



Formation Python-Django - 0

Initiation

Résumé: Cette première journée vous permettra d'aborder les bases du développement Web. Au menu : HTTP, HTML, css et l'intégration de scripts Javascript existants dans vos pages.

Version: 1

Table des matières

I	Préambule	2
II	Règles communes	3
III	Exercice 00	4
IV	Exercice 01	5
V	Exercice 02	6
VI	Exercice 03	8
VII	Exercice 04	9
VIII	Exercice 05	10
IX	Rendu et peer-évaluation	11

Chapitre I

Préambule

Voici ce que nous dit wikipédia de la *Balaenoptera musculus* :

La baleine bleue (*Balaenoptera musculus*), appelée aussi rorqual bleu, est une espèce de cétacés de la famille des *Balaenopteridae*. Pouvant dépasser 30 mètres de longueur et 170 tonnes, c'est le plus gros animal vivant à notre époque et, dans l'état actuel de nos connaissances, le plus gros ayant jamais vécu sur Terre.

Long et mince, le corps de la baleine bleue peut prendre diverses teintes de gris-bleuté sur le dos et un peu plus clair en dessous. On dénombre au moins trois sous-espèces distinctes : *B. m. musculus* dans l'Atlantique Nord et le Pacifique Nord, *B. m. intermedia* de l'océan Antarctique et *B. m. brevicauda* découverte dans l'océan Indien et dans le sud de l'océan Pacifique. *B. m. indica*, découverte dans l'océan Indien, pourrait être une autre sous-espèce. Comme les autres baleines, la baleine bleue se nourrit essentiellement d'un petit crustacé, le krill, mais également de petits poissons et parfois de calmars.

Les baleines bleues furent abondantes dans presque tous les océans avant le début du XXe siècle. Pendant près de quarante ans, elles furent chassées par les baleiniers qui ont amené l'espèce au bord de l'extinction avant qu'elle ne soit protégée par la communauté internationale en 1966. Un rapport de 2002 estimait qu'il y avait entre 5 000 et 12 000 baleines bleues à travers le monde, localisées dans au moins cinq groupes. Des études plus récentes sur la sous-espèce *B. m. brevicauda* suggèrent qu'il pourrait s'agir d'une sous-estimation. Avant la chasse industrielle à la baleine, la plus forte population se trouvait dans l'Atlantique, qui en comptait approximativement 240 000 (entre 202 000 et 311 000). L'espèce est considérée comme menacée.

Aucune baleine n'a été maltraitée lors de la rédaction de ce sujet.


Chapitre II

Règles communes

- Votre projet doit être réalisé dans une machine virtuelle.
- Votre machine virtuelle doit avoir tout les logiciels necessaire pour réaliser votre projet. Ces logiciels doivent être configurés et installés.
- Vous êtes libre sur le choix du systmème d'exploitation à utiliser pour votre machine virtuelle.
- Vous devez pouvoir utiliser votre machine virtuelle depuis un ordinateur en cluster.
- Vous devez utiliser un dossier partagé entre votre machine virtuelle et votre machine hôte.
- Lors de vos évaluations vous allez utiliser ce dossier partager avec votre dépôt de rendu.
- Vos fonctions de doivent pas s'arrêter de manière inattendue (segmentation fault, bus error, double free, etc) mis à part dans le cas d'un comportement indéfini. Si cela arrive, votre projet sera considéré non fonctionnel et vous aurez 0 au projet.
- Nous vous recommandons de créer des programmes de test pour votre projet, bien que ce travail **ne sera pas rendu ni noté**. Cela vous donnera une chance de tester facilement votre travail ainsi que celui de vos pairs.
- Vous devez rendre votre travail sur le git qui vous est assigné. Seul le travail déposé sur git sera évalué. Si Deepthought doit corriger votre travail, cela sera fait à la fin des peer-evaluations. Si une erreur se produit pendant l'évaluation Deepthought, celle-ci s'arrête.

Chapitre III

Exercice 00

	Exercice : 00
Exercice 00 : Premier script shell	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <i>myawesomescript.sh</i>	
Fonctions Autorisées : <i>curl, grep, cut</i>	

Si **Twitter** n'a aucun secret pour vous, vous connaissez très probablement bit.ly : un raccourcisseur d'URLs bien pratique.

Le but de cet exercice est d'écrire et de rendre un script shell qui affiche l'adresse réelle d'une adresse bit.ly supposé valide (comprenez "L'adresse vers laquelle renvoie le lien bit.ly").

Comme explicitement écrit dans le cartouche de cet exercice, vous n'avez droit d'utiliser que les trois commandes shell suivantes pour réaliser cet exercice : **curl**, **grep** et **cut**. Votre meilleur point de départ est de lire le manuel de la commande **curl**. Pour cela, tapez **man curl** dans votre terminal.

Voici un exemple du comportement attendu de votre script shell :


```
$> ./myawesomescript.sh bit.ly/1072s3U
http://42.fr/
$>
```

L'exemple ci-dessus montre clairement que votre script doit être exécutable. L'interprète à utiliser est **/bin/sh**.

Rendez-votre script dans un dossier **ex00** à la racine de votre dépôt.

Chapitre IV

Exercice 01

	Exercice : 01
Exercice 01 : Votre CV en HTML	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <i>cv.html</i>	
Fonctions Autorisées : <i>n/a</i>	

Vous devez réaliser un CV en HTML/css et respecter les contraintes suivantes :


- Vous devez respecter la sémantique de vos balises HTML, ainsi que la séparation du fond et de la forme.
- Vous devez produire un fichier HTML cohérent dans sa structure avec un contenu minimum imposé : nom, prénom, compétences et parcours.
- Vous devez afficher au moins un titre avec la balise `title` et un titre avec la balise `h1`.
- Vous devez utiliser au moins un tableau avec les balises `table`, `th`, `tr` et `td`.
- Vous devez utiliser au moins une liste avec la balise `ul` et une liste avec la balise `ol`. Les éléments doivent utiliser la balise `li`.
- Les bordures des tableaux doivent être visibles (`solid`). Les bordures des tableaux doivent être fusionnées (`collapse`).
- La cellule la plus en bas à droite de chaque tableau doit avoir `#424242` comme couleur de bordure.
- Vous devrez utiliser une solution syntaxique différente pour chacune des deux consignes précédentes : pour la première utilisez la balise `style` dans le `head` de votre page. Pour la seconde, utilisez un attribut `style` dans une balise qui vous paraît adaptée.



Pas de consigne particulière pour la véracité des informations. Vous pouvez réaliser un CV farfelu si le coeur vous en dit.

Chapitre V

Exercice 02

	Exercice : 02
Exercice 02 : Formulaire d'envoi d'emails	
Dossier de rendu : <code>ex02/</code>	
Fichiers à rendre : <code>form.html</code>	
Fonctions Autorisées : <code>n/a</code>	

Réalisez un formulaire HTML qui représente les informations usuelles d'un contact quelconque. Ce formulaire doit proposer tous les champs suivants :

- `Firstname` : un champ texte.
- `Name` : un champ texte aussi.
- `Age` : vous devez utiliser le champ numérique spécifique au HTML5.
- `Phone` : vous devez utiliser le champ `tel` spécifique au HTML5.
- `Email` : vous devez utiliser le champ email spécifique au HTML5.
- `Student at 42 ?` : vous devez utiliser le champ checkbox.
- `Gender` : vous devez utiliser des boutons radio avec les valeurs `Male`, `Female` et `Other`.
- Un bouton de soumission du formulaire. L'attribut `onclick` de votre bouton doit être : `'displayFormContents();'`.

La tarball `d00.tar.gz` en annexe de ce sujet contient un sous-dossier `ex02/` qui lui-même contient un fichier Javascript `popup.js` écrit par le fils de votre patron en stage dans votre entreprise. Comme il serait inacceptable que vous fassiez passer le fils du patron pour un incompetent en programmation, vous ne pouvez pas du tout modifier son fichier qui doit donc être utilisé tel quel.



Une lecture attentive et une compréhension superficielle du code Javascript fourni sont requises pour réussir cet exercice.

Vous devez intégrer correctement ce fichier **Javascript** à votre page **HTML**. Si votre code **HTML** est correct, l'appui sur le bouton du formulaire fera apparaître une popup ultra-moderne contenant les champs et valeurs de votre formulaire. Dans tous les autres cas, votre code **HTML** est faux.

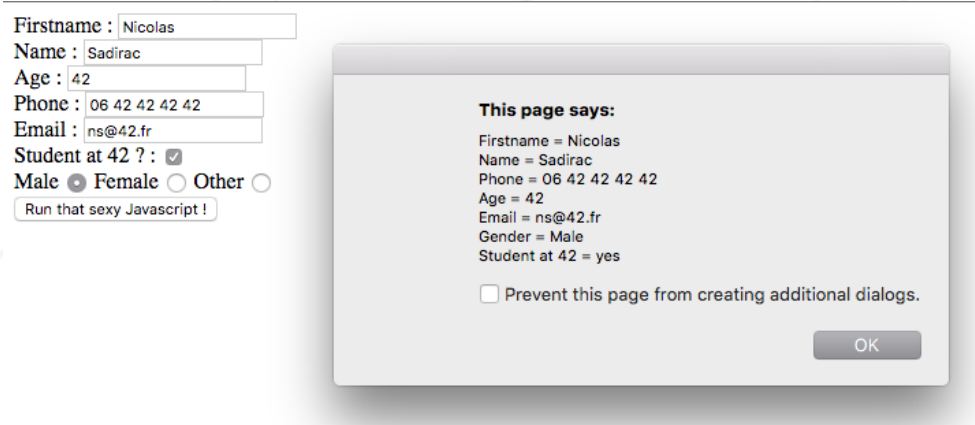



FIGURE V.1 – Illustration non contractuelle du résultat attendu.

Chapitre VI

Exercice 03

	Exercice : 03
Exercice 03 : Reproduction d'une page web	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : <i>copy.html</i>	
Fonctions Autorisées : <i>n/a</i>	

Une entreprise concurrente a mis en ligne une page web plus jolie que la votre. Grâce à un espionnage industriel digne du cinema Hollywoodien, votre patron se procure un screen shoot de la page et le fichier `css` qui va avec. Ces deux fichiers sont à votre disposition dans les annexes de ce sujet dans l'archive `d00.tar.gz` et son sous-dossier `ex03/`.

Vous devez reproduire cette page le plus fidèlement possible !

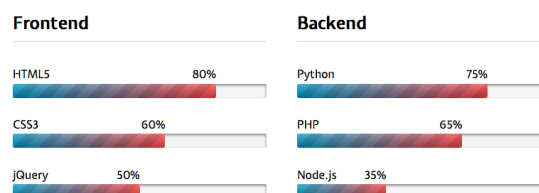



FIGURE VI.1 – Le screenshot de la page à reproduire. Echelle de l'image non contractuelle.

Vous devrez là encore séparer le fond et la forme, respecter la sémantique des balises utilisées et maintenir une structure logique dans votre document.

Vous devez obligatoirement utiliser le fichier `css` fourni sans le modifier. Une version “fraîche” de ce `css` sera utilisée en soutenance pour vérifier que vous avez bien respecté cette consigne.

Chapitre VII

Exercice 04

	Exercice : 04
Exercice 04 : Integration de snippets JS.	
Dossier de rendu : <i>ex04/</i>	
Fichiers à rendre : snippets.html	
Fonctions Autorisées : n/a	

La tarball `d00.tar.gz` en annexe de ce sujet contient un sous-dossier `ex04/` qui lui-même contient quatre fichiers : `file1.js`, `file2.js`, `file3.js` et `file4.js`.


Vous devez créer et rendre un fichier `snippets.html` qui doit importer les quatre scripts de telle manière que la pop-up s'affiche **correctement** (pas de caractères bizarres donc).



Vous ne pouvez qu'importer les scripts en question, vous ne pouvez pas les modifier ou ajouter du Javascript dans votre code HTML.

Chapitre VIII

Exercice 05

	Exercice : 05
Exercice 05 : Validation W3C.	
Dossier de rendu : <i>ex05/</i>	
Fichiers à rendre : Votre index.html corrigé.	
Fonctions Autorisées :	

Du code c'est bien, du beau code c'est mieux, et pour faire du beau code, le mieux est de suivre une belle norme.

La **norme W3C** est un incontournable du domaine, et il est impératif de la respecter lorsque vous écrivez ou que vous générez du **HTML**.

Vous trouverez dans la tarball **d00.tar.gz** en annexe de ce sujet un sous-dossier **ex05/** qui contient les sources d'une page web complète. Malheureusement, cette page a été écrite par un développeur bien moins bon que vous !

Corrigez le code **HTML** du fichier **index.html** pour qu'il passe la [validation du W3C](#) ! Cela signifie donc aucune erreur et aucun *warning*.

Vous devez *corriger* le fichier et non le tronquer. Cela signifie que le contenu du fichier à corriger *doit* être présent dans votre rendu dans son intégralité.

Chapitre IX

Rendu et peer-évaluation

Rendez votre travail dans votre dépôt `Git` comme d'habitude. Seul le travail présent dans votre dépôt sera évalué en soutenance. Vérifiez bien les noms de vos dossiers et de vos fichiers afin que ces derniers soient conformes aux demandes du sujet.



L'évaluation se déroulera sur l'ordinateur du groupe évalué.