



# Développement Web Côté client

## Module 7 – Positionnement des éléments



# Objectifs

## Positionnement des éléments

Développement Web côté client  
**POSITIONNEMENT DES ÉLÉMENTS**



Découvrir le positionnement en CSS



Découvrir float



Découvrir display



Découvrir position



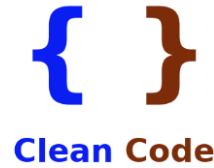
Découvrir flex

# Introduction

Positionnement des éléments

# Introduction

- Le positionnement des éléments est l'affaire du CSS
- Le HTML pose, quant à lui, la structure du document
- Ce positionnement doit être évolutif en fonction de l'appareil qui affichera le site Web.
  - Les éléments seront positionnés de manière différente selon que le site s'affiche sur un smartphone ou un écran de PC
- La position des éléments doit toujours passer par un travail préalable de réflexion :
  - Maquettage
  - Structure du document HTML (regroupement d'éléments dans un conteneur par exemple)



# Introduction

- Le positionnement des éléments est l'affaire du CSS
- Le HTML pose, quant à lui, la structure du document
- Ce positionnement doit être évolutif en fonction de l'appareil qui affichera le site Web.
  - Les éléments seront positionnés de manière différente selon que le site s'affiche sur un smartphone ou un écran de PC
- La position des éléments doit toujours passer par un travail préalable de réflexion :
  - Maquettage
  - Structure du document HTML (regroupement d'éléments dans un conteneur par exemple)



# Introduction

Le positionnement est une question difficile à maîtriser



# Les différentes alternatives

Positionnement des éléments

# Les différentes alternatives

Float  
Display  
Position  
Flex

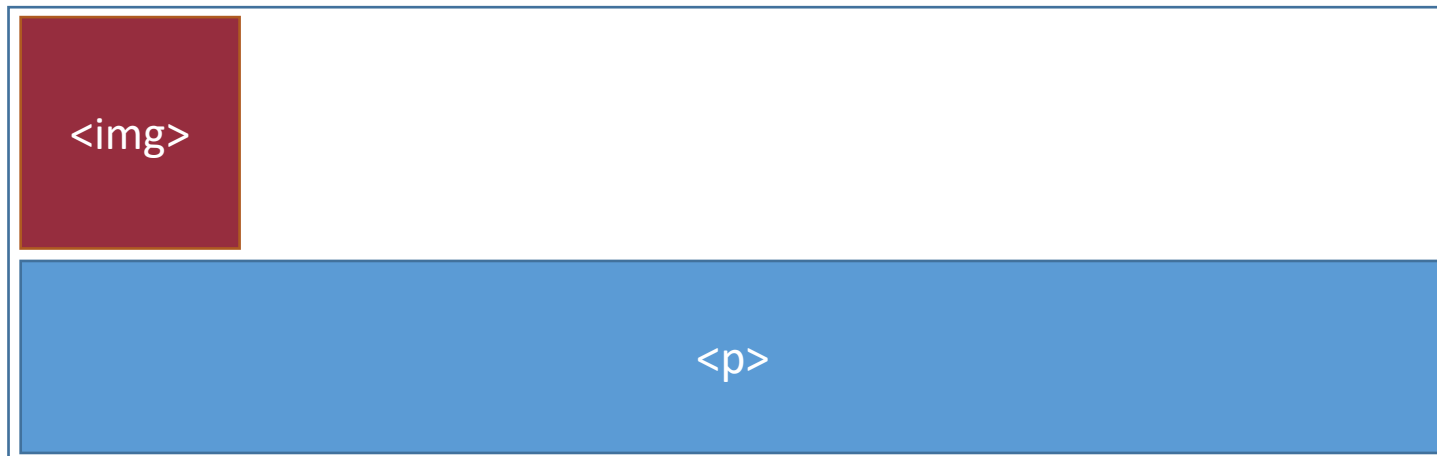


# Float

Positionnement des éléments

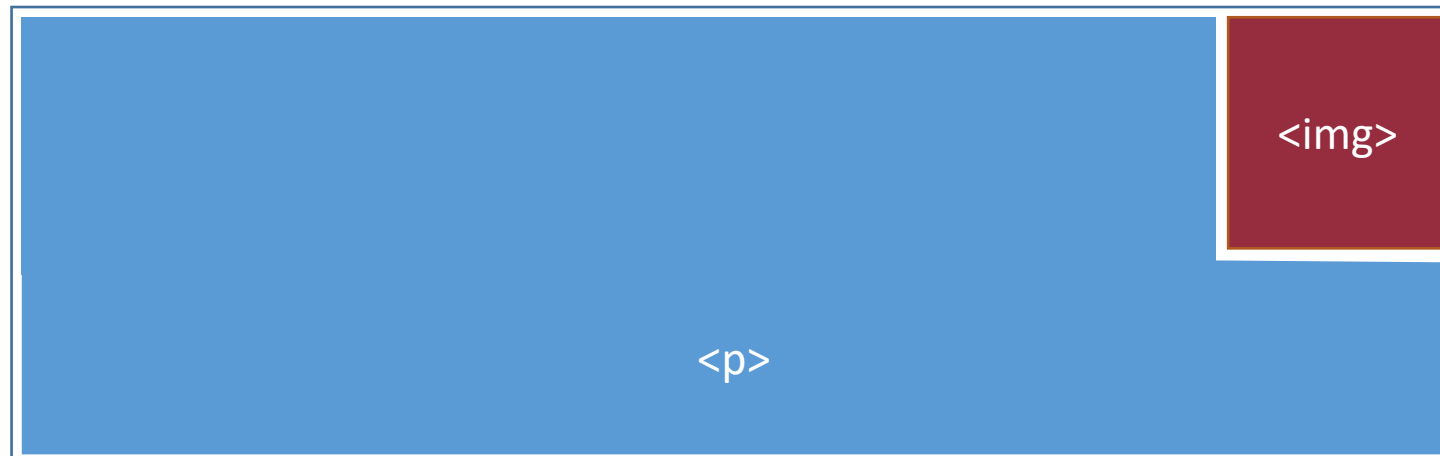
# Float

- **Float** est une propriété particulière qui modifie le flux du document (C'est-à-dire l'ordre et la façon dont les éléments s'affichent)
- Il est peu (voire plus du tout) utilisé en entreprise
- Il prend la valeur **left** ou **right**
- Après utilisation, il faut utiliser la propriété **clear** pour rétablir le flux



# Float

- **Float** est une propriété particulière qui modifie le flux du document (C'est-à-dire l'ordre et la façon dont les éléments s'affichent)
- Il est peu (voire plus du tout) utilisé en entreprise
- Il prend la valeur **left** ou **right**
- Après utilisation, il faut utiliser la propriété **clear** pour rétablir le flux



# DÉMONSTRATION

Float



# Display

Positionnement des éléments

# Display

- **Display** est une propriété qui influe sur le comportement d'un élément HTML.
- Chaque élément HTML possède un display par défaut (**block** ou **inline**)
- **Cette propriété ne s'hérite pas.** Modifier le display d'un élément parent ne modifiera pas celui des enfants

# Display

- **block** :
  - Fait un saut de ligne par rapport à l'élément qui précède et celui qui suit
  - Permet la modification de la taille de l'élément
- **inline** :
  - Se place sur la même ligne que les autres éléments voisins inline ou inline-block
  - Ne permet pas la modification de la taille. Prend automatiquement la taille de son contenu
- **inline-block** :
  - Se place sur la même ligne que les autres éléments voisins inline ou inline-block
  - Permet la modification de la taille de l'élément
- **none** :
  - Permet de cacher l'élément et de supprimer l'emplacement qui lui était prévu
  - Pour cacher un élément en gardant la place de celui-ci, il faut utiliser visibility

# Display

*block*



*inline*



*block + inline*





# DÉMONSTRATION

Display

# Position

Positionnement des éléments

## Position

- **Position** est une propriété qui permet de définir la manière dont un élément est placé dans le document.
- Cette propriété est très utile pour placer un élément précis.
- Il existe 5 valeurs possibles :
  - **Static** (valeur par défaut)
  - **Relative**
  - **Absolute**
  - **Fixed**
  - **Sticky**
- Pour spécifier la position on s'appuie sur les valeurs **top**, **bottom**, **left** et **right**

## Position - Relative

- **Relative** permet de décaler un élément par rapport à sa position d'origine.
- Sa place initiale est conservée et il n'y a pas de décalages des autres éléments
- Exemple :
  - position: relative;
  - top: 50px;
  - left: 50px;



## Position - Relative

- **Relative** permet de décaler un élément par rapport à sa position d'origine.
- Sa place initiale est conservée et il n'y a pas de décalages des autres éléments
- Exemple :
  - position: relative;
  - top: 50px;
  - left: 50px;



## Position - Absolute

- **Absolute** permet de décaler un élément par rapport à sa position d'origine.
- Sa place initiale est conservée et il n'y a pas de décalages des autres éléments
- Exemple :
  - position: absolute;
  - top: 50px;
  - left: 50px;



## Position - Absolute

- **Absolute** permet de décaler un élément par rapport à sa position d'origine.
- Sa place initiale est conservée et il n'y a pas de décalages des autres éléments
- Exemple :
  - position: absolute;
  - top: 50px;
  - left: 50px;



## Position - Fixed

- **Fixed** permet de fixer un élément sur la page en dehors du flux
- Sa place initiale n'est pas conservée et il y a un décalage des autres éléments
- Exemple :
  - position: fixed;
  - top: 0px;
  - left: 0px;





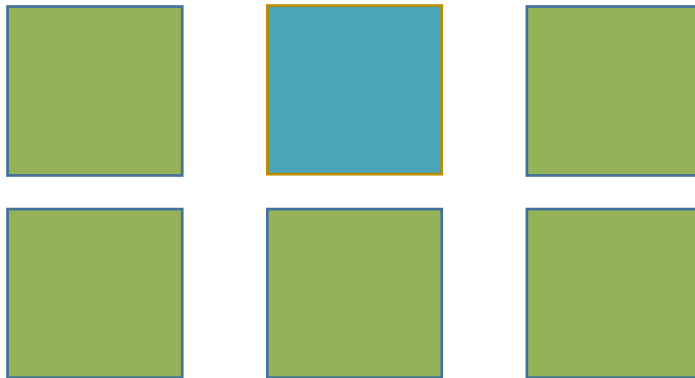
## Position - Fixed

- **Fixed** permet de fixer un élément sur la page en dehors du flux
- Sa place initiale n'est pas conservée et il y a un décalage des autres éléments
- Exemple :
  - `position: fixed;`
  - `top: 0px;`
  - `left: 0px;`



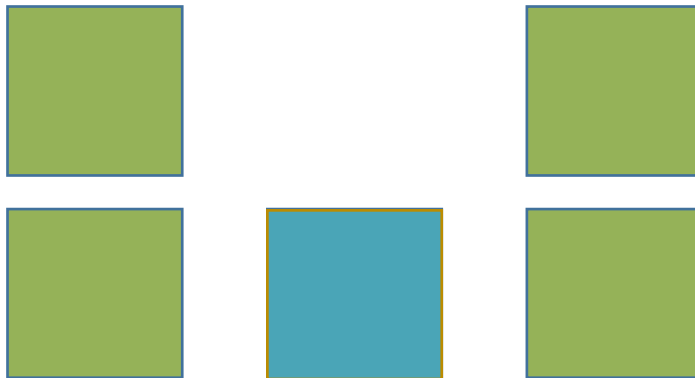
## Position - Sticky

- **Sticky** permet de fixer un élément sur la page en restant dans le flux
- Sa place initiale est conservée
- Exemple :
  - `position: fixed;`
  - `top: 50px;`
  - `left: 50px;`



## Position - Sticky

- **Sticky** permet de fixer un élément sur la page en restant dans le flux
- Sa place initiale est conservée
- Exemple :
  - `position: fixed;`
  - `top: 50px;`
  - `left: 50px;`



# DÉMONSTRATION

Position

# Flex

Positionnement des éléments

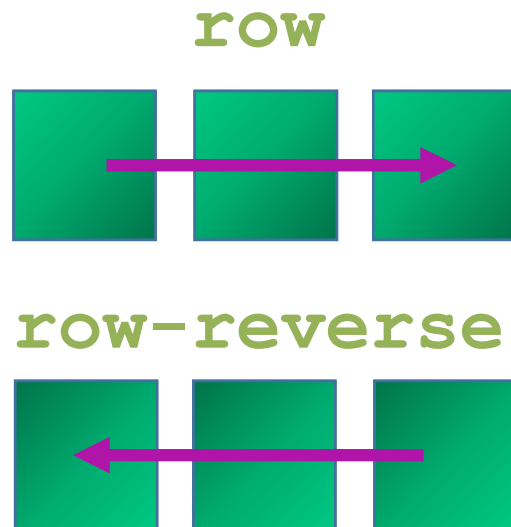
# Flex

- **Flex** est une valeur de display
- Cette valeur se place sur le **conteneur parent**. Une fois mise en place, ses enfants deviennent flexibles
- Une grande partie des propriétés associées à flex se place sur l'élément **parent** afin de gérer les enfants dans un même ensemble
- Quelques propriétés existent pour les enfants pour une gestion individuelle plus fine

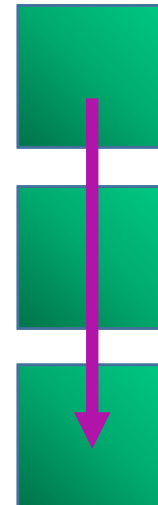
```
.parent {  
    display: flex;  
}
```

# Flex

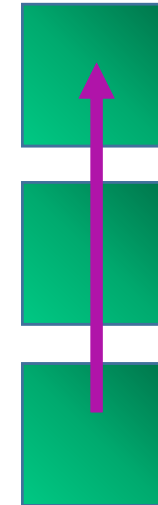
- Une fois que le conteneur parent est devenu flexible, on peut choisir l'axe principal d'alignement des éléments enfants grâce à **flex-direction**



**column**



**column-reverse**



# Flex

- Parfois, le contenu est trop large. Il faut donc prévoir le comportement de retour à la ligne avec **flex-wrap**





# Flex

- Choisir la répartition sur **l'axe principal**, `justify-content`



`Flex-start`



`Flex-end`



`center`



`Space-around`



`Space-between`



`Space-evenly`

# Flex

- Choisir la répartition sur **l'axe secondaire** :
  - Pour les éléments entre eux : **align-items**
  - Pour l'ensemble du contenu : **align-content** (similaire à justify-content)

# Flex

- Propriétés pour les enfants flexibles
  - **Order** : Permet de modifier l'ordre d'un élément
    - Fonctionne comme un coefficient (par défaut 0)
  - **Flex-grow** : Permet de modifier le volume d'un élément
    - Fonctionne avec un coefficient (par défaut 0)

# DÉMONSTRATION



## **TRAVAUX PRATIQUES**






# Flexy Frog

**TRAVAUX  
PRATIQUES**

# TP Fil rouge

# CSS

## CONCLUSION

-  Vous avez découvert le positionnement en CSS
-  Vous avez découvert le float
-  Vous avez découvert le display
-  Vous avez découvert la propriété position
-  Vous avez découvert le flex