JavaScript et HTML

Cours 6

Jean-Jacques Lévy

jean-jacques.levy@inria.fr

http://jeanjacqueslevy.net/lp-js

Plan

- pages de style, CSS
- animation en JavaScript
- exemple HTML et CSS
- événements
- gestion d’événements

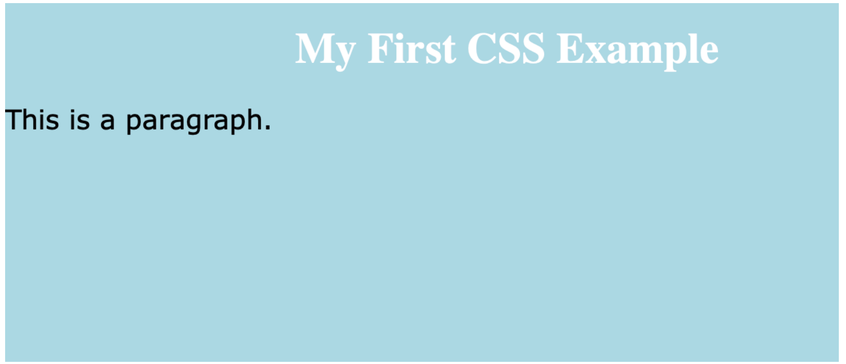
CSS

- Cascade Style Sheet (CSS) permet de séparer le style et la structure du document

```
body {
    background-color: lightblue;
  }

h1 {
    color: white;
    text-align: center;
  }

p {
    font-family: verdana;
    font-size: 20px;
  }
```

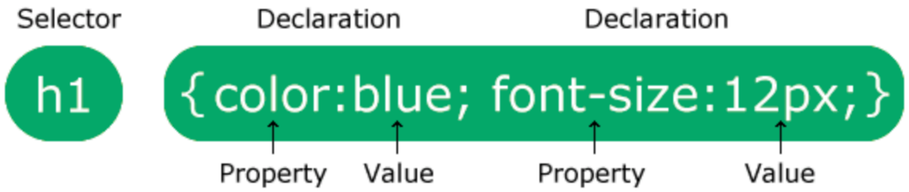


- voir `http://www.w3schools.com/css/default.asp`

  cf. la démo sur la page suivante pour voir la distinction entre structure du document et style d'affichage

CSS

- syntaxe très simple




```
body {
    background-color: lightblue;
  }

h1 {
    color: white;
    text-align: center;
  }

p {
    font-family: verdana;
    font-size: 20px;
  }
```

- sélecteurs

`h1`	élément
`#demo`	identifiant
`.center`	class
`p.center`	class dans un élément
`*`	tous les éléments

- grouper les sélecteurs

```
h1, h2, p {
    color: red;
  }
```

- voir `http://www.w3schools.com/css/default.asp`

cf. la démo sur la page suivante pour voir la distinction entre structure du document et style d'affichage

CSS

- 3 manières d’utiliser les CSS

- fichier externe (suffixe .css)

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="mystyle.css">
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

- style dans la page HTML

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: linen;
}

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

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

CSS

- 3 manières d’utiliser les CSS (suite)

- style comme attribut d’un élément

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;text-align:center;">This is a heading</h1>
<p style="color:red;">This is a paragraph.</p>

</body>
</html>
```

[si plusieurs styles applicables à un même élément, on prend le plus interne]

CSS

- quelques propriétés bien utiles

```
color
background-color
border
margin-top   margin-right    margin-bottom    margin-left
padding-top  padding-right   padding-bottom   padding-left
height       width
font-family
```

- sélecteur avec états

```
a:link    a:visited    a:hover    a:active
```

pour les ancres (*hyperlinks*)

- voir `http://www.w3schools.com/css/default.asp`

  pour une description détaillée

HTML Document Object Model

document.getElementById(*id*)

document.getElementsByTagName(*name*)

document.getElementsByClassName(*name*)

element.innerHTML = *new html content*

element.*attribute* = *new value*

element.style.*property* = *new style*

element.setAttribute*(attribute, value)*

document.createElement(*element*)

document.removeChild(*element*)

document.appendChild(*element*)

document.replaceChild(*new, old*)

document.write(*text*)

HTML avec Animation javascript

```
<!DOCTYPE html>
<html>

<style>
#container {
  width: 400px;
  height: 400px;
  position: relative;
  background: yellow;
}
#animate {
  width: 50px;
  height: 50px;
  position: absolute;
  background-color: red;
}
</style>


<body>

<p><button onclick="myMove()">Click Me</button></p>

<div id ="container">
  <div id ="animate"></div>
</div>
```


```
<script>
function myMove() {
  let id = null;
  const elem = document.getElementById("animate");
  let pos = 0;
  clearInterval(id);
  id = setInterval(frame, 5);
  function frame() {
    if (pos == 350) {
      clearInterval(id);
    } else {
      pos++;
      elem.style.top = pos + "px";
      elem.style.left = pos + "px";
    }
  }
}
</script>

</body>
</html>
```

- pour être plus exhaustif, voir*:* `http://www.w3schools.com/js/js_htmldom_animate.asp`

HTML avec Animation javascript

la fonction `frame` est argument de `setInterval`

```
<script>
function myMove() {
  let id = null;
  const elem = document.getElementById("animate");
  let pos = 0;
  clearInterval(id);
  id = setInterval(frame, 5);
  function frame() {
    if (pos == 350) {
      clearInterval(id);
    } else {
      pos++;
      elem.style.top = pos + "px";
      elem.style.left = pos + "px";
    }
  }
}
</script>

</body>
</html>
```

HTML

- l'attribut `class` permet de déclarer des styles communs pour plusieurs éléments
- l'attribut `id` donne un nom à un élément
- l'élément `div` démarre par une nouvelle ligne et s'étend au maximum sur sa gauche et sur sa droite, il a aussi une marge au dessus et en dessous
- l'élément `span` est un élément à l'intérieur d'un paragraphe. Pas de marges sur les côtés, ni au desus, ni en dessous

[l'attribut `class` de HTML n'a aucune relation avec les `class` de JavaScript]

HTML exemple (1/3)

Une page HTML

Un sous-titre

Un paragraphe normal mais avec du **g r a s** et de l'*italique* ainsi qu'un logo JS: dans une autre page

ou directement ici

- un premier item d'une liste non numérotée
- et le deuxième

Inscrits	IFT3220	IFT6810
Femmes	6	7
Hommes	44	16

HTML exemple (2/3)

```css
h1 { color: red; }

th {text-decoration: underline;
    color: olive;
   }

th.typeDonnee {
    font-style:italic;
    text-align:center;
    color:navy;
}

td { text-align: right; }

ul {background-color: gray;}

b { letter-spacing: 3pt; }

img {display:block;
    margin-left:auto;
    margin-right:auto;
}
```

dom1.css

```html
<!DOCTYPE html>
<html>
   <head>
      <meta http-equiv="content-type" content="text/html;charset=utf-8">
      <link rel="stylesheet" type="text/css" href="dom1.css" >
      <title>Page simple en HTML</title>
   </head>

   <body>
      <h1>Une page HTML</h1>
      <h2><i>Un sous-titre</i></h2>
      <p>Un paragraphe normal mais avec du <b>gras</b> et de
          l'<i>italique</i> ainsi qu'un logo JS:
          <a href="../pics/jslogo.jpg">dans une autre page</a>
      </p>
      <p>ou directement ici</p>
      <p>
           <img src="../pics/jslogo.jpg" alt="" height="71" width="54">
       </p>
      <ul>
         <li>un premier item d'une liste non numérotée</li>
         <li>et le deuxième</li>
      </ul>
      <table width="180" border="1" cellspacing="2" cellpadding="0">
```

dom1.html

HTML exemple (3/3)

```css
h1 { color: red; }

th {text-decoration: underline;
    color: olive;
   }

th.typeDonnee {
    font-style:italic;
    text-align:center;
    color:navy;
}

td { text-align: right; }

ul {background-color: gray;}

b { letter-spacing: 3pt; }

img {display:block;
    margin-left:auto;
    margin-right:auto;
}
```

dom1.css

```html
      <table width="180" border="1" cellspacing="2" cellpadding="0">
         <tr>
            <th class="typeDonnee">Inscrits</th>
            <th>IFT3220</th>
            <th>IFT6810</th>
         </tr>
         <tr>
            <th>Femmes</th>
            <td>6</td>
            <td>7</td>
         </tr>
         <tr>
            <th>Hommes</th>
            <td>44</td>
            <td>16</td>
         </tr>
      </table>
   </body>
</html>
```

dom1.html

HTML

- l'élément `table` se décompose en lignes `<tr>` (*table rows*)
- chaque ligne est composée de en-têtes `<th>` (*table headers*) ou de données `<td>` (*table data*)

Inscrits	IFT3220	IFT6810
Femmes	6	7
Hommes	44	16

```
<table width="180" border="1" cellspacing="2" cellpadding="0">
   <tr>
      <th class="typeDonnee">Inscrits</th>
      <th>IFT3220</th>
      <th>IFT6810</th>
   </tr>
   <tr>
      <th>Femmes</th>
      <td>6</td>
      <td>7</td>
   </tr>
   <tr>
      <th>Hommes</th>
      <td>44</td>
      <td>16</td>
   </tr>
</table>
</body>
</html>
```

Événements JS / HTML

- action extérieure (clic sur un bouton, page Web chargée, entrée de données)

- l'attribut `event` associe un code JavaScript à un élément

```
<button onclick="document.getElementById('demo').innerHTML = Date()">The time is?</button>
```

- ou en séparant plus le code JavaScript

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript HTML Events</h2>
<p>Click the button to display the date.</p>

<button onclick="displayDate()">The time is?</button>

<script>
function displayDate() {
  document.getElementById("demo").innerHTML = Date();
}
</script>

<p id="demo"></p>

</body>
</html>
```

Événements JS / HTML

- événements possibles:

onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

- voir `http://www.w3schools.com/js/js_htmldom_events.asp`

  où il y a un exemple pour chacun de ces événements

Événements JS / HTML

- le traitement d'un événement peut se faire en JavaScript grâce à la méthode addEventListener

- alors description de l'élément et code JavaScript sont mieux séparés

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript addEventListener()</h2>

<p>This example uses the addEventListener() method to attach a click event to a button.</p>

<button id="myBtn">Try it</button>

<script>
document.getElementById("myBtn").addEventListener("click", function() {
  alert("Hello World!");
});
</script>

</body>
</html>
```

Prochain cours

- un bon tutoriel JavaScript: `http://www.programiz.com/javascript`
- un autre tutoriel JavaScript: `http://www.w3schools.com/js`
- notions plus avancées de JavaScript (exceptions, prototypes, modules)
- quelques bibliothèques JavaScript
- essais de pages web sexy !