

CR TDM03 JAVASCRIPT

Noémie PRIN et Maria-Bianca ZUGRAVU

Bonnes pratiques

- placer le script Java dans le **<footer>** (plutôt ici) ou dans le **<head>**
- à cause du traitement séquentiel des pages par le navigateur si on place les scripts dans le **<footer>** ils vont être interprétés après les chargements des éléments visuels
- appel d'une fonction ou exécution d'une commande Javascript dans le **<body>**
- utilisation du JAVASCRIPT dans une URL: si on clique

`<p>`

Dans une fonction appelée en cliquant ` ici `

`<p>`

Notions importantes!! :

- **getElementByTagName(tagname)** ressort un tableau
- **getElementById(elementID)** ressort un seul élément
- Une **nodeList** est un tableau d'éléments, comme celui qui est renvoyé par la méthode **document.getElementsByTagName()**. Les éléments d'une **nodeList** sont accessibles par un index de deux manières différentes :
 - `list.item(1)`
 - `list[1]`

SCRIPTS & commentaires

Calculatrice V1

Enoncé

Réalisez une page HTML contenant du code Javascript ouvrant une pop-up permettant la saisie d'une expression à évaluer. Cette expression est ensuite calculée et affichée avec son résultat (utilisation de `eval()`).

Quelques méthodes utiles et remarques:

- **getElementById()**: retourne un élément représentant l'élément dont l'ID correspond à la chaîne entrée

Syntaxe : `var element = document.getElementById(id);`

- **createTextNode()**:

Syntaxe : var text = document.createTextNode(donnees);

- texte est un nœud de texte;
- donnees est une chaîne contenant les données à placer dans le nœud de texte.
- **insertBefore()** : pour insérer un noeud comme étant un fils, juste avant un fils existant, que l'on spécifie; cette méthode pourrait être utile pour déplacer un élément existant

Script à nous:

```
<!DOCTYPE html>
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8"/>
    <title>Calculatrice V1</title>
  </head>
  <body onload="javascript:popup();" >
    <header>
      <h1>Calculatrice V1</h1>
    </header>
    <footer>
      <script type="text/javascript">
        // <!--
        function popup() {
          while (true){
            op=window.prompt("Expression","");
            res=eval(op);
            window.alert(res);
          }
        }
        // -->
      </script>
    </footer>
  </body>
</html>
```

Correction

```
<!DOCTYPE html >

<html >
  <head >
    <title> Calculatrice </title>
    <meta http-equiv= "Content-Type" content="text/html; charset=utf-8">
  </head>
  <body>
    <header>
```

```

    <h1> Calculette en javascript</h1>
</header>
<p id= "paragraphe">
<br id= "saut"/>
</p>
<p>
    <a href="correctionV1.html"> Nouveau calcul</a>
<!--
Pour un nouveau calcul ça revient à utiliser de nouveau ce script.
De cette manière, sans placer le onload dans la balise <body> on evite
de lancer automatiquement une fenetre pop up (on le fait seulement si l'utilisateur le
/-->
</p>
<footer>
    <script type= "text/javascript">
        var expression = prompt("Entrer un calcul : " ," " ) ;
        var equation = document.createTextNode ( expression + " = " + eval(expression)) ;
        var paragraphe = document.getElementById("paragraphe") ;
        var saut = document.getElementById("saut") ;
        paragraphe.insertBefore(equation , saut);
        /* Pour afficher l'équation juste après la balise de titre et avant la balise
        autofermante br. L'équation fera partie du paragraphe*/
    </script>
</footer>
</body>
</html >

```

Calculatrice V2

Enoncé

Reprenez le code du premier exercice et proposez la saisie de l'équation à l'intérieur du document HTML, dans une boîte de saisie. Le calcul est effectué par un bouton et inséré, selon la valeur d'une série de boutons choix, soit dans un champs texte, soit dans une fenêtre d'alerte.

Quelques méthodes utiles et remarques:

- à l'aide de **<onclick="calculer()"** on fait l'appel au Javascript pour donner le résultat
- evenements d'exécution de la fonction:
 - **onclick**
 - **onmouseover**
 - **onfocus**
 - **onselect**
 - **onchange** : modification du contenu

- **onsubmit** : pour un formulaire
- **onload** : au chargement d'un formulaire
- **onunload** : à la fermeture d'un élément chargé ou au chargement d'une autre page que la courante
- à l'aide de la méthode **value** on peut interpreter l'opération entrée par l'utilisateur :

Syntaxe : `var expression = document.getElementById("expression").value`

- **à propos du Javascript**: On a 2 cas à traiter en fonction du choix de l'affichage. Pour faire ceci on rajoute une fonction test qui retourne un int correspondant au choix (0: champ texte, 1: fenêtre alerte et -1: si aucun choix a été effectué).
- Petites remarques:
 - on peut utiliser **radio.length** pour compter combien d'éléments de type radio ont été définis
 - on utilise une notation pointée par **affichage** pour indiquer qu'on veut passer comme paramètres seulement les éléments qui ont le **name** = "affichage" qui sont compris dans **<form>**
 - les éléments qui ont le même nom sont regroupés dans un tableau qui a pour nom le type de ces éléments.

if (**radio[i].checked**)

Syntaxe: `test(document.getElementById("formulaire").affichage)`

Script

```
<!DOCTYPE html>
<html>
  <head>
    <title> Calculette 2 </title>
    <meta http-equiv= "Content-Type" content= "text/html; charset=utf-8">
  </head>
  <body>
    <header>
      <h1> Calculette 2 en javascript</h1>
    </header>
    <form id= "formulaire">
      <fieldset>
        <label> Emplacement de la réponse : </label>
        <input type= "radio" name= "affichage" id= "texte" checked= "checked" />
        <label for="texte"> formulaire</label>
        <input type="radio" name= "affichage" id= "pop-up"/>
        <label for="pop-up">pop-up</label>
      </fieldset>
    </form>
  </body>
<!-- L'ordre des définition des balises compte ici car on va simuler un case of
```

en fonction du choix effectué sur l'affichage. En fait, chaque element inclu dans l'element référencé par l'id="formulaire" qui a le name="affichage" on va lui attribuer un index dans la fonction test pour être capables de créer les conditions if. Les indexes commencent à 0!

```
-->
<hr/>
    <input id= "expression" type= "text" size= "15"/>
    <input type= "button" value= "=" onclick= "calculer()" />
    <!-- On lance le script javascript en cliquant sur le button = -->
    <input id= "reponse" type= "text" size= "6" />
</fieldset>
</form>
<footer>
    <script type="text/javascript">
        function calculer () {
            var expression = document.getElementById("expression").value;
            var affichage = test(document.getElementById("formulaire").affichage);
            if (affichage==0){
                var reponse = document.getElementById("reponse");
                reponse.value = eval(expression);
            } else if(affichage==1)
                alert (expression + "=" + eval(expression));
            else
                alert("Choisissez un affichage !");
        }
        function test(radio) {
            for (var i=0; i<radio.length;i++)
                if (radio[i].checked)
                    return i;
            return -1;
        }
    </script>
</footer>
</body>
</html>
```

Gestion d'un horloge

Enoncé

Réalisez une page HTML contenant du code Javascript affichant la date et l'heure, qui sera mise à jour à intervalles réguliers (toutes les secondes).

Syntaxe : var maintenant = **new** Date()

- Si on appelle simplement Date() la valeur retournée est un string et pas un objet de type Date. On peut aussi faire un appel : *new Date(value)* (ex:

var date = new Date('December 17, 1995') qui nous retourne un objet de type Date qui a pour valeur le string entré)

Quelques méthodes utiles et remarques:

- **setInterval(function, milliseconds, param1, param2, ...)** : appelle la fonction à des intervalles réguliers spécifiés par *milliseconds*. Cette méthode va continuer à appeler la fonction jusqu'au moment où **clearInterval()** est appelée ou la fenêtre web est fermée.
- **body.appendChild(date)** : on insère la date comme fils de *<body>* après le 1er fils de *<body>* car la variable *body* était définie comme suit: *var body = document.getElementsByTagName("body")[0]*; en fait ici on accède à la balise *<header>*
- **getDate()** : pour retourner l'objet de type Date, obtenir le jour
- **getMonth()** **getFullYear()** **getHours()** etc
- **getMonth()** : renvoie les mois de 0 à 11, donc si on veut afficher le mois correct on fait +1. **ATTENTION** on fait le transtypage en *string* de l'objet retourné par *getMonth()* pour pouvoir le concatener
- **body.replaceChild(date)** : remplacer le fils de *<body>* par *date*
- **body.replaceChild(nouvelleHeure,vieilleHeure)**
- **getElementByTagName()** : *document.getElementsByTagName("body")[0]*
: on accède au premier élément du *<body>*

Script

```
<!DOCTYPE html>
<html>
  <head>
    <title> Horloge </title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
  </head>
  <body onload="javascript:depart()">
    <header> <h1> Horloge</h1> </header>
    <footer>
    <script type="text/javascript">
      function depart () {
        afficheDate();
        setInterval(majHeure,1000);
        /*appel de la fonction majHeure chaque seconde = 1000 ms*/
      }
      function afficheDate() {
        var maintenant=new Date() ;
        var body = document.getElementsByTagName("body")[0];
        var date = document.createTextNode(maintenant.getDate() + "/"
          + String(maintenant.getMonth()+1)+ "/" + maintenant.getFullYear() + " ");
        /*maintenant.getMonth() + 1 car la méthode getMonth() renvoie les mois numérotés
        de 0 à 11*/
        body.appendChild(date);
      }
    </script>
  </body>
</html>
```

```

/*on affiche la date formatée après l'enfant descendant direct càd la balise <h1>*/
var heure = document.createTextNode(maintenant.getHours()+":"
    + maintenant.getMinutes()+ ":" + maintenant.getSeconds());
body.appendChild(heure);
/* après la date on affiche l'heure formatée*/
}
function majHeure() {
    var maintenant=new Date();
    var body = document.getElementsByTagName("body")[0];
    var vieilleHeure = body.lastChild;
    /*l'heure ancienne est le dernier enfant dans le <body>*/
    var nouvelleHeure = document.createTextNode(maintenant.getHours() + ":"
    + maintenant.getMinutes() + ":" + maintenant.getSeconds());
    body.replaceChild(nouvelleHeure,vieilleHeure);
    /*on remplace l'heure ancienne par la nouvelle*/
}
</script>
</footer>
</body>
</html>

```

Gestion du fenêtrage

Enoncé

Réalisez une page HTML contenant du code Javascript présentant dans un menu sous la forme de 3 liens la liste des applications Javascript précédentes. En passant (donc sans cliquer) votre pointeur sur un des liens, le document choisi sera affiché dans une seconde fenêtre/onglet.

Quelques méthodes utiles et remarques:

- **window.open**("correctionV1.html" ,"Calcullette") : pour ouvrir un script html; **Syntaxe générale** : *window.open(URL, name, specs, replace)*
- L'URL peut être un site: *window.open("https://www.w3schools.com")*;

Script

```

<!DOCTYPE html>
<html>
<head>
    <title> Menu </title>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>
<body>
    <header>
        <h1> Menu </h1>
    </header>

```

```

<p>
  <a href= " " onmouseover= "calcullette();" > Calcullette </a> <br/>
  <a href= " " onmouseover= "horloge();" > Horloge </a> <br/>
  <a href= " " onmouseover= "calcullette2();" > Calcullette 2 </a> <br/>
</p>
<footer>
<script type= "text/javascript">
  function calcullette() {
    window.open("correctionV1.html" ,"Calcullette");
  }
  function horloge() {
    window.open("correction3.html" ,"Calcullette");
  }
  function calcullette2() {
    window.open("correctionV2.html" ,"Calcullette");
  }
</script>
</footer>
</body>
</html>

```

Forum ASI v2

Enoncé

Reprenez la page HTML réalisée lors du TDM correspondant permettant de déposer un message sur le forum ASI sans s'enregistrer. Modifiez-la afin de vérifier le contenu du formulaire par du code Javascript, avant tout envoi. La vérification portera sur la présence obligatoire d'un email. En cas d'absence, un message sera ajouté à la page, comme présenté ci-dessous (annexe) 2. Un clic sur le bouton Effacer ouvrira un pop-up Javascript afin d'obtenir une confirmation. 3. Pensez à bien supprimer le message d'erreur en cas de saisie d'un email suivie d'une nouvelle tentative d'envoi, ou lorsque l'on appuie sur le bouton Effacer. *

La seule façon d'instancier un objet **Date** c'est de passer par l'opérateur **new**:

Quelques méthodes utiles et remarques:

- **overflow** dans la CSS: This property specifies whether to clip content or to add scrollbars when an element's content is too big to fit in a specified area.
- **object.style.outline = "width style color|initial|inherit"** : width style color initial inherit sont des params à passer

Example: `document.getElementById("myDiv").style.outline = "thick solid #0000FF";`

`emailBox.style.outline= "initial";`

- ```
document.forms[nom_formulaire].submit()
```

- pour remettre tous les champs du formulaire à vide :

- afficher un bouton pour cocher si on veut recevoir une copie

## Script

9

```

 }
};
function nettoyage() {
 if(confirm('Etes vous sur ?')) {
 document.getElementById("email").style.outline= "initial";
 var probleme=document.getElementById("probleme");
 if (probleme.hasChildNodes()) {
 probleme.removeChild(probleme.childNodes.item(0));
 }
 document.forms['formulaire'].reset();
 }
};
</script>
</head>
<body>
 <div id="page">
 <div id="top">

 <h1> Forum ASI</h1>
 </div>
 <p id= "probleme"> </p>
 <form name="formulaire" action="mailto : Alexandre.Pauchet@insa-rouen.fr" method=
 "post" enctype="text/plain">
 <fieldset>
 <legend> Coordonnées : </legend>
 <label for="name"> Name : </label>
 <input placeholder="Saisissez votre nom" id="name" name="name" type="text"
 size="30"/>

 <label for="email"> E-mail : </label>
 <input placeholder="Saisissez votre email" id="email" name="email"
 type="email" size="30"/>

 <label> Etudiant : </label>
 <input type="radio" id="etudiantinsaoui" name="etudiantinsa" value="Oui"/>
 <label class="labelradio" for="etudiantinsaoui">Oui</label>
 <input type="radio" id="etudiantinsanon" name="etudiantinsa" value="Non"/>
 <label class="labelradio" for="etudiantinsanon"> Non</label>

 <label for="annee"> Année : </label>
 <select name="annee" id="annee">
 <option value="1"> 1 </option>
 <option value="2"> 2 </option>
 <option value="3"> 3 </option>
 <option value="4"> 4 </option>
 <option value="5"> 5 </option>
 </select>
 </fieldset>
 </fieldset>
 </body>

```

```

 <legend> Message : </legend>
 <textarea rows="4" cols="50"> </textarea>

 <label for="copie"> Je souhaitez recevoir une copie du message : </label>
 <input type="checkbox" name="copie" id="copie" checked="checked"/>

 </fieldset>
 <input type="button" value="Poster le message"
 onclick="Javascript:verification();" />
 <input type="button" value="Effacer" onclick="Javascript:nettoyage();" />
</form>
</div>
</body>
</html>

```

## Points importants CM