

LPW

Langages de programmation web

101 Les bases

Table des matières

1 Objectifs, contenus & organisation.....	3
2 Mise en jambes.....	4
2.1 HTML/CSS: Dessiner des drapeaux.....	4
2.1.1 HTML.....	4
2.1.2 CSS.....	5
2.1.3 Résultat.....	8
2.2 JS/DOM: Colorier du texte.....	9
2.2.1 Comportement.....	9
2.2.2 HTML.....	9
2.2.3 CSS.....	9
2.2.4 JS/DOM.....	9
2.2.5 Résultat.....	10
2.3 JS: Animer une horloge digitale.....	11
2.3.1 HTML.....	11
2.3.2 CSS.....	11
2.3.3 JS.....	11
2.3.4 JS Intermédiaire.....	11
2.3.5 JS Avancé.....	12
2.3.6 Résultat.....	12
2.3.7 Bonus: Horloge analogique.....	13
2.4 JS/DOM: Damier.....	14
2.4.1 HTML.....	14
2.4.2 CSS.....	14
2.4.3 JS.....	14
2.5 JS/DOM: Todo List.....	15
2.5.1 HTML / CSS.....	15
2.5.2 JS.....	15
2.5.3 Persistence.....	15

1 Objectifs, contenus & organisation

Objectifs

Approfondir ses connaissances et sa maîtrise des langages utilisés dans les applications web, en apprenant et en appliquant le paradigme de la programmation orientée objet.

Contenus associés

- Réaliser des interfaces au moyen d'un moteur de templating PHP ;
- Comprendre et appliquer les principes de base de la programmation orientée objet ;
- Réaliser des applications single-page avec React ;
- Connaître et savoir situer d'autres langages web populaires.

Durée

- 144h
- 36 cours

Evaluation

- Exercices journaliers (135/240)
- Examen sommatif (105/240)

2 Mise en jambes

Exercices en autonomie permettant de revoir et d'approfondir les connaissances acquises en HTML, CSS & Javascript, en préparation du module React.

2.1 HTML/CSS: Dessiner des drapeaux

Création des drapeaux du BeNeLux

- sans utiliser les unités **px**
- en utilisant que des **div** avec des couleurs de fond
- sans mediaqueries

2.1.1 HTML

- Valide !
- Nom du fichier: **flag.html**
- Charger le CSS depuis un fichiers externe **flag.css**
 - pas de <style>
 - pas d'inline
- Uniquement des **div** dans le body
- Nom des classes
 - Le conteneur des 3 drapeaux: **flags**
 - Chaque drapeau: **flag** suivi du code ISO 3166 alpha 3 du pays
Belgique: **bel**, Pays-Bas: **nld**, Luxembourg: **lux**
 - Les mâts: **pole**
 - Le conteneur des bandes tricolores: **stripes**
 - si horizontales: **horizontal**
 - si verticales: **vertical**
 - Les bandes de couleurs: **start, middle, last**

2.1.2 CSS

Disposition

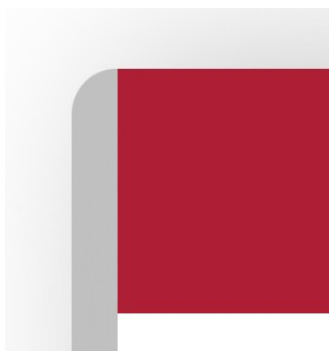
- Utiliser *flexbox* (pas de margin ou padding, sauf pour annuler celles du *body*)
- Suivre l'ordre de l'acronyme BeNeLux
- L'espace entre les drapeaux sont égaux
- Centrage vertical (donc 25%, 50%, 25%, puisque les drapeaux prennent 50%)

Drapeaux

- en hauteur, ils prennent la moitié de l'écran
- hauteur minimum de 300px (en cas de zoom)
- couleurs réglementaires:
 - https://en.wikipedia.org/wiki/Flag_of_Belgium
 - https://en.wikipedia.org/wiki/Flag_of_the_Netherlands
 - https://en.wikipedia.org/wiki/Flag_of_Luxembourg
- leur proportion (sans le mât) est de 2:3
- un ombrage argenté (silver) de 2em, sans offset.
- chaque bande de couleur mesure 1/3 du drapeau

Mâts

- Argenté (silver), 0.5em de large
- la partie supérieur en *quart de lune*



- la partie inférieure en *demi-lune*



Responsive

Si la taille de la fenêtre change,

- les drapeaux restent en colonne
- les ombrages ne se touchent pas



2.1.3 Résultat



2.2 JS/DOM: Colorier du texte

2.2.1 Comportement

Lorsque le formulaire est soumis avec du contenu dans l'*input*:

- Le texte de l'*input* sera placé dans le *div*
- Chaque lettre aura une couleur différente,
- Les couleurs sont en séquence (ex: red, green, blue, red, green, blue, red, green, ..)

Lorsque le formulaire est soumis sans contenu dans l'*input*, le contenu *div* est supprimé

2.2.2 HTML

- Créer un formulaire ne contenant qu'un *input* et un *button*
- Créer un *div#colorized*, sous le formulaire qui accueillera le texte colorié
- Placer le CSS dans **colorizer.css**
- Placer le JS dans **colorizer.js**

2.2.3 CSS

- Trois classes pour appliquer les couleurs (ex: color_1, color_2, color_3)
- Texte colorié: taille 2em, alignement centré

2.2.4 JS/DOM

Utiliser les techniques et fonctions DOM natives suivantes:

- getElementById
- addEventListener
- appendChild & removeChild
- createElement & createTextNode

2.2.5 Résultat



2-CE-X75

2.3 JS: Animer une horloge digitale

2.3.1 HTML

- L'heure est un lien qui mène à l'horloge analogue
- Le CSS et le JS dans des fichiers externes

2.3.2 CSS

- Texte centré horizontalement & verticalement (flexbox)
- Taille: 10em;

2.3.3 JS

- Le format d'affichage sera HH:MM:SS
- Créer un objet de la classe Date pour récupérer heure, minute et seconde
- Définir deux fonctions:
 - leadingZeros, qui s'assure que les chiffres deviennent des nombres
 - startClock, qui démarre l'application
- Utiliser setTimeout pour l'animation, (startClock est le callback)

2.3.4 JS Intermédiaire

- utiliser un tableau pour stocker heure, minute et seconde
- utiliser map pour les formater (leadingZeros est le callback)
- utiliser join pour générer l'heure en string au format HH:MM:SS

2.3.5 JS Avancé

- utiliser la notation *arrow function* pour vos fonctions
- écrire la fonction `leadingZeros` sur une seule ligne
- écrire la fonction `startClock` sur 3 lignes

2.3.6 Résultat



23:13:18

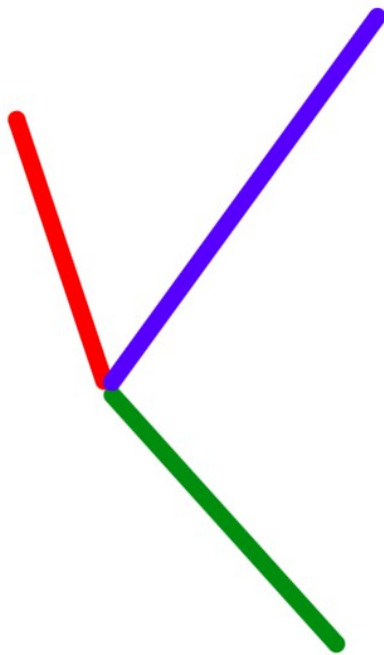
2.3.7 Bonus: Horloge analogique

Animation:

- JS pour le timing
- CSS pour la rotation

Présentation

- 3 aiguilles de taille différente, de couleur différente
- Positionnement: absolute/relative
- Taille 50% de la viewport



2.4 JS/DOM: Damier

2.4.1 HTML

- Uniquement le squelette html, head, body
- Les fichiers CSS et JS sont externes

2.4.2 CSS

- #FFF et #000 pour les couleurs du damier
- Utiliser CSS Grid pour la disposition

2.4.3 JS

- Générer l'HTML du damier grace au DOM
- La dimension du damier passée par URL
- Si la taille correspond à celles d'un jeu d'échec, placer les pions
- Les pions seront "dessinés" grâce à leurs caractères unicode

2.5 JS/DOM: Todo List

2.5.1 HTML / CSS

- Créer un formulaire contenant un seul champs d'entrée et un bouton de soumission
- Sous le formulaire, créer une liste qui contiendra les tâches et les actions

2.5.2 JS

- Ajouter une tâche grâce au formulaire
- Supprimer une tâche d'un simple click
- Modifier une tache, dans un modal
- Marque une tâche comme accomplie
 - elle apparait en bas de liste,
 - sont contenu est barré

2.5.3 Persistence

Lorsque la page est rechargée, la liste ne disparaît pas.

Appliquer et justifier le choix de la technique utilisée parmi

- CookieStorage
- LocalStorage
- SessionStorage