Rapport personnel

Date de création : 15.05.2023  
Version 1 du 05.06.2023

Eldi Kabashi

|  |  |  |
| --- | --- | --- |
| S:\EMF\CPROF\Identite_EMF\2015-Identites\Logos des sections COULEURS\Logo_EMF-Informatique_FR_RVB_25.jpg |  | Module du 15.05.2023 au 13.06.2023 |

Table des matières

[1 Introduction 5](#_Toc136882945)

[2 Exercices 5](#_Toc136882946)

[2.1 Exercice 1 5](#_Toc136882947)

[2.2 Exercice 2 5](#_Toc136882948)

[2.3 Exercice 3 6](#_Toc136882949)

[2.4 Exercice 4 7](#_Toc136882950)

[2.5 Exercice 5 7](#_Toc136882951)

[2.6 Exercice 6 8](#_Toc136882952)

[2.7 Exercice 7 8](#_Toc136882953)

[2.8 Exercice 8 9](#_Toc136882954)

[2.9 Exercice 9 9](#_Toc136882955)

[2.10 Exercice 10 9](#_Toc136882956)

[2.11 Exercice 11 9](#_Toc136882957)

[2.12 Exercice 12 9](#_Toc136882958)

[2.13 Exercice 13 9](#_Toc136882959)

[2.14 Exercice 14 9](#_Toc136882960)

[2.15 Exercice 15 9](#_Toc136882961)

[2.16 Exercice 16 9](#_Toc136882962)

[2.17 Exercice 17 9](#_Toc136882963)

[2.17.1 Généralité API 9](#_Toc136882964)

[2.17.2 REST 9](#_Toc136882965)

[2.17.3 GET 9](#_Toc136882966)

[2.17.4 POST 9](#_Toc136882967)

[2.17.5 PUT 9](#_Toc136882968)

[2.17.6 DELETE 9](#_Toc136882969)

[2.17.7 SOAP 9](#_Toc136882970)

[2.18 Exercice 18 10](#_Toc136882971)

[2.19 Exercice 20 10](#_Toc136882972)

[3 Introduction du projet 10](#_Toc136882973)

[3.1 Définition 10](#_Toc136882974)

[3.2 Explications API 10](#_Toc136882975)

[3.2.1 Site de référence 10](#_Toc136882976)

[3.2.2 Fonctionnement 10](#_Toc136882977)

[3.2.3 Exemple d’utilisation 10](#_Toc136882978)

[4 Analyse 10](#_Toc136882979)

[4.1 Diagramme 10](#_Toc136882980)

[4.2 Maquette 10](#_Toc136882981)

[5 Conception 10](#_Toc136882982)

[5.1 Diagramme de navigation 10](#_Toc136882983)

[6 Implémentation 10](#_Toc136882984)

[6.1 HTML / CSS 10](#_Toc136882985)

[6.2 Javascript général 10](#_Toc136882986)

[6.3 Javascript spécifiques 10](#_Toc136882987)

[6.4 Descente de code complète pour une action qui fait appel à un service Web 11](#_Toc136882988)

[7 Conclusion 11](#_Toc136882989)

[7.1 Ce que j’ai aimé 11](#_Toc136882990)

[7.2 Ce que j’ai moins aimé 11](#_Toc136882991)

[7.3 Ce que j’ai appris 11](#_Toc136882992)

# Introduction

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

# Exercices

## Exercice 1

Comment ajouter un écouteur "click" via Javascript.

Il faut utiliser la commande « addEventListener », le premier paramètre est l’évènement, ici ce sera « click » et le deuxième paramètre est la fonction que va exécuter l’évènement, ici ce sera « testez ».

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

Il faut rajouter un attribut sur notre balise « button » dans le fichier html. L’attribut se nomme « onclick », nous ajoutons ensuite la fonction qu’exécutera le bouton en click, il faut rajouter les parenthèses après le nom de la fonction.

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

Pour exécuter du code js dans la page html, il faut utiliser l’attribut « onload » dans la balise « body », nous mettrons ensuite notre fonction js principale.

Expliquez ce que signifie DOM.

Document Object Model est le modèle des objets du fichier html, ce modèle se construit sous la forme d’arbre. Cela permet de retrouver et d’interagir avec les éléments html depuis notre fichier js et rendre dynamique l’affichage en html.

Où le script Javascript peuvent être chargés et pourquoi.

Dans la balise body car cela permet de charger tous les éléments de notre page avant de pouvoir exécuter notre code js qui fera référence à certains éléments de notre page. Il faut aussi préciser notre script js dans la balise « head » à l’aide de la balise « script ».

## Exercice 2

À quoi sert l’attribut « placeholder » ?

Il sert à avoir du texte dans un textfield qui disparait lorsqu’on commence à écrire dans le textfield.

À quoi sert l’attribut « autofocus » ?

Il permet d’avoir le focus lors du chargement de la page. Il n’est pas conseillé de le mettre sur plusieurs éléments.

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

Oui, nous avons l’attribut « onclick » suivi de la fonction « validerUtilisateur() » dans notre bouton.

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

Le bouton de l’exercice 1 était un élément « button », ici nous avons l’élément « input » qui se trouve dans un « fieldset » se trouvant dans un « form ».

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

La fonction alert() sert à faire apparaitre une popup avec un message. Confirm() permet de faire apparaitre une popup avec deux boutons : « OK » et « cancel ». Prompt() permet de faire apparaitre une popup où l’utilisateur peut écrire dans un textfield, il peut ensuite appuyer sur le bouton « OK ».

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

Nous ne pouvons pas mettre de contenu dans un input.

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

Pour récupérer un élément HTML, il faut utiliser la commande « document.getElementById() ».

Comment récupérer le texte contenu dans un champ de formulaire ?

Il suffit de rajouter « .value » à la fin de la commande précédente et mettre dans comme paramètre l’id du champ.

Comment écrire sur la console de votre navigateur ? Quelle est l'utilité ? Où se trouve la console de débogage de Chrome ?

Il faut utiliser la fonction « console.log() » et mettre notre message en paramètre. La console se trouve dans l’onglet « console » lorsqu’on inspecte la page.

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

function nomDeLaFonction(paramètres){}

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

let/const nomDeLaVariable = valeur.

Let est utilisé lorsqu’on peut changer la valeur de la variable, const lorsque la valeur est constante.

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

Oui mais nous ne déclarons pas leur type.

Comment convertir un texte en minuscule ?

A l’aide de la fonction toLowerCase().

## Exercice 3

Dans un formulaire html, il existe plusieurs types de boutons :

* + « button » : bouton normal
  + « submit » : bouton qui envoie un formulaire à un serveur, le client est redirigé à la page indiqué dans l’attribut « action ».
  + « reset » : bouton qui réinitialise les valeurs d’un formulaire avec leur valeurs par défaut.
  + « image » : bouton représenté par une image.
  + « checkbox » : bouton qui peut être coché ou décoché.
  + « radio » : bouton qui peut être sélectionné dans un groupe de boutons.
  + « file » : bouton qui permet de sélectionner un fichier.

Un code PHP commence par « <?PHP » et fini par « ?> ».

Pour récupérer des données sur PHP avec les méthodes POST ou GET, il faut utiliser la commande $\_POST/GET[« »] et mettre dans les crochets le nom de notre textfield.

Pour concaténer des strings en PHP, il suffit de mettre un point entre les strings.

Pour convertir un string en minuscule, il faut utiliser la méthode « strtolower ».

Pour renvoyer quelque chose au client depuis le PHP, il faut utiliser la commande « echo ».

## Exercice 4

JSFiddle est un outil permettant de tester du code js dans notre navigateur sans devoir associer des pages html ou css. Nous avons juste à mettre notre bout de code html, css et mettre le code que nous voulons tester en js. Cela permet de ne pas avoir à toujours associer des pages html à notre code js juste pour tester une petite partie.

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

Description générée automatiquement

## Exercice 5

Comment déclarez-vous une variable en Javascript ?

On peut utiliser les mots-clés « let/const », let permet de faire des variables dont la valeur peut changer et const permet de garder toujours la même valeur.

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

Pour faire une constante il faut donc utiliser le mot-clé « const ». Comme en Java, il faut mettre le nom de la constante en majuscules.

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

« var » était utilisé avant 2015, il n’est plus utilisé car il laisse trop de liberté.

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

Cela permet de faire des variables globales.

Combien y a-t-il de types primitifs en Javascript et quels sont-ils ?

Il y en a 7 : string, number, bigint, boolean, undefined, symbol et null.

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

On peut utiliser le mot-clé « typeof » suivi de la valeur ou la variable. Exemple : console.log(typeof variable) ;

## Exercice 6

Comment récupérez-vous le numéro du jour actuel ?

Pour cela, il faut créer une variable qui aura comme valeur « new Date() », il faudra ensuite utiliser la méthode « getDay() ».

Comment pouvez-vous insérer du texte dans une balise html ?

Il faut retrouver la balise, puis utiliser la commande « innerHTML » suivi du texte que l’on veut ajouter.

Comment pouvez-vous insérer du code html dans une balise html ? Par exemple "<b>On est mardi</b>"

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

Pour créer un tableau, il suffit de créer une variable et de lui donner comme valeur « [] ». Nous pouvons mettre des valeurs dans les crochets.

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

Pour ajouter une valeur à la fin, il faut utiliser la commande « push() » avec la valeur entre les parenthèses, pour rajouter au début « unshift() ». Pour supprimer la dernière valeur « pop() » et pour la première « shift ».

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

Comme en Java, il faut mettre le nom du tableau et mettre entre crochet la position de la valeur que l’on veut.

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

Pour cela, il faut faire une boucle qui affichera tous les éléments. Exemple :

noms.forEach(function(nom) {

console.log(nom);

});

## Exercice 7

Les boucles for sont utilisées lorsqu’on connait le nombre d’itérations. Pour la faire, il faut utiliser la commande suivante :

for (let i=0; i<5; i++) {

console.log(i);

}

Ici le nombre d’itération est de 5.

Les boucles while servent lorsqu’on ne connait pas le nombre d’itérations, elles vont exécuter des instructions tant qu’une condition est vraie.

while(i<5){

console.log(i);

i++;

}

Les boucles do while agissent comme les boucles while, cependant, elles vont exécuter une fois les instructions avant de vérifier que la condition est vraie.

## Exercice 8

En javascript, nous pouvons utiliser une boucle « for » en utilisant la taille du tableau présent dans le json pour trouver le nombre d’itérations. Ensuite, pour récupérer la valeur des propriétés des objets présents dans le json, nous pouvons utiliser ce qui s’apparente à une boucle « for-each » en java. Il va s’écrire comme cela :

for (const f in objetDuJson) {

s+= objetDuJson[f] + " ";

}

La constante f permet de passer à travers chaque propriété de notre objet.

## Exercice 9

En javascript, nous pouvons créer des objets à l’aide de classe exactement comme en java. Pour cela, il nous faudra créer la classe, elle comportera un constructeur, des attributs et des fonctions. Pour créer la classe, il nous suffit de créer une variable avec un nom et lui donner comme valeur « new NotreClasse() ». Exemple :

class Eleve {

constructor(prenom, nom, age) {

this.prenom = prenom;

this.nom = nom;

this.age = age;

}

toString() {

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

}

}

## Exercice 10

## Exercice 11

## Exercice 12

## Exercice 13

## Exercice 14

## Exercice 15

Pour récupérer la valeur d’un champ avec jquery, il faut utiliser le symbole « $ » suivi de notre champ entre parenthèses.

## Exercice 16

## Exercice 17

### Généralité API

Une API permet de faire le lien entre le client et le serveur en lui permettant d’interagir avec les données de la base de données du serveur. Le client peut alors faire des requêtes pour récupérer, ajouter, modifier ou supprimer ces données.

### REST

Les Webservices REST fonctionne avec une architecture clien-serveur. Ils utilisent des méthodes HTTP : GET, POST, PUT, PATCH, DELETE. Ils représentent les données dans un format JSON, XML ou HTML. Les méthodes doivent être accessible par les clients, elles doivent donc être publiques.

### GET

Les requêtes GET permettent de récupérer des données. Lors d’une requête GET, l’API retourne les données dans un format JSON ou XML. Dans les requêtes, nous retrouvons le lien https suivi de paramètres permettant de retrouver certaines données. Après la requête, l’API nous retourne un code HTTP pour nous indiquer des informations sur la requête, si la requête a abouti, on reçoit « 200 ».

### POST

Les requêtes POST permettent d’ajouter des

### PUT

### DELETE

### SOAP

## Exercice 18

## Exercice 20

# Introduction du projet

## Définition

Mon site web permet de répertorier les jeux gratuits et de les trier par catégorie, par plateforme et par développeur.

## Explications API

### Site de référence

### Fonctionnement

### Exemple d’utilisation

# Analyse

## Diagramme

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

Description générée automatiquement

## Maquette

Une image contenant diagramme, Rectangle, texte, Parallèle

Description générée automatiquement

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

Description générée automatiquement

Une image contenant texte, capture d’écran

Description générée automatiquement

# Conception

## Diagramme de navigation

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

Description générée automatiquement

# Implémentation

## HTML / CSS

## Javascript général

## Javascript spécifiques

## Descente de code complète pour une action qui fait appel à un service Web

# Conclusion

## Ce que j’ai aimé

## Ce que j’ai moins aimé

## Ce que j’ai appris