

Annexe : Introduction au JavaScript

1 Exemple

Le code JavaScript peut être écrit en ligne, ou à l'aide d'un fichier externe.

```
<script>                                <script src="prog.js"></script>
    alert("Bonjour!");                  avec le fichier prog.js contenant :
</script>                                alert("Bonjour!");
```

2 Commandes de base

Les commentaires sont définis par // et /* ... */.

Les instructions se terminent avec un point virgule;

Les variables sont déclarées avec le mot clé **let**.

Les constantes ne peuvent être initialisées qu'une seule fois.

Les tests et les boucles s'écrivent comme en PHP (**if**, **while**, ...)

Il existe plusieurs fonctions que nous pouvons utiliser avec les chaînes de caractère :

Par exemple,

```
let test="Bonjour les RT1!!!"
alert(test.length); // 19
alert(test.substring(12,15)); //RT1
```

Les tableaux sont définis avec des indices entiers partant de 0 et pouvant avoir des valeurs de différent type.

```
let formations = ["RT", "GIM", "QLIO"];
alert(formations[0]); // RT
formations[formations.length]="GEA"; // ["RT", "GIM", "QLIO","GEA"];
```

Les fonctions s'écrivent comme en PHP avec la syntaxe suivante :

```
function essaifonction(param1,param2,...){
    ...
}
```

3 BOM (Browser Object Model)

Nous pouvons interagir avec le navigateur en utilisant des fenêtres, des minuteurs ou des boîtes de dialogue.

Boîtes de dialogue :

```
alert ('Bonjour! Comment allez-vous?');
if (confirm("Etes-vous certain?"))
    alert("Oui.");
else
    alert("En fait non ...");

let Prenom = prompt("Votre Prénom?", "Toto");
if (Prenom!= null)
    alert("Bienvenue " + Prenom);
```

Minuteurs :

```
function Message () {
    alert("Bonjour");
}

setTimeout(Message, 1000);
setInterval(Message, 5000);
```

4 DOM (Document Object Model)

Le DOM est une représentation de la page web sous forme d'arborescence d'objets.
Nous pouvons sélectionner différents éléments HTML, changer leur style et leur contenu.

Exemple 1 :

```
<p id="x">Ceci est un test pour le DOM.
<p>Un autre paragraphe de test.
<script>
  // Sélection d'un élément unique, par son id
  let p=document.getElementById ("x");
  // Changer le style d'éléments HTML
  p.style.color="blue";
  p.style.backgroundColor "red";
  p.innerHTML='Paragraphe modifié';
</script>
```

Exemple 2 :

```
<ul>
  <li> Ligne 1
  <li> Ligne 2
  <li> Ligne 3
</ul>
<script>
  // Sélection d'éléments, par leur balise
  let l=document.getElementsByTagName ("li");
  // Changer le style d'éléments HTML
  for (let i=0;i<l.length;i++)
    l[i].style.color = "green";
</script>
```

Il est également possible d'ajouter ou d'enlever des éléments HTML.

Nous pouvons ajouter un texte.

```
let nouveautexte = document.createTextNode("Nouveau texte");
document.body.appendChild(nouveautexte);
```

Nous pouvons ajouter un tableau.

```
let tableau=document.createElement("table");
for (let i=1;i<=10;i++) {
  let ligne=document.createElement("tr");
  for (let j=1;j<=10;j++) {
    let cellule=document.createElement("td");
    cellule.appendChild(document.createTextNode(i*j));
    ligne.appendChild(cellule);
  }
  tableau.appendChild(ligne);
}
document.getElementById("x").appendChild(tableau);
```

Les **événements** sont provoqués soit par l'utilisateur, ou soit par le navigateur (en utilisant des minuteurs).

Les exemples d'événements sont nombreux et différents suivant le type d'éléments HTML : fenêtre (onload, onresize), formulaire (onchange, onfocus, onblur, onsubmit, clavier (onkeypress), souris (onclick, ondblclick, onmouseover/onmouseenter), vidéo (onplay, onpause, onended).

Nous pouvons insérer ces événements dans les attributs d'éléments (DOM niveau 0) :

```
<p onClick="alert('Attention !')"> ... </p>
```

Nous pouvons utiliser des scripts pour gérer ces événements (DOM niveau 1) :

```
<p id='para'>
<script>
  document.getElementById('para').onclick=fonction;
  window.onload=fonction;
</script>
```

Si la fonction à exécuter lors d'un événement retourne la valeur faux, l'action n'est pas suivie.

```
<script>
  function validation (form) {
    let a=form.age.value;
    if (a!= "" && a>=0 && a<=120)
      return true;
    return false;
  }
</script>

<form onsubmit='return validation(this)''>
  Votre âge: <input type="text" name="age">
  <input type='submit'>
</form>
```

En plus de l'objet `window` on a aussi accès à un objet `document` qui permet de récupérer des éléments HTML et de les manipuler.

On partira systématiquement de cet objet afin de sélectionner nos éléments.

```
document.body // Récupère l'élément body

document.getElementById('demo') // Sélectionne l'élément avec l'id demo

document.querySelector('.demo') // Sélectionne le premier élément correspondant au sélecteur CSS
```