

PROGRAMMATION WEB AVANCEE

TP1

EXERCICE 1

Utiliser une structure DOM pour écrire une page web permettant de créer un lien pointant vers la page de www.gmail.com en cliquant sur le texte Lien vers gmail, et ce de deux manières :

- a) en utilisant `getElementsByName()`
- b) `getElementById`

EXERCICE 2

Afin de s'entraîner à créer avec le DOM, on vous donne le code HTML suivant et on vous demande de recréer cette structure en utilisant le DOM en Javascript.

Le code que vous devez recréer est le suivant :

```
<div id="divEX2">
  Le <strong>World Wide Web Consortium</strong>, abrégé par le sigle <strong>W3C</strong>, est un
  <a href="http://fr.wikipedia.org/wiki/Organisme_de_normalisation" title="Organisme de normalisation">
    organisme de standardisation</a> à but non-lucratif chargé de promouvoir la compatibilité des
    technologies du <a href="http://fr.wikipedia.org/wiki/World_Wide_Web" title="World Wide Web">World
    Wide Web</a>.
</div>
```

EXERCICE 3

L'objectif de cet exercice consiste à pouvoir accéder aux éléments à partir de l'ensemble du document. On tentera de travailler avec les méthodes `getElementById`, `getElementsByName` et `getElementsByTagName`.

Travail demandé :

1. Créer une fonction `modif_paragraphe`, appelée en cliquant sur le titre. Cette fonction sélectionne le paragraphe en utilisant son identifiant, puis le modifie avec la propriété `innerHTML`, en remplaçant le mot **original** en caractères droit par le mot **corrigé**, en caractère italique.

2. Créer une fonction `centrage_h1`, appelée en cliquant sur le premier paragraphe. Cette fonction détecte d'abord les éléments portant le nom de balise `h1`. Elle sélectionne ensuite le premier d'entre eux, dans notre cas, il n'y en a qu'un seul paragraphe sur la page, et modifie son attribut `align`, en lui affectant la valeur `"center"`, à l'aide de la méthode `setAttribute`.

EXERCICE 4

Question 1

Soit le fichier

```
<!DOCTYPE html>

<html lang="fr">
<head>

<title>Gestion d'événement avec les propriétés des objets éléments</title>

<meta charset="UTF-8">

<script>

function Init()
{
}

</script>
</head>
<body onload="Init()">
<form id="formulaire">
<input type="text" value="Mot quelconque" id="champsaisie">
<input type="button" id="soumet" value="Soumettre">
<input type="button" id="raz" value="Remise à zéro">
</form></body></html>
```

1. Dans la fonction `Init()` :

- identifier chacun des trois éléments du formulaire ;
 - affecter le gestionnaire soumission à l'événement click sur le bouton de soumission, et le gestionnaire remise_a_zero à l'événement click sur le bouton de... remise à zéro.
2. La fonction soumission se contente d'afficher le texte saisi dans le champ texte ;
 3. La fonction de remise à zéro affiche une boîte d'alerte informant de l'action en cours, et remet le champ de saisie à sa valeur par défaut.

Question 2 (à remettre, question notée)

Soit le fichier

```
<!DOCTYPE html>

<html lang="fr">

<head>

<title>Un petit dessin</title></title>

<meta charset="UTF-8">

<style>

#grandcadre {

position: absolute;

top: 0px;

left: 0px;

border: dotted 2px black;

width: 600px;

height: 450px;

}

#petitcadre {

position: absolute;

top: 450px;

left: 600px;
```

```
border: solid 2px black;

width: 200px;

height: 150px;

}

.point {

position: absolute;

width: 3px;

height: 3px;

background-color: blue;

}

</style>

<script>

function Init()

{}

function crayon(evenement)

{}

function dessine(evenement)

{}

</script>

</head>

<body onload="Init()">
<div id="grandcadre"><p id="info"></p></div>

<div id="petitcadre"></div>
</body></html>
```

1. Dans la fonction Init() :
 - Identifier les deux cadres ;
 - Ajouter un gestionnaire d'événement associé à l'événement mousemove, lançant la fonction dessine, appliqué à l'élément d'identifiant petitcadre.
2. Dans la fonction dessine, afficher une boîte d'alerte donnant les coordonnées du pointeur.

Tracé de dessin

A la place de la boîte d'alerte, nous allons lancer le dessin dans le calque d'identifiant grandcadre. Pour cela :

1. Créer un nouvel élément div ;
2. L'ajouter au grand cadre ;
3. La propriété de style position de cet élément vaut « absolute », sa classe est « point » ;
4. Son abscisse doit être égale à $3 \times (\text{abscisse du pointeur} - 600)$, son ordonnée à $3 \times (\text{ordonnée du pointeur} - 450)$.

EXERCICE 5 : Interagir avec les médias

Imaginons que nous voulions proposer un lecteur sans contrôles natifs, mais uniquement avec nos boutons HTML que nous pourrions styliser via du CSS. Il faudrait alors que ces boutons interagissent avec la vidéo correctement. Admettons que nous voulions ajouter les contrôles suivants :

- **Lecture** : lit la vidéo ;
- **Pause** : met la vidéo en pause ;
- **Stop** : arrête la vidéo ;
- **-10s** : recule la vidéo de 10 secondes ;
- **+10s** : avance la vidéo de 10 secondes ;

Structure de base

Voici la structure de base que nous allons respecter :

```
<video id="mavideo" controls>
  <source src="http://clips.vorwaerts-gmbh.de/VfE_html5.mp4" type="video/mp4">
  <source src="http://clips.vorwaerts-gmbh.de/VfE.webm" type="video/webm">
  <source src="http://clips.vorwaerts-gmbh.de/VfE.ogv" type="video/ogg">

  <p class="alert">
    Votre navigateur ne supporte pas la balise vidéo ! Mettez-vous à jour !
  </p>
```

```
</video>
<div class="controles" hidden>
</div>

function lecture() {
    // Lit la vidéo
}

function pause() {
    // Met la vidéo en pause
}

function stop() {
    // Arrête la vidéo
}

function avancer(duree) {
    // Avance de 'duree' secondes
}

function reculer(duree) {
    // Recule de 'duree' secondes
}

function creerBoutons() {
    // Crée les boutons de gestion du lecteur
}
```

1. Créez une page web contenant une vidéo muette et un fichier son à démarrage automatique. Votre page devra pouvoir être compatible avec tous les navigateurs principaux.
2. Ajouter une balise object pour ouvrir un fichier swf(flash) si aucun format vidéo n'est reconnu par le navigateur
3. A l'aide du DOM ajouter les boutons et le code JavaScript correspondant pour contrôler votre vidéo :

Lecture : pour lancer la lecture de la vidéo

Pause : pour faire une pause de lecture

Stop : pour arrêter la lecture

"-10s" : pour revenir dans la lecture de 10 secondes

"+10s" : pour avancer dans la lecture de 10 secondes

Les boutons seront créés avec la fonction « ceerBouton » qui sera exécutée à a fin du chargement de la page.

4. Ajouter un *checkbox* « Répéter » qui fera répéter la vidéo lorsqu'elle se termine si elle est cochée ;
5. Ajouter du CSS3

EXERCICE 6 : Vidéo et CSS3

Tous les styles CSS, notamment les effets de transition, transformations géométriques et animation de CSS3 s'appliquent aux vidéos

1. Tester la transition ci-dessous à une video :

```
video {  
  width: 75px;  
  -o-transition: all 0.5s ease-in-out;  
  -webkit-transition: all 0.5s ease-in-out;  
  -moz-transition: all 0.5s ease-in-out;  
  transition: all 0.5s ease-in-out;  
}  
video:hover, video:focus { width:600px; }
```

2. Créer des transitions au choix sur une vidéo

EXERCICE 7

Écrire une page HTML contenant :

- un titre de niveau 1
- un titre de niveau 2
- faire en sorte que le titre h2 effectue une série de 20 rotations,

- Les rotations impaires se feront en grossissant le titre,
- Les rotations paires se feront en réduisant le titre,
- Les rotations ne commenceront que 5 secondes après le chargement de la page.

Exercice 8 (noté) événements CSS3

Soit le code source :

```
<!doctype html>
<html>
<head>
  <title>CSS animations: Animation events</title>
  <style type="text/css">
    .slidein {

        animation-duration: 3s;
        animation-name: slidein;
        animation-iteration-count: 3;
        animation-direction: alternate;

    }

    @keyframes slidein {
      from {
        margin-left:100%;
        width:300%
      }

      to {
        margin-left:0%;
        width:100%;
      }
    }
  </style>
  <script>
    function listener(e) {
      var l = document.createElement("li");
      switch(e.type) {
        case "animationstart":
          l.innerHTML = "Started: elapsed time is " + e.elapsedTime;
          break;
        case "animationend":
          l.innerHTML = "Ended: elapsed time is " + e.elapsedTime;
          break;
        case "animationiteration":
          l.innerHTML = "New loop started at time " + e.elapsedTime;
          break;
      }
      document.getElementById("output").appendChild(l);
    }

    function setup() {
```



```
var e = document.getElementById("watchme");
e.addEventListener("animationstart", listener, false);
e.addEventListener("animationend", listener, false);
e.addEventListener("animationiteration", listener, false);

e.className = "slidein";
}
</script>
</head>
<body onload="setup()">
  <h1 id="watchme">Watch me move</h1>
  <p>This example shows how to use CSS animations to make <code>H1</code>
elements
  move across the page.</p>
  <p>In addition, we output some text each time an animation event fires, so
you can see them in action.</p>
  <ul id="output">
  </ul>
</body>
</html>
```

1. Que fait ce code ?
2. Attacher un événement "animationstart", "animationiteration" et "animationend" à un élément DIV : faire une rotation de 45 degrés au début, à chaque itération : si l'itération est paire changer la couleur d'arrière plan du div, si elle est impaire faire une rotation du div de 70 degrés. A la fin changer la couleur du texte.
3. Appliquer les événements animationstart" et "animationend" des animations sur une vidéos. J'exige le maximum d'effets.

ANNEXE

QU'EST CE QUE LA BALISE OBJECT

Cet élément est conçu pour insérer toute sorte de document. On spécifie de quelle sorte avec l'attribut `type` en indiquant le type mime, et la source du document avec l'attribut `data` en indiquant son URI.

Si le navigateur, ou un de ses plugins, sait interpréter ce document, il le fait sinon le contenu imbriqué est affiché. Tout code HTML peut être insérer comme contenu alternatif, par exemple un lien vers le document, ou une image à la place d'une animation.

Exemples:

Incorporer un document HTML

```
<object data="data/test.html" type="text/html" width="300" height="200">  
  alt : <a href="data/test.html">test.html</a>  
</object>
```

Incorporer un document pdf

```
<object data="data/test.pdf" type="application/pdf" width="300" height="200">  
  alt : <a href="data/test.pdf">test.pdf</a>  
</object>
```

Vous pouvez spécifier des paramètres relatifs au document par l'intermédiaire de la balise `param`. IE a parfois besoin du paramètre `src` pour localiser correctement la source.

Incorporer un document wav

```
<object type="audio/x-wav" data="data/test.wav" width="200" height="20">  
  <param name="src" value="data/test.wav">  
  <param name="autoplay" value="false">  
  <param name="autoStart" value="0">
```

```
alt : <a href="data/test.wav">test.wav</a>  
</object>
```

L'objet event

I. Propriétés et méthodes générales

Description

Un objet de type événement est passé en argument à la fonction. Plusieurs propriétés décrivent cet objet et son état. On peut les utiliser, notamment, afin de déterminer d'où est issu l'événement, et à quelle étape précise de son « voyage » il en est (montée ou descente...). Il est également possible de l'intercepter, et de faire en sorte qu'il cesse sa propagation.

Nom de la propriété	Description
bubbles	Un booléen indiquant si l'événement remonte l'arborescence ou non.
cancelable	Un booléen indiquant si l'événement <i>peut</i> être annulé.
currentTarget	Le nœud auquel est affecté le gestionnaire d'événement.
eventPhase	Un entier indiquant l'étape où l'événement se trouve dans le flot : il vaut soit CAPTURING_PHASE(1) , soit AT_TARGET(2) , soit BUBBLING_PHASE(3) .
target	Le nœud d'où est parti l'événement.
timeStamp	L'heure à laquelle l'événement a eu lieu.
type	Une chaîne de caractères donnant le type d'événement, comme " mouseover " ou " click ", etc.
Nom de la méthode	Description
preventDefault()	Peut être utilisé pour annuler l'événement. Cela empêche le navigateur de procéder à l'action par défaut pour l'événement, comme par exemple charger une URL quand un lien a été cliqué. Attention, l'événement va continuer sa propagation le long de l'arbre.
stopPropagation()	Coupe le flot de l'événement. Cette méthode peut être utilisée lors de la phase de remontée ou de descente.

Événements souris

a. Propriétés

Les événements souris sont :

- **click**
- **dblclick**
- **mousedown**
- **mouseup**

- **mouseover**
- **mouseout**
- **mouseleave**
- **mousemove**
- **select** qui est activé lors de la sélection de texte

Nom de la propriété	Description
altKey, ctrlKey, metaKey, shiftKey	Booléens. Ils valent true si la touche correspondante était enfoncée quand l'événement souris a été activé.
button	Un entier indiquant quel bouton de souris était enfoncé : 0=gauche, 1=milieu, 2=droit. Une souris à un seul bouton (type Mac) ne retournera que la valeur 1 ; une souris à deux boutons retournera les valeurs 0 ou 2.
clientX, clientY	Donne les coordonnées du pointeur de la souris quand l'événement a eu lieu, par rapport à la fenêtre du navigateur effectivement disponible (hors barre de défilement).
relatedTarget	Sur un mouseover , cette propriété indique le nœud que vient de quitter la souris ; sur un mouseout , il s'agit du nœud sur lequel la souris vient de se placer. Cette propriété n'est pas supportée par Internet Explorer
screenX, screenY	Donne les coordonnées du pointeur de la souris quand l'événement a eu lieu, par rapport à l'écran.

Remarques

Les événements souris sont toujours attachés à l'élément de plus bas niveau dans l'arborescence. Par exemple, cela sera le lien si l'utilisateur a cliqué sur un lien inclus dans un paragraphe.

Lors du clic, trois événements ont lieu, toujours dans le même ordre : **mousedown**, **mouseup** et **click**. Ces événements sont toujours traités successivement, le traitement de chacun d'eux ne pouvant débuter que si celui du précédent est terminé.