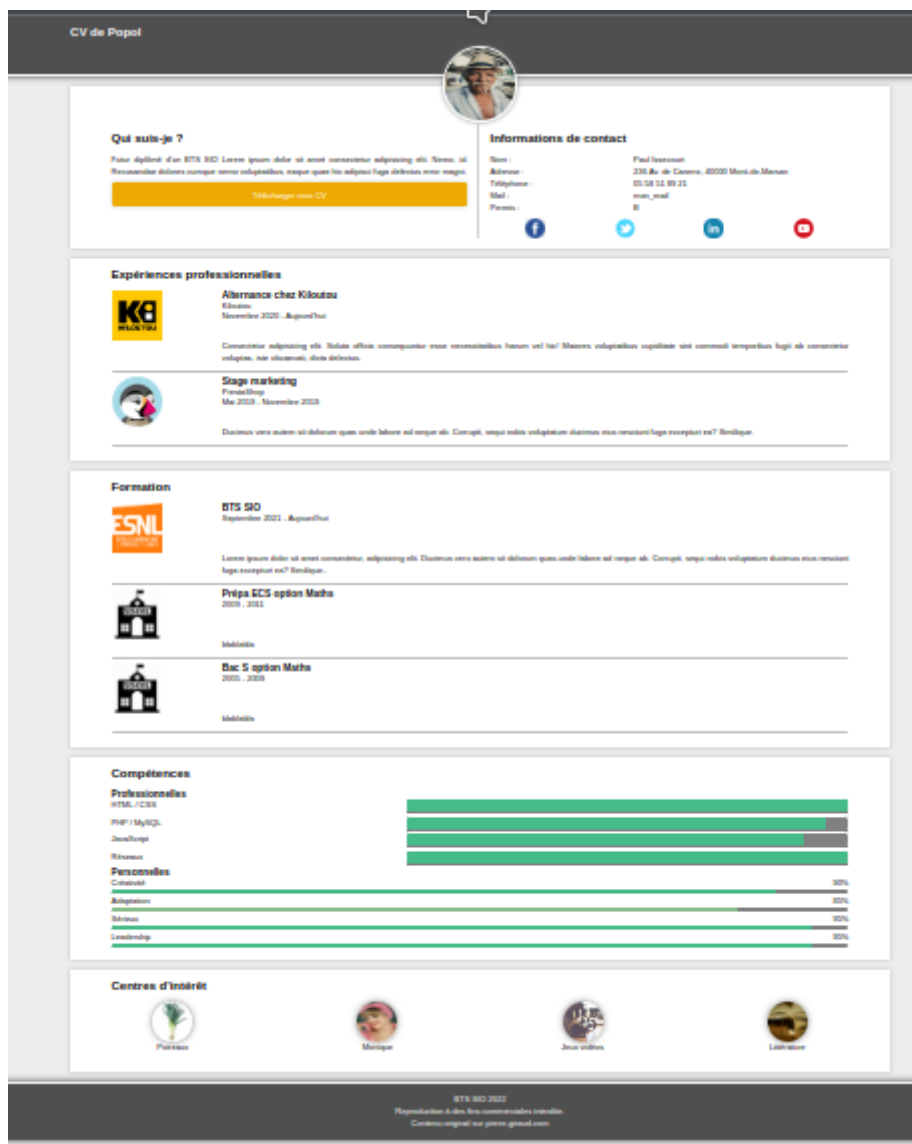


TD Création d'un CV en ligne.

Nous allons construire un CV en ligne avec un site statique déployé via Netlify.

Voici le résultat auquel on souhaite parvenir (le CV, pas l'image)



Préambule

Dans le TD, on vous propose une structure issue d'exercices menés avec d'autres classes. En pratique, bien évidemment, on ne trouvera quasiment jamais la structure optimale du premier coup et l'on est souvent obligé de revenir sur le code HTML et surtout sur le code CSS pour ajuster et affiner le rendu.

Par conséquent une phase de réflexion et de maquettage s'avère nécessaire avant de commencer à coder afin de ne pas perdre trop de temps ou de produire du code brouillon, .

Les codes HTML et CSS présentés après ne sont pas directement copiables. C'est voulu. Si besoin, demandez le code source directement à l'intervenant.

Avec un chèque à l'ordre de..

Maquettage

Nous définissons deux maquettes, une pour les mobiles et la version pour PC.

Version Mobile

Header



Présentation

Contact



Expérience



Expérience

Expérience

Formation

Expérience

Compétences Pro



Compétences Perso




Centres intérêt



Footer

Header

Présentation



Contact

F

F

F

Expérience

Expérience

Expérience

Formation

Expérience

Compétences Pro

Compétences Perso

Centres intérêt

Footer

TD : création de la page statique

Structure HTML de notre page de CV

Comme d'habitude, nous commençons par créer la structure HTML avant de définir les styles particuliers en CSS.

Nous allons déjà ici commencer avec la structure minimale d'une page valide et allons également renseigner le `meta name="viewport"` et lier notre fichier HTML à un fichier CSS.

Dans votre dossier projet :

- Créez un dossier, nommez le `my_static_cv` (ou autre) et placez vous à l'intérieur.

```
mkdir my_static_cv && cd my_static_cv
```

- créez un fichier `index.html` et un dossier `assets` en suivant l'arborescence suivante

```
--assets-
  |-- css
  |-- images
  |-- js
```

dans le sous-dossier `assets/css` créez un fichier `styles.css`

Index.html

En vous aidant de l'autocomplétion de Visual-studio-code (VSC)

Commencez à entrer `html` et dans le menu contextuel, vous sélectionnez `html:5`

Vous obtenez le squelette du site.



```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CV de Popol</title>
</head>

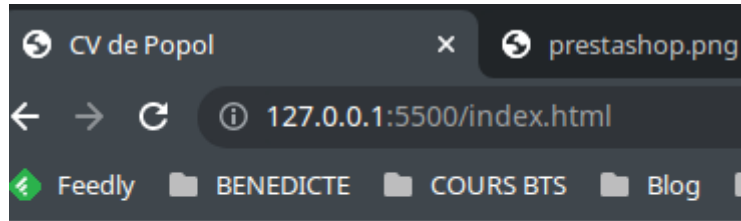
<body>

</body>

</html>
```

Notez que VSC renseigne par défaut le `meta name="viewport"`

Modifiez le titre et visualisez le site avec l'extension GoLive par exemple.



Ensuite :

- Dans le `<head>` liez votre fichier HTML avec votre fichier css
`<link rel="stylesheet" href="./assets/css/styles.css">`
- Créez un `<header>` et placez-y le titre principal de notre page de CV.

```
...  
<body>  
  <header>  
    <h1> CV de Popol</h1>  
  </header>  
</body>  
...
```

La structure

En se basant sur la maquette, on crée la structure en disposant les différentes sections avec leurs titre `h2` respectifs.

```
...
<body>
  <header>
    <h1> CV de Popol</h1>
  </header>

  <section>
    <!-- Partie présentation du CV-->
    <!-- + Bouton envoyer cv      -->
  </section>

  <section>
    <h2>Expériences professionnelles</h2>
  </section>

  <section>
    <h2>Formation</h2>
  </section>

  <section>
    <h2>Compétences</h2>
  </section>

  <section>
    <h2>Centres d'intérêt</h2>
  </section>

  <footer>
  </footer>

</body>
...
```

On va implémenter chaque sections.

Création de la section de présentation du CV

Cette section va contenir :

- Un bloc contenant une photo de profil ;
- Un bloc contenant un paragraphe de présentation et un lien de téléchargement du CV au format PDF ;

Explications

La section est composée de trois blocs délimités par des `<div>`

- On crée un premier bloc dans lequel on place notre photo de profil. On donne une classe `class="photo-profil"` pour pouvoir styliser celle-ci dans le css.
- On déclare un bloc présentation qui possède une structure très simple : un titre, un paragraphe et un lien.
- Le bloc contact est un peu plus complexe : on place les différentes informations de contact dans différents `div class="contact-flex"`. L'idée va être de créer des conteneurs flexibles pour aligner les éléments.
- On crée également un bloc avec les icônes sociales qui seront cliquables (les liens sont laissés vides).

Contrairement à ce que la maquette pourrait donner à penser, on ne crée pas de bouton le téléchargement, mais on stylisera le lien.

Sous cette première section de présentation, nous allons avoir deux sections qui vont avoir une structure similaire : première une section décrivant nos expériences professionnelles et une seconde avec notre parcours scolaire.

Début du style

Normalement, on doit s'occuper du style après avoir complètement défini la structure HTML du site.

Néanmoins, pour faciliter la lecture (et notamment réduire la taille des images réseaux sociaux qui nous oblige à beaucoup scroller), on va appliquer quelques règles:

Dans styles.css

Un reset CSS des marges internes et externes sur tous nos éléments pour avoir le même comportement dans tous les navigateurs et pouvoir les positionner précisément ensuite.

On va choisir :

- Une fonte sans-serif : Avenir (si vous voulez utiliser les google-font [regardez ici](#))
- des marges intérieures (padding) et extérieures (margin) initialisées à 0.
- Un alignement du texte sur justifié (plus [d'explications ici](#)). Cette propriété est comparable à l'alignement dans un traitement de texte
- La propriété `box-sizing` et sa valeur `border-box` pour que la largeur définie inclue les marges internes et bordures dans son calcul et afin que notre paragraphe ne dépasse pas de son élément parent.
- Une couleur de fond : gris très léger

```
/*Generics styles*/
*{
  font-family: Avenir, sans-serif;
  padding: 0px;
  margin: 0px;
  text-align: justify;
  box-sizing: border-box;
}
body{
  background: #ededed;
}
```

Nous allons tout de suite réduire la tailles des logo des réseaux sociaux `class="social"` pour les limiter à 40px;

```
...
.social img{
  width: 40px;
  height: 40px;
}
...
```

La section 'Expérience'

Cette section décrit vos expériences professionnelles

Chaque bloc expérience va suivre le même schéma :

- Un logo
- Des informations « méta » par rapport à l'expérience
- Une description de celle-ci.

```

...
<section>
  <h2>Expériences professionnelles</h2>

  <div class="exp">
    <div class="exp-logo">
      <a href="#"></a>
    </div>
    <div class="exp-info">
      <h3>Alternance chez Kiloutou</h3>
      <h4>Kiloutou</h4>
      <h4>Novembre 2020 - Aujourd'hui</h4>
    </div>
    <div class="exp-desc">
      <p>Consectetur adipisicing elit. Soluta officia consequuntur esse necessitatibus harum vel hic!
      Maiores voluptatibus cupiditate sint commodi temporibus fugit ab consectetur voluptas,
      iste obcaecati, dicta delectus.</p>
    </div>
  </div>
  <div class="exp">
    <div class="exp-logo">
      <a href="#"></a>
    </div>
    <div class="exp-info">
      <h3>Stage marketing</h3>
      <h4>PrestaShop</h4>
      <h4>Mai 2019 - Novembre 2019</h4>
    </div>
    <div class="exp-desc">
      <p>Ducimus vero autem sit dolorum quas unde labore ad neque ab.
      Corrupti, sequi nobis voluptatum ducimus eius nesciunt fuga excepturi ea? Similique.</p>
    </div>
  </div>
</section>
...

```

La section formation

Elle a une structure similaire à la section expérience.

Ici on aura dans chaque bloc :

- Un logo
- Le titre de la formation et la date
- La description

```

...
<section>
  <h2>Formation</h2>
  <div class="exp">
    <div class="exp-logo">
      <a href="#"></a>
    </div>
    <div class="exp-info">
      <h3>BTS SIO</h3>
      <h4>Septembre 2021 - Aujourd'hui</h4>
    </div>
    <div class="exp-desc">
      <p>Lorem ipsum dolor sit amet consectetur, adipisicing elit.
        Ducimus vero autem sit dolorum quas unde labore ad neque ab.
        Corrupti, sequi nobis voluptatum ducimus eius nesciunt fuga excepturi ea? Similique..</p>
    </div>
  </div>
  <div class="exp">
    <div class="exp-logo">
      <a href="#"></a>
    </div>
    <div class="exp-info">
      <h3>Prépa ECS option Maths</h3>
      <h4>20019 - 2020</h4>
      <h4 class="Etablissement">Etablissement : Lycée Gerard Menez - Pau</h4>
    </div>
    <div class="exp-desc">
      <p>blablabla</p>
    </div>
  </div>
  <div class="exp">
    <div class="exp-logo">
      <a href="#"></a>
    </div>
    <div class="exp-info">
      <h3>Bac S option Maths</h3>
      <h4>2014 - 2018</h4>
      <h4 class="Etablissement">Etablissement : Lycée Paul Presboit - Nanterre</h4>
    </div>
    <div class="exp-desc">
      <p>blablabla</p>
    </div>
  </div>
</section>
...

```

La section compétence

Pour la section présentant les compétences, on utilise un système de graph/barres.

Nous verrons après deux design différents en CSS et nous mettrons en place en même temps les deux type de barre-graphe.

La structure de cette section se limitera pour l'instant à :

- Un titre `<h2>`
- un bloc (classe `comp-perso` ou `comp-pro`) qui liste la compétence
- Une valeur associée.

```
...

<section>
  <h2>Compétences</h2>
  <h3>Professionnelles</h3>
  <div class="comp-pro">
    <p>HTML / CSS</p>
    <p>90%</p>

  </div>
  <div class="comp-pro">
    <p>PHP / MySQL</p>
    <p>85%</p>

  </div>
  <div class="comp-pro">
    <p>JavaScript</p>
    <p>50%</p>
  </div>
  <div class="comp-pro">
    <p>Réseaux</p>
    <p>100%</p>
  </div>
  <h3 class="h3gauche">Personnelles</h3>
  <div class="comp-perso">
    <p>Créativité</p>
    <p>90%</p>

  </div>
  <div class="comp-perso">
    <p>Adaptation</p>
    <p>85%</p>

  </div>
  <div class="comp-perso">
    <p>Sérieux</p>
    <p>95%</p>

  </div>
  <div class="comp-perso">
    <p>Leadership</p>
    <p>95%</p>

  </div>
</section>

...
```

La section centres d'intérêt.

Nous utiliserons des vignettes pour présenter nos centre d'intéret.

Préparez vos images ou utilisez celles fournies.

Chaque vignette

```
...  
  
<section>  
  <h2>Centres d'intérêt</h2>  
  <figure class="interet">  
      
    <figcaption>Poireaux</figcaption>  
  </figure>  
  <figure class="interet">  
      
    <figcaption>Monique</figcaption>  
  </figure>  
  <figure class="interet">  
      
    <figcaption>Jeux vidéos</figcaption>  
  </figure>  
  <figure class="interet">  
      
    <figcaption>Littérature</figcaption>  
  </figure>  
</section>  
  
...
```

Le footer

Pour terminer, nous allons créer un footer

Nous y mettrons quelques liens et infos légales

```
...  
  
<footer>  
  <p><a href="https://esnl.campuslandes.com">BTS SIO</a> 2022</p>  
  <p>Reproduction à des fins commerciales interdite.</p>  
</footer>  
  
...
```

Style du CV

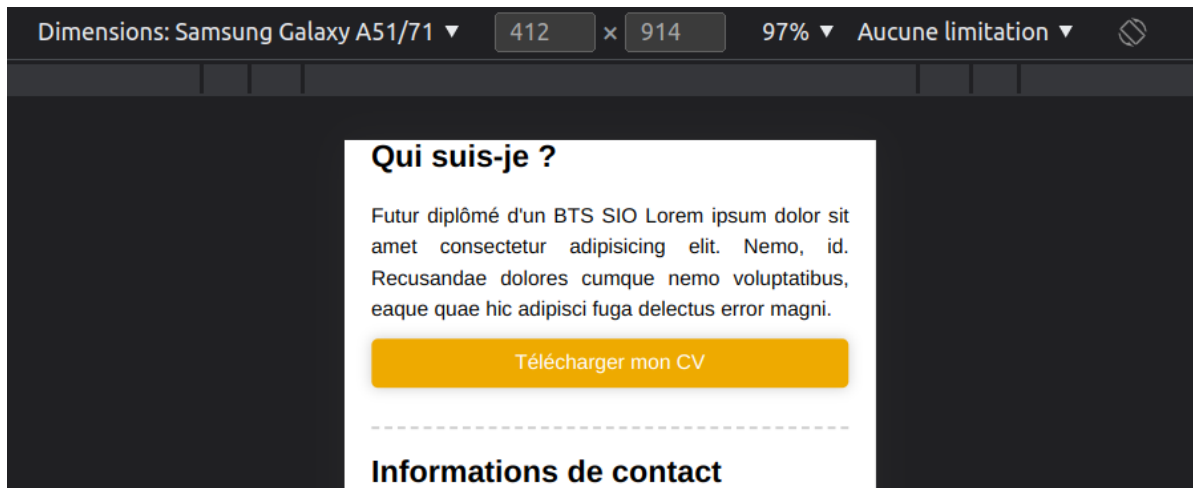
Notre CV sera développé suivant le paradigme "**mobile first**" :

Nous allons commencer par créer les styles de la version mobile de notre CV qui seront donc les styles standards puis allons ensuite utiliser les Media Queries pour les styles de la version bureau.

En version mobile, on va vouloir que le CV s'affiche en pleine page, c'est-à-dire que les différents blocs occupent tout l'espace en largeur et allons ensuite centrer les éléments à l'intérieur de façon à ce qu'on ait un espace vide égal à 5% de la largeur totale de la page de chaque côté.

Les blocs seront donc affichés les uns en dessous des autres.

Pour tester votre page utiliser les Dev-Tools de chrome ou de firefox pour tester votre site en utilisant la vue "mobile"



Nous allons nous occuper de régler les styles du `header` et du `footer` qui sont les parties les plus simples de notre CV.

```
*STYLES GENERAUX*/
*{
  font-family: Avenir, sans-serif;
  padding: 0px;
  margin: 0px;
  text-align: justify;
  box-sizing: border-box;
}
body{
  background: #ededed;
}
p{
  font-size: 1em;
  line-height: 1.5em;
}
a{
  text-decoration: none;
  color: #000;
}

/*HEADER & FOOTER*/
header, footer{
  width: 100%;
  padding: 20px 0px;
  background: url("https://campuslandes.com/") #4f4f4f;
  box-shadow: 0px 0px 15px #333;
}
header{
  border-bottom: 2px solid #fff;
  height: 120px;
}
header h1{
  color: white;
  font-size: 1.5em;
  width: 90%;
  margin: 0 auto;
}

footer{
  border-top: 2px solid #fff;
}
footer p{
  width: 90%;
  margin: 0 auto;
  color: #fff;
  text-align: center;
}
footer a{
  color: #fff;
}
```


Style des sections

Les éléments section vont tous être des conteneurs flexibles.

- L'axe horizontal sera l'axe principal.
- On aligne les éléments au centre selon l'axe principal (l'axe horizontal donc) avec `justify-content : center`.
- On définit les styles : couleur de fond, ombres, etc.

```
...
/*SectionsS*/
section{
  display: flex;
  flex-flow: row wrap;
  justify-content: center;
  width: 100%;
  padding-bottom: 20px;
  margin: 20px auto;
  box-shadow: 0px 0px 10px #bbb;
  background-color: #fff;
}

section h2{
  margin: 20px 0;
  width: 90%;
}
section > div{
  width: 90%;
}
...
```

display: flex;

Les `<section>` sont des conteneurs flexibles (`display: flex;`).

Héritage

Tous les enfants directs de `.section` sont des éléments flexibles.

our notre première section, nos 3 `div class="photo-profil"`, `div class="prez"` et `div class="contact"` sont donc des éléments flexibles.

Ces trois blocs occupent par défaut 90% de l'espace disponible en largeur dans le conteneur avec `width: 90%` tant qu'aucun `flex-basis` n'est défini. De plus, ils sont alignés au centre selon leur axe principal qui est l'axe horizontal grâce à `justify-content : center`.

Photo de profil

```
/*Section informations*/
.photo-profil{
  text-align: center;
}
.photo-profil img{
  width: 150px;
  height: 150px;
  border: 3px solid #fff;
  box-shadow: 0px 0px 10px #777;
  border-radius: 50%;
  margin-top: -80px;
}
```

Au centre

Il va suffire d'ajouter un `text-align : center` à notre premier `div` pour centrer notre photo dans la section.

De l'ombre

D'un point de vue esthétique, on va régler les dimensions de la photo sur `150*150px` et allons lui attribuer une bordure blanche ainsi et allons créer un effet d'ombre autour.

```
border: 3px solid #fff;
box-shadow: 0px 0px 10px #777;
```

Un rond...

```
border-radius: 50%;
```

On applique un `border-radius` égal à la moitié de ses dimensions, ce qui aura pour effet « d'arrondir » notre image.

Le chevauchement de la photo

```
margin-top: -80px;
```

On veut que la photo se trouve à cheval entre notre `header` et notre première `section`.

On utilise un petit trick CSS qui consiste à attribuer une marge haute négative à l'image pour la faire sortir de son conteneur.

Bloc de présentation

Ajout d'une bordure basse.

```
.presentation{
  border-bottom: 2px dashed #ccc;
}
```

Style du lien de téléchargement

```
.presentation a{
  display: block;
  text-align: center;
  padding: 10px;
  margin: 10px auto 30px auto;
  color: #fff;
  background-color: #EA0;
  box-shadow: 0px 0px 10px #ccc;
  border-radius: 5px;
}
```

Les liens, par défaut, sont des éléments inline, on le transforme donc en block

```
display: block;
```

Couleurs : blanc pour les lettres, Jaune pour le fond

```
color: #fff;
```

```
background-color: #EA0;
```

Bloc de présentation : Contact

```
.contact-flex{
  display: flex;
  flex-flow: row wrap;
  align-items: center;
}
```

On veut aligner les informations de contact les unes sous les autres.

Pour cela, on va appliquer un `display : flex` à nos éléments portant la classe `contact-flex` en définissant à nouveau l'axe horizontal comme axe principal.

```
flex-flow: row wrap;
```

Les éléments internes du flex-box se positionnent en ligne (côte à côte). Lorsque la largeur du container est remplie, les éléments suivants sont autorisés à commencer une nouvelle ligne. (Voir annexe ou, mieux, la Doc MDN).

On utilise `align-items: center` pour centrer les éléments en hauteur au cas où certains prendraient plusieurs lignes.

```
.contact-flex p:first-child{
  flex: 0 0 40%;
}
.contact-flex p:last-child{
  flex: 0 0 60%;
}
```

first & last child

```
.contact-flex p:first-child
```

Cible le premier paragraphe de chaque `.contact-flex`

```
flex: 0 0 40%;
```

Attribue une largeur de base égale à 40% du conteneur flexible.

```
.contact-flex p:last-child
```

Cible le dernier paragraphe de chaque `.contact-flex`

Le conteneur `div class=social`

```
.social{
  display: flex;
  justify-content: space-around;
  margin: 10px auto 0px auto;
}
.social img{
  width: 40px;
  height: 40px;
}
```

```
justify-content: space-around;
```

Permet de répartir l'espace libre selon l'axe principal dans le conteneur équitablement entre les différents éléments avec `justify-content: space-around`. (voir annexe)

Section Expérience et section formation

Les deux sections vont recevoir la même mise en forme. on ne définit qu'une seule classe pour les deux, `.exp`

Ici, chaque expérience est composée de trois blocs:

- Un logo composé d'une image cliquable.
- Des informations importantes
- Une description.

```
...
/*Section expérience et formation*/
.exp{
  display: flex;
  flex-flow: row wrap;
  border-bottom: 1px solid #bbb;
  padding-bottom: 10px;
  margin-bottom: 10px;
}
.exp-info{
  display: flex;
  flex-flow: column wrap;
}
.exp-logo{
```

```

    flex: 0 0 25%;
    max-width: 100px;
  }
  .exp-info{
    flex: 0 0 70%;
    margin-left: auto;
  }
  .exp img{
    width: 100%;
  }
  .exp h3{
    font-size: 1.2em;
  }
  .exp h4{
    font-size: 1em;
    font-weight: normal;
  }
  ...

```

.exp-logo

25% de la largeur du conteneur jusqu'à 100px.

.exp-info

70% de la largeur du conteneur, marge à gauche `auto` ce qui signifie que la marge va absorber l'espace restant dans le conteneur.

.exp img

Va occuper par défaut tout l'espace disponible dans l'élément parent (`.exp-logo`).

La section compétences

Nous n'avons pas pour l'instant défini notre code HTML pour les barres de compétence.

Nous modifions le code HTML de `index.html`

Les barre de compétences, auront deux styles différents, un pour les compétences pro et un pour les compétences perso.

Nous créons une `<div>` conteneur aura pour enfant un `` représentant la barre, dont la classe est représentative du % de compétence.

Ex : une compétence de 85%

```
<div class="conteneur-barre-pro"><span class="barre c85"></span></div>
```

Bien entendu, il faudra une classe par valeur présentée, essayez donc de limiter le nombre de valeurs différentes. Arrondissez, si besoin.

```

<section>
  <h2>Compétences</h2>

```

```

<h3 class="h3gauche">Professionnelles</h3>
<div class="comp">
  <p>HTML / CSS</p>
  <div class="conteneur-barre-pro"><span class="barre c100"></span></div>
</div>
<div class="comp">
  <p>PHP / MySQL</p>
  <div class="conteneur-barre-pro"><span class="barre c95"></span></div>
</div>
...

Idem pour toutes les compétences pro

...

<h3 class="h3gauche">Personnelles</h3>
<div class="comp-perso">
  <p>Créativité</p>
  <p>90%</p>
  <div class="conteneur-barre-perso"><span class="barre c90"></span></div>
...
Idem pour toutes les compétences perso
...

</section>

```

Le style des compétences

Le niveau de maîtrise de nos compétences va être représenté par une barre plus ou moins remplie et plus ou moins verte. Pour nos compétences professionnelles, nous allons vouloir placer les barres à côté des compétences tandis que pour les compétences personnelles nous allons placer les barres en dessous.

Compétences pro

```

*Section compétences*/
.h3gauche{
  width: 90%;
}
.comp-pro{
  display: flex;
  flex-flow: row wrap;
  margin-bottom: 10px;
  justify-content: space-between;
}
}
.comp-pro p{
  flex: 0 0 40%;
}
.conteneur-barre-pro{
  flex: 0 0 60%;
  border-radius: 5px;
  background-color: grey;
}

```

```

border-bottom: 1px ridge grey;
}
.barre{
display: block;
height: 100%;
border-top-left-radius: 5px;
border-bottom-left-radius: 5px;
}
.c60{
width: 60%;
background-color: #8B8;
}
.c85{
width: 85%;
background-color: #8B8;
}
.c90{
width: 90%;
background-color: #4B8;
}
.c95{
width: 95%;
background-color: #4B8;
}
.c100{
width: 100%;
background-color: #4B8;
border-radius: 5px;
}

```

- On définit l'axe horizontal comme axe principal.
- Les textes décrivant les compétences occupent 40% de l'espace
- Le conteneur de barre occupe 60% de l'espace du conteneur.

La barre

- `display : block` pour nos `span`
- Une hauteur de 100% afin que les barres aient bien la même hauteur que les paragraphes.
- On applique finalement la couleur sur une certaine largeur du conteneur grâce aux classes `c85`, `c90`, etc.
- Pour les barres, on pense bien à répliquer le `border-radius` du conteneur lorsqu'elles touchent un angle du conteneur afin d'avoir le meilleur rendu possible (ici 5px).

Compétences perso

```
.comp-perso{
  display: flex;
  flex-flow: row wrap;
  justify-content: space-between;
}
.conteneur-barre-perso{
  flex: 0 0 100%;
  height: 0.5em;
  margin-bottom: 0.25em;
  border-radius: 5px;
  background-color: grey;
}
```

- Les dimensions sont différentes.
- On passe un `flex: 0 0 100%` à nos conteneurs de barre afin qu'ils occupent **tout l'espace disponible**. Ce qui implique de passer à la ligne.

Section centre d'intérêt.

On va afficher des vignettes avec une légende en dessous et afficher deux vignettes par ligne en centrant le tout.

```
/*Section interets*/
.interet{
  flex: 0 1 50%;
  display: flex;
  flex-flow: column wrap;
  align-items: center;
  margin-bottom: 20px;
}
.interet img{
  width: 80px;
  height: 80px;
  border-radius: 50%;
  box-shadow: 0px 0px 15px #555;
}
```

`flex: 0 1 50%;`

La propriété **flex** est une propriété raccourcie qui définit la capacité d'un élément flexible à modifier ses dimensions afin de remplir l'espace disponible de son conteneur.

Les propriétés détaillées correspondantes à cette propriété raccourcie sont [flex-grow](#), [flex-shrink](#) et [flex-basis](#).

Chaque vignette occupe 50 % de l'espace qui leur est dédié en pratique deux par ligne.

`flex: 0 1 25%;` - quatre par ligne

`flex: 0 1 33%;` - trois par ligne

`flex: 0 1 60%;` - une seule vignette par ligne.

Styles CSS de la version bureau de notre CV

Pour la version bureau de notre CV HTML et CSS, nous allons placer la présentation et les contactsc ôte-à-côte plutôt que les uns en dessous des autres.

Pour créer cette version bureau, nous allons utiliser la règle CSS `@media screen and (min-width: 980px){}`.

Les styles définis dans cette règle vont donc s'appliquer pour tous **les appareils disposant d'un écran** et lorsque **la fenêtre est plus large que 980px**.

```
/*Version bureau CV*/
@media screen and (min-width: 980px){
  section{
    width: 80%;
    box-shadow: 0px 0px 10px #bbb;
  }
  a:hover{
    color: #EA0;
  }
  header h1{
    width: 80%;
  }
  .presentation, .contact{
    flex: 0 0 45%;
  }
  .presentation{
    border-bottom: none;
    border-right: 2px solid #ccc;
    padding-right: 20px;
    margin-left: auto;
  }
  .contact{
    padding-left: 20px;
    margin-right: auto;
  }
  .presentation a{
    border: 2px solid transparent;
  }
  .presentation a:hover{
    color: #f28835;
    background-color: #fff;
    border: 2px solid #f28835;
    box-shadow: 0px 0px 20px #666;
  }
  .exp-logo{
    flex: 0 0 10%;
  }
}
```

```
.exp-info, .exp-desc{
  flex: 0 0 85%;
  margin-left: auto;
}
.interet{
  flex: 0 1 25%;
}
}
```

- On modifier la largeur de nos sections afin qu'elles n'occupent que 80% de la largeur de la page.
- On aligne titre principal sur la nouvelle largeur des sections
- On définit une couleur (jaune) pour les textes des liens au survol (`a:hover`).

Alignement des deux sections côte à côte.

Pour nos deux sections suivantes, on se contente de réduire la taille de base des logos dans la section et d'augmenter la taille prise par les informations. On en profite également pour cette fois-ci aligner la description sous les informations de chaque expérience.

On ne touche pas au design de la section compétence qui nous convient très bien en l'état.

Finalement, on place chacun des blocs représentant nos centres d'intérêt côte-à-côte en définissant un `flex: 0 1 25%` pour chacun de nos éléments `interet`.

Styles complets

```
/*STYLES GENERAUX*/
*{
  font-family: Avenir, sans-serif;
  padding: 0px;
  margin: 0px;
  text-align: justify;
  box-sizing: border-box;
}
body{
  background: #ededed;
}
p{
  font-size: 1em;
  line-height: 1.5em;
}
a{
  text-decoration: none;
  color: #000;
}

/*HEADER & FOOTER*/
header, footer{
  width: 100%;
  padding: 20px 0px;
  background: url("https://campuslandes.com/") #4f4f4f;
  box-shadow: 0px 0px 15px #333;
}
header{
  border-bottom: 2px solid #fff;
  height: 120px;
}
header h1{
  color: white;
  font-size: 1.5em;
  width: 90%;
  margin: 0 auto;
}

footer{
  border-top: 2px solid #fff;
}
footer p{
  width: 90%;
  margin: 0 auto;
  color: #fff;
  text-align: center;
}
footer a{
  color: #fff;
}

/*SECTIONS*/
section{
  display: flex;
```

```
flex-flow: row wrap;
justify-content: center;
width: 100%;
padding-bottom: 20px;
margin: 20px auto;
box-shadow: 0px 0px 10px #bbb;
background-color: #fff;
}

section h2{
margin: 20px 0;
width: 90%;
}
section > div{
width: 90%;
}

/*Section informations*/
.photo-profil{
text-align: center;
}
.photo-profil img{
width: 150px;
height: 150px;
border: 3px solid #fff;
box-shadow: 0px 0px 10px #777;
border-radius: 50%;
margin-top: -80px;
}
.presentation{
border-bottom: 2px dashed #ccc;
}
.presentation a{
display: block;
text-align: center;
padding: 10px;
margin: 10px auto 30px auto;
color: #fff;
background-color: #EA0;
box-shadow: 0px 0px 10px #ccc;
border-radius: 5px;
}
.contact-flex{
display: flex;
flex-flow: row wrap;
align-items: center;
}
.contact-flex p:first-child{
flex: 0 0 40%;
}
.contact-flex p:last-child{
flex: 0 0 60%;
}
.social{
display: flex;
justify-content: space-around;
margin: 10px auto 0px auto;
}
.social img{
width: 50px;
height: 50px;
}
```

```

}

/*Section expérience et formation*/
.exp{
  display: flex;
  flex-flow: row wrap;
  border-bottom: 1px solid #bbb;
  padding-bottom: 10px;
  margin-bottom: 10px;
}
.exp-info{
  display: flex;
  flex-flow: column wrap;
}
.exp-logo{
  flex: 0 0 25%;
  max-width: 100px;
}
.exp-info{
  flex: 0 0 70%;
  margin-left: auto;
}
.exp img{
  width: 100%;
}
.exp h3{
  font-size: 1.2em;
}
.exp h4{
  font-size: 1em;
  font-weight: normal;
}

/*Section compétences*/
.h3gauche{
  width: 90%;
}
.comp-pro{
  display: flex;
  flex-flow: row wrap;
  margin-bottom: 10px;
  justify-content: space-between;
}
.comp-pro p{
  flex: 0 0 40%;
}
.conteneur-barre-pro{
  flex: 0 0 60%;
  border-radius: 5px;
  background-color: grey;
  border-bottom: 1px ridge grey;
}
.barre{
  display: block;
  height: 100%;
  border-top-left-radius: 5px;
  border-bottom-left-radius: 5px;
}
.c60{
  width: 60%;
  border-radius: 5px;
  background-color: grey;
}

```

```

        background-color: #8B8;
    }
    .c85{
        width: 85%;
        background-color: #8B8;
    }
    .c90{
        width: 90%;
        background-color: #4B8;
    }
    .c95{
        width: 95%;
        background-color: #4B8;
    }
    .c100{
        width: 100%;
        background-color: #4B8;
        border-radius: 5px;
    }
    .comp-perso{
        display: flex;
        flex-flow: row wrap;
        justify-content: space-between;
    }
    .conteneur-barre-perso{
        flex: 0 0 100%;
        height: 0.5em;
        margin-bottom: 0.25em;
        border-radius: 5px;
        background-color: grey;
    }

    /*Section interets*/
    .interet{
        flex: 0 1 50%;
        display: flex;
        flex-flow: column wrap;
        align-items: center;
        margin-bottom: 20px;
    }
    .interet img{
        width: 80px;
        height: 80px;
        border-radius: 50%;
        box-shadow: 0px 0px 15px #555;
    }

    /*Version bureau CV*/
    @media screen and (min-width: 980px){
        section{
            width: 80%;
            box-shadow: 0px 0px 10px #bbb;
        }
        a:hover{
            color: #EA0;
        }
        header h1{
            width: 80%;
        }
        .presentation, .contact{
            flex: 0 0 45%;
        }
    }

```

```

}
.presentation{
  border-bottom: none;
  border-right: 2px solid #ccc;
  padding-right: 20px;
  margin-left: auto;
}
.contact{
  padding-left: 20px;
  margin-right: auto;
}
.presentation a{
  border: 2px solid transparent;
}
.presentation a:hover{
  color: #f28835;
  background-color: #fff;
  border: 2px solid #f28835;
  box-shadow: 0px 0px 20px #666;
}
.exp-logo{
  flex: 0 0 10%;
}
.exp-info, .exp-desc{
  flex: 0 0 85%;
  margin-left: auto;
}
.interet{
  flex: 0 1 25%;
}
}

```

Annexes

Syntaxes pour flex-flow.

- **flex-flow: row wrap;**

1	2	3
4	5	

Dans ce premier exemple de syntaxe, les éléments internes du flex-box se positionnent en ligne (côte à côte). Lorsque la largeur du container est remplie, les éléments suivants sont autorisés à commencer une nouvelle ligne.

- **flex-flow: row nowrap;**

1	2	3	4
---	---	---	---

Les éléments internes du flex-box se positionnent côte à côte, sur une seule ligne. Les retours à la ligne n'étant pas autorisés, les éléments internes peuvent déborder de la largeur du container.

- **flex-flow: row-reverse wrap;**

3	2	1
	5	4

Les éléments internes du flex-box se positionnent côte à côte en commençant par la fin du container, Les retours à la ligne sont autorisés.

- **flex-flow: row-reverse wrap-reverse;**

	5	4
3	2	1

Les éléments internes se positionnent côte à côte en commençant par la fin du flex-box, Les retours à la ligne se font également en sens inverse : la ligne suivante se positionne au dessus de la ligne précédente.

- **flex-flow: column wrap;**

1	4
2	5
3	

Les éléments internes se positionnent en colonnes. Les retours étant autorisés, les éléments qui débordent la hauteur du container se positionnent sur une deuxième colonne.

- **flex-flow: initial; flex-flow: inherit; flex-flow: revert; flex-flow: unset;**

La valeur initiale est : Voir les propriétés individuelles.

Pour une description plus complète de ces valeurs, reportez-vous à leur description :

[initial](#), [inherit](#), [revert](#), [unset](#).

Liens
