IUT Paris Descartes - Formation initiale novembre 2020
Etudiants de 1ère année J-M Ilié, M. Rincon, Y. Alt el Mahjoub
module CDIN

Travail personnel noté, à présenter en séance de TP pour la semaine du 14 décembre 2020.

Il s'agit de terminer le site ePortFolio commencé en TPs, ou alternativement votre propre site ePortFolio.
Les redoublants devront faire leur propre site. Vous trouverez ci-dessous les caractéristiques attendues dans le contexte du site du TP. Des caractéristiques sensiblement équivalentes seront demandées au cas où vous préféreriez faire votre propre site. Vous (relirez) vos polycopiés électroniques de cours sur css, notamment CSS-strategie et CSS-dynamique, css-bootstrap qui vous ont été envoyés. A noter qu'un complément de cours et TPs pendant la semaine du 30 novembre vous introduira la notion de formulaires HTMx.

HTML5 :

- Le site ePortFolio comporte **plusieurs pages** écrites en HTML5 et possédant une chartre graphique commune à ces pages. Voici un exemple de pages :
  - `accueil.html`
  - `competences.html`
  - `experiences.html`
  - `loisirs.html`

- Prévoir un **contenu minimum** par page que vous mettrez dans des balises <article>. A noter qu'il ne s'agit pas à ce niveau de nécessairement produire un contenu très élaboré de ces pages, ni à l'inverse d'introduire un charabia.
- Faire **la chasse et éliminer** les balises superflues et les identifiants superflus. Préférer l'imbrication à la sur-identification des balises. Privilégier l'exploitation de classes de style.
- **Créer un tableau** dans une des pages du site. Si la page est `loisirs.html`, ce pourra être par exemple, un contenu d'articles affichant les résultats des compétitions de derniers tournois d'échecs. Attention, ce tableau doit montrer votre efficacité, n'hésiter pas à créer un tableau riche qui nécessite des fussions de cellules.
- **Créer un formulaire** dans une des pages du site (les formulaires sont vus en TPs dans la semaine du 5 décembre et sont décrit en fin du poly CSS-bases). Ce formulaire pourrait servir à renseigner un livre d'or. Différents champs (input, select, textarea) et différents types de champs (pasword,date,email…) seront utilisés dans le formulaire.
- Utiliser **menu et sous-menus** pour indexer les pages. Organiser la partie `<nav>` pour que la balise `<il>` de chaque menu imbriqué son ou ses sous menus.
- En se référant à la fin du poly de CSS-stratégies, utiliser les **media-queries** pour charger de façon exclusive un fichier de style CSS pour une taille d'écran >1200px : **`stylel.CSS`**, et un fichier pour une taille d'écran inférieur à 1200px (voir aussi dans les entêtes des pages du TP), **styleh.CSS**. Prévoir éventuellement un CSS commun en sus. Vous pourrez simuler le passage de l'un à l'autre CSS en zoomant sur votre page.
- Ecrire **l'entête** de vos pages, notamment avec les balises <title> pour le titre et <meta> pour les informations d'auteur et de description de contenu.
- **Valider** votre page.

CSS3 :

- **Ordonner** vos règles de styles CSS pour qu'ils soient facilement maintenables et ajouter quelques **commentaires** afin de séparer clairement les parties traitées.
- Utiliser **indentations** pour rendre vos règles de style bien visibles et maintenables.

- (re) Faire la présentation de votre **menu <nav>**. Pour un item menu donné, Faire pointer les liens des sous menus vers des sections différentes que l'on aura identifié (notion d'ancres construites à partir de balises <a> avec identifiant, au sein d'une page HTML). Une alternative sera de créer des pages HTML5 différentes pour les contenus associés aux sous-menus.
- Afficher les articles dans des **boites de même dimension**. Des ascenseurs apparaîtront si la boite est de taille insuffisante pour le texte contenu (spécifier la propriété de style overflow). Sous-dimensionner la page pour vérifier l'apparition des ascenseurs.
- **Maîtriser le redimensionnement de vos pages**, en cas de zoom ou de modification de la taille de la fenêtre.
    - Utiliser des **dimensions proportionnelles** majoritairement.
    - Modifier par exemple le style de la balise en imposant une largeur minimum, grâce la propriété `min-width`, en deçà de laquelle un ascenseur horizontal doit apparaître. Le principe est d'avoir une largeur minimum suffisante pour permettre et préserver l'alignement de vos blocks et de leurs contenus comme les parties de votre entête.
    - Tester la différence et remarquer la transformation de votre présentation en réduisant la largeur de votre fenêtre ou en zoom-avant sur la page.
    - Imposer éventuellement une largeur maximum avec `max-width`.
    - Dans un premier temps, laisser libre la hauteur de votre page, avec la propriété `height:auto`. La conséquence est que votre page peut nécessiter un défilement avec l'ascenseur vertical. En imposant une hauteur minimum avec `min-height`, l'ascenseur vertical apparaîtra, si la hauteur de la fenêtre est réduite en deçà de cette hauteur.
- **Tester** votre site sur 2 navigateurs significatifs, dont firefox.
- **Valider** votre pages avec fichiers CSS attachés.

CSS3 animation

- **Animer vos pages** et vos images de différentes façons (voir un exemple en fin du poly CSS-dynamique). Par exemple,

    - animer le menu en faisant apparaître différemment le menu survolé ou le menu activé (le menu activé correspond à la page en cours d'affichage).

    - Animer le formulaire pour que un bloc d'informations initialement caché (`display:none`) apparaisse en cas de survol d'un champ. Pour faciliter cette animation, placer le champ et son bloc d'information dans un même bloc parent, par exemple placé dans un élément de liste (entre `<li>` et `</li>`).

      Note : on peut changer les styles pour des balises dans un état donné.
      li:hover {} permet de modifier le styles des balises li quand leur contenu est survolé par la souris.
      li:hover a {} permet de modifier le styles des liens quand les li qui les imbriquent sont survolés.

BOOTSTRAP :

- Démontrer vos capacités à exploiter la bibliothèque CSS bootstrap, pour simplifier la gestion de la présentation de vos pages ou spécifier la structure de vos pages suivant un mode de responsive design. L'idéal est de faire du bootstrap sur une page complète.