


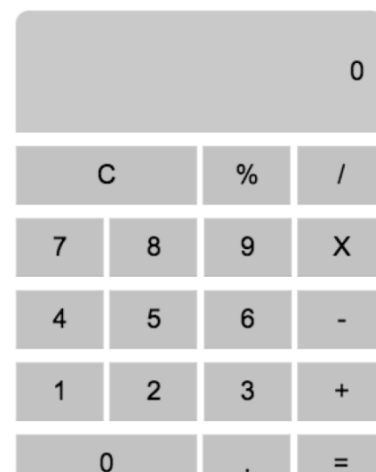
Seq 4	Le WEB	TP
NSI	Dynamiser vos pages avec le Javascript	

1. INTRODUCTION

Pour faire en sorte d'attirer des lecteurs ou internautes il faut souvent que le site soit dynamique, des animations, des choix via des clics de souris.....si vous désirez aussi réaliser des petits jeux mais aussi des calculs, alors il est nécessaire de faire appel au langage Javascript.

2. Réalisation d'une calculatrice

Cette activité est divisée en 2 parties: le Front End et le Back End. Nous allons créer la structure de notre calculatrice en utilisant HTML et l'embellir en utilisant CSS dans la partie Front-end.
 Dans la partie Back End, nous allons lui donner une vie, c'est-à-dire le faire fonctionner pour résoudre les équations. La partie back-end se fera en utilisant javascript.
 Notre projet final ressemblera à ceci:



Ouvrir l'éditeur de votre choix et nommer le avec l'extension .html.

Nous devons définir une zone où nous allons montrer la sortie à l'utilisateur.

Nous ferons une division avec affichage de classe

Notre première rangée de boutons aura 3 boutons, bien que la rangée suivante aura 4 boutons. En effet, le bouton Effacer de la première rangée remplacera 2 boutons car il s'agit du bouton principal de notre calculatrice (0 également). Vous l'avez peut-être également vu dans la capture d'écran ci-dessus.



On définit un nouveau div avec la classe «première ligne». Ici, tous nos boutons appartenant à la première ligne seront placés ici. Les boutons seront «C», «%» et «/».

```
<div class="first-row"> </div>
```

L'identifiant doit suivre le format: «row_num-button_num».

Exemple: si le bouton est pour la ligne 1 et est le premier bouton de la ligne, il aura un identifiant «r1-1».

Voir l'exemple ci-dessous :

```
<button id="r1-1">C</button>
```

Votre calculatrice à ce stade devrait ressembler à:

Output

0



Passons à la section CSS.

Vous devez donc associer le nouveau fichier css (dites «style.css») à la page HTML.
Pour cela, à l'intérieur de la balise head, mettre le code ci-dessous:

```
<link rel="stylesheet" href="style.css" type="text/css">
```

Nous devons le faire ressembler à la figure comme dans la page 1.

Donc, 5 choses seront impliquées dans la création de l'écran de sortie :

- C'est la couleur de fond
- Rembourrage
- Largeur
- Alignement du texte
- Taille de police

Alors le code CSS sera:

```
.display { background-color: rgb(201, 201, 201);  
padding: 15px; width: 250px;  
text-align: right;  
font-size: 20px;  
border-top-left-radius:10px;  
border-top-right-radius:10px;}
```

Maintenant, nous allons écrire du code CSS commun pour tous les boutons, car nous voulons qu'ils se ressemblent. (sauf 0 & C, nous y reviendrons).

Cela impliquera des 7èmes:

- Rembourrage
- Largeur
- la taille
- Marge
- Couleur de l'arrière plan
- Couleur du texte
- Taille de police

```
button {  
padding: 10px;  
width: 67px;  
height: 45px;  
margin-top: 10px;  
background-color: rgb(194, 194, 194);  
color: black; font-size: 20px;
```

```
}
```

Référez-vous au code:

```
<button id="r1-1" style="width: 50px;">C</button>
<button id="r5-1" style="width: 50px;">O</button>
```

Pour cela, nous devons changer le curseur en pointeur et la couleur d'arrière-plan un peu plus sombre quand il est survolé. En outre, il devrait paraître un peu plus sombre lorsque vous cliquez dessus.

```
button:hover { cursor: pointer; background-color: rgb(182, 178, 178);}
button:active { background-color: rgb(136, 136, 136);}
```

Maintenant le JavaScript, qu'il faudra lier à notre fichier HTML, avec le code à l'intérieur de la balise head.

```
<script src="script.js">
```

Maintenant, faisons notre première fonction, qui efface tout les infos sur l'écran de sortie, lorsque le bouton «C» est déclenché.

Appelons notre fonction *forclear* ().

```
function forclear() {
    document.getElementById("output").innerHTML = "0";}
<button id="r1-1" onclick="forclear()" style="width: 50px;"> C </button>
```

Nous devons maintenant définir une fonction qui supprime le 0 par défaut sur l'écran de sortie, sinon toute valeur mise à jour sur l'écran de sortie sera concaténée avec 0, comme ci-contre :

0252

Nous devons supprimer ce zéro avant le début de toute opération. Ainsi, nous définissons une nouvelle fonction «removeZero ()», qui remplace le 0 par juste un espace si 0 est présent avant le début de toute opération.

```
function removeZero()
{ var value = document.getElementById("output").innerHTML;
  if (value == "0")
    { value = " " document.getElementById("output").innerHTML = value;
    }
}
```

Pour « % » définissons une fonction `perc()`, qui récupère la valeur de l'écran de sortie, la divise par 100 puis la renvoie.

```
function perc()
{
    var value = document.getElementById("output").innerHTML;
    value = value / 100;
    document.getElementById("output").innerHTML = value;
}

<button id="r1-3" onclick="perc()"> % </button>
```

Il reste maintenant 2 choses:

- Mettre à jour la valeur sur l'écran de sortie lorsqu'un bouton est cliqué
- Résoudre l'équation sur l'écran de sortie une fois que le bouton «`=`» est cliqué.

Puis afficher les valeurs sur l'écran de sortie.

Pour cela, nous définissons une nouvelle fonction d'affichage (*valeur*), où *valeur* est le paramètre qui n'est autre que les valeurs numériques ou les opérateurs arithmétiques.

Voyons le code:

```
function fordisplay(valeur)
{
    removeZero()
    document.getElementById("output").innerHTML += valeur;
}
```

Ensuite, il met à jour la valeur de l'écran de sortie avec la valeur ou l'opérateur qui est déclenché.

Mais nous devons définir la propriété «`onclick`» pour chaque nombre et opérateur dans la section HTML.

La «*valeur*» sera remplacée lors du déclenchement du bouton correspondant.

Par exemple :

```
<button id="r3-1" value="4" onclick="fordisplay('4') ">4</button>
```

Votre code ressemblera à ceci:

```
<div class="third-row">
  <button id="r3-1" value="4" onclick="fordisplay('4') ">4</button>
  <button id="r3-2" value="5" onclick="fordisplay('5') ">5</button>
  <button id="r3-3" value="6" onclick="fordisplay('6') ">6</button>
  <button id="r3-4" value="-" onclick="fordisplay('-') ">-</button>
</div>
```

Maintenant, quoi que nous appuyions sur la calculatrice, il est mis à jour sur l'écran de sortie.

Nous devons afficher le résultat de l'opération une fois que le bouton «`=`» est déclenché.

Pour cela, nous définissons une fonction *résoudre()* pour cela.

Mais la question est de savoir comment nous allons résoudre l'équation?

Grâce à JavaScript, nous avons une fonction *eval* (équation) qui évalue / exécute l'équation passée en paramètre.

Nous utiliserons ce concept ici.

```
function solve() {  
    removeZero()  
    var equation = document.getElementById("output").innerHTML;  
    var solved = eval(equation);  
}
```

Ensuite, nous mettons à jour l'écran de sortie avec la valeur stockée dans *résolu*. C'est la réponse de l'utilisateur d'équation envoyé pour résolution.

Dernière étape, nous devons dire au bouton «=» ce qu'il doit faire une fois qu'il a cliqué.

```
<button id="r5-3" value="=" onclick="solve()" >=</button>
```