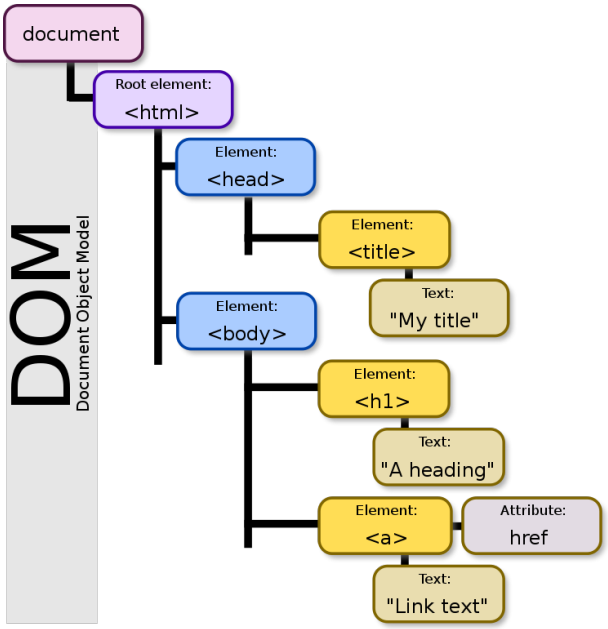
La gestion d'évènements avec Javascript

Table des matières

1) Fonctionnement d'un navigateur web

L'un des principaux attraits de l'interpréteur Javascript d'un navigateur est la possibilité de modifier dynamiquement et en temps réel le rendu d'une page web.

Le navigateur commence par télécharger le fichier HTML correspondant à l'URL visitée. Une fois recupérée, la page est transformée en une structure de donnée d'arbre.



Une fois l'arbre construit, ses noeuds (les éléments du document HTML) sont parcourus, les règles CSS sont appliquées et le rendu graphique de la page est effectué. Dans un même temps, les scripts Javascript présents dans la page sont exécutés. L'une de leur activité principale est la définition de gestionnaires d'évènements.

Ces gestionnaires d'évènements peuvent modifier librement l'arbre représentant le document. Ajouter ou supprimer des nœuds dans l'arbre correspond à ajouter ou supprimer les balises correspondant dans le document (encore une fois le fichier original, qu'il soit stocké localement ou sur un serveur n'est pas modifié). Les attributs des balises peuvent aussi être modifiés. Parmi ces attributs, la modification de l'attribut style permet de changer le rendu graphique de l'élément. Ces changements sont immédiatement répercutés sur l'affichage.

2) Premier programme

2.1) Codes

.html

```html
<html>
<head>
    <title>Age</title>
</head>
<body>
  <button onClick="suivant();">Clic</button>
  <span id="valeur">0</span>
  <script src="evenmt1.js" ></script>
</body>
</html>
```

.js

```javascript
let compteur=0;
function suivant(){
  compteur=compteur+1;
  let v=document.getElementById("valeur");
  v.innerHTML=compteur;
}
```

Exercice 1 : Observer et modifier ce code. Commenter le code correspondant.

2.2) Description du programme

Le script commence par définir une variable compteur initialisée à 0. Il définit ensuite une fonction `suivant()`. Lorsque elle est appelée, cette dernière commence par incrémenter la variable `compteur`. L'instruction suivante récupère un objet correspondant à l'élément de la page dont l'`id` vaut valeur. Cela correspond à l'élément `<span id="valeur">` se trouvant après le bouton. Une fois cet élément récupéré, il est possible d'écraser le contenu textuel (initialement à 0) par une nouvelle valeur. Toutes ces modifications sont reflétées directement dans l'affichage de la page, où l'on peut voir directement la valeur du compteur augmenter sans que la page ne se recharge. Par contre, si on recharge la page, alors le programme recommence du début.

3) Les évènements

3.1) La syntaxe

Les événements sont des actions de l'utilisateur, qui vont pouvoir donner lieu à une interactivité. L'événement par excellence est le clic de souris, car c'est le seul que le HTML gère. Grâce au Javascript il est possible d'associer des fonctions, des méthodes à des événements tels que le passage de la souris au-dessus

d'une zone, le changement d'une valeur, ...

La syntaxe d'un gestionnaire d'événement est la suivante :

```
onEvenement="Action_Javascript_ou_Fonction()";
```

Par exemple, pour créer un évènement clic de souris dans un bouton, on a utilisé l'instruction :

```
<button onClick="suivant();">Clic</button>
```

Mais l'attribut `onclick` n'est pas spécifique aux boutons. En effet, on peut créer un tel évènement sur n'importe quel élément HTML (`<p>` et `</p>` par exemple...).

Exercice 2 :

Réaliser un premier événement souris. Pour cela, vous réaliserez un page web minimale avec deux paragraphes et quand vous cliquerez sur un paragraphe, une nouvelle fenêtre apparaîtra et affichera 'mettre de la couleur'

3.2) Différents types d'évènements

Dans l'exemple précédent, le script contenu dans le Javascript sera exécuté quand l'utilisateur cliquera sur le bouton grâce à l'attribut `onClick`.

De la même manière, on pourrait ajouter des attributs `onKeyPress` (pression sur une touche du clavier) ou `onMouseover` (la souris survole l'élément) ou tout autre attribut d'évènement parmi les évènements récapitulés dans le tableau 1

Commande	Description
Load	chargement d'une page
Click	Clic avec la souris, sélection d'un élément
Dblclick	Double-clic sur l'élement
Mouseover	Entrée du pointeur de la souris sur un élément
Mouseout	Sortie du pointeur de la souris sur un élément
Keypress	Appui sur une touche produisant un caractère
Submit	Envoi d'un formulaire

Tableau 1 – Évènements

Exercice 3 : réaliser un bouton qui renvoie vers un site

3.3) Un exemple d'utilisation de fonction avec un bouton

Nous allons intégrer à notre page un bouton qui va nous permettre de fermer la page par un ou deux clics.

Cela nous permettra de voir un exemple avec une structure conditionnelle "if".

Exemple : pour commencer, intégrer cette fonction dans votre .js

```
        function quit()
        {
                if (confirm("Tu veux vraiment partir ? :"))
                {
                        alert("à bientôt !");
                        close();
                }
        }
```

Puis ajouter la ligne suivante dans votre fichier html.

```
        <button onclick="quit()">Tester le bouton</button>
```

Exercice 4 : prenez le temps d'observer et de commenter ces scripts.

3.4) Fonction et l'instruction querySelector()

La méthode `querySelector()` sélectionne le premier élément dans le document correspondant au sélecteur - ou groupe de sélecteurs - spécifié(s), ou null si aucune correspondance n'est trouvée.

Exemple créer une fonction dans votre fichier .js comme celle-ci :

```
function text_rouge(){
        let paragraphe=document.querySelector("p");
        paragraphe.style.color="red";
}
```

et modifier le programme html de la manière suivante :

```
<p onclick="text_rouge()">Je veux du rouge !</p>
```

Remarque : nous pouvons sélectionner le contenu d'un attribut div. Exemple, pour sélectionner l'élément *d* HTML (`div="d"`) dans un script Javascript, on peut utiliser *querySelector(#d)*. Nous pouvons aussi sélectionner le contenu d'un attribut class. Exemple, pour sélectionner l'élément *d* HTML (`class="d"`) dans un script Javascript, on peut utiliser *querySelector(.d)*

3.5) Lignes à tester

- changement de la couleur de fond de la page :

```
document.body.style.background="red";
```

- réinitialisation de la couleur par défaut :

```
document.body.style.background="";
```

- récupération d'un élément *d*. Cette instruction qui peut remplacer querySelector doit être faite une fois pour que d soit définie, avant de tester les instructions suivantes.

```
let d=document.getElementById("mondiv");
```

- changement de style de l'élément *d* :

```
d.style.color="blue";
d.style.fontWeight="bold";
d.style.border="1 pt dashed green"
```

- Changement de contenu de l'élément *d* :

```
d.innerHTML="Salut, tu vas bien";
```

Exercice 5 : tester les différentes instructions précédentes dans des fonctions.

Exercice 6 :

Réaliser petit programme de conversion de dollars en euros. Pour le réaliser, pour vous allez :

- réaliser une page HTML qui aura deux paragraphes ;
- créer un bouton (qui s'appellera *conversion*) dans cette page html qui lorsqu'on clique dessus, appelle une fonction ;
- Cette fonction convertira la somme en euros et affichera la somme dans le fichier HTML à l'intérieur d'un paragraphe.

4) Bonnes pratiques

Cependant, même si cette technique fonctionne et reste utilisée, elle tend fortement à disparaître. En effet, une séparation nette entre HTML, CSS et Javascript est prônée. Le code de l'exemple précédent devient ainsi :

.html

```
<html>
        <head>
        <title>Age</title>
</head>
<body>
        <button id="bouton">Clic</button>
        <span id="valeur">0</span>
        <script src="evenmt1.js" ></script>
</body>
</html>
```

.js

```
let compteur=0;
function suivant(){
        compteur=compteur+1;
        let v=document.getElementById("valeur");
        v.innerHTML=compteur;
}

let b=document.getElementById("valeur");
b.addEventListener("click",suivant);
```

Le premières lignes du script sont identiques aux précédentes. Les deux dernières sont intuitives. On récupère l'objet correspondant au bouton grâce à son `id`, puis on lui ajoute le gestionnaire d'évènements suivant.