

Javascript

Partie 1

Javascript

2

Les bases du JavaScript

- JavaScript est un langage de programmation qui permet d'implémenter des mécanismes complexes sur une page web.
- À chaque fois qu'une page web fait plus que simplement afficher du contenu statique — afficher du contenu mis à jour à des temps déterminés, des cartes interactives, des animations 2D/3D, des menus vidéo défilants, etc... — JavaScript a de bonnes chances d'être impliqué.

Javascript

3

Les bases du JavaScript

Le JavaScript, créé en 1995 par Brendan Eich (pour la Netscape Communication Corporation), est un langage de programmation de scripts orienté objet.

Si le terme Java est commun au langage du même nom, le JavaScript est radicalement différent.

On crée une instruction Javascript à l'intérieur des balises `<script>`

`<script>instruction</script>`

Javascript

4

Les bases du JavaScript

- Le JavaScript, créé en 1995 par Brendan Eich (pour la Netscape Communication Corporation), est un langage de programmation de scripts orienté objet.
- Si le terme Java est commun au langage du même nom, le JavaScript est radicalement différent.
- On crée une instruction Javascript à l'intérieur des balises <script>

`<script>instruction</script>`

La syntaxe du JavaScript

- Les instructions doivent être séparées par un point-virgule que l'on place à la fin de chaque instruction :

```
<script>
```

```
    instruction 1;
```

```
    instruction 2;
```

```
    ...;
```

```
</script>
```

Javascript

6

Fonction JavaScript

- La syntaxe d'une fonction se compose de deux choses : son nom, suivi d'un couple de parenthèses (une ouvrante et une fermante). Entre les parenthèses se trouvent les arguments, que l'on appelle aussi paramètres.

<script>

myfunction();

</script>



```
<script>
```

```
    alert('Bonjour à tous');
```

```
</script>
```

Les fichiers JavaScript (js)

- Il est possible, et même conseillé, d'écrire le code Javascript dans un fichier externe, portant l'extension .js. Ce fichier est ensuite appelé depuis la page Web au moyen de l'élément `<script>` et de son attribut `src`.

- **Fichier hello.js**

```
<script>  
    alert('Bonjour à tous');  
</script>
```

- **Fichier html**

```
<script src="hello.js"></script>
```


Les variables

- Une variable consiste en un espace de stockage, qui permet de garder en mémoire tout type de données. La variable est ensuite utilisée dans les scripts.
- Une variable contient seulement des caractères alphanumériques, le \$ (dollar) et le _ (underscore) ; elle ne peut pas commencer par un chiffre ni prendre le nom d'une fonction existante de Javascript.

`<script>`

`var myVariable();`

`myVariable=2;`

`</script>`

`<script>`

`myVariable=2;`

`</script>`

Les variables

- Une variable peut être de type numérique, mais aussi une chaîne de caractères :

```
<script>
```

```
var text1='Bonjour tout le monde';
```

```
var text2='je m'appelle antoine';
```

```
</script>
```

- Une variable peut enfin être de type booléen (boolean), avec deux états possibles : vrai ou faux (true ou false).

Les opérateurs

- On peut utiliser 5 opérateurs arithmétiques : l'addition (+), la soustraction (-), la multiplication (*), la division (/) et le modulo (%). (Le modulo est le reste d'une division.)

```
<script>
var number1 = 3,
    number2 = 2, result;
result = number1 *
number2;
alert(result); //
Affiche : « 6 »
</script>
```

ou :

```
<script>
var number = 3;
number = number + 5;
alert(number); //
Affiche : « 8 »
</script>
```

qui équivaut à :

```
<script>
var number = 3;
number += 5;
alert(number); //
Affiche : « 8 »
</script>
```

Javascript

11

La concaténation

- Une concaténation consiste à ajouter une chaîne de caractères à la fin d'une autre

```
<script>
    var hi = 'Bonjour ', name =
'toi', result;
    result = hi + name;
    alert(result);
    // Affiche : « Bonjour toi »
</script>
```

ou :

```
<script>
    var text = 'Bonjour ';
    text += 'toi';
    alert(text);
    // Affiche « Bonjour toi ».
</script>
```

Javascript

12

La concaténation en JavaScript

- Une concaténation consiste à ajouter une chaîne de caractères à la fin d'une autre

```
<script>
  var hi = 'Bonjour ', name =
'toi', result;
  result = hi + name;
  alert(result);
// Affiche : « Bonjour toi »
</script>
```

ou :

```
<script>
  var text = 'Bonjour ';
  text += 'toi';
  alert(text);
// Affiche « Bonjour toi ».
</script>
```

```
<script>
  var text, number1 = 4, number2 = 2;
  text = number1 + ' ' + number2; /* on ajoute une chaîne de caractères vide
entre les deux nombres pour permettre une concaténation sans calcul */
  alert(text); // Affiche : « 42 »
</script>
```

Javascript

13

La concaténation

- Une concaténation consiste à ajouter une chaîne de caractères à la fin d'une autre

```
<script>
  var hi = 'Bonjour ', name =
'toi', result;
  result = hi + name;
  alert(result);
// Affiche : « Bonjour toi »
</script>
```

ou :

```
<script>
  var text = 'Bonjour ';
  text += 'toi';
  alert(text);
// Affiche « Bonjour toi ».
</script>
```

```
<script>
  var text, number1 = 4, number2 = 2;
  text = number1 + ' ' + number2; /* on ajoute une chaîne de caractères vide
entre les deux nombres pour permettre une concaténation sans calcul */
  alert(text); // Affiche : « 42 »
</script>
```


Javascript

14

La fonction prompt et concaténation

```
<script>
  var userName = prompt('Entrez votre prénom :');
  alert(userName); // Affiche le prénom entré par l'utilisateur
</script>
```

```
<script>
  var start = 'Bonjour ', name, end = ' !', result;
  name = prompt('Quel est votre prénom ?');
  result = start + name + end;
  alert(result);
</script>
```

Javascript

15

La fonction prompt et concaténation

```
<script>
var first, second, result;
first = prompt('Entrez le premier chiffre :');
second = prompt('Entrez le second chiffre :');
result = parseInt(first) + parseInt(second); /* La fonction parseInt()
transforme la chaîne de caractères en nombre */
alert(result);
</script>
```


Javascript

16

Les 8 opérateurs de comparaison

<code>==</code> : égal à	<code>===</code> : contenu <u>et</u> type de variable égal à	<code>></code> supérieur à	<code><</code> : inférieur à
<code>!=</code> : différent de	<code>!==</code> : contenu <u>ou</u> type de variable différent de	<code>>=</code> supérieur ou égal à	<code><=</code> : inférieur ou égal à

```
<script>
var number1 = 2, number2 = 2, number3 = 4, result;
  result = number1 == number2; // Au lieu d'une seule valeur, on en écrit
deux avec l'opérateur de comparaison entre elles
  alert(result); // La condition est donc vérifiée car les deux variables
contiennent bien la même valeur
  result = number1 == number3;
  alert(result); // La condition n'est pas vérifiée car 2 est différent de 4
  result = number1 < number3;
  alert(result); // La condition est vérifiée car 2 est bien inférieur à 4
</script>
```

Les opérateurs logiques

Il y en a 3 :

`&&` qui signifie ET avec par exemple : `valeur1 && valeur2`

Cet opérateur vérifie la condition lorsque toutes les valeurs qui lui sont passées valent `true`.

`||` qui signifie OU avec par exemple : `valeur1 || valeur2`

Cet opérateur est plus souple car il renvoie `true` si une des valeurs qui lui est soumise contient `true`, qu'importent les autres valeurs.

`!` qui signifie NON avec par exemple : `!valeur`

Cet opérateur se différencie des deux autres car il ne prend qu'une seule valeur à la fois. S'il se nomme « NON » c'est parce que sa fonction est d'inverser la valeur qui lui est passée, ainsi `true` deviendra `false` et inversement.

Les condition if – else

La condition est composé :

- de la structure conditionnelle if ;
- de parenthèses qui contiennent la condition à analyser, ou plus précisément le booléen retourné par les opérateurs conditionnels ;
- d'accolades qui permettent de définir la portion de code qui sera exécutée si la condition se vérifie.

Les condition if – else

La fonction `confirm()` permet une interaction de l'utilisateur à l'exécution du code (true si OK, false si Annuler) :

```
<script>  
if (confirm('Voulez-vous exécuter le code Javascript de cette page ?')) {  
    alert('Le code a bien été exécuté !');  
}  
</script>
```

Javascript

20

Les condition if – else

La structure else permet de simplifier l'alternative :

```
<script>
  if (confirm('Pour accéder à ce site vous devez être une fille, cliquez sur
"OK" si c\'est le cas.')) {
    alert('Vous allez être redirigé vers le site.');
```

```
  }
```

```
  else {
```

```
    alert("Désolé, vous n'avez pas accès à ce site.");
```

```
  }
```

```
</script>
```


Les condition if – else

La structure else permet de simplifier l'alternative :

```
<script>
var floor = parseInt(prompt("Entrez l'étage où l'ascenseur doit se rendre (de
-2 à 30) :"));
if (floor == 0) {
    alert('Vous vous trouvez déjà au rez-de-chaussée.');
```

```
} else if (-2 <= floor && floor <= 30) {
    alert("Direction l'étage n°" + floor + ' !');
```

```
} else {
    alert("L'étage spécifié n'existe pas.");
}
</script>
```

Les condition switch

Cette structure permet de gérer une courte liste de possibilités :

```
<script>
  var drawer = parseInt(prompt('Choisissez le tiroir à ouvrir (1 à 4) :')); //on
précise bien le type de la valeur, ici un nombre avec la fonction parseInt()
  switch (drawer) {
    case 1: // on pose chaque cas l'un après l'autre ; on met des apostrophes si
L'on vérifie des chaînes de caractères au lieu de nombres
      alert('Contient divers outils pour dessiner : du papier, des crayons, etc.');
      break; // on arrête la fonction pour passer à un autre cas
    case 2:
      alert('Contient du matériel informatique : des câbles, des composants, etc.');
      break;
    case 3:
      alert('Ah ? Ce tiroir est fermé à clé ! Dommage !');
      break;
    case 4:
      alert('Contient des vêtements : des chemises, des pantalons, etc.');
      break;
    default: // on pose une autre possibilité, pour gérer une erreur de
L'utilisateur
      alert("Info du jour : le meuble ne contient que 4 tiroirs et, jusqu'à preuve
du contraire, les tiroirs négatifs n'existent pas.");
  }
</script>
```


Les ternaires

Cette structure permet de simplifier certaines conditions :

```
<script>
  var startMessage = 'Votre genre : ',
      endMessage,
      adult = confirm('Êtes-vous une fille ?');
  endMessage = adult ? 'Fille' : 'Garçon';
  alert(startMessage + endMessage);
</script>
```

Exercice à réaliser :

- **Demander l'âge de la personne**
- **Fournir un commentaire selon l'âge de la personne.**
- **Vous devez fournir un commentaire sur 4 tranches d'âge qui sont les suivantes : Tranche d'âge**
- **Exemple de commentaire**
- **1 à 6 ans « Vous êtes un jeune enfant. »**
- **7 à 11 ans « Vous êtes un enfant qui a atteint l'âge de raison. »**
- **12 à 17 ans « Vous êtes un adolescent. »**
- **18 à 120 ans « Vous êtes un adulte. »**

Javascript

Fin de la présentation
Partie 1