

Chapitre 2

HTML5 & CSS3

Techniques de programmation Web

Enseignante:

Hela LIMAM

Année universitaire
2021-2022

Introduction

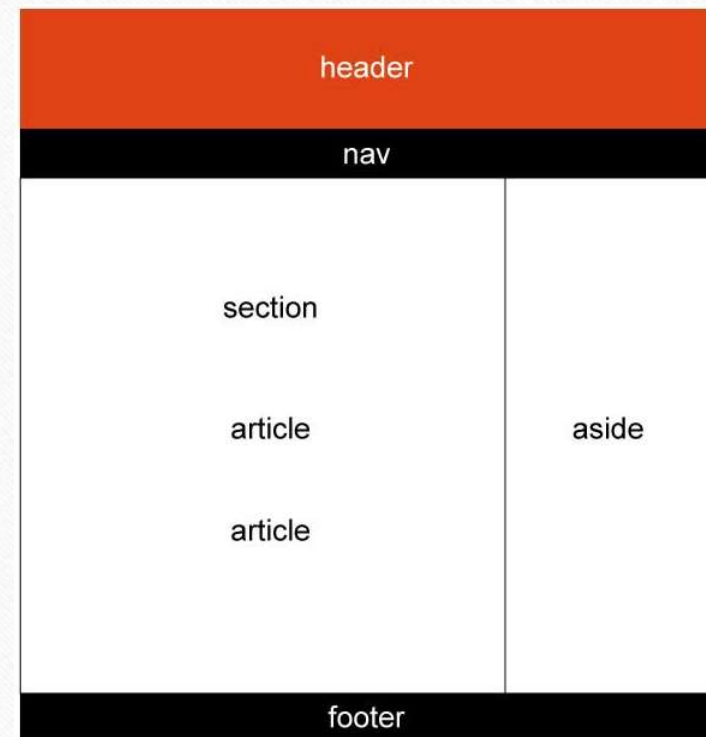
- **HTML5** (HyperText Markup Language 5) est la dernière révision majeure d'HTML
- La dernière version a été finalisée le 28 octobre 2014 par W3C.
- HTML5 comprend également une couche d'application avec de nombreuses API, ainsi qu'un algorithme afin de pouvoir traiter les documents à la syntaxe non conforme.
- W3C met à disposition un outil de validation des documents écrits en [HTML5](#) et en [CSS](#)

Les nouveautés

- La déclaration DOCTYPE pour HTML5 est très simple: **<!DOCTYPE html>**
- La définition de l'encodage est de même plus simple: **<meta charset="UTF-8">**
- De éléments sémantiques comme **<header>**, **<footer>**, **<article>** et **<section>**
- Nouveaux attributs des éléments de formulaire tels que **nombre**, **date**, **heure**, **time**...
- API HTML 5 : HTML 5 permet d'utiliser de nouvelles API qui permettront de créer de nouvelles applications et qui peuvent être utilisées ensemble avec de nouveaux éléments introduits pour les applications dont :
 - ✓ une API de dessin 2D utilisé avec la nouvelle balise **canvas**
 - ✓ une API pour jouer des vidéos et des sons/musiques utilisé avec les nouvelles balises **video** et **audio**

Balises sémantiques

Nom	Détails
<section>	Section générique regroupant un même sujet, une même fonctionnalité.
<article>	Articles contenus dans les sections.
<nav>	Cette balise est utilisée pour afficher les liens de navigation et avant tout pour indiquer qu'il y a un menu.
<aside>	Cette balise est utilisée pour afficher des compléments d'informations, ou des contenus pas forcément en relation avec la page, par exemple dans le cas où vous affichez une publicité ou encore un menu de navigation secondaire.
<header>	En-tête de page
<footer>	Pied de page



Balise audio

- Pour lire un fichier audio , on utilise l'élément <audio>

<audio controls>

<source src="horse.ogg" type="audio/ogg">

<source src="horse.mp3" type="audio/mpeg">

Le navigateur ne supporte aucun format audio

</audio>

- Avec **controls** , on ajoute les boutons « Lecture », « Pause » , « volume» et la barre de défilement.
- **<source>** permet de spécifier d'autres fichiers audio parmi lesquels le navigateur peut choisir. Le navigateur utilisera le premier format reconnu.

Balise video

- Pour lire un fichier video, on utilise l'élément <video>

```
<video width="320" height="240" poster="sintel.jpg" autoplay controls >
```

```
<source src="movie.mp4" type="video/mp4">
```

```
<source src="movie.ogg" type="video/ogg">
```

```
Your browser does not support the video tag.
```

```
</video>
```

- Avec **autoplay** la vidéo sera jouée dès le chargement de la page.
- Avec **controls** on ajoute les boutons « Lecture », « Pause » et la barre de défilement.

Les feuilles de style CSS

- Cascading Style Sheet
- Préconisée pour les styles des documents HTML depuis la version 4.0
- Le principe consiste à regrouper dans un même document des caractéristiques de mise en forme associées à des groupes d'éléments
- Définir par un nom un ensemble de définitions et de caractéristiques de mise en forme et de l'appeler pour l'appliquer à un texte
- Norme du W3C depuis 1996
- Supportées par les versions récentes des navigateurs
- Utilise son propre langage différent de HTML

Avantages

- Compenser les manques du langage HTML en ce qui concerne la mise en page et la présentation;
- Simplifie l'édition des pages Web et la maintenance du site en séparant le contenu et la présentation;
- Une uniformité de présentation pour l'ensemble des pages d'un site;
- Alléger les fichiers HTML dont le contenu se perdait dans un balisage complexe;
- Diminuer le temps de chargement;
- Permettre le changement de l'aspect d'un site entier par la seule modification de quelques lignes.

Emplacement des styles

Dans un fichier à part(.CSS)(feuille de styles externe)

- ✓ Allège le document HTML
- ✓ Permet de partager une même feuille de style entre plusieurs documents HTML
- ✓ Créer un lien vers la feuille de style au moyen de l'élément<LINK> dans l'entête du document HTML

Dans l'entête du document, au moyen de l'élément <style>(feuille de style interne)

Feuille de style externe

- `<link rel="stylesheet" href= " feuille .css" type="text/css">`
- L'élément LINK doit comprendre:
 - ✓ Le type de lien, soit vers une feuille de styles
 - ✓ Pour une feuille CSS, ce doit être `rel= "stylesheet"`
 - ✓ L'emplacement de la feuille: variable qui doit coïncider avec le nom du fichier et l'emplacement de la feuille `href= "feuille .css"`
 - ✓ Le type de feuille de style: Pour une feuille CSS, ce doit être `type="text/css"`

Syntaxe

- Une feuille de style est une séquence de règles
- Une règle peut s'appliquer à plusieurs éléments ou même (dans quelques cas) à une partie d'éléments seulement
- Une règle est composée d'un sélecteur de balises et d'une déclaration
- Sélecteur: Fonction qui limite la portée de la déclaration, généralement un élément HTML.
- Une déclaration est mise entre accolades et se compose d'une propriété et d'une valeur , séparées par :



Syntaxe

- Déclaration: Fonction qui associe une caractéristique de stylage(propriété) à une valeur spécifique(valeur)

```
p {text-align:left}  
BODY {background-color:red}
```

- On peut associer plusieurs couples propriété-valeur dans une même déclaration en les séparant par un ;

```
P {color:blue;text-align:justify}
```


Syntaxe

- On peut définir des sélecteurs multiples pour une même déclaration en les séparant par ,

```
H1,H3 {font-weight:bold}
```

- On peut définir plus d'une règle pour un même sélecteur:

```
H1,H2,H3 {color: red}  
H1 {font-size:200%}
```

Syntaxe

- Commentaire dans une feuille de style:
 - ❖ Précédé de /* et suivi de */
 - ❖ Utile pour désactiver temporairement une partie d'une feuille en développement
- Les majuscules et minuscules sont équivalentes dans tout le stylage CSS
- Unités de mesure

Valeurs relatives em :la largeur de la lettre m dans la police de l'élément, px :pixels, en relation avec la résolution

Valeurs absolues cm :centimètres, mm :millimètre, pt :points typographiques

Pourcentages Se calcule le plus souvent en relation avec la taille de la police ou l'élément parent

Les sélecteurs

sélecteur	description
p{}	Sélectionner tous les paragraphes
h1,P	Sélectionner tous les <h1> et tous les <P>
.bloc	Sélectionner tous les balises ayant l'attribut class = bloc
#bloc	Sélectionner tous les balises ayant l'attribut id = bloc
*{}	Sélectionne toutes les balises sans exception
h1 H3{}	Sélectionne toutes les balises <H3> situées à l'intérieur d'une balise <h1>
h3+p	Sélectionne la première balise <p> située après un titre <h3>
a[title]	Sélectionne tous les liens<a> qui possède un attribut title
a[title="Cliquer ici"]	Idem, mais l'attribut doit avoir exactement pour valeur "Cliquer ici"

Les sélecteurs

sélecteur	description
a[title*="ici"]	Idem, mais l'attribut doit avoir pour valeur "ici" , peu importe sa position
ul > li	permettra de cibler tous les éléments qui sont des éléments enfants directs de .
Autres types de sélecteurs (à voir dans la partie suivante)	

Propriétés pour les polices

- font-family

- ✓ fonction:détermine le nom de la police
- ✓ Valeurs possibles:nom d'une police spécifique ou nom générique(famille de polices)
- ✓ Noms génériques:serif(Times),sans-serif(Arial,Helvetica),cursive,monospace.
- ✓ On peut juxtaposer jusqu'à trois polices comme valeur, ainsi si l'agent utilisateur ne possède pas la première il passe à la suivante

- font-style

- ✓ Fonction:détermine le style de caractères
- ✓ valeurs possibles:normal |italic|oblique
- ✓ valeur par défaut:normal

Propriétés pour les polices

- font-weight

- ✓ Fonction: détermine l'épaisseur du trait des caractères
- ✓ Valeurs possibles: normal|bold|bolder|lighter|100|200|300|400|500|600|700|800|900
- ✓ Valeur par défaut: normal

```
H3 {font-weight: bold}
```


Propriétés pour le texte

- text-align

- ✓ Fonction:détermine l'alignement du texte
- ✓ Valeurs possibles:left|right|center|justify

```
H1,H2{text-align:center}
```

- text-decoration

- ✓ Fonction:détermine la décoration du texte
- ✓ Valeurs possibles:none|underline|overline

```
a:link:{text-decoration:none}
```

Propriétés pour le texte

- Transparence opacity: 0,7;
- Retrait de la première ligne d'un texte text-indent: 50px; //propriété est utilisée pour spécifier le retrait de la première ligne d'un texte
- La casse text-transform: capitalise (1 lettre de chaque mot en majuscule) / uppercase / lowercase / none;

Propriétés pour la couleur

■ color

- ✓ Fonction: détermine la couleur d'un élément en l'occurrence de la police, si elle s'applique à un élément à contenu textuel.
- ✓ Valeurs possibles: couleurs.

```
a:link {color:black}
```

```
a:visited {color:black}
```

```
a:active {color:mediumblue}
```

Propriétés pour l'arrière plan

- background-color:

- ✓ Fonction:détermine la couleur de l'arrière plan d'un élément
- ✓ Valeurs possibles:[couleur]

```
BODY {background-color:gray}
```

- Background-image

- ✓ Fonction:détermine l'image d'arrière-plan
- ✓ Valeurs possibles:[url]

```
BODY {background-image:url}
```


Spécifications bloc ou paragraphe (1)

- Le modèle de boîte: s'applique entièrement aux éléments bloc (div, titre, élément de liste)
- **Bordure standard** **border : style color size**
 - border-style: none / dotted (pointillé) / solid (trait plein) / double;
 - border-color: couleur
 - border-width: / thin / medium / thick / px;
 - border-top-color:yellow;
- **Coins arrondis**
 - border-radius: 10px;
 - border-radius: 10px/20px; (%en horizontale et en verticale)

Spécifications bloc ou paragraphe (2)

■ Marges

✓ Marges intérieures : padding-top/padding-right/padding-bottom/padding-left

❖ padding: 10px; / padding: 10px ; / padding: 10px 20px;

✓ Marge extérieures : margin-top/margin-right/margin-bottom/margin-left

❖ (idem)

■ **Ombrage box-shadow:** Le mot inset est facultatif : il fait entrer l'ombrage à l'intérieur du bloc ou même un texte, horiz et vert représentent le décalage de l'ombrage, le dégradé adoucie la couleur; il est normale s'il est égale au décalage, faible si inférieur, fort si supérieur,

✓ Pour un bloc box-shadow: 5px 5px 5px black inset;

❖ Pour un texte text-shadow: 3px 3px 3px orange;

Propriétés pour les listes

- List-style-type

- ✓ Fonction:détermine le type de puces ou de numérotation pour les éléments d'une liste
- ✓ Valeurs possibles:disc|circle|square|decimal|lower-roman|upper-roman|lower-alpha|upper-alpha|none

```
OL{list-style-type:lower-alpha}  
UL{list-style-type:square}
```

Propriétés pour l'espacement

- Margin ou margin-top, margin-bottom, margin-left, margin-right
 - ✓ Fonction: détermine la largeur de la marge par rapport au bloc conteneur
 - ✓ Valeurs possibles: [longueur] [pourcentage]

```
H1 {margin-top:1em;margin-bottom:2em}  
BODY {margin:2em}
```


Propriétés pour les bordures

- border-color
 - ✓ Fonction:détermine la couleur de la bordure
 - ✓ Valeurs possibles:[couleur]
- border-width
 - ✓ Fonction:détermine l'épaisseur de la bordure
 - ✓ Valeurs possibles:thin|medium|[longueur]

```
IMG {border-color:black; border-width:thin}
```

Les pseudos classes

- Une pseudo-classe est utilisée pour définir un état spécifique d'un élément.
- Exemples
 - ✓ `a:link { color: #FF0000;}`
 - ✓ `a:visited { color: #00FF00;}`
 - ✓ `a:hover { color: #FF00FF;}`
 - ✓ `a:active { color: #0000FF;}`
 - ✓ `td:first-child { color: blue;}`
 - ✓ `td:empty { background: #ff0000;}`

Les pseudo éléments


- Un pseudo-élément CSS est utilisé pour le style spécifié parties d'un élément.
- Exemple
 - ✓ `p::first-line` : La première ligne de chaque élément `<p>`.
 - ✓ `p::first-letter` : première lettre
 - ✓ `a:after/before { content: "→"; }` : Ajoute une flèche après les liens
 - ✓ `::selection { color: red; background: yellow; }` : le texte sélectionné sera rouge sur fond jaune

La propriété display

- La propriété display spécifie si / comment un élément est affiché.
- Un élément de niveau block commence toujours sur une nouvelle ligne et occupe toute la largeur disponible
- Un élément en ligne ne commence pas sur une nouvelle ligne et ne prend que la largeur nécessaire.
- Par rapport à display: inline, la principale différence est que display: inline-block permet de définir une largeur et une hauteur sur l'élément, de plus, les marges / marges supérieures et inférieures sont respectées, mais avec display: inline, elles ne le sont pas.
- Par rapport à display: block, la principale différence est que display: inline-block n'ajoute pas de saut de ligne après l'élément, de sorte que l'élément peut être placé à côté d'autres éléments.
- L'exemple suivant montre les différents comportements d'affichage: inline, display: inline-block et display: block

display: inline-block

```
.nav {  
background-color: yellow; list-style-type: none; text-align: center; border: 1px solid; }  
.nav li {  
display: inline-block; font-size: 20px; padding: 20px; }  
<ul class="nav">  
  <li><a href="#home">Home</a></li>  
  <li><a href="#about">About Us</a></li>  
  <li><a href="#clients">Our Clients</a></li>  
  <li><a href="#contact">Contact Us</a></li>  
</ul>
```



[Home](#) [About Us](#) [Our Clients](#) [Contact Us](#)

Alignement des blocs

- Pour centrer horizontalement un élément de bloc , utilisez `margin: auto;`
- Pour centrer une image, définissez les marges gauche et droite sur `auto` et faites-en un élément de bloc.
- Une méthode d'alignement d'éléments consiste à utiliser `position: absolute`.
- Une autre méthode pour aligner des éléments consiste à utiliser la propriété `float`.
- Si un élément est plus grand que l'élément qui le contient et qu'il est flottant, il débordera à l'extérieur de son conteneur. Vous pouvez utiliser '**clearfix**' pour résoudre ce problème.

✓ Exemple de **clearfix**

Positionnement

- La propriété **position** spécifie le type de méthode de positionnement utilisé pour un élément (statique, relatif, fixe, absolu).
 - ✓ Position static: l'élément HTML est positionné statique par défaut.
 - ✓ Position relative : l'élément est positionné par rapport à sa position normale.
 - ✓ Position fixe: l'élément est positionné par rapport à la fenêtre d'affichage, ce qui signifie qu'il reste toujours au même endroit même si la page est défilée.
 - ✓ Position absolue: l'élément est positionné par rapport à l'ancêtre positionné le plus proche
- [Exemple1](#) [Exemple2](#)

Les animations

- Pour créer une animation CSS, il faut utiliser la propriété `animation` sur un ou plusieurs éléments. Cette propriété permet de configurer la durée, le minutage et d'autres détails à propos de l'animation.
- Pour déterminer l'apparence visuelle de l'animation on applique `@keyframes`
- Les propriétés détaillées rattachées à la propriété `animation` sont :
 1. **animation-delay**: Cette propriété spécifie un délai pour le début d'une animation
 - ❖ Exemple: **animation-delay: 2s;** (un délai de 2 secondes avant de démarrer l'animation)
 2. **animation-direction**: Cette propriété indique si l'animation doit alterner entre deux directions de progressions
 - ❖ **Normal**: L'animation est lue normalement (en avant). C'est par défaut
 - ❖ **Reverse**: L'animation est lue en sens inverse (en arrière)
 - ❖ **Alternate**: L'animation est jouée en avant puis en arrière

Animation -2-

3. **animation-duration**: Cette propriété définit la durée d'un cycle de l'animation, Si la animation-duration propriété n'est pas spécifiée, aucune animation ne se produira car la valeur par défaut est 0s.
4. **animation-name**: Cette propriété permet de déclarer un nom qui pourra être utilisé comme référence à l'animation pour la règle @keyframes.
5. **animation-iteration-count**: Cette propriété détermine le nombre de fois que l'animation est répétée.
 - Exemple: **animation-iteration-count: 3;** (l'animation 3 fois avant son arrêt)
 - Exemple: **animation-iteration-count: infinite;** (l'animation continue à jamais)

Les animation -3-

6. animation-timing-function: propriété spécifie la courbe de vitesse de l'animation, elle peut avoir les propriétés suivantes:

- ❖ **Ease:** Spécifie une animation avec un début lent, puis rapide, puis se termine lentement (par défaut)
- ❖ **Linear:** Spécifie une animation avec la même vitesse du début à la fin
- ❖ **ease-in:** Spécifie une animation avec un démarrage lent
- ❖ **ease-out:** Spécifie une animation avec une fin lente
- ❖ **ease-in-out:** Spécifie une animation avec un début et une fin lents

Les animations -4-

- L'exemple [suivant](#) lie l'animation **example** à l'élément <div>.
- L'animation durera 4 secondes et changera progressivement la couleur de fond de l'élément <div> de "rouge" à "jaune":

```
<style>
div {
  width: 500px;
  height: 100px;
  background-color: gray;
  animation-name: example;
  animation-duration: 4s;
  animation-delay: 2s;
  animation-iteration-count: infinite;
  animation-direction: reverse;
}
/* Standard syntax */
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}
</style>
<div></div>
```

Les animations -5-

- L'exemple [suivant](#) modifie à la fois la couleur de fond et la position de l'élément <div> lorsque l'animation est terminée à 25%, à 50% et à nouveau lorsque l'animation est terminée à 100%:

```
<style>
div {
  width: 100px;
  height: 100px;
  background-color: red;
  position: relative;
  animation-name: example;
  animation-duration: 4s;
}
/* Standard syntax */
@keyframes example {
  0%   {background-color:red; left:0px; top:0px;}
  25%  {background-color:yellow; left:200px; top:0px;}
  50%  {background-color:blue; left:200px; top:200px;}
  75%  {background-color:green; left:0px; top:200px;}
  100% {background-color:red; left:0px; top:0px;}
}
</style>
<div></div>
```


Les transitions

- Les transitions CSS vous permettent de modifier les valeurs de propriété en douceur (d'une valeur à une autre), sur une durée donnée.
- Pour créer un effet de transition, vous devez spécifier deux choses:
 - ✓ la propriété CSS à laquelle vous voulez ajouter un effet
 - ✓ la durée de l'effet (Si la partie de durée n'est pas spécifiée, la transition n'aura aucun effet car la valeur par défaut est 0.)

Les transitions -2-

- L'exemple suivant montre un élément <div> rouge 100px * 100px red. L'élément <div> a également spécifié un effet de transition pour la propriété width, d'une durée de 2 secondes.

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

- L'effet de transition commencera lorsque la propriété CSS spécifiée (largeur) changera de valeur.
- Maintenant, spécifions une nouvelle valeur pour la propriété width lorsqu'un utilisateur passe la souris sur l'élément <div>:

```
div:hover { width: 300px; }
```


Les transitions -3-

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

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

Les transitions -4-

- La propriété **transition-timing-function** spécifie la courbe de vitesse de l'effet de transition.
- La propriété **transition-timing-function** peut avoir les valeurs suivantes:
 - ✓ **ease** - spécifie un effet de transition avec un début lent, puis rapide, puis une fin lente (valeur par défaut)
 - ✓ **linear** - spécifie un effet de transition avec la même vitesse du début à la fin
 - ✓ **ease-in** - spécifie un effet de transition avec un démarrage lent
 - ✓ **ease-out** - spécifie un effet de transition avec une fin lente
 - ✓ **ease-in-out** - spécifie un effet de transition avec un début et une fin lents
- Exemple

Les transitions -5-

- La propriété **transition-delay** spécifie un délai (en secondes) pour l'effet de transition.

```
div {  
  transition-delay: 1s;  
}
```

- L'exemple suivant ajoute également une transformation à l'effet de transition.

la propriété abrégée transition: transition: width 2s linear 1s;

Animation Sténographie

- L'exemple ci-dessous utilise six propriétés d'animation:

```
div {  
  animation-name: example;  
  animation-duration: 5s;  
  animation-timing-function: linear;  
  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 animation abrégée :

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


Les canvas avec CSS3

<canvas> est un nouvel élément qui peut être utilisé pour **dessiner des graphismes via des scripts JavaScript**. Par exemple, Il peut être utilisé pour dessiner des graphes, faire des compositions de photos, des animations, ou même faire du traitement ou de l'affichage de vidéos en temps réel.

```
<!DOCTYPE html>
```

```
<html>
```

```
<body>
```

```
<canvas id="myCanvas" width="200" height="100" style="border:1px solid black;">
```

Your browser does not support the HTML5 canvas tag.

```
</canvas>
```

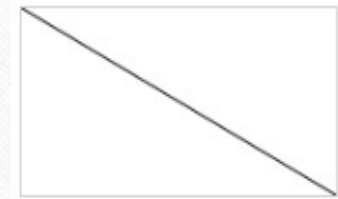
```
</body>
```

```
</html>
```



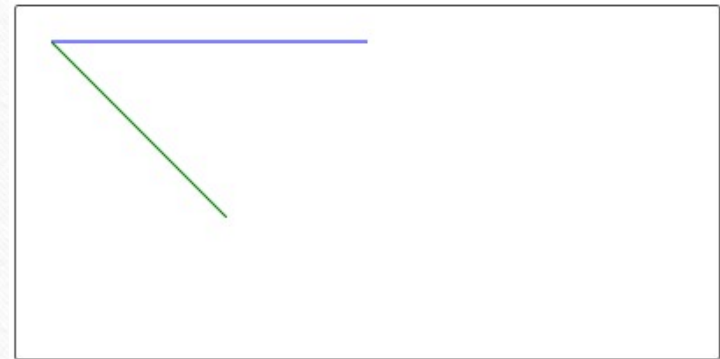
Tracer une ligne

```
<canvas id="myCanvas"> Texte alternatif pour les navigateur </canvas>
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.moveTo(0,0);//déplace le point de départ d'un nouveau sous-chemin vers les coordonnées (0,0).
ctx.lineTo(200,100);
ctx.stroke();// pour dessiner une ligne
</script>
```



Utiliser beginPath

```
var canvas = document.getElementById('canvas');  
var ctx = canvas.getContext('2d');  
// Premier chemin  
ctx.beginPath();  
ctx.strokeStyle = 'blue';  
ctx.moveTo(20, 20);  
ctx.lineTo(200, 20);  
ctx.stroke();  
// Second chemin  
ctx.beginPath();  
ctx.strokeStyle = 'green';  
ctx.moveTo(20, 20);  
ctx.lineTo(120, 120);  
ctx.stroke();
```



Dessiner un rectangle

stroke pour tracer un chemin.

```
ctx.rect(10, 10, 100, 100);  
ctx.stroke();
```



Utiliser fillStyle

fillStyle pour spécifie la couleur ou style à utiliser à l'intérieur des formes. La valeur par défaut est #000 (black).

```
ctx.fillStyle = 'blue';
```

```
ctx.fillRect(10, 10, 100, 100);
```

Ou bien

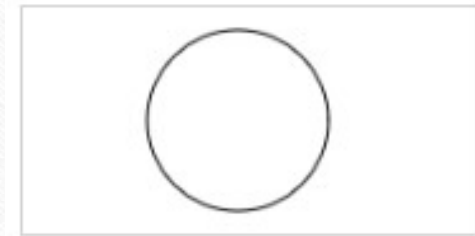
```
ctx.rect(10, 10, 100, 100);
```

```
ctx.fill(); // fill pour remplir un chemin
```



Dessiner un cercle

```
<canvas ...> Texte alternatif pour les navigateur </canvas>
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(95,50,40,0,2*Math.PI);
//ctx.arc(x, y, rayon, angleDépart(radion), angleFin(radion), sensAntiHoraire);
ctx.stroke();
</script>
```



Dessiner dégradé linéaire

```
<script>
var c = document.getElementById("myCanvas");
var ctx = c.getContext("2d");
// Create gradient
var grd = ctx.createLinearGradient(0,0,200,0);
grd.addColorStop(0,"red");
grd.addColorStop(1,"white");
// Fill with gradient
ctx.fillStyle = grd;
ctx.fillRect(10,10,150,80); //appliquer la couleur de remplissage
</script>
```



Texte Stroke

```
<canvas ...> Texte alternatif pour les navigateur </canvas>
```

```
<script>
```

```
var c = document.getElementById("myCanvas");
```

```
var ctx = c.getContext("2d");
```

```
ctx.font = "30px Arial";
```

```
ctx.strokeText("Hello World",10,50); //écrit un texte donné à la position (x, y) donnée
```

```
</script>
```



Hello World

Merci pour votre attention
