Rapport Personnel

Jacquier Romain

|  |  |
| --- | --- |
| Une image contenant texte  Description générée automatiquement | Module du 15.05.2023 au 13.06.2023 |

Date de création du 15.05.2023

Version 1 du 07.06.2023

Tables des matières

1 Introduction 6

1.1 Objectifs du module 6

2 Exercice 0 6

3 Exercice 1 Base HTML5 7

3.1 Objectifs 7

3.2 Plugins 8

3.3 Questions 8

4 Exercice 2 Login 9

4.1 Objectifs 9

4.2 Questions 9

5 Exercice 3 Login avec PHP 11

5.1 Objectifs 11

5.2 Réalisation 12

5.3 Question 12

5.4 Code 14

6 Exercice 4 JSFiddle 15

6.1 Objectif 15

6.2 Réalisation 15

6.3 Questions 16

7 Exercice 5 Variable en Javascript 16

7.1 Objectif 16

7.2 Réalisation 17

7.3 Questions 17

7.4 Code 18

8 Exercice 6 La structure de contrôle switch 20

8.1 Objectif 20

8.2 Réalisation 20

8.3 Questions 20

9 Exercice 7 Les boucles 21

9.1 Objectif 21

9.2 Réalisation 22

9.3 Questions 23

10 Exercice 8 Tableau d’objets JSON 24

10.1 Objectif 24

10.2 Réalisation 24

10.3 Questions 24

10.4 Exercice 9 Démo sur les objets en JavaScript 25

10.5 Objectif 25

10.6 Réalisation 25

10.7 Questions 26

10.8 Exercice 10 Programmation orientée « objets » 26

10.9 Objectif 26

10.10 Réalisation 27

10.11 Question 28

11 Exercice 11 : Programmation orientée "classe" en Javascript 28

11.1 Objectif 28

11.2 Réalisation 29

11.3 Questions 30

12 Exercice 12 : Fonctions et IIFE en JavaScript 31

12.1 Objectif 31

12.2 Réaliser 31

12.3 Question 32

13 Exercice 13 : Les cookies 33

13.1 Objectif 33

13.2 Réalisation 34

13.3 Questions 35

14 Exercice 14 : Bases de jquery 36

14.1 Objectif 36

14.2 Réalisation 36

14.3 Questions 36

15 Exercice 15 : Première utilisation de JQuery 40

15.1 Objectif 40

15.2 Réalisation 40

15.3 Questions 41

16 Exercice 16 : Utilisation de jquery 42

16.1 Objectif 42

16.2 Réalisation 42

16.3 Questions 43

17 Exercice 17 WebServices 44

17.1 SOAP/REST 44

17.2 Get 44

17.3 Post 45

17.4 Put 45

17.5 Delete 45

18 Postman 46

19 Exercice 18 WebServices - Application HTML appelant WS PHP 46

19.1 Objectif 46

19.2 Réalisation 46

19.3 Questions 46

20 Exercice 20 Application de login sur un serveur PHP 47

20.1 Objectif 47

20.2 Réalisation 47

20.3 Questions 48

21 Projet « Villes » 49

22 Analyse 49

22.1 Description 49

22.2 API 49

22.3 Use Case 49

22.4 Maquettes 51

22.5 Test web Service 52

23 Conception 53

23.1 Diagramme nav 53

24 Code 54

25 Conclusion 55

# Introduction

C’est un module ou l’on va réaliser de la programmation Web. Il est dans la continuité du module 101 que l’on a réalisé en première année. Et durant lequel on avait fait un site web avec de l’html et du css. On va donc continuer à travailler avec cela mais en travaillant également avec Java Script.

## Objectifs du module

Voici les objectifs de ce module.

1. Développer le caractère fonctionnel des pages Web interactives conformément aux données du problème.
2. Développer une maquette pour la saisie et la présentation des données compte tenu des aspects ergonomiques.
3. Choisir les éléments de formulaire appropriés pour la réalisation des données du problème, et garantir la validation des données entrées.
4. Programmer l’application de manière modulaire et conformément aux directives de codification.
5. Définir et en mettre œuvre des cas de tests appropriés pour des pages Web interactives et documenter dans le procès-verbal de tests

# Exercice 0

Connexion fileZilla

Il faut aller dans fichier sous gestionnaire de sites et ensuite mettre les informations suivantes pour pouvoir se connecter.

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

C’est dans public html que l’on a tous les fichiers de notre site web.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

# Exercice 1 Base HTML5

On a réalisé cet exercice durant la 1ère semaine de ce module.

Voici l’interface que l’on avait à disposition pour ce module.

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

## Objectifs

Voici les objectifs de cet exercice.

* Lire et comprendre du code HTML/CSS/Javascript.
* Adapter les sources d'un programme selon des directives.
* Ajouter des écouteurs Javascript pour déclencher des opérations.
* Exécuter un code Javascript lorsqu'une page HTML est chargée.
* Donner une définition du DOM.

## Plugins

Dans l’application visio code, voici ou l’on va avoir la possibilité d’installer les plugins.

Une image contenant texte, capture d’écran, logiciel, Icône d’ordinateur

Description générée automatiquement

On en a installer 7 qui vont nous être utile.

Live Server, Bracket Pair Colorizer, JavaScript (ES6) code snippets, Prettier Code formatter, Material icon Theme, Jquery code snippets, CSS Peek et French Language

## Questions

Voici les questions auxquelles on a dû répondre pour cet exercice :

* Comment ajouter un écouteur "click" via Javascript.

Voici le code qui permet de faire cela.

function initCtrl() {

  // Ecouteur du bouton "Testez-moi..."

  document.getElementById("testez").addEventListener("click", testez);

}

* Comment ajouter directement un écouteur "click" sur un bouton.

Voici le code qui permet de faire cela, on va utiliser le onclick avec entre guillmets le nom de la fonction javascript qu’il va utiliser.

<button id="testez" onclick="testez()" style="color:red;">Teste-moi, James</button>

Et voici la fonction qui est appelée.

function testez() {

   document.getElementById("info").innerHTML =

      "C'est <b>James Bond</b> qui a pressé le bouton !";

}

* Expliquez comment exécuter du code Javascript lorsque la page HTML a fini d'être chargée.

Pour faire cela, on va utiliser le code ci-dessous avec le initCtrl.

<body onload="initCtrl()">

Il sera au début du body et permettra donc d’initialiser certaines choses dans le JavaScript que lorsque la page sera chargée.

* Expliquez ce que signifie DOM.

DOM est l’abréviation de Document Object Model. Et grâce à cela, JavaScript va avoir la capacité de créer du HTML dynamique.

* Où le script Javascript peuvent être chargés et pourquoi.

Cela va être chargé dans le body, car si il faut que tout les éléments aient déjà été chargé avant le onload. Cependant, dans le head, on va quand même devoir mettre une référence au fichier JS.

<script type="text/javascript" src="js/indexCtrl.js" async></script>

# Exercice 2 Login

On a réalisé cet exercice durant la semaine 1 de ce module.

## Objectifs

Voici les objectifs de cet exercice.

* Créer un formulaire HTML avec des champs de saisie et des boutons.
* Exécuter une opération lorsqu'un bouton HTML est pressé.
* Récupérer les informations saisies dans un formulaire HTML.
* Afficher des informations sur la console d'un navigateur.
* Déclarer des variables.
* Exécuter du code Javascript en fonction de la valeur d'une variable.
* Afficher des messages d'information dans le navigateur.

## Questions

À quoi sert l’attribut « placeholder » ?

Cela permet de spécifier un bref indice de la valeur qui est attendue dans le champ d’entrée. Cela va être afficher dans le champ de saisie avant que l’utilisateur n’entre une valeur. Dans l’exemple ci-dessous, il s’agit du texte « Un nom SVP »

Une image contenant texte, logiciel, Police, capture d’écran

Description générée automatiquement

<input type="text" size="30" id="username"

              placeholder="un nom svp" autofocus />

À quoi sert l’attribut « autofocus » ?

Il s’agit d’un attribut booléen, losqu’il sera présent, il va permettre de spécifier qu’un élément input doit automatiquement obtenir le focus lors du chargement de la page

Y a-t-il déjà du JavaScript dans ce code ?

Non il n’y en a pas.

Quelle est la différence entre le bouton de l'exercice 1 et celui-ci ?

Lors de l’exercice 1, c’est la balise button qui a été utiliser contrairement à celui-ci ou il s’agit de la balise input qui est de type button.

<input type="button" value="Valider" id="valider" onclick="validerUtilisateur();" class="valider">

Comment définit-on une « fonction » (méthode) comme « validerUtilisateur() » ?

Voici comment on définit une fonction qui est vide.

function validerUtilisateur() {

}

Comment définit-on une « variable » (par exemple pour « username » et « password ») ?

Voici comment on définit une variable. Dans ce cas, on définit deux variables, nomutilisateur et mdp.

let nomUtilisateur = document.getElementById("username").value;

  let mdp = document.getElementById("password").value;

Les variables sont-elles « typées » (String par exemple) ?

Non elles ne le sont pas, pour déclarer un string ou un int, on va devoir mettre let.

Où se trouve la console de débogage de Chrome ?

C’est sous Console, et lorsque l’on aura une erreur dans le code, cela va nous afficher à quelle ligne elle se trouve.

Une image contenant texte, capture d’écran

Description générée automatiquement

Expliquer les fonctions alert(), confirm() et prompt().

La fonction alert va permettre d’afficher une popup à l’utilisateur avec un message que l’on aura prédéfinis.

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

La fonction confirm va afficher une boxe de dialogue avec un message et l’utilisateur aura le choix entre le bouton OK et le bouton Cancel.

La fonction prompte va ouvrire une box de dialogue qui va demander à l’utilisateur d’entrer une valeur, par exemple son nom.

Quelle est la différence entre bouton "button" et un bouton "input" ?

Dans l’input on ne va pouvoir mettre que du texte contrairement au bouton ou l’on pourra mettre des images ou encore d’autres balises HTML.

Une explication pour récupérer un élément HTML depuis Javascript grâce à son id.

On va utiliser le document.getElementById pour pouvoir récupérer l’élément grâce à son id.

document.getElementById("username").value

Comment récupér le texte contenu dans un champ de formulaire ?

Pour cela, on va utiliser le .value après avoir été recherché l’élément à partir de son id.

document.getElementById("username").value

Comment écrire sur la console de votre navigateur ? Quelle est l'utilité ?

Cela va nous permettre d’avoir des outputs et il faudra que dans notre code on utilise console.log

Où peut-on charger les script Javascript et pourquoi ?

Dans le head on va juste faire la référence au fichier JS et ensuite on va charger les scripts dès que l’on va utiliser le onload ou le onclick.

<input type="button" value="Valider" id="valider" onclick="validerUtilisateur();" class="valider">

Comment convertir un texte en minuscule ?

Voici comment on peut faire dans le CSS.

div.b {

  text-transform: lowercase;

}

# Exercice 3 Login avec PHP

On a réalisé cet exercice durant la première semaine de ce module.

## Objectifs

Voici les objectifs de cet exercice.

* Créer un formulaire HTML avec des champs de saisie et des boutons de type submit.
* Comprendre le traitement des informations dans un fichier PHP, côté serveur.
* Enumérer les différents types de boutons qui sont programmables dans un formulaire HTML.
* Traiter les données d'un formulaire HTML par un scripte PHP.

## Réalisation

On peut voir que l’on utilise la méthode post. (200 veut dire que cela fonctionne, 404 est l’erreur File Not Found.

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

Et ensuite lorsque l’on va sous Payload, on peut voir les valeurs que l’on a entré pour le mot-de passe.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

La différence entre le post et le get est que dans la barre de recherche on ne verra pas les données que l’on a entrée lorsque l’on fait un post, contrairement au get ou l’on verra tout. Le get permet d’aller rechercher des informations tandis que l’on va utiliser le post pour donner des informations.

## Question

Comment centrer le bouton de validation et le styliser ?

On va devoir utiliser deux méthodes différentes comme on peut le voir sur l’image ci-dessous. La première va permettre de centrer le bouton et la deuxième va permettre de le styliser.

.user-form .button {

  text-align: center;

}

.user-form input[type=submit] {

  background-color: rgb(146, 144, 144);

  border: none;

  color: white;

  font-size: 16px;

  margin-top: 5px;

}

Comment envoyer le formulaire à un script PHP qui fera la validation ?

Pour faire cela, on va utiliser la ligne de code ci-dessous. C’est le submit qui permet de faire cela, le action permet de spécifier qu’elle serveur va être interrager.

<input type="submit" value="Valider">

Une explication sur les différents types de boutons programmables dans un formulaire HTML.

Voici une liste qui regroupe ces explications.

* **type="button" :** ce type crée un simple bouton cliquable sans fonctionnalité particulière.
* **type="submit" :** ce type crée un bouton qui envoie un formulaire vers un serveur lorsqu'il est cliqué. Le visiteur sera conduit à la page indiqué dans l'attribut ***action***.
* **type="reset" :** ce type crée un bouton qui réinitialise les valeurs d'un formulaire à leurs valeurs par défaut.
* **type="image" :** ce type crée un bouton qui est représenté par une image spécifiée dans l'attribut src.
* **type="checkbox" :** ce type crée un bouton qui peut être coché ou décoché.
* **type="radio" :** ce type crée un bouton qui peut être sélectionné dans un groupe de boutons liés.
* **type="file" :** ce type crée un bouton qui permet à l'utilisateur de sélectionner un fichier sur son ordinateur.

Par quoi commence et doit se terminer un code PHP ?

Par la balise suivante.

<?PHP

?>

Comment récupère-t-on une information dans le flux des informations GET ou POST ?

Dans le code html, on va utiliser name = « nom de l’info que l’on recherche »

<input name = "username" type="text" size="30" id="username"

              placeholder="un nom svp" autofocus />

ou encore

<input name = "password" type="password" size="30" id="password"

              placeholder="un mot de passe svp" />

Comment effectue-t-on une concaténation de chaines de caractères en PHP ?

Pour pouvoir concaténer une chaine de caractère en php, on utilise le point « . ».

Avec quelle fonction intrinsèque (dans PHP) convertit-on une chaine en minuscules ?

Il s’agit de strtolower.

$username = strtolower($\_POST['username']);

Quelle est la commande qui permet de renvoyer quelque chose vers le client ?

Pour le php il s’agit du echo. Et grâce à cette commande, cela va nous afficher les données que l’on a entré pour se connecter.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

Quel(s) autre(s) langage(s) trouve-t-on encore dans ce code PHP ?

On retrouve du JavaScripte qui permet de faire les popup.

## Code

Voici le code de index.html que l’on a fait pour le post.

 <form class="user-form" action="https://jacquierr.emf-informatique.ch/Module\_307/Exercices/Exercice\_3/login.php" method="POST">

        <fieldset>

          <legend>Identification:</legend>

          <div class="field" >

            <label for="username">Nom d'utilisateur:</label>

            <input name = "username" type="text" size="30" id="username"

              placeholder="un nom svp" autofocus />

          </div>

          <div class="field" >

            <label for="username">Mot de passe:</label>

            <input name = "password" type="password" size="30" id="password"

              placeholder="un mot de passe svp" />

          </div>

          <div class="button">

            <input type="submit" value="Valider">

          </div>

        </fieldset>

      </form>

Et voici le code de login.php.

<?PHP

// test si on a reçu une donnée de formulaire nommée "username"

if (isset($\_POST['username'])) {

  // récupération des données transmises dans des variables locales

    $username = strtolower($\_POST['username']);

    $password = $\_POST['password'];

  // affichage des infos reçues

  echo "username: ".$username."</br>";

  echo "password: ".$password."</br>";

  // test username et mot de passe

  if (($username == "admin") && ($password == "emf123")) {echo "<script>alert('Validation OK');</script>";} else {

    echo "<script>alert('Utilisateur ou mot de passe incorrect !!!');</script>";

  }

}

?>

# Exercice 4 JSFiddle

On a réalisé cet exercice durant la première semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Tester du code HTML, CSS et Javascript avec JSFiddle.

## Réalisation

On s’est créer un compte sur le site suivant.

Liens : <http://jsfiddle.net/>

Une image contenant texte, capture d’écran, Police, logiciel

Description générée automatiquement

On peut voir que le code s’affiche dans la console.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

Code permettant de réaliser un carré qui, lorsque l’on clique dessus, affiche une popup.

Une image contenant texte, capture d’écran, logiciel, Logiciel multimédia

Description générée automatiquement

## Questions

Voici les questions auxquels on a dû répondre durant cet exercice.

Qu'est-ce que JSFiddle ? A quoi sert-il ?

JSFiddle est un outil en ligne pratique pour le développement et le partage de code HTML, CSS et JavaScript, facilitant le prototypage rapide, le débogage et la collaboration entre les développeurs web.

Quel est l'avantage d'utiliser cet outil ?

L'avantage de cette solution est qu'il est possible de tester du code HTML, CSS ou Javascript rapidement sans devoir mettre en place des fichiers et ouvrir son IDE.

# Exercice 5 Variable en Javascript

On a réalisé cet exercice durant la première semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Déclarer des variables avec les mots clés var, let et const.
* Connaître la différence entre les différentes déclarations d'une variable.
* Utiliser et traiter des variables de type date.
* Afficher le contenu des variables sur la console.
* Utiliser des fonctions littérales pour imprimer des variables sur la console.
* Connaître les types primitifs de Javascript.
* Identifier le type d'une variable Javascript en utilisant le mot clé typeof.

## Réalisation

Voici deux manières différentes qui permettent de réaliser le même calcule. La différence est que la deuxième fois on va utiliser un litéral avec `…${…}…`

console.log(a+"+"+b+"="+(a+b));

console.log(a+"+"+b+`=${a+b}`);

## Questions

Comment déclarez-vous une variable en Javascript ?

Il y a différente manière pour le faire, cependant on utilisera le let nomvariable = ?

Comment déclarez-vous une constante et quelle convention est utilisée ?

La convention pour les constantes est de les mettre en majuscules.

Quelle est la différence entre les mots clés ***var*** et ***let*** ?

Il faut utiliser le let car le var est beaucoup plus ancien et si on utilise le var on pourrait déclarer 2X la même variable ce que l’on ne veut pas.

Quelle particularité y a-t-il à déclarer une variable sans les mots clés ***var*** et ***let*** ?

Cela va créer une variable globale non-déclarée, ces variables peuvent mener à des comportements inattendus et sont considérées comme une mauvaise pratique.

Combien y a-t-il de type primitif en Javascript et quels sont-ils ?

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Quelle instruction permet de connaître le type d'une variable ? Donnez un exemple.

Pour connaitre le type d’une variable, on utilise le typeof comme on peut le voir dans l’exemple ci-dessous.

Une image contenant texte, capture d’écran, Logiciel multimédia, logiciel

Description générée automatiquement

## Code

Voici le code que l’on nous demandait de réaliser pour cet exercice sur JSFiddle.

//Effacer le contenu de la console avec « console.clear() » ;

console.clear;

//Créer une variable nommée « a » ;

let a = ("variable a");

//Afficher le contenu de « a » avec console.log;

console.log(a);

//Stocker la valeur 15 dans cette variable ;

a = 15;

//Afficher le contenu de cette variable dans la console sous la forme « Ma variable a = ? »

console.log("Ma variable a = "+a);

//Créer une variable nommée « b » et lui assigner directement la valeur 9 ;

let b = 9;

//Afficher le contenu de cette variable dans la console sous la forme « Ma variable b = ? »

console.log("Ma variable b = "+b);

//Faire l’addition de ces 2 variables en affichant directement le résultat dans la console sous cette forme : « 15 + 9 = ? » ; (essayer d’utiliser un littéral avec `…${…}…`)

console.log(a+"+"+b+"="+(a+b));

//Compléter en faisant de même pour une soustraction, une multiplication et une division des deux variables ;

console.log(a+"+"+b+`=${a+b}`);

console.log(a+"-"+b+`=${a-b}`);

console.log(a+"\*"+b+`=${a\*b}`);

console.log(a+"/"+b+`=${a/b}`);

//Stocker « Bonjour » dans la variable a ;

a = "Bonjour";

//Stocker « les amis » dans la variable b ;

b = "les amis";

//Afficher « bonjour les amis » dans la console en concaténant les variables ;

console.log(a+b);

//Faites la même chose en utilisant un littéral avec `…${…}…`;

console.log(`${a+b}`);

//Stocker « true » dans la variable a ;

a = true;

//Stocker « false » dans la variable b ;

b = false;

//Effectuer une opération AND entre les 2 variables et afficher le résultat sous cette forme « true AND false = ? » ;

console.log(a+" AND "+b+` = ${a && b}`);

//Effectuer une opération OR entre les 2 variables et afficher le résultat sous cette forme « true OR false = ? » ;

console.log(a+" OR "+b+` = ${a || b}`);

//Stocker la date du jour dans la variable a avec new Date();

a = new Date();

b = new Date();

//Calculer une nouvelle date dans la variable b qui est 61 jours avant la date courante (utilisation getDate, setDate)

b.setDate(a.getDate()-61);

//Afficher les dates contenues dans les variables a et b en vous aidant de « toLocaleString(), toLocaleDateString() et toLocaleTimeString()» . Afficher la date et l'heure, la date uniquement et l'heure uniquement.

console.log(a.toLocaleString());

console.log(b.toLocaleString());

console.log(a.toLocaleDateString());

console.log(b.toLocaleDateString());

console.log(a.toLocaleTimeString());

console.log(b.toLocaleTimeString());

//Le mot réservé « typeof » permet de connaitre le type utilisé momentanément pour une variable. Stocker la valeur de Math.PIdans a, « bonjour » dans b, créer et assigner true dans c, créer, assigner la date courante dans d et déclarez la variable e sans rien lui affectez, puis afficher le type pour les 5 variables :

a = Math.PI;

b = "bonjour";

let c = true;

let d = new Date();

let e;

console.log(typeof a);

console.log(typeof b);

console.log(typeof c);

console.log(typeof d);

console.log(typeof e);

Et voici le résultat que l’on avait sur la console.

"variable a"

"Ma variable a = 15"

"Ma variable b = 9"

"15+9=24"

"15+9=24"

"15-9=6"

"15\*9=135"

"15/9=1.6666666666666667"

"Bonjourles amis"

"Bonjourles amis"

"true AND false = false"

"true OR false = true"

"16.05.2023 09:46:43"

"16.03.2023 09:46:43"

"16.05.2023"

"16.03.2023"

"09:46:43"

"09:46:43"

"number"

"string"

"boolean"

"object"

"undefined"

# Exercice 6 La structure de contrôle switch

On a réalisé cet exercice durant la première semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Déclarer des variables avec les mots clés var, let et const.
* Connaître la différence entre les différentes déclarations d'une variable.
* Utiliser et traiter des variables de type date.
* Afficher le contenu des variables sur la console.
* Utiliser des fonctions littérales pour imprimer des variables sur la console.

## Réalisation

Voici l’interface de l’application qui affiche quels jours de la semaine on est.

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

## Questions

Comment récupérez-vous le numéro du jour actuel ?

On utilise la méthode .getDay

dateActuelle.getDay()

Expliquez comment créer un tableau, vide ou avec des valeurs.

Création d’un tableau avec les jours de la semaine.

  const jourSemaine = [

    "Sunday",

    "Monday",

    "Tuesday",

    "Wednesday",

    "Thursday",

    "Friday",

    "Saturday",

  ];

Expliquez comment ajouter et supprimer des valeurs, au début ou à fin, d'un tableau.

On va pouvoir faire cela grâce au code ci-dessous.

noms.push('Eric'); // ajoute Eric à la fin du tableau

noms.unshift('AA'); // ajoute AA au début du tableau

noms.pop(); // efface le dernier élément du tableau

noms.shift(); // efface le premier élément du tableau

Expliquez comment récupérer une valeur, dans un tableau, à position donnée.

Pour cela, on va mettre entre crochet le numéro de la cellule à laquelle on aimerait récupéré la valeur.

Expliquez comment afficher l'ensemble des valeurs d'un tableau.

Pour faire cela, on peut faire une boucle fori avec à l’intérieur, un console.log qui permettra d’afficher cela.

for (let i=0; i<noms.length; i++) {

  console.log(i + " : " + noms[i]); // 0 : Alice …

}

# Exercice 7 Les boucles

On a réalisé cet exercice durant la première semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Connaître et utiliser toutes les sortes de boucles en JavaScript
* Gérer les évènements JavaScript

## Réalisation

Voici l’interface graphique que l’on avait à disposition.

Une image contenant texte, Police, capture d’écran

Description générée automatiquement

Et ci-dessous, on peut voir les différents résultats qui était attendu lorsque l’on appuyait sur le bouton for, while et le do while.

Une image contenant texte, capture d’écran, nombre, Police

Description générée automatiquement

## Questions

Comment utiliser la boucle for en JavaScript

Voici le code que l’on a utiliser pour la boucle for en JavaScript.

function testerFor() {

  let resultat = "for (let i = 0; i < 5; i++) {...} <br>";

  for (let i = 0; i < 5; i++) {

    resultat += "i = " + i + "<br>";

  }

  resultat +=

    "--> A utiliser si on sait que l'on veut itérer x fois (x connu avant de commencer la boucle)";

  document.getElementById("info").innerHTML = resultat;

}

Comment utiliser la boucle while en JavaScript

Voici le code que l’on a utilisé pour la boucle for en JavaScript.

function testerWhile() {

  let i = 0;

  let resultat = "while(i<5) {...} <br>";

  while (i < 5) {

    resultat += "i = " + i + "<br>";

    i++;

  }

  resultat +=

    "--> A utiliser si on ne sait pas le nombre d'itérations au démarrage de la boucle";

  document.getElementById("info").innerHTML = resultat;

}

Comment utiliser la boucle doWhile en JavaScript

Voici le code que l’on a utilisé pour la boucle do While en JavaScript.

function testerDoWhile() {

  let resultat = "do {...} while(i<5);  <br>";

  let i = 0;

  do {

    resultat += "i = " + i + "<br>";

    i++;

  } while (i < 5);

  resultat +=

    "--> A utiliser si on ne sait pas le nombre d'itérations au démarrage de la boucle mais avec un passage obligatoire";

  document.getElementById("info").innerHTML = resultat;

}

# Exercice 8 Tableau d’objets JSON

On a réalisé cet exercice durant la première semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Connaître et utiliser les structures de tableau JSON
* Parcourir et ressortir des éléments de tableau JSON en Javascript

## Réalisation

Voici l’interface que l’on avait à disposition pour faire cet exercice, et le résultat de celui-ci lorsque l’on appuie sur le bouton Parcourir un tableau JSON.

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

## Questions

Comment faire une boucle sur un tableau JSON et comment reprendre des informations de chaque itération de la boucle ?

Voici le code que l’on a réalisé pour faire ça.

function parcourirUnTableauJSON() {

  const tabjson = {

    personnes: [

      { prenom: "John", nom: "Doe", age: 44 },

      { prenom: "Anna", nom: "Smith", age: 32 },

      { prenom: "Peter", nom: "Jones", age: 29 },

    ],

  };

  let resultat = "";

  for (let i = 0; i < tabjson.personnes.length; i++) {

    let personne = tabjson.personnes[i];

    resultat+= i+" ";

    for (let f in personne) {

      resultat += personne[f] + " ";

    }

    resultat+="<br>";

  }

  document.getElementById("info").innerHTML = resultat;

}

On peut voir que l’on a dû faire deux boucles pour pouvoir afficher les données sur la page web. La première boucle permet de passer pour chaque personne et la deuxième boucle, va permettre de passer pour chaque information qui sont dans la personne.

## Exercice 9 Démo sur les objets en JavaScript

On a réalisé cet exercice durant la deuxième semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Connaître et utiliser toutes les sortes de boucles en JavaScript
* Gérer les évènements JavaScript

## Réalisation

Voici l’application que l’on avait à disposition pour pouvoir tester les différentes manières de créer un objet avec une classe.

Une image contenant texte, Police, ligne, capture d’écran

Description générée automatiquement

## Questions

Voici les questions auxquelles on a dû répondre pour cet exercice.

Comment créer un objet avec la classe (selon la dernière manière décrite dans cet exemple)

Voici le code qui permet de faire cela. Cela va beaucoup ressembler à ce que l’on fait lorsque l’on utilise où créer des classes en java.

class Personne {

constructor(prenom, nom, age) {

this.prenom = prenom;

this.nom = nom;

this.age = age;

}

toString() {

return this.nom + " " + this.prenom + " (" + this.age + ")";

}

}

Et voici le résultat que l’on a.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

## Exercice 10 Programmation orientée « objets »

On a réalisé cet exercice durant la deuxième semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Découvrir le concept de POO en JavaScript
* Organiser son code de manière structurée

## Réalisation

Voici l’application que l’on a utilisé pour cet exercice.

Une image contenant texte, capture d’écran, affichage, logiciel

Description générée automatiquement

Voici la classe personne que l’on a fait. Le toString permet d’afficher les informations dans la partie texte de l’application.

function Personne(prenom, nom, age) {

    this.prenom = prenom ;

    this.nom = nom;

    this.age = age;

  }

  Personne.prototype.toString = function(){

    return this.prenom+" "+this.nom +" ("+this.age+")"+"\n";

  }

Dans le worker.js, on utilise une fonction qui est privé. Pour avoir une fonction privée, il faut mettre un underscore avant.

function \_afficherPersonnes() {

  let txt = "<ul>";

  for (let i = 0; i < personnes.length; i++) {

    txt+="<li>";

    txt+="<a href='#' onclick='selectionnerPersonne("+i+")'/>";

    txt += personnes[i].toString();

    txt+="</a>";

    txt+="</li>";

  }

  txt += "</ul>";

  document.getElementById("info").innerHTML = txt;

}

Voici le code du indexCtrl.js qui nous permet d’afficher les noms des personnes avec des liens.

function \_afficherPersonnes() {

  let txt = "<ul>";

  for (let i = 0; i < personnes.length; i++) {

    txt+="<li>";

    txt+="<a href='#' onclick='selectionnerPersonne("+i+")'/>";

    txt += personnes[i].toString();

    txt+="</a>";

    txt+="</li>";

  }

  txt += "</ul>";

  document.getElementById("info").innerHTML = txt;

}

## Question

Question à laquelle on a dû répondre pour cet exercice.

Qu’est-ce que le document.onreadystatechange = function()

Voici le code qui se trouve dans le indexCtrl.js.

document.onreadystatechange = function () {

  if (document.readyState === "complete") {

    \_afficherPersonnes();

  }

};

Cela va être utiliser chaque fois que l’on rafraichit la page ou a chaque fois qu’il y a une modification sur l’état du document. Et dans la fonction, une fois que le document aura été chargé, on lancera le code. Et cela va s’effectuer avant le onload. Car si il y a des images ou des vidéos qui sont lourdes, pendant qu’elles seront en train de se charger, le code JS pourra s’effectuer en parallèle.

# Exercice 11 : Programmation orientée "classe" en Javascript

On a réalisé cet exercice durant la deuxième semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Créer et instancier des classes Javascript.
* Découper une application selon le modèle MVC.

## Réalisation

Pour cet exercice, on avait la même interface graphique que pour l’exercice précédent. On a également gardé presque le même code. On a cependant dû modifier le code pour faire des classes. Voici un exemple avec la classe Worker().

class Worker{

  constructor(){

    this.personnes = [

      new Personne("John", "Doe", 44),

      new Personne("Anna", "Smith", 32),

      new Personne("Peter", "Jones", 29),

    ];

    this.personnes.sort();

  }

// premier tri de la liste de personnes

// fonction privée pour retrouver l'index d'une personne dans le tableau, -1 autrement

// il faut comparer avec toString()

trouverPersonne(p) {

  let idx = -1;

  for (let i = 0; i < this.personnes.length; i++) {

    if (this.personnes[i].toString() == p.toString()) {

        idx = i;

    } else {

    }

  }

  return idx;

}

// ajouter une personne dans la liste des personnes si pas trouvée

ajouterPersonne(p) {

  let idx = this.trouverPersonne(p);

  if (idx == -1) {

    this.personnes.push(p);

  } else {

  }

}

// supprimer une personne dans la liste des personnes si trouvée

supprimerPersonne(p) {

    let idx = this.trouverPersonne(p);

  if (idx == -1) {

  } else {

    this.personnes.splice(idx, 1);

  }

}

}

Pour les méthodes, on va enlever le \_ pour les privates et le function. Et ensuite on devra mettre le this pour toutes les méthodes et les attributs qui sont dans la classe dans laquelle on se trouve. Et sinon, on devra mettre this.leNomDeLaClasseDansLaquelleSeTrouveLObject.leNomDeL’objet.

txt += this.wrk.personnes[i].toString();

Durant cet exercice on a également vu la différence entre == et ===. Lorsque l’on utilise trois égale, cela va permettre de comparer le type en plus de la valeur. Cela va être utile étant donné qu’en js les variables ne sont pas typé de bases.

## Questions

Voici les questions auxquelles on a dû répondre.

Une explication sur comment créer une classe en Javascript.

Voici le code qui montre comment on peut créer une classe.

class Ctrl {

  constructor() {

    this.wrk = new Worker();

  }

}

Comment instancier une classe pour qu'elle soit accessible dans la vue, le cas échéant, un HTML.

On va pouvoir faire cela avec le code ci-dessous, le code va également se trouver dans l’indexCtrl.js en dessus de la classe Ctrl.

document.onreadystatechange = function () {

  if (document.readyState === "complete") {

    window.ctrl = new Ctrl(); // ou ctrl = new Ctrl();

    ctrl.afficherPersonnes();

  }

};

Pour que l’objet Ctrl soit disponible dans la vue, l’objet « ctrl » doit être stocké dans l’objet « window » du navigateur car les méthodes de l’ « indexCtrl » sont appelées depuis l’ « index.html ». L’objet window doit être ajouté à la variable ou on ne met rien et la variable devient globale

# Exercice 12 : Fonctions et IIFE en JavaScript

On a réalisé cet exercice durant la deuxième semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

Déclarer des méthodes traditionnelles, anonymes et "flèches".

## Réaliser

Pour réaliser cet exercice, on a uniquement essayé le code dans le navigateur. On a également essayé des fonctions qui sont créer et exécuter directement. IIFE qui est l’abréviation de « Immediately-Invoked-Function  Expression».

Voici un exemple qui montre comment on utilise cela.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

Manière 1

(function() {

let val = 3;

console.log(val) ;

})() ; // => exécution et affichage de 3 dans la console

Manière 2

(() => {

let val = 4;

console.log(val) ;

})() ; // => exécution et affichage de 4 dans la console via une fonction flèche

## Question

Voici la question à laquelle on a dû répondre pour cet exercice.

Quels sont les 3 types de déclarations de fonction possibles en Javascript. Donnez pour chacun un ou plusieurs exemples.

**Déclaration d’une fonction**

Il s’agit de la manière traditionnelle que l’on a utilisé depuis le début de ce module.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

function a() {

let val = 1;

console.log(val) ;

}

a(); // => exécution de la méthode et affichage de 1 dans la console

// si on écrit a(); avant la déclaration de la fonction, cela fonctionne !

**Déclaration d’une expression fonction (fonction anonyme)**

Dans ce cas, on va stocker la fonction dans une variable. Elle sera donc anonyme car elle n’aura pas de nom. Elles seront appelées avec le nom de la variable.

Une image contenant texte, Police, ligne, blanc

Description générée automatiquement

let b = function() {

let val = 2;

console.log(val) ;

} ;

b(); // => exécution de la méthode et affichage de 2 dans la console

**Déclaration d’une fonction flèche (arrow function) ; 3 cas de figure**

Cela va principalement être utile car la fonction sera plus courte à écrire.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

let c = (a,b) => a + b ;

console.log(c(2,5)) ; // = 7

let d = val => val\*val ; // avec un paramètre

console.log(d(5)); // = 25

let e = (a,b) => { // avec plusieurs instructions

let somme = a+b;

return somme;

};

console.log(e(5,7)); // = 12

# Exercice 13 : Les cookies

On a réalisé cet exercice durant la deuxième semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Créer et stocker des informations dans un cookie,
* Lire les données contenues dans un cookie.

## Réalisation

Voici l’interface graphique que l’on avait à disposition pour réaliser cet exercice.

Une image contenant texte, capture d’écran, Police, Rectangle

Description générée automatiquement

Voici le code qui permet de créer la date d’expiration du cookie. On utilise setTime et il faut faire \*60 et \*1000 pour transformer les 3 minutes en milliseconde. On fait cela car setTime travaille avec les millisecondes.

  let dateAujourdhui = new Date();

  dateAujourdhui.setTime(dateAujourdhui.getTime() + minutes \*60\*1000)

Pour pouvoir vérifier que les cookies aient bien été créer, on va devoir publier le site sur internet pour avoir le serveur et ensuite lorsque l’on va sous l’onglet information on va pouvoir voir les informations que contient le cookie.

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

Et ensuite lorsque le temps d’expiration n’est pas arrivé, si on recharge la page, les informations du cookie vont se réafficher.

## Questions

Voici les questions auxquelles on a dû répondre pour cet exercice.

Qu'est-ce qu'un cookie et à quoi sert-il ?

Il s’agit de petit fichier texte qui peuvent entreposer un nombre limité de données. Cela va être utile pour conserver des contenus de variables mémoires réutilisables au fil de la navigation au travers des nombreuses pages de site web.

Comment créer un cookie et stocker une chaîne de caractère. Donner un exemple.

Voici le code permettant de créer le cookie qui stockera la chaîne de caractère.

 document.cookie = "cookie\_exercice\_13=" + contenu + "; expires=" + dateAujourdhui + "; path=/";

Comment récupérer une chaîne de caractère depuis un cookie. Donner un exemple.

C’est la méthode getCookie qui va faire cela.

function getCookie() {

  // Tester la présence du cookie. Pour cela, il faut tester si la taille du cookie est supérieure à 0.

  if (document.cookie.length > 0) {

    // Récupérer le contenu du cookie et en extraire la partie qui se situe après le "="; Vous pouvez utiliser la méthode "split"

    // liée aux chaînes de caractères.

    let contenuCookie = document.cookie;

    let tableauContenu = contenuCookie.split("=");

    tableauContenu = tableauContenu[1].split(" ");

    // Convertir la chaîne de caractères en objet Json.

    personneJson = {

      prenom: tableauContenu[0],

      nom: tableauContenu[1]}

    // Charger les deux champs avec la valeur du prénom et du du nom contenu dans l'objet json.

    document.getElementById("prenom").value = personneJson.prenom;

    document.getElementById("nom").value= personneJson.nom;

    console.log("Data retrieved from cookie");

  }

Une explication du code à utiliser pour convertir un objet JSON en chaîne de caractères et pour convertir une chaîne de caractères en objet JSON.

Durant cet exercice on s’est compliqué la vie en réalisant cela « à la main », cependant, il y a deux méthodes qui permettes de réaliser ces conversions. La première permet de transformer un objet JSON en String. La méthode stringify permet de faire cela.

const obj = {name: "John", age: 30, city: "New York"};  
const myJSON = JSON.stringify(obj);

Pour faire le contraire et transformer un String en Objet JSON, on utilisera la méthode parse.

const obj = JSON.parse('{"name":"John", "age":30, "city":"New York"}');

# Exercice 14 : Bases de jquery

On a réalisé cet exercice durant la deuxième semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Expliquer ce que la librairie jquery.
* Ajouter et utiliser jquery dans un un scripte javascript.

## Réalisation

Pour pouvoir répondre à ces questions, on a principalement utilisé le site web suivant.

Lien : <http://www.jcsinfo.ch/demo/jquery>.

Le code ci-dessous, permet de voir les différences entre JavaScript et JQuery pour le traitement de base des éléments du DOM.

let info = document.getElementById("info");

// let info = document.querySelector("#info");

info.textContent="Une info JavaScript!";

info.innerHTML=("<b>C'est bold</b>");

info.style.backgroundColor="blue";

info.style.color="red";

info.addEventListener("click", () => { info.textContent="click JS"});

info.append(document.createElement("div"));

$("#info").text("C'est une info JQuery");

$("#info").html("<b>C'est bold</b>");

$("#info").css("color","red");

$("#info").css({"color": "red"});

$("#info").css({color:"red", backgroundColor:"blue"});

$("#info").css({"color":"green", "background-color":"cyan", "font-size":"1.5em"});

$("#info").click(() => { $("#info").text("click JQuery")});

$("#info").append("<div>C'est un div</div>");

## Questions

Voici les questions auxquelles on a dû répondre pour cet exercice.

Expliquez brièvement ce qu'est jquery et à quoi il sert.

C’est une librairie open source qui permet de simplifier l’interaction entre du HTML et du Javascript. Cela permet une simplification de manipulation programmée du DOM.

Comment reconnaît-on du code jquery ?

Il est facilement reconnaissable car c’est lorsque le $ est utiliser.

Quelles sont les 2 tâches principales de jquery ?

* Trouver un ou plusieurs éléments d'une page HTML.
* Faire quelque chose avec.

[$](http://api.jquery.com/jQuery)("div").[addClass](http://api.jquery.com/addClass)("special");

Expliquer comment trouver des éléments du DOM avec jquery selon un nom de tag, une classe CSS ou un id.

Le $ représente l’objet JQuery qui est une fonction Java. Le « div » va permettre de réaliser la recherche, lui qui est un sélecteur CSS. Et le code suivant va retourner un set JQuery de 0 à N nœud du DOM

$("div")

Décrivez la fonction jquery qui est appelée lorsque la page a été chargée. Donnez un exemple.

Il s’agit de l’événement « ready » qui va partir à l’instant même ou le DOM sera prêt et que l’on pourra travailler avec. Et cela même si des images sont encore en chargement. Il faut donc mettre le code JQuery dans un bloc ready pour que le chargement du code se passe bien.

$(document).[ready](http://api.jquery.com/ready)(function(){   
    // Votre code jQuery doit se trouver ici   
});

Comment retrouvez un élément parent, le suivant, le précédent, un enfant ou les éléments frères avec jquery ?

Parfois une sélection par sélecteurs CSS n’est pas suffisante donc JQuery propose une suite de méthodes pour parcourir le DOM. Voici les différentes méthodes.

* .parent()
* .next()
* .prev()
* .children()
* .siblings

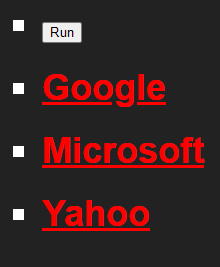
[$](http://api.jquery.com/jQuery)("button").[parent](http://api.jquery.com/parent)().[css](http://api.jquery.com/css)("border", "3px solid red");

Modifier le css d'un élément de l'élément sélectionné ?

Pour cela, on peut utiliser la méthode .css comme le montre le code suivant.

[$](http://api.jquery.com/jQuery)("li a").[css](http://api.jquery.com/css)({   
  color: "red",   
  fontWeight: "bold"   
});

Et voici l’output de ce code lorsque l’on appuye sur le bouton Run.



Ajouter un élément avant ou après l'élément sélectionné ?

Le code suivant utilise la méthode .after() qui ajoute un élément après, à chaque fois que l’on appuye sur le bouton run cela ajoutera une petite icone à côté du texte.

[$](http://api.jquery.com/jQuery)("a[target=\_blank]")   
  .[after](http://api.jquery.com/after)("<img src='images/open.png'/>");

Une image contenant texte, Police, capture d’écran, Graphique

Description générée automatiquement

Et pour mettre avant, il faudra utiliser la méthode .before.

$("button").click(function(){  
  $("p").before("<p>Hello world!</p>");  
});

Supprimer un élément ?

Pour cela il faut utiliser la méthode .remove().

Gérer les événements (click, hover, etc.)

Pour gérer les événements on va avoir différente méthode tel que le .click(), le .hover() ou encore le .toggle().

**.click**

[$](http://api.jquery.com/jQuery)("a.menu").[click](http://api.jquery.com/click)(function(){   
  $(this).[next](http://api.jquery.com/next)().[toggle](http://api.jquery.com/toggle)();   
  return false;   
});

**.hover()**

[$](http://api.jquery.com/jQuery)("li").[hover](http://api.jquery.com/hover)(function(){   
  $(this).[animate](http://api.jquery.com/animate)({marginLeft: 38, marginRight: 0});   
}, function(){   
  $(this).[animate](http://api.jquery.com/animate)({marginLeft: 18, marginRight: 18});   
});

Cacher ou afficher un élément ?

Pour cela il s’agit de la méthode .hide() pour cacher l’élément et de la méthode .show() pour l’afficher.

[$](http://api.jquery.com/jQuery)("div.block").[hide](http://api.jquery.com/hide)("slow", function(){   
  $(this).[show](http://api.jquery.com/show)("slow");   
});

Avec ce code, lorsque l’on appuye sur Run, cela va cacher puis réafficher l’image.

Une image contenant texte, capture d’écran, Police, Graphique

Description générée automatiquement

Comment lire un fichier avec ajax ?

AJAX = Asynchronous JavaScript and XML. Et ci-dessous, il y a le code permettant de lire un fichier XML avec du Ajax.

$.[ajax](http://api.jquery.com/jQuery.ajax)({   
    url: "file.xml",   
    success: function( xml ) {   
        $(xml).[find](http://api.jquery.com/find)("tab").[each](http://api.jquery.com/each)(function(){   
            [$](http://api.jquery.com/jQuery)("ul").[append](http://api.jquery.com/append)(   
              "<li>" + $(this).[text](http://api.jquery.com/text)() + "</li>");   
        });   
    }   
});

Comment lire un fichier JSON avec la méthode $.getJSON()?

Voici le code qui utilise cette méthode pour lire le fichier JSON.

$.[getJSON](http://api.jquery.com/jQuery.getJSON)("file.json", function( obj ) {   
    for ( var prop in obj ) {   
        [$](http://api.jquery.com/jQuery)("ul").[append](http://api.jquery.com/append)(   
            "<li>" + prop + ": " + obj[prop] + "</li>");   
    }   
});

Comment charger un fichier HTML dans un div existant ?

[$](http://api.jquery.com/jQuery)("div.load").[load](http://api.jquery.com/load)("file.html");

Comment intégrer jquery dans vos pages ?

Voici la balise qui permet de faire cela.

<script src='http://code.jquery.com/jquery.js'></script>

# Exercice 15 : Première utilisation de JQuery

On a réalisé cet exercice durant la deuxième semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Référencer la librairie jquery dans du code HTML.
* Accéder à des éléments du code HTML avec jquery.
* Modifier les propriétés d'un document à l'aide de jquery.

## Réalisation

Voici l’application que l’on avait a disposition.

Une image contenant texte, capture d’écran, jaune, Police

Description générée automatiquement

Et lorsque l’on appuye sur une couleur, cela change la couleur de l’application. On a dû réaliser ce code de deux manières différentes, une fois en javascripte et une fois avec jquery.

Voici la méthode changerCouleur que l’on a fait avec du Jquery.

   changerCouleur(couleur){

        $("#container").fadeOut(1);

        $("#container").hide();

        $("#container").css("background-color", couleur);

        if(couleur== "green" || couleur== "blue" || couleur == "red"){

            $("#titre").css("color", "white");

        }else{

            $("#titre").css("color", "black");

        }

        $("#container").slideToggle();

    }

  }

Et voici la même méthode mais avec le Javascript.

changerCouleur(couleur) {

  document.getElementById("container").style.backgroundColor = couleur;

  if(couleur== "green" || couleur== "blue" || couleur == "red"){

    document.getElementById("titre").style.color = "white";

}else{

  document.getElementById("titre").style.color = "black";

}

}

Cependant dans cette méthode-là, il n’y a pas les animations.

## Questions

Voici les questions auxquelles on a dû répondre pour cet exercice.

Comment récupérer la valeur d'un champ avec jquery.

Pour cela, il faut utiliser la méthode .val

Comment une propriété css avec jquery.

  $("#container").css("background-color", couleur);

        if(couleur== "green" || couleur== "blue" || couleur == "red"){

            $("#titre").css("color", "white");

        }else{

            $("#titre").css("color", "black");

        }

Comment utiliser une animation avec jquery.

Voici deux examples

$("#container").fadeOut(1);

$("#container").slideToggle();

# Exercice 16 : Utilisation de jquery

On a réalisé cet exercice durant la deuxième semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Ajouter des écouteurs via jquery.
* Sélectionner des éléments du DOM via jquery.
* Ajouter et supprimer des classes CSS à des éléments HTML via jquery.
* Récupérer des valeurs d'un formulaire via jquery.
* Ajouter des éléments au DOM via jquery.
* Extraire la valeur d'un attribut HTML via jquery.

## Réalisation

Durant cet exercice, on a utilisé JQuery. Voici l’interface que l’on avait à disposition.

Une image contenant texte, Appareils électroniques, capture d’écran, Police

Description générée automatiquement

Dans le code ci-dessous, on a toute la structure de notre fichier main.js, avec d’abord le document.ready à l’intérieur duquel on va mettre tous les écouteurs. Et ensuite on va créer les fonctions que l’on utilise dans les écouteurs. Et dans les fonctions on va utiliser les différentes méthodes que JQuery nous met à disposition.

$(document).ready(function () {

  $("#btnShow").on("click", function () {

    afficher();

  });

  $("#btnHide").on("click", function () {

    cacher();

  });

  $("#btnToggle").on("click", function () {

    afficherCacher();

  });

});

function cacher() {

  $("#div\_1").hide();

}

function afficher() {

  $("#div\_1").show();

}

function afficherCacher() {

  $("#div\_1").toggle();

}

## Questions

Voici les questions auxquelles on a dû répondre pour cet exercice.

Comment ajouter un écouteur via jquery.

Voici le code qui ajoute un écouteur, on va devoir mettre cet écouteur à l’intérieur du document ready.

$("#btnShow").on("click", function () {

    afficher();

  });

Sélectionner des éléments du DOM via jquery.

Ajouter et supprimer des classes CSS à des éléments HTML via jquery.

function ajouteClasse() {

  $("li:first").addClass("boldBlueText");

}

function supprimerClasse() {

  $("li:first").removeClass("boldBlueText");

}

function ajouteSupprimerClasse() {

  $("li:last").toggleClass("boldBlueText");

}

Récupérer des valeurs d'un formulaire via jquery.

Ajouter des éléments au DOM via jquery.

Extraire la valeur d'un attribut HTML via jquery.

# Exercice 17 WebServices

Voici le lien d’un site web qui explique bien ce qu’est une API.

Lien : <https://blogs.mulesoft.com/fr/learn-apis/api-led-connectivity/what-are-apis-how-do-apis-work/>

## SOAP/REST

SOAP est l’abbréviation de Simple Object Access Protocol, REST est l’abbréviation de representational state transfer. Ce sont donc deux architectures logicielles qui sont utiliser pour le WebServices. Cependant durant ce module, on utilisera le REST.

## Get

Voici un schéma qui explique une requête get.

Une image contenant texte, capture d’écran, logiciel, Police

Description générée automatiquement

Et sur l’image ci-dessous, on peut voir les différents retours que l’on peut avoir.

Une image contenant texte, capture d’écran, Police, nombre

Description générée automatiquement

## Post

On va utiliser le post lorsque l’on va créer de nouvelles entrées. La différence avec le put est que si l’on effectue 10 fois le même post qui crée un joueur, le joueur sera créé 10 fois.

## Put

Le put va lui donc être utiliser pour modifier une requête. Si on effectue 10 fois le même put, cela va le faire 1 seule fois et ensuite les 9 autres fois n’auront aucun effet.

## Delete

C’est ce qui va nous permettre de supprimer les données. On va pouvoir ajouter des paramètres à la requête comme on peut le voir sur l’image ci-dessous qui a été faite sur l’application postman.

Une image contenant texte, capture d’écran, ligne, nombre

Description générée automatiquement

# Postman

C’est une application que l’on va utiliser pour tester les requêtes.

# Exercice 18 WebServices - Application HTML appelant WS PHP

On a réalisé cet exercice durant la 4ème semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* Mettre en ligne un webservice PHP
* Effectuer des appel AJAX en JS

## Réalisation

Voici l’interface que l’on avait à disposition.

Une image contenant texte, capture d’écran, ligne, Police

Description générée automatiquement

On a réalisé cet exercice avec des posts, mais on aurait également pu le faire avec le fichier php pour les get.

## Questions

Voici les questions auxquelles on a dû répondre pour cet exercice.

* Comment faire un appel GET / POST en Ajax

Voici comment on va faire pour le Post. On fait donc ce code dans le fichier httpServ.js.

* class HttpServ  {
* constructor(){
* }
* celcius2Fahrenheit(degres, successCallback, errorCallback) {
* let url = "https://jacquierr.emf-informatique.ch/Module\_307/Exercices/Exercice\_18/convert-temp\_p\_xml.PHP";
* let param = "Temperature=" + degres + "&FromUnit=C&ToUnit=F";
* // envoi de la requête
* $.ajax(url, {
* type: "POST",
* contentType: "application/x-www-form-urlencoded; charset=UTF-8",
* data: param,
* success: successCallback,
* error: errorCallback
* });
* }
* }
  + Comment donner des paramètres

C’est avec le data : param et on aura définit la variable au-dessus.

let param = "Temperature=" + degres + "&FromUnit=C&ToUnit=F";

* + Comment donner une méthode de retour en cas de succès
* success: successCallback,
  + Comment donner une méthode de retour en cas d'erreur
* error: errorCallback

# Exercice 20 Application de login sur un serveur PHP

On a réalisé cet exercice durant la 4ème semaine de ce module.

## Objectif

Voici les objectifs de cet exercice :

* D'expliquer et implémenter une application selon le modèle MVC.
* Utiliser des requêtes AJAX entre un client Web et un serveur.
* Charger des vues dynamiquements dans une application.

## Réalisation

Voici l’interface que l’on avait à disposition pour cet exercice.

Une image contenant texte, capture d’écran, logiciel, affichage

Description générée automatiquement

Lorsque l’on met créer un compte, il y a une page différente qui s’ouvre.

Une image contenant texte, Appareils électroniques, capture d’écran, affichage

Description générée automatiquement

Et voici ce que cela nous affiche quand le login est correct.

Une image contenant texte, capture d’écran, affichage, Police

Description générée automatiquement

## Questions

Voici les questions auxquelles on a dû répondre pour cet exercice.

* Que signifie SPA ? Expliquez le concept.

SPA est l’abréviation de Single Page Application, il s’agit donc d’une application web qui est accessible via une page web unique. Le but est d’éviter le chargement d’une nouvelle page à chaque action demandée et ainsi de fluidifier l’éxperience utilisateur.

* Quelle méthode de Jquery pouvez-vous utiliser pour charger le code HTML d'une page, dans un élément HTML d'une autre page. Donnez un exemple.
* // avec arrow function
* loadLogin() {
* this.vue.chargerVue("login", () =>  new LoginCtrl());
* }
* // avec function classique
* loadAccueil(langue) {
* this.vue.chargerVue("accueil", function() {
* new AccueilCtrl(langue);
* });
* }
* loadCompte() {
* this.vue.chargerVue("compte", () =>  new CompteCtrl());
* }
* A quoir sert la méthode ajax $.ajaxSetup() ?

La méthode ajaxSetup() définit les valeurs par défaut pour les futures requêtes AJAX. Voici la syntaxe de ce code.

$.ajaxSetup({name:value, name:value, ... })

# Projet « Villes »

On a réalisé le projet durant les dernières semaines de ce module.

# Analyse

Voici la partie analyse de ce projet.

## Description

Le projet se nomme Villes, c’est un site web qui permet de recenser quelques villes de suisse ainsi que les pays du monde. Pour cela, il y a deux API différentes qui sont utilisée. On va donc avoir diverses informations sur les villes et les pays. Il y aura également une carte qui montre où se trouve les villes de suisses.

**Attention :** Pour pouvoir avoir les données des API, il faut installer le plugin pour le CORS. Sinon cela ne fonctionne pas et il y aura des erreurs pour aller chercher les données des API. Voici l’extension qui doit être installé sur le navigateur.

Une image contenant texte, capture d’écran, Police, ligne

Description générée automatiquement

## API

Voici les deux apis que j’ai choisis pour ce projet.

Lien API pays du Monde : <https://mettraux.best/pays>

Lien API villes Suisse : <https://mettraux.best/ville/CHE>

Etant donné le fait que le site nominatim openstreetmap a été bloqué pour l’EMF, les valeurs de longitude et de latitude des différentes villes de suisse ont été mises en dure. Cependant pour une amélioration, on pourrait utiliser l’api d’openstreetmap.

## Use Case

Voici le diagramme Use Case du projet.

Une image contenant texte, diagramme, capture d’écran, ligne

Description générée automatiquement

## Maquettes

On a utilisé moqups pour réaliser les maquettes de notre site web. Voici la première page (Home).

Une image contenant texte, capture d’écran, conception

Description générée automatiquement

Et voici le modèle de page qui sera utiliser pour afficher les données des API.

Une image contenant texte, capture d’écran, logiciel, Page web

Description générée automatiquement

## Test web Service

Pour cela on a utilisé l’application Postman. Grâce à cette application et étant donné que pour les API que j’ai choisie il n’y avait pas de documentation, on va pouvoir se référer à cela pour savoir comment se nommes les données et pour que l’on puisse donc les intégrés dans le site web.

Une image contenant texte, capture d’écran, logiciel, nombre

Description générée automatiquement

Et on va donc faire cela avec les deux API.

# Conception

Voici la partie conception de ce projet.

## Diagramme nav

Voici le diagramme de navigation pour le projet.

Une image contenant diagramme, conception

Description générée automatiquement

# Code

Le code en entier se trouve sur GitHub, cependant voici quelques bouts de code qui sont intéressant. Ci-dessous ou peut voir la deuxième page du site web qui comporte une carte.

Une image contenant carte, atlas, texte

Description générée automatiquement

Dans le streetCtrl.js on va donc définir l’icône qui affichera ou la ville se trouve, dans ce cas-ci, un logo ville.

    const RedMarkerIcon = L.icon({

      iconUrl: "images/ville\_318-108709.webp",

      iconSize: [36, 60],

      iconAnchor: [9, 30],

      popupAnchor: [0, -20],

    });

Ensuite on va définir l’endroit ou la carte sera initialement ainsi que le zoom qu’elle aura. On va donc faire que la carte se trouve initialement sur la suisse.

mapid = L.map("mapid").setView([lat, lon], 8);

Et ensuite on définit avec la longitude et la latitude des villes ou il y aura les marqueurs et on fait également que lorsque l’on clique sur le marqueur, le nom de la ville s’affiche.

    var berne = L.marker([latBern, lonBern], { icon: RedMarkerIcon }).addTo(mapid);

  berne.bindPopup("Bern");

  var zurich = L.marker([latZurich, lonZurich], { icon: RedMarkerIcon }).addTo(mapid);

  zurich.bindPopup("Zürich");

  var geneve = L.marker([latGeneve, lonGeneve], { icon: RedMarkerIcon }).addTo(mapid);

  geneve.bindPopup("Genève");

  var basel = L.marker([latBasel, lonBasel], { icon: RedMarkerIcon }).addTo(mapid);

  basel.bindPopup("Basel");

  var lausanne = L.marker([latLausanne, lonLausanne], { icon: RedMarkerIcon }).addTo(mapid);

  lausanne.bindPopup("Lausanne");

Il y a également les entêtes qui était assez compliquer à trouver. On va donc créer un Ctrl.js par page html. Et dans ce fichier on va d’abord devoir supprimer les classes qu’il y a et ensuite ajouter la classe que l’on veut. Dans le CSS on ajoute donc un background-image. Et dans le js on modifie aussi le titre pour chaque page.

class homeCtrl {

    constructor() {

      $("#entete").removeClass();

      $("#entete").addClass("home");

      $("#title").text("Villes");

    }

  }

# Conclusion

C’est un module que j’ai apprécié même si il n’était pas facile. Je trouve que cela allait un peu trop vite avec les exercices et qu’à la fin il ne nous restait pas assez de temps pour le projet. Mais j’ai trouvé très intéressant de retravailler avec VS Code et de travaillé avec du javascript car on avait uniquement essayé cela lors du module 101 en première année.