

## 1. Objectifs du TP

- Apprendre à faire une page web sans mise en forme avec HTML5 ;
- Vérifier et corriger la syntaxe d'une page HTML5 ;
- Connaître les principales balises HTML5 et leurs utilisations ;
- Découvrir Visual Studio Code et les "aides de codage" qu'il propose pour HTML5 ;

Ce sujet est consultable en version PDF [sur Chamilo](#).

## 2. Documentation conseillée

- Le cours, disponible sur la plate-forme Chamilo
- Les [tutoriaux HTML5](#) sur le site w3schools.
- La [liste des éléments HTML5](#) sur le site w3schools.
- La [documentation de Mozilla](#) en français

## 3. Préparation et introduction

### À faire

1. Créer un répertoire **r1\_02** dans votre répertoire personnel ;
2. Créer un répertoire **TP1** dans le répertoire **r1\_02** ;
3. Se positionner dans ce répertoire **TP1** ;
4. Recopier dans votre répertoire **TP1** le contenu de [l'archive](#).

On rappelle le squelette d'une page HTML5 :

```
1  <!DOCTYPE html>
2  <html lang="fr">
3    <head>
4      <meta charset="UTF-8">
5      <title>Titre de ma page</title>
6      <link rel="stylesheet" href="style.css">
7    </head>
8    <body>
9      <!--
10     Tout ce qui sera visible à l'écran de
11     l'utilisateur qui visite la page.
12     -->
13  </body>
14 </html>
```

## 4. Validation

La validation d'une page web est importante. Cela permet d'être sûr que la structure de la page est correcte et qu'il n'y a pas de faute, c'est-à-dire que la syntaxe du langage HTML5 est bien respectée. Nous allons vérifier que la page web *page1.html* est valide. Pour cela, nous utiliserons le [validateur HTML5 nu](#).

## À faire

- Extraire le contenu de l'archive (5 fichiers: *page1.html*, *page2.html*, *viaferrata-bastille.jpg*, *etudiut.jpg* et *etudiut-90-103.jpg*).
- Visualiser dans le navigateur Firefox le fichier *page1.html*. Différentes méthodes possibles :
  - Dans un terminal, saisir la commande `firefox page1.html &`
  - Depuis l'explorateur de fichiers, commande File/Open With/Firefox.
  - Depuis l'explorateur de fichiers, en cliquant-glissant le fichier *page1.html* dans Firefox.
- Ouvrir le validateur HTML5 *nu* accessible à l'adresse <https://validator.w3.org/nu/> ;
- Sélectionner le mode *File upload* ;
- Importer le fichier *page1.html*.

**La page Web *page1.html* est-elle valide ?**

Nous allons maintenant nous intéresser à la page Web *page2.html*

## À faire

- Visualiser dans le navigateur Firefox le fichier *page2.html*
- Réitérer les étapes de validation précédentes avec le fichier *page2.html*.

**La page Web *page2.html* est-elle valide ?**

## À faire

- Ouvrir sous l'éditeur *Visual Studio Code* la page Web *page2.html*. Pour cela :
  - Ouvrir un terminal et saisir la commande `code` ;
  - Ouvrir l'explorateur de fichier en cliquant sur l'icône en haut à gauche, ou bien en utilisant le raccourci clavier `Ctrl + Shift + E` ;
  - Ouvrir un répertoire en cliquant sur le bouton *Open Folder*. Par simplicité, ouvrir le répertoire **r1\_02** qui contiendra tous les TP. ainsi, cette étape n'aura besoin d'être faite qu'une seule fois ;
  - Un bandeau apparaît pour indiquer que le répertoire n'est pas un répertoire de confiance. Ajouter votre répertoire **r1\_02** parmi les "Trusted Folders" en cliquant sur l'icône en forme de roue crantée en bas à gauche, puis cliquer sur "Manage Workspace Trust" ;  
**Note** : cette étape n'est pas obligatoire pour cet exercice, mais le deviendra plus tard, donc il est préférable de la faire tout de suite.
- Corriger jusqu'à ce que la page soit valide.

Il peut devenir fastidieux de recourir au validateur en ligne pour corriger une page Web. Heureusement, les éditeurs de code tels que Visual Studio Code deviennent de plus en plus performants et intègrent des extensions pour vérifier en temps-réel la syntaxe de vos pages.

## À faire

- Dans Visual Studio Code, ouvrir le panneau des extensions avec le raccourci clavier `Ctrl + Shift + X` ;
- Rechercher l'extension *W3C Validation* et l'installer ;
- Fermer *Visual Studio Code* et relancer-le ;
- Retourner sur le fichier *page2.html* et introduire volontairement une erreur de syntaxe ;
- **Important** : survoler l'erreur et lire le message d'erreur associé !

## ⚠ Attention

Les extension de Visual Studio Code sont désactivées si le répertoire n'est pas un répertoire de confiance. Assurez-vous d'avoir bien réalisé les étapes précédentes pour avoir indiqué que le répertoire **r1\_02** est un répertoire de confiance.

L'extension permet également de remonter les erreurs de syntaxe dans le panneau *Problems*. Pour l'ouvrir, cliquer dans la barre principale sur *View* puis *Problems* ou utiliser le raccourci clavier **Ctrl** + **Shift** + **M**.

## 5. Page Web à faire

Nous allons rédiger un **curriculum vitae** (CV) dans les technologies Web. Vous devez faire votre propre CV, ou celui de l'étudiant "idéal" du Département Informatique de l'IUT2 ! (suivre dans ce cas l'image ci-dessous).

## ⚠ Attention

**Uniquement de l'HTML5** (aucune mise en forme, aucun CSS pour ce TP).

Quelques précisions sur l'outil Visual Studio Code :

1. Votre fichier doit absolument avoir l'extension **.html**, pour que l'éditeur puisse "savoir" que vous faites de l'HTML et se configure correctement.
2. Vous profiterez des possibilités qu'offrent Visual Studio Code en saisissant directement les éléments du langage, et non pas les balises. Par exemple:
  - **html** + **enter** (et non pas **<html>**)
  - **article** + **enter** (et non pas **<article>**)
  - **ul** + **enter** (et non pas **<ul>**)
  - ...

## ✍ À faire

- Mettre en place le squelette de la page Web appelée **cv.html** ;
- Réaliser le CV en suivant les indications ci-dessous.

Pour vous aider à réaliser votre CV, une maquette de type "zoning" est fournie plus bas. Il s'agit d'une maquette grossière découpant les zones importantes du CV. Elle est généralement réalisée en amont de la réalisation et se fait directement à l'aide de crayons et papiers.

Votre CV comportera 7 divisions (à positionner dans la partie **<body> ... </body>**):

- Une première division (**header**) regroupant :
  - un titre **h1**
  - une photo
  - un paragraphe avec des informations générales (âge, adresse, téléphone,...)
- Une division principale (**main**) contenant elle-même cinq divisions (**article**) qui seront structurées toutes de la même façon, avec :
  - un titre **h2**
  - des informations (voir ci-dessous les différents éléments HTML à utiliser obligatoirement)
- Une dernière division (**footer**) comprenant, un lien interne vers le haut de la page, la date de mise à jour de la page.

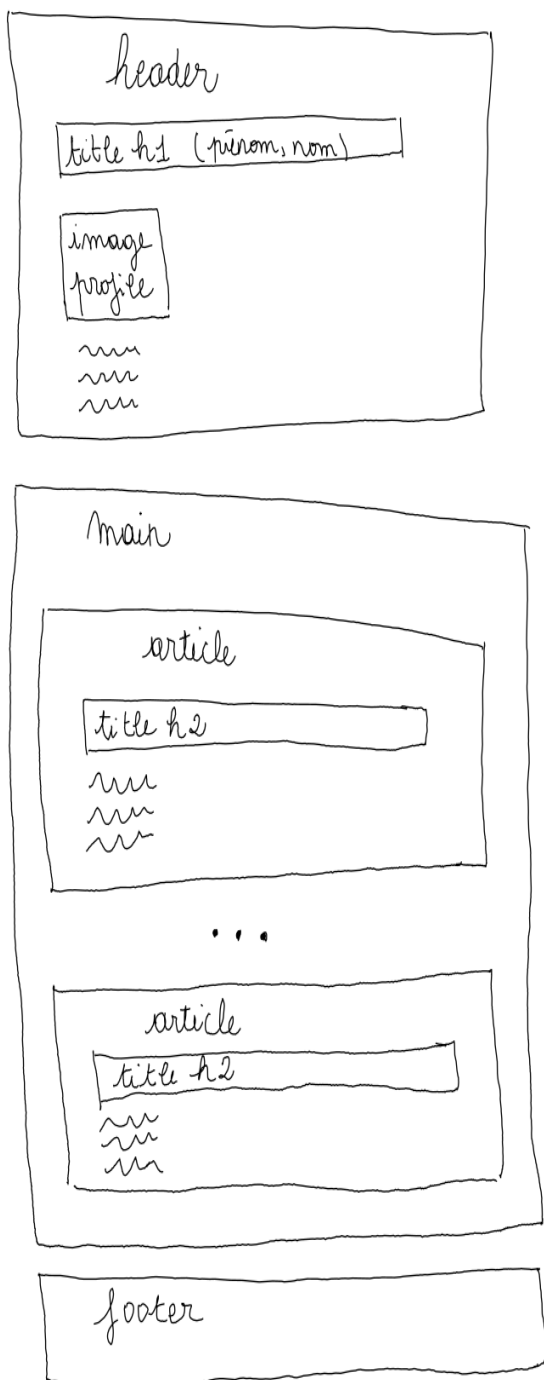
Les cinq divisions **article** seront :

1. Formation:  
Cette division vous permet de présenter votre formation.  
Vous utiliserez **des listes non numérotées**.  
Vous ferez également **des liens externes** qui renvoient sur les sites des écoles/universités que vous avez fréquentées.
2. Compétences:  
Cette division présente vos compétences informatiques.  
Vous utiliserez **une liste de description**.
3. Expériences professionnelles:  
Cette division présente vos expériences professionnelles des plus récentes aux plus anciennes.  
Vous utiliserez **une liste numérotée**.
4. Langues:  
Cette division présente vos compétences linguistiques.  
Vous utiliserez **un tableau**.
5. Centres d'intérêt:  
Cette division présente vos centres d'intérêt.  
Vous utiliserez les éléments de votre choix.

Pour marquer le changement de sujet entre chacune des 7 divisions, vous utiliserez la balise <hr>.

### Attention

Vérifier régulièrement la validité de votre page Web grâce au validateur (en ligne ou embarqué dans Visual Studio Code).



Maquette "zoning" du CV.

## 6. Version 2 du CV

Nous nous intéressons à une seconde version du CV, qui modifie la liste du premier article. On veut cette fois des listes imbriquées. (Voir la figure ci-dessous).

## CV de Pierre MICHEL - Développeur Web



23 ans

320 avenue des Fleurs - 38000 Grenoble

+33 (0)6 00 11 22 33  
pierre.michel@gmail.com  
<http://www.monsite.fr>

### Formation

- 2016-2017  
[Licence professionnelle Métiers Informatiques](#)
- 2015-2016  
[DUT - 2<sup>ème</sup> année](#)
- 2014-2015  
[DUT - 1<sup>ère</sup> année](#)
- 2012-2015  
[Lycée du Grésivaudan](#)

### Compétences

Développement web:  
HTML5 - CSS - PHP - Javascript  
Respect des standards:  
W3C - WCAG (Accessibilité Handicap)  
Base de données:  
SQL - Postgres  
Développement logiciel:  
ADA - C - C++ - JAVA - Python

### Expériences professionnelles

1. **avril 2016 - juin 2016**  
Stage de développement web dans l'entreprise Dracula (Paris) - utilisation du CMS Drupal
2. **juillet 2015 - août 2015**  
CDD de plongeur - Restaurant "Le marmillon" (Annecy)

### Langues

Voici les langues que je pratique:

	Notions Technique Courant		
Anglais	-	-	X
Espagnol	X	-	-
Italien	X	-	-

### Centres d'intérêt

#### Sports

Compétiteur ski alpin depuis 2010; participation annuelle au championnat de France des ESF

#### Vie associative

Je participe activement, en tant que bénévole, à l'association [Handi'Chiens](#)

Date de Mise à jour : novembre 2017 - [début du CV](#)

Exemple de CV de l'étudiant idéal.

## Formation

- 2016-2017  
[Licence professionnelle Métiers Informatiques](#)
  - Grenoble
  - Isère
- 2015-2016  
[DUT](#) - 2<sup>ème</sup> année
  - Grenoble
  - Isère
- 2014-2015  
[DUT](#) - 1<sup>ère</sup> année
  - Grenoble
  - Isère
- 2012-2015  
[Lycée du Grésivaudan](#)
  - Meylan
  - Isère

Variante du CV avec des listes imbriquées.

### À faire

- Créer un fichier *cv2.html* à partir du contenu du fichier *cv.html* ;
- Modifier la section *Formation* avec des listes imbriquées ;
- Valider la page Web.

### Attention

Lire l'article [Mes listes UL, OL, ou listes imbriquées ne sont pas valides](#) du site Alsacrations en cas de difficulté à construire une liste imbriquée valide.

## 7. Nouvelle page

On vous demande de faire une nouvelle page *pageLibre.html* (toujours sans CSS!), thème libre, par exemple: votre sport ou sportif préféré, votre région, votre ville,... .

Outre les éléments classiques vus précédemment, vous utiliserez des éléments que vous n'avez pas encore testés:

- details, summary
- em, strong
- sub, sup
- kbd, code, pre
- q, blockquote, cite, del, ins
- figure, img
- audio, vidéo
- utilisez également des caractères UTF-8 spéciaux comme € ou ♣ ou ↔ ou encore ☺ (voir [UTF-8 General Punctuation](#)).
- dans un lien, ajoutez un attribut 'target' pour que la page cible s'ouvre dans une fenêtre différente

Vous pouvez vous référer à la [liste des éléments HTML5](#) sur le site w3schools.

## 8. En bonus

- Vérifier que vos pages sont valides ;
- Lire l'article [HTML5 se dévoile](#) sur le site Alsacrations ;
- Etudier les [exemples](#) sur le site w3schools ;
- Faire des [exercices](#) proposés par le site w3schools ;
- Et enfin, consulter la dernière documentation du [HTML Living Standard](#).