TP – découverte HTML CSS GIT

Création d'un CV web

Objectifs de cette activité :

- découverte des langages HTML et CSS : les 2 incontournables pour créer toute page web
  - HTML est le langage utilisé pour décrire le contenu de la page à afficher.
  - CSS est le langage utilisé pour mettre en forme le contenu (esthétisme et position)
- premier contact avec le programme GIT et en particulier son utilisation par le service en ligne Github afin de pouvoir stocker, partager et gérer les versions de votre code source
- création et mise en ligne d'un CV qui pourra vous servir dans votre recherche de stage

1 Contenu de base d'une page HTML

Créer un fichier nommé **index.html** et l'ouvrir avec Visual Studio Code ainsi qu'avec un navigateur web.

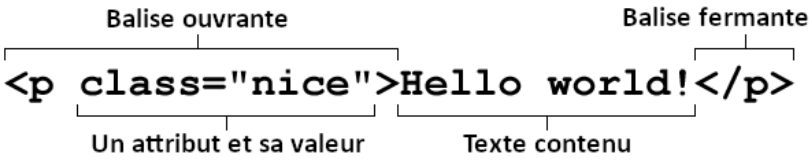
La page d'accueil d'un site se nomme toujours index.html.

Copier-coller le texte ci-dessous dans VSCode et observer le résultat dans le navigateur.

```
<!DOCTYPE html>
<html lang="fr">
        <head>
                <meta charset="utf-8" />
                <title>ici c'est le titre </title>
        </head>

        <body>
                le body             contient
                tout le contenu <br/> de la page
                <!—tout apparait ? -->
        </body>
</html>
```

Le HTML se compose d'une série d'**éléments** imbriqués les uns dans les autres.



Dans l'exemple de code ci-dessus la **balise ouvrante** <html> et la **balise fermante** </html> définissent l'élément global dont **le contenu** est l'ensemble des informations de la page web. Cet élément contient lui-même 2 éléments :

- l'élément <head> : qui contient des informations nécessaires au bon affichage de la page
- l'élément <body> : qui contient ce qui doit être affiché dans la page

Une balise peut contenir 1 ou plusieurs **attributs** avec leur **valeur**.

Remarque : les balises <meta /> et
 sont auto-fermantes, comme elles n'ont pas de contenu, il n'y a qu'une seule balise.

La balise <!DOCTYPE html> indique qu'il s'agit d'un document HTML5. Les balises DOCTYPE des versions précédentes contenaient davantage d'informations.

La balise <meta charset="utf-8" /> permet de spécifier un encodage en UTF-8 des caractères de la page, ce qui permettra de gérer tous les alphabets nécessaires.

Question 1

Comparer le code html du body et son affichage dans le navigateur. Quelles sont les 4 informations sur l'interprétation du code que vous pouvez en déduire ?Il ne retiens pas toutes les spécificité le titre va en haut de l'onglet ,etc

Question 2

Indiquer les 3 rôles de la balise <title>, pour cela :

- regarder le code de la balise <title> et le rendu de la page dans le navigateur
- mettre votre page en favori
- analyser le contenu d'un site référencé dans un moteur de recherche :
  - faire une recherche de « w3c validator » avec un moteur de recherche et observez l'affichage dans le moteur du résultat correspondant au site https://validator.w3.org/
  - dans un autre onglet, entrer dans le site et utiliser l'outil d'inspection du code (fourni dans le navigateur) pour trouver le contenu de sa balise <title>
  - comparer ce contenu et l'affichage du résultat du moteur de recherche

On peut remarquer que tout le monde peut avoir accès au code html d'une page consultée.
Il ne faut donc jamais y mettre d'informations confidentielles.

Le W3C (World Wide Web Consortium) est l'organisme en charge de la spécification de la norme HTML. Le site https://validator.w3.org/ fournit un outil permettant de valider la conformité d'un code html vis-à-vis de la norme. Vous devrez valider le code de vos pages avec cet outil.

Modifier le contenu du fichier index.html pour y afficher un texte du type :

Bonjour, vous trouverez bientôt sur cette page le CV de Bob Dansleciel.
Si vous voulez voir le CV en cours de construction, suivez le lien !

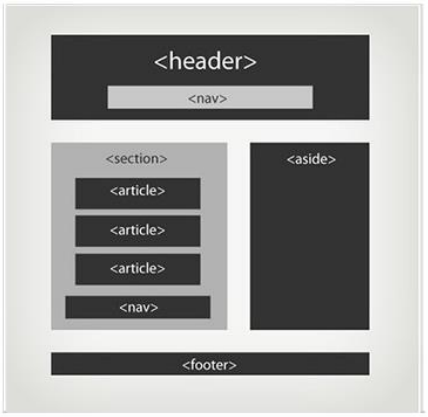
2 Enrichir le contenu du site avec HTML

2.1 Organiser les blocs de la page

L'organisation de la page se fait, dans le body, à l'aide de balises définissant les différentes zones standards de la page :

- l'entête dont le contenu est défini entre des balises <header>
- le contenu principal (spécifique à cette page, ne peut être inclue dans un autre élément) : <main>
- le pied de page : <footer>
- une barre (souvent latérale) pouvant donner des informations complémentaires sur le contenu : <aside>
- un menu de navigation : <nav>
- une zone de contenu indépendante du reste de la page : <article>
- une section : <section>
- une zone de bloc quelconque :

L'objectif pour un site quelconque est d'obtenir à la fin, par exemple le genre d'organisation ci-dessous (après application d'un style CSS) :



Rappel : le rôle de l'HTML est de donner le contenu de la page web (le texte, les images, …) et de le structurer, mais pas de le mettre en forme (cela sera fait dans un fichier css). Même en mettant ces balises dans votre code, il faudra attendre le css pour positionner les blocs les uns par rapport aux autres. Ici les balises n'ont qu'un rôle sémantique, elles ne font que donner du sens aux blocs à la lecture du code. Par exemple, quand on lit « footer » on comprend que ce sera un bloc de bas de page, mais écrire footer ne va pas forcer le bloc à se trouver en bas de page.
Historiquement, on ne trouvait dans les sites que des blocs « div » qui définissaient des blocs indépendamment de leur rôle.

Dans le bloc « body », ajouter un bloc « main » contenant le texte actuel de la page ainsi qu'un bloc « footer » contenant le texte « Site réalisé par Bob Dansleciel. ». Les 2 textes doivent être séparés par une ligne vide.

2.2 Structurer le texte

Pour structurer le texte, différentes balises sont disponibles :

- pour définir un paragraphe
-
 : pour revenir à la ligne dans un paragraphe
- <h1> un titre principal </h1>
  <h2> un sous titre </h2>
  … jusqu'à …
  <h6> un sous sous sous …. </h6>
- déclaration d'une « unordered list »
   élément de la liste
   autre élément de la liste

- déclaration d'une « ordered list »
   élément de la liste
   autre élément de la liste

Testez les différentes balises ci-dessus.

Question 3

Quelle différence y a-t-il entre une <ul> et une <ol> ?

ul : liste non organisée et ol c'est les listes organisées

Définir comme titre principal la première phrase et comme titre de rang 2 la 2eme phrase de la page.

Vous avez peut être remarqué que lorsque vous appliquez des titres, ils apparaissent avec une certaine mise en forme dans le navigateur alors que html n'est pas sensé gérer la mise en forme. Il s'agit en fait d'une mise en forme par défaut appliqué par le navigateur afin que le site puisse être affiché convenablement même s'il n'y a pas de feuille de style CSS.

2.3 Mettre en valeur des éléments

Des balises permettent d'indiquer au navigateur que certains mots ont une importance particulière.

 marque la particularité d'un texte, indique une citation, un titre de livre
 met en avant un texte, le fait ressortir
<mark> attire l'attention sur un texte </mark>

Testez les différentes balises ci-dessus.

Question 4

Quelle est la mise en forme appliquée, par défaut (modifiable par le CSS), pour ces 3 balises ?

em italique , strong gras , mark surligne

Dans votre page, faire apparaitre votre nom en italique.

2.4 Insérer des liens

Les liens sont au cœur du web, ils permettent de se déplacer de pages en pages à l'intérieur d'un site ou entre différents sites. Ils utilisent les balises :

 texte du lien

Remarque :
- Le « chemin du lien » peut être une url ou bien un chemin relatif dans l'arborescence du site par rapport à l'endroit où se trouve le fichier html dans lequel se trouve le lien
- Un attribut facultatif supplémentaire target="_blank" permet de forcer l'ouverture du lien dans un nouvel onglet ou une nouvelle fenêtre (selon les réglages du navigateur).

Créer un nouveau fichier html nommé cv-en-construction.html contenant le texte suivant (à adapter bien sûr) en utilisant une balise paragraphe :

Bonjour. Je m'appelle Bob Dansleciel, je suis étudiant en BTS CIEL (Cybersécurité, Informatique et Réseaux, Electronique).
Je recherche un stage de 6 semaines à partir du mois de mai.

Créer un lien de la page index.html vers cette nouvelle page quand on clique sur « suivez le lien ».

3 Mettre en forme le site avec CSS

Le langage CSS (Cascading Style Sheet) va maintenant être utilisé pour mettre en forme le contenu définit dans le code HTML.

Une bonne habitude consiste à distinguer 2 parties dans le fichier CSS (on peut même faire 2 fichiers distincts) :
- une partie « chartre graphique » dans laquelle on va définir les tailles et polices d'écriture, les couleurs, appliquer différents effets, …
- une partie « structure » dans laquelle on gère le positionnement des zones de la page.

Dans le même dossier que vos fichiers html, créez un fichier « styles.css » et ouvrez-le avec VSCode.

Afin de faire le lien entre le fichier html et le fichier css, ajoutez la balise suivante dans le <head> du fichier index.html.

```
<link rel="stylesheet" href="styles.css">
```

3.1 Syntaxe d'une règle CSS

Le principe de base du css est de définir des règles dans lesquelles **un sélecteur** indique les éléments de la page pour lesquels on affecte **des valeurs** à **des propriétés.**
La syntaxe d'une règle est :

```
sélecteur {
    propriete1: valeur1;
    propriete2: valeur2;
    propriete3: valeur3;
}
```

3.2 Modification des couleurs

Ajouter dans votre fichier css :

- une règle permettant de sélectionner tous les titres principaux (h1) pour les mettre dans une couleur (color) à partir de son nom.
  Il existe :
  - 16 couleurs nommées standards prisent en charge par tous les navigateurs (white, silver, gray, black, red, maroon, lime, green, yellow, olive, blue, navy, fuchsia, purple, aqua, teal)
  - 140 couleurs nommées prisent en charge par les navigateurs récents

  https://htmlcolorcodes.com/fr/noms-de-couleur/
- une règle permettant de colorer le texte contenu dans <body> à partir du code RGB de la couleur, exemple : rgb(12, 200, 0)
- une règle permettant de colorer tous éléments en italique (em) à partir du code hexadécimal de la couleur, exemple : #45A1BB

Vous pouvez voir que tout le texte de votre page prend la couleur que vous avez défini en RGB, à l'exception des éléments <h1> et . Cela est dû au principe d'**héritage** qui régit le fonctionnement de CSS. Tous les éléments qui sont dans <body> récupèrent par défaut sa mise en forme, à moins qu'ils n'aient leur propre mise en forme (comme <h1> et ici).

Avec la propriété background-color, créer la règle permettant de modifier la couleur du fond d'écran de toute la page.
Remarque : background-color permet de changer la couleur de fond de n'importe quel élément.

Compléments sur les sélecteurs

Si on souhaite appliquer une mise en forme à un ou plusieurs éléments du même type, mais pas à tous, par exemple mettre un seul paragraphe d'une certaine couleur, il faut :

- Dans HTML, l'identifier avec l'un des 2 attributs ci-dessous :
  - <p class="un-nom-qui-peut-être-commun-a-plusieurs-éléments">
  - <p id="un-nom-unique-sur-la-page">
- Dans CSS, utiliser le sélecteur approprié :
  - .nom-de-class
  - #nom-de-id

Il est possible de combiner les sélecteurs pour cibler certains éléments :

- h3, em : sélectionne tous les éléments <h3> et tous les éléments
- h3 em : sélectionne tous les éléments situés à l'intérieur d'un élément <h3>
- h3 + p : sélectionne le premier élément <p> situé après un titre <h3>.
- a[title] : sélectionne tous les liens <a> qui possèdent un attribut title
- a[title="Cliquez ici"] : sélectionne les liens <a> qui possèdent un attribut title valant "cliquez ici"
- a[title*="ici"] : sélectionne les liens <a> qui possèdent un attribut title qui contient "ici"

Mettez en place quelques règles de coloration pour votre page d'accueil temporaire.

Validez tout votre code source à l'aide du site https://validator.w3.org/

Une première version temporaire de votre site est maintenant prête. Nous allons voir comment gérer les versions du code source et mettre votre CV en ligne.

4 Gérer son code source avec GIT/Github et mettre en ligne le site

4.1 GIT, c'est quoi ?

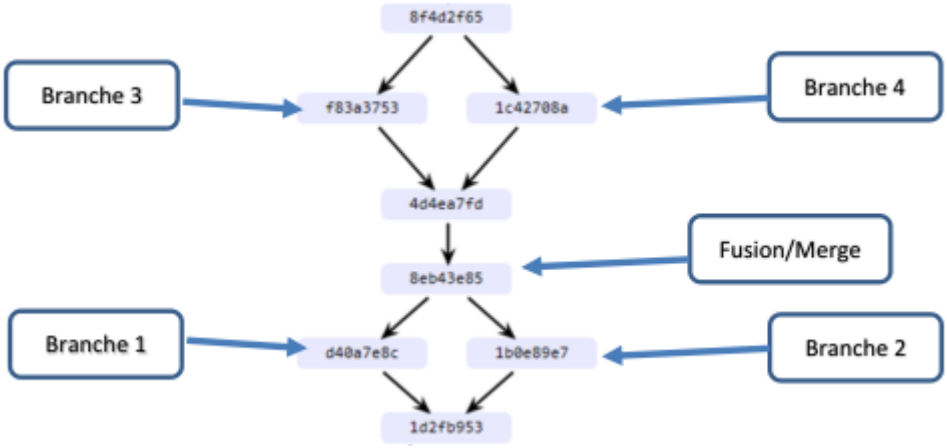
GIT est un **logiciel de contrôle de version** (*version control system* ou VCS en anglais). C'est de loin le plus connu et utilisé à l'heure actuelle.

Son rôle est de garder une trace de la progression de son développement en gardant une sauvegarde des principales étapes. Il permet de voir quelles modifications ont été faites, quand, et par qui si le développement se fait à plusieurs.

On créé **un dépôt (*repository* ou « repo »)** qui va garder l'ensemble des fichiers et/ou dossiers.
Ce dépôt peut être :

- en local sur sa machine si on travaille seul
- sur un serveur local pour travailler à plusieurs
- sur un site tel que **Github** par exemple pour que le code soit en permanence sauvegardé dans le cloud, accessible depuis n'importe où, pour faire du travail collaboratif.

Si plusieurs développeurs travaillent en même temps, chacun travaillera sur sa **branche**.



Une fois son travail terminé, un développeur **livre** sa version et une **fusion** (*merge* en anglais) sera faite pour que tout le monde puisse continuer à travailler sur le code complet et à jour.

Dans cette activité nous allons travailler avec Github afin de stocker votre code en ligne ce qui permettra également ensuite de rendre votre CV accessible sur Internet. Cela nécessite tout de même d'avoir GIT en local sur son PC pour avoir un dépôt local des fichiers qui se synchronise quand on le souhaite avec les fichiers du dépôt distant.

4.2 Déposer le CV sur Github

Aller sur le site https://github.com/ et se créer un compte.
Remarque : si vous mettez un pseudo comme nom d'utilisateur, choisissez en un « sérieux » car ce compte Github pourra être votre future carte de visite quand vous rechercherez un stage, une alternance ou un emploi…

Créer un nouveau dépôt (*repository*) de type « publique » nommé « monCV ».

Dans la partie « Quick setup », demander à uploader un fichier existant afin de sélectionner tous les fichiers de votre site. Avant de valider le commit avec le bouton vert, penser à mettre un commentaire expliquant le commit : ici ce sera par exemple « dépôt fichiers initiaux ».

Un **commit** est une capture de l'état de votre code a un instant donné. On fait un commit lorsque l'on a terminé une certaine modification dans le code et que l'on souhaite indiquer au dépôt que l'on veut garder une trace de cet avancement. Ce n'est pas un bouton de sauvegarde sur lequel on appuie n'importe quand.

Vous devez maintenant voir :

- que vos fichiers sont dans le dépôt
- qu'il y a une branche de développement nommée « main »
- que l'on vous propose d'ajouter un fichier readme afin d'indiquer aux personnes qui arrivent sur ce dépôt ce qu'ils vont y trouver

Créer le fichier readme en y ajoutant par exemple :

Mon CV en ligne
Mon premier projet HTML – CSS

4.3 Installer GIT

Bien que nous utilisions Github, nous devons installer GIT sur notre ordinateur afin que les fichiers puissent se synchroniser entre les deux.

S'il ne l'est pas déjà, installer GIT sur votre ordinateur avec les options par défaut :
https://git-scm.com/download/win

Configurer GIT avec votre mail et votre nom d'utilisateur Github avec les commandes suivantes à taper dans l'invite de commande :

- git config --global user.email "vous@exemple.com"
- git config --global user.name "votre nom sur github"

4.4 Cloner en local les fichiers déposés sur Github

Les fichiers sont maintenant dans Github, mais ils ne sont pas en lien avec ceux se trouvant sur le PC.
Nous allons cloner sur le PC les fichiers se trouvant sur Github afin de créer un lien entre Github et notre dépôt local.

Dans Github, dans le dépôt monCV, cliquer sur le bouton vert « Code » afin de copier l'url https correspondant au dépôt.

Dans l'invite de commande Windows, se placer dans l'arborescence sur le serveur de la classe dans le dossier à votre nom dans un sous-dossier « web ».
Taper la commande : git clone suivi de l'url précédemment copiée dans Github.

Vous devez maintenant avoir un dossier monCV sur le serveur. Si vous affichez les fichiers cachés de ce dossier, vous devez voir un fichier .git vous indiquant que ce dossier est bien suivi par GIT.

Supprimer les fichiers html et css que vous aviez créés à l'origine qui sont maintenant inutiles (pas ceux que l'on vient de cloner).

Il aurait été possible (et même plus simple) de demander directement à VSCode de mettre nos fichiers sur Github en créant le lien entre le dépôt local et le dépôt distant. Le but était de vous montrer ici aussi l'interface côté Github.

4.5 Modifier un fichier Github depuis VSCode

Ouvrir le dossier monCV dans VSCode, et modifier l'un des fichiers, par exemple une couleur dans le CSS.

Dans VSCode, c'est le menu à gauche « Source Control » qui donne accès à l'outil GIT.
Dans « Source control », vous devez observer qu'un « M » indique que le fichier a été modifié par rapport à celui se trouvant dans le dépôt GIT local.

Dans Source Control :

- cliquer sur la ligne correspondant au fichier afin de visualiser la modification
- valider le commit en mettant un commentaire approprié
- confirmer le pop-up de warning
- valider la synchronisation des changements pour que le commit remonte sur Github
- vérifier sur Github que le changement est bien pris en compte.

Encore une fois, attention de ne pas utiliser le commit comme un simple bouton de sauvegarde. Vous ne devez faire des commits que sur des évolutions notables, terminées, du code (ajout d'une fonctionnalité, correction de textes, ensemble de modifications de couleurs,…), pas juste parce que vous ne voulez pas perdre ce que vous avez fait et que c'est l'heure d'aller manger.

4.6 Mettre un site en ligne sur Github

Maintenant que vos fichiers sont sur Github, nous allons utiliser le service Github Pages proposé par le site. Il s'agit d'un service d'hébergement de site statique qui va permettre de rendre votre CV accessible sur Internet.

Dans Github, sur la page du dépôt monCV :

- aller sur l'onglet « Settings » puis le menu « Pages »
- choisir un « déploiement à partir d'une branche » et sélectionner la branche « Main » puis sauvegarder.
- Cliquer à nouveau sur le menu « Pages », vous devez voir apparaitre l'url de votre CV en ligne, il n'y a plus qu'à tester qu'elle fonctionne !

4.7 Création d'une branche de développement

Actuellement les fichiers sur Github servant pour l'affichage du site sont dans la branche Main. Cela signifie que tout nouveau commit sur le dépôt sera immédiatement répercuté sur le site … ce que l'on ne veut pas forcément. On souhaite pouvoir contrôler à quel moment le site visible du public évolue.
Pour cela, nous allons créer une nouvelle branche spéciale pour le développement du site.
Cette manipulation est possible aussi bien depuis Github que depuis VSCode.

- Depuis l'onglet Source Control de VSCode, créer une nouvelle branche nommée « dev » que vous « pushez » immédiatement sur Github (dans GIT, l'action « push » fait remonter une modification du dépôt local vers le dépôt distant).
- Vérifier dans Github que la branche est bien présente et la sélectionner
- Vérifier que cette branche contient bien les mêmes fichiers que la branche Main

Maintenant il est possible de travailler sur les fichiers de la branche dev, sans impacter ceux de Main.

- Dans VSCode, faire une modification quelconque dans un fichier puis faire un commit + push vers la branche dev.
- Dans Github, vérifier que le fichier dans la branche dev est modifié mais pas celui de la branche Main.
- Vérifier que le site en ligne n'a pas bougé.

5 Structurer une page web avec CSS

La page mise en ligne ne contient qu'un site temporaire le temps que votre vrai CV soit prêt. Il faut donc s'y mettre …

Nous allons maintenant travailler dans le fichier cv-en-construction de la branche dev afin de préparer la structure du CV.

Si vous ne l'avez pas déjà fait, associer la feuille de style CSS au fichier cv-en-construction.html.

5.1 Format des blocs

Parmi les balises vues jusqu'à présent, il existe :

- des balises bloc : elles définissent une zone sur la page et provoque un retour à la ligne (paragraphe, titre, section, ….)
- des balises inline, qui sont inclues dans une balise bloc, et qui ne provoque pas de retour à la ligne (strong, lien, …).

Pour visualiser l'effet des propriétés suivantes, définissez une couleur de fond au bloc paragraphe (ajouter les balises nécessaires si besoin) contenant le texte de la page cv-en-construction.

Il existe des propriétés permettant d'imposer les dimensions des balises de type bloc, soit de façon fixe, soit relativement aux dimensions de l'écran :

```
height : 20 px;
width : 50 %;
```

On peut spécifier des dimensions minimales et maximales pour la largeur et la hauteur :

```
min-width : 500px
max-height : 300px
```

Les blocs ont une bordure ainsi que des marges intérieurs (padding) et extérieurs (margin) que l'on définit en pixel :

```
p {
        border: 2px solid black ;
        padding: 5px;
        padding-top : 10px;
        margin: 10px;
        margin-left : 30px;
}
```

Pour centrer un bloc sur la page, celui-ci doit avoir une largeur indiqué, et on spécifie margin : auto.

Tester les instructions précédentes.

5.2 Mise en place des blocs

La mise en place des blocs sur la page s'effectue de façon moderne avec les techniques Flexbox ou Grid.
D'autres méthodes ont été utilisées par le passé : utilisation de tableaux, ou l'utilisation de certaines propriétés CSS mais elles ne sont plus préconisées aujourd'hui.

Flexbox est à privilégier quand on veut placer des blocs en ligne les uns par rapport aux autres.
Grid est efficace dans des structures plus complexes avec un positionnement en 2 dimensions des blocs.
Dans cette activité nous allons tester Flexbox, ceux qui le souhaitent pourront se renseigner sur Grid.

Le principe consiste en la définition d'un conteneur dans lequel sont placés des blocs.
Les différentes balises body, header, section, main … peuvent être le conteneur d'autres balises bloc.
Il est également possible d'utiliser la balise bloc qui créée un bloc mais sans signification particulière.

Pour tester l'utilisation de Flexbox, supprimer le contenu de la page puis définissez dans le body 3 blocs de 300 pixels de large sur 100 pixels de haut ayant chacun une couleur de fond différente.
Définissez une hauteur de 100% pour les balises html et body.

Le body servira de conteneur et doit être défini comme flexbox dans CSS en lui appliquant la propriété :

```
display: flex ;
```

La propriété display, comme toutes les autres, s'applique en fonction d'un sélecteur. N'oubliez pas que si besoin vous pouvez créer des sélecteurs ciblant précisément certains éléments en utilisant des attributs class ou id. Cela permet par exemple de définir une certaine section comme étant un conteneur, sans que toutes les sections soient des conteneurs.

- Choix de la direction et l'ordre des blocs (à mettre pour le bloc conteneur)
    - flex-direction: column;
    - flex-direction: row-reverse;
    - flex-direction: column-reverse;

- Choix de l'alignement des blocs (à mettre pour le bloc conteneur)

Que les blocs soient en ligne ou en colonne, on va placer les éléments selon l'axe principal.

  - justify-content : flex-start;
  - justify-content : flex-end;
  - justify-content : center;
  - justify-content : space-between;
  - justify-content : space-around;

On peut les placer également suivant l'axe secondaire, perpendiculaire à l'axe principal des blocs.

  - align-items: stretch;
  - align-items: flex-start;
  - align-items: flex-end;
  - align-items: center;

- Grossissement des blocs (à mettre pour le bloc contenu)

La propriété flex permet de faire grossir un élément afin qu'il occupe tout l'espace disponible.

Par exemple si un bloc à la propriété flex: 1; il prend tout l'espace disponible.
Si un bloc est flex: 1 et qu'un autre est flex: 2, alors le second peut grossir 2 fois plus que le premier.

Tester les propriétés précédentes pour comprendre le fonctionnement de Flexbox.

Vous avez maintenant la plupart des outils de base permettant de créer votre vraie page de CV.

Réfléchir à la structure du CV que vous souhaitez faire et la mettre en place à l'aide de Flexbox :

- Pour les différentes zones de votre page, n'oubliez pas d'utiliser les blocs header, main, footer, … vus au point 2.1, plutôt que de simple
- Comme précédemment vous pouvez mettre des fonds de couleur différents pour bien visualiser les blocs et vous aider à les mettre en place.
- Il est possible de mettre des Flexbox dans des Flexbox si besoin.
- A ce stade, il ne doit y avoir presque pas de contenu dans les différents blocs, il s'agit juste de positionner et dimensionner les différentes zones constituant votre page.

Quand votre structure est prête, vérifier votre code sur https://validator.w3.org/ et faire un commit de votre code sur la branche dev.

Ajouter les principales informations des différentes rubriques (sans rentrer dans tous les détails pour le moment), vérifier la qualité de votre code, puis faire un commit sur la branche dev.

5.3 Fusion de la branche dev sur la branche Main

A ce stade le CV commence à ressembler à quelque chose, nous allons donc mettre à jour la page cv-en-construction de la branche Main afin que les visiteurs de votre site puissent voir l'avancement de la construction.
Pour cela il faut faire une fusion (merge) de dev sur Main.

Dans Github :

- aller sur la branche dev, puis cliquer sur « Compare & Pull request ». Une pull request est une demande de fusion.
- Après avoir éventuellement mis un commentaire à la demande, cliquer sur « Create pull request »
- S'il n'y a pas de conflit entre les branches, la fusion est possible, cliquer alors sur « Merge pull request ».
- Cliquer sur « Confirm merge »
- Vous pouvez vérifier que Main est maintenant identique à dev et que votre site en ligne est modifié (ça prend quelques minutes parfois).

6 A vous de jouer

Votre CV prend forme mais n'est pas encore très rempli, très beau, très …

A vous de travailler, sur la branche dev, pour améliorer le contenu, la mise en forme, …
Nous avons vu ici quelques balises html et quelques propriétés CSS, à vous d'être curieux et de chercher sur internet comment faire ce que vous voulez faire.

Vous pouvez par exemple :

- Ajouter une photo de vous
- Jouer sur les polices de caractère utilisées
- Jouer sur la taille des caractères
- Ajouter un lien vers les sites web des établissements que vous mentionnez
- Ajouter le contenu de la formation de 1ere année de BTS afin que les recruteurs sachent ce que vous saurez faire pour le stage
- Permettre le téléchargement d'un fichier pdf de votre CV (pas le CV du site, mais un fichier pdf fait de façon classique)

N'oubliez pas votre ami : https://validator.w3.org/

Quand votre CV est prêt il faut basculer tout le code final dans le fichier index.html afin que ce soit directement votre beau CV qui s'affiche et non plus la page temporaire que l'on avait créée.

Si vous utilisez réellement ce CV en ligne pour votre recherche de stage, il est évidemment important d'en soigner la qualité.
Il est également important de soigner la qualité (indentation, choix des blocs, …) du code source que vous stockez sur Github car vous pouvez mettre votre page Github en lien dans votre CV afin que les personnes qui vont vous recruter (stage, alternance après le BTS, emploi) voient comme vous savez faire du beau code…