

Chapitre 9 . Css : Positionnement des blocs

1- LES ELEMENTS INLINE :

L'image ci-contre est enregistrée dans le fichier *smiley.png* . Sa taille est de 300 x 300 pixels.
Le navigateur affiche la page suivante :



Le smiley		
Un smile est un dessin	extrêmement	stylisé d'un visage souriant , exprimant l'amitié .
		

On donne ci-contre
le contenu des
fichiers
exercice.html et
exercice.css de cette
page.

```
<body>
  <header>
    <h1>Le smiley</h1>
  </header>
  <section>
    <p>
      Un smile est un dessin <span class = "mot">extrêmement</span>
      stylisé d'un <b>visage souriant</b>,
      exprimant <span class = "mot" >l'amitié</span>.
    </p>
    
    
  </section>
</body>
```

exercice.html

```
/****** Css de section *****/
section {
}
section p{
}
section p span{
}

section img:nth-child(2){
  width:100px;
}
section img{
  width:150px;
}

section .mot{
  font-size: 50px;
  color: blue;
}
```

exercice.css

Point Cours : Type d'affichage des éléments --- Propriété Css `display:`

Les règles d'affichage des éléments d'une page dépendent de la balise utilisée en html.

Par défaut, sans Css, les éléments sont de type INLINE pour les balises `<input>` , `<label>` , `` , `` et aussi pour toutes ces petites balises qui s'insèrent dans un texte : `` , `` , `<u>` , `<sub>` , `<sup>` , ...

La plupart des autres balises sont de type BLOCK. C'est le cas des balises `<p>`, `<div>`, ``, ``, `<nav>` , `<section>`,

La propriété Css `display:` permet de **modifier** le type d'un élément :

- `display:inline;` : rend l'élément ciblé INLINE
- `display:block;` : rend l'élément ciblé BLOCK
- `display:none;` : le navigateur ignore l'élément lors de l'affichage
- `display:inline-block;` : type nouveau, sera vu dans ce chapitre
- `display:flex;` et `display:grid;` : seront vus ultérieurement

Point Cours : Caractéristiques des éléments INLINE :

- Ils n'occupent pas toute la largeur de la page. La largeur de l'élément dépend de son contenu.
- Les propriétés `width:` et `height:` n'ont aucun effet (sauf pour les images)
- La propriété `text-align:` appliquée au bloc parent, permet de centrer ou décaler sur la gauche, l'élément INLINE à l'intérieur du bloc parent.
- Si plusieurs éléments INLINE se trouvent dans un même bloc, la propriété `vertical-align:` appliquée à un élément INLINE, permet de le centrer verticalement, par rapport aux autres éléments qui se trouvent dans l'intérieur du bloc parent

On donne ci-dessous plusieurs copies d'écran de la page. A chaque étape une ou des propriétés Css ont été ajoutées.

⇒ On demande pour chaque étape, de donner la ou les lignes Css qui a été ajoutées.

⇒ ETAPE 1. :

Le smiley			
Un smile est un dessin	extrêmement	stylisé d'un visage souriant , exprimant	l'amitié .
			

⇒ ETAPE 2. :

Le smiley			
Un smile est un dessin	extrêmement	stylisé d'un visage souriant , exprimant	l'amitié .
			

⇒ ETAPE 3. :

Le smiley			
Un smile est un dessin	extrêmement	stylisé d'un visage souriant , exprimant	l'amitié
			
			

⇒ ETAPE 4. :

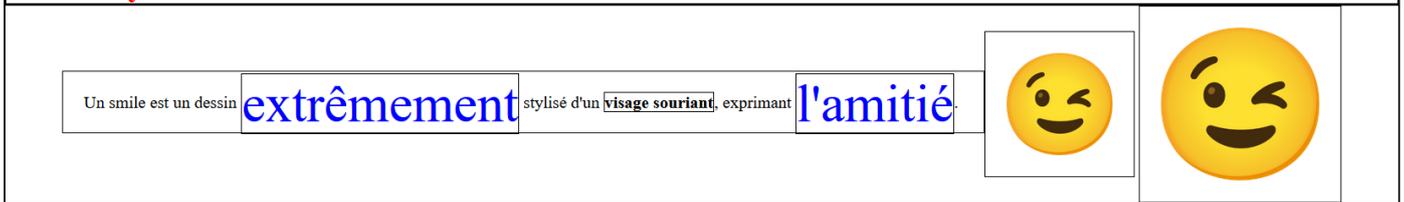
Le smiley			
Un smile est un dessin	extrêmement	stylisé d'un visage souriant , exprimant	l'amitié
			
			

⇒ ETAPE 5. :

Le smiley			
Un smile est un dessin	extrêmement	stylisé d'un visage souriant , exprimant	l'amitié
			
			

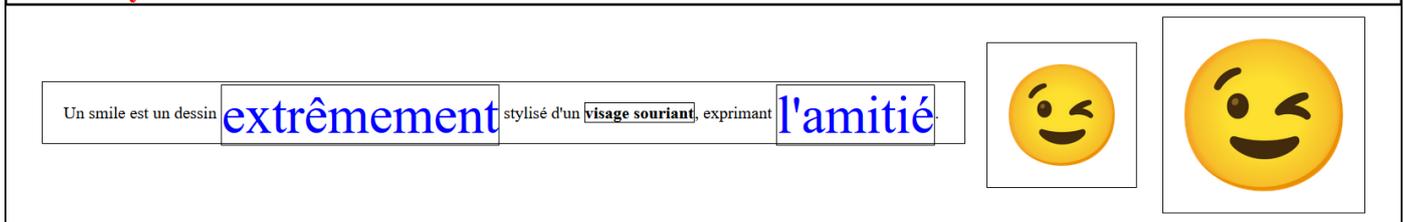
⇒ ETAPE 6. :

Le smiley



⇒ ETAPE 7. :

Le smiley



2- LES ELEMENTS BLOCK :

Point Cours : Caractéristiques des éléments BLOCK :

- Ils occupent par défaut toute la largeur de la page.
- Les propriétés `width:` et `height:` permettent de modifier leur dimensions :
 - Par défaut, la valeur de `width:` est à 100% de l'élément parent.
 - Avec `width:50%` , l'élément occupe 50% de la largeur de l'élément parent.
 - Avec `width:max-content` , l'élément prend les dimensions de son contenu.
 - Avec `width:min-content` , les dimensions sont réduites au minimum avec des retours à la ligne sur le contenu.
 - Avec `max-width:100%` , la largeur est au maximum égale à la largeur de l'élément parent.
 - Avec `max-width:calc(100% - 20px)` , la largeur est au maximum égale à la largeur de l'élément parent que l'on réduit de 20px.
- Les propriétés `text-align:` et `vertical-align:` appliquée à l'élément BLOCK, n'ont aucun effet.
- Le propriété `margin:auto` appliquée à l'élément BLOCK, permet de le centrer dans l'élément parent.

On donne ci-dessous plusieurs copies d'écran de la page. A chaque étape une ou des propriétés Css ont été ajoutées. On demande pour chaque étape, de donner la ou les lignes qui a été ajoutées.

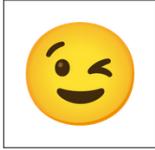
⇒ ETAPE 1. :

Le smiley	
Un smile est un dessin extrêmement stylisé d'un visage souriant , exprimant l'amitié .	 

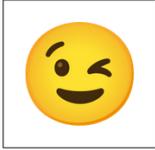
⇒ ETAPE 2. :

Le smiley	
Un smile est un dessin extrêmement stylisé d'un visage souriant , exprimant l'amitié .	 

⇒ ETAPE 3. :

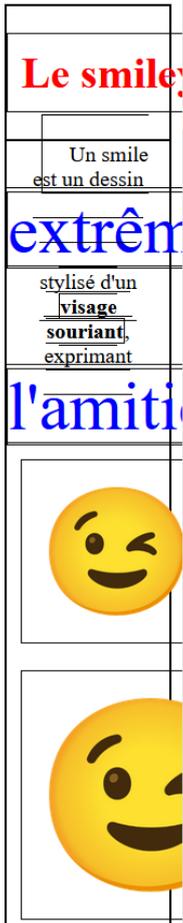
Le smiley	
Un smile est un dessin extrêmement stylisé d'un visage souriant , exprimant l'amitié .	 

⇒ ETAPE 4. :

Le smiley	
Un smile est un dessin extrêmement stylisé d'un visage souriant , exprimant l'amitié .	 

⇒ ETAPE 5. :

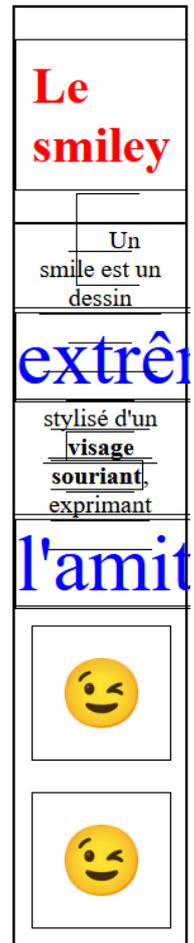
Le smiley	
Un smile est un dessin extrêmement stylisé d'un visage souriant , exprimant l'amitié .	 



La page affichée doit s'adapter à toutes les largeurs d'écran. En réduisant la largeur de la fenêtre, on constate sur la figure de gauche, que le bloc <h1>, fils de <header> est rogné. De même, les éléments images, fils de <section>, le sont aussi.

En rajoutant des propriétés **max-width:** dans le Css, on évite ce défaut et on obtient l'affichage de la figure de droite.

Quelles propriétés ont été rajoutées ?



Point Cours :

La propriété **word-wrap : break-word** appliqué à un élément qui contient du texte, donne l'autorisation au navigateur de « casser » les mots si la largeur de l'élément devient trop petite.

3- LES ELEMENTS INLINE-BLOCK :

Point Cours : Caractéristiques des éléments INLINE-BLOCK :

- Ils se comportent comme des INLINE par rapport à leur largeur. Ils n'occupent pas par défaut, toute la largeur de la page. Il est possible d'utiliser les propriétés **text-align:** , **vertical-align:**
- Ils se comportent comme des BLOCKS par rapport à la possibilité d'utiliser les propriétés **width:** , **height:** et **margin:auto**



On donne ci-après plusieurs copies d'écran de la page. A chaque étape une ou des propriétés Css ont été ajoutées. On demande pour chaque étape, de donner la ou les lignes qui a été ajoutées.

⇒ ETAPE 6. :

Le smiley	
Un smile est un dessin extrêmement stylisé d'un visage souriant , exprimant l'amitié	 

⇒ ETAPE 7. :

Le smiley	
Un smile est un dessin extrêmement stylisé d'un visage souriant , exprimant l'amitié	 

4- LA PROPRIETE POSITION :

Point Cours : La propriété CSS **position** définit la façon dont un élément est positionné sur une page. Les propriétés **top**, **right**, **bottom** et **left** déterminent alors l'emplacement final de l'élément positionné. Parmi les différentes valeurs que peut prendre cette propriété, on retient :

- **position:static** ; : valeur par défaut de tous les éléments.

En fonction des valeurs données à **top** et / ou **left** , ...

- **position:relative** ; : l'élément est décalé de sa position d'origine.
- **position:absolute** ; : l'élément est retiré du flux d'affichage de la page. Si *top* et *left* sont nuls, il se positionne généralement dans le coin haut-gauche de la page.
- **position:fixed** ; : pareil que *absolute*, mais ici l'élément se positionne dans le coin haut-gauche de l'écran.

Cela donne les affichages suivants pour l'exemple traité :

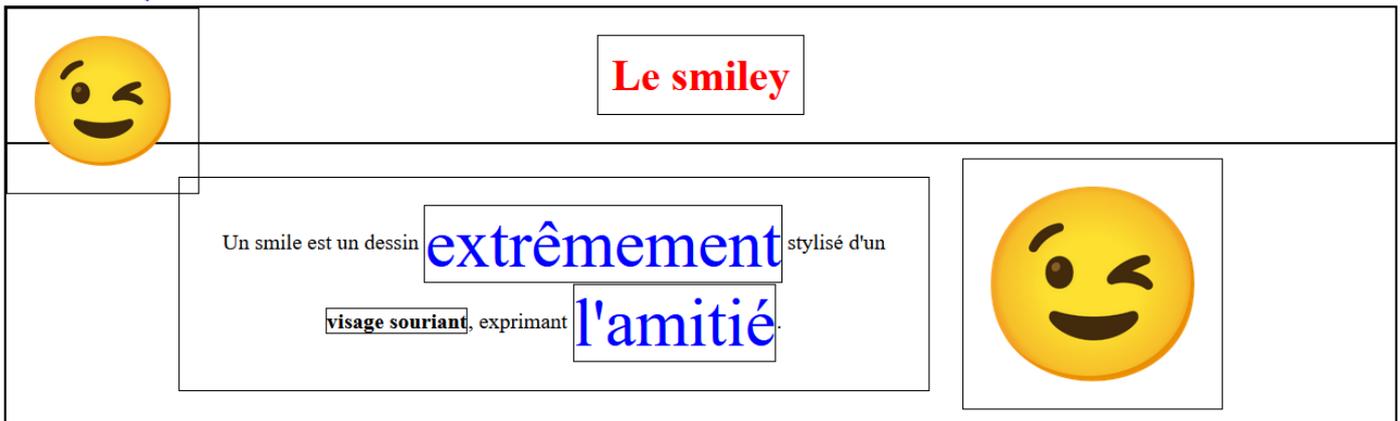
⇒ CAS 1 :

```
section img:nth-child(2){
  width:100px;
  position:relative;
  left:100px;
  top:-20px;
}
```



⇒ CAS 2 :

```
section img:nth-child(2){
  width:100px;
  position:absolute;
  left:0px;
  top:0px;
}
```



⇒ CAS 3 :

```
section img:nth-child(2){
  width:100px;
  position:fixed;
  left:0px;
  top:0px;
}
```

