

Programmation Javascript

Rappels de HTML et CSS

JS

HTML : description du contenu d'une page web

- Langage à **balises**
- Les balises décrivent les éléments placés dans la page
 - Un nom de balise pour chaque type d'éléments
 - Exemples (structuration et mise en forme du texte):
 - `<p>...</p>` : paragraphe
 - `<h1>...</h1>`, ..., `<h6>...</h6>` : titres de niveau 1 à 6
 - `...` : liste à puces
 - `...` : liste numérotée
 - `...` : item de liste (à l'intérieur de `..` ou `..`)
 - `...` : mise en évidence légère du texte (en italique par défaut)
 - `...` : mise en évidence forte du texte (en gras par défaut)
 - `<mark>...</mark>` : marquage du texte (surlignage jaune par défaut)

HTML : description du contenu d'une page web

- Attributs des balises

- Placés à l'intérieur de la balise **ouvrante**
- Servent à donner des informations supplémentaires sur l'élément
- Ex :
 - `` : **image** avec attribut **src** indiquant la source du fichier
 - `Retour à l'accueil` : **lien hypertexte** avec attribut **href** indiquant la page cible du lien

HTML : description du contenu d'une page web

- Document HTML5 minimal

```
<!DOCTYPE html>
<html>
  <head>
    <!-- En-tête de la page -->
    <meta charset="utf-8" />
    <title>Titre (barre du navigateur)</title>
  </head>
  <body>
    <!-- Corps de la page -->
    ...
  </body>
</html>
```

HTML : description du contenu d'une page web

- Balises structurantes (sectionnement sémantique du contenu)
 - Servent à définir les différentes parties de la page
 - Ex :
 - `<header>...</header>` : **en-tête** de la page Web ;
 - `<nav>...</nav>` : zone de **navigation** (ex : menu);
 - `<main>...</main>` : zone **principale** de la page;
 - `<section>...</section>` : **section** regroupant des éléments ayant une thématique commune ;
 - `<article>...</article>` : **article** indépendant ;
 - `<aside>...</aside>` : zone d'**informations complémentaires** ;
 - `<footer>...</footer>` : **pied de page**.

HTML : description du contenu d'une page web

- Attribut d'identifiant : **id**
 - Sert à donner un identifiant **unique** à un élément
 - Ex :
 - `<section id="intro">`
- Attribut de classe : **class**
 - Sert à placer l'élément dans un ou plusieurs **groupes** (classes)
 - Ex :
 - `` (une classe)
 - `` (deux classes)
- **id** et **class** très utilisés pour la mise en forme CSS

CSS : mise en forme du contenu d'une page web

- Principe : désigner tel ou tel élément du HTML et définir sa mise en forme par un ou plusieurs **styles**
 - style = liste de propriétés de mise en forme avec leur valeur
- On placera le code de mise en forme CSS dans un **fichier séparé** d'extension .css
- Lien HTML/CSS :
 - Ajout d'une balise `<link .../>` dans l'élément `<head>` du **fichier HTML**

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

CSS : mise en forme du contenu d'une page web

- Déclaration d'un style CSS (dans le fichier .css)
 - Sélecteur d'élément(s) + liste de paires propriété/valeur

```
sélecteur {  
    propriété1 : valeur1 ;  
    propriété2 : valeur2 ;  
    ...  
}
```

CSS : mise en forme du contenu d'une page web

- Types de sélecteur
 - Par nom de balise : **p**, **h1**, **img**, ...
 - Par **identifiant** (utilisation du hashtag) : **#intro**
 - Par **classe** (utilisation du point) :
 - .vignette (une classe)
 - .vignette.logo (deux classes)
 - Par **attribut** : **img[title]**
 - Par **attribut+valeur** : **img[title="bandeau"]**
 - Sélecteur universel (sélectionne tout) : *****

CSS : mise en forme du contenu d'une page web

- Combinaisons de sélecteurs

- Liste de sélecteurs

sél1, sél2, sél3 { ... }

- Inclusion

sél1 sél2 { ... } : éléments de sél2 sont descendants de sél1

sél1 > sél2 { ... } : éléments de sél2 sont fils directs de sél1

CSS : mise en forme du contenu d'une page web

- Sélecteurs fonction d'une caractéristique de l'élément (pseudo-classes) :

sélecteur:caractéristique { ... }

Exemple :

- `li:hover { ... }` : sélectionne les éléments `li` survolés (hover) par la souris
- `p:last-child { ... }` : sélectionne l'élément `p` qui est le **dernier enfant** de son élément parent

CSS : mise en forme du contenu d'une page web

- Propriétés de mise en forme

- Texte :
 - `font-family`, `font-size`, `font-style`, `font-weight`, `text-decoration`, `text-align`, ...
- Couleurs et fond :
 - `color` (texte), `background-color`, `background-image`, `background-attachment`, `background-repeat`, `opacity`, ...
- Bordures et ombres :
 - `border-style`, `border-width`, `border-color`, `border-radius`, `box-shadow`, `text-shadow`, ...
- Dimensionnement, marges, débordement :
 - `width`, `height`, `min-width`, `min-height`, `max-width`, `max-height`, `margin` (extérieur), `padding` (intérieur), `overflow`, ...

CSS : mise en forme du contenu d'une page web

- Propriétés de mise en forme
 - Positionnement (1) :
 - Par défaut (flux normal):
 - Éléments **in-line** : **en ligne horizontalement**, comme les mots d'un texte
 - Éléments **block** : **les uns en dessous des autres** avec passage à la ligne et 100 % de la largeur
 - Changement du positionnement par défaut d'un élément : **display** : **block** ou **inline** ou **inline-block** ou **none**

CSS : mise en forme du contenu d'une page web

- Propriétés de mise en forme

- Positionnement (2) :

- Mode de positionnement :

- position** : absolute ou relative ou fixed

- top, left, bottom, right** (avec positions en pixels ou %)

- float** : left ou right

- Modèle de boite flexible :

- display** : flex ;

- flex-direction, flex-wrap, justify-content, align-items,**

- ...

CSS : mise en forme du contenu d'une page web

- Propriétés de mise en forme

- Positionnement (3) :

- Modèle de grille :

- Dans l'élément conteneur :

```
display : grid ;  
grid-template-areas : ". menu"  
                     "main comp"  
                     "piedDePage piedDePage";}  
  
grid-template-columns, grid-template-rows, ...
```

- Dans les éléments contenus (nommage pour la grille) :

```
grid-area : piedDePage ;
```

CSS : mise en forme du contenu d'une page web

- Media query (responsive design)
 - Règles de "déclenchement" de mises en forme (styles) CSS
 - But : adapter l'aspect d'un site web au support (smartphone, tablette, ordinateur, ...)
 - ✓ redimensionner un élément
 - ✓ cacher un élément
 - ✓ déplacer un élément
 - ✓ changer l'aspect d'un élément
 - ✓ ...
 - Il y a souvent plusieurs media query pour s'adapter à différents supports



CSS : mise en forme du contenu d'une page web

- Media query (responsive design)

- Syntaxe :

```
@media (condition de déclenchement) {  
    // liste de styles déclenchés si la condition est vraie  
    sélecteur1 {  
        propriété1:valeur1 ;  
        propriété2:valeur2 ;  
    }  
    sélecteur2 {  
        propriété3:valeur3 ;  
        propriété4:valeur4 ;  
    }  
    ...  
}
```

CSS : mise en forme du contenu d'une page web

- Media query (responsive design)
 - Approche classique : "**mobile first**"
 - Le CSS de base est fait pour téléphone mobile
 - Des media query servent à adapter l'aspect du site aux écrans plus grands (tablette, puis ordinateur par exemple)
 - L'**ordre des media query est important !**
 - Les conditions sont évaluées les unes après les autres, dans le sens d'écriture du code
→ du plus petit écran au plus grand

CSS : mise en forme du contenu d'une page web

- Media query (responsive design)

- Approche "mobile first" : exemple de media query

// styles pour mise en forme des petits écrans (<500px de large ici)

```
sélecteur1 {
```

```
    propriété1:valeur1 ;
```

```
}
```

@media (min-width : 500px){ // styles déclenchés pour des écrans >= 500px de large

```
sélecteur1 {
```

```
    propriété1:valeur2 ;
```

```
}
```

```
}
```

@media (min-width : 900px){ // styles déclenchés pour des écrans >= 900px de large

```
sélecteur1 {
```

```
    propriété1:valeur3 ;
```

```
}
```

```
}
```

Mementos HTML/CSS

- Référence HTML :

<https://developer.mozilla.org/fr/docs/Web/HTML/Element>

- Référence CSS :

<https://developer.mozilla.org/fr/docs/Web/CSS/Reference>

