

TD/TP3 – Introduction au JavaScript

A. Dynamisme avec JavaScript (TD de 2h)

Résultat d'apprentissage visé : En répondant à chaque question, les élèves doivent comprendre de formuler des fonctions JavaScript en fonction des spécifications requises pour rendre la page web plus dynamique

Cet exercice a pour but d'introduire la notion de dynamisme dans la page web. En cliquant sur un bouton, un écouteur est invoqué, transférant le contrôle à un fonction JavaScript qui fait une action nécessaire.

1. En utilisant HTML (et CSS, si nécessaire), créer un formulaire comme indiqué ci-dessous :

Calcul de base

1er Numéro:

2e Numéro:

Le résultat est :

Calcul de base

1er Numéro:

2e Numéro:

Le résultat est :
200

- Vous allez entrer deux numéros
- En cliquant, par exemple, sur le bouton « Multiplier », la page calcule la multiplication de ces deux numéros et la page affiche le résultat. Le même principe s'applique pour les autres boutons « Ajouter », « Soustraire » et « Diviser ». Pour cet exercice, vous devez écrire une fonction associée au dynamisme de chaque bouton.
- Ecrivez votre code JavaScript à l'intérieure de votre page HTML. Comment écrire le même code à l'extérieure de votre page HTML ?

Étapes à suivre :

- Dans la partie `<body></body>` de votre page web, créer un formulaire en utilisant `<form></form>`
- Ajouter `<table></table>`, `<tr></tr>` et `<td></td>` pour créer un tableau avec des lignes et de colonnes.
- Dans la première ligne de votre tableau, dans la première colonne, tapez « 1er Numéro ». Dans la seconde colonne, utiliser `<input type="number" id="firstNumber">` pour saisir le premier numéro dans votre formulaire et le sauvegarder dans une variable nommée "firstNumber".
- Répétez le même processus pour saisir le second numéro. Vous pouvez, cette fois-ci, sauvegarder le numéro saisi dans une variable nommée "secondNumber".
- Dans votre formulaire, ajouter le bouton « Ajouter » en utilisant `<input type="submit" onclick="addition()" value="Ajouter">`. Notez bien que "addition()" est une fonction qu'on va appeler quand on clique sur ce bouton ; on va définir cette fonction dans la partie JavaScript de notre code plus tard.
- Répétez le même processus pour créer des boutons "Soustraire", "Multiplier", et "Diviser". Pour chaque opération, il faut définir une fonction appropriée. Par exemple, vous pouvez associer la fonction "multiplication()" pour le bouton "Multiplier".
- Après `<form></form>`, ajouter un paragraphe. Dans le paragraphe, ajouter le texte Le résultat est
 . Quelle est l'intérêt d'utiliser la balise ?
- Tester votre code HTML. Votre page web doit afficher le même résultat que notre page d'exercice.
- Pour la partie JavaScript, dans la partie `<head></head>` de votre page web, ajouter `<script type="text/javascript"></script>`. Vous êtes maintenant prêt de coder en JavaScript.
- On va implémenter le code de notre fonction multiplication(). Dans la partie `<script></script>`, ajouter `function multiplication()`. Entre { et }, on va ajouter des codes pour récupérer le premier et second numéro, faire la multiplication et afficher le résultat de l'opération.
- On va déclarer d'abord deux variables "num1" et "num2" pour sauvegarder les deux numéros tapés dans notre formulaire.
- Dans votre fonction, tapez `num1 = document.getElementById("firstNumber").value`; C'est un code pour récupérer la donnée à partir de votre `<input>` avec son `id="firstNumber"`.

- Répétez le même processus pour récupérer le second numéro.
- Pour calculer le produit et l'afficher, ajouter le code `document.getElementById("result").innerHTML = num1 * num2;`
- Testez votre code. A cette étape, votre page web doit être fonctionnelle pour faire la multiplication de deux numéros.
- Répétez le même processus pour finir les opérations : `addition()`, `soustraction()` et `division()`.

- Améliorez la présentation de votre page web afin que maintenant elle présente également l'opération concernée. Par exemple, si on clique sur le bouton « Diviser », ça affiche :

Calcul de base

1er Numéro:

2e Numéro:

Le résultat est :

10 / 20 = 0.5

- Pour afficher l'opération concernée, ajouter `` dans votre code avant ``. Pourquoi ?
- Dans votre code JavaScript, partie multiplication(), avant l'affichage de résultat, ajouter :
`document.getElementById("operation").innerHTML = num1 + " * " + num2 + " = ";`

2. Cette partie a pour but d'introduire le concept des fonctions `prompt()` et `alert()`. La fonction `prompt()` peut afficher un texte et de saisir l'entrée de l'utilisateur. La fonction `alert()` présente un texte ou un résultat. En cliquant sur un bouton, demandez le nom de l'utilisateur. Cela étant, on le salue. Par exemple :

Cliquez sur le bouton pour lancer la fonction

Cette page indique

Entrez votre nom, SVP

Et après, un message pertinent est affiché, selon le cas :

Cette page indique

Bonjour Manolo

ou

Cette page indique

L'utilisateur a annulé l'invite.

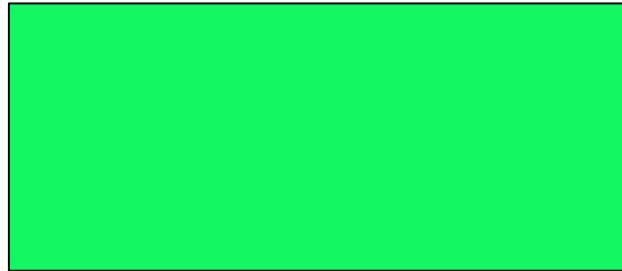
Toujours avec le `prompt()`, au lieu d'afficher le message par `alert()`, affichez votre message dans votre page web.

- Étapes à suivre :**
- Dans la partie `<body></body>` de votre page web, ajouter `<p></p>` et écrire le texte indiqué : **"Cliquez sur le bouton pour lancer la fonction"**.
 - Utiliser `<button>Cliquez-moi</button>` pour créer un bouton. C'est une autre façon de créer un bouton. Ajoutez `onclick="salutation()"` comme un paramètre de la balise `<button>`.
 - Créer la partie JavaScript avec `<script></script>`. Dans la partie `<script>`, ajouter la fonction `salutation()` avec `function salutation() {}`.
 - Dans votre script, ajouter `var personne = prompt("Entrez votre nom, SVP", "Harry Potter");`

- Ce code affichera le résultat #2 (voir l'image). Il attend que vous tapiez votre nom ou garder "Harry Potter", et cliquer sur le bouton "OK" ou "Annuler". Votre réponse est sauvegardée dans la variable **personne**. Notez bien qu'avec ou sans **var**, votre JavaScript fonctionne la même chose.
- On va ajouter la partie décisionnelle. Tapez le code suivant :

```
if (personne == null || personne == "") { alert("L'utilisateur a annulé l'invite."); }
else { alert("Bonjour " + personne); }
```

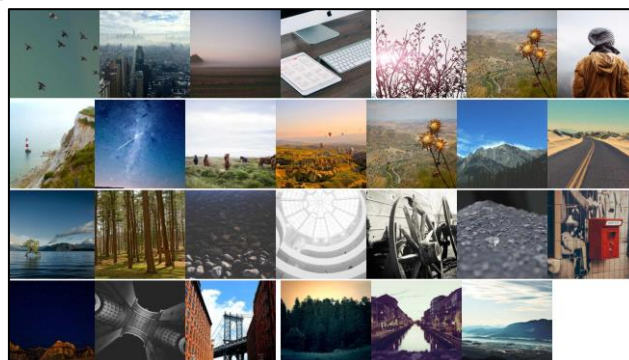
3. En utilisant la fonction **random()**, générer une valeur aléatoire, entre 0 et 255, de la quantité de couleur rouge, couleur verte et couleur bleu pour **rgb(valeur1, valeur2, valeur3)**. Utiliser cette valeur **rgb()** pour changer la couleur de fond de votre page web. Voici un exemple de vos résultats rgb():



Etapes à suivre :

- Dans la partie `<body></body>` de votre page web, ajouter `<p></p>` et écrire de texte approprié, par exemple : "**Cliquez le bouton pour changer la couleur de fond**".
- Utiliser `<button>Cliquez-moi</button>` pour créer un bouton. Ajoutez `onclick="colorMyWorld()"` comme un paramètre de la balise `<button>`.
- Dans la partie JavaScript, créer une fonction `colorMyWorld()`. Dans cette fonction, nous allons récupérer des valeurs aléatoires pour les couleurs rouge, verte et bleu. On va commencer avec la couleur rouge. Ajouter le code suivant : `let r = Math.floor(Math.random() * 256);` Avec ou sans `let`, le code fonctionne de la même façon. A l'exécution de ce code, la valeur de notre variable `r` est une valeur aléatoire entre 0 et 255.
- Répétez le même processus pour récupérer la valeur aléatoire de couleur verte et de couleur bleu.
- Ajouter le code suivant : `let color = "rgb(" + r + ", " + g + ", " + b + ")";` dont `r` contient la valeur **red**, `g` la valeur **green** et `b` la valeur **blue**.
- Effacer l'écran avec le code suivant : `document.body.innerHTML = "";`
- Afficher la couleur de fond avec le code suivant : `document.body.style.background = color;`

4. Le service [Picsum](https://picsum.photos/) permet d'obtenir des images libres de droit depuis une adresse web. Par exemple, le lien ci-dessous affiche l'image avec l'identifiant #42 en résolution 300 x 300 : <https://picsum.photos/id/42/300/300>. Créer une fonction qui affiche des images aléatoires depuis Picsum. Par exemple :



(NB : C'est possible que certaines images soient défectueuses et n'affichent pas)

Etapes à suivre :

- Dans la partie `<body></body>` de votre page web, ajouter `<p></p>` et écrire de texte approprié de votre choix. Utiliser `<button>Cliquez-moi</button>` pour créer un bouton. Ajoutez `onclick="galerieImages()"` comme paramètre de votre balise `<button>`.
- Dans la partie JavaScript, créer une fonction `galerieImages()`. Dans cette fonction, nous allons afficher les images spécifiées dans le site web mentionnée ci-dessus.
- D'abord on a besoin le nombre d'images avec `let count = 30;`
- On a besoin une variable de concaténation avec `let html = "";`
- On va générer une boucle des images. Pour chaque boucle, le numéro de cette image soit aléatoire.

- Tapez le code suivant :


```
for (let i = 0; i < count; i++) {
  let tmp = Math.floor(Math.random() * 1024);
  html = html + '';
}
```
- Afficher toutes les images avec `document.body.innerHTML = html;`

5. Créer une table de multiplication affiché dans votre page web. D'abord, demander l'utilisateur de saisir le nombre de ligne et le nombre de colonne. Après, afficher cette table de multiplication.

- Étapes à suivre :**
- Dans la partie `<body></body>` de votre page web, ajouter `<p></p>` et écrire de texte approprié de votre choix. Utiliser `<button>Cliquez-moi</button>` pour créer un bouton. Ajoutez `onclick="tableMultiplication()"` comme paramètre de votre balise `<button>`.
 - Dans la partie JavaScript, créer la fonction `tableMultiplication()`.
 - Dans cette fonction, utiliser `prompt` pour demander combien lignes d'une table de multiplication qu'on souhaite avec : `let lignes = prompt("Nombre de lignes ?");`
 - Répétez le même processus pour récupérer le nombre de colonnes de cette table.
 - Notez bien que JavaScript comprend bien toutes les balises HTML.
 - On commence en créant une variable de concaténation avec : `let tableau = '<table>';`
 - On ajoute une ligne dans ce tableau : `tableau += '<tr>';`
 - On fait une boucle pour afficher le tableau par ce code :


```
for (let i = 1; i <= lignes; i++) {
  for (let k = 1; k <= colonnes; k++) {
    tableau += '<td>' + i * k + '</td>';
  }
  tableau += '</tr>';
}
```
 - On complète notre code avec : `tableau = tableau + '</table>';`
 - Finalement, on affiche ce tableau : `document.body.innerHTML = "Table de Multiplication
" + tableau;`

6. L'objet **console** permet d'accéder à la console de débogage de votre navigateur web. La méthode `log()` de l'objet console écrit (journalise) un message sur la console. La méthode `log()` est utile à des fins de test.

Ecrire une fonction qui écrit/journalise un tableau dans la console. Ecrire également un tableau sous forme de tableau dans la console. Pour voir la sortie d'écran, tapez sur le clavier « F12 » et sélectionner « console ».

- Étapes à suivre :**
- Dans la partie `<body></body>` de votre page web, ajouter `<p></p>` et écrire de texte approprié de votre choix. Utiliser `<button>Cliquez-moi</button>` pour créer un bouton. Ajoutez `onclick="consoleLog()"` comme paramètre de votre balise `<button>`.
 - Dans la partie JavaScript, créer la fonction `consoleLog()`.
 - Ajouter ce code dans votre fonction :


```
const myArr = ["orange", "banane", "mange", "kiwi"];
console.log(myArr);
console.table(["Audi", "Peugeot", "Toyota"]);
```
 - Voir le console log de votre code.

7. Cette partie a pour but de valider (ou pas) les données dans un formulaire. Créer une page HTML contenant un formulaire avec 4 champs : nom, âge, téléphone, date de naissance. Voir l'image ci-dessous :

Dynamisme avec JavaScript :: Formulaire

Nom:

Age:

Téléphone:

Date de naissance:

- Lors du clic sur le bouton valider du formulaire, une fonction JavaScript doit être appelée.
- Si l'un de 4 champs est vide, la fonction doit afficher un pop-up « Un champ est vide ».
- La fonction doit pouvoir distinguer lequel des champs est vide lorsque le formulaire n'est pas complet.
- Créer un bloc « errordiv » dans la page HTML, puis rendre ce block invisible
- Dans le cas où l'un des champs n'est pas vide, afficher le bloc « errordiv »
- Le bloc « errordiv », lorsqu'il est affiché, doit contenir le ou les noms des champs non remplis.
- Modifier le div pour qu'il apparaisse sur fond rouge.
- A présent, modifier le contrôle du champ téléphone pour vérifier qu'il contient bien 10 chiffres.
- Faire de même avec le nom (uniquement des caractères), l'âge (3 chiffres maximum) et la date de naissance (au format jj/mm/aaaa).

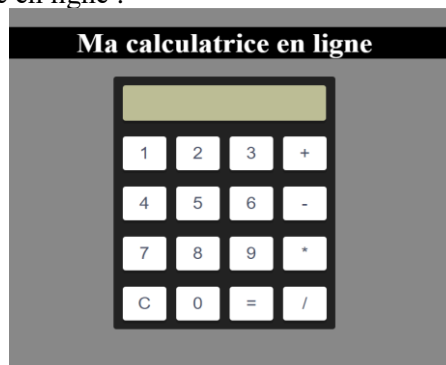
- **Étapes à suivre :**
- Dans la partie `<body></body>` de votre page web, créer le formulaire qu'on a besoin avec `<form></form>`. Ajouter `<table></table>`, `<tr></tr>` et `<td></td>` pour créer un tableau.
- Ecrire les textes appropriés et utiliser des `<input>` avec ses paramètres **type** pour identifier le type de donnée à récupérer et des **id** pour associer les variables avec les `<input>`.
- Créer le bouton "Submit". Associer le paramètre `onclick= "validate()"` pour ce bouton.
- Après `</form>`, ajouter un `<div></div>` et nommer cet élément **errordiv** avec son paramètre **id**.
- Ajouter `<style></style>` pour ajouter des styles css. Dans votre code css, ajouter ce style :
- ```
#errordiv {
 padding: 10px;
 background-color: rgba(200, 0, 0, 0.75);
 border: solid 1px;
 border-radius: 5px;
 width: 200px;
 min-height: 100px;
 visibility: hidden;
}
```
- Cette fois, nous voulons un code Javascript qui est sauvegardé dans un fichier **"myScripts.js"**. Pour le faire, ajouter cette partie :
- ```
<script type="text/javascript" src="myScripts.js"> </script>
```

- Dans votre code JavaScript sauvegardé dans le fichier **myScripts.js**, taper ce code:
- ```
message="";
function validate() {
 testEmpty();
 validateName();
 validateAge();
 validateTelephone();
 validateBirthday();
 message="";
} //fin function
```
- Dans la partie **testEmpty()**, nous voulons tester s'il y a un champ non rempli dans le formulaire. Dans la partie **validateName()**, nous voulons savoir s'il y a des erreurs dans le champ Nom. Même principe pour les parties **validateAge()**, etc.
- Voici le code pour la partie **testEmpty()** :
- ```
function testEmpty() {
    document.getElementById("errordiv").style.visibility = 'hidden';
    if ((document.getElementById("name").value === "") ||
        (document.getElementById("age").value === "")) ||
```

```

        (document.getElementById("phone").value === "") ||
        (document.getElementById("birthday").value === "") {
            alert("Un champ est vide");
        }
    } //fin function
    ➤ Voici le code pour la partie validatename() :
        function validateName() {
            if (document.getElementById("name").value === "") {
                var messageName1 = "<br>Le champ Nom est vide.";
                message += messageName1;
                document.getElementById("errordiv").innerHTML = message;
                document.getElementById("errordiv").style.visibility = 'visible';
            } //fin if
        } //fin function
    ➤ Compléter les codes pour les autres fonctions. Vous devez identifier toutes sortes des erreurs dans chaque champ.
    
```

8. Avec le principe de calcul de base, êtes-vous capable de designer votre propre calculatrice en ligne ?
Voici l'image d'une calculatrice en ligne :



➤ **Etapes à suivre :**

- Dans la partie `<body></body>` de votre page web, écrire le texte nécessaire avec : `<h1>Ma calculatrice en ligne</h1>`
- Ajouter un formulaire avec `<form></form>`. Ajouter `<table></table>`, `<tr></tr>` et `<td></td>` pour créer un tableau.
- Pour la première ligne, on a une ligne avec une seule colonne qui occupe quatre places. On peut le faire avec ce code : `<tr><td colspan="4"><input type="text" name="affiche" id="affiche"></td></tr>`
- Pour la deuxième ligne, on va afficher "1", "2", "3" et "+", on peut le faire avec ce code :


```

            <tr>
                <td><input type="button" name="un" value="1" onclick="affiche.value += '1' "></td>
                <td><input type="button" name="deux" value="2" onclick="affiche.value += '2' "></td>
                <td><input type="button" name="trois" value="3" onclick="affiche.value += '3' "></td>
                <td><input type="button" name="plus" value="+" onclick="affiche.value += '+' "></td>
            </tr>
            
```
- Pour réaliser le bouton "=", taper `<td><input type="button" name="egale" value="=" onclick="affiche.value = eval(affiche.value)"></td>`
- Pour réaliser le bouton "C", comment effacer toutes choses dans votre écran ?
- Compléter tout le reste de code manquant
- Ajouter les styles suivants dans votre code css. Etudiez bien chaque propriété utilisée dans ce code.

```

body {
    background-color: #888;
}
h1 {
    text-align: center;
    color: white;
    background-color: black;
    font-size: 40px;
}
table {
    margin: auto;
    background-color: #222;
    width: 295px;
    max-width: 295px;
    height: 325px;
    text-align: center;
    
```

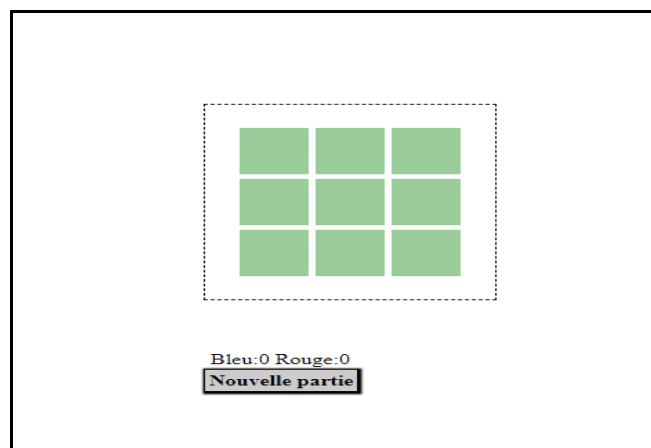
```
border-radius: 4px;
padding-right: 10px;
}
input {
  outline: 0;
  position: relative;
  left: 5px;
  top: 5px;
  border: 0;
  color: #495069;
  background-color: #fff;
  border-radius: 4px;
  width: 60px;
  height: 50px;
  float: left;
  margin: 5px;
  font-size: 25px;
  box-shadow: 0 4px rgba(0, 0, 0, 0.2);
  margin-bottom: 15px;
}
#affiche {
  width: 280px;
  max-width: 280px;
  font-size: 28px;
  text-align: right;
  background-color: #bc9d5b;
  float: left;
}
```

B. HTML, CSS, et JavaScript : Jeu de Morpion/ Tic-Tac-Toe | (TP de 1h + Travail à la maison de 1h)

➤ [Travail d'une équipe en binôme et à soumettre sur Boostcamp](#)

Résultat d'apprentissage visé : Pour cet exercice, les élèves doivent utiliser HTML, CSS et JavaScript pour réaliser un jeu de morpion.

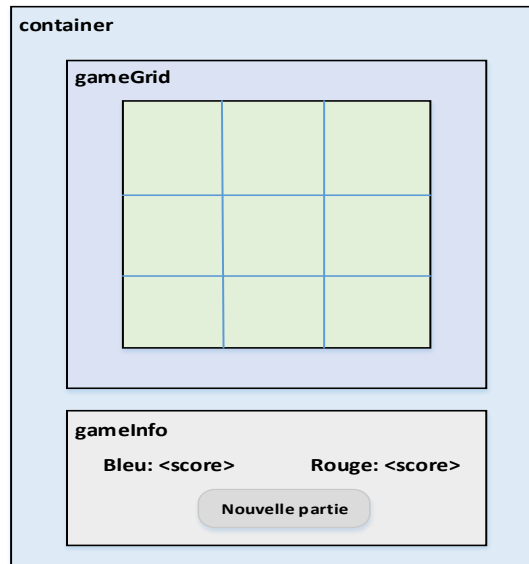
Dans cet exercice, nous voulons créer un jeu très simple de morpion (Tic-Tac-Toe en anglais). Pour ce faire, vous allez écrire des codes, un pour HTML, CSS et JavaScript. Les spécifications du code sont indiquées ci-dessous:



A. HTML

- Dans votre page Web, vous créez trois blocs principaux, dont «container», «gameGrid» et «gridInfo», comme indiqué ci-dessous. Le tableau morpion sera construit en neuf panneaux carrés (ou en neuf cases).
- Le bloc gameGrid crée une boîte autour du tableau de morpion.

- Au début, le tableau est tout vert. Les joueurs bleu et rouge jouent en alternance. Une fois que vous avez cliqué sur une case du tableau, la couleur de cette case devient celle de la couleur du joueur. Un message approprié s'affiche lorsqu'un joueur clique sur une case qui a déjà la couleur d'un joueur.



- Les scores sont affichés dans le bloc gameInfo. Un clic sur le bouton « Nouvelle partie » réinitialise le jeu et le jeu recommence.

B. CSS

- Définissez la marge gauche du bloc conteneur sur une certaine valeur, telle que 25% ou 30%, en fonction du style qui vous convient (par exemple, `#container {margin-left: 25%;}`).
- Voici quelques suggestions que vous pouvez appliquer au bloc gameGrid: sa largeur (« width ») et sa hauteur (« height ») en pixels doivent être identiques (il s'agit d'un rectangle). Vous pouvez configurer son remplissage (« padding ») à 20px, par exemple. La bordure montrée ci-dessus était configurée comme suit : «`border: 2px solid black;`». Vous pouvez spécifier sa couleur d'arrière-plan (background-color), ainsi que son bord supérieur (margin-top). Vous êtes libre d'ajouter d'autres styles.
- Chaque planche ou case de morpion doit avoir la même largeur et la même hauteur (par exemple, 50px;). Vous pouvez attribuer une marge à gauche (margin-left) et une marge en haut (margin-top) (telle que 5 pixels) à chaque cas. Vous devez également affecter un flottant à gauche à chaque cas.
- Chaque planche ou caisse de morpion peut être colorée à l'aide de la propriété de couleur d'arrière-plan (background-color).
- Configurez le style du reste de votre plateau de jeu, selon vos préférences.

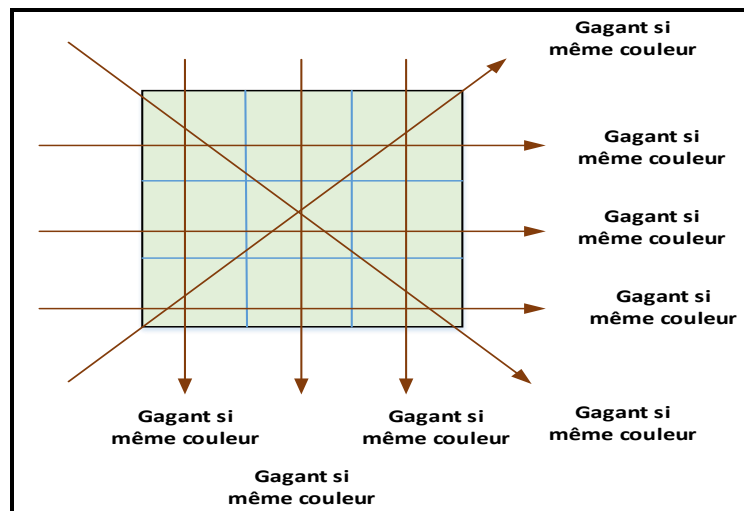
C. JavaScript

- Votre code JavaScript doit être dans un fichier séparé.
- Créez une fonction qui rechargera le tableau de morpion.

Web dynamique

ING2 | 2024

- Lorsqu'un clic ou un tableau est cliqué dans la page web, une fonction JavaScript correspondante est appelée. La couleur du joueur est utilisée pour colorier la case d'un tableau correspondant.
- Créez une fonction pour déterminer s'il y a un gagnant. Un gagnant existe si l'une des 8 conditions ci-dessous est remplie (c'est-à-dire que trois cases ou panneaux sont de la même couleur)



- Si aucun gagnant n'est appelé, alors votre jeu devrait indiquer que c'est à l'autre joueur de jouer.
- Si un gagnant est appelé, le score de couleur du joueur est incrémenté de 1.
- Créez d'autres fonctions adaptées à votre travail

Voici un exemple de résultat :

