



# D00 - Piscine Python-Django

## Bases du Web

*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

<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>Consignes</b>	<b>3</b>
<b>III</b>	<b>Exercice 00</b>	<b>4</b>
<b>IV</b>	<b>Exercice 01</b>	<b>5</b>
<b>V</b>	<b>Exercice 02</b>	<b>6</b>
<b>VI</b>	<b>Exercice 03</b>	<b>8</b>
<b>VII</b>	<b>Exercice 04</b>	<b>9</b>
<b>VIII</b>	<b>Exercice 05</b>	<b>10</b>

# 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

Sauf contradiction explicite, les consignes suivantes seront valables pour tous les jours de cette Piscine.

- Seul ce sujet sert de référence : ne vous fiez pas aux bruits de couloir.
- Le sujet peut changer jusqu'à une heure avant le rendu.
- 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. L'url de votre dépôt GIT pour cette journée est disponible sur votre intranet.
- Vos exercices seront évalués par vos camarades de Piscine.
- En plus de vos camarades, vous pouvez être évalués par un programme appelé la Moulinette. La Moulinette est très stricte dans sa notation car elle est totalement automatisée. Il est donc impossible de discuter de sa note avec elle. Soyez d'une rigueur irréprochable pour éviter les mauvaises surprises.
- Les exercices shell doivent s'exécuter avec `/bin/sh`.
- Vous ne devez laisser aucun autre fichier que ceux explicitement spécifiés par les énoncés des exercices dans votre dépôt de rendu.
- Vous avez une question ? Demandez à votre voisin de droite. Sinon, essayez avec votre voisin de gauche.
- Toutes les réponses à vos questions techniques se trouvent dans les `man` ou sur Internet.
- Pensez à discuter sur le forum Piscine de votre Intra et sur Slack !
- Lisez attentivement les exemples car ils peuvent vous permettre d'identifier un travail à réaliser qui n'est pas précisé dans le sujet à première vue.
- Réfléchissez. Par pitié, par Thor, par Odin !

# 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](https://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](https://bit.ly) supposé valide (comprenez "L'adresse vers laquelle renvoie le lien [bit.ly](https://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.

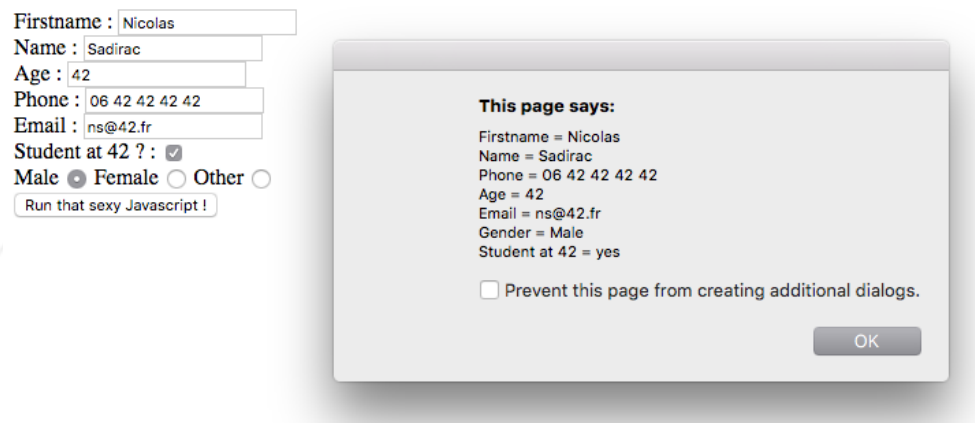



FIGURE V.1 – Illustration non contractuelle du résultat attendu.



# 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 !

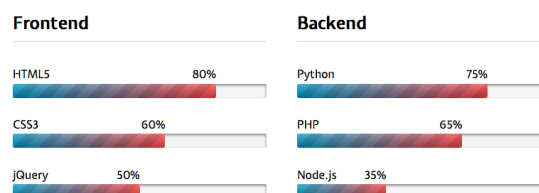



FIGURE VI.1 – Le screenshot de la page à reproduire. Echelle de l'image non contractuelle.

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 : <b>snippets.html</b>	
Fonctions Autorisées : <b>n/a</b>	

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 : <b>Votre index.html corrigé.</b>	
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.