

Secteur : **Digital & IA**

Manuel des travaux pratiques

# M105 : Programmer en JavaScript

1<sup>ère</sup> Année

Filière :

Développement  
Digital  
(Tronc commun)



# Remerciements

---

La DRIF remercie les personnes qui ont contribué à l'élaboration du présent document :

## Équipe de conception :

BOUDIAF Saida *Digital learning manager/  
Project manager*

MIHOUBI Fattoum, *Cheffe de projet pédagogique/  
Ingénieure pédagogique*

CROUZOULON Jonathan, *Directeur pédagogique/  
Chef de projet pédagogique*

---

## Équipe de rédaction :

ENAANAI Adil, *Assistant Professor in Computer science (  
Faculty of science Tetouan)/Enseignant-chercheur en informatique*

Les utilisateurs de ce document sont invités à communiquer à la DRIF et au CDC Digital & IA toutes les remarques et suggestions afin de les prendre en considération pour l'enrichissement et l'amélioration de ce module.

# SOMMAIRE

## 1. DÉFINIR LE RÔLE DE JAVASCRIPT DANS LE DÉVELOPPEMENT

- Comparer un langage de script avec un langage compilé  
Comprendre l'architecture client/serveur
- Découvrir l'écosystème de développement

## 2. ACQUÉRIR LES FONDAMENTAUX DE JAVASCRIPT

- Maîtriser la syntaxe JavaScript et ses notions fondamentales
  - Maîtriser les structures de contrôle
  - Utiliser des fonctions & Manipuler les objets

## 3. MANIPULER LES ÉLÉMENTS D'UNE PAGE AVEC DOM

- Comprendre l'arbre DOM, les nœuds parents et enfants
- Connaître les bases de la manipulation du DOM en JavaScript
  - Manipuler les éléments HTML

## 4. GÉRER LES ÉVÉNEMENTS UTILISATEUR

- Comprendre la notion d'événement pour gérer l'interactivité
  - Gérer les éléments d'un formulaire

## 5. MANIPULER JQUERY

- Découvrir jQuery
- Découvrir AJAX



## PARTIE 1

### Définir le rôle de javascript dans le développement

Dans ce module, vous allez :

- Comparer un langage de script avec un langage compilé
- Comprendre l'architecture client/serveur
- Découvrir l'écosystème de développement



04 heures



## Activité 1

### Comparer un langage de script avec un langage compilé

#### Compétences visées :

- Définir un langage de script
- Comprendre le fonctionnement d'un langage de script

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



01 heures

# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Comprendre le fonctionnement d'un script
  - Maîtriser l'écriture des scripts JS associés au code HTML



## Activité 1 :

### Comparaison entre langage de script et langage compilé



#### Exercice 1 :

1. Qu'est-ce que JavaScript ?
2. Quelle est la différence entre Java et JavaScript ?
3. Quels sont les avantages de JavaScript ?
4. Quelle entreprise a développé JavaScript ?
5. Quelles sont les compétences techniques nécessaires pour devenir développeur front-end ?
6. Dans quelle balise HTML peut-on placer le code JavaScript ?
7. Le fichier externe de JavaScript doit-il contenir la balise `<script>` ?
8. Quel est le bon endroit pour insérer un code JavaScript ?
9. Est-ce que Javascript peut accéder au système de fichiers ?

## Activité 1 : Comparaison entre langage de script et langage compilé



### Solution (Exercice 1) :

#### 1. Qu'est-ce que JavaScript ?

- JavaScript est un langage de script côté client qui peut être inséré dans des pages HTML et être interprété par les navigateurs Web.

#### 2. Quelle est la différence entre Java et JavaScript ?

Java	JavaScript
Java est un langage de programmation compilé	JavaScript est un langage de programmation interprété
Les applications Java peuvent s'exécuter sur n'importe quelle machine virtuelle (JVM) ou navigateur	Le code JavaScript est exécuté sur le navigateur uniquement
Les objets de Java sont basés sur les classes, même si nous ne pouvons créer aucun programme en Java sans créer une classe	Les objets JavaScript sont basés sur des prototypes
Java est un langage autonome	JavaScript est inclut dans une page Web et s'intègre à son contenu HTML
Un programme Java utilise plus de mémoire	JavaScript nécessite moins de mémoire, il est donc utilisé dans des pages Web



## Activité 1 :

### Comparaison entre langage de script et langage compilé



#### Solution (Exercice 1) :

##### 3. Quels sont les avantages de JavaScript ?

- JS est un langage côté client.
- JS est relativement rapide pour l'utilisateur final.
- JS ajoute des fonctionnalités aux pages Web.
- JS ne nécessite aucune compilation nécessaire.
- JS est facile à déboguer et à tester.
- JS est indépendant de la plateforme.

##### 4. Quelle entreprise a développé JavaScript ?

- Javascript a été développé par la société NetScape.

##### 5. Quelles sont les compétences techniques nécessaires pour un développeur front-end ?

- Maîtriser les langages HTML, CSS, Javascript, JQuery, NodeJS, et les frameworks tels que Angular, React et viewJS.

##### 3. Quelles sont les compétences techniques nécessaires pour un développeur front-end ?

- Maîtriser les langages HTML, CSS, Javascript, JQuery, NodeJS, et les frameworks tels que Angular, React et viewJS.

##### 4. Dans quelle balise HTML plaçons-nous le code JavaScript ?

- La balise `<script>`.

##### 5. Le fichier externe de JavaScript doit contenir la balise `<script>` ?

- Non.

##### 6. Est-ce qu'un programme Javascript crée un fichier compilé ?

- Dans les sections `<head>` et `<body>` , ou dans un fichier externe.

##### 7. Est-ce que Javascript peut accéder au système de fichiers ?

- Non.



## Activité 2

### Comprendre l'architecture client/serveur

#### Compétences visées :

- Comprendre la composition d'une architecture client/serveur
- Comprendre le fonctionnement d'un système client/serveur pour le cas d'une architecture Web

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



01 heures

# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Maîtriser la notion du client et du serveur ?
  - Comprendre la communication client / serveur ?



## Activité 2 : Comprendre l'architecture client/serveur



### Exercice 1 :

1. Est-ce qu'une machine personnelle pourra jouer le rôle d'un serveur ?
2. Quel est le nombre maximum de clients que peut accepter un serveur ?
3. Citer les différentes limites matérielles d'un serveur quand le nombre de clients est trop grand.
4. Citer les différents types de serveurs de point de vue logiciel.
5. Avec un serveur Web, quel est le logiciel qu'il faut installer chez les clients ?
6. Quelle est la nature des requêtes envoyées par les clients vers le serveur ?
7. Est-ce qu'on peut installer plusieurs serveurs Web dans une même machine ?

## Activité 2 : Comprendre l'architecture client/serveur



### Solution (Exercice 1) :

**1. Est-ce qu'une machine personnelle pourra jouer le rôle d'un serveur ?**

- Oui, il suffit d'installer le serveur logiciel et de configurer l'accès distant.

**2. Quel est le nombre maximum de clients que peut accepter un serveur ?**

- Cela dépend de la capacité matériel du serveur et aussi de la performance du serveur logiciel utilisé. Pour la partie matérielle, on cite le nombre de processeurs et leur vitesse, la capacité de la mémoire RAM disponible, la capacité du disque disponible et la bande passante.

**3. Citer les différentes limites matérielles d'un serveur quand le nombre de clients est trop grand**

- Processeur surchargé, mémoire RAM saturée, Disque dur saturé, bande passante insuffisante.

**4. Citer les différents types de serveurs en point de vue logiciel**

- Serveur Web, serveur d'application, serveur de base de données, serveur de monitoring, ...

**5. Avec un serveur Web, quel est le logiciel qu'il faut installer chez les clients ?**

- Le navigateur web.

**6. Quelle est la nature des requêtes envoyées par les clients vers le serveur web ?**

- Les clients communiquent avec le serveur web à travers les requêtes HTTP.

**7. Est-ce qu'on peut installer plusieurs serveurs Web dans une même machine ?**

- Oui, il faut juste leurs donner des ports de communication différents.



## Activité 3

### Découvrir l'écosystème de développement

#### Compétences visées :

- Découvrir l'environnement de développement
- Découvrir les librairies appropriés (jQuery, React, Vue JS, Angular, ...)

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



02 heures

# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 3 : Découvrir l'écosystème de développement



### Exercice 1 :

1. Quel est le meilleur environnement de développement des applications web ? Citer pourquoi.
2. Quel est le format de transfert de données le plus utilisé entre les applications front-end et back-end ? Citer pourquoi.
3. Est-ce qu'on peut écrire un serveur web avec Javascript ?
4. A quoi sert JQuery ?
5. **Activité de recherche :**
  - Comparer les deux frameworks :
    1. Angular
    2. React



## Activité 3 : Découvrir l'écosystème de développement



### Solution (Exercice 1) :

**1. Quel est le meilleur environnement de développement des applications web ? Citer pourquoi.**

- Le meilleur environnement de développement est l'UNIX, parce que la plupart des sites Internet et des applications web sont hébergés et tournent sur des environnements Linux.

**2. Quel est le format de transfert de données le plus utilisé entre les applications front-end et back-end ? Citer pourquoi.**

- C'est le format JSON, car il est plus léger et permet de transférer les objets complexe d'une manière simple.

**3. Est-ce qu'on peut écrire un serveur web avec Javascript ?**

- Oui, c'est l'idée de NodeJS.

**4. A quoi sert JQuery ?**

- JQuery est une bibliothèque javascript qui permet de fournir des fonctionnalités de haut niveau afin de faciliter le développement Javascript.



## PARTIE 2

### Acquérir les fondamentaux de javascript

Dans ce module, vous allez :

- Maîtriser la syntaxe JavaScript et ses notions fondamentales
- Maîtriser les structures de contrôle
- Utiliser des fonctions
- Manipuler les objets



18 heures



## Activité 1

### Maîtriser la syntaxe javascript et ses notions fondamentales

#### Compétences visées :

- Maîtriser les notions de variables et de données
- Manipuler les expressions et les opérateurs
- Utiliser les types primitifs et objets de base

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



04 heures

# CONSIGNES

## Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## Pour l'apprenant :

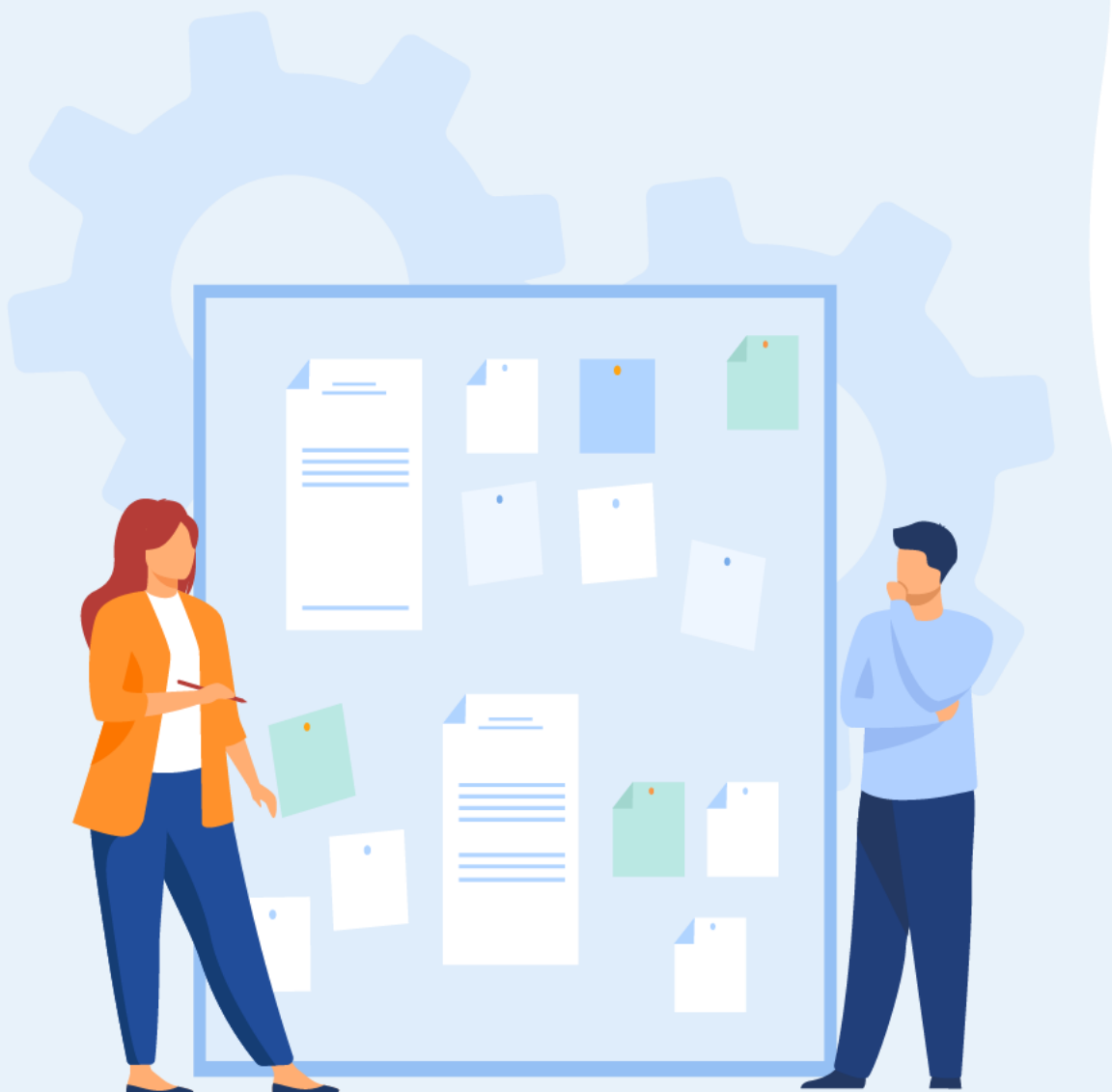
- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## Conditions de réalisation :

- Support de résumé théorique accompagnant

## Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 1 :

### Maîtriser la syntaxe javascript et ses notions fondamentales



#### Exercice 1 :

Quel est le résultat de l'exécution de chacune de ces lignes dans la console ? Pourquoi ?

1. `var a; typeof a;`
2. `var s = '1s'; s++;`
3. `!!"false"`
4. `!!undefined`
5. `typeof -Infinity`
6. `10 % "0"`
7. `undefined == null`
8. `false === ""`
9. `typeof "2E+2"`
10. `a = 3e+3; a++;`

## Activité 1 :

### Maîtriser la syntaxe javascript et ses notions fondamentales



#### Solution (Exercice 1) :

1. undefined
2. NaN
3. true
4. false
5. number
6. NaN
7. true
8. false
9. string
10. 3000

## Activité 1 :

### Maîtriser la syntaxe javascript et ses notions fondamentales



### Exercice 2 :

Donner le résultat logique de chacune des déclarations suivantes :

1. `4 > 3 && 10 < 12`
2. `4 > 3 && 10 > 12`
3. `4 > 3 || 10 < 12`
4. `4 > 3 || 10 > 12`
5. `!(4 > 3)`
6. `!(4 < 3)`
7. `!(false)`
8. `!(4 > 3 && 10 < 12)`
9. `!(4 > 3 && 10 > 12)`
10. `!(4 === '4')`

## Activité 1 :

### Maîtriser la syntaxe javascript et ses notions fondamentales



#### Solution (Exercice 2) :

1. true
2. false
3. true
4. true
5. false
6. true
7. true
8. false
9. true
10. true



## Activité 1 : Maîtriser la syntaxe javascript et ses notions fondamentales



### Exercice 3 :

1. Créer l'interface HTML suivante puis créer une fonction qui permet de calculer la somme des valeurs saisies dans les deux champs et de mettre le résultat dans un élément <label>

+  =8

2. Créer l'interface suivante puis le code javascript qui permet de permuter les valeurs des deux champs de saisie

3. Créer l'interface suivante, puis une fonction qui permet de séparer le nom et le prénom et les mettre dans des éléments <label>. On considère que le nom et le prénom sont constitués d'un seul mot (Utiliser les fonctions indexOf et slice)

Nom:ENAANAI  
Prénom:Adil

## Activité 1 :

### Maîtriser la syntaxe javascript et ses notions fondamentales



#### Solution (Exercice 3) :

1.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Exercice 3-Q1</title>
  <script src="test.js"></script>
</head>
<body>
  <input id="n1" type="text">
  +<input id="n2" type="text">
  =<label id="resultat"></label>
  <button onclick="calculer()">Calculer</button>
</body>
</html>
```

```
function calculer()
{
  n1 = parseFloat(document.getElementById("n1").value);
  n2 = parseFloat(document.getElementById("n2").value);
  document.getElementById("resultat").innerHTML = n1+n2;
}
```

## Activité 1 :

### Maîtriser la syntaxe javascript et ses notions fondamentales



#### Solution (Exercice 3) :

2.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Exercice 3-Q2</title>
  <script src="test.js"></script>
</head>
<body>
  <input id="n1" type="text">
  <input id="n2" type="text">
  <button onclick="permuter()">Permuter</button>
</body>
</html>
```

```
function permuter()
{
  const temp = document.getElementById("n1").value;
  document.getElementById("n1").value=document.getElementById("n2").value;
  document.getElementById("n2").value=temp;
}
```

## Activité 1 :

### Maîtriser la syntaxe javascript et ses notions fondamentales



#### Solution (Exercice 3) :

3.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Exercice 3-Q3</title>
  <script src="test.js"></script>
</head>
<body>
  <input id="nom_complet" type="text" placeholder="Nom et prénom">
  <button onclick="separer()">Séparer</button><br>
  Nom:<label id="nom"></label><br>
  Prénom:<label id="prenom"></label>
</body>
</html>
```

```
function separer()
{
  let nom_complet = document.getElementById("nom_complet").value
  let position_espace=nom_complet.indexOf(" ")
  document.getElementById("nom").innerHTML = nom_complet.slice(0,position_espace)
  document.getElementById("prenom").innerHTML = nom_complet.slice(position_espace+1)
}
```



## Activité 2

### Maîtriser les structures de contrôle

#### Compétences visées :

- Maîtriser les structures alternatives
- Maîtriser les structures itératives

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



04 heures

# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

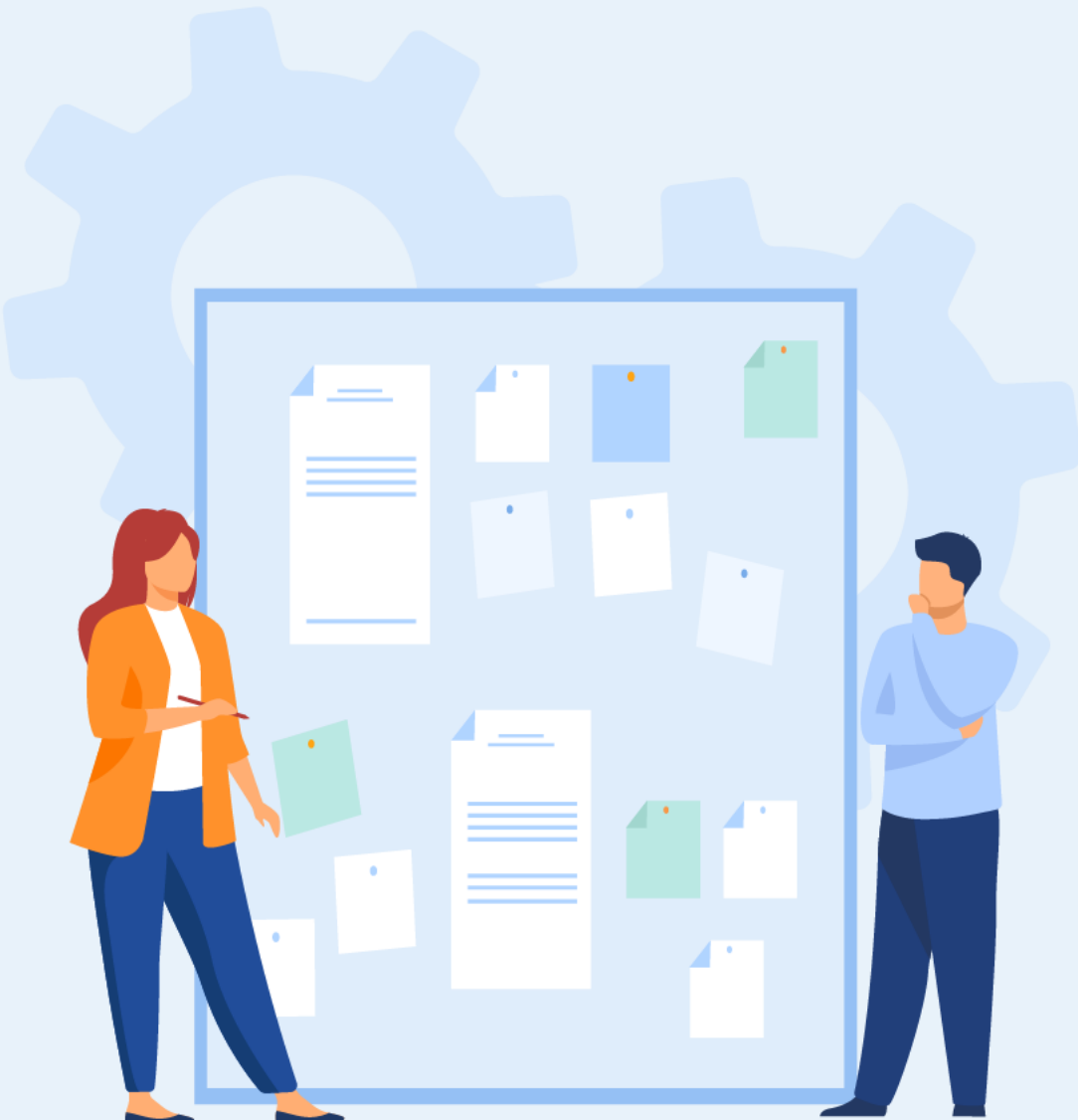
- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisations :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 2 : Maîtriser les structures de contrôle



### Exercice 1 :

1. Ecrire un programme Javascript qui permet de calculer la valeur absolue d'un nombre saisi dans un champ.

Valeur absolue

Valeur absolue:7

2. Ecrire un programme Javascript qui permet de calculer la somme des valeurs absolues de deux nombres.

SVA

Résultat:7

3. Ecrire un programme Javascript qui permet de réaliser le calcul selon l'opération choisie.

/

+  
-  
x  
/

Calculer

Résultat:0.75

## Activité 2 : Maîtriser les structures de contrôle



### Solution (Exercice 1) :

1.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Exercice1-Q1</title>
  <script src="test.js"></script>
</head>
<body>
  <input id="nombre" type="nombre" placeholder="Nombre">
  <button onclick="calculer()">Valeur absolue</button><br>
  Valeur absolue:<label id="valeur_absolue"></label><br>
</body>
</html>
```

```
function calculer()
{
  const nombre = document.querySelector("#nombre").value
  if (nombre<0)
  {   document.querySelector("#valeur_absolue").innerHTML=-nombre }
  else
  {   document.querySelector("#valeur_absolue").innerHTML=nombre }
}
```



## Activité 2 : Maîtriser les structures de contrôle



### Solution (Exercice 1) :

2.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Exercice1-Q2</title>
  <script src="test.js"></script>
</head>
<body>
  <input id="nombre1" type="text" placeholder="Nombre1">
  <input id="nombre2" type="text" placeholder="Nombre2">
  <button onclick="calculer()">SVA</button><br>
  Résultat:<label id="SVA"></label><br>
</body>
</html>
```

```
function calculer()
{
  let nombre1 = parseFloat(document.querySelector("#nombre1").value)
  let nombre2 = parseFloat(document.querySelector("#nombre2").value)
  if (nombre1<0) nombre1=-nombre1
  if (nombre2<0) nombre2=-nombre2
  document.querySelector("#SVA").innerHTML =nombre1+nombre2
}
```

## Activité 2 : Maîtriser les structures de contrôle



### Solution (Exercice 1) :

3.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Exercice1-Q3</title>
  <script src="test.js"></script>
</head>
<body>
  <input id="nombre1" type="text" placeholder="Nombre1">
  <input id="nombre2" type="text" placeholder="Nombre2">
  <select id="operation">
    <option value="+>+</option>
    <option value="->-</option>
    <option value="x">x</option>
    <option value="/">/</option>
  </select>
  <button onclick="calculer()">Calculer</button><br>
  Résultat:<label id="resultat"></label><br>
</body>
</html>
```

## Activité 2 : Maîtriser les structures de contrôle



### Solution (Exercice 1) :

3.

```
function calculer()
{
    let nombre1 = parseFloat(document.querySelector("#nombre1").value)
    let nombre2 = parseFloat(document.querySelector("#nombre2").value)
    const operation = document.querySelector("#operation").value
    let resultat
    switch (operation){
        case '+':resultat = nombre1+nombre2
        break
        case '-':resultat = nombre1-nombre2
        break
        case 'x':resultat = nombre1*nombre2
        break
        case '/':resultat = nombre1/nombre2
        break
    }
    document.querySelector("#resultat").innerHTML=resultat
}
```

## Activité 2 : Maîtriser les structures de contrôle



### Exercice 2 :

1. Dans une page HTML, réaliser un script permettant d'afficher les nombres de 20 à 0 avec décrémentation d'un pas égal à 2.
2. Créer un script qui demande à l'utilisateur de saisir trois variables numériques via une boîte prompt. L'utilisateur est averti s'il ne saisit pas un nombre, et le script lui demande de saisir une valeur numérique. Une fois que l'utilisateur saisit trois valeurs numériques, le script teste les trois variables :
  - Si elles sont égales, écrire : "Les trois variables sont identiques."
  - Si seulement deux sont identiques, écrire : « Deux des variables sont de valeurs égales. »
  - Sinon, écrire : "Les trois variables sont différentes."
3. Utilisez une (ou deux) boucle for() pour produire l'affichage suivant :

```
*  
**  
***  
****  
*****  
*****  
*****  
*****  
*****  
*****
```



#### Remarque

- utilisez la méthode d'affichage : `document.write ("élément à afficher")`.

4. Ecrire un programme Javascript qui permet de demander à l'utilisateur de saisir une note comprise entre 0 et 20 à l'aide de la commande "prompt". Si l'utilisateur saisit une valeur hors de l'intervalle [0,20], on lui demande de saisir à nouveau la valeur et ainsi de suite.

## Activité 2 : Maîtriser les structures de contrôle



### Solution (Exercice 2) :

1.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Exercice2-Q1</title>
</head>
<body>
  <script>
    for (i=20; i>=0; i-=2)
    {
      if(i!=0)
        document.writeln(i+" - ");
      else
        document.writeln(i);
    }
  </script>
</body>
</html>
```

## Activité 2 : Maîtriser les structures de contrôle



### Solution (Exercice 2) :

2.

```
<script>
  var a,b,c;
  do {
    a = parseFloat(window.prompt('Entrez un premier entier '));
  } while(isNaN(a));
  do {
    b = parseFloat(window.prompt('Entrez un deuxième entier '));
  } while(isNaN(b));
  do {
    c = parseFloat(window.prompt('Entrez un troisième entier '));
  } while(isNaN(c));
  if (a === b && b === c)
    alert("Les 3 variables sont identiques.");
  else
    if (b === c || a === c || a === b)
      alert("2 des variables sont de valeurs égales.");
    else
      alert("Les 3 variables sont différentes.");
</script>
```

## Activité 2 : Maîtriser les structures de contrôle



### Solution (Exercice 2) :

3.

```
<script>
  for(i=0;i<10;i++)
  {
    for(j=0;j<i;j++)
      document.write('*')
    document.write('<br>')
  }
</script>
```

4.

```
<script>
  note = parseFloat(prompt("Saisir une note comprise entre 0 et 20; "))
  while(note >20 || note <0)
    note = parseFloat(prompt("Saisir une note comprise entre 0 et 20; "))
  document.write('Merci')
</script>
```

## Activité 3

### Utiliser des fonctions

#### Compétences visées :

- Manipuler les fonctions et les expressions lambdas
- Manipuler les appels asynchrones (callBack, Promise)
- Maîtriser la gestion des exceptions

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



06 heures





# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

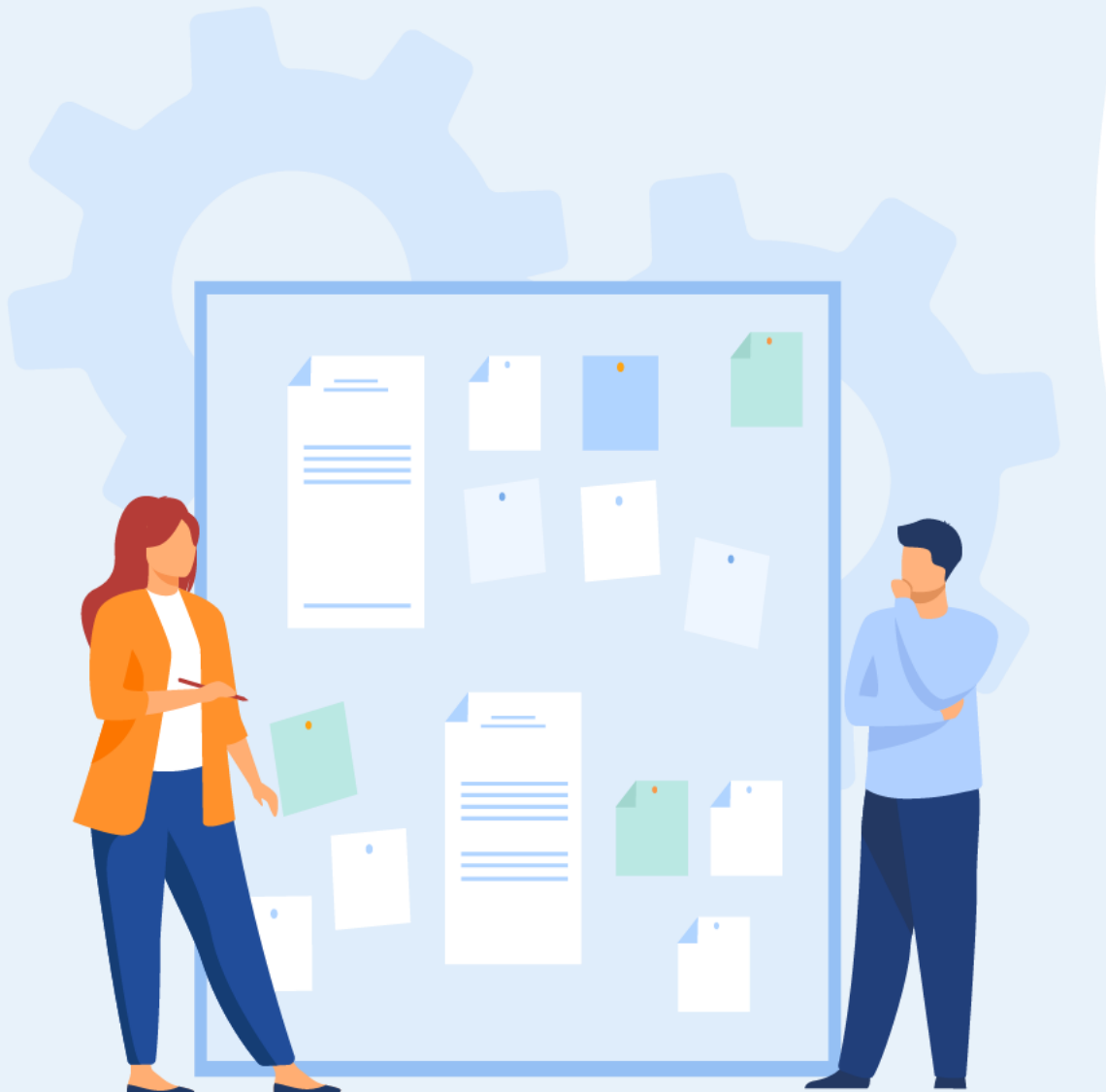
- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 3 : Utiliser des fonctions



### Exercice 1 :

1. Créez une fonction qui :
  - demande la saisie d'un rayon.
  - retourne la surface du cercle correspondant.
2. Affichez le résultat de l'appel à cette fonction en cliquant sur un bouton.

### Exercice 2 :

1. Créez deux variables globales a et b, initialisées respectivement à 3 et à 2.
2. Créez une fonction « multiplie » qui prend un argument x avec la valeur par défaut 8, et qui envoie le résultat de la multiplication de x par 3
3. Créez une fonction affiche, appelée au clic sur le bouton, qui affiche dans des boîtes d'alerte successivement le résultat de « multiplie » appliquée aux variables a et b, puis sans aucun paramètre (en exécutant donc la fonction avec la valeur de x par défaut).

### Exercice 3 :

Écrivez trois fonctions fléchées, appelées fonction1, fonction2 et fonction0 :

1. fonction1 prend un paramètre x et renvoie  $x+5$  ;
2. fonction2 prend deux paramètres x et y et renvoie  $x+y$  ;
3. fonction0 ne prend aucun paramètre, demande la saisie par l'utilisateur d'un nombre et renvoie l'addition de 2 à ce nombre.

## Activité 3 :

### Utiliser des fonctions



#### Exercice 4 :

- Créer un programme javascript avec « promesse » qui permet d'afficher l'heure, la minute et la seconde courante et à chaque fois que la seconde passe à la valeur 00, le programme affiche un bouton à l'utilisateur pour marquer sa présence. S'il ne clique pas sur le bouton au bout de 5 secondes, on affiche "Absent", sinon, on affiche "Présent".

Voici la page HTML de base :

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Exercice 4</title>
  <script src="test.js"></script>
  <link rel="stylesheet" href="test.css">
</head>
<body>
  <div id="presence"></div>
  <div id="montre"></div>
  <button id="marquer" onclick="marquer()">Présent</button>
</body>
</html>
```

## Activité 3 :

### Utiliser des fonctions



#### Exercice 5 :

- Ecrire un programme Javascript qui calcule la surface d'un rectangle tout en gérant l'exception si les valeurs entrées ne sont pas de type numérique.

#### Exercice 6 :

- En utilisant la gestion des exceptions, écrire un programme Javascript qui permet d'afficher le message "Note hors intervalle" si l'utilisateur saisie une note supérieure 20 ou inférieure à 0.

## Activité 3 : Utiliser des fonctions



### Solution (Exercice 1) :

```
function surfaceCercle()  
{  
    var rayon=parseFloat(prompt("Entrez le rayon du cercle : "));  
    return Math.PI*rayon*rayon;  
}
```

### Solution (Exercice 2) :

```
var a=3;  
var b=-2;  
function multiplie(x=8) { return 3*x; }  
function affiche()  
{  
    alert(multiplie(a));  
    alert(multiplie(b));  
    alert(multiplie())  
}
```

### Solution (Exercice 3) :

```
var fonction1 = (x) => x+5;  
var fonction2 = (x, y) => x+y;  
var fonction0 = () => parseFloat(prompt("Saisissez un nombre"))+2;
```

## Activité 3 :

### Utiliser des fonctions



### Solution (Exercice 4) :

```
setInterval(()=>{
    date = new Date()
    document.querySelector("#montre").innerHTML=date.toLocaleTimeString('fr-FR')
    if (date.getSeconds()%10==0){
        myPromise = new Promise(function(resolve,reject){
            document.getElementById("marquer").style.display="block"
            setTimeout(()=>{reject()},5000)
            setInterval(()=>{
                if (document.getElementById("marquer").style.display=="none"){resolve()}
            },200)
        })
        myPromise.then(()=>{
            document.getElementById("presence").innerHTML="Présent"
        }).catch(()=>{
            document.getElementById("presence").innerHTML="Absent"
            document.getElementById("marquer").style.display="none";
        })
    }
},1000)
function marquer()
{
    document.getElementById("marquer").style.display="none"
}
```

## Activité 3 :

### Utiliser des fonctions



### Solution (Exercice 5) :

```
function marquer()
{
    document.getElementById("marquer").style.display="none"
}

function calculerSurface(dim1, dim2)
{
    if (isNaN(dim1) || isNaN(dim2))
    {
        throw 'Paramètre erroné!';
    }
    return dim1*dim2
}

try
{
    calculerSurface(3, 'A');
}
catch (e)
{
    console.error(e);
}
```

## Activité 3 :

### Utiliser des fonctions



### Solution (Exercice 6) :

```
function verification(a)
{
    note = parseInt(document.querySelector("#note").value)
    if (note<0 || note>20) throw new Error("Valeur hors intervalle")
}
```





## Activité 4

### Manipuler les objets

#### Compétences visées :

- Maîtriser la création d'objet et la manipulation des objets
- Apprendre à utiliser les objets natifs
- Utiliser les données JSON

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



04 heures

# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 4 :

### Manipuler les objets : Création d'objet



#### Exercice 1 :

- Créez une classe appelée « Personne » qui accepte le nom d'une personne sous forme de chaîne et son âge sous forme de nombre.
- La classe « Personne » doit avoir une méthode appelée « Décire » qui renvoie une chaîne avec la syntaxe suivante : " nom, âge ans". Ainsi, par exemple, si la personne Ali a 19 ans alors la fonction « Décire » de son objet renverra « Ali, 19 ans".

#### Exercice 2 :

- Créez un objet appelé « Formateur » dérivé de la classe « Personne » et implémentez une méthode appelée « Former » qui reçoit une chaîne appelée « Sujet » et qui affiche :  
[Nom du formateur] enseigne [Sujet]

## Activité 4 :

### Manipuler les objets : Création d'objet



#### Exercice 3 :

1. Créez une autre classe appelée « Livre ». Chaque livre doit avoir les propriétés suivantes :
  - Titre, Genre, Auteur
  - Lu (vrai ou faux)
  - Date de lecture: peut être vide, sinon doit être un objet Date de JS.
2. Créez une classe appelée « ListeLivres » via le modèle d'appel du constructeur. Cette classe a les propriétés suivantes :
  - Nombre de livres marqués comme lus
  - Nombre de livres marqués pas encore lus
  - Livre suivant à lire (objet livre)
  - Livre en cours de lecture (objet livre)
  - Dernier livre lu (objet livre)
  - Un tableau de tous les livres.
3. Chaque liste de livres devrait avoir les méthodes suivantes :
  - ajouter(livre) qui ajoute un livre à la liste des livres.
  - terminerLivreEnCours()
    - Doit marquer le livre en cours de lecture comme lu
    - Modifiez la propriété dernierLivre en lui affectant le livre lu
    - Modifiez la propriété livreEnCours pour pointer vers le contenu de la propriété livreSuivant ;
    - Modifiez la propriété livreSuivant pour qu'elle soit le premier livre non lu de la liste des livres.
4. Proposez d'autres méthodes pour compléter les traitements au niveau des classes listeLivres.

## Activité 4 :

### Manipuler les objets : Création d'objet



#### Solution (Exercice 1) :

```
var Personne = function(nom, age)
{
    this.nom=nom;
    this.age=age;
    this.decrire = function()
    {
        return this.nom + ", " + this.age + " ans";
    }
}
var obj1 = new Personne("Ali", 25);
var obj2 = new Personne("Nadia", 24);
console.log(obj1.decrire());
console.log(obj2.decrire());
```

## Activité 4 :

### Manipuler les objets : Création d'objet



### Solution (Exercice 2) :

```
var Personne = function() {};  
Personne.prototype.initialize = function(nom, age)  
{  
    this.nom = nom;  
    this.age = age;  
}  
  
var Formateur = function()  
{  
    this.former = function(sujet)  
    {  
        console.log(this.nom + " enseigne " + sujet);  
    }  
}  
  
Formateur.prototype = new Personne();  
var obj = new Formateur();  
obj.initialize("Ahmed", 45);  
obj.former("JavaScript");
```

## Activité 4 :

### Manipuler les objets : Création d'objet



### Solution (Exercice 3) :

```
var livre = function(titre, genre, auteur, lu, dateLecture)
{
  this.titre = titre || "Sans titre";
  this.genre = genre || "Fiction";
  this.auteur = auteur || "Sans auteur";
  this.lu = lu || false;
  this.dateLecture = new Date(dateLecture);
}
var ListeLivres = function(livres)
{
  if (!typeof livres == "array")
  {
    livres = false;
  }
  this.tousLivres = livres || [];
  //vérifier s'il y a des livres à traiter
  if (this.tousLivres.length > 0)
  {
    //Parcourir la liste de tous les livres de l'argument « livres »
    //Utiliser la fonction "ajouter" pour ajouter un livre à la liste
  }
}
```

## Activité 4 :

### Manipuler les objets : Création d'objet



### Solution (Exercice 3) :

```
this.ajouter = function(livre)
{
    this.tousLivres.push(livre);
}
this.terminerLivreEnCours = function()
{
    if (this.livreEnCours)
    {
        this.livreEnCours.lu = true;
        this.dernierLivre = this.livreEnCours
        this.livreEnCours = this.livreSuivant
    }
    else
    {
        console.log("Il faut se mettre à la lecture")
    }
}
}
```





## PARTIE 3

### Manipuler les éléments d'une page avec dom

Dans ce module, vous allez :

- Comprendre l'arbre DOM, les nœuds parents et enfants
- Connaître les bases de la manipulation du DOM en JavaScript
- Manipuler les éléments HTML



30 heures



## Activité 1

### Comprendre l'arbre dom, les nœuds parents et enfants

#### Compétences visées :

- Comprendre l'arbre DOM et l'objet Document
- Manipuler la navigation dans le DOM (parentNode, childNodes, ... )

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



08 heures

# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

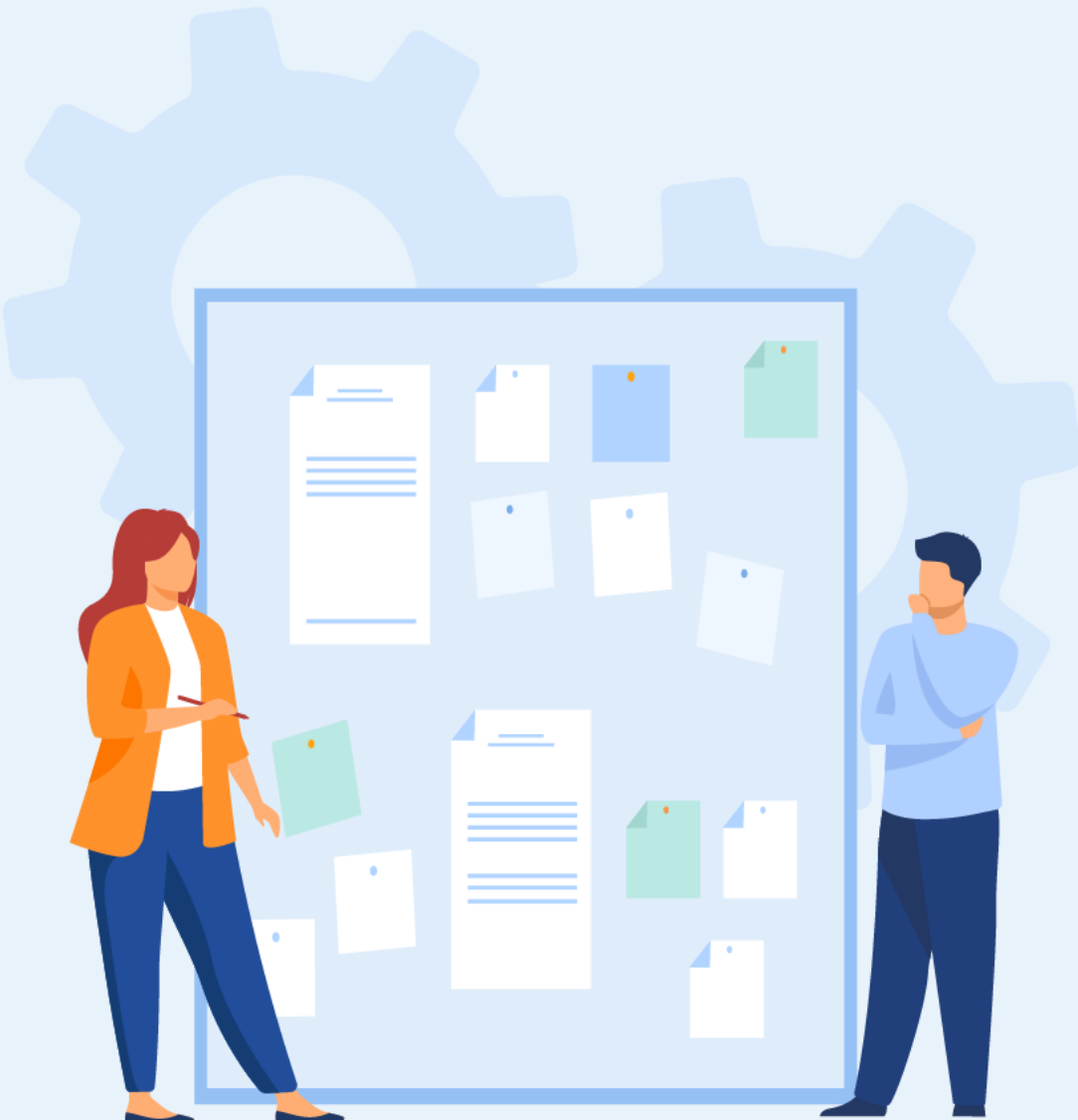
- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 1 : Comprendre l'arbre dom, les nœuds parents et enfants



### Exercice 1 :

Créer l'application suivante en respectant les consignes ci-dessous :

Casablanca

Rabat

Tétouan

Agadir

Marrakech

- Le clic sur le bouton « Ajouter » permet d'insérer la ville saisie dans la zone de texte à la fin de la liste.
- Le clic sur une ville provoque le changement de sa couleur afin de montrer l'état de sélection (On peut sélectionner plusieurs villes)
- Le clic sur le bouton « Supprimer » permet de supprimer les villes sélectionnées.

## Activité 1 :

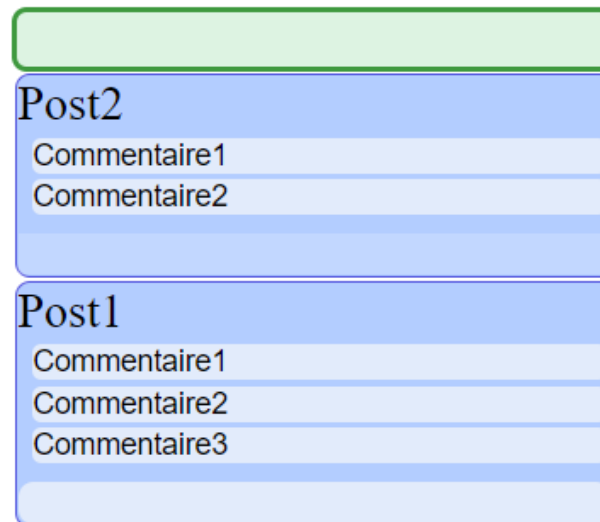
### Comprendre l'arbre dom, les nœuds parents et enfants



### Exercice 2 :

L'objectif est de créer un comportement similaire à celui de Facebook :

- On ajoute les posts sur le champ de saisie le plus haut.
- Pour chaque post on ajoute des commentaires à la fin.



## Activité 1 :

### Comprendre l'arbre dom, les nœuds parents et enfants



#### Solution (Exercice 1) :

```
function ajouterVille(){
    var liste = document.querySelector("#liste");
    var li = document.createElement('li');
    li.className="unselected";
    li.innerHTML=document.querySelector("#ville").value;
    li.addEventListener("click",changerCouleur);
    liste.appendChild(li);
}

function changerCouleur()
{
    const c=event.target.className;
    if (c=="unselected")
    {
        event.target.className="selected";
    }
    else
    {
        event.target.className="unselected";
    }
}
```

```
<body>
  <input type="text" id="ville">
  <button onclick="ajouterVille()">Ajouter</button>
  <button onclick="supprimerVille()">Supprimer</button>
  <ul id="liste"></ul>
</body>
```

```
function supprimerVille()
{
    const listeElements=document.querySelectorAll("#liste > li");
    const liste=document.querySelector("#liste");
    //Expression lambda
    listeElements.forEach(e=>{
        if (e.className=="selected"){
            liste.removeChild(e);
        }
    });
}
```

## Activité 1 :

### Comprendre l'arbre dom, les nœuds parents et enfants



### Solution (Exercice 2) :

```
<body>
  <input type="text" id="post" onkeypress="ajouterPost(event,value,this)">
  <div id="contenuPost"></div>
</body>
```

```
function ajouterCommentaire(ev,val)
{
  if (ev.keyCode==13)
  {
    var element=document.activeElement.parentElement;
    var texte=document.createTextNode(val);
    var comment=document.createElement("p");
    comment.setAttribute("class","commentElement");
    comment.appendChild(texte);
    element.insertBefore(comment,document.activeElement);
    document.activeElement.value="";
  }
}
```

## Activité 1 :

### Comprendre l'arbre dom, les nœuds parents et enfants



### Solution (Exercice 2) :

```
function ajouterPost(e,v)
{
    if (e.keyCode==13)
    {
        var contenuPost=document.getElementById("contenuPost");
        var post=document.createElement("div");
        post.setAttribute("class","postElement");
        var texte=document.createTextNode(v);
        var comment=document.createElement("p");
        comment.setAttribute("class","comment");
        comment.appendChild(texte)
        var inputPost=document.createElement("input");
        inputPost.setAttribute("class","inputComment")
        inputPost.setAttribute("type","text");
        inputPost.setAttribute("onkeypress","ajouterCommentaire(event,value)");
        post.appendChild(comment);
        post.appendChild(inputPost);
        contenuPost.insertBefore(post,contenuPost.firstChild);
        document.getElementById("post").value="";
    }
}
```



## Activité 1 : Comprendre l'arbre dom, les nœuds parents et enfants



### Exercice de synthèse :

Créer l'application suivante en respectant les consignes :

- Quand l'utilisateur clique sur le bouton « Ajouter », on affiche le formulaire d'ajout ;
- Quand on clique sur le bouton « Ajouter », les éléments saisis seront ajoutés à la fin du tableau ;
- Quand on clique sur « Annuler », le formulaire d'ajout disparaît ;
- Quand on sélectionne une ligne, elle est coloriée en vert ;
- Quand on clique sur « Supprimer », la ligne entière sera supprimée.

#### Liste des étudiants

Code	Nom	Prénom	Semestre	Filière
3	NACIRI	Hamza	S6	Math
2	SAFIR	Meriem	S2	Info
Ajouter		Supprimer		

2	
SAFIR	
Meriem	
S2	
Info	
Ajouter	Annuler

#### Liste des étudiants

Code	Nom	Prénom	Semestre	Filière
3	NACIRI	Hamza	S6	Math
2	SAFIR	Meriem	S2	Info
Supprimer				

## PARTIE 4

### Gérer les événements utilisateur

Dans ce module, vous allez :

- Comprendre la notion d'événement pour gérer l'interactivité
- Gérer les éléments d'un formulaire



**20 heures**



# Activité 1

## Comprendre la notion d'événement pour gérer l'interactivité

### Compétences visées :

- Utiliser la méthode `addEventListener`
- Manipuler les événements `MouseEvents` et l'interaction avec le clavier

### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



**18 heures**

# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 1 :

### Comprendre la notion d'événement pour gérer l'interactivité



### Exercice 1 :

Créer la page Web correspondante au code suivant :

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <h1>Histoires</h1>
    <ul>
      <li>Nom : <input type="text" id="nom">
      <li>Adjectif: <input type="text" id="adj">
      <li>Nom d'une personne: <input type="text" id="pers">
    </ul>
    <button id="raconter">Raconte une histoire</button>
    <div id="histoire"></div>
  </body>
</html>
```

1. Ajoutez une balise de script en bas de la page pour votre code.
2. Ajoutez un écouteur d'événement au bouton afin qu'il appelle une fonction « raconterHistoire » lorsqu'il est cliqué.
3. Dans la fonction « raconterHistoire », récupérez les valeurs des éléments d'entrée du formulaire, puis créez une histoire et affichez-la dans la div correspondante.

## Activité 1 :

### Comprendre la notion d'événement pour gérer l'interactivité



#### Solution (Exercice 1) :

```
<script>
function raconterHistoire()
{
    var histoireDiv = document.getElementById("histoire");
    var personne = document.getElementById("pers").value;
    var adjectif = document.getElementById("adj").value;
    var nom = document.getElementById("nom").value;
    histoireDiv.innerHTML = personne + " mange " + adjectif + " " + nom + "... bizarre !";
}
var btn = document.getElementById('raconter');
btn.addEventListener('click', raconterHistoire);
</script>
```

## Activité 1 :

### Comprendre la notion d'événement pour gérer l'interactivité



#### Exercice 2 :

- Ajoutez un élément d'ancrage à une page HTML qui renvoie vers un site Web de votre choix. Lorsque vous cliquez sur l'ancre, vous devez empêcher le chargement de la nouvelle page Web.

#### Exercice 3 :

- Ajoutez un écouteur d'événements pour les événements du clavier à l'élément html d'une page Web. Lorsqu'une combinaison spécifique de touches est enfoncée (par exemple, a, b, c, d), un message s'affiche pour indiquer que la combinaison a été saisie correctement.

#### Exercice 4 :

- Créer un programme Javascript qui permet d'afficher un menu contextuel suite au clic droit de la souris dans un objet DIV et le masque quand le curseur sort de la zone DIV.
- Améliorer le programme pour qu'il affiche la position du curseur (x et y) sur un objet label.

## Activité 1 :

### Comprendre la notion d'événement pour gérer l'interactivité



### Solution (Exercice 2) :

```
<!DOCTYPE html>
<html>
  <head>
    <title></title>
  </head>
  <body>
    <a href="http://www.ofppt.ma">Site de l'OFPPT</a>
    <script>
      var lien = document.querySelector('a');
      lien.addEventListener('click', function(event)
      {
        event.preventDefault();
        console.log("Redirection interdite");
      })
    </script>
  </body>
</html>
```



## Activité 1 :

### Comprendre la notion d'événement pour gérer l'interactivité



#### Solution (Exercice 3) :

```
<p id="affichage"></p>
<script>
  addEventListener('keydown', keyPressed);
  addEventListener('keyup', keyReleased);
  var keys =[];
  function keyPressed(event)
  {
    var msg = document.getElementById('affichage');
    var texte = document.createTextNode('Vous avez cliqué sur les touches a et b');
    keys[event.keyCode] = true;
    if (keys[65] && keys[66] && msg.textContent === '') //touches 'a' et 'b'
    {
      msg.append(texte);
    }
  }
  function keyReleased(event)
  {
    var msg = document.getElementById('affichage');
    keys[event.keyCode] = false;
    msg.innerHTML= '';
  }
</script>
```

## Activité 1 :

### Comprendre la notion d'événement pour gérer l'interactivité



#### Solution (Exercice 4) :

```
<div id="myDiv">Element DIV</div>
  <ul id="menu">
    <li>Choix1</li>
    <li>Choix2</li>
    <li>Choix3</li>
  </ul>
```

```
<script>
  window.addEventListener("load",()=>{
    document.getElementById("myDiv").addEventListener("contextmenu",(event)=>{
      event.preventDefault()
      document.getElementById("menu").style.display="inline-block"
    })
    document.getElementById("myDiv").addEventListener("mouseleave",()=>{
      document.getElementById("menu").style.display="none"
    })
  })
</script>
```

## Activité 1 :

### Comprendre la notion d'événement pour gérer l'interactivité



#### Exercice 5 :

1. Ajoutez un écouteur d'événement pour les événements de clic sur un élément d'image.
2. Ajoutez un bouton à la page qui permet de supprimer l'écouteur d'événement sur l'image.
3. Ajoutez un écouteur d'événement pour l'événement DOMContentLoaded afin d'afficher l'heure actuelle sur une page Web lorsque la page est chargée.
4. Ajoutez un écouteur d'événement « blur » à un élément d'entrée (input) qui vérifie que la valeur entrée dans l'élément est une chaîne composée de 9 caractères au moins. Ajoutez une bordure rouge à l'élément si ce n'est pas le cas.
5. Ajoutez un écouteur d'événement qui gère les événements tactiles mais qui n'est plus invoqué pour les événements de clic.

## Activité 1 :

### Comprendre la notion d'événement pour gérer l'interactivité



#### Solution (Exercice 5) :

```
img id="" src="oiseau.jpg" width="200" height="200"/>
<button>Supprimer evenement</button>
<input type="text" id="input1"/>
<p id="element"></p>
<script>
    var img = document.querySelector('img');
    var btn = document.querySelector('button');
    var input1 = document.getElementById('input1');
    img.addEventListener('click', EntrerImage);
    btn.addEventListener('click', supprimerEvenImg);
    document.addEventListener('DOMContentLoaded', afficherHeure);

    input1.addEventListener('blur', function(e) {
        var input = e.target;
        var input1 = document.getElementById('input1');
        var inputStyle = input1.style;
        if(input.value.length < 9) {
            inputStyle.border = '1px solid red';
            console.log(input.value);
        }
        else { inputStyle.border = '1px solid green'; }
    });
</script>
```

## Activité 1 :

### Comprendre la notion d'événement pour gérer l'interactivité



#### Solution (Exercice 5) :

```
document.addEventListener('touchstart', function(e) {
    console.log('touche');
})
document.addEventListener('touchstart', function(e) {
    if(e.sourceCapabilities.fireTouchEvents) {
        return false;
    }
    console.log('clic');
})
function afficherHeure(e) {
    var p = document.getElementById('element');
    p.textContent = event.type + " " + new Date().toLocaleTimeString() ;
}
function EntrerImage(e) {
    var img = e.target;
    console.log(img);
}
function supprimerEvenImg(e) {
    img.removeEventListener('click', EntrerImage);
}

</script>
```

## Activité 2

### Gérer les éléments d'un formulaire

#### Compétences visées :

- Maîtriser la validation d'un formulaire
- Manipuler l'Interruption et la soumission d'un formulaire

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



**02 heures**



# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 2 :

### Comprendre la notion d'événement pour gérer l'interactivité



#### Exercice 1 :

- Soit le formulaire suivant :

```
<body>
  <h1 style="text-align: center;">Formulaire d'inscription</h1>
  <form name="formInscp" action="" onsubmit="return validation()" method="post">
    <p>Nom: <input type="text" size="65" name="nom" /></p>
    <p>Adresse: <input type="text" size="65" name="adresse" /></p>
    <p>Adresse E-mail : <input type="email" size="65" name="email" /></p>
    <p>Mot de passe : <input type="password" size="65" name="motPasse" /></p>
    <p>Telephone: <input type="text" size="65" name="telephone" /></p>
    <p>Langue :
      <select type="text" value="" name="langue">
        <option>Anglais</option>
        <option>Français</option>
        <option>Arabe</option>
      </select>
    </p>
    <p>Commentaire: <textarea cols="55" name="commentaire"> </textarea></p>
    <p><input type="submit" value="Valider" name="Valider" />
    <input type="reset" value="Annuler" name="Annuler" /></p>
  </form>
</body>
```

- Créer le code javascript de validation pour ce formulaire sachant que tous les champs sont obligatoires.



## Activité 2 :

### Comprendre la notion d'événement pour gérer l'interactivité



#### Solution (Exercice 1) :

```
function validation() {  
    var nom = document.forms["formInscp"]["nom"];  
    var email = document.forms["formInscp"]["email"];  
    var tel = document.forms["formInscp"]["telephone"];  
    var lang = document.forms["formInscp"]["langue"];  
    var pass = document.forms["formInscp"]["motPasse"];  
    var adr = document.forms["formInscp"]["adresse"];  
    if (nom.value == "") {  
        window.alert("Merci de saisir le nom.");  
        nom.focus();  
        return false;  
    }  
    if (adr.value == "") {  
        window.alert("Merci de saisir l'adresse.");  
        adr.focus();  
        return false;  
    }  
    if (email.value == "") {  
        window.alert("Merci de saisir une adresse mail.");  
        email.focus();  
    }  
}
```

```
        return false;  
    }  
    if (pass.value == "") {  
        window.alert("Merci de saisir le mot de passe");  
        pass.focus();  
        return false;  
    }  
    if (tel.value == "") {  
        window.alert("Merci de saisir le numéro de  
téléphone.");  
        tel.focus();  
        return false;  
    }  
    if (lang.selectedIndex < 1) {  
        alert("Merci de choisir une langue.");  
        lang.focus();  
        return false;  
    }  
    return true;  
}
```

## PARTIE 5

### Manipuler jquery

Dans ce module, vous allez :

- Découvrir jQuery
- Découvrir AJAX



10 heures

# Activité 1

## Découvrir jquery

### Compétences visées :

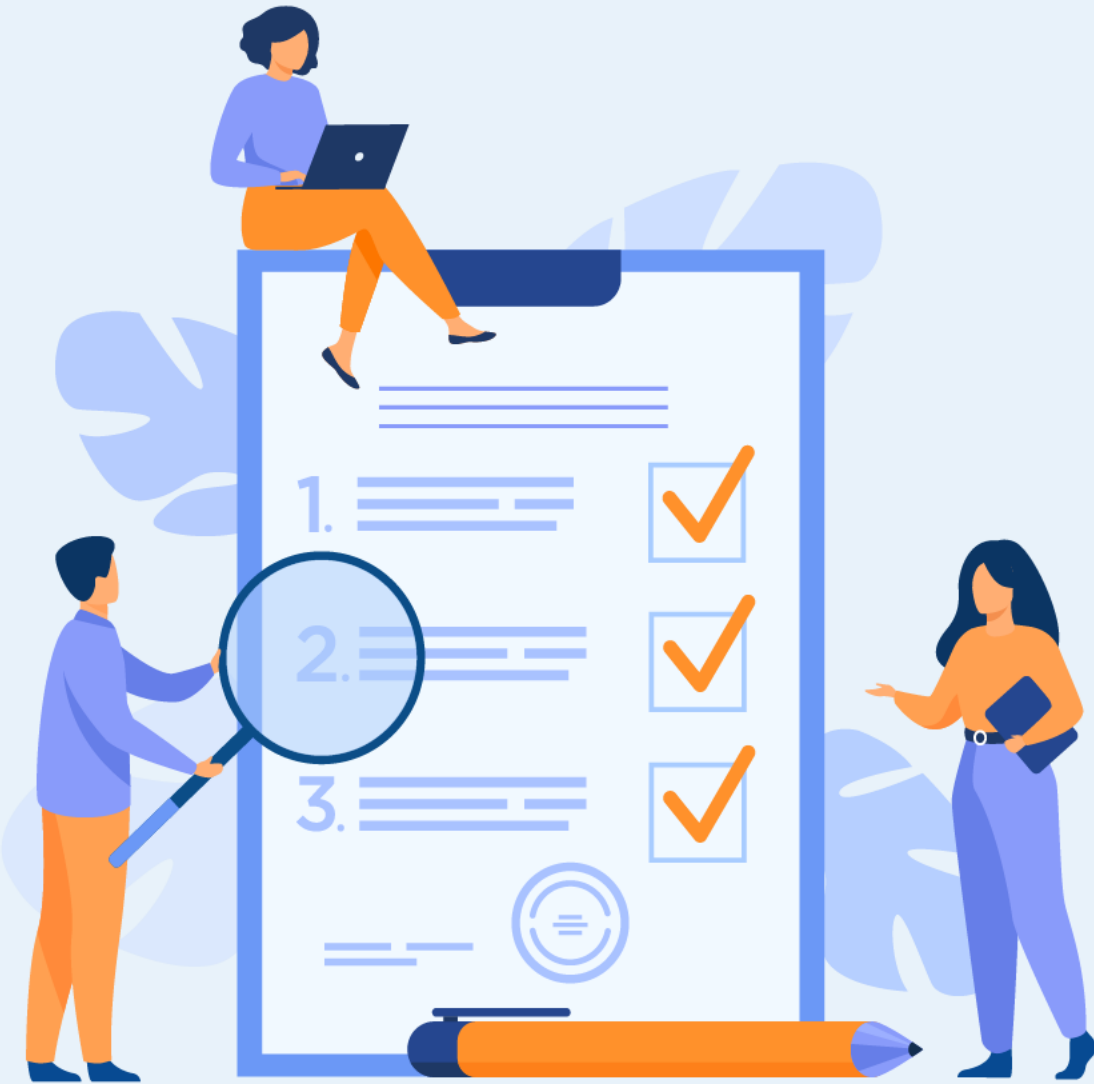
- Manipuler les fonctions essentielles et chaînage
- Maîtriser l'association d'évènements et déclenchement
- Intégrer et utiliser les plugins existants

### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



08 heures



# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 1 : Découvrir jquery



### Exercice 1 :

Programmez les événements suivants :

1. En cliquant sur un bouton, affichez un message de bienvenue.
2. En cliquant dans une zone de texte, sa couleur de fond passe au jaune.
3. Soit un élément html (paragraphe, titre, ...). Supprimez et ajoutez un style différent à cet élément en utilisant JQuery. Pour cela, créez deux classes de style et appliquez par défaut une première à un élément. En cliquant sur un bouton, changez la propriété associée des éléments à un autre style.
4. Basculer les propriétés de style de d'un élément suite à un événement de clic.
5. Créez un bouton et une zone de texte. En cliquant sur le bouton, le texte écrit dans la zone de texte doit s'afficher sur le bouton.

## Activité 1 : Découvrir jquery



### Solution (Exercice 1) :

1.

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
...
<button type='button' id=boutton>Cliquez ici</button>
<span id=affichage></span>
```

```
<script>
    $(document).ready(function() {
        $("#boutton").click(function(){
            $("#affichage").html('Bienvenue au cours de JQuery');
        })
    })
</script>
```

## Activité 1 : Découvrir jquery



### Solution (Exercice 1) :

2.

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
...
<input type='text' class='t_input' />
```

```
<script>
  $(document).ready(function() {
    $(".t_input").click(function(){
      $(this).css("background-color", "yellow");
    })
  })
</script>
```

### Solution (Exercice 1) :

3.

```
<html>
  <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
    <style>
      .c1{          background-color:yellow;          }
      .c2{          background-color:aqua;             }
    </style>
  </head>
  <body>
    <p class=c1>cliquez pour changer la couleur</p>

    <script>
      $(document).ready(function()
      {
        $(".c1").click(function(){
          $(''.c1').removeClass('c1').addClass('c2');
        })
      })
    </script>
  </body>
</html>
```



### Solution (Exercice 1) :

4.

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
<style>
  .c1{    background-color:yellow;    }
  .c2{    background-color:aqua;      }
</style>
```

```
<p id="zone">C'est l'élément identifié par "zone". cliquez pour changer sa couleur</p>
<script>
  $(document).ready(function() {
    $('#zone').addClass('c1')
    $('#zone').click(function(){
      var classeActuelle = $('#zone').attr('class');
      if(classeActuelle=='c1'){
        nouvelleClasse='c2'
      }else{
        nouvelleClasse='c1'
      }
      $('#zone').removeClass(classeActuelle).addClass(nouvelleClasse);
    })
  })
</script>
```

## Activité 1 : Découvrir jquery



### Solution (Exercice 1) :

5.

```
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</head>

<body>
  <input type='text' id='t1' />
  <button type='button' id='b1'>Cliquez ici</button>
  <script>
    $(document).ready(function() {
      $("#b1").click(function(){
        var texte=$('#t1').val();
        $('#b1').html(texte);
      })
    })
  </script>
</body>
```

## Activité 1 : Découvrir jquery



### Exercice 2 :

1. Créez une balise div et qui affiche un message de bienvenue (en utilisant JQuery).
2. Créez deux balises div et avec couleur d'arrière-plan de votre choix à l'aide de JQuery.
3. Afficher et masquer le message affiché dans une balise div en cliquant sur deux boutons. Utilisez les fonctions JQuery show() et hide().
4. Affichez et masquez le message affiché dans une balise div à l'aide d'un seul bouton. Utilisez la fonction JQuery toggle().

## Activité 1 : Découvrir jquery



**WEBFORCE**  
BE THE CHANGE

### Solution (Exercice 2) :

1.

```
<div id=affichage></div>

<script>
  $(document).ready(function() {
    $("#affichage").html('Bienvenue à l\'ofppt');
  })
</script>
```

### Solution (Exercice 2) :

2.

```
<div id='affichage1'> 1er div</div>
<div id='affichage2'> 2ème div</div>

<script>
$(document).ready(function() {
    $("#affichage1,#affichage2").css("background-color", "yellow");
})
</script>
```

### Solution (Exercice 2) :

3.

```
<button type='button' id='b1'>Show</button>
<button type='button' id='b2'>Hide</button>
<div id="affichage"> Bienvenue à l'ofppt </div>

<script>
  $(document).ready(function() {
    $("#b1").click(function(){
      $("#affichage").show();
    })
    $("#b2").click(function(){
      $("#affichage").hide();
    })
  })
</script>
```

### Solution (Exercice 2) :

4.

```
<button type='button' id='b3'>Show & Hide</button>
<div id="affichage"> Bienvenue à l'ofppt </div>

<script>
  $(document).ready(function() {
    $("#b3").click(function(){
      $("#affichage").toggle();
    })
  })
</script>
```

## Activité 1 : Découvrir jquery



### Exercice 3 :

1. Créez une classe CSS et nommez-la « erreur » , donnez sa couleur de bordure en rouge et sa couleur d'arrière-plan en jaune. Appliquez cette classe à la zone de texte lorsque le focus est déplacé hors de celle-ci et si les données saisies sont supérieures à 10.
2. Créez deux zones de texte et un bouton avec la valeur « Addition ». En cliquant sur le bouton, il faut afficher la somme des deux nombres entrés dans les zones de texte. De même, ajoutez les boutons « Soustraction », « Multiplication » et « Division ».
3. Vos boutons dans le code ci-dessus resteront désactivés jusqu'à ce que les deux entrées soient remplies. (Utilisez une fonction pour activer ou désactiver les boutons. Déclenchez cette fonction avec l'événement « Blur » des zones de texte)



## Activité 1 : Découvrir jquery



### Solution (Exercice 3) :

1.

```
<style>
    .erreur
    {
        border : 1px solid red;
        background-color: yellow;
    }
</style>
```

```
<input type='text' id='t1' />

<script>
    $(document).ready(function() {
        $('#t1').blur(function(){
            var my_val=parseInt($('#t1').val());
            if(my_val >10 ){
                $('#t1').addClass('erreur');
            }
        })
    })
</script>
```

## Activité 1 : Découvrir jquery



**WEBFORCE**  
BE THE CHANGE

### Solution (Exercice 3) :

2.

```
<input type='text' id='t1'/>
<input type='text' id='t2'/>
<br>
<button type='button' id='btnAdd'>+</button>
<button type='button' id='btnSous'>-</button>
<button type='button' id='btnMult'>*</button>
<button type='button' id='btnDiv'>/</button>
<br>
<input type='text' id='t3'/>
```

### Solution (Exercice 3) :

2.

```
<script>
    $(document).ready(function() {
        $('#btnAdd').click(function(){
            var val1=parseInt($('#t1').val());
            var val2=parseInt($('#t2').val());
            $('#t3').val(val1 + val2);
        })
        $('#btnSous').click(function(){
            var val1=parseInt($('#t1').val());
            var val2=parseInt($('#t2').val());
            $('#t3').val(val1 - val2);
        })
        $('#btnMult').click(function(){
            var val1=parseInt($('#t1').val());
            var val2=parseInt($('#t2').val());
            $('#t3').val(val1 * val2);
        })
        $('#btnDiv').click(function(){
            var val1=parseInt($('#t1').val());
            var val2=parseInt($('#t2').val());
            $('#t3').val(val1 / val2);
        })
    })
</script>
```

### Solution (Exercice 3) :

3.

```
<script>
    $(document).ready(function() {
        $("#btnAdd,#btnSous,#btnMult,#btnDiv").prop('disabled',true);

        valider=function valider(){
            var val1=$('#t1').val();
            var val2=$('#t2').val();
            if(val1.length >0 && val2.length >0){
                $("#btnAdd,#btnSous,#btnMult,#btnDiv").prop('disabled',false);
            }else{
                $("#btnAdd,#btnSous,#btnMult,#btnDiv").prop('disabled',true);
            }
        }

        $('#t1,#t2').blur(function(){
            valider()
        })
    })
</script>
```

### Solution (Exercice 3) :

3.

```
$('#btnAdd').click(function(){  
    var val1=parseInt($('#t1').val());  
    var val2=parseInt($('#t2').val());  
    $('#t3').val(val1 + val2);  
})  
$('#btnSous').click(function(){  
    var val1=parseInt($('#t1').val());  
    var val2=parseInt($('#t2').val());  
    $('#t3').val(val1 - val2)  
})  
$('#btnMult').click(function(){  
    var val1=parseInt($('#t1').val());  
    var val2=parseInt($('#t2').val());  
    $('#t3').val(val1 * val2)  
})  
$('#btnDiv').click(function(){  
    var val1=parseInt($('#t1').val());  
    var val2=parseInt($('#t2').val());  
    $('#t3').val(val1 / val2)  
})  
})  
</script>
```

## Activité 1 : Découvrir jquery



### Exercice 4 :

Réaliser un formulaire d'ajout des notes qui contient les champs suivants :

- nom (contient au moins 3 caractères)
- score (entre 0 et 20)
- Bouton « Ajouter » pour saisir les données dans la table.
  1. Ecrire le code Html et css qui produit le formulaire
  2. En utilisant jquery :
    - a) Ecrire le code du bouton « Ajouter »
    - b) Ecrire le code qui calcule la moyenne.

**Joueurs:**  
nom:  score:

Leila	17
Driss	18

moyenne:17.5

### Solution (Exercice 4) :

```
<table>
  <tr><td colspan="3">Joueurs:</td></tr>
  <tr>
    <td>
      <label for="">Nom:</label><input type="text" id="nom" required="required">
    </td>
    <td>
      <label for="">Score:</label><input type="text" id="score" required>
    </td>
    <td>
      <input type="submit" id="Aj" value="Ajouter">
    </td>
  </tr>
</table>
```

## Activité 1 : Découvrir jquery



**WEBFORCE**  
BE THE CHANGE

### Solution (Exercice 4) :

```
<div id="table">
  <table>
    <tr style="background: #9966ff;">
      <td>Leila</td><td>17</td>
    </tr>
    <tr style="background: #CCCCFF;">
      <td>Driss</td><td>18</td>
    </tr>
  </table>
</div>

<div>
  <p>Moyenne:<span id="moyenne"></span>
</p>
</div>
```



## Activité 1 : Découvrir jquery



### Solution (Exercice 4) :

```
<script>
$(document).ready(function()
{
    var T, r1,r2,res;
    r1 = Number($('#table tr:first-child() td:last-child()').text());
    r2 =Number($('#table tr:nth-child(2) td:last-child()').text());
    res=r1+r2;

    $('#moyenne').text(res/2);
    // bouton ajouter
    $('#Aj').click(function(e)
    {
        var nom = $('#nom').val();
        var score = $('#score').val();
        if(isNaN(nom) && score >= 0 && score <= 20 )
        {
            var row = ""+""+nom+""+""+score+""+"";

```

### Solution (Exercice 4) :

```
$("#table table").append(row);
res+=Number($('#table tr:last-child() td:last-child()').text());
$('#table table tr:nth-child(odd)').css('background','#CCCCFF');
$('#table table tr:nth-child(even)').css('background','#9966ff');
var len = $('#table tr').length;
$('#moyenne').text(res/len);
    }
else
    {
        alert("un champ invalide");
        e.preventDefault();
    }
});
});
</script>
```

## Activité 2

### Découvrir AJAX

#### Compétences visées :

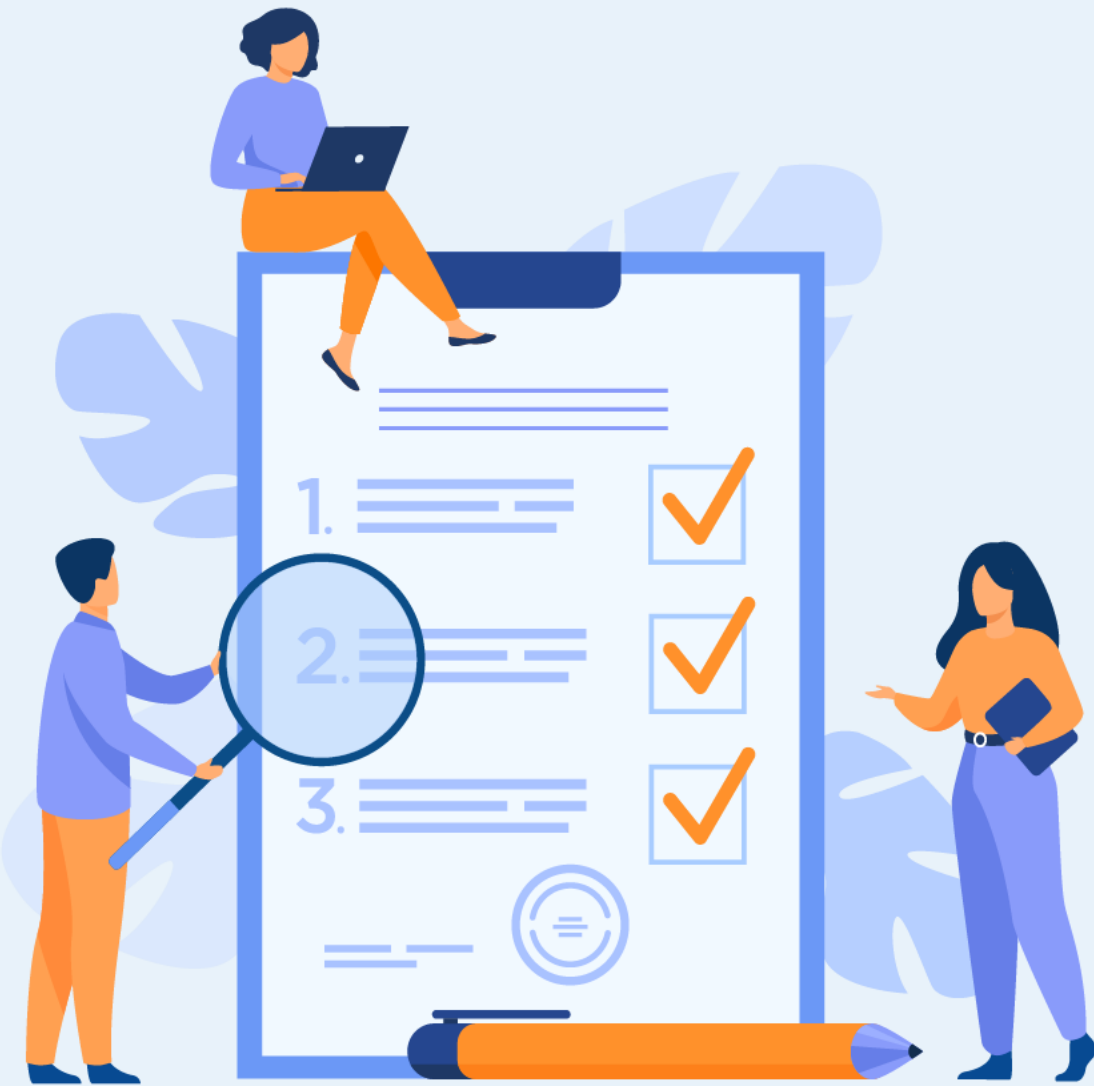
- Comprendre le fonctionnement d'AJAX
- Implémenter AJAX via jQuery

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



02 heures



# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 2 : Découvrir AJAX



### Exercice 1 :

- Créer un programme jquery qui récupère la personne de l'ID 5 depuis une API REST sur le lien: <https://jsonplaceholder.typicode.com/todos/1>
- Mettre les données de la personne récupérée dans une structure HTML de votre choix après avoir cliqué sur un bouton. Utilisez la méthode « GET » .

### Solution (Exercice 1) :

```
<button type='button' id='b1'>Cliquez ici</button>
<p id='userid'></p><p id='id'></p><p id='title'></p><p id='completed'></p>

<script>
    $(document).ready(function() {
        $("#b1").click(function(){
            id = 5;
            $.ajax({
                type: "GET",
                url: "https://jsonplaceholder.typicode.com/todos/"+id,
                success: function (response)
                {
                    console.log(response.completed)
                    $("#userid").html(response.userId)
                    $("#id").html(response.id)
                    $("#title").html(response.title)
                    $("#completed").html(response.completed+"")
                }
            });
        })
    })
</script>
```

# Activité 1

## Diaporama

### Compétences visées :

- Propriétés des images
- Fonctions de JS

### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



02 heures



# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?





## Activité 1 : Diaporama

### Exercice 1 :

- Créer la page html suivante et programmer les événements de clic sur les boutons « image précédente » et « image suivante ». L'idée est de déclarer un tableau contenant les noms des photos, par exemple :
- `var mesphotos = new Array('images/img1.jpg','images/img2.jpg','images/img3.jpg','images/img4.jpg','images/img5.jpg');`
- Puis programmer les fonctions : `suivant()` et `precedent()`

#### Diaporama manuel



image précédente image suivante

## Activité 1 : Diaporama



### Exercice 2 :

- Même idée que l'exercice précédent, utiliser les fonction Up() et Down() :

### Déplacement d'images



en haut

en bas

## Activité 1 : Diaporama



**WEBFORCE**  
BE THE CHANGE

### Exercice 3 :

- Dans cet exercice, utiliser les fonctions `setInterval` et `clearInterval`. Le bouton « stop » permet d'arrêter le diaporama, le bouton « On y retourne » permet de le démarrer.

### Diaporama automatique



stop! on y retourne!

## Activité 2

### Quiz en JS

#### Compétences visées :

- Fonctions JS

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



02 heures



# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 2 :

### Quiz en JS

- L'objectif de l'activité est mettre en place un QCM interactif en utilisant HTML / CSS / JS. Le thème, dont le choix est libre, doit contenir trois parties (ou plus). L'idée est de mettre 10 questions par partie, et proposer à l'utilisateur des réponses sous forme de choix (des boutons radio, des cases à cocher ou liste select).
- Pour valider le test, il faut répondre au moins à sept questions par catégorie, sinon le score ne sera pas calculé. A la fin, vous devez indiquer à l'utilisateur son score.
- Vous pouvez proposer à l'utilisateur d'afficher la réponse correcte.







## Activité 3

### Carte interactive

#### Compétences visées :

- Événement de la souris en JS

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



02 heures

# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



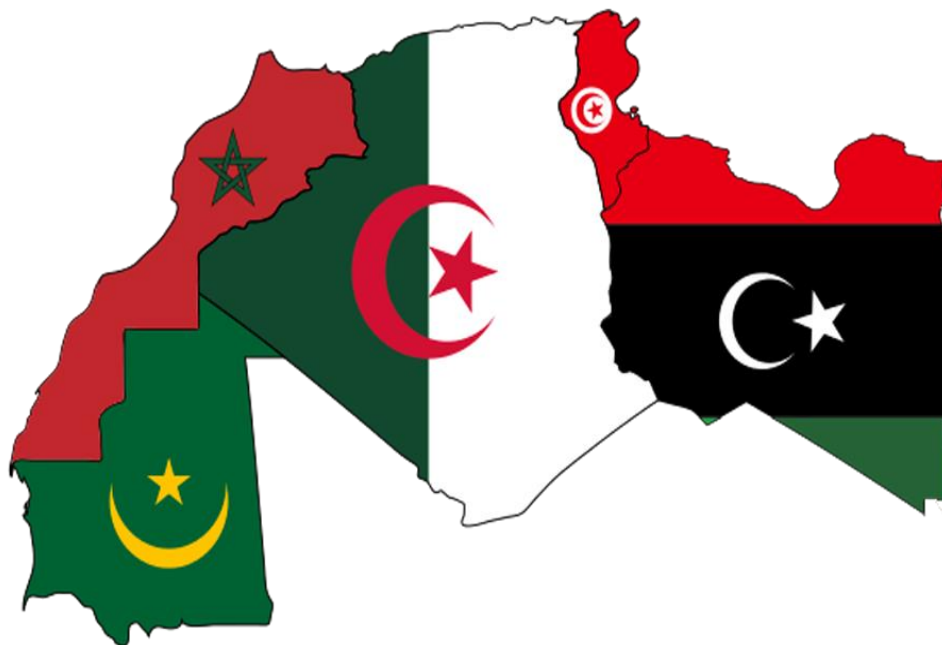


## Activité 3 : Carte interactive

Dans cet activité, nous allons découper la carte de du « Grand Maghreb » en zones correspondant chacune à un pays. Lorsque la souris passe sur un pays précis, les informations concernant cet état s'affichent dans des champs textes d'un formulaire.

### Étapes :

- Récupérez l'image 'maghreb.png'
- Découpez l'image en zones et récupérez les coordonnées de chaque zone.
- Utilisez les évènements « onMouseOver » et « onMouseOut ».



Pays :	<input type="text"/>
Population :	<input type="text"/> hab.
Superficie :	<input type="text"/> km <sup>2</sup>
Gouvernement :	<input type="text"/>
Capitale :	<input type="text"/>
Monnaie :	<input type="text"/>



## Activité 4

### Jeu « Memory »

#### Compétences visées :

- Développement des scripts JS et JQuery

#### Recommandations clés :

- Suivre les instructions du TP et organiser le dossier de travail
- Utiliser le résumé théorique pour réaliser le projet de synthèse



04 heures

# CONSIGNES

## 1. Pour le formateur :

- Demander aux apprenants de suivre les étapes décrites dans le résumé théorique du cours et d'appliquer les procédures
- Demander aux apprenants de réaliser le travail de synthèse

## 2. Pour l'apprenant :

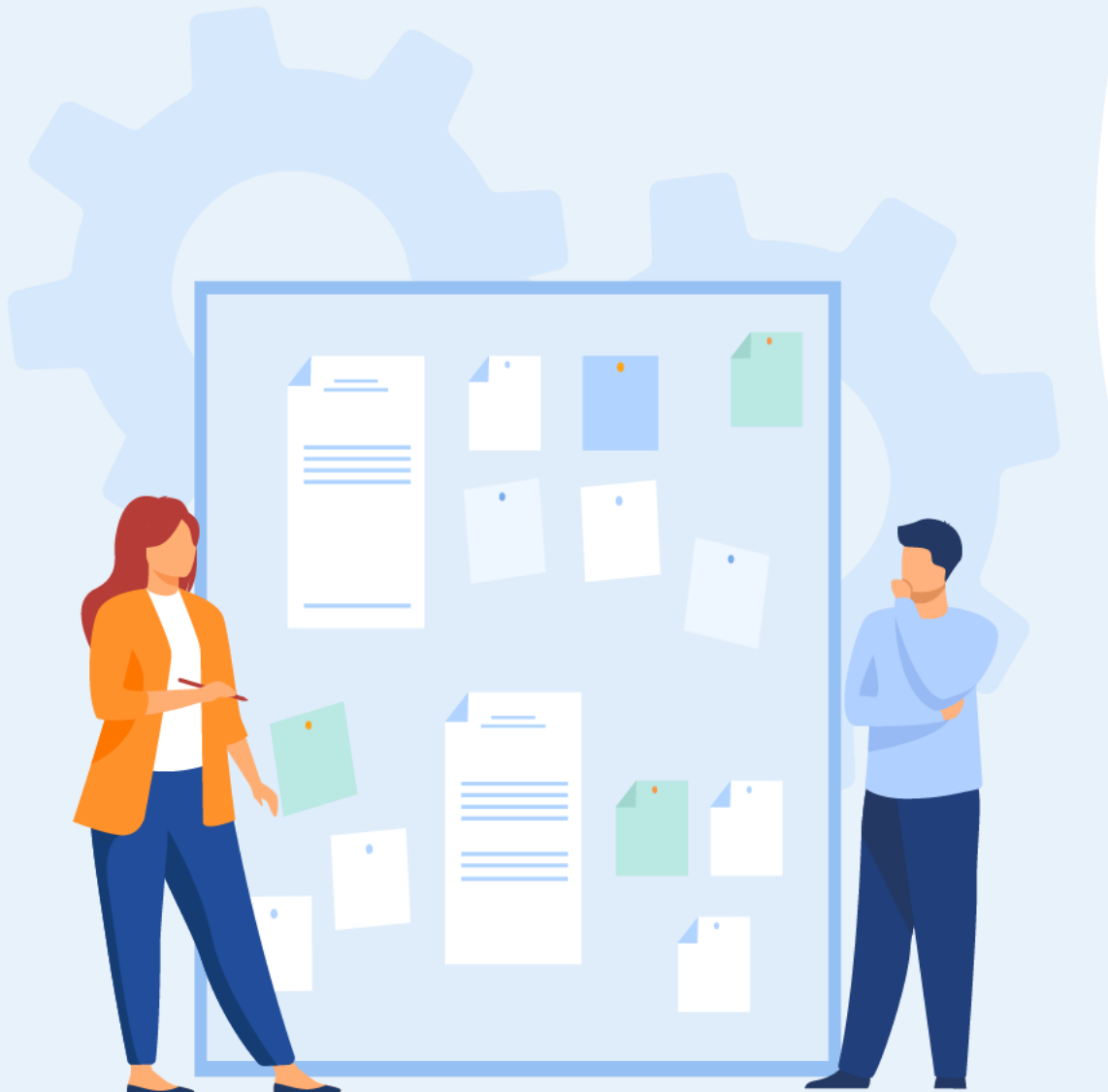
- Installer VS Code et y ajouter les extensions recommandées
- Créer un dossier de travail dans VS Code et y créer les fichiers des réalisations.

## 3. Conditions de réalisation :

- Support de résumé théorique accompagnant

## 4. Critères de réussite :

- Le stagiaire est-il capable de :
  - Présenter les bibliothèques JS les plus utilisées et discuter de leurs différences ?



## Activité 4 : Jeu « Memory »



### Version 1 :

Cette activité consiste à programmer le jeu de « memory ». Les règles du jeu sont les suivantes :

- Au début, toutes les cartes sont cachées
- Si les deux cartes sont identiques elles restent visibles, sinon elles disparaissent après 3 secondes
- La session du jeu se termine quand le joueurs réussit à découvrir toutes les paires des images

### Étapes à suivre :

- Récupérer huit images de même taille et coder en HTML/CSS la disposition des seize cartes (Grille de 4 lignes x 4 colonnes)
- Programmer en JavaScript le mélange des cartes. Par exemple en choisissant eux cartes au hasard et les échanger (modifier la propriété src de l'image) et en répétant cette action plusieurs fois. Ce mélange doit avoir lieu au chargement de la page
- Rendre les cartes invisibles en utilisant les propriétés CSS
- Programmer en JavaScript le dévoilement d'une image suite à un clic sur zone
- Suite au 2ème clic (sur une autre image), tester si les deux images dévoilées sont les mêmes. Si les images sont différentes, les faire disparaître après une 3 secondes
- Interdire le clic sur une troisième carte pendant que les deux cartes cliquées sont dévoilées
- Compter le nombre de paires découvertes, compter les nombre de clics utilisés

## Activité 4 : Jeu « Memory »



### Version 2 :

#### Programmer une deuxième version du jeu :

- Ajouter deux boutons : l'un qui permet de commencer une nouvelle partie, l'autre qui donne la solution en découvrant toutes les cartes
- Prévoir des niveaux du jeu : facile, difficile moyen
- Donner la possibilité du help
- Afficher le score obtenu
- Produire un effet visuel quand la carte apparaît
- Compléter en produisant des messages : découverte d'une paire, nombre de clics utilisés, message de victoire, etc.