

JUSTIN

Présentation orale

HEBERGEMENT – SITE WEB

Ilia – Justin – Mohamed – Valentin

JUSTIN

Au Programme

Les sujets traités :

1

API

2

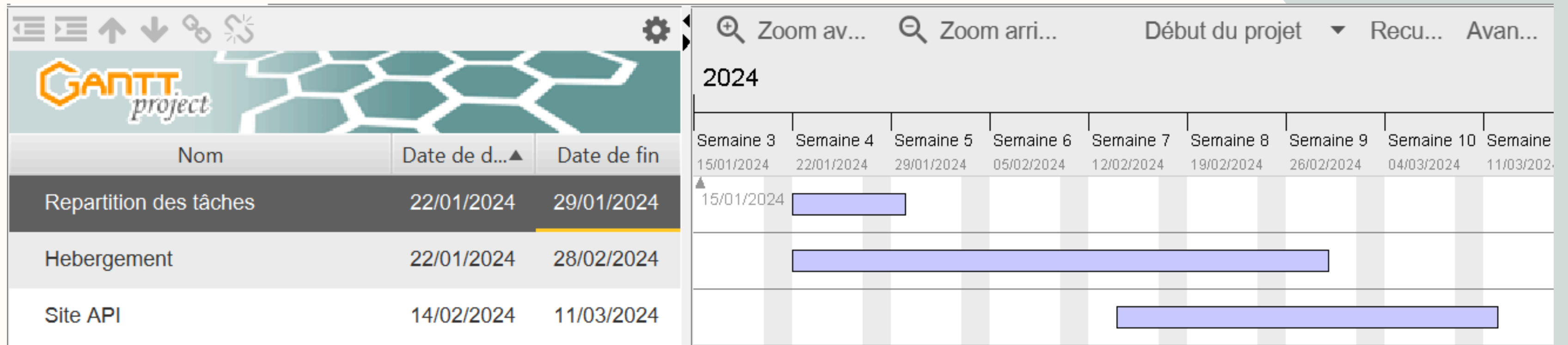
Hebergement

3

Site web

JUSTIN

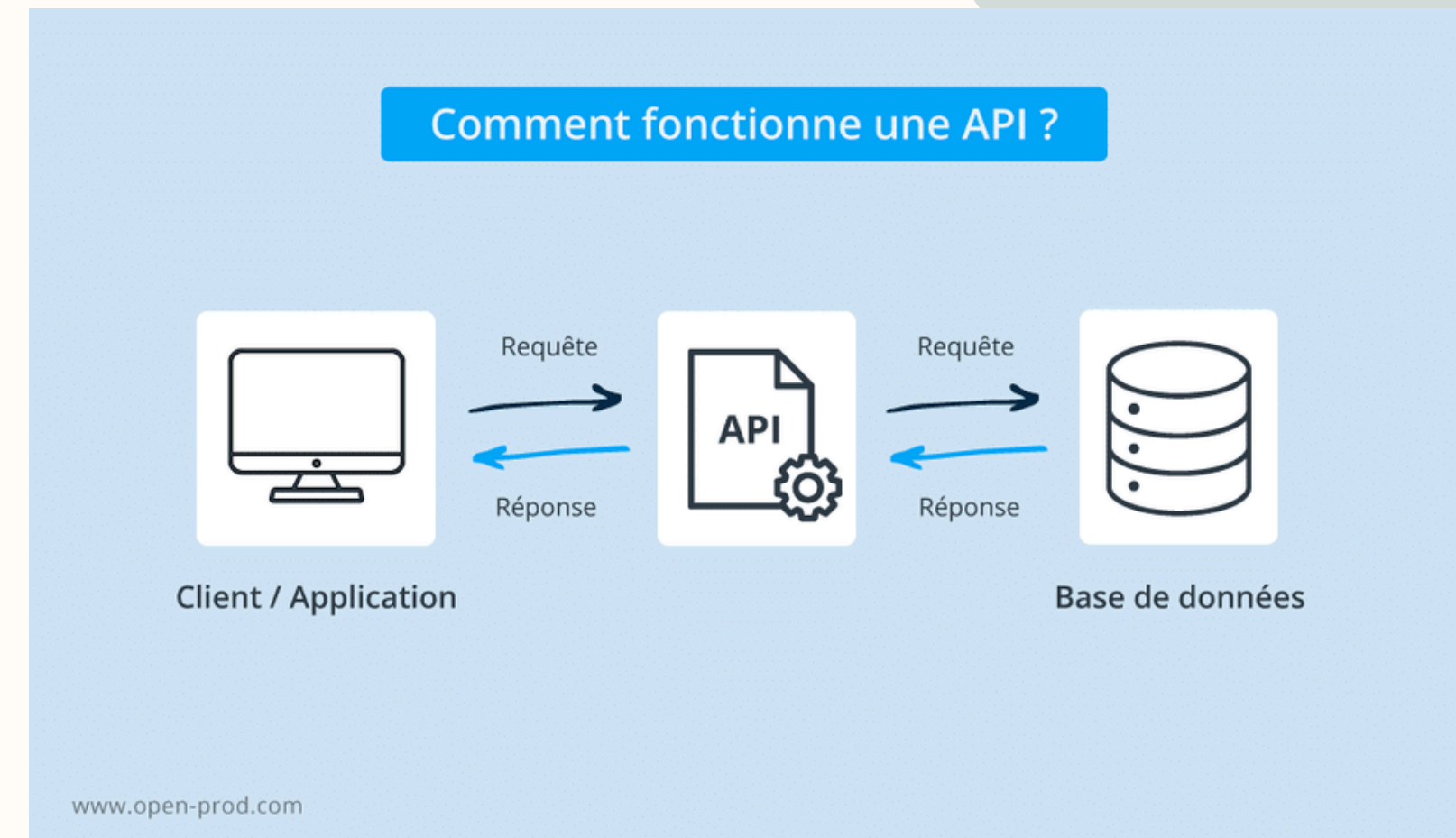
Gantt



API

Qu'est-ce qu'une API ? (Application Programming Interface)

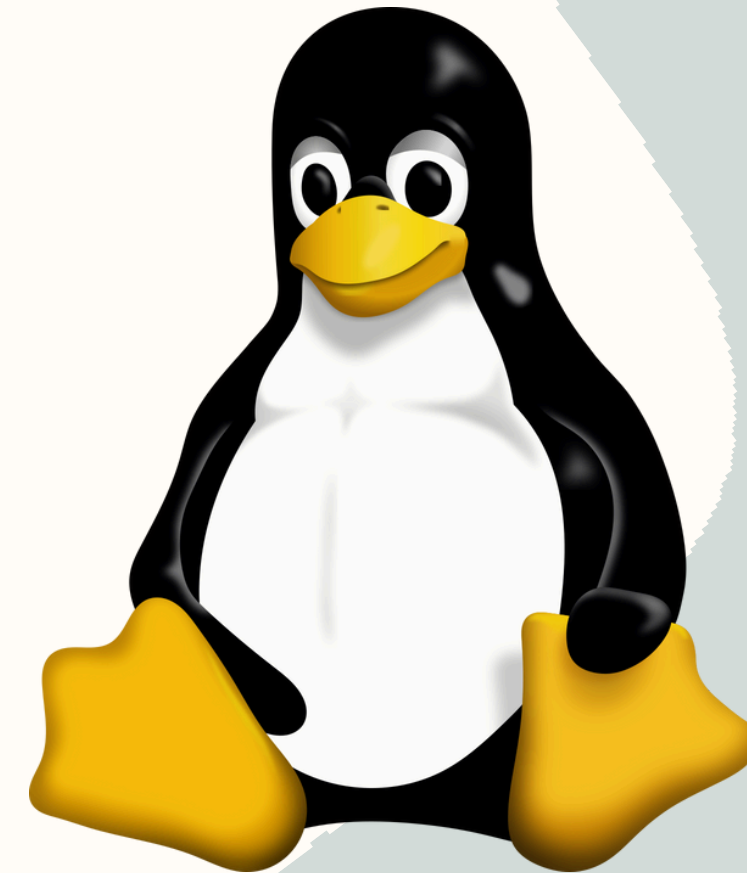
- Définition : Ensemble de règles permettant à des applications de communiquer entre elles.
- Utilité :
 - Simplifie le développement en réutilisant des fonctionnalités.
 - Permet l'interopérabilité entre différents systèmes et logiciels.
 - Automatise les tâches et étend les services via la connectivité.



ILIAN

Linux

Linux est un système d'exploitation open-source inspiré de UNIX, utilisé dans le monde entier pour les serveurs, les ordinateurs de bureau, les smartphones et les appareils intégrés



VAL

Ubuntu

Ubuntu est une distribution Linux fondée
sur Debian



Pourquoi choisir Linux et non pas Windows ?

VAL

Sécurité :

- Ubuntu est reconnu pour sa robustesse en matière de sécurité.
- Mises à jour régulières de sécurité pour combattre les dernières menaces

Coûts :

- Ubuntu est un système d'exploitation open-source, donc gratuit à télécharger et à utiliser.
- Réduction des coûts de licence et d'exploitation par rapport à Windows.



Hébergement Local :

Installation sous Ubuntu/Debian

Installation de Apache :

```
sudo apt install apache2  
sudo service apache2 start
```

Installation de PHP et MYSQL :


```
sudo apt install apache2 php libapache2-mod-php mysql-server php-mysql
```

On a installer PHP et MYSQL pour gérer les bases de données si on décide de publier le site

Momo

Une fois que tout fonctionne on aura
cette page en écrivant
localhost/phpmyadmin si la barre de
recherche

Verification

**ubuntu**

Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/  
|-- apache2.conf  
|   |-- ports.conf  
|-- mods-enabled  
|   |-- *.load  
|   |-- *.conf  
|-- conf-enabled  
|   |-- *.conf  
|-- sites-enabled  
|   |-- *.conf
```



Accéder au site depuis une autre machine du même réseau local :

- Avoir son adresse IP
- On a ensuite ouvert le port 80 ,
qui est le port utilisé pour le
trafic web HTTP

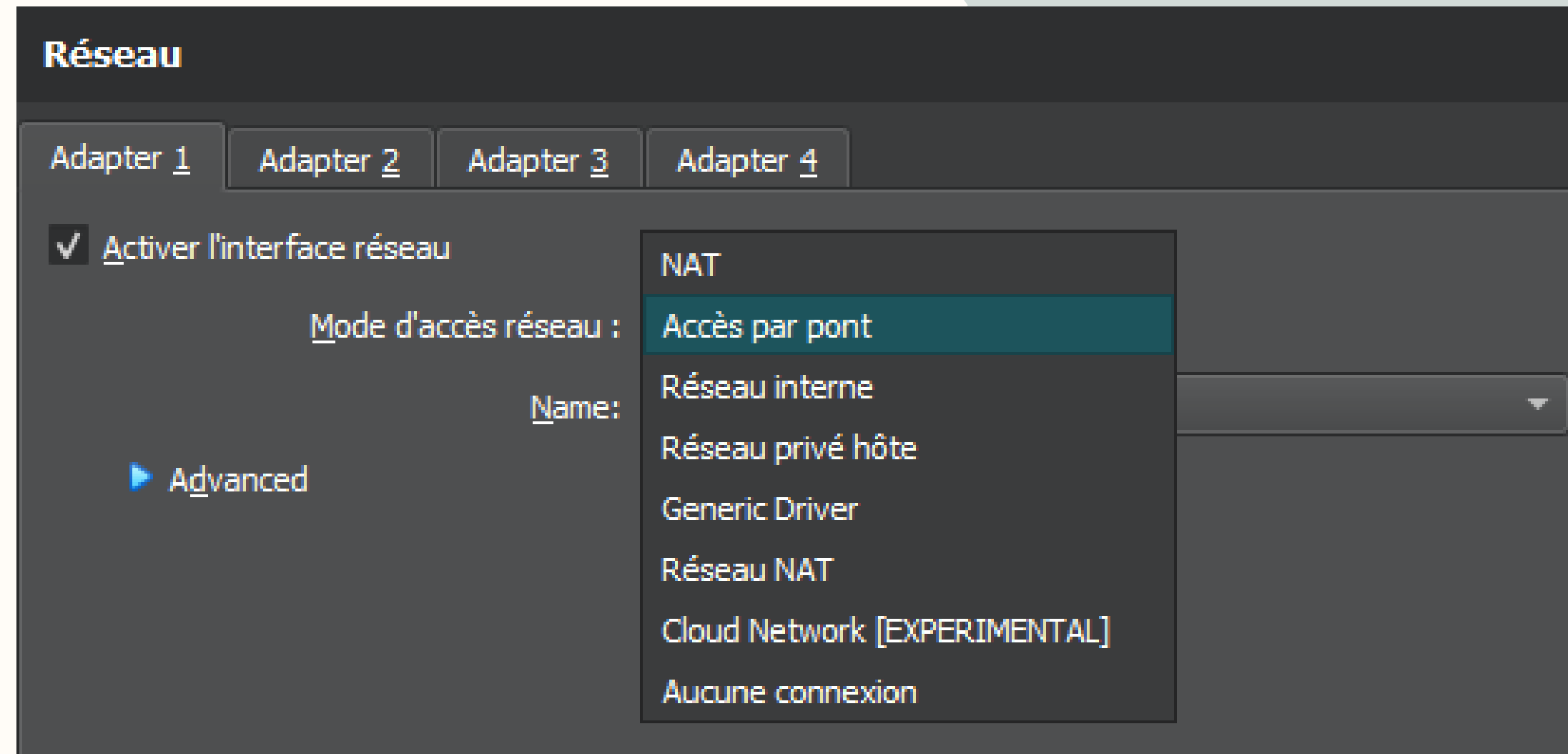
Pour avoir simplement l'adresse IP de l'ordinateur sur le réseau, depuis un terminal

```
hostname -I
```

```
status: active  
[linuxconfig.org]$ sudo ufw allow 80  
Rule added  
Rule added (v6)
```

Accéder au site depuis une autre machine
du même réseau local :

- Activer le Mode Bridge :













Justin/

RICK & MORTY

Val

Rick & Morty

Page précédente Page suivante Genre : Tous Espèce : Toutes Appliquer les filtres

				
Rick Sanchez	Morty Smith	Summer Smith	Beth Smith	Jerry Smith
				



Conception du site :

1

Header

2

Implementation de l'api

3

Pop-Up

4

Filtre

Val

Header

Attribut : Sticky

Elle permet à la barre de filtres de se comporter comme un élément position: static; jusqu'à ce qu'il atteigne un certain point lors du défilement (défini par top: 0;),

```
.controls {  
    position: sticky;  
    top: 0;  
    background-color: #000;  
    z-index: 1000;  
    text-align: center;  
    margin: 0 auto;  
    padding: 10px 0;  
    width: 100%;  
}
```


Justin Implémentation de L'API

- Déclarer une fonction pour récupérer les personnages de l'API
- 2 conditions qui permettent de changer l'URL

```
function fetchCharacters(page) {  
  let url = `https://rickandmortyapi.com/api/character/?page=${page}`;  
  
  if (currentGenderFilter) {  
    url += `&gender=${currentGenderFilter}`;  
  }  
  
  if (currentSpeciesFilter) {  
    url += `&species=${currentSpeciesFilter}`;  
  }  
}
```

On a ensuite le `fetch(url)`, qui va envoyer la requête à l'api

```
fetch(url)  
  .then(response => response.json())
```

Justin Implémentation de L'API

3 variables :

- 1 variables servant pour le style CSS
- 2 autres variables qui vont servir a afficher le nom et l'image du personnage



```
const characterElement = document.createElement('div');
characterElement.classList.add('character');
const image = document.createElement('img');
image.src = character.image;
characterElement.appendChild(image);
const name = document.createElement('h3');
name.textContent = character.name;
characterElement.appendChild(name);
```


POP-UP

On a déclaré les variables pour la pop-up qui vont simplement reprendre les attributs de la variable initial et y rajouté un style CSS

```
characterElement.addEventListener('click', () => {  
  document.getElementById('popupName').textContent = character.name;  
  document.getElementById('popupImage').src = character.image;  
  document.getElementById('popupStatus').textContent = 'Status: ' + character.status;  
  document.getElementById('popupSpecies').textContent = 'Species: ' + character.species;  
  document.getElementById('popupGender').textContent = 'Gender: ' + character.gender;  
  popup.style.display = 'block';  
})
```

On y a configuré un bouton close qui change de couleur quand y met le curseur dessus

```
.close-button {  
  color:  #aaa;  
  float: right;  
  font-size: 28px;  
  font-weight: bold;  
}  
.close-button:hover,  
.close-button:focus {  
  color:  black;  
  text-decoration: none;  
  cursor: pointer;  
}
```

Les Filtres

On a commencé par select tout ce qu'on veut dans les filtres :

```
Genre :
<select id="genderFilter">
  <option value="">Tous</option>
  <option value="Male">Homme</option>
  <option value="Female">Femme</option>
  <option value="Genderless">Sans genre</option>
  <option value="unknown">Inconnu</option>
</select>
Espèce :
<select id="speciesFilter">
  <option value="">Toutes</option>
  <option value="Human">Humain</option>
  <option value="Alien">Alien</option>
  <option value="Robot">Robot</option>
  <option value="unknown">Inconnue</option>
</select>
```

Puis créer des variables pour récupérer les valeurs mises dans les ID

```
const genderFilterSelect = document.getElementById('genderFilter');
const speciesFilterSelect = document.getElementById('speciesFilter');
```

Les Filtres

On va mettre à jour les variables de filtre puis retourner à la page 1 de la présentation des personnages

```
applyFiltersButton.addEventListener('click', () => {  
  currentGenderFilter = genderFilterSelect.value;  
  currentSpeciesFilter = speciesFilterSelect.value;  
  currentPage = 1;  
  fetchCharacters(currentPage);  
});
```

L'URL qui va changer est celle utilisée dans les requêtes de l'API

```
if (currentGenderFilter) {  
  url += `&gender=${currentGenderFilter}`;  
}  
  
if (currentSpeciesFilter) {  
  url += `&species=${currentSpeciesFilter}`;  
}
```



MERCI !