odepile : https://www.codepile.net/pile/dwm12ajax

ATTENTION SYNTAXE:
forEach

Tableau associatif = objet donc foreach ne fonctionne pas

API +- BDD

Requêtes HTTP : https://fr.wikipedia.org/wiki/Liste_des_codes_HTTP

API Star Wars = https://swapi.co/

Pour faire des requêtes sur des api on utilise postman

répertoire API : https://github.com/public-apis/public-apis

requête synchrone : on lance la requête et elle s'exécute
2 requêtes synchrones s'exécutent l'une à la suite de l'autre

requête asynchrone : on lance une requête et elle s'exécute
2 requêtes asynchrones s'exécutent en même temps peut importe la fin de l'exécution

ex : requête 1 se termine en 1 sec
requête 2 se termine en 0.5sec
lancement des 2 requêtes en même temps, résultat 2 PUIS résultat 1 car requête 2 plus rapide à exécuter

!= FIFO

RAPPEL SYNTAXE FONCTION :
function(blabla){
instruction;
};
⇒ Ancienne syntaxe, bof bof

(blabla)=>{
instruction;
};
⇒ Nouvelle version, bien !

Attention à la position des ; ⇒ on ne doit le trouver qu'à la fin de l'enchaînement des instructions

```
fetch('https://pokeapi.co/api/v2/pokemon/499')
.then((response)=>{
    response.json()
    .then((data)=>{
        console.log(data);
    })
})
```

console.log(data.name) ⇒ afficher le nom du pokemon qui est "directement" accessible

```
VM162:1
{abilities: Array(2), base_experience: 146, forms: Array(1), game_indices: Array(4)
, height: 10, …}
  abilities: (2) [{…}, {…}]
  base_experience: 146
  forms: [{…}]
  game_indices: (4) [{…}, {…}, {…}, {…}]
  height: 10
  held_items: []
  id: 499
  is_default: true
  location_area_encounters: "https://pokeapi.co/api/v2/pokemon/499/encounters"
  moves: (68) [{…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}, {…}…
  name: "pignite"
  order: 590
  species: {name: "pignite", url: "https://pokeapi.co/api/v2/pokemon-species/499/"}
  sprites: {back_default: "https://raw.githubusercontent.com/PokeAPI/sprites/master…
  stats: (6) [{…}, {…}, {…}, {…}, {…}, {…}]
  types: (2) [{…}, {…}]
  weight: 555
  __proto__: Object
undefined
> console.log(data.name)
pignite                                                    VM210:1
undefined
>
```

```
fetch('https://pokeapi.co/api/v2/pokemon/499')
.then((response)=>{
    response.json()
    .then((data) => {
        console.log(`Le nom du pokemon est : ${data.name}`);
    })
})
```

```
Le nom du pokemon est : pignite                            script.js:74
```

Pour afficher un élément qui se trouve dans un array dans les datas :

```
> console.log(data.moves[67].move.name)
  power-up-punch                                           VM340:1
```

[67] est entre crochets car l'array moves est indexés
Tous les autres éléments d'identification du truc qu'on veut afficher sont indiqués après un point

textContent = affiche le contenu d'une balise
InnerHTML = Remplace par autre chose

CREER DES ELEMENTS HTML AVEC JS

```
document.querySelector("button").addEventListener("click", ()=>{
    let h1 = document.createElement("h1");
    h1.innerHTML = "Ce titre a été généré par un eventListener";
    document.querySelector("main").appendChild(h1);
    console.log("it worked !");
});
```

```
document.querySelector("button").addEventListener("click", () => {

    // Création d'une variable qui contiendra la création d'un <h1> en HTML
    let h1 = document.createElement("h1");

    // On ajoute du contenu à cette balise <h1>
    h1.innerHTML = "Ce titre a été généré via un event listener...";

    // On va ajouter notre <h1> à notre balise <main>
    document.querySelector("main").appendChild(h1);
})
```

```
document.querySelector("button").addEventListener("click", ()=>{

    document.querySelector("main").innerHTML= '';

    let h1 = document.createElement("h1");
    h1.innerHTML = "Ce titre a été généré par un eventListener";
    document.querySelector("main").appendChild(h1);

    let p1 = document.createElement("p");
    p1.innerHTML = "paragraphe 1";
    document.querySelector("main").appendChild(p1);

    let h2 = document.createElement("h2");
    h2.innerHTML = "Titre 2";
    document.querySelector("main").appendChild(h2);

    let p2 = document.createElement("p");
    p2.innerHTML = "paragraphe 2";
    document.querySelector("main").appendChild(p2);

    let ul = document.createElement("ul");
    document.querySelector("main").appendChild(ul);

    let li1 = document.createElement("li");
    let li2 = document.createElement("li");
    let li3 = document.createElement("li");
    li1.innerHTML = "li 1";
    li2.innerHTML = "li 2";
```

```
    li3.innerHTML = "li 3";
    document.querySelector("ul").appendChild(li1);
    document.querySelector("ul").appendChild(li2);
    document.querySelector("ul").appendChild(li3);

    console.log("it worked !");
});
```

⇑Pour créer via JS des balises

```
document.querySelector("main").innerHTML= '';
```

⇑Pour éviter que les balises précédement crées se recréent à chaque clic
ATTENTION, ne pas mettre le bouton dans le main !!

//EXERCICE//

Vous allez devoir insérer dans votre HTML certaines données du Pokémon recherché.

Ces données, seront à stocker dans un <table> HTML.
Elles devront comporter les th suivants :
- name
- type1
- type2 (si existant)
- move1
- move2
- move3
- move4
- une petite image (sprite) du Pokémon en question

Également, vous tenterez d'ajouter un MINIMUM de CSS à votre <table>, pour que quand même, il soit assez joli et pas simplement un <table> tout moche en noir et blanc.

//JQUERY//

JQuery cdn sur google

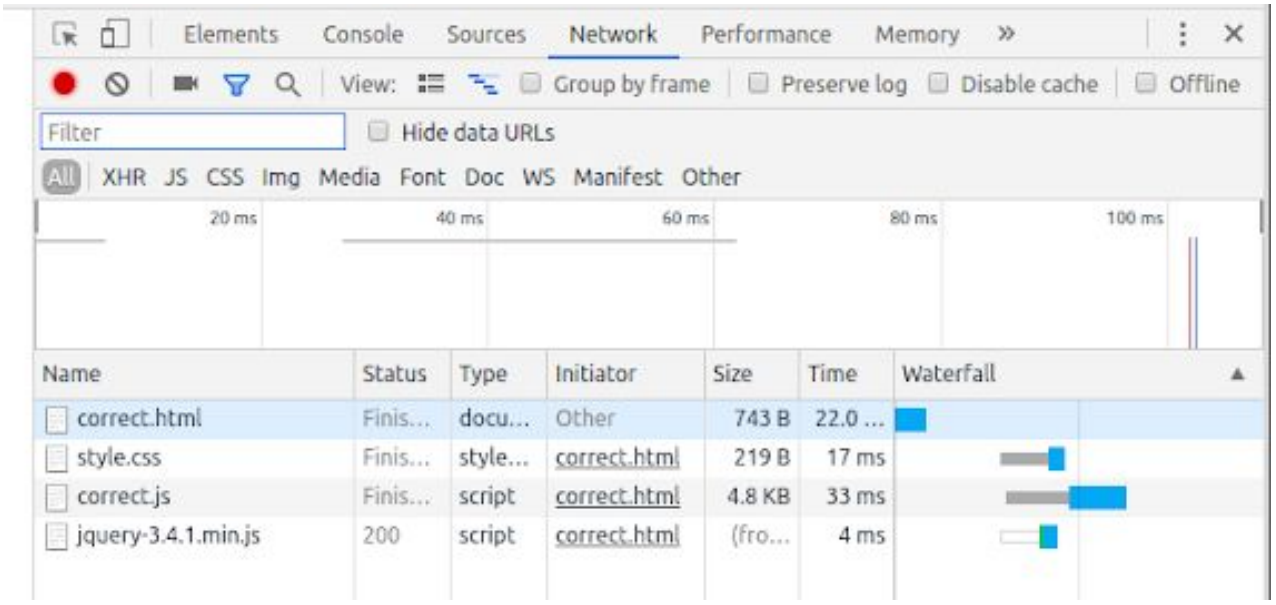
Cliquer sur JQuery Core 3x ; minified et coller le lien dans le code au-dessus du lien vers les script.js

```
    <script
    src="https://code.jquery.com/jquery-3.4.1.min.js"
    integrity="sha256-CSXorXvZcTkaix6Yvo6HppcZGetbYMGWSFlBw8HfCJo="
    crossorigin="anonymous"></script>
    <script src="./correct.js"></script>
</body>
</html>
```

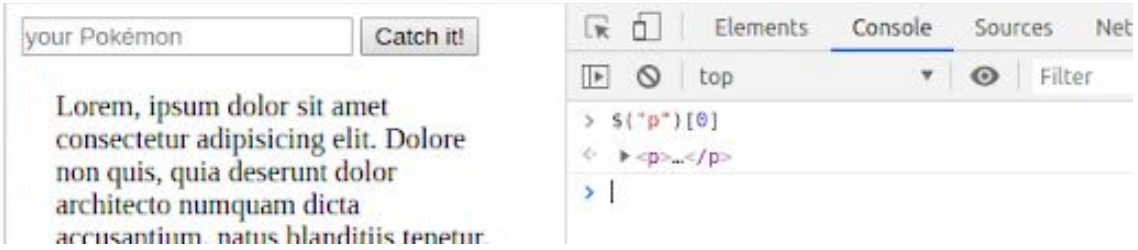
Dans le script :
https://learn.jquery.com/using-jquery-core/document-ready/

```
$(document).ready(()=>{
    //Ici on tape tout le code en jQuery ou JS
})
```

Dans l'inspecteur ; network
vérifier que JQuery apparaisse dans les "trucs" chargés



pour retrouver des éléments dans la consoles (ex le 1er <p> du doc) :



toujours le $ puis ("element")[index]

//MODIFIER ELEMENTS DU SITE DEPUIS JQUERY//

$("p")[0].replaceWith("KUUUUUULBUTOKÉÉÉÉÉ")

On peut utiliser toutes les instructions JS en JQuery:

```
$(document).ready(()=>{
```

```
    //Ici on tape tout le code en jQuery ou JS
    $("p")[0].innerHTML = "MIAOUSS ! OUI LA GUERRE !"
})
```

création enventListener :

```
$(document).ready(()=>{
    //Ici on tape tout le code en jQuery ou JS
    $("p")[0].innerHTML = "MIAOUSS ! OUI LA GUERRE !"

    $("button").on("click", ()=>{
        console.log("Bouton cliqué");
    })
})
```

Attention, les [] semblent ne pas fonctionner avec l'application de fonctions

//TP//
Vous devez construire un petit site contenant à minima :
-Une navbar
-Un footer
-4 images
-Du texte
Celui-ci doit être responsive jusqu'à 350px de large

PDO

https://www.codepile.net/pile/dwm12pdo

faire une MV avec un LAMP
(si besoin faire un sudo apt update avant de faire les sudo apt install de apache2, mysql, php7.3)

se connecter à mysql depuis la VM dans terminal :
mysql -uroot -p
ensuite entrer password

Créer une DB :
CREATE database "nomDB";
NE PAS OUBLIER LE ;

Supprimer une DB :
drop databse "nomDB";

Voir les DB:
show databases;
⇒
mysql> show databases;
+--------------------+
| Database |

```
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| students           |
| sys                |
+--------------------+
5 rows in set (0.01 sec)
```

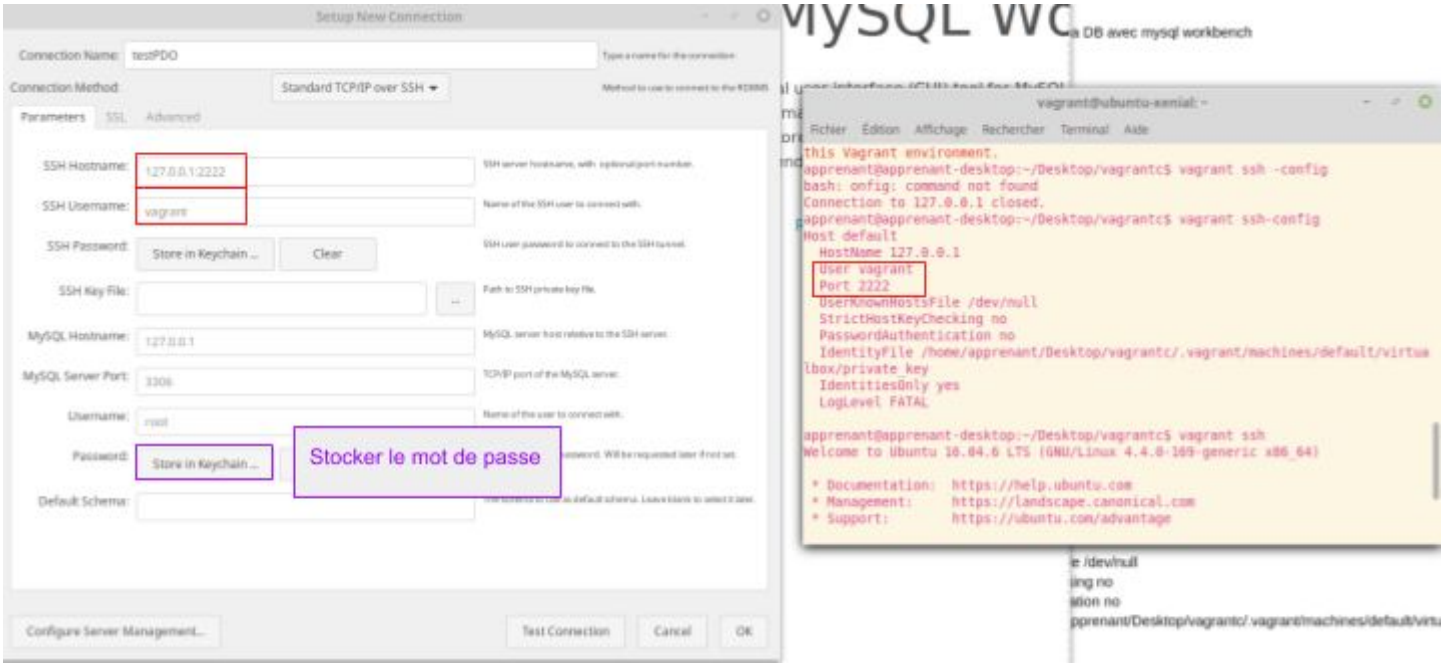
SGBD ⇒ Système de Gestion de Base de Données

On fera la manip de la DB avec mysql workbench

Sortir de mysql :
mysql> exit

MySQL Connections

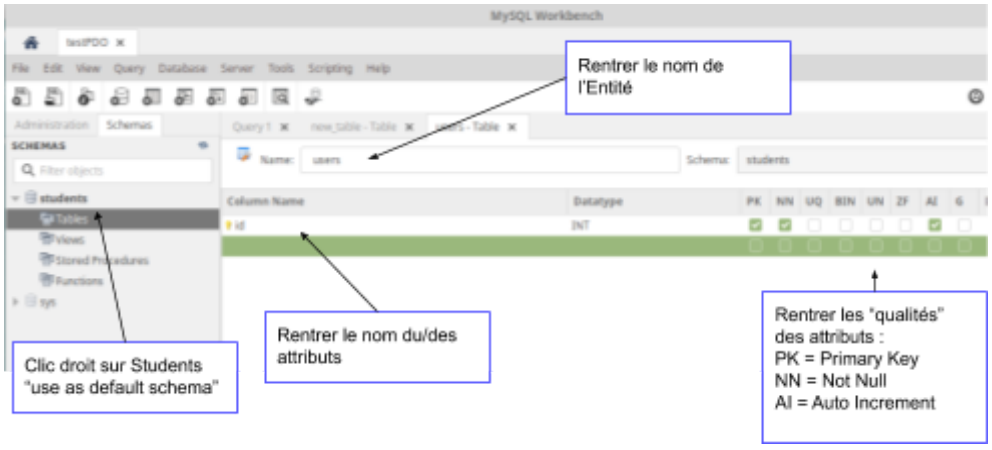
Obtenir les infos de connection :
sortir de la vagrant (exit) et taper : vagrant ssh-config
Host default
  HostName 127.0.0.1
  User vagrant
  Port 2222
  UserKnownHostsFile /dev/null
  StrictHostKeyChecking no
  PasswordAuthentication no
  IdentityFile /home/apprenant/Desktop/vagrantc/.vagrant/machines/default/virtualbox/private_key
  IdentitiesOnly yes
  LogLevel FATAL



clic sur SSH Keyfile [...]
Aller dans le dossier de la vagrant
Clic droit afficher les fichiers cachés ou CTRL+H pour afficher .vagrant
clic jusqu'à pouvoir sélectionner "private_key"

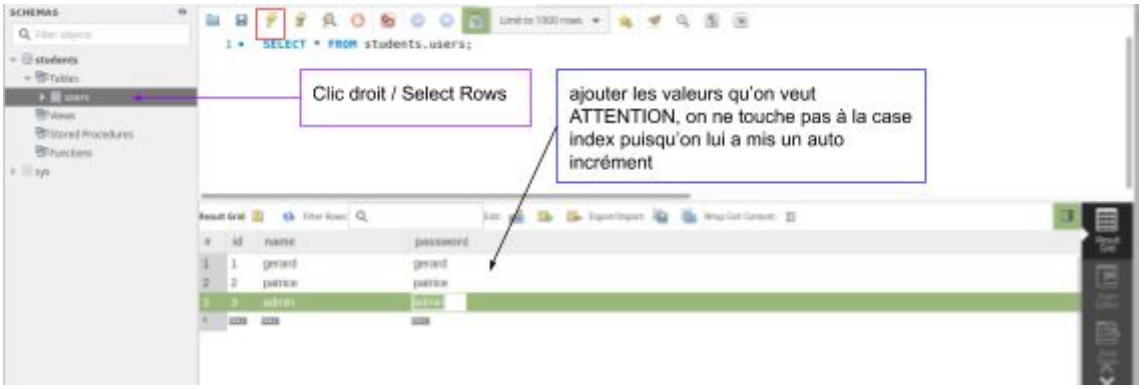
Si une erreur de fingerprint apparaît **DANS UN NOUVEAU TERMINAL**
sudo nano ~/.ssh/known_hosts
Puis supprimez toutes les lignes de ce fichier, sauvegardez et retentez la connexion

On est arrivé à connecter Workbench à la vagrant !!



Quand on a tout entré, on clique sur Apply ⇒ Affiche le script
Puis re Apply pour appliquer

Pour ajouter des données à la table users, clic droit sur users, select rows, dans la partie du "milieu" de la page, rentrer les données voulues puis coup de foudre :



(EN POO, () ⇒ création objets)

```php
$dbuser= 'root';
$dbpassword='0000';

//Créer une connexion à notre DB
$db= new PDO ('mysql:host=localhost;dbname=students', $dbuser, $dbpassword);
```

Si ce message dans navigateur :
Fatal error: Uncaught PDOException: could not find driver in /var/www/html/index.php:7 Stack trace: #0 /var/www/html/index.php(7): PDO->__construct('mysql:host=loca...', 'root', '0000') #1 {main} thrown in **/var/www/html/index.php** on line **7**

faire dans terminal dans la vagrant :
sudo apt install php7.3-mysql
puis
sudo service apache2 restart

Limiter la casse des erreurs de connection : https://www.php.net/manual/fr/pdo.connections.php

```php
try {
    $dbuser= 'root';
    $dbpassword='0000';

    //Créer une connexion à notre DB
    $db= new PDO ('mysql:host=localhost;dbname=students', $dbuser, $dbpassword);

} catch (PDOException $e) {
    print "Erreur !: " . $e->getMessage() . "<br/>";
    die();
}
```

Créer un repo “db” dans le repo qui contient notre code et faire un include. Ne pas laisser ⇑ dans l’index. JAMAIS.

Pour pouvoir utiliser la db pour afficher/ créer/ supprimer des users, on colle le “try” d’au-dessus dans une fonction :

```php
function showUsers (){
    try {
        $dbuser= 'root';
        $dbpassword='0000';
        //Créer une connexion à notre DB
        $db= new PDO ('mysql:host=localhost;dbname=students', $dbuser, $dbpassword);
    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage() . "<br/>";
        die();
    }
}
```

On appellera la fonction quand on en aura besoin, en attendant, la fonction ne se lance pas
BIEN PENSER À INSTALLER PHP-XDEBUG PUIS SUDO SERVICE APACHE2 RESTART

index.php :

```php
<?php
include 'db/database.php';
//showUsers(); pour vérifier l'affichage du coucou
showUsers();
```

database.php :

```php
<?php
function showUsers (){
    try {
        $dbuser= 'root';
        $dbpassword='0000';
        //Créer une connexion à notre DB
        $db= new PDO ('mysql:host=localhost;dbname=students', $dbuser, $dbpassword);
        //echo 'coucou'; pour tester le fonctionnement de la fonction

        //Afficher tous les users:
        $sql = "SELECT * FROM users";

        //var_dump($sql);die; voir que $sql n'est pour l'instant qu'une string
        //Faire en sorte que PDO prépare notre requête SQL
        $allUsers = $db->prepare($sql);
        //Faire éxecuter la requête au PDO
        $allUsers->execute();
        //Récupérer le contenu de notre requête
        $allUsers=$allUsers->fetchAll();
        var_dump($allUsers);die; //"voir" le tableau

    } catch (PDOException $e) {
        print "Erreur !: " . $e->getMessage() . "<br/>";
        die();
```

```
    }
}
```

ATTENTION !! ERREUR !! NE PAS déclarer la fonction dans le fichier database, faire un fichier qui porte le nom de la fonction puis directement try :

index.php:

```php
<?php
include 'db/showUsers.php';
```

showUsers.php

```php
<?php

try {
    $dbuser= 'root';
    $dbpassword='0000';
    //Créer une connexion à notre DB
    $db= new PDO ('mysql:host=localhost;dbname=students', $dbuser, $dbpassword);
    //echo 'coucou'; pour tester le fonctionnement de la fonction

    //Afficher tous les users:
    $sql = "SELECT * FROM users";

    //var_dump($sql);die; voir que $sql n'est pour l'instant qu'une string
    //Faire en sorte que PDO prépare notre requête SQL
    $allUsers = $db->prepare($sql);
    //Faire éxecuter la requête au PDO
    $allUsers->execute();
    //Récupérer le contenu de notre requête
    $allUsers=$allUsers->fetchAll();
    //var_dump($allUsers);die; //"voir" le tableau
    // foreach($allUsers as $users) {
    //     //var_dump($users); //voir chaque user de manière indépendante
    // }

} catch (PDOException $e) {
    print "Erreur !: " . $e->getMessage() . "<br/>";
    die();
}
```

Pour afficher les users “dans” l’index.php

```php
foreach($allUsers as $key => $users) {
    echo "$users[$key]<br />";
}
```

18.11.2019 ⇒ suite cours pdo

Les affichages de contenu de mysql ne se font pas sur l'index donc créer par exemple un dossier admin dans lequel on aura un visu sur les users
https://www.php.net/manual/fr/pdo.prepared-statements.php