Les cahiers d'Exercices en Programmation: Le langage Javascript

Apprenez et entraînez vos acquis

- De très nombreux exercices à réaliser par vous-même

- Les corrigés sont présents dans cet ouvrage et expliqués Pas à Pas.

Professeur : Bart Kimps

AVANT-PROPOS

Ce livre est un cahier d'exercices : il vous propose des énoncés d'exercices et leurs corrigés. Vous allez apprendre le logiciel en vous entrainant à travers des exercices regroupés par thème.

Chaque énoncé vous présente l'exercice à réaliser. Vous trouverez à la fin du cahier le corrigé de chaque exercice. Certaines explications peuvent-être présentes.

METHODOLOGIE

Lors de la réalisation des exercices, vous pourrez remédier à certain problème à l'aide des corrections à la fin du cahier.

Après avoir réalisé tous les exercices de chaque chapitre vous allez pouvoir vérifier les compétences acquises à l'aide du tableau des objectifs.

Celui-ci sert à la cotation du professeur (grille d'évaluation).

Des **légendes ou recommandations** peuvent être présentes dans certains exercices. Celles-ci vous aideront dans vos recherches. Elles ne doivent pas être reproduites dans votre travail.

Chaque point de matière acquis dans un exercice peut être utilisé dans des exercices suivants sans explication.

Table des matières

Avant-propos

Bienvenue dans Javascript. Dans ce cours tu apprendras à programmer en Javascript, le langage du Web.

Javascript est un langage de programmation utilisé partout. Les navigateurs web tels que Chrome, Firefox et Internet Explorer s'en servent tous. Avec la puissance de Javascript, les développeurs web peuvent transformer leurs plus simples pages web en véritables applications interactives et en jeux.

Javascript peut aussi être exécuté à partir d'un serveur web pour créer des sites web complets. Il peut même être utilisé pour contrôler des robots et autres machines.

Des fichiers du code des jeux et autres exemples sont disponibles sur le serveur ou PC du professeur.

Amuse-toi bien ?

Une dernière chose à ne pas oublier : amuse-toi ! La programmation peut être une activité amusante et créative. Tout comme dessiner ou jouer. Une fois que tu auras compris comment coder, ta seule limite sera ton imagination.

Bienvenue dans le monde fantastique de la programmation informatique – j'espère que tu vas t'éclater.

Chapitre 1 : Fondamentaux

1. Qu'est-ce que le Javascript ?

Les ordinateurs sont des machines incroyablement puissantes, capables d'accomplir des exploits tels que participer à des concours d'échecs, publier des milliers de pages web ou réaliser des millions de calculs complexes en un temps record.

Mais, en réalité, les ordinateurs sont bêtes : ils exécutent uniquement ce que les humains leur ordonnent de faire.

Pour transmettre des ordres, nous utilisons des programmes informatiques. Sans eux, ils ne peuvent rien faire.

2. A la rencontre de Javascript

Les programmes informatiques sont écrits dans des langages de programmation, tels que Javascript.

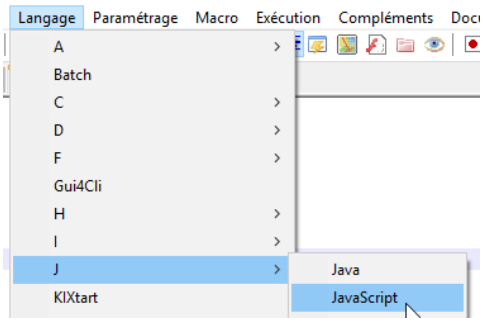
Le Javascript sert à écrire des programmes qui sont exécutés dans des pages web. Il permet de contrôler l'apparence de ces pages et de les faire réagir lorsqu'un visiteur clique sur un bouton ou bouge sa souris.

3. Pourquoi apprendre le Javascript

Le Javascript n'est pas le seul langage de programmation mais il est facile et amusant à apprendre et la meilleur raison est que pour écrire et exécuter des programmes Javascript il te faut uniquement un navigateur web.

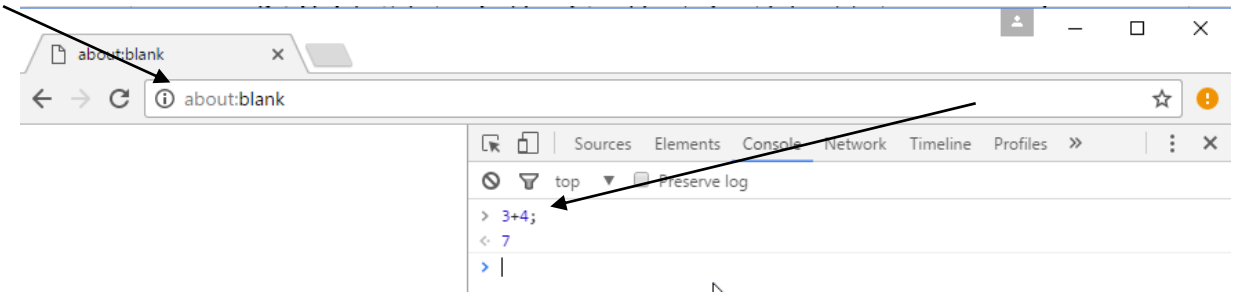
4. Écrire du Javascript

Dans Notepad++ ou Jedit, avant d'introduire du code, n'oubliez pas avant, de définir le langage utilisé



À partir de Google Chrome pour de petits programmes à tester :

Tape about:blank dans la barre d'adresse. Et ensuite appuie simultanément sur les touches CTRL + Maj + J sous Windows et Linux et sur les touches cmd + Alt + J. Une fenêtre apparaîtra. Teste en introduis ce petit calcul.



Teste aussi les opérations mathématiques suivantes :

100 -17; 123*456; 1234 + 57 * 3 – 31:

8/(1+3);

Donc, quand je ne le dis pas utilise TOUJOURS un éditeur comme Notepad++, Jedit, …

Chapitre 2 : Introduction au Javascript

Objectif(s) des exercices de ce chapitre.

À la fin de ce chapitre, l'élève sera capable de :

Créer un exercice en Javascript
Utiliser et comprendre la commande <button>
Utiliser et comprendre l'évènement OnClick
Afficher un message dans un navigateur : Commande Alert

Maintenant que tu maîtrises les notions de base en HTML (utiles pour choisir ce que tu veux mettre sur une page web), ajoutons du code en Javascript. Ce langage permet d'interagir avec la page, en cliquant sur un bouton par exemple. Il peut s'utiliser avec HTML.

1.Clique ICI

Ouvre une nouvelle page web dans ton éditeur de texte. Saisis le code ci-dessous.

N'oublie pas de définir le langage utilisé pour ces exercices : Javascript

Mais enregistre ton programme au format HTML et nomme-le Ex1.

```
1  <html>
2      <button>Clique ici</button>
3  </html>
```

Double-clique sur ce fichier pour le tester.

Essaie de cliquer sur le bouton … Il ne se passe rien ? Nous devons dire au bouton ce qu'il doit faire quand on clique dessus ! Pour cela, allez voir le chapitre suivant.

Code Javascript	Description	Exemple
<Button>	Création d'un bouton	<button>Clique ici</button>

2. Donner une action à un bouton

Il faut ajouter ce que l'on appelle un « listener », qui exécutera du code Javascript lorsqu'un évènement particulier se produit.

Nous allons utiliser un listener « onclick ».

Code Javascript	Description	Exemple
Onclick= ' '	Evènement particulier se produit lors du clique sur le bouton	<button onclick='alert("Bonjour")'
Alert	Indique au navigateur d'afficher un message	''alert("Bonjour")'

On peut remarquer que l'on utilise des guillemets simples autour du Javascript

Crée un nouvel exercice avec ce code.

```
1  <html>
2      <button onclick='alert("Bonjour")'>Clique ici</button>
3  </html>
```

Enregistre ton programme au format HTML et nomme-le Ex2.

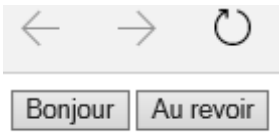
Crée une nouvelle page web nommée Ex3.

Saisis ce code :

```
1  <html>
2      <button>Bonjour</button>
3      <button>Au revoir</button>
4  </html>
```

Double-clique sur le fichier pour l'ouvrir.

Tu devrais voir à peu près ceci :



Pour fonctionner, les boutons ont besoin de code.

Donc, il y a deux « listeners de clics » : un pour chaque bouton.

Modifie le code précédent comme ceci :

```
<html>
    <button onclick='alert("Bonjour")'>Bonjour</button>
    <button onclick='alert("Au revoir")'>Au revoir</button>
</html>
```

Bien faire attention entre les guillemets simples et doubles.

Les guillemets simples sont autour du Javascript.

Les guillemets doubles sont autour du message.

Enregistre ton travail Ex4 et teste-le.

3. Exercice :

Ajoute un troisième bouton dans cet exercice. Lors du clique sur celui-ci, il doit afficher : Il faut grandir.

Sauvegarde ton travail sous le nom Ex5.

4. Solution :

```
<html>
    <button onclick='alert("Bonjour")'>Bonjour</button>
    <button onclick='alert("Au revoir")'>Au revoir</button>
    <button onclick='alert("Il faut grandir")'>Il faut grandir</button>
</html>
```

5. Introduire du code Javascript dans une page Web

Il faut utiliser l'élément <Script> dans le Body de la page

Exercice-toi avec la phrase habituelle à afficher Hello World !

N'oublie pas ici de créer une page en langage HTML.

Voici le code :

Hello World.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Hello World!</title>
</head>

<body>
    <script>
        alert('Hello world!');
    </script>
</body>
</html>
```

6. Lien vers un fichier js externe

Créer deux fichiers : le premier en js et le second en HTML dont voici le code à introduire :

Fichier HTML nommé « Page Maître.html »

```
<!DOCTYPE html>
<html>
<head>
    <title>Hello World!</title>
</head>

<body>
      <Script src="hello.js"></script>
</body>
</html>
```

Fichier Javascript nommé « hello.js »

```
alert('Hello world!');
```

Chapitre 3 : Boucles, variables et Javascript

Objectif(s) des exercices de ce chapitre.

À la fin de ce chapitre, l'élève sera capable de :

Réaliser des opérations et afficher le résultat à l'écran
Afficher des valeurs à l'écran avec un espace entre
Déclarer une variable
Assigner une valeur à une variable
Créer une boucle For
Arrêter une boucle For quand …
Déclarer et faire croître une variable
Confirmer une action
Boucle while

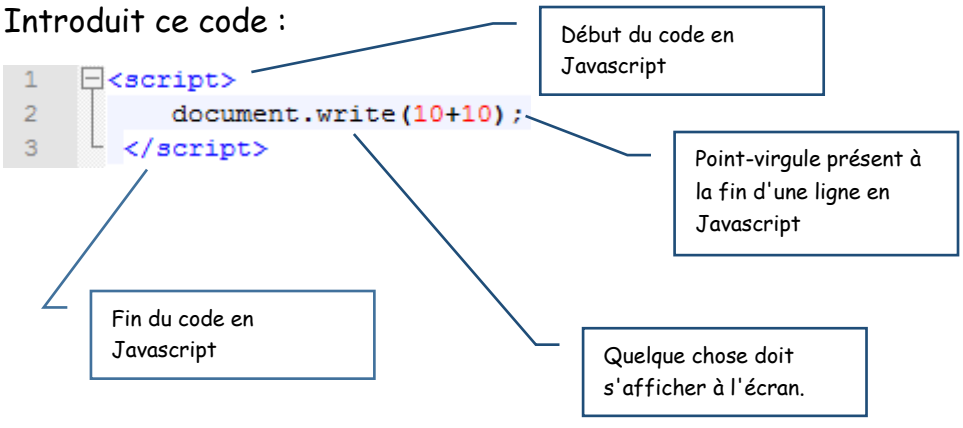
Si tu as codé sous le langage Scratch, tu connais déjà les boucles et les variables.

Une boucle est une séquence de commandes que l'on veut que l'ordinateur répète. Une variable est une valeur stockée par l'ordinateur. Tu vas découvrir comment utiliser ces techniques en Javascript.

1. Additions

Ouvre une nouvelle page Web nommée Ex6.

Introduit ce code :



2. Exercices

Ex7 : Additionner 50 et 40.

Ex8: Soustraire 25 à 80.

3. Solutions

Ex7 :

```
<script>
    document.write(50+40);
</script>
```

Ex8 :

```
<script>
    document.write(80-25);
</script>
```

4. Boucles et répétitions

Tu peux utiliser des boucles pour afficher quelque chose à l'écran de manière répétée. Pour cela, il faut coder une boucle "for". Modifie ton code ainsi. Vérifie que tu as bien saisi le code. Voici ce qui doit s'afficher : 123456789.

Ex9 :

```
<script>
    for(var n=1;n<10;n++)
    document.write(n);
</script>
```

Cela crée une boucle utilisant une variable appelée n, qui commence à 1. Elle augmente de 1 à chaque boucle, et sa valeur s'affiche à chaque fois sur l'écran. La boucle se poursuit ainsi jusqu'avant 10.

Change la commande write(n) en writeln(n).

Que se passe-t-il ?

Pour la suite des exercices, tu utiliseras la commande writeln().

5. La fonction Confirm

Son utilisation est simple : on lui passe en paramètre une chaîne de caractères qui sera affichée à l'écran et elle retourne un booléen en fonction de l'action de l'utilisateur.

Essayer ce code. Il peut-être ajouté dans une page existante.

```
<!DOCTYPE html>
<html>
<head>
    <title>Hello World!</title>
</head>

<body>
      <Script>
        if (confirm('Voulez-vous exécuter le code JavaScript de cette page ?'))
                {
                alert('Le code a bien été exécuté !');
                }

      </script>
</body>
</html>
```

6. Exercices

Ex10 : Créer une boucle utilisant une variable appelée n, qui commence à 10. Elle augmente de 1 à chaque boucle, et sa valeur s'affiche à chaque fois sur l'écran. La boucle se poursuit ainsi jusqu'avant 20.

Ex11 : Créer une boucle utilisant une variable appelée n, qui commence à 20. Elle augmente de 1 à chaque boucle, et sa valeur s'affiche à chaque fois sur l'écran. La boucle se poursuit ainsi jusqu'avant 40.

Ex12 : Créer une boucle utilisant une variable appelée n, qui commence à 1. Elle augmente de 1 à chaque boucle, et sa valeur soustrait le nombre 10. La réponse de la soustraction est affichée à chaque fois sur l'écran. La boucle se poursuit ainsi jusqu'avant 10.

Ex13 : Créer une boucle utilisant une variable appelée n, qui commence à 20. Elle diminue de 1 à chaque boucle, et sa valeur s'affiche à chaque fois sur l'écran. La boucle se poursuit ainsi jusqu'avant 0.

Ex14 : Crée une boucle qui compte de 1 à 100.

Ex15 : Crée une boucle qui compte de 1 à 1000.

7. Solutions

Ex10 :

```
<script>
    for(var n=10;n<20;n++)
    document.writeln(n);
</script>
```

Ex11 :

```
<script>
    for(var n=20;n<40;n++)
    document.writeln(n);
</script>
```

Ex12:

```
<script>
    for(var n=1;n<10;n++)
    document.writeln(10-n);
</script>
```

Ex13:

```
<script>
    for(var n=20;n>0;n--)
    document.writeln(n);
</script>
```

Ex14 :

```
<script>
    for(var n=1;n<101;n++)
    document.writeln(n);
</script>
```

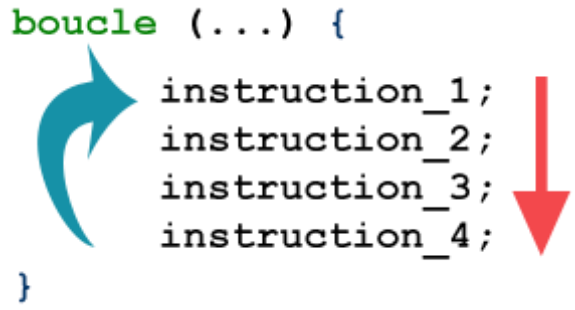
Ex15 :

```
<script>
    for(var n=1;n<1001;n++)
    document.writeln(n);
</script>
```

8. La boucle While

La boucle while se répète tant que la condition est validée. Cela veut donc dire qu'il faut s'arranger, à un moment, pour que la condition ne soit plus vraie, sinon la boucle se répéterait à l'infini, ce qui serait fâcheux.

Une boucle est une structure analogue aux structures conditionnelles vues dans le chapitre précédent sauf qu'ici il s'agit de répéter une série d'instructions. La répétition se fait jusqu'à ce qu'on dise à la boucle de s'arrêter. À chaque fois que la boucle se répète on parle d'itération (qui est en fait un synonyme de répétition).



Créer un nouveau fichier HTML dans lequel nous allons insérer du Javascript

Ex16 :

```
<!DOCTYPE html>
<html>
<head>
    <title>While</title>
</head>

<body>
      <Script>
        var number = 1;
        while (number < 10)
                {
                number++;
                }

        alert(number); // Affiche : « 10 »

      </script>
</body>
</html>
```

Un autre exemple que tu vas essayer de comprendre :

Ex17 :

```
<!DOCTYPE html>
<html>
<head>
    <title>While</title>
</head>

<body>
      <Script>
        var nicks = '',
        nick,
        proceed = true;

        while (proceed)
        {
        nick = prompt('Entrez un prénom :');

                if (nick)
                {
                nicks += nick + ' '; // Ajoute le nouveau prénom ainsi qu'une espace juste après
                }
                else
                {
                proceed = false; // Aucun prénom n'a été entré, donc on fait en sorte d'invalider la condition
                }
        }

        alert(nicks); // Affiche les prénoms à la suite

      </script>
</body>
</html>
```

Explication : La variable proceed est ce qu'on appelle une variable témoin, ou bien une variable de boucle. C'est une variable qui n'intervient pas directement dans les instructions de la boucle mais qui sert juste pour tester la condition. Nous avons choisi de la nommer proceed, qui veut dire « poursuivre » en anglais.

À chaque passage dans la boucle, un prénom est demandé et sauvé temporairement dans la variable nick. On effectue alors un test sur nick pour savoir si elle contient quelque chose, et dans ce cas, on ajoute le prénom à la variable nicks. Remarquez que nous ajoutons aussi un espace, pour séparer les prénoms. Si par contre nick contient la valeur null— ce qui veut dire que l'utilisateur n'a pas entré de prénom ou a cliqué sur Annuler - on change la valeur de proceed en false, ce qui invalidera la condition, et cela empêchera la boucle de refaire une itération.

9. La boucle do while

La boucle do while ressemble très fortement à la boucle while, sauf que dans ce cas la boucle est toujours exécutée au moins une fois. Dans le cas d'une boucle while, si la condition n'est pas valide, la boucle n'est pas exécutée. Avec do while, la boucle est exécutée une première fois, puis la condition est testée pour savoir si la boucle doit continuer.

Voici la syntaxe d'une boucle do while :

```
do {
    instruction_1;
    instruction_2;
    instruction_3;
} while (condition);
```

On note donc une différence fondamentale dans l'écriture par rapport à la boucle while, ce qui permet de bien faire la différence entre les deux. Cela dit, l'utilisation des boucles do while n'est pas très fréquente, et il est fort possible que vous n'en ayez jamais l'utilité car généralement les programmeurs utilisent une boucle while normale, avec une condition qui fait que celle-ci est toujours exécutée une fois.

Pas d'exercice.

10. À retenir

Code en Javascript	Description	Exemple
document.write()	Afficher quelque chose à l'écran	document.write(40-25)
Document.writeln()	Afficher quelque chose à l'écran avec un espace entre chaque valeur	
Var n	Déclarer une variable n	
For(var n=1;	Faire débuter une variable à 1	
For(var n=1;n<10;	Faire arrêter la boucle avant 10	
For(var n=1;n<10;n++)	n++ fait croître n	
While () {inst, inst2,…}		

Chapitre 4 : Fonctions et Javascript

Objectif(s) des exercices de ce chapitre.

À la fin de ce chapitre, l'élève sera capable de :

Créer un quiz sans fonction
Créer un quiz avec une fonction
Comprendre et créer une fonction
Comprendre et savoir utiliser une condition

Tu viens d'apprendre comment utiliser des boucles pour répéter des lignes de code avec Javascript. Tu auras parfois besoin de répéter seulement une partie de ton code, avec différentes valeurs. Pour cela, tu dois créer tes propres commandes, alors appelées "fonctions".

1. Faire un Quiz

Créons maintenant un quizEx18. Pour cela, il faut écrire du code pour poser une question, puis vérifier si la réponse est bonne ou mauvaise, et en informer le joueur. Voici à quoi cela peut ressembler :

```
1  <script>
2      var answer=prompt("Combien font 5 x 5?");
3      if (answer=="25") alert("Bien joué");
4      else alert("Faux");
5
6      var answer=prompt("Combien font 10 x 10?");
7      if (answer=="100") alert("Bien joué");
8      else alert("Faux");
9
10     var answer=prompt("Combien font 3 x 3?");
11     if (answer=="9") alert("Bien joué");
12     else alert("Faux");
13 </script>
```

Cette ligne réclame une entrée et stocke la réponse dans une variable appelée **answer** ("réponse")

Cette ligne vérifie si la réponse est correcte et affiche "Bien joué" si c'est le cas.

Si la réponse est incorrecte, le programme affiche "Faux"

Cela fonctionne mais chaque question requiert 3 lignes de code. Il faut faire plus simple

2. Utiliser une fonction dans un Quiz

Au lieu de répéter 3 lignes de code pour chaque question, tu peux créer une fonction ask ("demander"). Cette fonction sera "appelée" ou "exécutée" à chaque fois que l'on veut poser une question.

Ex17 :

```
<script>
    function ask(question,correct)
    {
        var answer=prompt(question);
        if(answer==correct)alert("Bien joué");
        else alert("Faux");
    }
    ask("Combien font 5 x 5?","25");
    ask("Combien font 10 x 10?","100");
    ask("Combien font 3 x 3?","9")
</script>
```

La ligne commence par définir la fonction **ask**.

La fonction utilise le même code qu'avant mais au lieu d'expression comme "Combien font 3 X 3 ?", elle se sert d'une variable appelée **answer**.

Ces lignes posent les questions. Elles "appellent" la fonction et lui transmettent les valeurs à utiliser pour chaque question et réponse.

Veille à bien saisir les { et }.

3. Exercice

Création d'une page Web nommée Ex19.

Crée un Quiz sur 5 pays d'Europe : La Belgique, l'Italie, l'Angleterre, Le Portugal et l'Espagne. Et demande leur capitale respective.

4. Solution

```
<script>
    function ask(question,correct)
    {
        var answer=prompt(question);
        if(answer==correct)alert("Bien joué");
        else alert("Faux");
    }
    ask("Qu'elle est la capitale de la Belgique ?","Bruxelles");
    ask("Qu'elle est la capitale de l'Italie ?","Rome");
    ask("Qu'elle est la capitale de l'Angleterre ?","Londres");
    ask("Qu'elle est la capitale du Portugal ?","Lisbonne");
    ask("Qu'elle est la capitale de l'Espagne ?","Madrid");
</script>
```

5. À retenir

Vocabulaire	Description
Fonction	Séquence de commandes effectuant une tâche spécifique à chaque fois qu'elle est "appelée"

Code en Javascript	Description	Exemple
prompt	Réclamer une entrée au clavier	var answer=prompt("Combien font 5 x 5?");
If … else	Si …. ;Sinon …. ; Si la réponse est … alors ; sinon ….;	if (answer=="100") alert("Bien joué"); else alert("Faux");

Chapitre 5 : Fonctions en Javascript avec HTML

Objectif(s) des exercices de ce chapitre.

À la fin de ce chapitre, l'élève sera capable de :

Insérer du code Javascript dans une page Web : la balise ouvrante t fermante SCRIPT.
Changer la couleur du fond d'écran d'une page web à l'aide de boutons
Mettre en lien le choix d'une couleur à un bouton sur une page web
Comprendre et utiliser la fonction **setbg** "définir un arrière-plan"

Tu as vu comment te servir des fonctions Javascript pour écrire un programme simple et linéaire. Linéaire signifie que les fonctions se lancent une par une. Tu vas à présent découvrir comment lancer différentes fonctions lorsqu'on clique sur différents boutons.

1. Changer de couleur

Réalisons une page web qui change de couleur lorsqu'on clique sur certains boutons. Pour commencer, tu vas créer une page avec un seul bouton dessus, et une fonction pour en modifier la couleur. Pour finir, tu programmeras le bouton pour qu'il exécute la fonction lorsqu'on clique dessus.

Nom de l'exercice : Ex20.

```
1   <html>
2   <button onclick="red()">Couleur rouge</button>
3
4   <script>
5       function red()
6       {
7           document.body.style.backgroundColor="red";
8       }
9   </script>
10
11  </html>
```

À l'aide de cette balise, tout ce qui se trouve dedans est traité comme du code Javascript. Tout le contenu de cette balise est donc exécuté par l'interpréteur Javascript du navigateur.

2. Plus de couleurs, plus de fonctions ?

Ce serait vraiment super de pouvoir ajouter d'autres boutons pour choisir la couleur de la page. Mais cela demanderait d'ajouter autant de fonctions que de couleurs souhaitées. Au lieu de cela, tu peux créer une fonction générale "modifier la couleur de fond "appelée **setbg**(set pour "définir" et bg pour "arrière-plan"). On pourrait "appeler" ou "exécuter" la fonction ainsi: **setbg('red')** ou **setbg('blue')**.

Voici un nouvel exercice nommé : Ex21.

```
<html>
<button onclick="setbg('blue')">Couleur bleu</button>

<script>
    function setbg(col)
    {
        document.body.style.backgroundColor=col;
    }
</script>

</html>
```

Ajoutons une couleur, par exemple le vert.

```
<html>
<button onclick="setbg('blue')">Couleur bleu</button>
<button onclick="setbg('green')">Couleur vert</button>

<script>
    function setbg(col)
    {
        document.body.style.backgroundColor=col;
    }
</script>

</html>
```

3. Exercice

Dans ce même exercice, essaie d'ajouter deux couleurs liées bien sûr à deux autres boutons.

Jaune : Yellow

Rouge : red

4. Solution

Ex22 :

```
<html>
<button onclick="setbg('blue')">Couleur bleu</button>
<button onclick="setbg('green')">Couleur vert</button>
<button onclick="setbg('yellow')">Couleur jaune</button>
<button onclick="setbg('red')">Couleur rouge</button>

<script>
    function setbg(col)
    {
        document.body.style.backgroundColor=col;
    }
</script>

</html>
```

5. À retenir

Code en Javascript	Description	Exemple
setbg	Définir un arrière-plan	Setbg('red')
Document.body.style.backgroundColor	Arrière-plan d'une page web	

Chapitre 6 : Projet animaux

Objectif(s) des exercices de ce chapitre.

À la fin de ce chapitre, l'élève sera capable de :

Mettre en lien des pages web

La plupart des sites web comportent plus d'une page. Toutes les pages sont reliées entre elles : on peut ainsi naviguer de l'une à l'autre. Tu vas à présent créer un site simple sur le thème des animaux, en te servant uniquement d'HTML.

1. Relier des pages Web

Crée une nouvelle page Web et nomme-la Index.html et saisis le code suivant :

```
1  <html>
2      <h1>Animaux</h1>
3  </html>
```

Crée une nouvelle page Web et nomme-la Chiens.html et sais le code suivant :

```
1  <html>
2      <h1>Chiens</h1>
3  </html>
```

Modifie le code de ton fichier Index.html comme ci-dessous :

```
1  <html>
2      <h1>Animaux</h1>
3      <a href="Chiens.html">Chiens</a>
4  </html>
```

Actuellement tu peux ouvrir ta page Chiens.html à partir d'un lien situé dans ta page Index.html.

2. Exercices

Ajoute 2 autres animaux dans la liste des animaux. Crée donc 3 autres pages html avec le nom de ces animaux. Crée une référence <a Href=" "> vers ces documents dans ta page Index.

Crée un nouveau titre insecte dans ton fichier Index

Ajoute 2 insectes à la liste et crée donc 2 pages HTML avec le nom de ces insectes.

3. Solution

Je ne présente ici que la page Index et un exemple d'une autre page en lien.

Index.html :

```
<html>
    <h1>Animaux</h1>
    <a href="Chiens.html">Chiens</a>
    <a href="Chats.html">Chats</a>
    <a href="Chevaux.html">Chevaux</a>
    <h1>Insectes</h1>
    <a href="Abeilles.html">Abeilles</a>
    <a href="Mouches.html">Mouches</a>
</html>
```

Abeilles.html :

```
<html>
    <h1>Abeilles</h1>
</html>
```

4. Insérer une image/photo dans une page Web

Télécharge une photo au format jpg de par exemple un cheval.

Ajoute cette balise <img> afin d'afficher l'image.

```
<html>
    <h1>Chevaux</h1>
    <img src="cheval.jpg">
</html>
```

Tu peux aussi ajouter cette balise
 afin d'insérer une ligne d'espace entre les liens.

```
<html>
    <h1>Animaux</h1>
    <a href="Chiens.html">Chiens</a>
    <br>
    <a href="Chats.html">Chats</a>
    <br>
    <a href="Chevaux.html">Chevaux</a>
    <h1>Insectes</h1>
    <a href="Abeilles.html">Abeilles</a>
    <br>
    <a href="Mouches.html">Mouches</a>
</html>
```

5. À retenir

Code en html	Description	Exemple
<a href="">	Créer un lien hypertexte	Chiens
 	Passer à la ligne	
<img src=" ">	Mettre en lien une photo ou une image jpg	<img src="cheval.jpg">

Chapitre 7 : Développer un jeu : À la recherche du trésor Enfoui.

Objectif(s) des exercices de ce chapitre.

À la fin de ce chapitre, l'élève sera capable de :

Comprendre et savoir utiliser les deux outils : DOM et jQuery
Gestion d'évènement
Comprendre et utiliser l'élément img
Gestion de nombres aléatoires
Créer et utiliser des fonctions

Ce chapitre va te permettre d'apprendre énormément de choses par le biais d'un jeu. Tu vas découvrir l'élément img, qui permet d'insérer des images dans une page web. Tu vas mesurer des distances entre deux points à l'aide de Javascript.

Tu vas mettre en pratique des nouvelles compétences en gestion d'évènement en créant un jeu.

1. Le DOM et jQUERY

Pour augmenter la puissance du code Javascript, il existe deux outils : le DOM et jQUERY.

DOM : Document Object Mode ou modèle objet de document. Le DOM permet à Javascript d'accéder au contenu d'une page web afin de manipuler ses éléments de différentes manières.

jQuery : Outils servant à modifier plus facilement les éléments sélectionnés.

Comment charger jQuery dans la page HTML :

<script src=https://code.jquery.com/jquery-2.1.0.js></script>

Mais ou insérer cela ? Dans le corps "Body" de ta page HTML.

2. Concevoir le jeu

Avant de commencer à écrire le code du jeu, nous allons mettre en place sa structure.

Étapes à suivre pour concevoir le jeu :

1. Crée une page web contenant une image (la carte au trésor).
2. Sélectionner une position aléatoire sur l'image de la carte pour y cacher le trésor.
3. Crée un gestionnaire de clics. Chaque fois que le joueur clique sur la carte, voici ce que dois faire le gestionnaire de clics :
    - Ajouter 1 au compteur de clics.
    - Calculer la distance entre la position du clic et celle du trésor.
    - Afficher un message pour dire à l'utilisateur s'il chauffe ou s'il refroidit.
    - Féliciter le joueur s'il clique sur le trésor ou juste à côté. Et lui dire combien de clics il a dû effectuer pour retrouver le trésor.

3. Créer la page web en HTML

Nous allons voir le code HTML de la page du jeu. Pour la carte au trésor, nous utiliserons un nouvel élément appelé img (image) et pour afficher les messages au joueur, ce sera l'élément p.

Crée un nouveau fichier appelé tresor.html et saisis le code suivant :

Structure de la page web

```
<!DOCTYPE html>
<html>
<head>
    <title></title>
</head>

<body>
    <script>
    </script>

    <script>
    </script>
</body>
</html>
```

On introduit maintenant les données dans l'en-tête de la page web

```
<head>
    <title>A la recherche du trésor enfoui !</title>
</head>
```

Et ensuite on passe au corps de la page web

```
<body>
    <h1 id="titre"> A la recherche du tresor enfoui !</h1>

    <img id="carte" width=800 height=800 src="http://nostarch.com/images/treasuremap.png">

    <p id="distance"></p>
```

L'élément img sert à insérer des images dans un document HTML. Pas besoin ici de balise fermante.

id = définition de l'attribut d'une balise qui a souvent pour valeur un nom explicite. Id identifie des éléments HTML

Nous n'utiliserons pas ici une fonction DOM servant à récupérer un élément par l'ID : document.getElementById.
Par exemple : var elementTitre = document.hetElementById("titre").

Tu demandes au navigateur de rechercher dans la page l'élément HTML dont l'attribut id a pour valeur "titre".

width et height = Réglage de la largeur et de la hauteur de l'image. . Ici l'image est affichée avec 400 pixels de largeur et 400 pixels de hauteur. En fonction de la grandeur de votre écran, vous pouvez par exemple augmenter sa grandeur à du 800 X 800 pixels.

h1 = définition d'un titre de premier niveau.

src = signaler au document qu'elle image il doit afficher (src = source). Sa valeur est l'adresse web de l'image. Tu crées donc un lien vers cette image situé sur le site web de chez Eyrolles (www.editions-eyrolles.com/dl/0011850).

Juste après l'élément img, tu places un élément p vide. Il possède un attribut id dont la valeur est "distance". Par la suite tu te serviras de cet attribut id dans le code Javascript pour ajouter du texte dans cet élément. C'est de cette manière que tu précises au joueur à quel point il est près de trouver le trésor enfoui.

Note : j'ai augmenté la taille de l'image à 800 X 800.

On insère les deux outils DOM et jQuery :

```
<body>
    <h1 id="titre"> A la recherche du tresor enfoui !</h1>

    <img id="carte" width=800 height=800 src="http://nostarch.com/images/treasuremap.png">

    <p id="distance"></p>

    <script src="https://code.jquery.com/jquery-3.1.1.js"></script>

    <script>
```

4. Placer le trésor de manière aléatoire

Nous allons commencer à écrire le code Javascript du jeu. Entre les balises précédemment construites.

Sélectionnons une position aléatoire sur l'image de la carte afin d'y cacher le trésor.

Les dimensions de la carte sont de 800 X 800 pixels.

Les coordonnées du pixel le plus en haut à gauche sont {x : 0; y : 0}, et celles du pixel en bas à droite sont {x : 799, y : 799}.

Sélection des nombres aléatoires

Pour créer les coordonnées d'un point aléatoire sur la carte au trésor, il faut sélectionner :

- Un nombre aléatoire entre 0 et 799 pour les coordonnées x ;
- Un nombre aléatoire entre 0 et 799 pour les coordonnées y ;

Pour cela, tu dois initialiser une fonction qui prend en entrée un argument dimension. Elle sélectionne un nombre aléatoire situé entre 0 et dimension (non incluse) et le renvoie.

En dessous de la définition des deux outils, insère les lignes de codes dans la seconde partie du script.

```
<script src="https://code.jquery.com/jquery-3.1.1.js"></script>

<script>
    //Sélection d'un nombre alétoire
    var creerNombreAleatoire = function (dimension) {
        return Math.floor(Math.random() * dimension);
    };
```

Explications :

Sélection d'un nombre aléatoire en 0 et 1 : Math.random()
Multiplication de ce nombre par l'argument dimension
Arrondir le résultat de la multiplication au premier nombre entier inférieur : Math.floor

Quand tu appelleras la fonction creerNombreAleatoire(800), tu obtiendras un nombre aléatoire entre 0 et 799.

Déterminer les coordonnées du trésor

Utilisons la fonction précédemment créée.

```
<script>
    //Sélection d'un nombre alétoire
    var creerNombreAleatoire = function (dimension) {
        return Math.floor(Math.random() * dimension);
    };

    //Déterminer les coordonnées du trésor
    //Définition des variables
    var largeur = 800;
    var hauteur = 800;

    //Créer un objet appelé cible et lui définir des coordonnées
    var cible = {
    x : creerNombreAleatoire(largeur),
    y : creerNombreAleatoire(hauteur)
    };
```

Le gestionnaire de clics

Le gestionnaire de clics et la fonction qui est appelée chaque fois que le joueur clique sur la carte au trésor.

```
    //Créer un objet appelé cible et lui définir des coordonnées
    var cible = {
    x : creerNombreAleatoire(largeur),
    y : creerNombreAleatoire(hauteur)
    };

    //Ajouter un gestionnaire de clics à l'élément img
    $("#carte").click(function (evenement) {
        //Le code du gestionnaire de clics sera ici
    });
</script>
```

Explications :

$("carte") : sélection de la zone qui contient la carte au trésor (l'élément img correspondant possède un attribut id dont la valeur est carte).

Fonction du gestionnaire de clics :

- incrémenter le compteur de clics ;
- calculer la distance entre le clic du joueur et le trésor enfoui ;
- afficher des messages à l'utilisateur.

Compter les clics

Sa première tâche est de compter le nombre total de clics. Tu auras donc besoin d'une variable `clics`. La valeur de se compteur doit être initialisée pour chaque début de partie.

```
24          var largeur = 800;
25          var hauteur = 800;
26          var clics = 0;
```

Par la suite dans le gestionnaire de clics, la valeur devra s'incrémenter de 1 à chaque clic : `clics++`.

```
//Ajouter un gestionnaire de clics à l'élément img
$("#carte").click(function (evenement) {
    //le code du gestionnaire de clics sera ici.
    clicks++;
});
```

Calculer la distance entre un clic et le trésor

Pour savoir si le joueur est proche du trésor, il faut mesurer la distance entre la position du trésor enfoui et l'endroit où tu as cliqué.

Pour cela tu vas écrire une fonction calculerDistance en dessous de la fonction creerNombreAleatoire créée précédemment.

```
<script>
    //Sélection d'un nombre alétoire
    var creerNombreAleatoire = function (dimension) {
        return Math.floor(Math.random() * dimension;
    };

    //Calculer la distance entre le clic et le tresor enfoui
    var calculerDistance = function (evenement, cible) {
        var diffX = evenement.offsetX - cible.x;
        var diffY = evenement.offsetY - cible.y;
        return Math.sqrt((diffX * diffX) + (diffY * diffY));
    };
```

Monsieur Pythagore nous a aidés avec son beau théorème.

Signifier au joueur à quel point il est près de gagner

Maintenant que tu connais la position entre le clic du joueur et la position du trésor, tu peux donner un indice afin qu'il sache à quel point il est près du trésor.

Tu vas donc créer une nouvelle fonction : creerIndiceDistance.

```
<script>
    //Sélection d'un nombre alétoire
    var creerNombreAleatoire = function (dimension) {
        return Math.floor(Math.random() * dimension;
    };

    //Calculer la distance entre le clic et le tresor enfoui
    var calculerDistance = function (evenement, cible) {
        var diffX = evenement.offsetX - cible.x;
        var diffY = evenement.offsetY - cible.y;
        return Math.sqrt((diffX * diffX) + (diffY * diffY));
    };

    var creerIndiceDistance = function(distance){
        if (distance < 10) {
            return "Tu brûles !";
        } else if (distance < 20) {
            return "Tu chauffres vraiment !";
        } else if (distance < 40) {
            return "Tu chauffes un peu !";
        } else if (distance < 80) {
            return "Tu chauffes !";
        } else if (distance < 160) {
            return "Tu refroidis !";
        } else if (distance < 320) {
            return "Tu refroidis vraiment !";
        } else {
            return "Tu gèles";
        }
    };
```

La chaîne de caractères renvoyée dépend de la distance en pixels entre ton clic et le trésor enfoui.

Afin de voir le message, le code suivant se place dans le corps du gestionnaire de clics. Son rôle est de calculer la distance, de sélectionner la chaîne des caractères correspondante et de la montrer à l'utilisateur.

```
//Ajouter un gestionnaire de clics à l'élément img
$("#carte").click(function (evenement) {
    //le code du gestionnaire de clics sera ici.
    clicks++;

    // calculer la distance entre un évènement de clic et la cible
    var distance = calculerDistance(evenement, cible);

    // Convertir la distance en un indice au joueur
    var indiceDistance = creerIndiceDistance(distance);

    //Compléter l'élément #distance avec l'indice au joueur
    $("#distance").text(indiceDistance);
```

Comme tu peux le voir, tu commences par appeler la fonction calculerDistance et tu stockes son résultat dans la variable Distance.

Ensuite, tu passes cette distance en argument à la fonction créerIndiceDistance afin qu'elle sélectionne la chaîne de caractères qui correspond à la distance. Cette chaine de caractères est enregistrée dans la variable indiceDistance.

Avec la dernière instruction (celle après le $) :

- tu sélectionnes l'élément de la page web dont l'attribut id a pour valeur "distance" ;
- puis tu lui donnes pour contenu le texte de la variable indiceDistance.

<u>Vérifier si le joueur a gagné</u>

Le joueur aura gagné s'il clique à moins de 8 pixels de la position du trésor. Ceci est réalisé à l'aide de ce morceau de code :

```
            //Compléter l'élément #distance avec l'indice au joueur
            $("#distance").text(indiceDistance);

            //Si le clic est très proche de la cible, dire au joueur qu'il a gagné
            if (distance < 8) {
            alert("Tu as retrouvé le trésor enfoui en " + clics + " !");
            }
        });
    </script>

</body>
</html>
```

5. Solution complète

```
<!DOCTYPE html>
<html>
<head>
    <title> A la recherche du tresor enfoui !</title>
</head>

<body>
    <h1 id="titre"> A la recherche du tresor enfoui !</h1>

    <img id="carte" width=800 height=800 src="http://nostarch.com/images/treasuremap.png">

    <p id="distance"></p>

    <script src="https://code.jquery.com/jquery-2.1.0.js"></script>

    <script>
        //Sélection d'un nombre alétoire
        var creerNombreAleatoire = function (dimension) {
            return Math.floor(Math.random() * dimension);
        };

    //Calculer la distance entre le clic et le tresor enfoui
    var calculerDistance = function (evenement, cible) {
      var diffX = evenement.offsetX - cible.x;
            var diffY = evenement.offsetY - cible.y;
            return Math.sqrt((diffX * diffX) + (diffY * diffY));
    };

    var creerIndiceDistance = function(distance){
            if (distance < 10) {
                return "Tu brûles !";
            } else if (distance < 20) {
                return "Tu chauffres vraiment !";
            } else if (distance < 40) {
                return "Tu chauffes un peu !";
            } else if (distance < 80) {
                return "Tu chauffes !";
            } else if (distance < 160) {
                return "Tu refroidis !";
            } else if (distance < 320) {
                return "Tu refroidis vraiment !";
            } else {
                return "Tu gèles";
            }
        };

```

```
47      //Déterminer les coordonnées du trésor
48      //Définition des variables
49      var largeur = 800;
50      var hauteur = 800;
51      var clicks = 0;
52
53    //Créer un objet appelé cible et lui définir des coordonnées
54          var cible = {
55          x : creerNombreAleatoire(largeur),
56          y : creerNombreAleatoire(hauteur)
57          };
58
59    //Ajouter un gestionnaire de clics à l'élément img
60          $("#carte").click(function (evenement) {
61              //le code du gestionnaire de clics sera ici.
62              clicks++;
63
64              // calculer la distance entre un évènement de clic et la cible
65              var distance = calculerDistance(evenement, cible);
66
67              // Convertir la distance en un indice au joueur
68              var indiceDistance = creerIndiceDistance(distance);
69
70              //Compléter l'élément #distance avec l'indice au joueur
71              $("#distance").text(indiceDistance);
72
73              //Si le clic est très proche de la cible, dire au joueur qu'il a gagné
74              if (distance < 8) {
75              alert("Tu as retrouvé le trésor enfoui en " + clics + " !");
76              }
77          });
78      </script>
79  </body>
80  </html>
```

6. Défis de programmation

Diminuer la précision :

La précision que tu dois avoir rend vraiment le jeu difficile. Modifie-la pour te faciliter un peu la vie de joueur.

Pas de solution mais une aide car ce travail n'est pas trop difficile.

Tu vas devoir modifier les valeurs dans les conditions : les pixels.

```
var creerIndiceDistance = function(distance){
    if (distance < 10) {
        return "Tu brûles !";
    } else if (distance < 20) {
        return "Tu chauffres vraiment !";
    } else if (distance < 40) {
        return "Tu chauffes un peu !";
    } else if (distance < 80) {
        return "Tu chauffes !";
    } else if (distance < 160) {
        return "Tu refroidis !";
    } else if (distance < 320) {
        return "Tu refroidis vraiment !";
    } else {
        return "Tu gèles";
    }
};
```

Ainsi que la valeur des pixels dans cette condition :

```
//Si le clic est très proche de la cible, dire au joueur qu'il a gagné
if (distance < 8) {
alert("Tu as retrouvé le trésor enfoui en " + clics + " !");
}
```

Ajouter des messages :

Essaie d'ajouter 2 messages à afficher (tels que "très très très chaud", "hyper chaud"). Modifie les distances correspondantes afin de personnaliser ton jeu.

```
var creerIndiceDistance = function(distance){
    if (distance < 10) {
        return "Tu brûles !";
    } else if (distance < 20) {
        return "Tu chauffres vraiment !";
    } else if (distance < 40) {
        return "Tu chauffes un peu !";
    } else if (distance < 80) {
        return "Tu chauffes !";
    } else if (distance < 160) {
        return "Tu refroidis !";
    } else if (distance < 320) {
        return "Tu refroidis vraiment !";
    } else if (distance < 640) {
        return "Tu te transformes en pierre";
    } else if (distance < 700) {
        return "BRRRRRRRRRR !!!"
    } else {
        return "Tu gèles";
    }
};
```

Limiter le nombre de clics :

Insère dans ton jeu une limite sur le nombre de clics et affiche le message "Game Over" si le joueur dépasse par exemple le nombre de 10 essais.

Déclaration d'une variable

```
//Déterminer les coordonnées du trésor
//Définition des variables
var largeur = 800;
var hauteur = 800;
var clicks = 0;
var clickLimit = 10;
```

Condition gérant le nombre de clic

```
//Ajouter un gestionnaire de clics à l'élément img
    $("#carte").click(function (evenement) {
        //le code du gestionnaire de clics sera ici.
        clicks++;

    //Nombre de clic limité
    if (clics > clickLimit) {
        alert("GAME OVER");
        return;
    }

    // calculer la distance entre un évènement de clic et la cible
    var distance = calculerDistance(evenement, cible);
```

Affiche le nombre de clics disponibles :

Affiche une ligne de texte supplémentaire, juste après l'indice de distance. Tu y informes au joueur le nombre de clics qu'il dispose encore. Ainsi il saura s'il est sur le point de perdre.

```
<p id="distance"></p>
<p id="clicks-restant"></p>
```

```
      // Nombre de clics restant
      $("#clicks-restant").text(clickLimit - clicks);

      //Si le clic est très proche de la cible, dire au joueur qu'il a gagné
      if (distance < 8) {
        alert("Tu as retrouvé le trésor enfoui en " + clicks + " clicks!");
      }
    });
    </script>
</body>
</html>
```

Bibliographie

Apprendre à coder : La programmation facile niveau 4 : Vigot

Javascript pour les Kids – Eyrolles

Site OpenClassRoom