JS variables,expressions, conversions de types, alert, prompt, console.log

Ingrid & Grégory
Promo Simplon 2018

Sommaire

Les variables JS

Une variable est un objet repéré par son nom, pouvant contenir des données, qui pourront être modifiées lors de l'exécution du programme.

Les noms de variables peuvent être aussi long que nécessaire, mais doivent répondre à certains critères :

· Un nom de variable doit commencer par une lettre (majuscule ou minuscule) ou un "_" .

· Un nom de variables peut comporter des lettres, des chiffres et les caractères _ et & (les espaces ne sont pas autorisés).

· Les noms de variables ne peuvent pas être des noms réservés (tel que *if*, *false*, *function*, *null*, *var*, *while*, etc).

Les noms de variables sont sensibles à la casse.

La déclaration de variables

La déclaration peut se faire de deux façons :

· Soit de façon explicite, en faisant précéder la variable du mot clé var qui permet d'indiquer de façon rigoureuse qu'il s'agit d'une variable :

```
var chaine= "bonjour"
```

· Soit de façon implicite, en laissant le navigateur déterminer qu'il s'agit d'une déclaration de variable.

Pour déclarer implicitement une variable, il suffit d'écrire le nom de la variable suivie du caractère = et de la valeur à affecter :

```
chaine= "bonjour"
```

Types de données dans les variables

En Javascript il n'est pas nécessaire de déclarer le type des variables, contrairement à des langages évolués tels que le langage C ou le Java pour lesquels il faut préciser s'il s'agit d'entier (*int*), de nombre à virgule flottante (*float*) ou de caractères (*char*).

Le Javascript n'autorise la manipulation que de 4 types de données :

- Des nombres : entiers ou à virgules.
- Des chaînes de caractères (string) : une suite de caractères.
- Des booléens : *true* et *false* qui permettent de vérifier une condition.
- Des variables de type *null* : un mot caractéristique pour indiquer que la variable ne contient aucune donnée.

La notion d'expression

Une expression est un morceau de code qui produit une valeur.

On crée une expression en combinant des variables, des valeurs et des opérateurs.

Toute expression produit une valeur et correspond à un certain type.

Le calcul de la valeur d'une expression s'appelle l'évaluation.

Lors de l'évaluation d'une expression, les variables sont remplacées par leur valeur.

Une expression peut comporter des parenthèses qui modifient la priorité des opérations lors de l'évaluation : en l'absence de parenthèses la priorité des opérateurs est la même qu'en mathématiques.

Expression : exemple

```
// c est une expression dont la valeur est le nombre 3
var c = 3;
// d est une expression dont la valeur est celle de c (ici 3)
var d = c;
// (d + 1) est une expression
// Sa valeur est celle de d augmentée de 1 (ici 4)
d = d + 1; // d contient la valeur 4
console.log(d); // Affiche 4
```

Conversions de types

Même si Javascript gère de façon transparente les changements de type des variables il est parfois nécessaire de forcer la conversion du type. Ainsi Javascript fournit deux fonctions natives permettant de convertir le type des variables passées en paramètre :

- *parseInt()* permet de convertir une variable en nombre.
- *parseFloat()* permet de convertir une variable en nombre décimal.

La syntaxe de la fonction *parseInt()* est la suivante :

```
parseInt(chaine[, base]);
```

La syntaxe de la fonction *parseFloat()* est la suivante :

```
parseFloat(chaine);
```

Exemple avec parseInt()

Soient les variables a et b :

```
var a = "123";
var b = "456";
```

Selon que l'on utilise la fonction *parseInt()* ou non, l'utilisation de l'opérateur + avec ces deux variables donnera un résultat différent :

```
document.write(a+b,"<BR>"); // Affiche 123456
document.write(parseInt(a)+parseInt(b),"<BR>"); // Affiche 579
```

Les boites de dialogue

Une boîte de dialogue est une fenêtre qui s'affiche au premier plan suite à un événement, et qui permet :

- Soit d'avertir l'utilisateur : Alert()
- Soit le confronter à un choix : confirm()
- Soit lui demander de compléter un champ pour récupérer une information : prompt()

La méthode Alert()

La méthode *alert()* permet d'afficher dans une boîte toute simple composée d'une fenêtre et d'un bouton *OK* et le texte qu'on lui fournit en paramètre. Dès que cette boîte est affichée, l'utilisateur n'a d'autre alternative que de cliquer sur le bouton OK.

Elle est est souvent utilisé pour souhaiter la bienvenue aux visiteurs ou encore indiquer une erreur dans la saisie d'un formulaire

LA méthode Alert()

Syntaxe : alert("message") ou alert(variable)

Exemple :

```
<script language="Javascript">

var temp = "Erreur!"
  alert(temp)
</script>
```

ce qui donnera la boite de dialogue :

La méthode confirm()

Elle est similaire à la méthode alert(), si ce n'est qu'elle permet un choix entre "OK" et "Annuler". Lorsque l'utilisateur appuie sur "OK" la méthode renvoie la valeur *true*. Elle renvoie *false* dans le cas contraire...

Syntaxe : confirm("message") ou confirm(variable)

On l'utilise généralement dans une condition.

Exemple :

```
if (confirm("êtes vous d'accord")) {
 alert("vous êtes d'accord")
}
else{
 alert("vous n'êtes pas d'accord")
}
```

La méthode prompt()

La méthode prompt est un peu plus évoluée que les deux précédentes puisqu'elle fournit un moyen simple de récupérer une information provenant de l'utilisateur, on parle alors de boîte d'invite.La méthode prompt() requiert deux arguments :

-le texte d'invite
-la chaîne de caractères par défaut dans le champ de saisie

syntaxe :
prompt('Posez ici votre question','chaîne par défaut')

prompt()

Cette boîte d'invite retourne la valeur de la chaîne saisie par l'utilisateur, elle retourne la valeur *null* si jamais aucun texte n'est saisi...

Fréquemment utilisée pour demander le nom d'un visiteur ou encore un mot de passe, cette méthode à l'avantage d'être bien sûr contextuelle et événementielle mais aussi le pouvoir de demander des données avant l'affichage de la page, données qui pourront être utilisées pour créer une page dynamique adapté aux choix du visiteur

prompt()

Valeur par défaut: Optionnelle, la valeur (caractère ou nombre) qui sera proposée par défaut au visiteur.

Si vous n'indiquez pas de valeur par défaut, aucune donnée ne sera proposée dans la boîte de saisie et la valeur sera <undefined>.

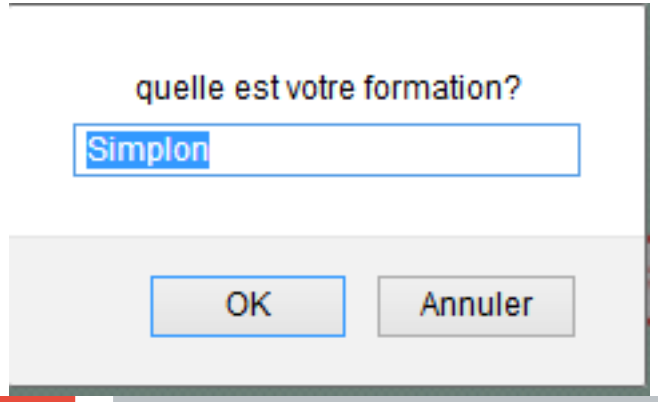
Si le visiteur annule (" cancel ") la saisie, la valeur retournée sera " null "

prompt()

exemple :

```
<script language="Javascript">
  nVarNom = prompt("Quelle est votre
formation?", "Simplon")
  alert(nVarNom)
</script>
```

ce qui donnera la boite de dialogue :



Console.log()

Une console de développement donne droit à des fonctions de log (journal) plus souples qu'un banal appel à alert() qui se révélera très souvent peu détaillé et bloquant.

La fonction la plus courante est sans conteste console.log() qui permettra d'afficher un message, une chaîne de texte, voire le contenu d'une variable (objet, tableau) de manière détaillée dans la console.

Utilisez console.log() au lieu de alert() pour afficher des informations pendant l'exécution de votre script : l'utilisateur ne verra rien et ne sera pas bloqué.

Console.log()

Syntaxe :

console.log(« chaine de caractère » ou variable)

Exemple :

```
<script language="Javascript">
nvarnom = prompt("quelle est votre formation?")
console.log(nvarnom)
</script>
```

Autres methodes console

Console.table() : pour mettre en forme des tableaux

Console.time() et console.timEnd() :

démarrent un chronomètre et le stoppent en affichant le résultat de la mesure. Très pratique pour mesurer le temps d'exécution d'un script.

Liste complète des methodes console :

https://developer.mozilla.org/fr/docs/Web/API/Console