

# FORMATION

## HTML5 / CSS3



**Openska**

# QU'ATTENDEZ-VOUS DE LA FORMATION ?

Qui êtes vous ?

Quel est votre profil ?

Où travaillez vous ?

Quel est votre parcours professionnel ?

Qu'attendez vous de la formation ?

# DÉROULEMENT D'UNE JOURNÉE

09:30 - 11:00

Pause

11:15 - 12:45

Pause déjeuner

14:00 - 15:15

Pause

15:30 - 17:00

# SOMMAIRE

Introduction

Syntaxe HTML : balises et attributs

HTML 5 : les éléments de base

HTML 5 : les formulaires

CSS 3 : concept de base

CSS 3 : positionnement, display et marge

CSS 3 : mise en forme

CSS 3 : fonctions avancées



# HTML 5

Hypertext Markup Language

Langage de balisage conçu pour représenter des pages web

Développé par W3C - World Wide Web Consortium

HTML5 encouragé à être utilisé depuis 2014



## Cascading Style Sheets

Langage conçu pour décrire la présentation des sites web

Développement dès 1999 mais développement extrêmement lent !



# CONTRAINTES NAVIGATEURS

Il est important de connaître sa cible avant tout

Écouter les contraintes, mais ne pas tomber dans l'abus ! Aujourd'hui Google Chrome occupe presque 60% d'utilisation, navigateurs ordinateurs et mobiles confondus

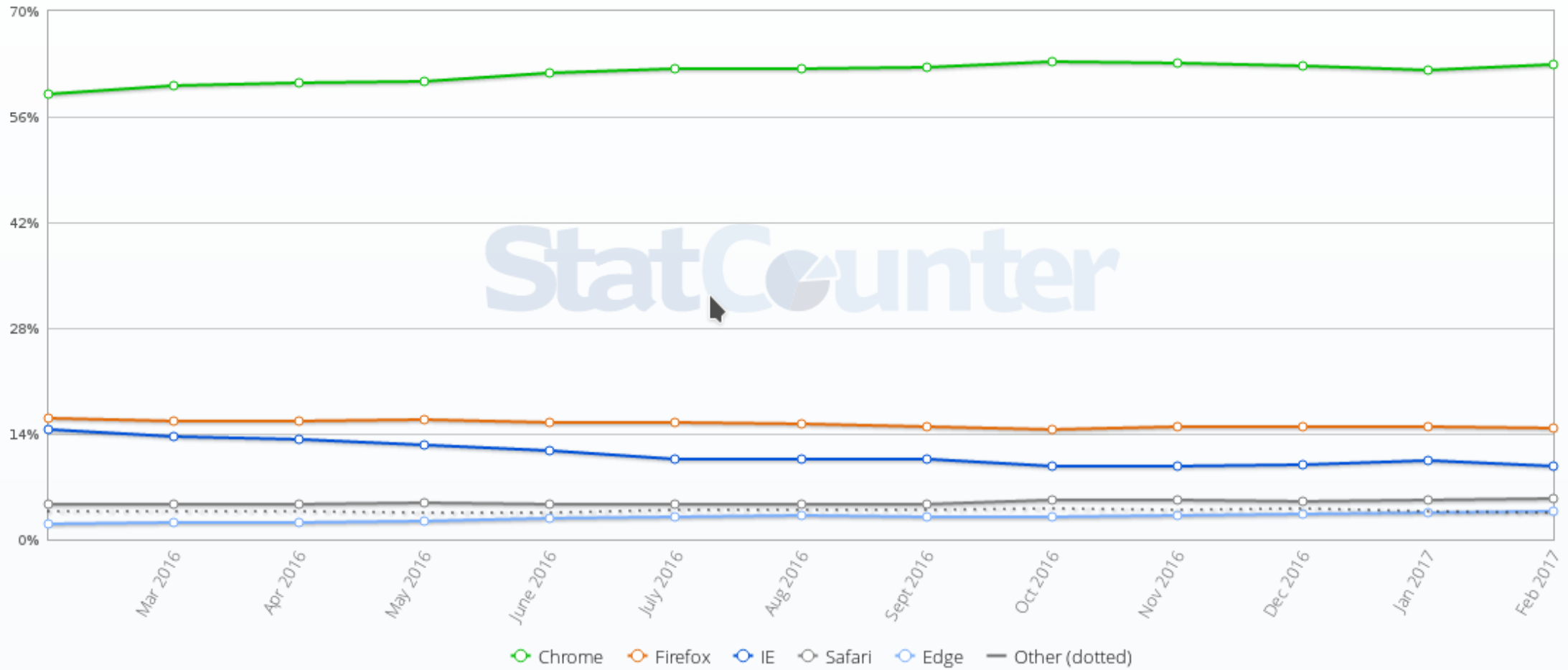
Internet Explorer ne maintient plus IE < 11 et Microsoft se concentre sur son nouveau navigateur: Edge

Safari est désormais le navigateur le plus en retard sur les dernières nouveautés HTML 5 / CSS 3



# Desktop Browser Market Share Worldwide

Feb 2016 to Feb 2017



# QUELQUES OUTILS

Un navigateur récent: Google Chrome, Firefox, Edge

Ou même encore plus récent: Google Chrome Canary, Firefox Developers

L'avantage d'un bon navigateur, les dernières nouveautés et un inspecteur d'éléments

Un éditeur de texte dernière génération: Atom, Visual Studio Code, Brackets, ...

# SYNTAXE HTML: BALISES ET ATTRIBUTS

# LES BALISES HTML

Langage de balisage

Les balises peuvent être imbriquées entre elles

Chaque balise peut avoir des attributs

Une balise peut être simple, ou double

Une balise doit toujours être fermée

Le HTML accepte les commentaires

On écrit les balises HTML en minuscules

```
<!-- Commentaires HTML -->
```

```
<!--  
    Commentaires sur  
    plusieurs lignes  
-->
```

```
<balise></balise>  
<balise />
```

```
<balise attribut="valeur"></balise>  
<balise attribut="valeur" />
```

```
<balise attribut="valeur">  
    <balise>Contenu textuel</balise>  
    <balise>Autre contenu ...</balise>  
</balise>
```

<balise>

<balise>

</balise>

<balise />

</balise>

# STRUCTURE D'UN DOCUMENT

Toujours indiquer au navigateurs qu'on utilise HTML 5

Le squelette d'un document HTML 5 est simple, mais doit être respecté

On pourra compléter la structure par un titre, la définition de la langue et du jeu de caractères

```
<!DOCTYPE html>
<html>

  <head>
    <meta charset="UTF-8" />
    <meta http-equiv="Content-Language" content="fr" />

    <title>Openska - Formation HTML 5 / CSS 3</title>
  </head>

  <body>

    <!-- LE CONTENU VISIBLE DE LA PAGE -->

  </body>

</html>
```

# HTML 5

## LES ÉLÉMENTS DE BASES

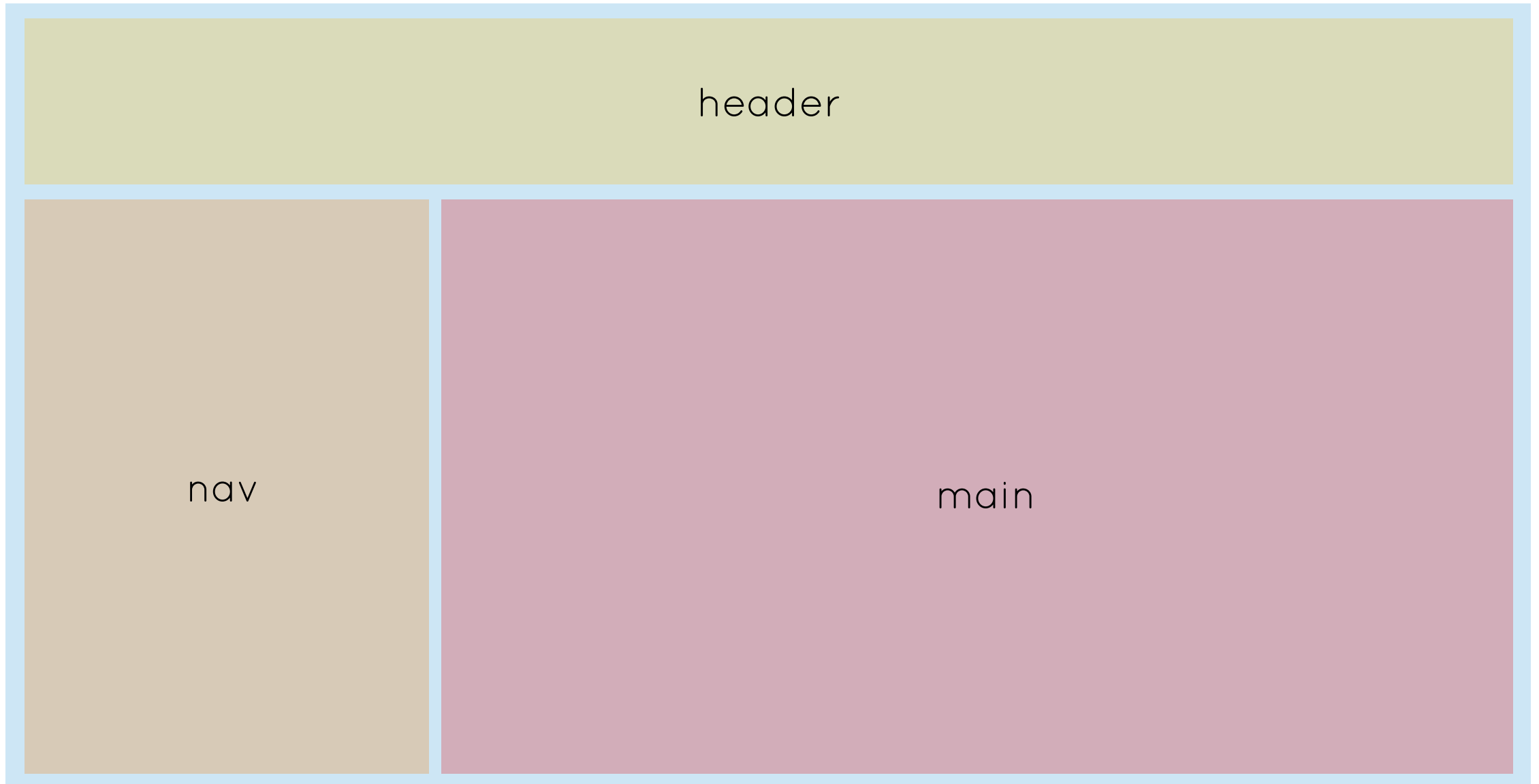


# STRUCTURER SA PAGE

HTML 5 met à disposition des balises pour structurer ses pages

Il est important de bien structurer ses pages pour qu'elles soient compréhensibles par les utilisateurs, mais également pour être correctement référencer sur les différents moteurs de recherches

On pourra par exemple indiquer des éléments de navigations, des entêtes, des sections, ...



```
<!-- ... -->
```

```
<body>
```

```
    <header></header>
```

```
    <nav></nav>
```

```
    <main>
```

```
        <article></article>
```

```
    </main>
```

```
    <footer></footer>
```

```
</body>
```

```
<!-- ... -->
```

```
<!-- EN TÊTE -->
```

```
<!-- MENU DE NAVIGATION -->
```

```
<!-- CONTENU PRINCIPAL -->
```

```
<!-- CONTENU PLUS PRÉCIS -->
```

```
<!-- PIED DE PAGE -->
```

# STRUCTURER SON TEXTE

Bien souvent, notre page contiendra principalement du texte

C'est le contenu qui fait vivre un site car c'est lui qui attire des utilisateurs

Attention, les Hommes ne sont pas les seuls à visiter les sites, il est donc important de structurer son contenu. Savoir quels sont les titres ,les citations, les mots importants, ...

Comment déterminez vous le titre, la citation, l'auteur de la citation dans la structure suivante ?

## Lorem Ipsum

Lorem ipsum dolor sit amet, consectetur adipiscing elit. **At reprehenderit** sequi ducimus aliquam veritatis voluptatem provident distinctio quia expedita! Harum rem nihil cupiditate, at, officiis modi ab vero nesciunt natus?

Sapiente doloribus, esse, nam iure repellendus, quo vero debitis corporis quae et perspiciatis.

Victor Lipsum

Les repères visuels sont un facteur important pour repérer ses éléments. Cependant les machines (comme les moteurs de recherches) ne les verront pas!

Les balises existent pour améliorer la compréhension des machines sur le contenu des pages

```
<article>

  <h1>Lorem Ipsum</h1>

  <p>
    Lorem ipsum dolor sit amet, consectetur adipisicing elit. <strong>At reprehenderit</strong> ...
  </p>

  <blockquote>

    <p>Sapiente doloribus, esse, nam iure repellendus ...</p>
    <footer>Victor Lipsum</footer>

  </blockquote>

</article>
```

# METTRE EN FORME SON TEXTE

Il est possible de mettre en forme son texte, sans pour autant en changer la structure

On pourra ainsi faire ressortir une partie du texte (utiliser du gras), sans changer de paragraphe ou faire un retour à la ligne

Jusqu'à maintenant les éléments de structure nous faisaient retourner à la ligne



# Lorem Ipsum dolor sir amet

Bienvenue chez **Openska**

La formation que vous suivez est HTML 5 / CSS 3

Il existe plusieurs balises pour mettre on texte en forme: b, em, i, sub, sup, ins, del, ...

```
<!-- ... -->
```

```
<h1>Lorem Ipsum <small>dolor sir amet</small></h1>
```

```
<p>
```

```
  Bienvenue chez <strong>Openska</strong>
```

```
</p>
```

```
<p>
```

```
  La formation que vous suivez est <mark>HTML 5 / CSS 3</mark>
```

```
</p>
```

```
<p>
```

```
  Il existe plusieurs balises pour mettre son texte ...
```

```
</p>
```

```
<!-- ... -->
```

# LES LIENS

Les liens sont très important sur internet, c'est grâce à eux qu'on peut naviguer, découvrir de nouveaux sites et de nouveaux contenus

Il existe différents types de liens: les ancres, les liens internes vers des pages de son site et des liens externes

Un lien sera au moins composé d'un contenu visible, d'un titre et du lien de la ressource vers laquelle il mène

```
<!-- Lien vers une page du même site -->
<a href="page.html" title="Page de mon site">Page de mon site</a>

<!-- Lien vers un contenu externe au site -->
<a href="http://openska.com" title="Openska">Openska</a>

<!-- Lien vers une ancre sur la même page -->
<a href="#chapitre-15" title="Chapitre 15">Aller au chapitre 15</a>
<!-- ... -->
<h3 id="chapitre-15">Chapitre 15</h3>

<!-- Lien vers une ancre d'une autre page -->
<a href="chapitres.html#chapitre-15" title="Chapitre 15">Chapitre 15</a>
```

# LES IMAGES

HTML permet également d'indiquer la présence d'une image

Les images doivent avoir un texte alternatif et un lien source

Les images peuvent être accompagnées de légendes

D'autres attributs peuvent compléter une image, comme un titre, la largeur, la hauteur, ...

```
<!-- ... -->



<figure>

  <figcaption>
    <!-- Légende de l'image -->
    Openska, organisme de formation.
  </figcaption>

</figure>

<!-- ... -->
```

# LES LISTES

Il existe deux types de listes, les listes à puces (non-ordonnées) et les listes ordonnées (numériquement ou alphabétiquement)

Les listes sont composées d'un élément qui contient toute la liste et de plusieurs éléments qui contiennent chaque objet de la liste

Les listes peuvent être imbriquées

```
<ul>
  <li>Liste non-ordonnée</li>
  <li>Liste non-ordonnée</li>
  <li>Liste non-ordonnée</li>
</ul>

<ol>
  <li>Liste ordonnée</li>
  <li>Liste ordonnée</li>
  <li>Liste ordonnée</li>
</ol>

<ul>
  <li>Liste non-ordonnée</li>
  <li>
    Liste non-ordonnée
    <ol>
      <li>Liste ordonnée</li>
      <li>Liste ordonnée</li>
      <li>Liste ordonnée</li>
    </ol>
  </li>
  <li>Liste non-ordonnée</li>
</ul>
```



# LES BALISES GÉNÉRIQUES

On a vu jusqu'à des balises qui avaient chacune leur utilité

Elles ont aussi chacune un rendu particulier sur le navigateur, des propriétés qui change de l'une à l'autre (C'est pourquoi un titre sera en gras et écrit plus gros)

Enfin certaines balises influent sur la structure même du document (retour à la ligne)

# LES BALISES GÉNÉRIQUES

Il existe deux balises génériques, qui ont un minimum de style associé

La balise "div" qui est une balise structurelle

la balise "span" qui est une balise de mise en forme

# HTML 5

## LES FORMULAIRES

# LES FORMULAIRES

Les formulaires en HTML 5 utilisent un grand nombre de balises et d'attributs

Grâce aux formulaires il existe de nombreuses possibilités pour interagir avec ses utilisateurs

Ils permettent l'envoi de données vers le serveur, qui pourront ensuite être traitées

# LES BALISES POUR FORMULAIRES

La balise "form" permet de définir un nouveau formulaire

La balise "fieldset" permet de diviser son formulaire en plusieurs parties

La balise "legend" permet de donner un titre à une partie

La balise "label" permet d'indiquer quelle information doit être renseignée dans le champ qu'elle indique

# LA BALISE INPUT

La balise "input" est la balise la plus utilisée dans les formulaires

Son attribut type peut prendre de nombreuses valeurs et changera grandement le comportement de la balise

Liste non-exhaustive des différents types : text, password, submit, radio, checkbox, email, color, range, tel, ...

```
<input type="text" />

<input type="text" id="input_username" name="username"/>
<input type="password" placeholder="mot de passe"/>
<input type="email" value="defaultvalue@email.com"/>

<input type="radio" name="formation" value="1" />
<input type="radio" name="formation" value="2" />

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

<input type="color" />
<input type="range" min="0" max="100"/>
```

# LISTE DÉROULANTE

Il existe une balise "select" pour réaliser des listes déroulantes dans nos formulaires

Cette balise s'utilise avec une ou plusieurs balises "option" qui correspondent aux différentes valeurs possible de notre liste

La balise "select" peut aussi avoir un attribut "multiple" qui permet de faire un choix multiple dans la liste



```
<select name="country">
  <option value="fr">France</option>
  <option value="be">Belgique</option>
</select>

<select name="activities" multiple>
  <option value="sport">Sport</option>
  <option value="games">Jeux</option>
  <option value="reading">lecture</option>
</select>
```

## ZONE DE TEXTE

Pour les textes plus long, comme la rédaction d'un article ou d'un commentaire, une balise "textarea" existe

On peut définir la taille de la zone de texte grâce aux attributs "rows" et "col"

Par défaut, la taille de la zone de texte peut être ajuster par l'utilisateur

```
<textarea></textarea>
```

```
<textarea name="article">Valeur par défaut</textarea>
```

```
<textarea rows="3" cols="20"></textarea>
```

# CSS 3

## CONCEPT DE BASE

# SYNTAXE

La syntaxe CSS comporte la sélection d'un ou plusieurs éléments suivi de différentes propriétés qui définissent son style

```
selector {  
    property: value;  
    property: value;  
  
    /* This is a comment */  
}
```

# OÙ METTRE SON STYLE ?

Trois choix s'offrent à nous !

Dans un attribut "style" qui contiendra les propriétés de style (mauvaise pratique)

Dans une balise "style" située dans l'entête du document HTML (mauvaise pratique, sauf pour du style dynamique)

Dans une feuille de style, un fichier CSS séparé, inclus dans notre HTML (bonne pratique)

```
<!DOCTYPE html>
<html>
  <head>
    <!-- METHODE : dans une balise style -->
    <style>
      selector {
        property: value;
        property: value;
      }
    </style>

    <!-- METHODE : dans une feuille de style à part -->
    <link rel="stylesheet" href="style.css" />
  </head>

  <body>
    <!-- METHODE : en tant qu'attribut -->
    <div style="property: value; property: value;"></div>
  </body>
</html>
```

# LES SÉLECTEURS

On utilise les sélecteurs pour indiquer sur quel(s) élément(s) on applique le style dans les propriétés qui suivent

On peut grâce aux sélecteurs, sélectionner très précisément un unique élément, ou de façon plus large, tout un ensemble d'éléments



# SÉLECTEUR UNIVERSEL

Le sélecteur universel permet de sélectionner tous les éléments

Il est utile pour définir un style par défaut sur l'ensemble du document, comme par exemple modifier la police sur l'ensemble du document

```
* {  
  property: value;  
}
```

# SÉLECTEURS PAR BALISES

Moins sélectif que le précédent, ce sélecteur va permettre de sélectionner toutes les balises qui lui correspondent

```
p {  
  property: value;  
}
```

# SÉLECTEURS PAR ID ET CLASSES

En HTML, un ID doit être unique par page. Avec le CSS on va pouvoir sélectionner cet unique élément

En HTML, une classe peut être multiple. Avec le CSS on va pouvoir sélectionner plusieurs éléments (normalement) semblables

```
#main-logo {  
    property: value;  
}
```

```
.button {  
    property: value;  
}
```

# SÉLECTEURS PAR ATTRIBUTS

Les ID et les classes ne sont pas les seuls attributs qui nous permettent de sélectionner des éléments

En réalité, n'importe quel attribut peut être la cible d'un sélecteur, il est juste plus simple de le faire avec des ID et des classes

```
[data-attribute] {  
  property: value;  
}
```

```
[data-attribute='specific-value'] {  
  property: value;  
}
```

# COMBINER LES SÉLECTEURS

Il est parfois utile ou nécessaire de sélectionner uniquement des combinaison de sélecteur, par exemple uniquement des "li" qui ont la classe "menu"

Très utile pour sélectionner uniquement les input de type texte par exemple !

```
input[type='text'] {  
  property: value;  
}
```

```
li.menu {  
  property: value;  
}
```

# SÉLECTION MULTIPLE

Une sélection ne couvre pas forcément tout nos besoins

On voudra par exemple sélectionner tous les boutons avec la classe submit et tous les input de type submit

```
button.submit, input[type='submit'] {  
  property: value;  
}
```

# SÉLECTION DE DESCENDANTS

On va pouvoir par exemple sélectionner tous les titres principaux uniquement présent dans une balise d'entête

Attention, ce sera bien tous les descendants, peut importe qu'il soit un enfant direct ou non

```
<!-- ... -->

<div>
  <header>
    <h1>Sélectionné</h1>
    <div>
      <h1>Également sélectionné</h1>
    </div>
  </header>
  <h1>Pas sélectionné :(</h1>
</div>

<!-- ... -->
```

```
header h1 {
  property: value;
}
```

# SÉLECTION D'ENFANTS DIRECTS

Des cas particuliers peuvent survenir où l'on doit sélectionner que des enfants directs, et pas tous les descendants

```
<!-- ... -->
<div>
  <header>
    <h1>Sélectionné</h1>
    <div>
      <h1>Pas enfant direct</h1>
    </div>
  </header>
  <h1>Pas sélectionné :( </h1>
</div>
<!-- ... -->
```

```
header > h1 {
  property: value;
}
```



# PSEUDO ÉLÉMENTS

Un pseudo élément peut être ajouté à un sélecteur pour modifier le comportement d'une partie du document ou de l'élément

Il en existe plusieurs (after, before, selection, ...) et certains sont encore expérimentaux (placeholder, grammar-error, ...)

```
p::selection {  
  property: value;  
}
```

# PSEUDO CLASSES

Une pseudo classe est semblable a un pseudo élément dans l'utilisation, il s'ajoute à l'élément, mais pour sélectionner celui ci selon un état précis

Il existe un plus grand nombre de pseudo classes (active, focus, not, visited, nth-child(), empty, ...)

```
a:visited {  
    property: value;  
}
```

```
button:hover {  
    property: value;  
}
```

# PRIORITÉ DES SÉLECTEURS

En CSS, ce qu'il compte en premier lieu n'est pas l'ordre dans lequel est écrit le code, mais la priorité des sélecteurs

Dans le cas où plusieurs sélecteurs ont la même priorité, c'est le dernier qui a été défini qui l'emporte sur les autres

Le sélecteur universel n'a aucune priorité

Le sélecteur par ID a une très forte priorité

# CSS 3

## POSITIONNEMENT, DISPLAY ET MARGES

# POSITIONNEMENT

Une fois la structure de notre site définie grâce au HTML, elle n'a pas forcément la disposition voulu !

On peut alors modifier la position de nos différents éléments avec notre CSS

Par défaut le positionnement d'un élément est statique

# POSITION FIXE

Les éléments HTML sont les uns à la suite des autres par défaut

Lorsque le contenu dépasse la taille de notre navigateur, une barre de défilement apparaît

On peut fixer un élément pour que sa position soit constante par rapport au navigateur

Cet élément ignorera le positionnement des autres, et inversement

```
<!-- ... -->
```

```
<div>
```

```
  <p>Lorem ipsum dolor sit amet ...</p>
```

```
  <div id="fixed-bar">CONTENU FIXE</div>
```

```
  <p>Lorem ipsum dolor sit amet ...</p>
```

```
</div>
```

```
<!-- ... -->
```

```
#fixed-bar {  
  position: fixed;  
  top: 10px;  
  left: 10px;  
}
```

# POSITION RELATIVE ET ABSOLUE

Position relative et absolue sont liées

Une position absolue déterminera sa position par rapport au premier parent ayant une position relative

Les positions relative et absolue sont parfois critiquées à tort, bien utilisées ces positions sont très utiles et intéressantes à utiliser



```
<!-- ... -->
```

```
<div>
```

```
  <p>Lorem ipsum dolor sit amet ...</p>
```

```
  <div id="relative-block">
```

```
    <p>Lorem ipsum dolor sit amet ...</p>
```

```
    <div id="absolute-content">CONTENU ABSOLU</div>
  </div>
```

```
</div>
```

```
<!-- ... -->
```

```
#relative-block {
  position: relative;
}
```

```
#absolute-content {
  position: absolute;
  top: 10px;
  left: 10px;
}
```

# POSITION COLLANTE

Position encore expérimentale mais au futur prometteur !

Mélange entre position fixe et position relative !?

Comportement anciennement géré en javascript, souvent mal fait ou au comportement détourné

# DISPLAY

Nous avons déjà abordé cette notion !

Différence entre éléments structurel et élément de mise en forme : certains nous ramène à la ligne, d'autre non

On va ainsi distinguer les éléments "block" et les éléments "inline"

Il existe d'autres valeurs de display, notamment "flex" et "none", et bien d'autre moins utilisés

# DISPLAY FLEX

Cette élément se comporte comme un élément "block", mais son contenu se comportera comme un système de boites flexibles

Nouveau, il révolutionne la manière de gérer les différentes colonnes !

Les éléments dans un élément flexible peuvent avoir des propriétés CSS supplémentaires

```
<div class="row">
  <div class="col">This is the left column</div>
  <div class="col">This is the right column</div>
</div>
```

```
.row {
  display: flex;
}

.col {
  flex: 1;
}
```

<https://www.alsacreations.com/tuto/lire/1493-css3-flexbox-layout-module.html>



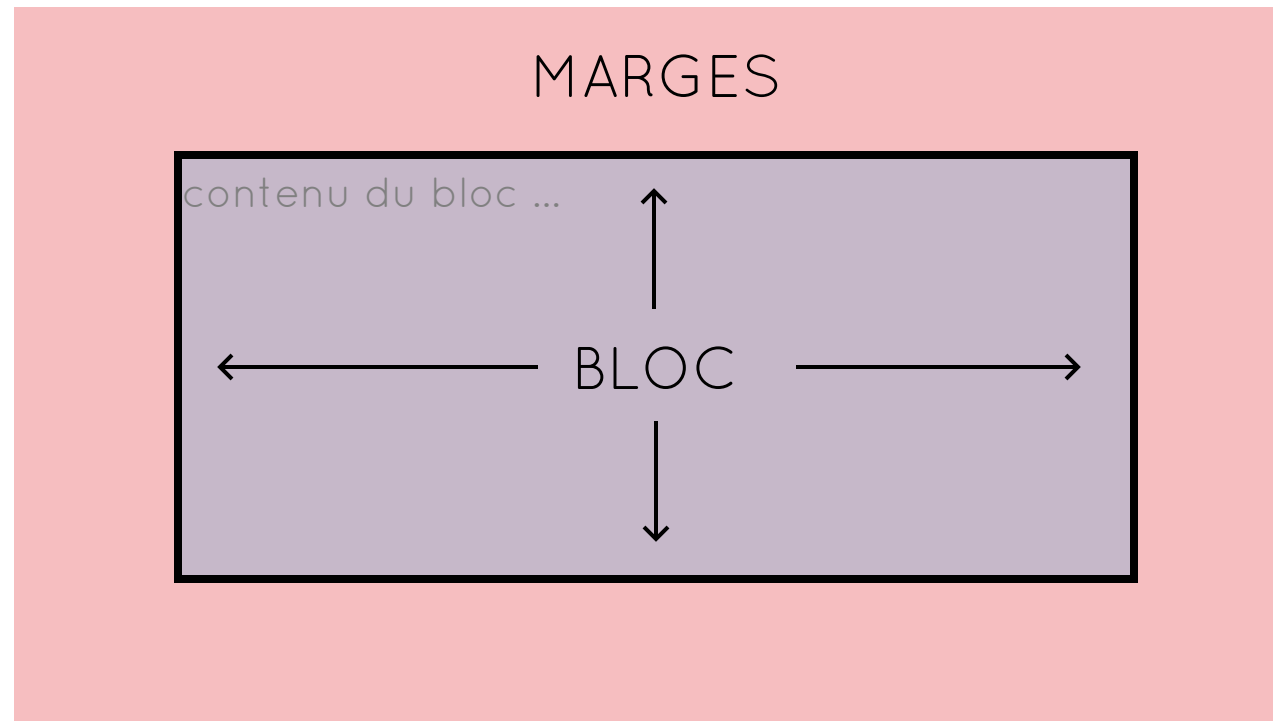
## DISPLAY NONE

L'élément et tous ses descendants (enfants directs et contenu) ne seront plus affichés

La page sera alors affichée comme si l'élément n'existait pas

# MARGES EXTÉRIEURES

Les marges extérieures sont utiles pour espacer les blocs entre eux



```
<div class="marges-externes">  
    contenu du bloc ...  
</div>
```

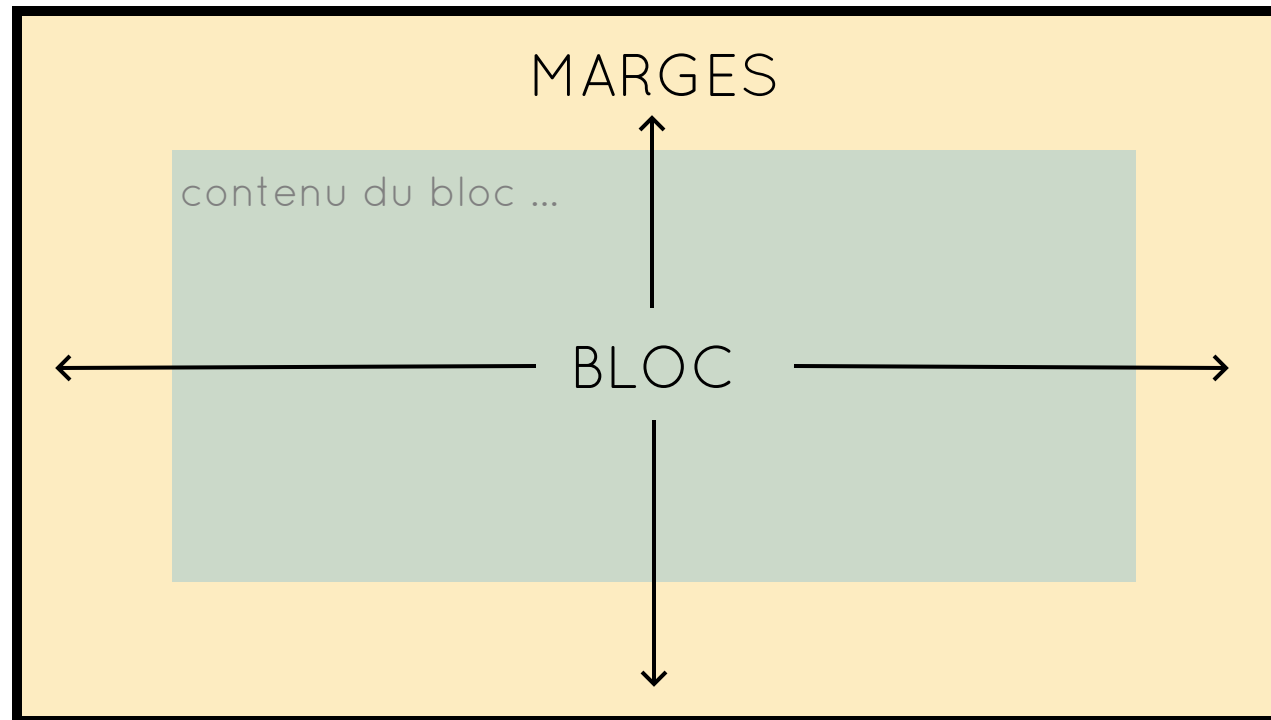
```
.marges-externes {  
    margin: 50px;  
}
```

```
.marges-externes {  
    margin-left: 50px;  
    /* MARGE UNIQUEMENT À GAUCHE */  
}
```



# MARGES INTÉRIEURES

Les marges intérieures sont utiles pour espacer le contenu à l'intérieurs de bloc



```
<div class="marges-internes">
    contenu du bloc ...
</div>
```

```
.marges-internes {
    box-sizing: border-box;
    padding: 50px;
}
```

```
.marges-internes {
    padding-right: 50px;
    /* MARGE UNIQUEMENT À DROITE */
}
```

# CSS 3

## MISE EN FORME

# TEXTES ET POLICES D'ÉCRITURES

Les textes peuvent être personnalisés, que cela soit en taille, en couleur, en police d'écriture, ...

Plus largement on va même pour intervenir sur la structure d'un bloc de texte, en changeant l'indentation, l'alignement ou encore l'espacement

# MODIFICATIONS DU TEXTE

```
h1 {  
  text-decoration: overline;  
  text-decoration: line-through;  
  text-decoration: underline;  
}
```

```
h3 {  
  text-transform: uppercase;  
  text-transform: lowercase;  
  text-transform: capitalize;  
}
```

```
h3 {  
  line-height: 15px;  
  letter-spacing: 3px;  
  word-spacing: 10px;  
}
```

```
p {  
  text-indent: 50px;  
  text-align: justify;  
}
```

# MODIFICATIONS DE LA POLICE

```
h4 {  
    font-family: arial, helvetica, sans-serif;  
}  
  
h5 {  
    font-style: normal;  
    font-style: italic;  
}  
  
h6 {  
    font-size: 20px;  
}  
  
div {  
    font-weight: normal;  
    font-weight: bold;  
  
    font-variant: normal;  
    font-variant: small-caps;  
}
```

# AJOUTER SA POLICE D'ÉCRITURE

Chaque système d'exploitation à des polices d'écritures différentes

On peut avoir besoin d'une police particulière, pour qu'elle soit identique pour tous nos utilisateurs, pour mieux se fondre avec le style de la page, ou pour respecter la charte graphique d'une marque

```
@font-face {  
  font-family: my-custom-font;  
  src: url(my-custom-font-file.woff);  
}  
  
* {  
  font-family: my-custom-font, sans-serif;  
}
```

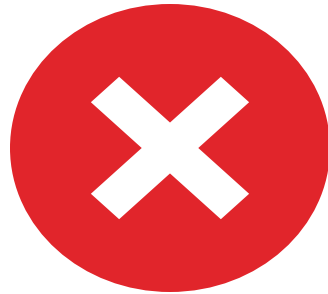
Fonctionnalité	Firefox (Gecko)	Chrome	Edge	Internet Explorer	Opera	Safari
Support simple	3.5 (1.9.1)	4.0	(Oui)	4.0	10.0	3.1
WOFF	3.5 (1.9.1)	6.0	(Oui)	9.0	11.10	5.1
WOFF2	39 (39)[1]	38	(Oui)	Pas de support	24	Pas de support
Polices SVG[2]	Pas de support[3]	(Oui)	Pas de support	Pas de support	(Oui)	(Oui)
unicode-range	36 (36)	(Oui)	(Oui)	9.0	(Oui)	(Oui)

Fonctionnalité	Android	Firefox Mobile (Gecko)	Edge	IE Mobile	Opera Mini	Opera Mobile	Safari Mobile
Support simple	(Oui)	1.0 (1.9.1)	(Oui)	?	Pas de support	10.0	(Oui)
WOFF	4.4	5.0 (5.0)	(Oui)	10.0	Pas de support	11.0	5.0
WOFF2	Pas de support	39.0 (39.0)[1]	(Oui)	Pas de support	?	?	Pas de support
Polices SVG[2]	?	Pas de support[3]	Pas de support	?	Pas de support	10.0	(Oui)
unicode-range	?	36.0 (36.0)	(Oui)	?	Pas de support	?	(Oui)



# COULEURS BORDURES ET FONDS

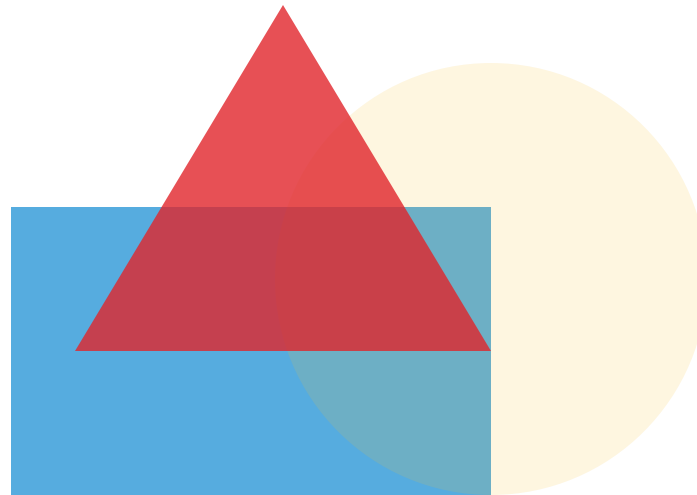
Maintenant que la structure est faite, notre site reste froid, sans couleur ...



# GESTION DES COULEURS ET TRANSPARENCES

Les couleurs peuvent être générées de plusieurs façon

On peut avoir des couleurs opaques, ou des couleurs plus ou moins transparentes



```
/* OPAQUE */
p {
    color: rgb(255, 0, 0);          /* RED GREEN BLUE, Valeur de 0 à 255 */
    color: hsl(0, 100%, 50%);       /* HUE SATURATION LIGHTNESS, Valeur de 0 à 360, puis 0 à 100 */
    color: #FF0000;                 /* RGB HEXADECIMAL, Valeur de 00 à FF */
    color: red;                     /* KEYWORD */
}
```

```
/* TRANSPARENT */
p {
    color: rgba(255, 0, 0, 0.5);
    color: hsla(0, 100%, 50%, 0.75);
    /* AJOUT CANAL ALPHA, Valeur entre 0 (transparence totale) et 1 (opaque) */
}
```

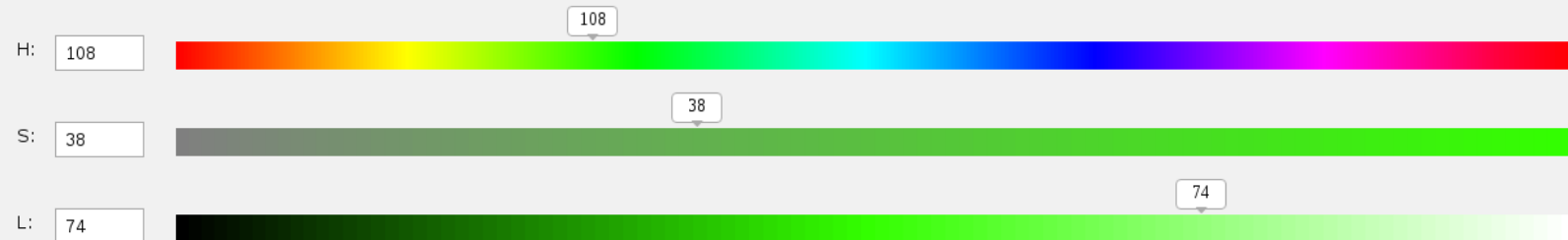
## HSL Calculator



hsl(108, 38%, 74%)

rgb(174, 214, 164)

#aed6a4



[https://www.w3schools.com/colors/colors\\_hsl.asp](https://www.w3schools.com/colors/colors_hsl.asp)

# LES BORDURES

Des bordures peuvent être ajoutées sur nos blocs

On pourra modifier leur épaisseur, leur couleur ou encore leur style (trait, pointillés, double ligne, ...)

On aura également le choix d'arrondir les angles de nos blocs en arrondissant les bordures

```
div {  
  border: 1px solid black;  
  
  border-width: 1px;  
  border-style: solid;  
  border-color: black;  
  
  border-radius: 20px;  
}
```

# LES FONDS

On commence à avoir un peu de couleurs, continuons !

Les fonds peuvent être colorés avec une couleur unie, une image, ou les deux

Une image de fond peut être répétée (pattern), ou placée à une position précise dans le bloc

```
body {  
  background-color: #F9CEFE;  
  background-image: url('http://i.imgur.com/OrsKBja.png');  
  background-posotion: bottom right;  
  background-repeat: no-repeat;  
  
  background: #F9CEFE url('http://i.imgur.com/OrsKBja.png') bottom right no-repeat;  
}
```



# CSS 3

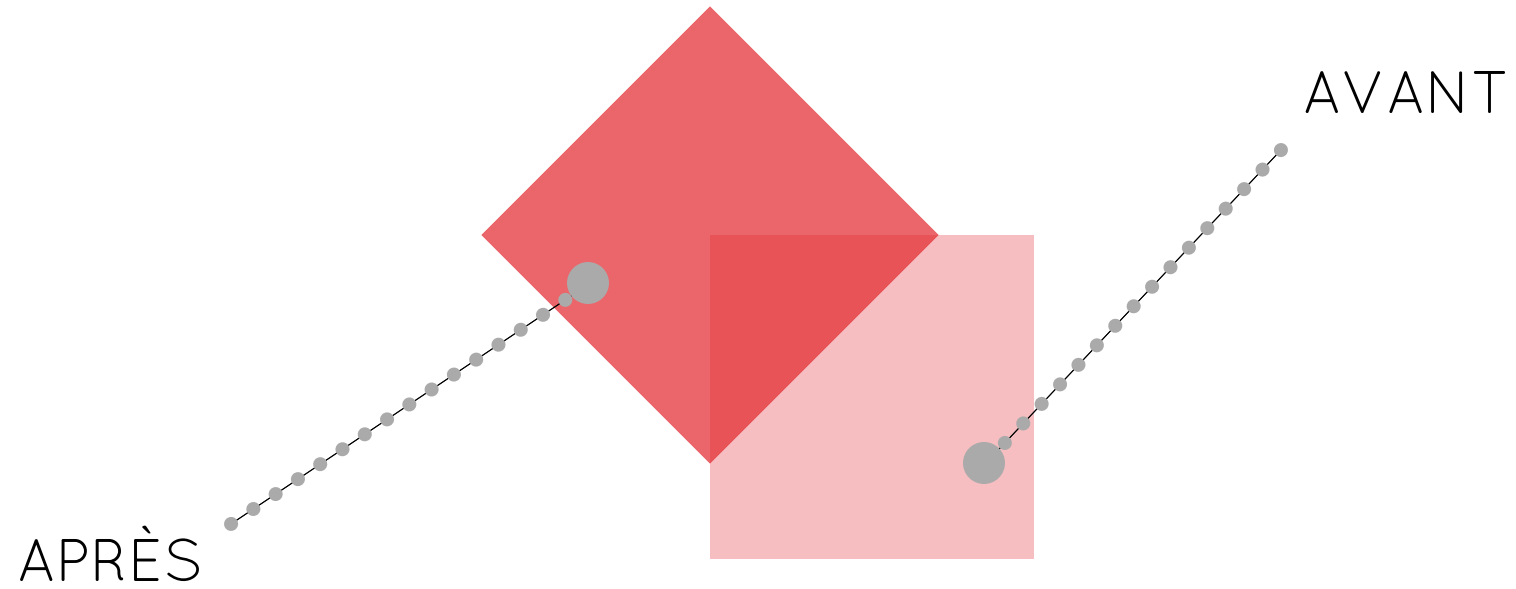
## FONCTIONS AVANCÉES

# TRANSFORMATIONS

Le CSS a énormément évolué au cours des dernières années et permet aujourd'hui de faire des transformations avancés

Ainsi nous allons pouvoir faire des translations, rotations, mise à l'échelle et autres inclinaisons, dans un plan 2D ou 3D uniquement en CSS

# TRANSLATION ET ROTATION



```
<!-- ... -->
```

```
<div class="base after"></div>
```

```
<div class="base before"></div>
```

```
<!-- ... -->
```

```
.base {
```

```
  width: 100px;
```

```
  height: 100px;
```

```
  position: fixed;
```

```
  top: 50%;
```

```
  left: 50%;
```

```
  background-color: rgba(225, 37, 43, 0.3);
```

```
}
```

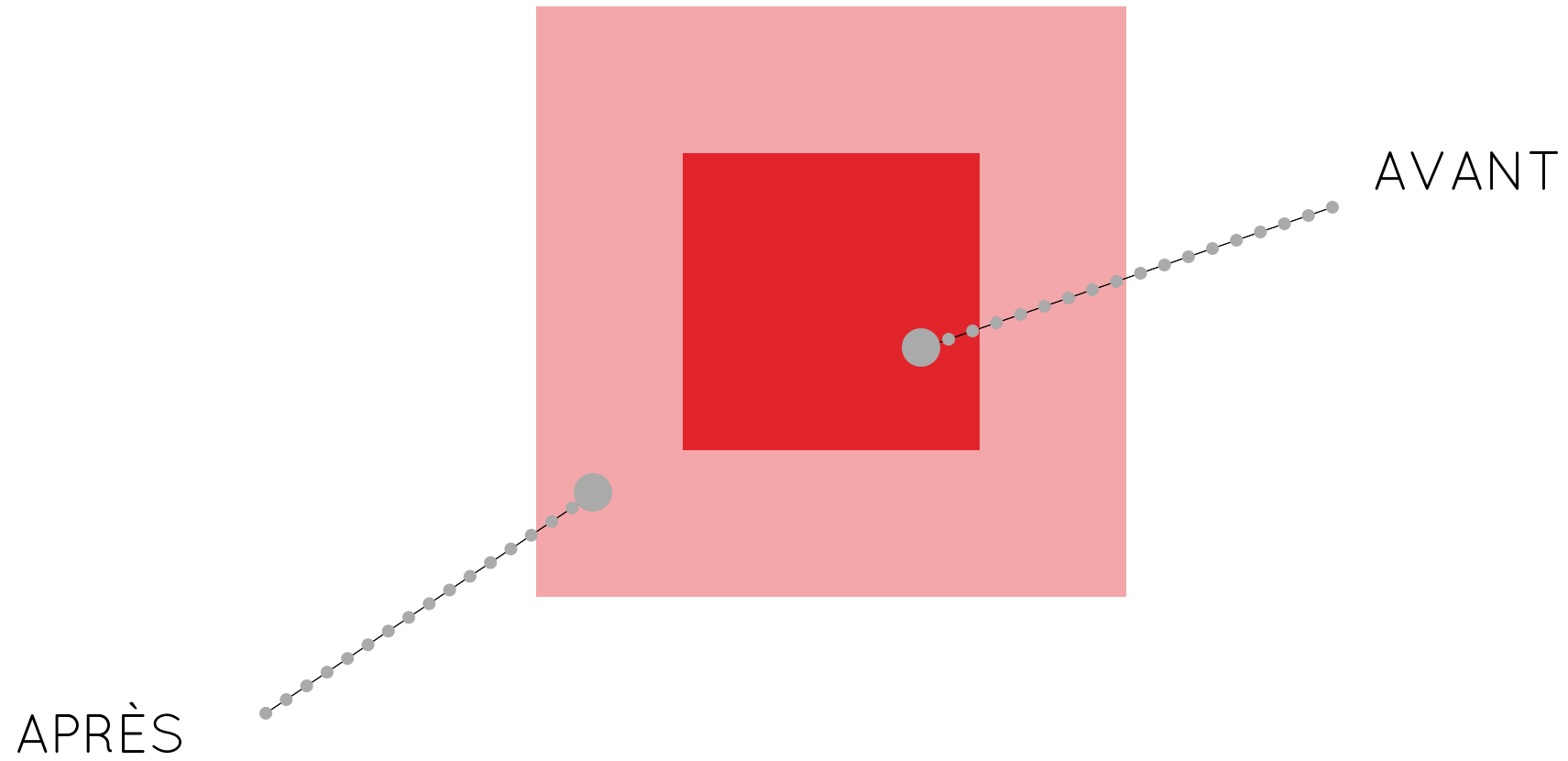
```
.after {
```

```
  background-color: rgba(225, 37, 43, 0.7);
```

```
  transform: translateX(-50%) translateY(-50%) rotateZ(45deg);
```

```
}
```

# ÉCHELLE



```
<!-- ... -->
```

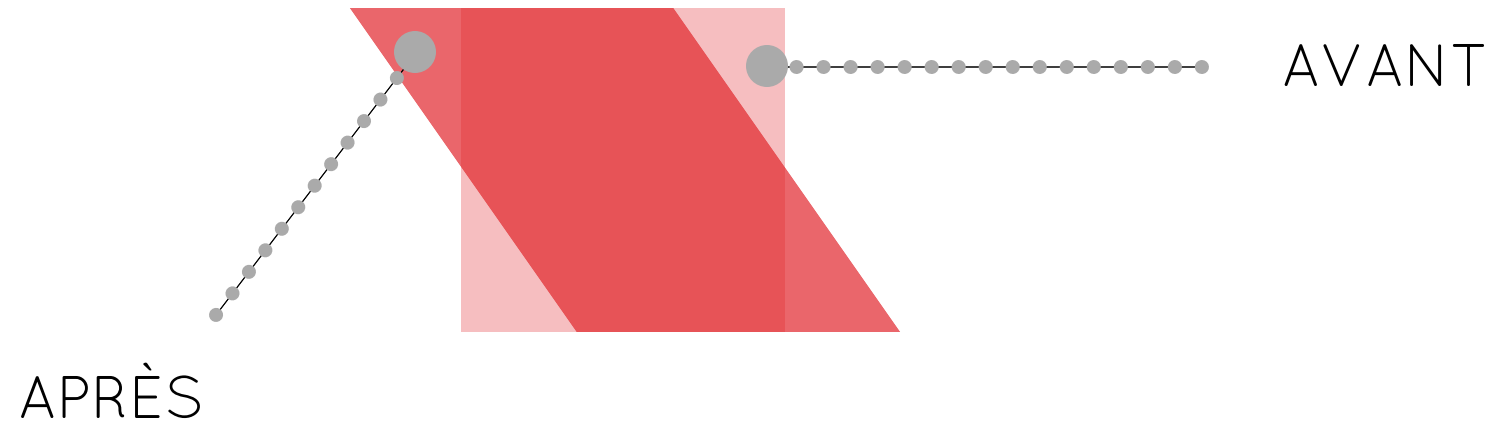
```
<div class="base after"></div>
```

```
<div class="base before"></div>
```

```
<!-- ... -->
```

```
.base {  
  width: 100px;  
  height: 100px;  
  position: fixed;  
  top: 50%;  
  left: 50%;  
  background-color: rgb(225, 37, 43);  
}  
  
.after {  
  background-color: rgba(225, 37, 43, 0.7);  
  transform: scale(3);  
}
```

# INCLINAISON



```
<!-- ... -->
```

```
<div class="base after"></div>
```

```
<div class="base before"></div>
```

```
<!-- ... -->
```

```
.base {  
  width: 100px;  
  height: 100px;  
  position: fixed;  
  top: 50%;  
  left: 50%;  
  background-color: rgba(225, 37, 43, 0.3);  
}
```

```
.after {  
  background-color: rgba(225, 37, 43, 0.7);  
  transform: skew(35deg);  
}
```



# TRANSITIONS

Lorsqu'un élément change de style, au survol par exemple, cela peut paraître brutal ou faire un effet de clignotement

Les transitions peuvent atténuer cet effet

On va favoriser les animations/transition CSS plutôt que de les implémenter en javascript

```
<!-- ... -->

<div class="base"></div>

<!-- ... -->
```

```
.base {
  width: 100px;
  height: 100px;
  position: fixed;
  top: 50%;
  left: 50%;
  background-color: red;
  transition: background-color 250ms ease;
}

.base:hover {
  background-color: yellow;
}
```

ALLEZ PLUS LOIN

# ANIMATIONS

Les animations sont un complément aux transitions, les transitions permettent seulement de passer d'un état A à un état B

Les animations peuvent être une suite de modification, avec plusieurs étapes intermédiaires possibles

Les animations sont de fait, un peu plus complexes

```
<!-- ... -->
<body>

    <div class="animations"></div>

</body>
<!-- ... -->
```

```
@keyframes my_custom_animation {
  0%   { background-color:#892369; width:5%; height:0; }
  40%  { background-color:#892369; height:100%; }
  80%  { width:100%; height:100%; }
  100% { background-color:#56acdf; width:100%; height:100%; }
}

.animations {
  position: fixed;
  top: 50%;
  left: 50%;
  transform: translateX(-50%) translateY(-50%);
  animation-name: my_custom_animation;
  animation-duration: 4s;
  animation-iteration-count: 1;
  animation-direction: normal;
  animation-fill-mode: forwards;
}
```

# PRINCIPALES PROPRIÉTÉS D'ANIMATIONS

@keyframe

animation

animation-delay

animation-direction

animation-duration

animation-duration

animation-fill-mode

animation-iteration-count

...

# MEDIA QUERIES

Adapter son style à la taille de l'écran

Certains sites ne sont pas fait pour les petits écrans

Aujourd'hui, tablettes et téléphones permettent de naviguer sur le web et on peut encore remarquer des sites avec un style uniquement pour les grands écrans

Attention, ne pas confondre sites "adaptatifs" et sites "responsive"

```
<!-- ... -->
```

```
<body>
```

```
  <div class="row">
```

```
    <div class="col-1">
```

```
      Lorem ipsum ...
```

```
    <div class="col-2">
```

```
      Lorem ipsum ...
```

```
    </div>
```

```
  </div>
```

```
</body>
```

```
<!-- ... -->
```

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

```
.col-1 {  
  flex: 2;  
  padding: 20px;  
  background-color: #fbc12d;  
}
```

```
.col-2 {  
  flex: 1;  
  padding: 20px;  
  background-color: #892369;  
  color: #F5F5F5;  
}
```

```
@media only screen and (max-width: 700px) {  
  .row {  
    flex-direction: column;  
  }  
}
```



# LIBRAIRIES DE GRILLES

Faire son propre système de colonnes peut être long et fastidieux, voir incomplet

C'est pourquoi il existe des librairies CSS qui implémentent un système de grilles pour gérer la structures (les colonnes) de notre site

# EXEMPLE AVEC FLEXBOX GRID

<http://flexboxgrid.com/>

```
<div class="row">  
  <div class="col-xs-12 col-sm-8 col-md-6 col-lg-4">  
    Lorem ipsum dolor sit amet, consectetur adipisicing elit.  
    Quo voluptate aspernatur omnis fugit, necessitatibus  
    doloribus. Officia, aliquam distinctio, in voluptatem fugit,  
    adipisci reiciendis sequi quo quaerat iure eius dolores modi.  
  </div>  
</div>
```

# UI FRAMEWORK

Il existe aussi ce qu'on appelle des "UI Frameworks", ils permettent d'avoir une base solide de style CSS, et pas seulement un système de grilles

Très pratique pour démarrer un projet rapidement et très simple à utiliser, ils servent pour un POC (proof of concept), MVP (minimum viable product), interface d'administration, ...

Doit absolument être personnalisé pour un projet client !

Il en existe une multitude, parmi eux : Bootstrap, Foundation, Skeleton, ...

# EXEMPLE AVEC BOOTSTRAP

<http://getbootstrap.com/>

```
<!-- ... -->  
  
<div class="container">  
  <div class="row">  
    <div class="col-xs-12">  
      <button class="btn btn-success">Valider</button>  
    </div>  
  </div>  
</div>  
  
<!-- ... -->
```

# SASS

<http://sass-lang.com/>

SASS permet d'écrire des feuilles de styles avec des outils plus complets

On peut écrire du CSS dans du SASS, mais le SASS apporte de nombreux avantages, parmi eux : les variables, les mixins, les boucles, ...

Il existe d'autres langages comme LESS ou STYLUS

AVEZ VOUS DES  
QUESTIONS ?

# MERCI



Openska