
TUTORIEL JAVASCRIPT

Javascript est un langage de script (interprété au fur et à mesure, ce n'est pas un programme à part entière). Il est supporté par les navigateurs web récents. Il peut donc être exécuté lors de la navigation sur les pages web par la plus part des personnes.

Une grande partie du javascript est comprise par les navigateurs. Cependant, chaque navigateur propose des fonctionnalités qui leur sont propres, notamment pour la gestion de la souris. Ceci rend difficile la réalisation de pages web incluant du javascript fonctionnant sur tous les navigateurs, quelque soit le système d'exploitation.

Il existe heureusement des éléments fonctionnant à peu près partout. Ce sont donc ceux-ci que nous allons découvrir, petit à petit. Dans l'ordre, nous verrons : le principe sommairement d'un fichier html, comment insérer un peu de javascript, comment javascript manipule les pages web et comment réaliser de petite animation en javascript.

Notons que les seuls outils dont vous ayez besoin sont : un navigateur web qui interprète le javascript et dont l'interprétation est activée (ce qui est le cas par défaut des navigateurs comme firefox, internet explorer, safari, opéra, google chrome, etc.) et d'un éditeur de texte.

INSERER DU JAVASCRIPT DANS UNE PAGE HTML

MISE EN PLACE DE VOS PAGES WEB

Dans votre répertoire `www/ext/technoWeb`, créez un répertoire « tutojs ». Récupérez le fichier [premierePage.html](#) et placez-le dans ce répertoire « tutojs ».

Une fois recopiée dans le répertoire, la page est accessible via le web par l'adresse « `http://kheops.unice.fr/~login/ext/technoWeb/tutojs/premierePage.html` ».

INSERTION DE JAVASCRIPT

Nous allons voir comment ajouter du javascript dans un page.

Le code javascript peut être écrit dans la page elle-même grâce à la balise « `<script type="text/javascript">` » et « `</script>` ». Cette balise peut être placée dans l'entête ou dans le corps de la page. Il peut y avoir autant de ces balises que nous le souhaitons dans une page. Mais attention, une balise « `<script type="text/javascript">` » et « `</script>` » ne peut pas contenir une autre balise « `<script type="text/javascript">` » et « `</script>` ».

Pour commencer, nous allons faire apparaître une petite fenêtre (*popup*) avec du texte. Copiez dans le presse papier (ctrl+C ou menu édition / copier) le code suivant :

```
<script type="text/javascript">  
alert("bonjour et bienvenue sur ma page ! ");  
</script>
```

Collez le dans le fichier premierPage.html dans l'entête de la page (entre la balise « <head> » et « </head> »). Ouvrez votre fichier dans votre navigateur. Qu'observez vous ? Voyez vous le reste du document avant de cliquer sur « ok » ?

Le code javascript s'exécute au fur et à mesure qu'il est lu dans la page. Déplacez ce morceau de code dans le corps de la page :

- d'abord entre la ligne « `<p>ceci marque le début de la page</p>` » et la ligne « `<h1>insertion de javascript</h1>` », rechargez votre page, que constatez-vous ?
- puis juste avant la ligne « `<p>ceci marque la fin de la page</p>` », rechargez votre page, que constatez-vous ?
- et finalement juste après la ligne « `<p>ceci marque la fin de la page</p>` », rechargez votre page, que constatez-vous ?

La fonction javascript « `alert` » est l'une des fonctions qu'offre javascript. Nous l'utilisons ici car son effet est immédiat. Il est bien sûr possible de faire beaucoup plus. Par exemple, nous pouvons faire des calculs. Nous allons définir une fonction qui calcule la circonférence d'un cercle et l'utiliser avec différentes valeurs de rayons.

Dans l'entête du fichier premierePage.html, copiez le code suivant :

```
<script type="text/javascript">  
function circonference(rayon)  
  {  
    return 2*Math.PI*rayon;  
  }  
</script>
```

Cette fonction prend un paramètre, rayon, et donne le résultat de la multiplication de ce rayon par 2π . La valeur de π est donnée par la constante (incluse dans javascript) « `Math.PI` », la multiplication est simplement notée par le symbole « `*` ».

Recopier le code suivant n'importe où dans le corps de la page :

```
<script type="text/javascript">  
alert("la circonference d'un cercle de rayon 3m est de "+circonference(3)+ " m");  
</script>
```

Le signe « `+` » sert ici à concaténer deux chaînes de caractères. Une chaîne de caractère, c'est du texte (manipulée par un programme). La concaténation, c'est simplement ajouter du texte à un autre.

Rechargez votre page et observez le résultat.

Nous allons maintenant appeler cette fonction grâce à une variable, « `x` ». Si vous ne savez pas ce qu'est une variable, souvenez-vous des représentations graphiques des fonctions que vous avez pu faire au lycée. En abscisse, vous aviez traditionnellement `x` qui variait et en ordonnée `y` qui était le

résultat de la fonction. Dans un tel cas, x est une variable, car nous faisons changer sa valeur entre une valeur de début et une de fin. x pouvait alors prendre n'importe quelle valeur entre ses deux bornes. Ici, nous allons faire varier x entre 0 et 5, en augmentant x de 1 à chaque fois. Recopier le code suivant dans le corps de la page premierePage.html :

```
<script type="text/javascript">  
x = 0;  
alert("la circonference d'un cercle de rayon "+x+"m est de "+circonference(x)+ "m");  
x = 1;  
alert("la circonference d'un cercle de rayon "+x+"m est de "+circonference(x)+ "m");  
x = 2;  
alert("la circonference d'un cercle de rayon "+x+"m est de "+circonference(x)+ "m");  
x = 3;  
alert("la circonference d'un cercle de rayon "+x+"m est de "+circonference(x)+ "m");  
x = 4;  
alert("la circonference d'un cercle de rayon "+x+"m est de "+circonference(x)+ "m");  
x = 5;  
alert("la circonference d'un cercle de rayon "+x+"m est de "+circonference(x)+ "m");  
</script>
```

Rechargez la page et observez le résultat.

En fait, les langages de programmation, javascript inclus, permettent de faire évoluer les valeurs et enchaîner les valeurs plus facilement que de recopier plusieurs fois la même ligne de code qui varie peu. Le code suivant est équivalent au code précédent :

```
<script type="text/javascript">  
for(x = 0; x <= 5; x = x+1)  
{  
    alert("la circonference d'un cercle de rayon "+x+"m est de "+circonference(x)+ "m");  
}  
</script>
```

Traduit en français, ces instructions javascript donne : « pour (« for ») chaque x allant de 0 à 5 (inclus, car « <= » signifie inférieur ou égale), x prenant (signe « = ») l'ancienne valeur de x plus 1 ; on affiche (« alert ») le calcul de la circonférence du cercle de rayon x ». Ceci s'appelle une boucle « for ».

Remplacez le dernier code que vous avez inclus et remplacez le par son équivalent avec la boucle « for ». Rechargez la page et observez le résultat. Modifiez la partie « x = x + 1 » par « x = x + 0.5 » et observez le résultat. C'est plutôt pratique, non ?

Nous avons vu comment mettre du javascript dans la page web. Cependant, à chaque fois, nous avons utilisé une fenêtre autre que la page. Maintenant, nous allons voir comment insérer le javascript dans la page elle-même.

JAVASCRIPT ET LA PAGE WEB

DOCUMENT.WRITE : ECRITURE DE MORCEAU DE HTML EN JAVASCRIPT

Il est possible d'écrire du html via javascript. A quoi cela sert-il ? A gérer du contenu qu'il est impossible de mettre à jour continuellement, comme la date et l'heure ou à générer du contenu qu'il serait pénible (copier coller répétitif) à écrire en html.

Récupérez la page [deuxiemePage.html](#), placez la dans votre répertoire « tutojs ». Pour commencer, ajoutons la date et l'heure entre les balises fermantes « </h1> » et « </body> » :

```
<script type="text/javascript">
time = new Date();
document.write("<p>Nous sommes le "+time.getDate()+" / "+(time.getMonth()+1)+"</p>");
document.write("<p>Il est "+time.getHours()+" : "+time.getMinutes()+"</p>");
</script>
```

La variable « time » reçoit le résultat d'une fonction incluse dans javascript, « *new Date()* ». Ceci permet de créer un objet informatique contenant la date et l'heure. Pour récupérer les valeurs de cet objet, il faut lui demander. Par exemple, pour avoir les heures, nous lui demandons à travers une fonction « *getHours()* » qu'il nous fournit. Comme souvent en informatique, les objets parlent anglais et il nous faut leurs parler en anglais.

Rechargez la page et observez le résultat. Attendez une minute, par exemple en lisant ces quelques phrases et à nouveau, rechargez la page et observez le résultat.

Ainsi, la date et l'heure s'affichent sans que nous ayons à modifier le document. En fait, le document est un objet informatique fourni par javascript, qui nous permet de manipuler le document, c'est-à-dire la page web. Le document est aussi anglophone, aussi, lorsque nous voulons ajouter, écrire une phrase dans le document, nous lui demandons par la fonction « *document.write()* » en passant en paramètre ce que nous voulons écrire. Ceci marche très bien au chargement de la page (quand on ouvre la page ou quand on actualise la page).

Revenons à nos circonférences de cercles. Affichez ceci dans une fenêtre « popup » était assez pénible, car d'une part, il fallait cliquer plusieurs fois sur ok et d'autre part le résultat n'était plus accessible une fois la page chargée. Avec « *document.write* », nous allons pouvoir ajouter les résultats dans la page. Pour cela, recopiez le code javascript suivant dans l'entête (entre les balises « <head> » et « </head> » :

```
<script type="text/javascript">
function circonference2(rayon)
{
    circ = 2*Math.PI*rayon;
    document.write("<p>La circonference du cercle de rayon "+rayon+"m est de
"+circ+"m.</p>");
    return;
}
</script>
```

Et recopiez le code suivant dans le corps de la page, entre « <body> » et « <h1> » :

```
<script type="text/javascript">
for(x = 0; x <= 5; x = x+0.5)
  {
    circonference2(x);
  }
</script>
```

La fonction « *circonference2* » est une variante de la fonction précédente. Elle calcule la circonférence, mais au lieu de retourner le résultat, elle écrit un paragraphe dans le document. De fait, elle ne retourne aucune valeur. Ensuite, nous appelons cette fonction plusieurs fois, grâce à l'instruction « pour (*for*) x variant... ».

Rechargez votre page *deuxiemePage.html* et observez le résultat.

Nous venons de le voir, « *document.write* » permet d'insérer dans la page, lors de son chargement, des éléments calculés. Cependant, cela ne permet pas une interaction plus conviviale une fois la page chargée. C'est pourquoi nous allons utiliser la notion d'événement.

JAVASCRIPT ET LES EVENEMENTS UTILISATEURS

Qu'est-ce qu'un événement ? C'est une action réalisée par l'utilisateur, la personne qui consulte la page web. Par exemple, nous pouvons vouloir déclencher une action, comme effectuer un calcul lorsque l'utilisateur clique sur un bouton.

Récupérer la page [troisiemePage.html](#) et placez là dans votre répertoire « tutojs ». Dans cette page, il y a un champ texte dans lequel on peut taper grâce au clavier du texte, en particulier un rayon. Il y a aussi un bouton qui servira de déclencheur au calcul.

Pour commencer, nous allons récupérer la valeur qui sera entrée dans le champ texte. Copiez le code javascript suivant dans l'entête de la page (entre les balises « <head> » et « </head> ») :

```
<script type="text/javascript">
function afficheChampDeTexte( )
  {
    champs = document.getElementById("rayon");
    alert("la valeur entree est : "+champs.value);
    return;
  }
</script>
```

Pour récupérer la valeur entrée dans le champ texte, nous utilisons une fonction que l'objet *document* nous fournit « *document.getElementById* ». « *id* » est un attribut de chaque balise html qui permet de nommer une des balises de la page. Cet « *id* » doit être unique, de fait, il ne reste plus qu'à demander au document quelle balise est nommée de telle ou telle façon pour l'obtenir.

La fonction « *afficheChampDeTexte* » récupère donc la valeur entrée puis l'affiche dans une fenêtre « *popup* ».

Il ne reste plus qu'à déclencher l'utilisation de cette fonction. Pour cela, nous allons utiliser un attribut particulier des balises html :

- « onclick » qui permet de déclencher une action lorsque l'utilisateur clique « sur » la balise (le texte contenu dans la balise)
- « onmouseover » qui permet de déclencher une action lorsque l'utilisateur fait passer la souris au dessus du texte contenu dans la balise
- « onmouseout » qui permet de déclencher une action lorsque l'utilisateur enlève la souris d'au dessus du texte contenu dans la balise.

Par exemple, nous allons ajouter ce code « html/javascript » `onclick="afficheChampDeTexte();"` dans la balise « `<input type="button" >` ». Rechargez votre page `troisiemePage.html`. Cliquez sur le bouton. Observez. Puis entrez un nombre dans le champ en haut de la page et recliquez sur le bouton. Finalement, tapez votre prénom dans le champ en haut de la page et recliquez sur le bouton.

Nous le voyons, que nous ayons tapé du texte ou non, que nous ayons tapé un nombre ou non, nous récupérons la valeur (vide ou non) de manière identique. Heureusement, javascript nous fournit une fonction « `isNaN()` (*is Not a Number*) » qui permet de savoir si son paramètre n'est pas un nombre ou si c'est un nombre.

Or, comme chaque langage de programmation impératif, javascript propose des structures de contrôle. Il s'agit en fait de faire des aiguillages ou des re-bouclages. L'instruction « `for` » en est une. L'instruction « `if(condition) {} else {}` » permet de faire un aiguillage. Par exemple, le matin, il m'est possible de consulter la météo. Et si la pluie est annoncée, alors je prends mon parapluie. Sinon, si la pluie n'est pas annoncée, alors je ne prends pas mon parapluie. Ce qui donnerait en mélangeant le français et la structure javascript : « `if (la pluie est annoncée) { je prends mon parapluie } else { je ne prends pas mon parapluie }` ».

Ainsi, nous pouvons tester si la valeur entrée dans le champ texte est un nombre ou non. Effacez tout le code javascript que vous insérez dans l'entête de la page `troisiemePage.html`. Remplacez le par le code suivant :

```
<script type="text/javascript">
function afficheChampDeTexte( )
{
  champs = document.getElementById("rayon");
  if (isNaN(champs.value))
  {
    alert("vous n'avez pas entre un nombre, je ne peux pas calculer la circonference !");
  }
  else
  {
    circonference = 2*Math.PI*champs.value ;
    alert("la circonference est : "+circonference+" m");
  }
  return;
}
</script>
```

Rechargez votre page `troisiemePage.html`. Cliquez sur le bouton. Observez. Puis entrez un nombre dans le champ en haut de la page et recliquez sur le bouton. Finalement, tapez votre prénom dans le champ en haut de la page et recliquez sur le bouton.

Cette fois, il n'y a que lorsqu'on ne tape rien dans le champ texte, c'est-à-dire un texte vide soit « "" » en javascript, que l'on calcule pour rien. Comme il est possible de mettre des structures de contrôle (« *for* », « *if* », etc.) dans des structures de contrôle, la solution consiste à utiliser un autre test avec l'instruction « *if* ». Remplacez la fonction « `afficheChampTexte` » par cette nouvelle version (sur la page suivante) :

```
<script type="text/javascript">
function afficheChampDeTexte( )
{
  champs = document.getElementById("rayon");
  if (isNaN(champs.value))
    {
      alert("vous n'avez pas tape un nombre, je ne peux pas calculer la circonference !");
    }
  else
    {
      if (champs.value != "")
        {
          circonference = 2*Math.PI*champs.value;
          alert("la circonference du cercle de rayon "+champs.value+" m est de
"+circonference+" m");
        }
      else
        {
          alert("vous n'avez rien tape, je ne peux pas calculer la circonference !");
        }
    }
  return;
}
</script>
```

Rechargez votre page `troisiemePage.html`. Cliquez sur le bouton. Observez. Puis entrez un nombre dans le champ en haut de la page et recliquez sur le bouton. Finalement, tapez votre prénom dans le champ en haut de la page et recliquez sur le bouton.

Comme précédemment, affichez le résultat, c'est déjà pas mal. Mais le résultat n'est plus consultable... alors nous pourrions vouloir utiliser « `document.write` ». Seulement, cette fonction offerte par javascript (à travers l'objet `document`) ne fonctionne « convenablement » qu'au chargement de la page. Essayez par vous-même en remplaçant « `alert` » par « `document.write` ». Rechargez votre page et essayez... tout disparaît sauf le résultat et on ne peut plus recalculer ou voir ce qu'il y avait préalablement.

Pour ajouter le résultat dans la page, il va falloir utiliser l'attribut des objets informatiques (javascript) `innerHTML`, ce que nous voyons dans la section suivante.

MODIFIER LA PAGE : L'ATTRIBUT INNERHTML

En effet, la propriété `innerHTML` des objets javascript représentant les balises html permet de manipuler le code html incluse dans ces balises.

Téléchargez la page [quatriemePage.html](#) et observez là dans votre navigateur ainsi que son code source. Vous y voyez une ligne, sur fond orange et encadrée de rouge, c'est un paragraphe (balise « `<p class="miseEnValeur">` » et « `</p>` ») qui est mis en forme avec du CSS. Nous allons essayer cette propriété `innerHTML` sur ce paragraphe. Dans cette balise paragraphe, ajoutez l'appel à une fonction javascript : `onclick="montreHtml(this);"`, c'est-à-dire après le « `<p class="miseEnValeur"` » et avant le « `>` », en laissant des espaces pour séparer les attributs. Le mot « *this* », ceci, permet de passer en paramètre le paragraphe lui-même, lorsqu'on clique dessus car avec l'attribut « `onclick` » qu'on déclenche l'appel.

De plus rajoutez le code correspondant à la fonction « *montreHtml* » dans l'entête de la page :

```
<script type="text/javascript">
function montreHtml(balise)
    {
        alert("le code html inclus dans la balise "+balise.tagName+" est
<<" + balise.innerHTML + ">>");
    }
</script>
```

L'attribut « *balise.tagName* » permet de connaître la balise html représenté par l'objet javascript.

Rechargez votre page et cliquez sur le paragraphe sur fond orange. Modifiez le texte de ce paragraphe. Par exemple, ajoutez la phrase « *Toutes les informations sur html sont sur la page* » w3c. Rechargez et recliquez. Observez les évolutions.

Cette fonction peut fonctionner avec n'importe quelle balise. Ajoutez `onclick="montreHtml(this);"` sur la première balise paragraphe du corps de la page, c'est-à-dire après le « `<p` » et avant le « `>` » en laissant un espace entre le « `p` » et le « `onclick` ». Rechargez et cliquez sur la première ligne de la page.

Vous l'avez constaté, « `innerHTML` » permet d'avoir accès au code html inclus dans la balise. Il s'agit en fait d'une chaîne de caractère. Et il est possible aussi de modifier ce code html. Par exemple, dans la fonction « *montreHTML* », ajoutez la ligne suivante :

```
balise.innerHTML = balise.innerHTML + " on a vu le code html " ;
```

Notez que nous utilisons ici la concaténation, c'est-à-dire l'ajout d'un texte à la suite d'un autre. Ainsi le code html présent avant le clic est conservé. Essayez à nouveau votre page et cliquez sur les différentes lignes de la page. Modifiez la ligne ci-dessus, par exemple en inversant de part et d'autre du « `+` » les chaînes de caractères « *balise.innerHTML* » et « *on a vu le code html* ». Observez le résultat dans votre navigateur.

Pour la suite, il est plus pratique de retirer le « `onclick` » sur le premier paragraphe de la page.

Pour en revenir à nos conférences, au lieu de faire une fenêtre pour afficher le résultat avec la fonction « *alert* » nous allons ajouter le résultat dans un paragraphe prévu à cet effet. Dans la page

web, il y a à la fin de la page, un paragraphe nommé (id) « resultats ». Nous allons adapter la fonction « *afficheChampDeTexte* » (et en profiter pour lui changer de nom) avec notre nouvel outil « *innerHTML* ». Dans l'entête de la page *quatriemePage.html*, ajoutez ceci :

```
<script type="text/javascript">
function afficheCirconference( )
{
    endroitOuEcrire = document.getElementById("resultats");
    champs = document.getElementById("rayon");
    if (isNaN(champs.value))
    {
        endroitOuEcrire.innerHTML = endroitOuEcrire.innerHTML + "vous n'avez pas tape
un nombre, je ne peux pas calculer la circonference !<br />";
    }
    else
    {
        if (champs.value != "")
        {
            circonference = 2*Math.PI*champs.value
            endroitOuEcrire.innerHTML = endroitOuEcrire.innerHTML + "la
circonference du cercle de rayon "+champs.value+" m est de "+circonference+" m.<br />";
        }
        else
        {
            endroitOuEcrire.innerHTML = endroitOuEcrire.innerHTML + "vous n'avez
rien tape, je ne peux pas calculer la circonference !<br />";
        }
    }
    return;
}
</script>
```

Rechargez la page. Le calcul ne se fait pas ? n'avez-vous rien oublié ? par exemple le déclenchement du calcul quand on clique sur le bouton ? (le code « `onclick="afficheCirconference();"` »).

Modifiez le code de la fonction « *afficheCirconference* » pour ne pas avoir de message d'erreur (quand on n'a pas tapé un nombre) dans la page. Par exemple en utilisant la fonction « *alert* » au lieu de modifier le code html...

Voilà, nous avons obtenu une page dont le contenu évolue en fonction des actions de l'utilisateur. Vous venez de manipuler dynamiquement une page web.

Cependant, avec javascript, il est possible de faire bien plus...

POUR ALLER PLUS LOIN : UNE HORLOGE

Dans cette partie, en guise d'exercice, nous allons faire une petite animation à travers un texte qui va se mettre à jour dans la page toutes les secondes. Notre objectif est de réaliser une petite horloge en texte.

MODIFIER L'HEURE APRES UN CLIC SUR UN BOUTON

Dans un premier temps, nous allons simplement mettre à jour l'heure quand on clique sur le bouton en « bas » de la page web [cinquiemePage.html](#) que vous allez récupérer et placer dans votre dossier « tutojs ».

Complétez le code javascript de la fonction « *rafraichirHorloge* » pour que l'heure soit rafraichie dans la page lorsque vous cliquez sur le bouton. Pour cela, inspirez-vous de la façon dont est écrite l'heure au chargement de la page (c.f. « *document.write* »), récupérez le paragraphe nommé "horloge" (c.f. « *document.getElementById* ») et utilisez l'attribut « *innerHTML* ».

MODIFIER L'HEURE AUTOMATIQUEMENT

Avant de pouvoir réaliser une mise à jour automatique de l'heure, nous avons besoin de connaître deux fonctionnalités de javascript (voire de html) supplémentaires :

- D'une part, il existe un déclencheur automatique à la fin du chargement de la page web dans le navigateur. Il s'agit de l'attribut « *onload* » de la balise « *<body>* » et « *</body>* » et uniquement pour cette balise. Cet attribut « *onload* » fonctionne comme « *onclick* », sauf que son déclenchement n'est pas un clic souris, mais comme écrit précédemment, le chargement de la page web.
- D'autre part, la fonction « *setTimeout* » permet de différer l'exécution d'un code javascript. Par exemple « `setTimeout("alert('coucou')", 1000);` » permettra d'ouvrir une fenêtre contenant le message « coucou » au bout d'une seconde (1000 est exprimé en millisecondes). Par exemple « `setTimeout("rafraichirHorloge()", 1000);` » permet d'appeler la fonction « *rafraichirHorloge* » au bout d'une seconde.
Une fonction peut se rappeler elle-même de la sorte.

Modifiez donc la balise ouvrante « *<body>* » et la fonction « *rafraichirHorloge* » pour que la mise à jour soit automatique.

Vous y êtes arrivé ? très bien. Vous êtes arrivé à la fin de ce tutoriel. Sachez qu'il existe encore beaucoup de facette de javascript, notamment tout ce qui est lié au document object model, à xml et à l'ajax.

Mais avant de vous laissez tranquille, que se passe-t-il maintenant si vous cliquez sur le bouton ? Qu'en concluez-vous ?