



## Module Web 01

# Je démarre en HTML et CSS



### 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
TD 2 - Le langage HTML et les balises de formatage du texte.....	5
Défi 1 - Mon CV de Geek : le code HTML.....	6
TD 3 - CSS ou comment donner du style à une page HTML.....	7
TD 4 - Les sélecteurs en CSS.....	9
Défi 2 - Un look de <geek/> à votre CV.....	10
Défi 3 - Mise en page d'un site web.....	11
TD 5 - Grilles CSS ou CSS Grid.....	12
Défi 4 - La grille CSS du site web.....	14
Défi 5 - Un site internet responsive.....	15
Défi 6 - Le design de votre page et son contenu.....	16

## Les compétences visées

</>	Catégorie	Je suis capable de :	
H T M L	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>	
C S S	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.	
G I T	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>Mon CV</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>

## TD 2 - Le langage HTML et les balises de formatage du texte

Le langage HTML fournit de nombreuses balises pour formater le texte d'une page web : les titres, les paragraphes, les listes, les tableaux, les liens, etc.

### Les titres

? Donner le nom de la balise permettant d'indiquer qu'un texte est un titre. Combien de niveau de titre différents sont possibles ?

? Donner le code HTML permettant d'écrire « MON CV » en titre principal.

? Donner le code HTML permettant d'écrire « Mes formations » en titre secondaire.

### Les paragraphes

? Écrire en HTML le paragraphe suivant : « Je suis actuellement en première année du BTS SNIR. »

### Les tableaux

Voici un exemple de tableau écrit en HTML.

```
<table>
  <tr>
    <th>Nom</th>
    <th>Distance du soleil (106 km)</th>
    <th>Température moyenne (°C)</th>
  </tr>
  <tr>
    <td>Mercure</td>
    <td>57,9</td>
    <td>167</td>
  </tr>
  <tr>
    <td>Vénus</td>
    <td>108,2</td>
    <td>464</td>
  </tr>
  <tr>
    <td>Terre</td>
    <td>149,6</td>
    <td>15</td>
  </tr>
</table>
```

? Donner la signification des balises du code précédent :\*

<table> :

<tr> :

<th> :

<td> :

### Les listes numérotées et non numérotées

? En recherchant sur Internet, trouver les balises utilisées pour créer des listes numérotées et non-numérotées.

? Écrire en HTML une liste non numérotée contenant : « rouge », « vert », « bleu ».

? Écrire en HTML une liste numérotée contenant : Mercure, Vénus, Terre.

## Défi 1 - Mon CV de Geek : le code HTML

### Test de la structure de base d'une page HTML (voir le TD1)

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 «CV.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** !



### Le contenu de votre CV (voir le TD2)

L'objectif de cette partie est de créer une page web contenant votre CV. Ce CV doit être rédigé dans le but de rechercher votre stage de fin d'année. Voici les éléments qui devront apparaître :

- 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

**</>** Écrire en HTML votre CV en respectant les contraintes HTML suivantes :

- 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>.

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

```
git commit -m "Defi 1 - le contenu du CV en HTML"
```

## TD 3 - 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 :

```
body {
  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 :

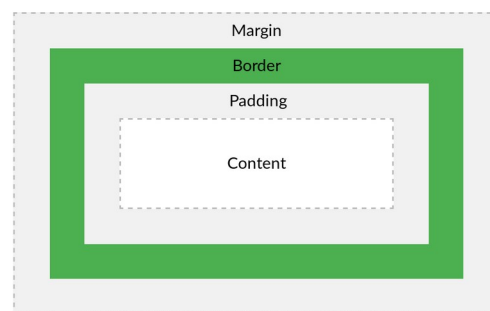
```
margin: 10px;
```

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 :

```
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 :



## TD 4 - 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 2 - Un look de <geek/> à votre CV

### Créer le fichier CSS (voir TD1)

**</>** Créer un fichier nommé `cv.css`, dans le même répertoire que votre fichier `cv.html`. Ajouter le code suivant :

```
body {  
    border: 10px solid #E0E0E0;  
    border-radius: 20px;  
    margin : 20px;  
}
```

**?** Expliquer en détail le code CSS ci-dessus.

### Lier le fichier HTML et le fichier CSS

Pour que les styles du fichier `cv.css` soient appliqués aux balises au fichier « `cv.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 « `cv.html` ».

Enregistrer les fichiers «`cv.html`» et «`cv.css`» puis rafraichir 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.

**</>** Modifier le style du `body` selon votre convenance

### Les contraintes CSS de votre CV

Voici la liste des contraintes CSS qui vous sont imposées :

- Tous les titres doivent être personnalisés : marges intérieures et extérieures, couleur et taille de la police selon votre goût
- La balise `<img>` de la photo doit être alignée à droite au même niveau que le nom. Elle doit être carré 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.

**</>** Travailler la mise en page des éléments dans la feuille de style `cv.css`.

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

```
git commit -m "Defi 2 - ajout de la feuille de style du cv"
```

**</>** BONUS : Trouver la procédure pour héberger votre CV sur GitHub puis écrire l'url permettant d'accéder à votre CV.

## Défi 3 - Mise en page d'un site web

### Objectifs

L'objectif de cette partie est de mettre en place un site internet de présentation et de gestion des drones que vous utiliserez au cours des 2 années de votre formation. On laisse donc de côté ce que nous venons de réaliser pour votre CV, et on part sur autre chose.

Le design de ce site devra ressembler à celui-ci :

### Les 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.

**</>** Dans votre répertoire de travail, créer un nouveau fichier nommé « index.html ». Compléter ce fichier avec le code de base d'une page HTML (voir le TD1)

**</>** En respectant la mise en page du code, ajouter dans le body de votre fichier « index.html » :

- 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...*

**</>** Dans la barre de navigation, ajouter les 4 liens (Présentation, Suivi, Connexion et Inscription) en utilisant la balise <a href="#">. La mention href="#" permet de faire un lien vers la page elle-même.

**</>** Dans le footer, ajouter un lien vers votre CV avec la balise <a> Cette fois-ci, faire le lien vers votre fichier « cv.html ». Vérifier que le lien fonctionne correctement.

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

```
git commit -m "Défi 3 - structure du site web"
```



## TD 5 - Grilles CSS ou CSS Grid

### Présentation

Les grilles CSS font parties de la spécification CSS. Elles permettent de mettre en page un site en divisant l'espace d'affichage en colonnes et en lignes, comme un tableau.

Cette manière de faire simplifie énormément la mise en page d'un site internet.

### Un exemple de grilles CSS

Partons d'un exemple pour voir comment s'utilise une grille CSS. Commençons par le code HTML partiel d'une page web :

```
<body>
  <header>En-tête</header>
  <nav>Navigation</nav>
  <main>Zone principale</main>
  <footer>Pied de page</footer>
</body>
```

Voici le code CSS utilisant la grille CSS :

```
body {
  display: grid;
  width: 100%;
  height: 250px;
  grid-template-areas: "head head"
                       "nav main"
                       "nav foot";
  grid-template-rows: 50px 1fr 30px;
  grid-template-columns: 150px 1fr;
}

header {
  grid-area: head;
}

nav {
  grid-area: nav;
}

main {
  grid-area: main;
}

footer {
  grid-area: foot;
}
```

? Dans le code CSS, quelle ligne indique que la mise en page utilisera les grilles CSS ?

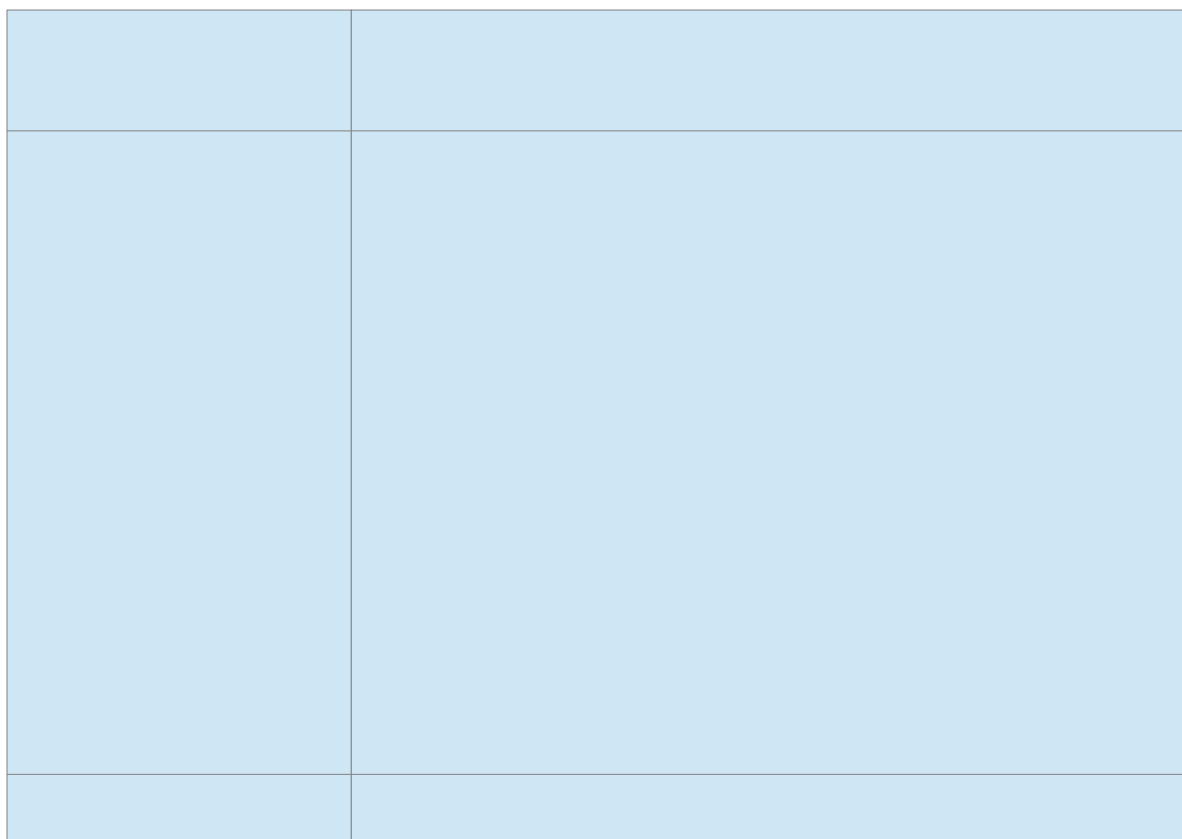
? Expliquer la signification des valeurs pour les propriétés width et height.

? D'après la propriété grid-template-rows, en combien de lignes est divisé le body ? Donner la taille de chaque ligne.

- ? D'après la propriété `grid-template-columns`, en combien de colonnes est divisé le body ? Donner la taille de chaque colonne.

- ? Expliquer le rôle de la propriété `grid-area` et `grid-template-areas`.

- ? Faire un dessin représentant la mise en page et indiquer les emplacements des header, nav, main et footer.



- ? Quel est l'avantage de la mesure `1fr` lorsque l'utilisateur augmente ou diminue la taille de la fenêtre de son navigateur web ?

## Défi 4 - La grille CSS du site web

### Objectif

L'objectif de cette partie est de mettre en place le corps de notre site internet. Des espaces suffisants doivent être réservés pour chaque zone de notre page comme le montre la capture d'écran du défi 3.

Pour atteindre cet objectif, vous utiliserez la technique de mise en page sous forme de grille (voir le TD 5).

### Ajout d'un fichier CSS

**</>** Créer un nouveau fichier CSS nommé « `ossature_grille.css` » dans votre répertoire de travail.

**</>** Ajouter un lien avec la balise `<link>` dans l'entête du fichier `index.html` avec le fichier `ossature_grille.css`.

### Les contraintes de mise en page

Voici les contraintes de mise en page qui vous sont imposées :

- Le body
  - un affichage sous forme de grille
  - 2 colonnes :
    - la colonne de droite prendra 25 % de la largeur
    - la colonne de gauche prendra tout le reste
  - 4 lignes :
    - la première ligne mesurera 80 pixels de haut
    - la seconde ligne mesurera 40 pixels de haut
    - la 3ème ligne mesurera au moins 400 pixels de haut ou plus en fonction de son contenu.
    - la 4ème ligne aura une hauteur qui s'adaptera à son contenu.
  - Les éléments seront disposés comme indiqué sur la capture d'écran du défi 3.
- Les 5 éléments constituant la page web (voir défi 3) se verront attribuer un nom avec la propriété `grid-area` :
  - Le header se nommera `header`.
  - Le nav se nommera `nav`.
  - La section se nommera `section`
  - Le aside se nommera `aside`
  - Le footer se nommera `footer`

**</>** Ajouter toutes ces contraintes de mise en page dans votre fichier `ossature_grille.css`.

Pour avoir un meilleur aperçu de la mise en page, ajouter une bordure grise de 1px sur les 5 éléments

**</>** constituant votre page web. *Remarque* : ces bordures ne sont que temporaires, elles vous permettront (à vous les développeurs) de vérifier que votre mise en page est correcte ! Une fois la vérification faite, vous pourrez mettre ces quelques lignes en commentaire.

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

```
git commit -m "Defi 4 - ossature grille du site"
```

## Défi 5 - Un site internet responsive

### La version mobile

Avec cette technique de mise en page en grille, il est assez simple de gérer l'affichage de notre site internet sur un téléphone. Un site internet qui sait adapter son affichage en fonction de la taille des écrans est appelé site *responsive*. Voyons comment le mettre en place.

Dans votre fichier `ossature_grille.css`, vous allez créer une zone qu'on réservera pour décrire l'affichage du site internet pour les écrans dont la largeur est inférieure à 800 pixels.

```
@media (max-width: 800px) {
  // Zone réservée pour les écrans inférieurs à 800 pixels de large
}
```

Voici les contraintes d'affichage pour ces tailles d'écrans :

- Le body
  - un affichage sous forme de grille
  - 1 seule colonne prenant 100 % de la largeur
  - 4 lignes contenant :
    - Une 1ère ligne de 80 pixels dédiée au header
    - Une seconde ligne dont la taille dépendra de son contenu et qui sera réservée à la navigation
    - Une troisième ligne pour la section dont la hauteur sera d'au moins 400 pixels.
    - Une 4ème ligne pour le footer dont la taille dépendra de son contenu.

**</>** Ajouter toutes ces contraintes de mise en page dans votre fichier `ossature_grille.css`, dans la zone `@media (max-width : 800px)`

Il est possible de d'afficher le rendu que votre site internet aurait sur un téléphone avec les navigateurs web. Pour cela, taper F12 (l'inspecteur de votre page s'ouvre) et une icône de située généralement en haut à droite de l'inspecteur permet de changer les résolutions d'affichage.

**</>** Vérifier que votre site internet s'adapte bien en fonction de la résolution de votre appareil.

**</>** Pour avoir un meilleur aperçu de la mise en page, ajouter une bordure grise de 1px sur les 5 éléments constituant votre page web comme dans le défi précédent.

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

```
git commit -m "Defi 5 - site responsive"
```



Que vous soyez dans un parc, au bureau ou à la maison, vous pouvez décoller à tout moment et découvrir le monde avec un œil nouveau. Tello est doté de deux antennes qui permettent une transmission vidéo

## Défi 6 - Le design de votre page et son contenu

### Objectif

L'objectif de cette partie est de travailler sur le design de votre site internet : le choix des couleurs de police et des couleurs de fond, de la police d'écriture et de sa taille, de marges intérieures et extérieures, et d'autres attributs du texte.

### Design du site

Tous les éléments de design du site seront décrits dans un nouveau fichier : design.css

**</>** Créer le fichier design.css dans votre répertoire de travail.

**</>** Ajouter le lien entre le fichier index.html et le fichier design.css avec la balise <link>.

Voici les contraintes de style que vous ajouterez dans le fichier design.css :

- body :
  - Utiliser une police de caractères (*font*) adaptée au web, par exemple la police Quicksand. Cette police est disponible gratuitement auprès de Google. Sur la page internet de cette police, vous trouverez toutes les informations pour l'utiliser dans votre site internet.
- Les header, nav et footer :
  - couleur de fond : #2874a6 (un bleu)
  - couleur de police : blanc
- Le titre h1 du header :
  - Une marge extérieure haute et basse de 0.7rem
  - Une marge intérieure haute de 0px
  - Une taille de police de 1.5rem
  - Un texte est en petite majuscule et centré
- La barre de navigation
  - un padding de 20px à droite et à gauche
- Les liens de la barre de navigation
  - taille de police de 1rem
  - couleur de police : blanc
  - marge extérieure de 0px
  - marge intérieure de 10px à droite et à gauche
  - Le texte sera centré, en petite majuscule et sans soulignement.
- Lors du passage de la souris sur les liens de la barre de navigation
  - Bordure en bas de 2px en blanc
  - durée de transition de 0.3 seconde
- Le aside
  - marge extérieure de 10px en haut, en bas et à gauche
  - marge intérieure de 10px.
  - bordure de 1px en #2874a6 (un bleu)
  - Arrondi des angles de la bordure de 10px.
- Les liens du footer prendront les mêmes styles que ceux de la barre de navigation



Donner la signification de l'unité rem.

**</>** Traduire l'ensemble de ces contraintes en CSS dans le fichier design.css et vérifier au fur et à mesure que ces contraintes de style s'appliquent correctement aux éléments que vous ciblez.

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

```
git commit -m "Defi 6 - design css de la page d'accueil"
```

**</>** Ajouter du contenu dans la section principale pour présenter les 2 drones que nous utiliserons durant votre BTS. Donner du style aux éléments que vous ajoutez en personnalisant le fichier design.css.

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

```
git commit -m "Defi 6 - ajout du contenu de la page d'accueil"
```



## Défi 7 - Le formulaire de connexion

### Objectif

L'objectif de cette partie est de créer un formulaire de connexion qui ressemblera au final à celui se trouvant à droite. Ce formulaire permettra aux membres du site de s'identifier pour pouvoir trouver et modifier des informations concernant nos drones.

### L'écriture du formulaire en HTML

Le formulaire de connexion sera créé dans le fichier « formulaire\_connexion.html ».

#### Identifiez-vous !

Pseudo :

Mot de passe :

Valider

**</>** Créer le fichier `formulaire_connexion.html` dans votre répertoire de travail.

**</>** En vous aidant du TD1, mettre le code de base d'un fichier HTML dans ce fichier.

Tous les éléments du formulaire de connexion devront apparaître à l'intérieur de la balise `body` de votre fichier.

**?** Donner le nom de la balise en HTML permettant de créer des formulaires

La balise `<label>` permet de créer un texte comme le texte Pseudo dans le formulaire ci-dessus. La balise `input` permet de créer des zones de saisie pour l'utilisateur. Dans notre cas, il s'agit d'une zone de texte. Le code qui correspond à la saisie du pseudo est le suivant :

```
<label for="pseudo">Pseudo : </label>
<input type="text" id="pseudo" name="pseudo_utilisateur">
```

**?** En recherchant sur internet, donner la liste des valeurs de la propriété `type` de la balise `input` qui nous intéressent dans ce formulaire.

**</>** En HTML, mettre en place tous les éléments du formulaire de connexion.

### Personnalisation du design du formulaire en CSS

**</>** Créer un fichier « `formulaire.css` » dans votre répertoire de travail.

**</>** Dans l'entête du fichier « `formulaire_connexion.html` » ajouter un lien vers le fichier « `formulaire.css` » avec la balise `<link>`.

**</>** En CSS, travailler la mise en forme du formulaire (dans « `formulaire.css` ») pour qu'il ressemble à celui donné en exemple.

L'envoi du formulaire et son traitement se feront dans un autre module.

### Lien vers le formulaire dans la page d'accueil

Dans la barre de navigation, vous allez diriger les utilisateurs qui cliquent sur le lien « Connexion » à votre page « `formulaire_connexion.html` ».

**</>** Modifier le lien « Connexion » de la barre de navigation vers la page « `formulaire_connexion.html` ». Vérifier le fonctionnement de ce lien.

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

```
git commit -m "Defi 7 - ajout du formulaire de connexion"
```

## Défi 8 - Le formulaire d'inscription

### L'objectif

L'objectif de cette partie est de créer un formulaire d'inscription qui ressemblera au final à celui ci-contre.

### L'écriture du formulaire en HTML

Le formulaire d'inscription sera créé dans le fichier « formulaire\_inscription.html ».

- </> Créer le fichier `formulaire_inscription.html` dans votre répertoire de travail.
- ? En recherchant sur internet, donner la liste des valeurs de la propriété type de la balise `input` qui nous intéressent dans ce formulaire.



- </> En HTML, mettre en place tous les éléments du formulaire d'inscription.

### Design du formulaire en CSS

Nous allons utiliser le même fichier CSS pour la personnalisation du design de tous les formulaires : « formulaire.css ».

- </> En CSS, travailler la mise en forme du formulaire pour qu'il ressemble à celui donné en exemple.

L'envoi du formulaire et son traitement se feront dans un autre module.

### Lien vers le formulaire dans la page d'accueil

Dans la barre de navigation, vous allez diriger les utilisateurs qui cliquent sur le lien « Inscription » à votre page « formulaire\_inscription.html ».

- </> Modifier le lien « Inscription » de la barre de navigation vers la page « formulaire\_inscription.html ». Vérifier le fonctionnement de ce lien.
- </> Sauvegarder de l'état actuel de votre projet avec  **git** en saisissant la commande :

```
git commit -m "Défi 7 - ajout du formulaire d'inscription"
```

### Inscrivez-vous !

Nom :

Prénom :

Adresse mail :

Date de naissance :

Pseudo :

Mot de passe :

Ressaisir le mot de passe :

M'inscrire