTML ET CSS

Fouziya Kouddane

**HTML**



**CSS**



**HTML**



**CSS**



Fouziya Kouddane

# CE QUE NOUS ALLONS VOIR

- ✓ Les formes en CSS
- ✓ Décimal VS hexadécimal
- ✓ Les différents formats d'images
- ✓ La propriété « Display »
- ✓ Les positions
- ✓ Quelques sélecteurs à revoir
- ✓ Margin VS padding
- ✓ La propriété background
- ✓ Flexbox

# LES FORMES EN CSS

Un carré:

```
#square {  
  background: lightblue;  
  width: 100px;  
  height: 100px;  
}
```

Un cercle:

```
#circle {  
  background: lightblue;  
  border-radius: 50%;  
  width: 100px;  
  height: 100px;  
}
```

Un triangle:

```
#triangle {  
  width: 0;  
  height: 0;  
  border-left: 40px solid transparent;  
  border-right: 40px solid transparent;  
  border-bottom: 80px solid lightblue;  
}
```

Pour aller plus loin: <https://blog.steloria.fr/2019/09/17/dessiner-en-css-les-formes/>  
<https://css-tricks.com/the-shapes-of-css/>

# DECIMAL VS HEXADECIMAL

En décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
En hexadécimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

En décimal	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	...
En hexadécimal	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	...

En décimal	...	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
En hexadécimal	...	EF	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF

# LES DIFFERENTS FORMATS D'IMAGES

FORMAT	AVANTAGES	INCONVENIENTS	QUAND L'UTILISER
<b>JPEG</b> <i>Joint Photographic Experts Group</i>  Extension : .jpg ou .jpeg	<ul style="list-style-type: none"> <li>• Léger</li> <li>• Possibilité de choisir le niveau de compression</li> <li>• 16 millions de couleurs</li> </ul>	<ul style="list-style-type: none"> <li>• Aucune gestion de la transparence</li> <li>• Rendu peu esthétique sur les aplats de couleur (sauf JPG haute qualité, plus lourd)</li> <li>• Mauvaise qualité si la compression est trop poussée (pixellisation)</li> </ul>	<ul style="list-style-type: none"> <li>• Photos sans transparence</li> <li>• Arrière-plans et images de grande taille</li> </ul>
<b>PNG</b> <i>Portable Network Graphics</i>  Extension : .png	<ul style="list-style-type: none"> <li>• Léger</li> <li>• Format sans perte (pas de pixellisation)</li> <li>• Possibilité de choisir le nombre de couleurs indexées (de 2 à 256)</li> <li>• Gestion partielle de la transparence (binaire)</li> </ul>	<ul style="list-style-type: none"> <li>• Limité à 256 couleurs maximum</li> <li>• Pas de couche alpha, uniquement des zones entièrement transparentes</li> </ul>	<ul style="list-style-type: none"> <li>• Visuels autres que photos avec <b>moins de 256 couleurs</b> et sans couche alpha (convient dans la majorité des cas) : logos, pictogrammes, schémas, icônes...</li> </ul>
<b>GIF</b> <i>Graphics Interchange Format</i>  Extension : .gif	<ul style="list-style-type: none"> <li>• Gestion des animations</li> <li>• Format sans perte (pas de pixellisation)</li> <li>• Possibilité de choisir le nombre de couleurs indexées (de 2 à 256)</li> <li>• Gestion partielle de la transparence (binaire)</li> </ul>	<ul style="list-style-type: none"> <li>• Limité à 256 couleurs max</li> <li>• Pas de couche alpha, uniquement des zones entièrement transparentes</li> <li>• Plus lourd que le PNG-8 pour un rendu identique</li> </ul>	<ul style="list-style-type: none"> <li>• Uniquement pour les animations</li> </ul>
<b>SVG</b> <i>Scalable Vector Graphics</i>  Extension : .svg	<ul style="list-style-type: none"> <li>• Format vectoriel</li> <li>• Très léger</li> <li>• Gestion de la transparence</li> <li>• Peut être utilisé sous forme de police de caractères (font-face)</li> <li>• Modifiable avec un éditeur de texte (syntaxe XML)</li> <li>• Balise &lt;svg&gt; dédiée en HTML5</li> </ul>	<ul style="list-style-type: none"> <li>• Non supporté par Internet Explorer 8 et ses versions antérieures</li> <li>• Uniquement pour les images vectorielles</li> <li>• Intégration plus complexe que pour une image classique</li> <li>• Implémentation variable de la balise &lt;svg&gt; selon les navigateurs</li> </ul>	<ul style="list-style-type: none"> <li>• Icônes, logos et pictogrammes (lorsque vous disposez des sources au format vectoriel), formes géométriques, polices de caractères</li> </ul>
<b>BMP</b> <i>Bitmap</i>  Extension : .bmp ou .dib	<ul style="list-style-type: none"> <li>• Format sans perte (pas de pixellisation)</li> <li>• Existe aussi en versions 256 couleurs, 16 couleurs ou monochrome</li> </ul>	<ul style="list-style-type: none"> <li>• Très lourd et long à charger</li> <li>• L'image se charge de bas en haut dans le navigateur</li> <li>• Pas de gestion de transparence</li> </ul>	<ul style="list-style-type: none"> <li>• Jamais sur le web !</li> </ul>

# LA PROPRIETE DISPLAY

La propriété CSS ***display*** permet de modifier le type de rendu d'un élément, les **balises s'affichant en « bloc »**

prennent toute la largeur disponible et s'affichent avec un saut à la ligne avant et après.

Les **balises s'affichant en « inline »** prennent uniquement la largeur dont elles ont besoin, sans ajouter de saut de ligne.

## Exemples de balises HTML qui s'affichent naturellement en block

- `<div>`
- Les balises de titre : `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`
- La balise paragraphe : `<p>`
- Les balises de liste : `<ul>`, `<ol>`, `<li>`
- La balise de citation : `<blockquote>`
- Les balises de tableaux : `<table>`
- **Exemples de balises HTML qui s'affichent naturellement en inline**
- `<span>`
- `<a>`
- `<strong>` , `<em>`, `<u>`
- `<img>`

# LA PROPRIETE DISPLAY

Voici quelques exemples de valeurs de la propriété display:

**display** : *block*; <- valeur par défaut

**display** : *inline*;

**display** : *inline-block*;

**display** : *table-cell*;

**display** : *list-item*;

**display** : *flex*;

**display** : *grid*;

**display** : *none*;

# LES POSITIONS

La propriété « *position* » permet de positionner un élément relativement à partir de sa position par défaut ou de façon absolue par rapport à un point donné dans la page en utilisation position conjointement avec les propriétés *top*, *left*, *bottom* et *right*.

*position* : *static* ; <- valeur par défaut

*position* : *relative* ;

*position* : *absolute* ;

*position* : *fixed* ;

*position* : *sticky*.



# REVISION SELECTEURS

Voyons maintenant les sélecteurs spécifiques aux champs de formulaire:

- `input[type=text]` sélectionne les champs de formulaire qui acceptent le **texte** .
- `input[type=password]` sélectionne les champs de formulaire qui acceptent les **mots de passe** .
- `input[type=number]` sélectionne les champs de formulaire qui acceptent les **nombres** .

- Passons à la pratique ici:

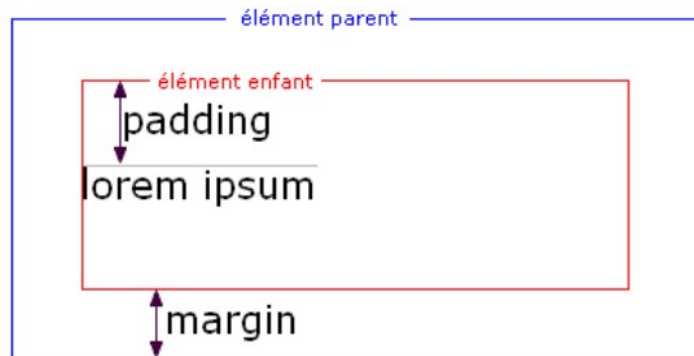
<https://www.bitdegree.org/learn/best-code-editor/css-forms-example-3>

- Pour aller plus loin:

<https://dev.to/oahehc/css-selectors-related-to-form-control-22pc>

# MARGIN VS PADDING

Une image vaut mieux qu'un long discours:



Margin = marge externe à l'élément.

Padding = marge interne à l'élément.

# LA PROPRIETE BACKGROUND

La propriété *background* permet de définir toutes les propriétés liées à l'arrière plan.

**background-color:** permet d'attribuer une couleur.

**Background-image:** ajouter une image d'arrière plan.

**background-repeat:** permet de gérer la répétition d'une.

**background-size:** permet de redimensionner une image.

**background-position:** permet d'indiquer la position.

# FLEXBOX

**Display:** `block` et **display:** `inline` ont été vu. Il existe aussi le **display:** `inline-block` et le `float` qui put prendre les valeur `left`, `right` ou `none`.

Ces derniers restent assez rigides, c'est ainsi qu'est apparu le **display:** `flex`;

Flexbox est un conteneur qui renferme des éléments (`inline` ou `block`) de telle sorte à ce que ces éléments là soient emboîtés d'une manière flexible sans générer de débordement ou de retour à la ligne incontrôlé.

# FLEXBOX

Avant d'appliquer des propriétés de flexbox à un élément, il faut que son parent soit en `display: flex;`

Voici quelques propriétés flexbox:

<https://jaetheme.com/blog/flexbox/>

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>

Amusons nous un peu à présent:

<https://flexboxfroggy.com/#fr>