



## Module Web 01

### Un CV de <geek/>



#### Table des matières

Les compétences visées.....	2
Le <i>versioning</i> avec GIT : les commandes de base.....	3
TD 1 - Le langage HTML et la structure de base d'un fichier.....	4
Défi 1 - La structure de base d'une page HTML.....	5
TD 2 - CSS ou comment donner du style à une page HTML.....	6
Défi 2 - Un look de <geek/> à votre page web.....	8
TD 3 - Les sélecteurs en CSS.....	10
Défi 3 - Votre CV de <geek/>.....	11

## Les compétences visées

</>	Catégorie	Je suis capable de :	
<b>H T M L</b>	Fichier HTML et sa structure	Créer et enregistrer un fichier HTML	
		Comprendre le rôle des balises <html>, <head> et <body>	
		Comprendre l'utilité du DOCTYPE	
	Les fondamentaux des balises	Ajouter une balise ouvrante et fermante	
		Ajouter une balise auto-fermante	
		Ajouter un attribut id et sa valeur	
		Ajouter un attribut class et sa valeur	
	Les balises du corps <body>	Ajouter une entête <header>	
		Ajouter une barre de navigation <nav>	
		Ajouter une section <section> principale <main>	
		Ajouter une section non-principale <aside>	
		Ajouter un pied de page <footer>	
	Les balises de texte	Ajouter des titres de niveau x <hx>	
		Ajouter des paragraphes <p>	
		Ajouter une liste numérotée <ol> et ses éléments <li>	
		Ajouter une liste non-numérotée <ul> et ses éléments <li>	
		Ajouter un tableau <table>, une ligne <tr> et une cellule <td>	
		Ajouter un lien <a>	
	La balise image	Ajouter une image <img>	
<b>C S S</b>	Fichier CSS et sa structure	Créer et enregistrer un fichier CSS	
		Ajouter un nouveau sélecteur et ses délimiteurs { }	
		Ajouter une nouvelle propriété dans un sélecteur	
		Ajouter une valeur à une propriété.	
	Les sélecteurs	Ajouter un sélecteur de class avec un '.'	
		Ajouter un sélecteur id avec un '#'	
		Comprendre l'imbrication des sélecteurs	
	Les boîtes	Connaître les éléments constituant une boîte : les marges intérieures et extérieures, la bordure et le contenu.	
		Ajouter une marge extérieure (margin) dans une boîte	
		Ajouter une bordure (border) dans une boîte	
		Ajouter une marge intérieure (padding) dans une boîte.	
<b>G I T</b>	Le dépôt local	Créer un nouveau dépôt local (git init)	
		Ajouter des fichiers dans le dépôt local (git add)	
		Sauvegarder les modifications dans le dépôts (git commit)	
	Le dépôt distant	Créer un dépôt distant et se synchroniser avec (git remote)	
		Publier un dépôt local dans un dépôt distant (git push)	

## Le versioning avec GIT : les commandes de base

### Le dépôt local

Git est un outil de versionning très populaire. Il s'utilise principalement en ligne de commande. Il existe néanmoins une version Bureau pour Windows : Git GUI.

#### Configurer un nouveau dépôt

Dans votre répertoire de travail, créer un nouveau dépôt GIT avec la commande :

```
git init
```

Pour le moment, ce dépôt est vide, c'est-à-dire qu'il n'y a encore aucun fichier dont on pourra suivre les versions.

#### Ajouter des fichiers dans le dépôt

Pour ajouter un fichier dans le dépôt (et donc commencer à en suivre ses versions), taper la commande :

```
git add index.html
```

Cette commande ajoute le fichier index.html dans le dépôt. N'oublier pas d'ajouter dans le dépôt Git tous vos nouveaux fichiers !

#### Enregistrer les modifications des fichiers dans le dépôt

Quand le code se trouve dans une situation stable, il est approprié de faire une sauvegarde dans le dépôt ainsi :

```
git commit -m 'mon premier code HTML'
```

Toutes les modifications des fichiers qui se trouvent dans le dépôt sont alors enregistrées. Il sera désormais possible de revenir à cette version ultérieurement si nécessaire !

#### Connaître l'état du dépôt

Il est possible de connaître l'état du dépôt en tapant la commande :

```
git status
```

Cette commande vous indique notamment

- si des fichiers présents dans votre répertoire de travail ne sont pas présents dans le dépôt git,
- si des fichiers ont été modifiés mais pas encore enregistrer avec un commit, etc...

### Le dépôt distant

GitHub est un des dépôts distants les plus connus et c'est celui que nous utiliserons. L'avantage est que votre code est alors disponible de n'importe où, que vous soyez au travail, à l'école ou chez vous !



#### Créer un compte GitHub

La première chose à faire est de créer un compte GitHub ! A vous de jouer : <https://github.com/>

#### Créer un dépôt sur GitHub

Avec l'interface graphique de GitHub, laissez-vous guider pour créer un nouveau dépôt. Vous pouvez lui donner le même nom que votre répertoire de travail local.

#### Connecter le dépôt local au dépôt GitHub

Pour connecter votre dépôt Git local au dépôt de GitHub, taper la commande suivante :

```
git remote add origin <github-repository-url>
```

#### Publier les changements de votre dépôt local sur GitHub

Pour envoyer votre dépôt git local sur GitHub, taper la commande :

```
git push origin master
```

#### Récupérer la dernière version du dépôt GitHub

Pour récupérer la dernière version du dépôt sur GitHub, taper la commande suivante :

```
git clone <github-repository-url>
```



Figure 1: Git GUI pour Windows

## TD 1 - Le langage HTML et la structure de base d'un fichier

### Le langage HTML

Quelques sites de référence pour cette partie :

[https://fr.wikipedia.org/wiki/Hypertext\\_Markup\\_Language](https://fr.wikipedia.org/wiki/Hypertext_Markup_Language)

[https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises\\_HTML](https://developer.mozilla.org/fr/docs/Apprendre/HTML/Balises_HTML)

? Donner la définition du sigle HTML.

? Expliquer l'utilité de ce langage

? Le HTML est un langage compilé ou interprété ? Et par qui ?

### La structure de base d'un fichier HTML

La structure de base d'un fichier HTML est la suivante :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>BTS SNIR - Dev WEB</title>
  </head>
  <body>
  </body>
</html>
```

### Anatomie d'une balise

D'une façon générale, une balise HTML possède le format suivant :

Diagram illustrating the structure of an HTML tag:

```

  Balise ouvrante      Balise fermante
  <p class="paragraphe">Mon CV</p>
  Un attribut et sa valeur (facultatif)  Le texte
  
```

? Dans le code donné ci-dessus, donner la liste des balises possédant une balise ouvrante et une balise fermante.

Certaines balises ne respectent pas ce format. Elles sont appelées les balises auto-fermantes parce qu'elle ne possède de balises fermantes.

? Dans le code donné ci-dessus, donner la liste des balises auto-fermantes :

### Le DOCTYPE

? Expliquez l'intérêt du DOCTYPE ([https://www.w3schools.com/tags/tag\\_doctype.asp](https://www.w3schools.com/tags/tag_doctype.asp))

### La signification des balises.

? Donner la signification de la balise <html>

? Donner la signification de la balise <title>

? Donner la signification de la balise <head>


? Donner la signification de la balise <body>

? Donner la signification de la balise <meta>

# Défi 1 - La structure de base d'une page HTML

## Test de la structure de base d'une page HTML

L'objectif de cette première partie sera de tester la page HTML décrite dans le TD précédent.

- </> Créer un répertoire de travail pour MW01.
- </> Créer un fichier « index.html » dans ce répertoire et éditer ce fichier avec un éditeur (Notepad++, Atom, Sublim Text, etc.).
- </> Retaper le code HTML de base présenté dans le TD précédent puis sauvegarder.
- </> Tester le code du fichier avec un navigateur : clic-droit sur le fichier, ouvrir avec un navigateur web (firefox, chrome, etc.) : vous devriez obtenir une page blanche avec un titre dans l'onglet.
- </> Modifier le titre de votre page dans le fichier HTML. Mettre « Mon CV de geek ».
- </> Sauvegarder et recharger la page dans votre navigateur. Vérifier que le titre de votre page a bien changé.
- </> Sauvegarder de l'état actuel de votre projet avec  **git** !



## Ajout des balises sémantiques dans le body

Dans cette partie, nous allons travailler sur le corps de la page. Pour le moment, le corps (c'est-à-dire le contenu de la balise <body>) est vide, nous allons maintenant le compléter avec :

1. **une entête <header>** contenant le titre du site
2. **une barre de navigation <nav>**. Elle contiendra les liens vers les pages de votre site que vous développerez au cours de l'année,
3. **une section <section>** contenant le **contenu principal** de votre site – votre CV dans ce module
4. **une autre section <aside>** contenant des liens vers d'autres sites web de référence pour le développement web
5. **un pied de page <footer>**.

La numérotation utilisée ci-dessus est reportée dans l'illustration ci-contre.

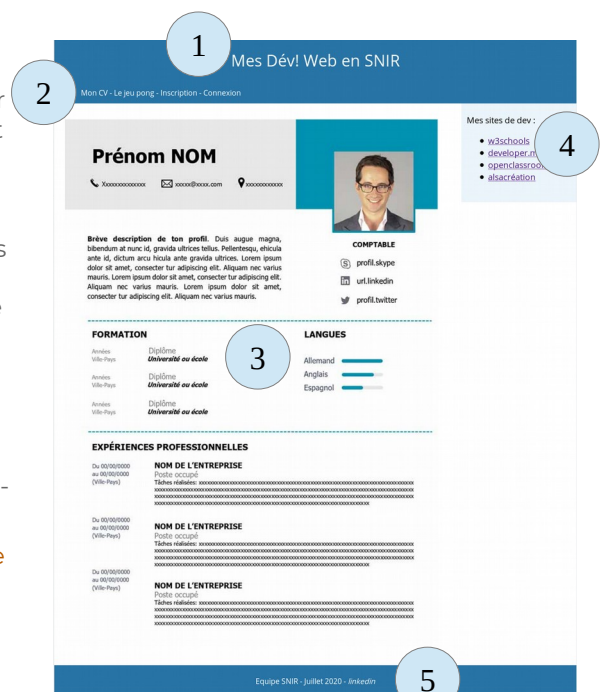
- </> En respectant la mise en page du code, ajouter dans le body de votre page :

- 1) une balise <header> ouvrante et fermante. Mettre le texte « L'entête de mon site » dans cette balise.
- 2) une balise <nav> ouvrante et fermante. Mettre le texte « Ma barre de navigation » dans cette balise.
- 3) une balise <aside> ouvrante et fermante. Mettre le texte « Mes liens » dans cette balise.
- 4) une balise <section> ouvrante et fermante. Mettre le texte « Mon CV » dans cette balise.
- 5) une balise <footer> ouvrante et fermante. Mettre le texte « Mon pied de page » dans cette balise.

- </> Sauvegarder et recharger la page dans votre navigateur. Vérifier que les 5 textes apparaissent les uns en dessous des autres. *Remarque : pour le moment, il s'agit juste de simples textes qui apparaissent... et c'est normal. Nous verrons comment les « styliser » par la suite...*

- </> Sauvegarder de l'état actuel de votre projet avec  **git** en saisissant la commande :

```
git commit -m "Defi 1 - structure HTML5"
```



## TD 2 - CSS ou comment donner du style à une page HTML

### Le langage CSS

? Donner la définition du sigle CSS.

? Expliquer l'utilité de ce langage

? Le CSS est un langage compilé ou interprété ? Et par qui ?

### Les éléments d'un fichier CSS

En CSS, les éléments sont décrits selon cette syntaxe :

```
sélecteur {
  propriété: valeur; /* Un commentaire */
}
```

Voici maintenant un exemple concret :

```
header {
  border: 1px solid grey;
  background-color: #e5e5e5;
}
```

? Dans l'exemple, donner le nom du sélecteur.

? Décrire l'effet des propriétés et de leurs valeurs

### Les boîtes en CSS

Tous les éléments HTML sont considérés comme des boîtes. Comme le montre l'illustration ci-contre, ces boîtes possèdent :

- Content ou contenu : elle contient le texte et les images
- Padding ou marge intérieure : elle se trouve autour du contenu dégageant un espace entre la bordure et le contenu de la boîte.
- Border ou bordure : elle fait le tour du padding et du contenu.
- Margin ou marge extérieure : elle se trouve autour de la bordure.

Les éléments margin, padding et border sont aussi des propriétés CSS. Par exemple, une marge de 10 pixels autour d'une boîte s'écrit :

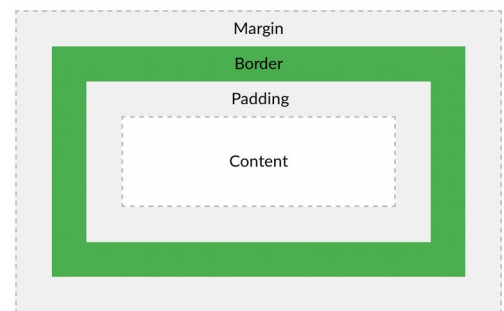
```
margin: 10px;
```

Maintenant, il est possible de faire des marges différentes pour le haut, le bas, la droite et la gauche comme ceci :

```
margin: 10px 25px 20px 15px; /* suivant l'ordre : haut droite bas gauche */
```

Il est aussi possible d'utiliser les propriétés margin-top, margin-right, margin-bottom et margin-left. Tout ce qui a été dit sur la propriété margin s'applique également à la propriété padding !

? On souhaite styliser le header comme suit : une marge extérieure de 0px et une marge intérieure en haut et en bas de 20 px et une couleur de fond valant #2874a6 (un bleu). Donner le code CSS correspondant :



### Les polices de caractères, leurs tailles et leurs couleurs

Pour donner du style à une page web, rien de tel que de travailler la police de caractères, la taille, les couleurs de polices et de fond. Voici quelques clés pour démarrer !

Pour qu'une police de caractères soit appliquée sur l'ensemble de la page, la définir dans le body. Par exemple :

```
body {
  font-family: Arial, Helvetica, sans-serif;
}
```

Les navigateurs web afficheront la page avec la police Arial. Si elle n'est disponible, ce sera Helvetica et sinon la famille sans-serif.

Après avoir déclaré une police de caractères, il est possible de la personnaliser. Par exemple :

```
section h1 {  
  text-align: left; /* valeurs possibles : left | right | center | justify */  
  color: blue;  
  font-size: 2em; /* 1em : taille par défaut de la police de caractères défini sur le navigateur */  
  font-style: italic; /* valeurs possibles : normal | italic */  
  font-weight: bold; /* valeurs possibles : normal | bold */  
  text-decoration: underline; /* valeurs possibles : none | underline | overline | line-through | blink */  
  text-transform: capitalize; /*text-transform: none | capitalize | uppercase | lowercase */  
  line-height: 200%; /*valeurs possibles : normal | nombre | % */  
}
```



On souhaite styliser le titre h1 du header comme suit : une police 3 fois plus haute qu'un caractère normal, blanche, en gras et centrée. Donner le code CSS correspondant :

## Défi 2 - Un look de <geek/> à votre page web

### Créer le fichier CSS

**</>** Créer un fichier nommé `style.css`, dans le même répertoire que votre fichier `index.html` et y copier le code suivant :

```
/* Rendu des éléments HTML5 en bloc */
body,header,footer,section,aside,nav {
  display: block;           /* Non nécessaire, valeur par défaut */
  border: 1px solid grey;    /* Une bordure pour 'voir' comment les blocs s'organisent */
  margin: 0;                /* Aucune marge extérieure */
  padding: 0;               /* Aucune marge intérieure */
}
```

À la ligne 2 se trouve une liste de balise HTML qui sont concernées par le style indiqué ensuite entre les accolades. Toutes ces balises HTML auront leur propriété `display` valant `block`. Un bloc prend toute la largeur de la page et sa hauteur dépend de son contenu. Les blocs se trouvent alors forcément les uns sous les autres.

### Lier le fichier HTML et le fichier CSS

Pour que ces styles soient appliqués aux balises au fichier « `index.html` » du 1<sup>er</sup> défi, il faut le lier avec la balise `<link>` au fichier CSS comme suit :

```
<link rel="stylesheet" type="text/css" href="style.css">
```

**</>** Ajouter la balise auto-fermante `<link>` ci-dessus à la fin du `<head>` dans votre fichier « `index.html` ».

Enregistrer les fichiers « `index.html` » et « `style.css` » puis rafraîchir la page dans votre navigateur.

**</>** *Remarque :* Votre page devrait avoir changer d'aspect. Si ce n'est pas le cas, vérifier : 1) que les fichiers `index.html` et `style.css` soient bien dans le même répertoire, 2) que la propriété `href` de la balise `link` de votre fichier HTML contient bien le nom du fichier CSS que vous avez créé.

### Modification des valeurs dans le fichier CSS

**</>** Modifier la bordure en rouge avec 3 pixels d'épaisseur, faire une marge extérieure haute et basse de 20 pixels et une marge intérieure à gauche de 50 pixels. Faire valider par votre professeur.

**</>** Sauvegarder de l'état actuel de votre projet avec  **git** en saisissant la commande :

```
git commit -m "Defi 2 - style CSS débutant"
```

### A vous de jouer !

Votre but est maintenant de faire quelque chose de beau, de rendre votre page belle, qui donne envie d'être consultée... Et pour rien ne vous cacher, ce n'est pas une mince affaire. Alors voici le cahier des charges que nous vous imposons avec une esquisse du résultat final :



En suivant les consignes ci-dessous, modifier la feuille de style pour obtenir le résultat suivant :

Le `<header>` aura les propriétés suivantes :

- une couleur de fond valant #2874a6,
- une marge intérieure en haut et en bas de 20 pixels,
- un alignement du texte au centre,
- une hauteur de police de 30 pixels,
- une couleur de police blanche.

Le `<footer>` aura les propriétés suivantes :

- une couleur de fond valant #2874a6,
- une marge intérieure en haut et en bas de 20 pixels,
- un alignement du texte au centre,
- une hauteur de police de 14 pixels,
- une couleur de police blanche.

Le `<aside>` aura les propriétés suivantes :

- une largeur de 20%
- une couleur de fond valant #ebf5fb,
- une marge intérieure en haut et en bas de 20 pixels et une marge intérieure gauche et droite de 10 pixels,
- Une position flottante à droite.

Le `<nav>` aura les propriétés suivantes :

- une couleur de fond valant #2874a6,
- une marge intérieure en haut et en bas de 10 pixels et une marge intérieure gauche et droite de 50 pixels.
- une hauteur de police de 14 pixels,
- une couleur de police blanche.

La `<section>` aura les propriétés suivantes :

- une marge intérieure en haut et en bas de 30 pixels et une marge intérieure gauche et droite de 20 pixels.



**</>** Pour placer la section `<aside>` à droite de la section principale, ajouter le code CSS suivant :

```
aside {
  float: right;
  width: 20%;
}
```

**</>** Sauvegarder de l'état actuel de votre projet avec **git** en saisissant la commande :

```
git commit -m "Defi 2 - CSS intermédiaire"
```

## TD 3 - Les sélecteurs en CSS

On peut utiliser indifféremment les attributs id et class pour appliquer des styles CSS aux éléments d'une page et interagir avec eux en JavaScript, mais :

- **un id** s'applique à un objet unique : il ne peut pas y avoir deux mêmes id dans une page
- **une classe** peut caractériser plusieurs objets (identiques ou non)

Prenons 2 exemples pour illustrer ces 2 notions essentielles du CSS.

### Le sélecteur class

Partons d'une page web qui affiche quelques citations d'auteurs. On souhaite que toutes les citations soient centrées et en italiques. On définit alors une classe nommée `citation` (précédé d'un '.') dans le fichier CSS.

style.css :

```
.citation {
  text-align: center;
  font-style: italic;
}
```

index.html :

```
<p>N'oubliez pas que la course n'appartient pas au plus rapide. C'est
ainsi que Jean de la Fontaine dit : </p>
<p class="citation">Rien ne sert de courir, il faut partir à point!</p>
<p>Dans le même ordre d'idée, on trouve cette marquise :</p>
<p class="citation">Même le dernier d'une course est devant ceux qui ne
courent pas.</p>
```

NOTE : Cette mise en forme citation peut être répétée *autant de fois que besoin* dans la page !

**?** Créer le sélecteur class `fond_bleu` avec une couleur de fond valant `#2874a6`, et une couleur de police blanche.

### Le sélecteur id

Supposons que dans votre page web, vous souhaitez mettre en jaune la couleur d'arrière plan de votre nom. Juste pour votre nom. Alors le sélecteur id est approprié !

style.css :

```
#nom {
  background-color: yellow;
}
```

index.html :

```
<p id="nom">DUPONT</p>
```

NOTE : L'id `nom` ne peut être utilisé *qu'une seule fois* dans le fichier source HTML !

Maintenant, supposons que dans notre page web, une image (une photo d'identité) possède des contraintes particulières : elle doit mesurer 50 pixels par 75 pixels, se trouver à droite en position flottante et posséder une marge de 20 pixels tout autour d'elle. Elle seule possède ces contraintes. On décide alors de donner à cette image l'id (un identifiant unique) `photo_identite`.

**?** Créer le sélecteur id `photo_identite` avec les propriétés CSS demandées. Ajouter la balise HTML `img` possédant cet id sur la photo dont le fichier source est « `maPhoto.jpg` ».

style.css :

index.html :

### La liste des sélecteurs CSS

- `X, Y, Z { ... }` : tous les éléments X, Y et Z.
- `X.foo { ... }` : tous les éléments X pour lesquels l'attribut class vaut foo
- `X#foo { ... }` : l'élément X dont l'id vaut foo
- `X Y { ... }` : les éléments Y qui sont à n'importe quel niveau d'imbrication sous un X
- `X > Y { ... }` : les éléments Y qui sont juste au-dessous d'un X
- `X:link { ... }` : un lien non encore visité
- `X:visited { ... }` : un lien déjà visité
- `X:hover { ... }` : élément survolé par le pointeur

## Défi 3 - Votre CV de <geek/>

### Optimisation du CSS

Dans cette première partie, vous allez optimiser le fichier style.css en créant des classes

**</>** Comme proposé dans le défi 2, créer le sélecteur class **fond\_bleu**.

**?** Indiquer le nombre de lignes que vous avez supprimées. En déduire l'intérêt des class et id.

### Le cahier des charges de votre CV

Créez une page Web contenant votre CV. Ce CV doit être rédigé dans le but de rechercher votre stage de fin d'année. Il devra donc contenir :

- A votre sujet :
  - votre nom et prénom,
  - votre adresse postale,
  - votre téléphone,
  - votre adresse mail,
  - une photo (celle du trombinoscope par exemple),
- Le type de stage que vous recherchez, la durée et la date de début de stage
- Vos formations (en commençant par la plus récente, qui est en cours) :
  - l'année d'obtention du diplôme
  - le nom du diplôme
  - le nom et adresse du centre de formation et son adresse
- Vos expériences professionnelles
  - la date de début et la durée
  - la mission confiée
  - le nom de l'entreprise et son adresse
- Vos langues parlées et/ou lues
- Vos compétences en développement (les compétences que vous aurez au moment du stage)
- Vos loisirs

### Les contraintes en HTML :

- Utiliser des titres avec la balise <h> (n est un nombre compris entre 1 et 6) de 2 niveaux différents
- Utiliser la balise <img> pour la photo, en précisant les attributs src et alt.
- au moins 2 différents styles d'écriture (police, gras, italique, couleur),
- un tableau de vos formations avec 3 colonnes : année, diplôme, établissement
- un tableau de vos expériences professionnelles avec 3 colonnes : durée, mission, entreprise
- un tableau de vos compétences en développement avec 3 colonnes : le langage, la durée de la formation et les compétences acquises.
- une liste numérotée de vos langues parlées et/ou lues avec les balises <ol> et <li>.
- une liste non numérotée de vos loisirs avec les balises <ul> et <li>.


### Les contraintes en CSS :

- Tous les titres doivent être personnalisés : marges intérieures et extérieures, couleur et taille de la police.
- La balise <img> de la photo doit être alignée à droite au même niveau que le nom. Elle doit être carrée avec 75 pixels de largeur. Affecter un attribut id à cette balise <img> et définissez le style CSS de cet identifiant.
- Les tableaux n'auront aucune bordure.
- Une ligne sur deux des tableaux aura le fond grisé pour faciliter la lecture. Créer une classe CSS appropriée.
- La police des cellules contenant une date et/ou une durée sera grasse et d'une autre couleur que les autres éléments. Créer une classe CSS appropriée.
- La liste des loisirs devra être disposée sur une ligne. Affecter un attribut id à cette balise <ul> et définissez le style CSS de cet identifiant.

**</>** En respectant ces contraintes, commencer par écrire le contenu de votre CV en HTML.

**</>** Travailler ensuite la mise en page des éléments avec la feuille de style. Les règles de style du CV seront écrites dans un nouveau fichier CSS « cv.css ».

Dans le <aside> de votre site, ajouter des liens avec la balise <a> vers des sites de références utiles pour le

**</>** développement web... Et ne pas oublier de sauvegarder de l'état actuel de votre projet avec  **git** en saisissant la commande :

```
git commit -m "Défi 3 - Mon CV de GEEK"
```