

Tutorat Github/Web

JACQUET Pierre : pierre.jacquet@etu.u-bordeaux.fr

JUNG Frédéric : frederic.jung@etu.u-bordeaux.fr



Démarrer un projet avec Github

Initialisation

- Créer un compte github
- Créer un nouveau *Repositories*



- Vous pouvez ensuite cloner le dépôt : `git clone http://www.url/du/repositories`

Envoyer son premier commit

Vous allez créer 3 nouveaux fichiers dans la racine du nouveau *repositories* :

- index.html
- template.css
- main.js

Ensuite vous allez enregistrer ces changements sur la *branch* courante (master par défaut) :

```
git add main.html template.css main.js
git commit -m "init files"
```

Puis envoyer ces modifications sur votre hébergeur (ici github)

```
git push
```

Ajouter des collaborateurs

Vous pouvez ajouter des collaborateurs pour travailler en groupe. Ils pourront eux aussi cloner le *repository*, créer de nouveaux commits et les synchroniser avec Github .

[Code](#) [Issues 0](#) [Pull requests 0](#) [Projects 0](#) [Wiki](#) [Insights](#) [Settings](#)

[Options](#)
Collaborators
[Webhooks](#)
[Integrations & services](#)
[Deploy keys](#)

Collaborators Push access to the repository

This repository doesn't have any collaborators yet. Use the form below to add a collaborator.

Search by username, full name or email address

You'll only be able to find a GitHub user by their email address if they've chosen to list it publicly. Otherwise, use their username instead.

[Add collaborator](#)

Héberger votre application web sur github

Vous pouvez héberger votre application web avec github en activant les *Github pages*

GitHub Pages

GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.

Source

GitHub Pages is currently disabled. You must first add content to your repository before you can publish a GitHub Pages site. [Learn more.](#)

None ▾

Save

Theme Chooser

Select a theme to publish your site with a Jekyll theme using the master branch. [Learn more.](#)

Choose a theme

Développement web

Ajouter le code suivant à `index.html` :

```
<!doctype html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Template HTML de base</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1 id="#coucou" class="swag2poule">Coucou!</h1>
    <p class="swag2poule">Vive internet!</p>

    <script src="script.js"></script>
  </body>
</html>
```

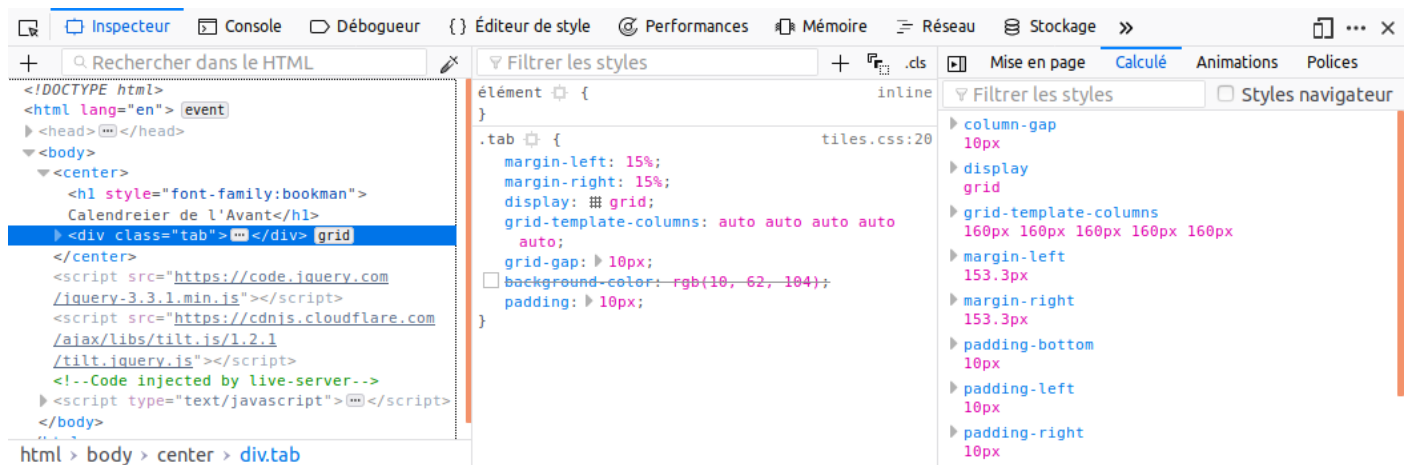
Ceci est un template basique et fonctionnel. En l'état vous n'aurez pas besoin de beaucoup plus pour créer une page web. On retrouve 3 points importants:

- la balise **HTML** qui encapsule les deux suivantes
- la balise **head**
- la balise **body**

1. Votre meilleure amie: la console développeur.

- 1) Ouvrez le fichier index.html dans votre navigateur (Chrome/Firefox c'est le mieu pour ce qu'on va faire).
- 2) Appuyez sur **F12** ou faites **Clique-droit "Examiner l'élément"**.

Vous avez en face la console WEB. C'est un ensemble d'outils extrêmement pratique pour développer. En faite si vous l'utilisez bien vous pouvez expérimenter de nombreuses choses directement depuis la console et apprendre le css notamment beaucoup plus rapidement.



Pour vous montrer à quoi ressemble la console des développeurs

- 3) Cliquez sur la balise **h1** dans la console dev. Regardez dans la section style/css qui devrait être présent dans la console. Créer une nouvelle règle pour mettre le texte en couleur.
- 4) Créer une nouvelle règle, mais cette fois-ci essayez de faire apparaitre avec la flèche bas du clavier le menu déroulant permettant de voir toutes les règles CSS existante. C'est extrêmement pratique cela permet de savoir tout ce que vous pouvez écrire comme règle.

Si vous actualisez la page, la couleur aura disparu. En effet, la console n'écrit rien "dans" le fichier. C'est seulement un outils "**bac à sable**" pour expérimenter ce que vous écrirez après dans le fichier.

Les règles/méthodes/manières permettant de créer un site web sont extrêmement vaste. Le meilleur moyen d'apprendre à coder rapidement en HTML/CSS c'est de renoncer à apprendre tout par coeur. N'hésitez surtout pas à regarder sur **stackoverflow** comment les autres ont pensé chaque bout de code.

De la même manière pour créer un "joli site", soyez attentif aux détails. Regardez quelque chose qui vous plait sur un site, et cherchez à le reproduire en regardant la règle qui permet de le faire sur le net.

- 5) Créez une "**div**" avec une height de 50px et une width de 50px.
- 6) Trouvez sur internet comment ajouter une ombre à une div. (Si possible une jolie, regardez bien comment faire et utilisez des termes anglais pour trouver les règles).
- 7) Tentez de modifier l'ombre dans la console dev.
- 8) Ajoutez un fond coloré sous forme de gradient

2. L'inclusion de script javascript (c'est rigolo à dire)

A la fin du body juste avant la balise de fermeture, vous allez insérer deux scripts:

- Le premier, **jquery**, est énormément utilisé et permet de simplifier l'écriture du javascript, on ne s'attardera pas dessus aujourd'hui. Il doit simplement être chargé car il sert au deuxième script.
- Le deuxième c'est **Tilt.js**, un script assez amusant à utiliser, vous allez voir.

```
<script src="https://code.jquery.com/jquery-3.3.1.min.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/tilt.js/1.2.1/tilt.jquery.js">
</script>
```

Remarquez que ces deux scripts sont chargés directement depuis internet (il n'y a pas besoin de les avoir en local), on appelle cela un CDN ("Content Delivery Network").

- Maintenant ajoutez à votre div coloré le mot clé "*data-tilt*" qui est une composante de tilt.js:

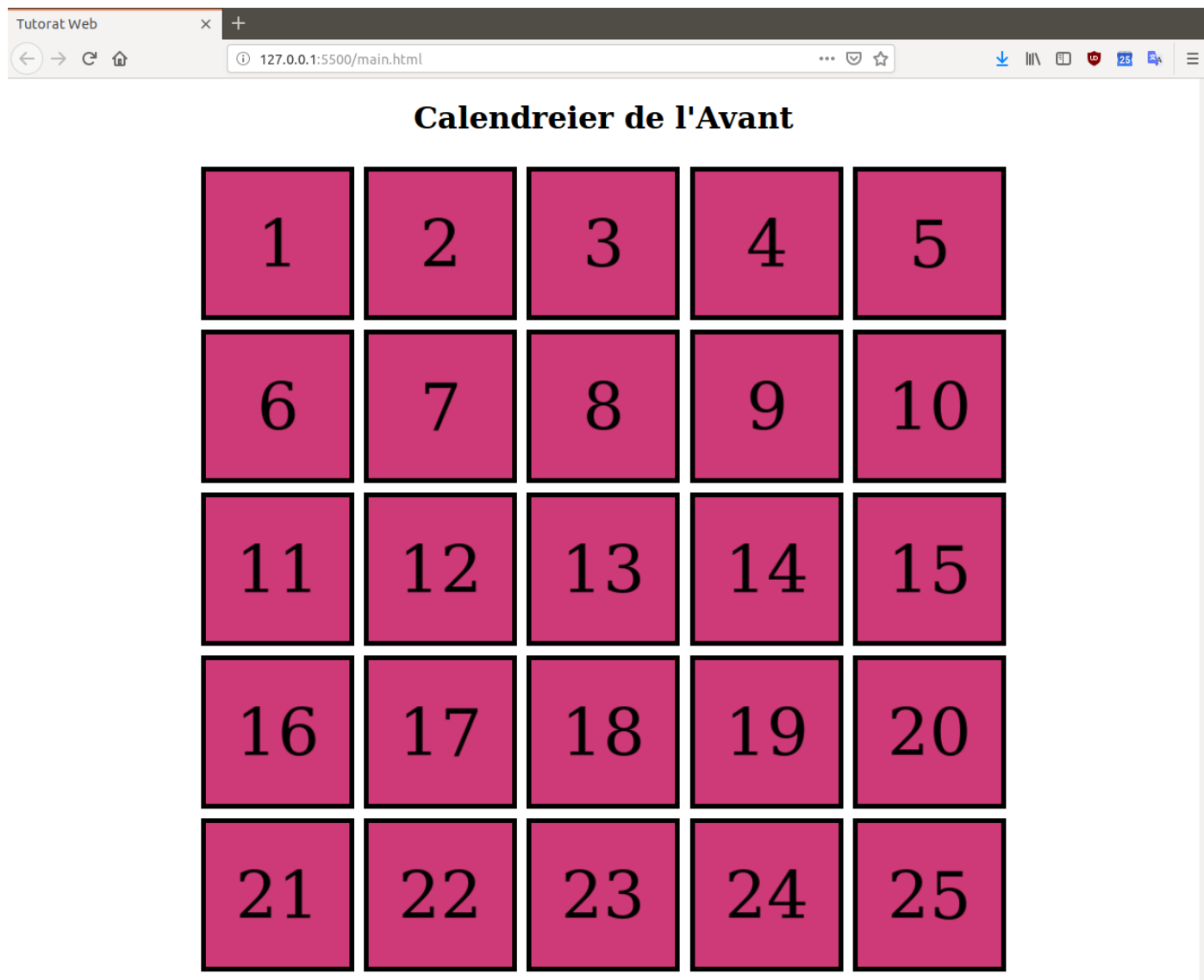
```
<div data-tilt></div>
```

- Observez ce qui se passe :)

Exercice 1 : Calendrier de l'avant

Cet exercice est tombé à la nuit de l'info 2017. Ici, vous allez créer un calendrier de l'avant. Vous démarrez un **nouveau projet avec github**.

Voici le rendu final le plus simple attendu :



Capture d'écran d'un exemple d'implémentation (choix des couleurs, police, marges, bordures... selon vos goûts)

Quelques consignes pour vous lancer :

- commencer par travailler avec deux fichiers : main.html et ma_feuille_de_style.css
- utiliser le layout `grid` CSS
- ne vous laissez pas influencer par la capture d'écran ou par les autres, HTML et CSS offre une liberté quasiment infini donc lâchez vous sur le style.
- utiliser la bibliothèque `tilt.js` pour donner du relief à vos cases

Bonus

Vous pouvez vous entraîner avec git [ici](#).