

Remise à niveau HTML/CSS

Table des matières

Remise à niveau HTML/CSS.....	1
I. Utilisation d'un système de versionning : Git	2
II. HTML	2
1. Présentation HTML/CSS.....	2
2. Utiliser la console du navigateur	2
3. Balises/tags de base	2
4. Les attributs des balises.....	2
5. Position block/inline	2
6. Exercice de mise en page.....	2
7. Table au HTML.....	3
8. Formulaire HTML.....	3
9. Autres balises (html5, em/strong).....	3
III. CSS	3
1. Les propriétés courantes	3
2. Style inline.....	3
3. Sélecteur par tag / id (unique) / class.....	3
4. Priorité entre sélecteurs / règle « !important »	4
5. Positionnement.....	4
6. Positionnement avancé	4
7. Responsive design / Medias queries	4
8. Frameworks HTML/CSS/JS	4

I. Utilisation d'un système de versionning : Git

Pour faciliter les échanges de fichiers du formateur, on va utiliser le logiciel git.

Cela permet de « versionner » les fichiers sur un serveur, afin que tous les développeurs travaillent sur les mêmes sources en même temps, avec chacun sa propre version sur son poste.

Le code de ce cours est disponible à l'url : <https://github.com/ice-devel/formation2020-cda>

II. HTML

1. Présentation HTML/CSS

Le HTML/CSS sont deux langages de description. Ils servent à indiquer aux navigateurs quels éléments afficher dans une page ainsi que leur disposition.

C'est la base du web, les spécifications HTML/CSS sont basés sur un standard : tous les navigateurs les comprennent.

2. Utiliser la console du navigateur

Dans chaque navigateur, il y a un outil de debug : on peut y accéder en cliquant droit sur la page, et en sélectionnant « Inspecter l'élément », ou encore en appuyant sur la touche F12. Cela permet d'analyser le code HTML/CSS/JS en temps réel, de faire des modifications et visualiser ce qu'elles donnent, pour ensuite reporter les bonnes modifications dans le code source.

3. Balises/tags de base

- Html/head/body (title/charset)
- Paragraphes
- Images
- Blocs
- Titres
- Liens (url absolue / relative – attribut target)
- Listes à puces / listes numérotées

4. Les attributs des balises

- Attributs obligatoires

Certains attributs sont obligatoires. Par exemple, pour la balise image, il faut nécessairement renseigner la source via la balise « src », pour les liens, il faut indiquer l'URL via la balise « href », etc.

- Attributs facultatifs

Les « id », « class », « style » sont donc des attributs facultatifs, on les renseigne si besoin.

5. Position block/inline

Les éléments de type bloc prennent par défaut toute la largeur, et sont disposés les uns en dessous des autres.

Les éléments inline prennent la largeur de leur contenu et sont affichés les uns à côté des autres.

6. Exercice de mise en page

Créer un fichier HTML et intégrer le design fourni (exercice.jpg). Respectez bien les balises html, head, body et le doctype.

7. Table au HTML

Les tableaux HTML permettent d'afficher des données sous forme de lignes et de cellules, un peu comme une feuille Excel.

8. Formulaire HTML

- Les différents types de champs
- Les attributs

(min, max, minlength, maxlength, accept, placeholder)

- Les validations côté client

Never trusts user input

Validation html5 : se déclenche au moment de la soumission du formulaire

- Labels / Placeholders
- Envoi des données au serveur / attribut name / attribut action

Clé/valeur

- Method GET/POST

Avantages/inconvénients

9. Autres balises (html5, em/strong)

Header, Footer, Section, Article, Nav, Aside : sémantique

<audio> <video>

<iframe>

III. CSS

Le CSS permet de mettre en forme, de positionner des éléments HTML.

1. Les propriétés courantes

- Font-size : taille de la police
- Color : couleur de la police
- Font-style : italic ou non
- Font-weight : gras ou non
- Border : bordure (taille, type, couleur)
- Margin : marge extérieure
- Padding : marge intérieur
- Background-color : couleur de fond

2. Style inline

On peut également appliquer du style directement dans le HTML, grâce à l'attribut style. Tous les html peuvent avoir cet attribut. On peut donc écrire le css directement dans cet attribut, c'est le style « inline ».

3. Sélecteur par tag / id (unique) / class

Chaque élément HTML peut avoir un ID (grâce à l'attribut id), une ou plusieurs classes (grâce à l'attribut class). Le tag est simplement le nom de la balise.

Grâce à ces 3 éléments, on peut sélectionner précisément un ou plusieurs éléments dans la page HTML pour leur appliquer du style.

- # (diese) : pour sélectionner par ID
- . (point) : pour sélectionner par classe
- rien : pour sélectionner par tag/balise. On écrit simplement le nom de la basie

4. Priorité entre sélecteurs / règle « !important »

Règle !important > Style Inline > Sélecteur ID > Sélecteur Classe > Sélecteur Tag

Les priorités ne font qu'écraser les propriétés déjà définies.

5. Positionnement

- Static
- Absolute
- Relative

6. Positionnement avancé

- Flexbox : permet de créer des « grilles » afin de positionner des blocs proprement et facilement. Les uns à côté des autres / en dessous des autres / dans l'ordre / dans l'ordre inverse, de répartir la largeur de façon équitable entre les différents éléments, etc.

7. Responsive design / Medias queries

Dans le but d'appliquer un style différent aux mêmes éléments en fonction de la largeur du périphérique/navigateur, on peut utiliser les médias queries. Grâce à cela, on peut avoir un seul code HTML dont l'affichage est propre sur toutes tailles d'écran.

8. Frameworks HTML/CSS/JS

- Bootstrap
- MaterializeCSS
- Foundation