



D00 - Piscine Python-Django

English version coming soon !

*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.*

Table des matières

I	Préambule	2
II	Consignes	3
III	Exercice 00	5
IV	Exercice 01	6
V	Exercice 02	7
VI	Exercice 03	9
VII	Exercice 04	10
VIII	Exercice 05	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


Consignes

- Seule cette page servira de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- Si aucune information contraire n'est explicitement présente, vous devez partir du principe que les versions des langages et framework utilisés sont les suivantes (ou ultérieures) :
 - Python 3
 - HTML5
 - CSS 3
 - Javascript ES6
 - Django 1.9
 - psycopg2 2.6
- Sauf indication contraire dans le sujet, les fichiers en python de chaque exercice sur Python seul (d01, d02 et d03) doivent comporter à leur fin un bloc `if __name__ == '__main__':` afin d'y insérer le point d'entrée dans le cas d'un programme, ou des tests dans le cas d'un module.
- Sauf indication contraire dans le sujet, chaque exercice des journées portant sur Django aura sa propre application dans le projet à rendre pour des raisons pédagogiques.
- Les exercices sont très précisément ordonnés du plus simple au plus complexe. En aucun cas nous ne porterons attention ni ne prendrons en compte un exercice complexe si un exercice plus simple n'est pas parfaitement réussi.
- Attention aux droits de vos fichiers et de vos répertoires.
- Vous devez suivre la procédure de rendu pour tous vos exercices : seul le travail présent sur votre dépôt GIT sera évalué en soutenance.
- Vos exercices seront évalués par vos camarades de piscine.
- Vous ne devez laisser dans votre répertoire aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices.
- Sauf indication contraire dans le sujet vous ne devez pas inclure dans votre rendu :

- Les dossiers `__pycache__`.
- Les éventuelles migrations.
Attention, il vous est tout de même conseillé de rendre le fichier `migrations/__init__.py`, il n'est pas nécessaire mais simplifie la construction des migrations.
Ne pas ajouter ce fichier n'invalidera pas votre rendu mais vous *devez* être capables de gérer vos migrations en correction dans ce cas.
- Le dossier créé par la commande `collectstatic` de `manage.py` (avec pour chemin la valeur de la variable `STATIC_ROOT`).
- Les fichier en bytecode Python (Les fichiers avec une extension en `.pyc`).
- Les fichiers de base de donnée (notamment avec `sqlite`).
- Tout fichier ou dossier devant ou pouvant être créé par le comportement normal du travail rendu.
Il vous est recommandé de modifier votre `.gitignore` afin d'éviter les accidents.
- Lorsque vous devez obtenir une sortie précise dans vos programmes, il est bien entendu interdit d'afficher une sortie précalculée au lieu de réaliser l'exercice correctement.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Votre manuel de référence s'appelle `Google / man / Internet /`
- Pensez à discuter sur le forum Piscine de votre Intra !
- Lisez attentivement les exemples. Ils pourraient bien requérir des choses qui ne sont pas autrement précisées dans le sujet...
- Par pitié, par Thor et par Odin ! Réfléchissez nom d'une pipe !

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 :

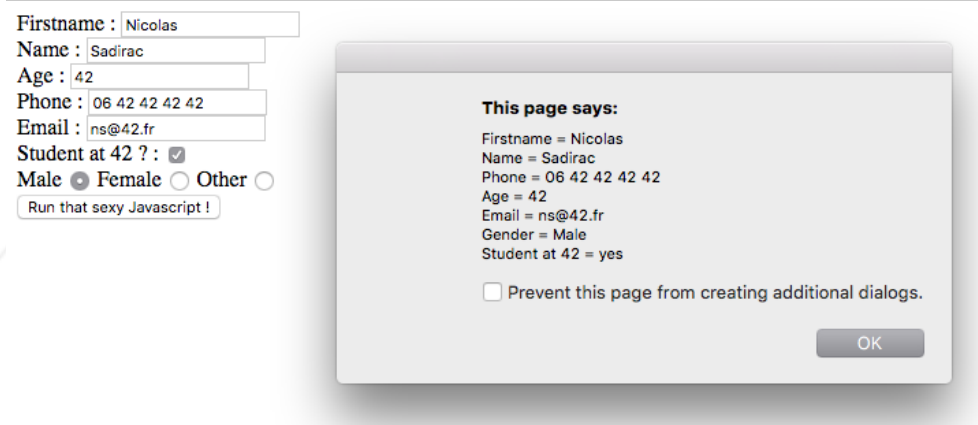
- `Firstname` : un champ texte.
- `Name` : un champ texte aussi.
- `Age` : vous devez utiliser le champ numérique spécifique au HTML5.
- `Phone` : vous devez utiliser le champ `tel` spécifique au HTML5.
- `Email` : vous devez utiliser le champ email spécifique au HTML5.
- `Student at 42 ?` : vous devez 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.



The image shows a web form on the left and a JavaScript alert dialog on the right. The form contains the following fields and values:

- Firstname : Nicolas
- Name : Sadirac
- Age : 42
- Phone : 06 42 42 42 42
- Email : ns@42.fr
- Student at 42 ? : ☒
- Gender : ☒ Male ☐ Female ☐ Other
- Run that sexy Javascript !

The alert dialog on the right has the title "This page says:" and contains the following text:


```
Firstname = Nicolas
Name = Sadirac
Phone = 06 42 42 42 42
Age = 42
Email = ns@42.fr
Gender = Male
Student at 42 = yes
```

Below the text is a checkbox labeled "Prevent this page from creating additional dialogs." and an "OK" button.

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 !

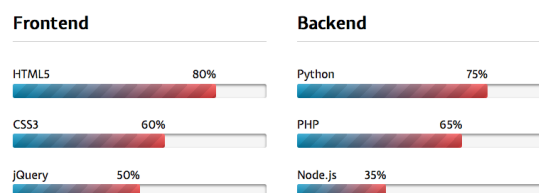



FIGURE VI.1 – Le screenshot de la page à reproduire. Echelle de l'image non contractuelle.

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.