

Découvrir JavaScript

Attribution - Partage dans les Mêmes Conditions :
<http://creativecommons.org/licenses/by-sa/4.0/fr/>

Table des matières

Objectifs	3
I - Associer un JS à une page HTML	4
II - Exercice : Changeling	6
III - Les bases de la syntaxe JS	7
IV - Exercice : Calculate	10
V - Interagir avec les événements	11
VI - Exercice : Ping pong	14
VII - Modifier dynamiquement la page	15
VIII - Exercice : Cards	17
Solutions des exercices	19
Contenus annexes	21



Objectifs

- Savoir associer un programme JavaScript à une page HTML
- Connaître la syntaxe de base de JavaScript
- Savoir sélectionner un élément de la page en fonction de ses attributs *id* ou *class* et associer un comportement lorsqu'on interagit avec celui-ci (par exemple afficher une fenêtre de dialogue lorsqu'un utilisateur clic sur un élément de la page)
- Savoir modifier un élément de la page lorsqu'un événement se produit (par exemple le cacher lorsqu'on clique dessus)

I Associer un JS à une page HTML

JavaScript

Az Définition

JavaScript est un troisième langage qui vient compléter HTML et CSS. Un programme JavaScript (JS) est un fichier qui peut être associé à une page HTML afin de permettre la programmation des interactions avec l'utilisateur.

Il permet notamment de :

- réagir à des événements produits par l'utilisateur, par exemple afficher un message lorsqu'il appuie sur une touche du clavier,
- modifier la page affichée, par exemple masquer un div HTML lorsque l'utilisateur clique dessus.

 Syntaxe

On peut déclarer dans une page HTML quel fichier JS utiliser en mobilisant l'élément `<script src="script.js" defer></script>` dans la partie head.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <title>...</title>
5     <meta charset="utf-8">
6     <script src="script.js" defer></script>
7   </head>
8   <body>
9     ...
10  </body>
11 </html>
```

Fichier script.js

 Exemple

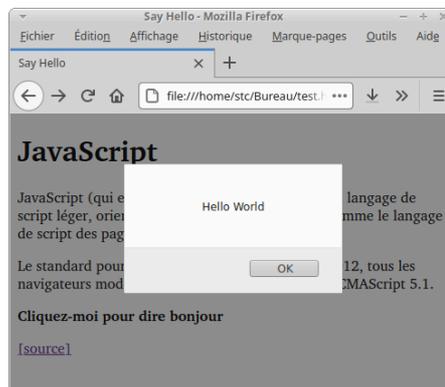
```
1 // Select an element in the HTML page
2 let p = document.querySelector('p.say-hello')
3
4 // Link the event "click" to the function "sayHello" on the element "p.say-
  hello"
5 p.addEventListener('click', sayHello)
6
7 // Code to execute when element "p.say-hello" is clicked
8 function sayHello() {
9   alert('Hello World')
10 }
11
```

Exemple

```

1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <title>Say Hello</title>
5     <meta charset="utf-8">
6     <script src="script.js" defer></script>
7   </head>
8   <body>
9     <h1>JavaScript</h1>
10    <p>JavaScript (qui est souvent abrégé en « JS ») est un langage de script
    léger, orienté objet, principalement connu comme le langage de script des pages
    web. </p>
11    <p>Le standard pour JavaScript est ECMAScript. En 2012, tous les
    navigateurs modernes supportent complètement ECMAScript 5.1.</p>
12    <p class="say-hello"><b>Cliquez-moi pour dire bonjour</b></p>
13    <p><a href="https://developer.mozilla.org/fr/docs/Web/JavaScript">[source]
    </a></p>
14  </body>
15 </html>
16

```



Tester le programme

Complément

Découvrir JavaScript sur MDN

Complément

https://developer.mozilla.org/fr/docs/Learn/JavaScript/First_steps

II Exercice : Changeling

Le fichier JavaScript ci-après permet :

- d'afficher une bordure autour d'un div doté d'une classe main lorsque la souris passe dessus,
- et de la supprimer lorsqu'on retire la souris.

```
1 let p = document.querySelector('div.main')
2
3 p.addEventListener('mouseover', addBorder)
4 p.addEventListener('mouseout', removeBorder)
5
6 function addBorder() {
7   this.style.border = 'solid'
8 }
9
10 function removeBorder() {
11   this.style.border = 'none'
12 }
```

Question

[solution n°1 p. 19]

Réaliser un fichier HTML qui affiche cet extrait de la chanson *The Changeling* des *Doors*.

« I'm a Changeling
See me change
I'm a Changeling
See me change



Indice :

III Les bases de la syntaxe JS

JavaScript est un langage interprété, c'est à dire un langage qui est exécuté par un autre programme : l'**interpréteur**.

L'interpréteur fonctionne de la façon suivante :

- il lit une ligne de code, l'analyse, l'exécute si elle est correcte et passe à la suivante ;
- il répète ce processus pour chaque instruction du programme.

Instruction

Az Définition

Une instruction est une ligne de code qui effectue une action : un affichage, un calcul, etc.

Les instructions sont exécutées **séquentiellement**, c'est-à-dire dans l'ordre et une par une.

Variables

Az Définition

Une instruction peut utiliser des **variables**, des objets auxquels on peut attribuer une valeur (un nom, une date, un nombre, etc.) et que l'on déclare avec le mot clé `let`. Sa valeur peut changer au cours du programme.

Déclaration de variables en JavaScript

📄 Syntaxe

```
1 let month = 'Avril'  
2 let dat = 12
```

Expression

Az Définition

Une expression est un morceau de code qui produit un **résultat**, par exemple une addition. Le résultat d'une expression peut être stockée dans une variable.

Instruction avec expression

👁 Exemple

```
1 let sum = 5 + 6
```

Fonctions

Az Définition

Une fonction est un morceau de code qu'un autre morceau de code peut appeler.

Les fonctions servent notamment à ne pas réécrire le code si on l'utilise plusieurs fois dans le programme..

Fonction d'affichage

[👁 Exemple](#)

Pour afficher des choses dans la console, on utilise la fonction prédéfinie `console.log()`.

```
1 console.log('Hello World')
2 console.log(4)
```

Règles de syntaxe

[💡 Fondamental](#)

Le JavaScript impose :

- d'aller à la ligne entre deux instructions ;
- de respecter la **casse**, c'est à dire les majuscules et minuscules (`console.Log()` n'équivaut pas à `console.log()`);
- de respecter les **espaces** (le nom d'une variable ne peut pas en contenir, en revanche elles sont nécessaires après des mots clés comme `let`).

[👁 Exemple](#)

```
1 const hello = 'Hello'
2 const world = 'World'
3 console.log(hello, world)
```

camelCase

[⊕ Complément](#)

La convention de casse utilisée en JavaScript est le camelCase.

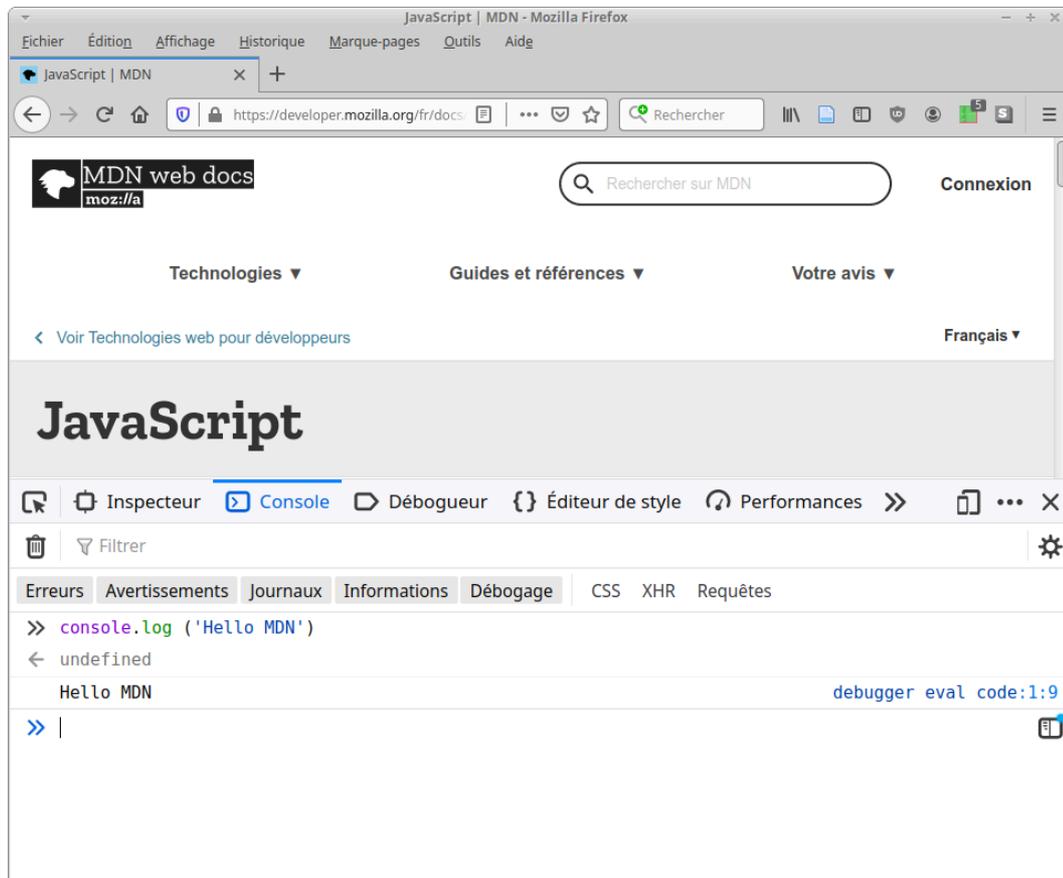
Exécuter du JavaScript dans un navigateur

[⊕ Complément](#)

On peut exécuter du JavaScript dans la **console web** du navigateur :

- sur **Chrome** : Clic droit > Inspecter ou `Ctrl+Shift+i`, puis onglet Console,
- sur **Firefox** : Clic droit > Examiner l'élément ou `Ctrl+Shift+k`, puis onglet Console.

On peut alors exécuter du code dans le champ de la console.



Repl.it

[Complément](#)

Il existe des interpréteur JavaScript en ligne, comme Repl.it¹.

¹. <https://repl.it>

IV Exercice : Calculate

Soit les fichiers HTML et JS suivants permettant d'effectuer des additions.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8">
5     <title>Exercice</title>
6     <script src="script.js" defer></script>
7   </head>
8   <body>
9     <div><label>x : </label><input id="x"/></div>
10    <div><label>y : </label><input id="y"/></div>
11    <div>x + y = <span id="xy"/>?<span></div>
12  </body>
13 </html>
14
15 let inputX = document.querySelector('#x')
16 let inputY = document.querySelector('#y')
17 let spanXY = document.querySelector('#xy')
18
19 inputX.addEventListener('change', calculate)
20 inputY.addEventListener('change', calculate)
21
22 function calculate() {
23   let x = parseInt(inputX.value)
24   let y = parseInt(inputY.value)
25   spanXY.textContent = x + y
26 }
27
```

Question 1

[solution n°2 p. 19]

Faites fonctionner le programme.

Question 2

[solution n°3 p. 19]

Écrivez une page web et un programme JS qui permettent de calculer, à partir d'un rayon : le diamètre, le périmètre et la surface du cercle, ainsi que le volume de la sphère.

V Interagir avec les événements

JavaScript permet de capter les événements qui se produisent en interaction à la page HTML et d'exécuter des fonctions lorsqu'ils se déclenchent.

Rappel

Il est possible d'associer un attribut `class` et/ou `id` aux éléments HTML, afin de les identifier :

- *Attribut class* (cf. p.21)
- *Attribut id* (cf. p.23)

Identifier un élément avec `document.querySelector`

Méthode

On peut stocker dans une variable l'adresse d'un élément HTML que l'on aura trouvé avec la fonction `querySelector`.

```
1 // Identify an element with a class
2 let element1 = document.querySelector('element1.class1')
3
4 // Identify an element with an id
5 let element2 = document.querySelector('element1#id1')
6
```

Exemple

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <title>Say Hello</title>
5     <meta charset="utf-8">
6     <script src="script.js" defer</script>
7   </head>
8   <body>
9     <h1>JavaScript</h1>
10    <p id="first-p">JavaScript (qui est souvent abrégé en « JS ») est un
    langage de script léger, orienté objet, principalement connu comme le langage de
    script des pages web. </p>
11    <p>Le standard pour JavaScript est ECMAScript. En 2012, tous les
    navigateurs modernes supportent complètement ECMAScript 5.1.</p>
12    <p class="say-hello"><b>Cliquez-moi pour dire bonjour</b></p>
13    <p><a href="https://developer.mozilla.org/fr/docs/Web/JavaScript">[source]
    </a></p>
14  </body>
15 </html>
16
1 let p1 = document.querySelector('#first-p')
2 let p2 = document.querySelector('p.say-hello')
3
4 alert(p1.textContent)
5 alert(p2.textContent)
```

Associer un événement avec addEventListener.

 Méthode

On peut associer des événements à des éléments HTML avec la fonction `addEventListener`.

```

1 // Call function1 when element1 receive event1
2 element1.addEventListener('event1', function1)
3

```

 Exemple

```

1 let p1 = document.querySelector('#first-p')
2 let p2 = document.querySelector('p.say-hello')
3 let body = document.querySelector('body')
4
5 p1.addEventListener('dblclick', printP1)
6 p2.addEventListener('click', sayHello)
7 body.addEventListener('keypress', printBody)
8
9 function sayHello() {
10   alert('Tu as cliqué sur moi.')
11 }
12
13 function printP1() {
14   alert('Merci de ne pas trop cliquer sur moi...')
15 }
16
17 function printBody() {
18   alert('Hey, pas avec le clavier !')
19 }
20
21 <!DOCTYPE html>
22 <html lang="fr">
23   <head>
24     <title>Say Hello</title>
25     <meta charset="utf-8">
26     <script src="script.js" defer></script>
27   </head>
28   <body>
29     <h1>JavaScript</h1>
30     <p id="first-p">JavaScript (qui est souvent abrégé en « JS ») est un
31     langage de script léger, orienté objet, principalement connu comme le langage de
32     script des pages web. </p>
33     <p>Le standard pour JavaScript est ECMAScript. En 2012, tous les
34     navigateurs modernes supportent complètement ECMAScript 5.1.</p>
35     <p class="say-hello"><b>Cliquez-moi pour dire bonjour</b></p>
36     <p><a href="https://developer.mozilla.org/fr/docs/Web/JavaScript">[source]
37     </a></p>
38   </body>
39 </html>
40

```

Quelques événements classiques

 Complément

- `click`, `dblclick` : quand l'utilisateur clique sur la souris
- `mouseover`, `mouseout` : quand l'utilisateur passe au entre et sort d'un élément avec sa souris

- `keypress` : quand l'utilisateur appuie sur une touche du clavier
- `change` : quand un élément HTML change de valeur (par exemple un élément d'un formulaire)
- `load` : quand la page HTML est totalement chargée

Tous les événements JavaScript

[⊕ Complément](#)

<https://developer.mozilla.org/fr/docs/Web/API/event>

Pour récupérer l'événement dans la fonction

[⊕ Complément](#)

```
1 element1.addEventListener('event1', (e) => fonction1(e))
2
```

Pour passer d'autres paramètres à la fonction

[⊕ Complément](#)

```
1 element1.addEventListener('event1', (e) => fonction1(e, x))
2
```

VI Exercice : Ping pong

[solution n°4 p. 19]

Question

Compléter le programme JS ci-après afin qu'il ait comme comportement :

- lorsque l'on clique sur Ping alors le cadre Ping s'allume en rouge et le cadre Pong d'éteint en noir
- lorsque l'on clique sur Pong alors le cadre Pong s'allume en rouge et le cadre Ping d'éteint en noir

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8">
5     <title>Exercice</title>
6     <link rel="stylesheet" type="text/css" href="style.css">
7     <script src="script.js" defer></script>
8   </head>
9   <body>
10    <div class="ping">Ping</div>
11    <div></div>
12    <div class="pong">Pong</div>
13  </body>
14 </html>
15
```

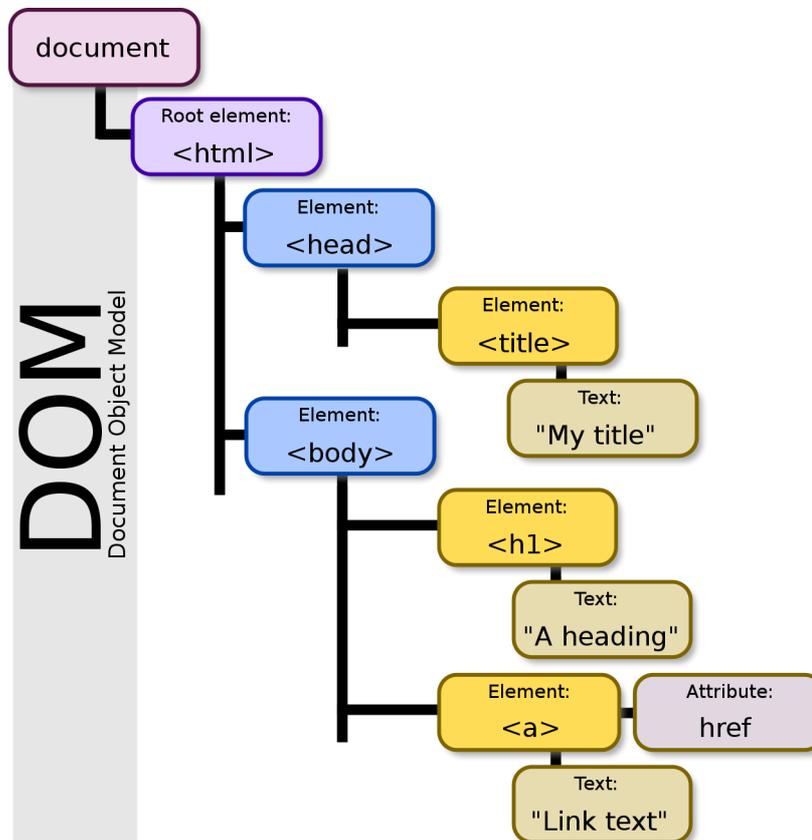
```
1 body {
2   display: flex;
3 }
4 div {
5   flex: 1;
6 }
7 .ping, .pong {
8   flex: 1;
9   font: 300% sans-serif;
10  text-align: center;
11  background-color: #d1d1d1;
12  border-style: dotted;
13 }
14
```

```
1 // something missing here...
2
3 ping.addEventListener('click', playPing)
4 pong.addEventListener('click', playPong)
5
6 function playPing() {
7   ping.style.color = 'red'
8   pong.style.color = 'black'
9 }
10
11 function playPong() {
12   //TODO
13 }
14
```

VII Modifier dynamiquement la page

Manipuler la page HTML avec le DOM

Avec JavaScript, la structure de la page est représentée par un arbre appelé DOM (pour *Document Object Model*), qui décrit la hiérarchie des balises au sein du document.



JavaScript fournit des fonctions pour manipuler les éléments du DOM. Par exemple, la méthode `document.querySelector('h1')` renvoie le premier élément `h1` de la page, que l'on pourra lire ou modifier.

👁 Exemple

```
1 let element1 = document.querySelector('#first-p')
2 element1.textContent = 'Here I am'
3 element1.style.color = 'green'
```

🔧 Méthode

Il est possible de combiner les actions sur le DOM avec les événements afin de créer une page interactive.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <title>Say Hello</title>
5     <meta charset="utf-8">
6     <script src="script.js" defer></script>
7   </head>
8   <body>
9     <h1>JavaScript</h1>
10    <p id="first-p">JavaScript (qui est souvent abrégé en « JS ») est un
    langage de script léger, orienté objet, principalement connu comme le langage de
    script des pages web. </p>
11    <p>Le standard pour JavaScript est ECMAScript. En 2012, tous les
    navigateurs modernes supportent complètement ECMAScript 5.1.</p>
12    <p class="say-hello"><b>Cliquez-moi pour dire bonjour</b></p>
13    <p><a href="https://developer.mozilla.org/fr/docs/Web/JavaScript">[source]
    </a></p>
14  </body>
15 </html>
16
```

```
1 let p1 = document.querySelector('#first-p')
2 let p2 = document.querySelector('p.say-hello')
3
4 p1.addEventListener('click', hideMe)
5 p2.addEventListener('click', displayHello)
6
7 function hideMe() {
8   this.hidden = true
9 }
10
11 function displayHello() {
12   this.textContent = 'Salut Monde'
13 }
14
```

VIII Exercice : Cards

Soit les trois fichiers HTML, CSS et JS suivants.

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8">
5     <title>Exercice</title>
6     <link rel="stylesheet" type="text/css" href="style.css">
7     <script src="script.js" defer</script>
8   </head>
9   <body>
10    <section class="cards"></section>
11    <button id="add">Ajouter</button>
12  </body>
13 </html>
14
```

```
1 .cards {
2   display: flex;
3   flex-wrap: wrap;
4   justify-content: center;
5   padding: 0 20px;
6 }
7 div {
8   padding: 70px 50px;
9   margin: 20px;
10  border-radius: 5%;
11  color: white;
12  font-weight: bold;
13 }
14 .green {
15   background: green;
16 }
17 .blue {
18   background: blue;
19 }
20 .red {
21   background: red;
22 }
23 .yellow {
24   background: yellow;
25 }
26 button {
27   background: white;
28   border: solid 1px black;
29   padding: 10px;
30   border-radius: 10px;
31 }
32
```

```
1 // Add a card to the section
2 function add() {
3   const colors = ["green", "red", "yellow", "blue"]
4   const cards = document.querySelector(".cards");
5   const num = Math.trunc(Math.random() * 9);
6   const randomColor = Math.trunc(Math.random() * 4);
7   const newCard = document.createElement("div");
8   newCard.classList = colors[randomColor];
9   newCard.innerText = num;
```

```
10 newCard.addEventListener('click', hide)
11 cards.appendChild(newCard);
12 }
13
14 // This lazy function does nothing
15 function hide() {
16   // Empty function
17 }
18
19 // Call function add once to display first card
20 add()
21
22 // When button is clicked, function add is called
23 button = document.querySelector('#add')
24 button.addEventListener('click', add)
25
```

Question 1

[solution n°5 p. 19]

Enregistrez et faites fonctionner ces fichiers.

Question 2

[solution n°6 p. 20]

Modifier le code JavaScript afin de permettre d'effacer une carte lorsque l'on clique dessus.

Solutions des exercices

Solution n°1

[exercice p. 6]

```
1 <!DOCTYPE html>
2 <html lang="fr">
3   <head>
4     <meta charset="utf-8">
5     <title>Exercice</title>
6     <script src="script.js" defer></script>
7   </head>
8   <body>
9     <div class="main">
10      <p>I'm a Changeling</p>
11      <p>See me change</p>
12      <p>I'm a Changeling</p>
13      <p>See me change</p>
14    </div>
15  </body>
16 </html>
17
```

Solution n°2

[exercice p. 10]

Solution n°3

[exercice p. 10]

Solution n°4

[exercice p. 14]

```
1 let ping = document.querySelector('.ping')
2 let pong = document.querySelector('.pong')
3
4 ping.addEventListener('click', playPing)
5 pong.addEventListener('click', playPong)
6
7 function playPing() {
8   ping.style.color = 'red'
9   pong.style.color = 'black'
10 }
11
12 function playPong() {
13   ping.style.color = 'black'
14   pong.style.color = 'red'
15 }
16
```

Solution n°5

[exercice p. 18]

Solution n°6

Il suffit de modifier la fonction `hide()` qui est appelée lorsqu'on clique sur une carte.

```
1 function hide() {  
2   this.hidden = true  
3 }
```

Contenus annexes

1. L'attribut Class

 Rappel

En HTML, l'attribut `class` peut être ajouté à tous les éléments : il permet d'associer une catégorie logique à un élément.

 Syntaxe

```
1 <p class="nom-de-la-classe">...</p>
```

 Exemple

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>W3C Cascading Style Sheets home page</title>
5     <meta charset="utf-8">
6     <link rel="stylesheet" href="style.css">
7   </head>
8   <body>
9     <h1 class="page-title">What is CSS?</h1>
10    <p>Cascading Style Sheets (CSS) is a simple mechanism for adding style
    (e.g., fonts, colors, spacing) to Web documents.</p>
11    <p>These pages contain information on how to learn and use CSS and on
    available software. They also contain news from the CSS working group.</p>
12    <p class="source"><a
    href="https://www.w3.org/Style/CSS/Overview.en.html">[source]</a></p>
13  </body>
14 </html>
15
```

Az Définition

On peut utiliser cet attribut `class` pour associer des styles différents à plusieurs éléments HTML identiques.

 Syntaxe

```
1 element.classe {
2   property: value;
3   ...
4 }
```

 Exemple

```

1 h1 {
2   font-family: Arial;
3   font-size: 150%;
4 }
5
6 p {
7   font-family: Monospace;
8 }
9
10 p.source {
11   font-weight: bold;
12 }

```


 Remarque

Il est possible de définir un style commun à plusieurs éléments différents qui ont la même classe.

```

1 .classe {
2   ...
3 }
4
1 .page-title {
2   font-family: Arial;
3   font-size: 150%;
4 }

```

Ici tous les éléments qui ont comme classe `page-title` activeront cette règle.

Kebab Case

[+ Complément](#)

On utilise la casse *kebab case* pour nommer les classes en HTML.

https://en.wikipedia.org/wiki/Letter_case²

2. Compléments

Associer un style à un élément unique (attribut id)

[Syntaxe](#)

Il est possible en HTML d'associer un identifiant unique à un élément et en CSS de le référencer en utilisant un croisillon (#) à la place du point utilisé pour cibler une classe.

```

1 <p id="personal-opinion">I don't like patent trolls.</p>
2
3
1 p#personal-opinion {
2   display: none;
3 }
```

Aller plus loin avec les sélecteurs

[+ Complément](#)

https://developer.mozilla.org/fr/docs/Web/CSS/S%C3%A9lecteurs_CSS

Gérer les unités de longueur

[Syntaxe](#)

Pour exprimer des longueurs en CSS, il existe de nombreuses unités, les plus courantes sont :

- % et em représentent une valeur relative à une valeur héritée d'un élément ancêtre (100% ou 1em représente la taille d'un paragraphe standard) ;
- mm, cm, px et pt représentent des tailles relatives à l'espace d'affichage (taille et résolution de l'écran).

Comprendre la gestion des unités

[+ Complément](#)

developer.mozilla.org/fr/docs/Web/CSS/length

Aller plus loin avec les polices

[+ Complément](#)

developer.mozilla.org/fr/docs/Web/CSS/@font-face

² https://en.wikipedia.org/wiki/Letter_case#Special_case_styles

Il est déconseillé d'utiliser des tableaux pour gérer la mise en page

 Attention

1. Les tableaux de mise en page diminuent l'accessibilité aux malvoyants.
2. Les tables produisent de la bouillie de balises (les pages ne sont plus humainement lisibles).
3. Les tableaux ne s'adaptent pas automatiquement aux terminaux.

« Les tableaux HTML ne doivent être utilisés que pour des données tabulaires.

Utiliser les tableaux pour la mise en page au lieu des techniques des CSS est une mauvaise idée.

(Quand NE PAS utiliser de tableaux en HTML ?, 
developer.mozilla.org/fr/docs/Apprendre/HTML/Tableaux/Basics)

Disposition spatiale

 Méthode

CSS intègre deux techniques pour gérer la mise en page : flexbox et grid.

- developer.mozilla.org/fr/docs/Web/CSS/CSS_Grid_Layout
- developer.mozilla.org/fr/docs/Learn/CSS/CSS_layout/Flexbox

Zen Garden (des exemples avancés)

 Complément

csszengarden.com³

³ <http://www.csszengarden.com>

