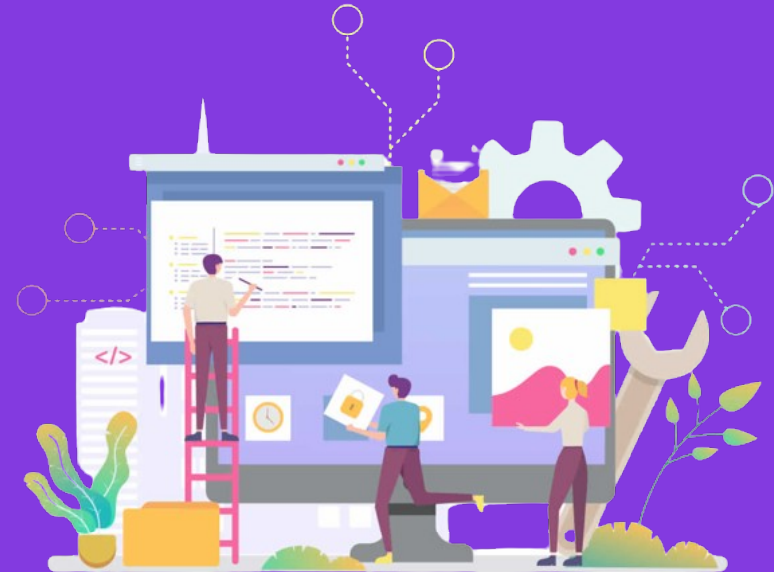


Création de site web CSS3 - Mettre en forme le Web

Enseignant : M. Chakib Ben Jijel



CSS



Année scolaire: 2024-2025

Qu'est-ce que CSS ?

CSS (Cascading Style Sheets) est un langage utilisé pour définir les styles de vos pages Web, y compris la conception, la mise en page et les variations d'affichage pour différents appareils et tailles d'écran.

Les feuilles de style externes sont stockées dans des fichiers CSS.



Où écrit-on le CSS ? (1/3)

Il existe **trois manières** d'insérer du code CSS :

CSS externe

Les styles externes sont définis dans l'élément `<link>`, à l'intérieur de la section `<head>` d'une page HTML.

```
<head>  
<link rel="stylesheet" href="monstyle.css">  
</head>
```

Une **feuille de style externe** peut être écrite dans n'importe quel éditeur de texte et doit être enregistrée avec une extension **.css**.



Où écrit-on le CSS ? (2/3)

CSS interne

Les styles internes sont définis dans la balise `<style>`, à l'intérieur de la section `<head>` d'une page HTML.

Le code CSS interne peut être utilisée si une seule page HTML a un style unique.

```
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 40px;
}
</style>
</head>
```



Où écrit-on le CSS ? (3/3)

CSS en ligne

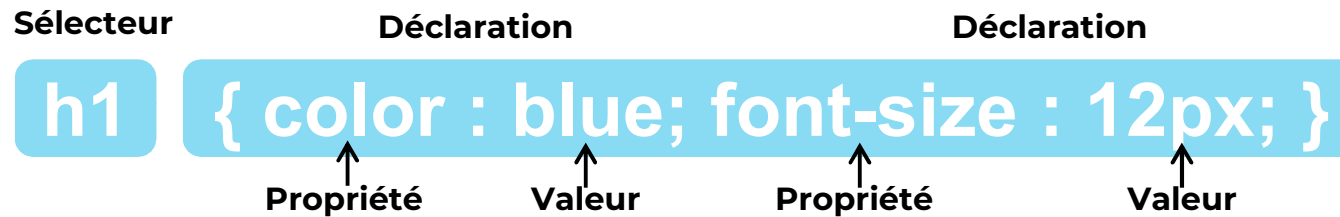
Les styles en ligne sont définis dans l'attribut « style » de l'élément concerné :

```
<h1 style="color:blue;text-align:center;">Ceci est un titre</h1>  
<p style="color:red;">Ceci est un paragraphe.</p>
```

Syntaxe de CSS

CSS est un langage basé sur **des règles**.

Une règle CSS se compose **d'un sélecteur** et d'un **bloc de déclaration**.



Le **sélecteur** pointe vers l'élément HTML que vous souhaitez styliser.

Le **bloc de déclaration** contient une ou plusieurs déclarations séparées par des **points-virgules**.

Chaque déclaration comprend **un nom de propriété CSS** et **une valeur**, séparés par deux points.



Sélecteurs (1/11)

Les sélecteurs CSS sont utilisés pour « trouver » (ou sélectionner) les éléments HTML que vous souhaitez styliser.

Sélecteur element

Le sélecteur d'éléments sélectionne les éléments HTML en fonction du nom de l'élément.

```
/* Ici, tous les éléments <p> de la page seront alignés au centre,  
   avec une couleur de texte rouge  
*/  
p {  
  text-align: center;  
  color: red;  
}
```



Sélecteurs (2/11)

Sélecteur `element[attribut]`

Le sélecteur `element[attribut]` est utilisé pour sélectionner des éléments avec un attribut spécifié.

```
/*L'exemple suivant sélectionne tous les éléments <a> avec un attribut « target » :*/  
a[target] {  
  background-color: yellow;  
}
```

```
<a href="https://edunet.tn">edunet.tn</a>  
<a href="http://www.disney.com" target="_blank">disney.com</a>  
<a href="http://www.wikipedia.org" target="_top">wikipedia.org</a>
```

Les liens avec un attribut "target" obtiennent un fond jaune :

edunet.tn [disney.com](http://www.disney.com) [wikipedia.org](http://www.wikipedia.org)



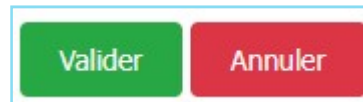
Sélecteurs (3/11)

Sélecteur `element[attribut="value"]`

Le sélecteur `element[attribut="value"]` est utilisé pour sélectionner des éléments avec un attribut et une valeur spécifiés.

```
input[type="submit"] {  
  background-color: #28a745;  
  color: white;  
}  
input[type="reset"] {  
  background-color: #dc3545;  
  color: white;  
}
```

```
<input type="submit" value="Valider">  
<input type="reset" value="Annuler">
```



Sélecteurs (4/11)

Sélecteur `element.class`

Le sélecteur `element.class` sélectionne les éléments spécifiés avec la valeur d'attribut de classe spécifiée.

```
/* sélectionne tous les éléments avec class="center" */  
.center {  
  text-align: center;  
}  
/* sélectionne tous les éléments <p> avec class="large" */  
p.large {  
  font-size: 200%;  
}
```

```
<h2 class="center">Ce titre sera aligné au centre</h2>
```

```
<p>Ce paragraphe ne sera pas affecté.</p>
```

```
<p class="center">Ce paragraphe sera aligné au centre.</p>
```

```
<p class="center large">Ce paragraphe sera aligné au centre et dans une  
grande taille de police.</p>
```

Ce titre sera aligné au centre

Ce paragraphe ne sera pas affecté.

Ce paragraphe sera aligné au centre.

Ce paragraphe sera aligné au centre et dans une
grande taille de police.



Sélecteurs (5/11)

Sélecteur #id

Le sélecteur **#id** utilise l'**attribut id** d'un élément HTML pour sélectionner un élément spécifique.

L'identifiant d'un élément est unique au sein d'une page, le sélecteur d'identifiant est donc utilisé pour sélectionner un élément unique !

Pour sélectionner un élément avec un identifiant spécifique, écrire **un caractère dièse (#)**, suivi de **l'identifiant de l'élément**.

```
/* La règle CSS ci-dessous sera appliquée à l'élément HTML avec id="para1" */  
#para1 {  
  text-align: center;  
  color: red;  
}
```

```
<p id="para1">Bonjour le monde!</p>  
<p>Ce paragraphe n'est pas affecté par le style.</p>
```

Remarque : Un nom d'identification ne peut pas commencer par un chiffre !



Sélecteurs (6/11)

Sélecteur `.class`

Le sélecteur de classe `.class` sélectionne les éléments HTML avec un attribut de classe spécifique.

Pour sélectionner des éléments avec une classe spécifique, écrire **un point (.)**, suivi du **nom de la classe**.

```
/* Dans cet exemple, tous les éléments HTML avec class="c1" seront rouges et alignés au centre */  
.c1 {  
  text-align: center;  
  color: red;  
}
```

```
<h1 class="c1">Titre rouge et aligné au centre</h1>  
<p class="c1">Paragraphe rouge et aligné au centre.</p>
```

Remarque : Un nom de classe ne peut pas commencer par un chiffre !



Sélecteurs (7/11)

Sélecteur de regroupement

Le sélecteur de regroupement sélectionne tous les éléments HTML avec **les mêmes définitions** de style.

Pour regrouper des sélecteurs, séparer chaque sélecteur par **une virgule**.

```
/* Les éléments h1, h2 et p ont les mêmes
   définitions de style */
h1 {
  text-align: center;
  color: red;
}

h2 {
  text-align: center;
  color: red;
}

p {
  text-align: center;
  color: red;
}
```

```
/* Dans cet exemple, nous avons regroupé
   les sélecteurs du code ci-contre */
h1, h2, p {
  text-align: center;
  color: red;
}
```



Sélecteurs (8/11)

Sélecteur universel (*)

Le sélecteur universel (*) sélectionne tous les éléments HTML de la page.

```
/* La règle CSS ci-dessous affectera chaque élément HTML de la page */  
* {  
  text-align: center;  
  color: blue;  
}
```

Sélecteurs (9/11)

Sélecteurs de pseudo-classe

Une **pseudo-classe** permet de définir **l'état particulier** d'un élément.

Par exemple, elle permet de :

- Appliquer un style à un élément **lorsqu'un utilisateur passe la souris dessus** ;
- Appliquer un style différent aux **liens visités** et **non visités** ;
- Appliquer un style à un élément lorsqu'il **obtient le focus**.



Sélecteurs (10/11)

Un sélecteur de pseudo-classe sélectionne les éléments en fonction **d'un certain état**.

Les liens peuvent être affichés de différentes manières :

```
/* Lien non visité */  
a:link {  
    color: red;  
}  
  
/* Lien visité */  
a:visited {  
    color: green;  
}  
  
/* Passez la souris sur le lien */  
a:hover {  
    color: hotpink;  
}  
  
/* Lien sélectionné */  
a:active {  
    color: blue;  
}
```



Sélecteurs (11/11)

La pseudo-classe CSS **:hover** est utilisée pour sélectionner des éléments lorsque vous **passez la souris dessus**.

Un exemple d'utilisation de la pseudo-classe **:hover** sur un élément **<div>** :

```
div:hover {  
  background-color: blue;  
}
```

La pseudo-classe CSS **:focus** est utilisée pour sélectionner et styliser l'élément qui **reçoit le focus**.

Un exemple d'utilisation de la pseudo-classe **:focus** sur **un champ de saisie** (input) :

```
input:focus {  
  background-color: yellow;  
}
```



Mise en forme du texte (1/8)

CSS possède de nombreuses propriétés pour formater le texte.

Couleur du texte

La propriété `color` est utilisée pour définir la couleur du texte. La couleur est spécifiée par :

- Un nom de couleur - comme "rouge"
- Une valeur HEX - comme "#ff0000"
- Une valeur RGB - comme "rgb(255,0,0)"

```
body {  
  color: blue;  
}  
  
h1 {  
  color: green;  
}
```



Mise en forme du texte (2/8)

Police du texte

La propriété `font-family` est utilisée pour spécifier la police d'un texte.

```
.p1 {  
  font-family: "Times New Roman", Times, serif;  
}  
  
.p2 {  
  font-family: Arial, Helvetica, sans-serif;  
}  
  
.p3 {  
  font-family: "Lucida Console", "Courier New", monospace;  
}
```

Il s'agit d'un paragraphe affiché dans la police Times New Roman.

Il s'agit d'un paragraphe affiché dans la police Arial.

Il s'agit d'un paragraphe affiché dans la police Lucida Console.



Mise en forme du texte (3/8)

Style de la police

La propriété `font-style` est principalement utilisée pour spécifier du texte en italique.

```
p.italic {  
  font-style: italic;  
}
```

C'est un paragraphe en style italique.

Poids de la police

La propriété `font-weight` spécifie le poids (`bold`, `bolder` ou `lighter`) d'une police.

```
.epais {  
  font-weight: bold;  
}  
.plus_epais {  
  font-weight: bolder;  
}  
.plus_legers {  
  font-weight: lighter;  
}
```

Un paragraphe avec des caractères épais.

Un paragraphe avec des caractères plus épais.

Un paragraphe avec des caractères plus légers.



Mise en forme du texte (4/8)

Taille de la police

La propriété `font-size` définit la taille du texte.

Définir la taille du texte **en pixels** vous donne un contrôle total sur la taille du texte.

```
h1 {  
  font-size: 40px;  
}  
  
h2 {  
  font-size: 30px;  
}  
  
p {  
  font-size: 14px;  
}
```



Mise en forme du texte (5/8)

Raccourci de police

La propriété `font` est utilisée pour **définir plusieurs propriétés de police dans une seule déclaration**.

La propriété `font` est une propriété abrégée pour :

- `font-style`
- `font-weight`
- `font-size`
- `font-family`

```
p.a {  
  font: 20px Arial, sans-serif;  
}  
  
p.b {  
  font: italic small-caps bold 12px/30px Georgia, serif;  
}
```



Mise en forme du texte (6/8)

Alignement du texte

La propriété `text-align` est utilisée pour définir l'alignement horizontal d'un texte.

Un texte peut être aligné **à gauche**, **à droite**, **centré** ou **justifié**.

```
h1 {  
  text-align: center;  
}
```

Titre 1 (centré)

```
h2 {  
  text-align: left;  
}
```

Titre 2 (à gauche)

```
h3 {  
  text-align: right;  
}
```

Titre 3 (à droite)

```
p {  
  text-align: justify;  
}
```

Paragraphe (justifié) : Dans mes années plus jeunes et plus vulnérables, mon père m'a donné des conseils que je pense depuis lors. « Chaque fois que vous avez envie de critiquer quelqu'un, m'a-t-il dit, rappelez-vous simplement que tous les gens de ce monde n'ont pas eu les avantages que vous avez eus. »



Mise en forme du texte (7/8)

Transformation de texte

La propriété `text-transform` est utilisée pour spécifier les lettres majuscules et minuscules dans un texte.

Il peut être utilisé pour tout transformer en **lettres majuscules** ou **minuscules**, ou **mettre en majuscule la première lettre de chaque mot** :

```
p.uppercase {  
  text-transform: uppercase;  
}
```

CE TEXTE EST TRANSFORMÉ EN MAJUSCULE.

```
p.lowercase {  
  text-transform: lowercase;  
}
```

ce texte est transformé en minuscule.

```
p.capitalize {  
  text-transform: capitalize;  
}
```

Dans Ce Texte, CSS Met En Majuscule La Première Lettre De Chaque Mot.



Mise en forme du texte (8/8)

Ombre de texte

La propriété `text-shadow` ajoute une ombre au texte.

Dans son utilisation la plus simple, vous spécifiez uniquement **l'ombre horizontale** (2px) et **l'ombre verticale** (2px) :

```
h1 {  
  text-shadow: 2px 2px;  
}
```

Effet d'ombre de texte !



Couleurs (1/7)

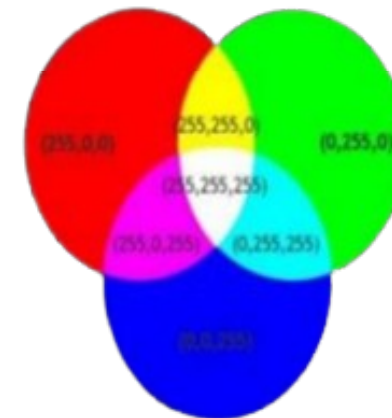
Les couleurs sont spécifiées à l'aide de **noms de couleurs prédéfinis** ou de **valeurs RGB, HEX, ...**

lime	
green	
yellow	
olive	
blue	
navy	
fuchsia	
purple	
aqua	
teal	

Nom des couleurs



Notation hexadécimale



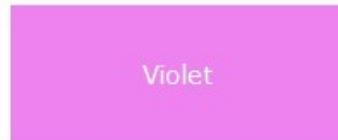
Notation RGB



Couleurs (2/7)

Noms de couleurs CSS

En CSS, une couleur peut être spécifiée en utilisant un nom de couleur prédéfini.



Couleurs (3/7)



Couleurs RGB

Une valeur de couleur **RGB** représente les sources de lumière ROUGE (Red), VERTE(Green) et BLEU(Blue).

En CSS, une couleur peut être spécifiée comme valeur RGB, en utilisant cette formule RGB:

rgb(red, green, blue)

Chaque paramètre (red, green et blue) définit l'intensité de la couleur **entre 0 et 255**.

Exemples :

- **rgb(255, 0, 0)** s'affiche en rouge, car le **rouge** est défini sur sa valeur la plus élevée (255) et les autres sont définis sur 0.
- Pour afficher le **noir**, définir tous les paramètres de couleur sur 0, comme ceci : **rgb(0, 0, 0)**.
- Pour afficher le **blanc**, définir tous les paramètres de couleur sur 255, comme ceci : **rgb(255, 255, 255)**.



Couleurs (4/7)

Couleurs HEX

Une couleur hexadécimale est spécifiée avec : **#RRGGBB**, où les entiers hexadécimaux **RR** (rouge), **GG** (vert) et **BB** (bleu) spécifient les composantes de la couleur.

En CSS, une couleur peut être spécifiée à l'aide d'une valeur hexadécimale sous la forme :

#rrggbb

Où **rr** (rouge), **gg** (vert) et **bb** (bleu) sont des **valeurs hexadécimales** comprises **entre 00 et ff** (identiques au nombre décimal 0-255).



Couleurs (5/7)

Couleur de l'arrière plan

La propriété `background-color` est utilisée pour définir la couleur d'arrière-plan des éléments HTML.

```
.box {  
  background-color: #567895;  
}
```

```
h2 {  
  background-color: Tomato;  
}
```

```
span {  
  background-color: rgb(255,255,0);  
}
```

```
<div class="box">  
  <h2>Bonjour le monde!</h2>  
  <p>Try changing the background essayez de changer <span>les  
couleurs</span> d'arrière-plan.</p>  
</div>
```

Bonjour le monde!

Essayez de changer les couleurs d'arrière-plan.



Couleurs (6/7)

Couleur du texte

La propriété `color` est utilisée pour définir la couleur du texte.

```
h3 {  
  color: Violet;  
}  
#para1 {  
  color: DodgerBlue;  
}  
#para2 {  
  color: MediumSeaGreen;  
}
```

```
<h3>Bonjour le monde</h3>
```

```
<p id="para1">Lorem ipsum dolor sit amet, consectetur adipiscing  
elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna  
aliquam erat volutpat.</p>
```

```
<p id="para2">Ut wisi enim ad minim veniam, quis nostrud exerci tation  
ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo  
consequat.</p>
```

Bonjour le monde

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.



Couleurs (7/7)

Couleur de la bordure

Vous pouvez définir la couleur des bordures.

```
#titre1 {  
  border: 2px solid Tomato;  
}  
  
#titre2 {  
  border: 2px solid DodgerBlue;  
}  
#titre3 {  
  border: 2px solid Violet;  
}  
  
<h1 id="titre1">Bonjour le monde!</h1>  
  
<h1 id="titre2">Bonjour le monde!</h1>  
  
<h1 id="titre3">Bonjour le monde!</h1>
```

Bonjour le monde!

Bonjour le monde!

Bonjour le monde!



Arrière-plans (1/5)

Couleur de l'arrière plan

La propriété `background-color` spécifie la couleur d'arrière-plan d'un élément.

Avec CSS, une couleur est le plus souvent spécifiée par :

- Un **nom de couleur** valide - comme "red"

```
/*Ici, les éléments <h1>, <p> et <div> auront des couleurs d'arrière-plan différentes*/
```

```
h1 {  
  background-color: green;  
}  
  
div {  
  background-color: lightblue;  
}  
  
p {  
  background-color: yellow;  
}
```

Exemple de couleur d'arrière-plan CSS !

Il s'agit d'un texte à l'intérieur d'un élément div.

Ce paragraphe a sa propre couleur de fond.

Nous sommes toujours dans l'élément div.



Arrière-plans (2/5)

Image d'arrière plan

La propriété `background-image` spécifie une image à utiliser comme arrière-plan d'un élément.

```
body {  
  background-image: url("paper.gif");  
}
```



Arrière-plans (3/5)

Répétition de l'image d'arrière-plan

La propriété `background-repeat` définit la manière dont une image d'arrière-plan sera répétée.

Par défaut, une image d'arrière-plan est répétée verticalement et horizontalement.

```
/*Ici, l'image d'arrière-plan est répétée verticalement et horizontalement. C'est le réglage par défaut.*/  
body {  
    background-image: url("paper.gif");  
    background-repeat: repeat;  
}  
/*Ici, l'image d'arrière-plan est répétée uniquement horizontalement.*/  
body {  
    background-image: url("paper.gif");  
    background-repeat: repeat-x;  
}  
/*Ici, l'image d'arrière-plan est répétée uniquement verticalement.*/  
body {  
    background-image: url("paper.gif");  
    background-repeat: repeat-y;  
}
```



Arrière-plans (4/5)

Taille de l'arrière-plan

La propriété **background-size** spécifie la taille des images d'arrière-plan.

```
/*Ici, l'image d'arrière-plan est définie sur 300px de largeur et 100px de hauteur.*/  
#example1 {  
  background: url(mountain.jpg);  
  background-size: 300px 100px;  
}
```

```
/*La taille de l'image d'arrière-plan est définie en pourcentage de l'élément parent ;  
75% de largeur et 50% de hauteur.*/  
#example2 {  
  background: url(mountain.jpg);  
  background-size: 75% 50%;  
}
```



Arrière-plans (5/5)

Raccourci d'arrière-plan

La propriété **background** est utilisée pour **définir plusieurs propriétés de d'arrière-plan dans une seule déclaration**.

La propriété **background** est une propriété abrégée pour :

- **background-color**
- **background-image**
- **background-size**
- **background-repeat**

```
body {  
    background: #ffffff url("img_tree.gif") 300px 100px repeat-x;  
}
```



Bordures (1/6)

Les propriétés de bordure CSS vous permettent de spécifier **le style**, **la largeur** et **la couleur** de la bordure d'un élément.

J'ai des frontières de tous les côtés.

J'ai une bordure inférieure rouge.

J'ai des bordures arrondies.

J'ai une bordure bleue à gauche.



Bordures (2/6)

Style de bordure

La propriété `border-style` définit le `style` des quatre bordures d'un élément. Cette propriété peut avoir de 1 à 4 valeurs.

```
p.dotted {border-style: dotted;}  
p.dashed {border-style: dashed;}  
p.solid {border-style: solid;}  
p.double {border-style: double;}  
p.none {border-style: none;}  
p.hidden {border-style: hidden;}  
p.mix {border-style: dotted dashed solid double;}
```

Une bordure en pointillés.

Une bordure constituée de tirets.

Une bordure solide.

Une double bordure.

Pas de bordure.

Une bordure cachée.

Une bordure mixte.



Bordures (3/6)

Largeur de la bordure

La propriété `border-width` spécifie la largeur des quatre bordures.

La largeur peut être définie comme une taille spécifique (**en px**) ou en utilisant l'une des trois valeurs prédéfinies : **thin, medium ou thick** :

```
p.un {
  border-style: solid;
  border-width: 5px;
}

p.deux {
  border-style: solid;
  border-width: medium;
}

p.trois {
  border-style: dotted;
  border-width: 2px;
}

p.quatre {
  border-style: dotted;
  border-width: thick;
}
```

```
/*La propriété border-width peut avoir de une à quatre valeurs (pour la bordure supérieure,
la bordure droite, la bordure inférieure et la bordure gauche)*/
p.un {
  border-style: solid;
  border-width: 5px 20px; /* 5px en haut et en bas, 20px sur les côtés */
}
p.deux {
  border-style: solid;
  border-width: 20px 5px; /* 20px en haut et en bas, 5px sur les côtés */
}
p.trois {
  border-style: solid;
  border-width: 25px 10px 4px 35px; /* 5px en haut, 10px à droite, 4px en bas et 35px à gauche */
}
```

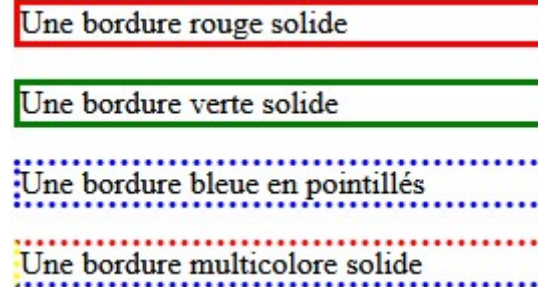


Bordures (4/6)

Couleur de la bordure

La propriété `border-color` est utilisée pour définir la couleur des quatre

```
p.un {  
  border-style: solid;  
  border-color: red;  
}  
p.deux {  
  border-style: solid;  
  border-color: green;  
}  
p.trois {  
  border-style: dotted;  
  border-color: blue;  
}  
p.quatre {  
  border-style: dotted;  
  /* rouge en haut, vert à droite, bleu en bas et jaune à gauche */  
  border-color: red green blue yellow;  
}
```



Bordures (5/6)

Bordures arrondies

La propriété `border-radius` est utilisée pour ajouter des bordures arrondies à un élément :

```
p.normale {  
  border: 2px solid red;  
}  
p.ronde1 {  
  border: 2px solid red;  
  border-radius: 5px;  
}  
p.ronde2 {  
  border: 2px solid red;  
  border-radius: 8px;  
}  
p.ronde3 {  
  border: 2px solid red;  
  border-radius: 12px;  
}
```

Bordure normale

Bordure ronde

Bordure ronde

Bordure la plus ronde



Bordures (6/6)

Raccourci de bordure

La propriété **border** est utilisée pour **définir plusieurs propriétés de bordure dans une seule déclaration**.

La propriété **border** est une propriété abrégée pour :

- **border-width**
- **border-style (obligatoire)**
- **border-color**

```
p {  
  border: 5px solid red;  
}
```

Cette propriété est une propriété abrégée pour border-width, border-style et border-color.



Listes (1/5)

En HTML, il existe deux principaux types de listes :

- **Listes non ordonnées ()** - les éléments de la liste sont marqués par des puces.
- **Listes ordonnées ()** - les éléments de la liste sont marqués par des chiffres ou des lettres .

Les propriétés de la liste CSS vous permettent de :

- Définir différents marqueurs d'éléments de liste pour les listes ordonnées;
- Définir différents marqueurs d'éléments de liste pour les listes non ordonnées;
- Définir une image comme marqueur d'élément de liste;
- Ajouter des couleurs d'arrière-plan aux listes et aux éléments de liste.



Listes (2/5)

Différents marqueurs d'éléments de liste

La propriété `list-style-type` spécifie le type de marqueur d'élément de liste.

```
ul.a {  
  list-style-type: circle;  
}  
  
ul.b {  
  list-style-type: square;  
}  
  
ol.c {  
  list-style-type: upper-roman;  
}  
  
ol.d {  
  list-style-type: lower-alpha;  
}
```

Exemple de listes non ordonnées :

- Café
- Thé
- Coca Cola

- Café
- Thé
- Coca Cola

Exemple de listes ordonnées :

- I. Café
- II. Thé
- III. Coca Cola

- a. Café
- b. Thé
- c. Coca Cola



Listes (3/5)

Une image comme marqueur d'élément de liste

La propriété `list-style-image` spécifie une image comme marqueur d'élément de liste :

```
ul {  
  list-style-image: url('sqpurple.gif');  
}
```

- Café
- Thé
- Coca Cola



Listes (4/5)

Positionner les marqueurs d'éléments de liste

La propriété `list-style-position` spécifie la position (`outside` ou `inside`) des marqueurs d'éléments de liste (puces).

```
ul.a {  
  list-style-position: outside;  
}
```

```
ul.b {  
  list-style-position: inside;  
}
```

`list-style-position` : à l'extérieur (par défaut) :

- Café
- Thé
- Coca Cola

position de style de liste : à l'intérieur :

- Café
- Thé
- Coca Cola



Listes (5/5)

Raccourci de liste

La propriété `list-style` est une propriété raccourcie. Il est utilisé pour définir **toutes les propriétés** de la liste **dans une seule déclaration**.

La propriété `list-style` est une propriété abrégée pour :

- `list-style-type`
- `list-style-position`
- `list-style-image`

```
ul {  
  list-style: square inside url("sqpurple.gif");  
}
```

- Café
- Thé
- Coca Cola



Tableaux (1/2)

Bordures de tableau

Pour spécifier les bordures d'un tableau en CSS, utilisez la propriété **border**.

```
table, th, td {  
  border: 1px solid;  
}
```

Prénom	Nom
Pierre	Griffin
Loïs	Griffin

La propriété **border-collapse** définit si les bordures du tableau doivent être réduites en une seule bordure (**separate** ou **collapse**) :

```
table {  
  width: 100%;  
  border-collapse: collapse;  
}
```

Prénom	Nom
Pierre	Griffin
Loïs	Griffin

```
table {  
  width: 100%;  
  border-collapse: separate;  
}
```

Prénom	Nom
Pierre	Griffin
Loïs	Griffin



Tableaux (2/2)

Propriété `table-layout`

La propriété `table-layout` définit la façon utilisée pour disposer les cellules, les lignes et les colonnes du tableau.

```
table.a {  
  table-layout: auto;  
  width: 180px;  
}  
  
table.b {  
  table-layout: fixed;  
  width: 180px;  
}
```



Images

Propriété filter

La propriété **filter** définit les effets visuels (comme le **flou** et la **saturation**) sur un élément (souvent ****).

```
/*Appliquer un effet de flou à l'image*/  
img {  
  filter: blur(5px);  
}  
|  
/* Convertir l'image en niveaux de gris*/  
img {  
  filter: grayscale(100%);  
}  
|  
/* Inverser les échantillons dans l'image */  
img {  
  filter: invert(100%);  
}
```



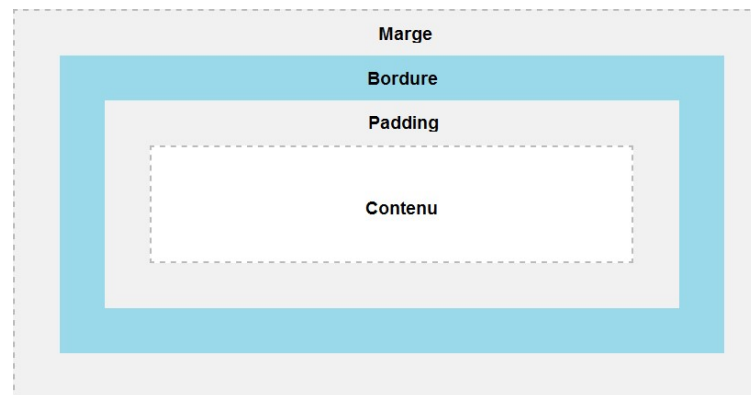
Modèle de boîte

En CSS, le terme « **modèle de boîte** » est utilisé pour parler de conception et de mise en page.

Le modèle de boîte CSS est essentiellement une boîte qui entoure chaque élément HTML.

Il se compose de :

- **UN contenu**
- **Un « Padding »**
- **Une bordure**
- **Une marge**



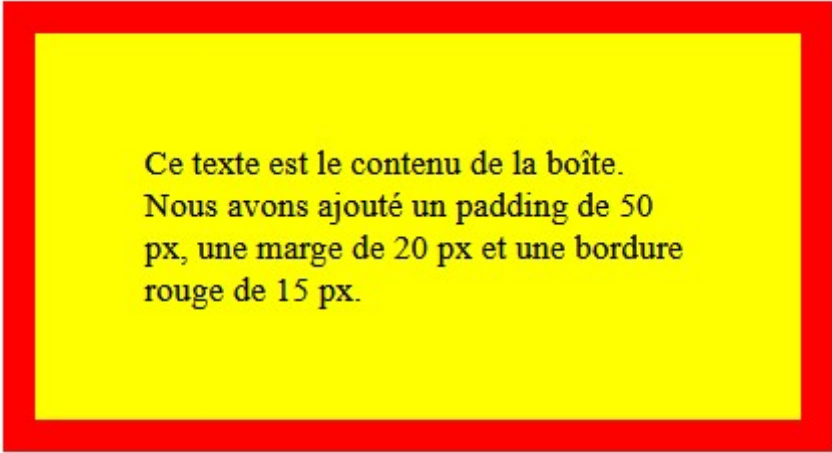
L'image ci-contre illustre le modèle de boîte :



Modèle de boîte

Démonstration du modèle boîte

```
div {  
  background-color: yellow;  
  width: 250px;  
  border: 15px solid red;  
  padding: 50px;  
  margin: 20px;  
}
```



Ce texte est le contenu de la boîte.
Nous avons ajouté un padding de 50
px, une marge de 20 px et une bordure
rouge de 15 px.



Modèle de boîte

Hauteur, largeur

Les propriétés CSS `height` et `width` sont utilisées pour définir la hauteur et la largeur d'un élément.

Les propriétés `height` et `width` peuvent avoir des valeurs en `px` ou en `pourcentage %`.

```
div {  
  height: 80px;  
  width: 50%;  
  background-color: yellow;  
}
```

```
<div>Cet élément div a une hauteur de 80px et une largeur de 50 %.</div>
```

Cet élément div a une hauteur de 80px et une largeur de 50 %.



Modèle de boîte

Marges CSS

La propriété CSS **margin** est utilisée pour créer de l'espace autour d'un élément, en dehors de toutes les bordures définies.

```
/*La propriété margin avec quatre valeurs */
div {
  margin: 25px 50px 75px 100px;
}
```

Cet élément div a une marge supérieure de 25 px, une marge droite de 50 px, une marge inférieure de 75 px et une marge gauche de 100 px.

```
/*La propriété margin avec trois valeurs*/
div {
  margin: 25px 50px 75px;
}
```

Cet élément div a une marge supérieure de 25 px, des marges droite et gauche de 50 px et une marge inférieure de 75 px.

```
/*La propriété margin avec deux valeurs*/
div {
  margin: 25px 50px;
}
```

Cet élément div possède des marges supérieure et inférieure de 25 px, et des marges droite et gauche de 50 px.

```
/*La propriété margin avec une seule valeur*/
div {
  margin: 25px;
}
```

Cet élément div possède des marges supérieure, inférieure, gauche et droite de 25 px.



Modèle de boîte

CSS Padding

La propriété CSS **padding** est utilisée pour générer de l'espace autour du contenu d'un élément, **à l'intérieur** de toutes les bordures définies.

```
/*La propriété padding avec quatre valeurs */
div {
  padding: 25px 50px 75px 100px;
}

/*La propriété padding avec trois valeurs */
div {
  padding: 25px 50px 75px;
}

/*La propriété padding avec deux valeurs */
div {
  padding: 25px 50px;
}

/*La propriété padding avec une valeur */
div {
  padding: 25px;
}
```



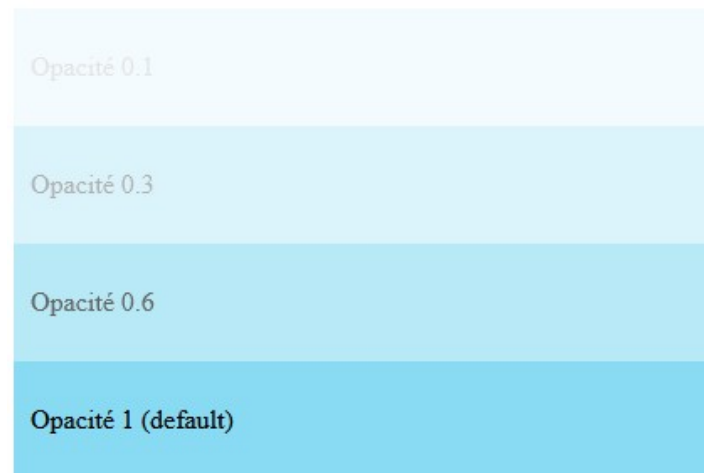
Modèle de boîte

Opacité / Transparence

La propriété `opacity` spécifie l'opacité/transparence d'un élément.

La propriété `opacity` peut prendre une valeur **comprise entre 0,0 et 1**. Plus la valeur est faible, plus elle est transparente :

```
div.premier {  
  opacity: 0.1;  
}  
  
div.deuxieme {  
  opacity: 0.3;  
}  
  
div.troisieme {  
  opacity: 0.6;  
}
```



Modèle de boîte

Ombre de la boîte

La propriété CSS `box-shadow` est utilisée pour appliquer une ou plusieurs ombres à la boîte d'un élément.

```
#exemple1 {  
  box-shadow: 5px 10px;  
}  
  
#exemple2 {  
  box-shadow: 5px 10px #888888;  
}  
  
#exemple3 {  
  box-shadow: 5px 10px red;  
}
```

Un élément div avec une ombre. La première valeur est le décalage horizontal et la deuxième valeur est le décalage vertical. La couleur de l'ombre sera héritée de la couleur du texte.

Vous pouvez également définir la couleur de l'ombre. Ici, la couleur de l'ombre est grise.

Une ombre rouge.



L'affichage (1/3)

La propriété **display** est utilisée pour spécifier comment un élément est affiché sur une page Web.

Chaque élément HTML a une valeur d'affichage **par défaut**, en fonction du type d'élément dont il s'agit. La valeur d'affichage par défaut pour la plupart des éléments est **block** ou **inline**.

La propriété **display** est utilisée pour modifier le comportement d'affichage par défaut des éléments HTML.



L'affichage (2/3)

La propriété **display** a plusieurs valeurs :

Valeur	Description
inline	Affiche un élément en tant qu'élément en ligne
block	Affiche un élément en tant qu'élément de bloc
inline-block	Affiche un élément en tant que conteneur de blocs de niveau en ligne. L'élément lui-même est formaté comme un élément en ligne, mais vous pouvez appliquer des valeurs de hauteur et de largeur .



L'affichage (3/3)

```
span.a {  
  display: inline; /* la valeur par défaut pour span */  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid blue;  
  background-color: yellow;  
}  
span.b {  
  display: inline-block;  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid blue;  
  background-color: yellow;  
}  
span.c {  
  display: block;  
  width: 100px;  
  height: 100px;  
  padding: 5px;  
  border: 1px solid blue;  
  background-color: yellow;  
}
```

display: inline

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. Aliquam venenatis gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.

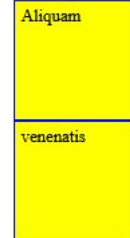
display: inline-block

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat. Aliquam venenatis gravida nisl sit

amet facilisis. Nullam cursus fermentum velit sed laoreet.

display: block

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum consequat scelerisque elit sit amet consequat. Aliquam erat volutpat.



gravida nisl sit amet facilisis. Nullam cursus fermentum velit sed laoreet.



Le positionnement (1/6)

La propriété **position** spécifie le type de méthode de positionnement utilisée pour un élément.

Il existe cinq valeurs de position différentes :

- **position : static ;**
- **position : relative ;**
- **position : fixed ;**
- **position : absolute ;**
- **position : sticky ;**

Les propriétés **top**, **right**, **bottom** et **left** déterminent l'emplacement final de l'élément positionné.



Le positionnement (2/6)

Toutes les propriétés de positionnement CSS

Propriété	Description
position	Spécifie le type de positionnement d'un élément
top	Définit le bord de la marge supérieure d'une boîte positionnée
right	Définit le bord de la marge droite pour une boîte positionnée
bottom	Définit le bord de la marge inférieure d'une boîte positionnée
left	Définit le bord de la marge gauche pour une boîte positionnée



Le positionnement (3/6)

Positionnement statique

Les éléments HTML sont positionnés statiquement **par défaut**.

Les éléments positionnés statiquement ne sont pas affectés par les propriétés **top**, **bottom**, **left** et **right**.

Un élément avec **position : static ;** n'est pas positionné d'une manière particulière ; il est toujours positionné **selon le déroulement normal de la page**.

Le positionnement (4/6)

Positionnement relatif

Un élément avec **position : relative** ; est positionné par rapport à sa position normale.

La définition des propriétés **top**, **right**, **bottom** et **left** d'un élément positionné relativement entraînera son ajustement par **rapport à sa position normale**.

Les autres contenus ne seront pas ajustés pour s'adapter aux espaces laissés par l'élément.



Le positionnement (5/6)

Positionnement absolu

Un élément avec **position : absolue ;** est positionné **par rapport à l'ancêtre positionné** le plus proche (au lieu d'être positionné par rapport à la fenêtre, comme fixe).

Positionnement fixe

Un élément avec **position : fixe ;** est positionné **par rapport à la fenêtre**, ce qui signifie qu'il reste toujours au **même endroit même si la page défile**. Les propriétés **top**, **right**, **bottom** et **left** sont utilisées pour positionner l'élément.



Le positionnement (6/6)

Positionnement adhérent (sticky)

Un élément avec `position : sticky ;` est positionné en fonction de la position de défilement de l'utilisateur.

Un élément `sticky` bascule entre relatif et fixe, en fonction de la position de défilement. Il est positionné par rapport jusqu'à ce qu'une position de décalage donnée soit atteinte dans la fenêtre - puis il "colle" en place (comme `position:fixed`).



CSS float (1/3)

La propriété **float** spécifie si un élément doit flotter vers la **gauche (left)**, la **droite (right)** ou **pas du tout (none)**.

La propriété **float** a les valeurs suivantes :

- **none** : L'élément **ne flotte pas** (il s'affiche uniquement à l'endroit où il apparaît dans le texte). C'est la **valeur par défaut**.
- **left** : L'élément flotte à gauche de son conteneur.
- **right** : L'élément flotte à droite de son conteneur.



CSS float (2/3)

Float Right

```
img {  
  float: right;  
}
```

L'ananas (*Ananas comosus*) est une espèce de plantes xérophytes, originaire d'Amérique du Sud, plus spécifiquement du Paraguay, du nord-est de l'Argentine et sud du Brésil. Il est connu principalement pour son fruit comestible, qui est en réalité un fruit multiple. Le mot ananas vient du tupi-guarani naná naná, qui signifie « parfum des parfums ». Le terme ananas est également un nom vernaculaire qui peut désigner plusieurs autres espèces de Bromeliaceae, aux fruits pas à peu comestibles comme l'espèce *Glomeropitcairnia penduliflora* désignée par les termes « ananas sauvage », « ananas bois » et « zananas mawon »..



Float left

```
img {  
  float: left;  
}
```

L'ananas (*Ananas comosus*) est une espèce de plantes xérophytes, originaire d'Amérique du Sud, plus spécifiquement du Paraguay, du nord-est de l'Argentine et sud du Brésil. Il est connu principalement pour son fruit comestible, qui est en réalité un fruit multiple. Le mot ananas vient du tupi-guarani naná naná, qui signifie « parfum des parfums ». Le terme ananas est également un nom vernaculaire qui peut désigner plusieurs autres espèces de Bromeliaceae, aux fruits pas à peu comestibles comme l'espèce *Glomeropitcairnia penduliflora* désignée par les termes « ananas sauvage », « ananas bois » et « zananas mawon ».



CSS float (3/3)

Float none (par défaut)

```
img {  
  float: none;  
}
```



L'ananas (*Ananas comosus*) est une espèce de plantes xérophytes, originaire d'Amérique du Sud, plus spécifiquement du Paraguay, du nord-est de l'Argentine et sud du Brésil. Il est connu principalement pour son fruit comestible, qui est en réalité un fruit multiple. Le mot ananas vient du tupi-guarani naná naná, qui signifie « parfum des parfums ». Le terme ananas est également un nom vernaculaire qui peut désigner plusieurs autres espèces de Bromeliaceae, aux fruits pas à peu comestibles comme l'espèce *Glomeropitcairnia penduliflora* désignée par les termes « ananas sauvage », « ananas bois » et « zananas mawon ».

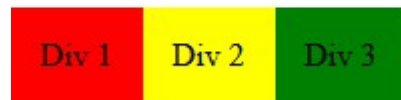
```
div {  
  float: left;  
  padding: 15px;  
}
```

```
.div1 {  
  background: red;  
}
```

```
.div2 {  
  background: yellow;  
}
```

```
.div3 {  
  background: green;  
}
```

Exemple : Trois divs flottent l'un à côté de l'autre



Débordement (1/3)

La propriété **overflow** spécifie ce qui doit se produire si le contenu déborde de la boîte d'un élément.

La propriété **overflow** a les valeurs suivantes :

- **visible** : le débordement **n'est pas tronqué**. Il s'affiche en dehors du cadre de l'élément. C'est la **valeur par défaut**.
- **hidden** : Le débordement est tronqué et le reste du contenu sera invisible.
- **clip** : Le débordement est tronqué et le reste du contenu est invisible. **Interdit le défilement, y compris par programmation**.
- **scroll** : Le débordement est tronqué et **une barre de défilement** est ajoutée pour voir le reste du contenu.
- **auto** : Similaire au scroll, mais il ajoute des barres de défilement uniquement **lorsque cela est nécessaire**.



Débordement (2/3)

```
div {  
  background-color: #88DBF2;  
  width: 200px;  
  height: 65px;  
  border: 1px solid;  
  overflow: visible;  
}
```

```
div {  
  background-color: #88DBF2;  
  width: 200px;  
  height: 65px;  
  border: 1px solid black;  
  overflow: hidden;  
}
```

```
div {  
  background-color: #88DBF2;  
  width: 200px;  
  height: 65px;  
  border: 1px solid;  
  overflow: clip;  
}
```

Overflow: visible (par défaut)

Vous pouvez utiliser la propriété overflow pour mieux contrôler la mise en page. Elle spécifie ce qui se passe si le contenu déborde de la boîte d'un élément.

Overflow: hidden

Vous pouvez utiliser la propriété overflow pour mieux contrôler la mise en page. Elle spécifie ce qui se passe si le

Overflow: clip

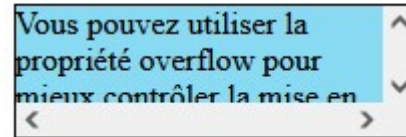
Vous pouvez utiliser la propriété overflow pour mieux contrôler la mise en page. Elle spécifie ce qui se passe si le



Débordement (3/3)

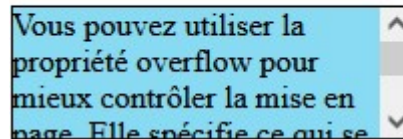
```
div {  
  background-color: #88DBF2;  
  width: 200px;  
  height: 65px;  
  border: 1px solid;  
  overflow: scroll;  
}
```

Overflow: scroll



```
div {  
  background-color: #88DBF2;  
  width: 200px;  
  height: 65px;  
  border: 1px solid;  
  overflow: auto;  
}
```

Overflow: auto



Transformation (1/5)

La propriété **transform** applique une **transformation 2D ou 3D** à un élément. Cette propriété vous permet de faire **pivoter**, **redimensionner**, **déplacer**, **incliner**, etc. des éléments.

Avec la propriété CSS **transform**, vous pouvez utiliser les méthodes de **transformation 2D** suivantes :

- **translate()**
- **rotate()**
- **scale()**
- **skew()**

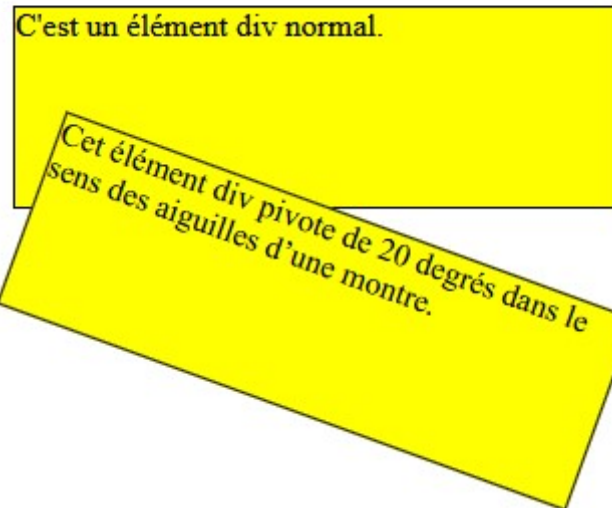


Transformation (2/5)

La méthode rotate()

La méthode `rotate()` fait pivoter un élément dans le sens des aiguilles d'une montre ou dans le sens inverse selon un degré donné.

```
div#myDiv {  
  transform: rotate(20deg);  
}
```



Transformation (3/5)

La méthode skew()

La méthode `skew()` incline un élément le long des axes X et Y selon les angles donnés.

```
div#myDiv {  
  transform: skew(20deg,10deg);  
}
```

C'est un élément div normal.

Cet élément div est incliné de 20 degrés le long de l'axe X et de 10 degrés le long de l'axe Y.



Transformation (4/5)

La méthode `scale()`

La méthode `scale()` **augmente** ou **diminue la taille** d'un élément (selon les paramètres donnés pour la largeur et la hauteur).

```
div {  
  transform: scale(2,3);  
}
```

Cet élément div fait deux fois sa largeur d'origine et trois fois sa hauteur d'origine.

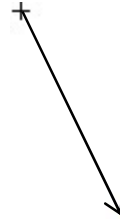


Transformation (5/5)

La méthode `translate()`

La méthode **`translate()`** déplace un élément de sa position actuelle (selon les paramètres donnés pour l'axe X et l'axe Y).

```
div {  
  transform: translate(50px,100px);  
}
```



Cet élément div est déplacé de 50 pixels vers la droite et de 100 pixels vers le bas par rapport à sa position actuelle.



Transition (1/4)

Les **transitions** CSS vous permettent de modifier les valeurs des propriétés en douceur, sur une durée donnée.

Comment utiliser les transitions CSS ?

Pour créer un effet de transition, vous devez spécifier :

- la **propriété CSS** à laquelle vous souhaitez ajouter un effet.
- la **durée** de l'effet.



Transition (2/4)

Les propriétés de transition CSS

Propriété	Description
transition-property	Spécifie le nom de la propriété CSS à laquelle l'effet de transition sera appliqué.
transition-duration	Spécifie combien de secondes ou de millisecondes un effet de transition prend pour se terminer.
transition-delay	Spécifie un délai (en secondes) pour l'effet de transition.
transition	Une propriété raccourcie pour définir les quatre propriétés de transition en une seule ligne.



Transition (3/4)

Les propriétés de transition CSS peuvent être spécifiées **une par une**, comme ceci :

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition-property: width;  
  transition-duration : 2s;  
  transition-delay : 1s  
}  
  
div:hover {  
  width: 300px;  
}
```

L'exemple précédent montre un élément **<div>** rouge de 100px x 100px. Cet élément **<div>** a également spécifié **un effet de transition** pour la **propriété** « **width** », d'une **durée de 2 secondes** et un **délai de 1 seconde**.

Une nouvelle valeur (300px) pour la propriété **width** est spécifié **lorsqu'un utilisateur passe la souris sur l'élément <div>** :



Transition (4/4)

ou en utilisant la propriété raccourcie **transition** :

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s 1s;  
}  
  
div:hover {  
  width: 300px;  
}
```

Modifier plusieurs valeurs de propriétés

L'exemple suivant ajoute un **effet de transition** pour les propriétés **width** et **height**, avec une **durée** de 2 secondes pour la largeur et de 4 secondes pour la

```
hauteur: div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: width 2s, height 4s;  
}  
  
div:hover {  
  width: 300px;  
  height: 300px;  
}
```



Transition + Transformation

L'exemple suivant ajoute **un effet de transition à la transformation** :

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
  transition: transform 2s;  
}  
  
div:hover {  
  transform: rotate(180deg);  
}
```



Animation (1/6)

Une **animation** permet à un élément de passer progressivement d'un style à un autre.

Vous pouvez modifier autant de propriétés CSS que vous le souhaitez, autant de fois que vous le souhaitez.

Pour utiliser une animation CSS, vous devez d'abord spécifier des images clés (**keyframes**).

Les images clés (**keyframes**) définissent les styles que l'élément adoptera à certains moments.



Animation (2/6)

La règle @keyframes

La règle CSS `@keyframes` est utilisée pour contrôler les étapes d'une séquence d'animation en définissant des styles CSS pour les points le long de la séquence d'animation.

```
/* Le code d'animation */
@keyframes exemple {
  from {background-color: red;}
  to {background-color: yellow;}
}

/* L'élément auquel appliquer l'animation */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: exemple;
  animation-duration: 4s;
}
```



Animation (3/6)

Les propriétés d'animation CSS

Propriété	Description
animation-name	Spécifie un nom pour l'animation @keyframes.
animation-delay	Spécifie un délai pour le démarrage d'une animation.
animation-duration	Définit la durée pendant laquelle une animation doit prendre pour terminer un cycle.
animation-iteration-count	Spécifie le nombre de fois qu'une animation doit être jouée.



Animation (4/6)

Les propriétés d'animation CSS

Propriété	Description
<code>animation-direction</code>	Définit si une animation doit être jouée vers l'avant (<code>normal</code>), vers l'arrière (<code>reverse</code>) ou dans des cycles alternés (<code>alternate</code>).
<code>animation</code>	Une propriété raccourcie pour définir les cinq propriétés d'animation en une seule ligne.



Animation (5/6)

L'exemple suivant modifiera la couleur d'arrière-plan de l'élément **<div>** lorsque l'animation est terminée à 25 %, à 50 % et à nouveau lorsque l'animation est terminée à 100 % :

```
/* Le code d'animation */
@keyframes example {
  0%   {background-color: red;}
  25%  {background-color: yellow;}
  50%  {background-color: blue;}
  100% {background-color: green;}
}

/* L'élément auquel appliquer l'animation */
div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
}
```



Animation (6/6)

L'exemple ci-dessous utilise **cinq des propriétés** d'animation :

```
div {  
  animation-name: exemple;  
  animation-duration: 5s;  
  animation-delay: 2s;  
  animation-iteration-count: infinite;  
  animation-direction: alternate;  
}
```

Le même effet d'animation que ci-dessus peut être obtenu en utilisant la propriété raccourcie **animation** :

```
div {  
  animation: exemple 5s 2s infinite alternate;  
}
```

