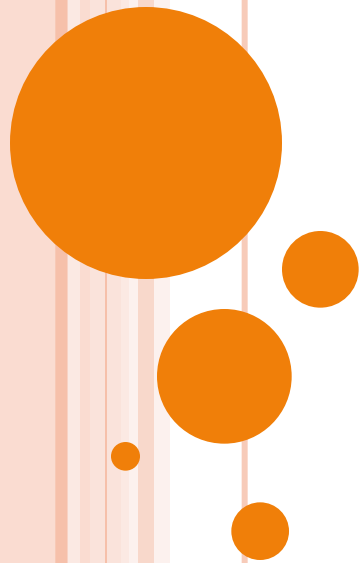


JAVASCRIPT



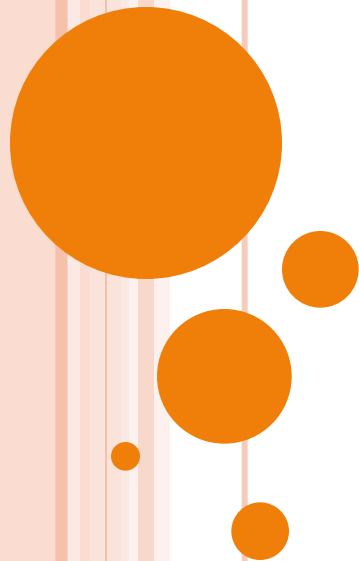
Pourquoi JAVASCRIPT?

introduire de l'interactivité avec les pages HTML

effectuer des traitements simples sur le poste de travail de l'utilisateur

Le Javascript est un langage de script simplifié orienté objet dont la syntaxe est basée sur celle du Java

JS contient des gestionnaires d'événement



L'encapsulation de code JavaScript

On peut placer du code JS :

- utiliser la balise `<SCRIPT>`
- utiliser un fichier externe `<SCRIPT SRC="fichier.js">`
- Associé au pseudo-protocole *javascript:* dans une URL
- Associé à une balise HTML qui gère un événement

JS :ENTRE LES BALISES

<SCRIPT> ET </SCRIPT>

<SCRIPT LANGUAGE="JavaScript">
instructions, déclarations de fonctions, etc..
</SCRIPT>

S'il est inclus dans la section <HEAD>, il n'est pas exécuté tout de suite.

Par contre, s'il fait partie du corps du document, il est immédiatement exécuté en même temps que le code HTML est interprété.

Il est nécessaire d'inclure les déclarations de fonctions dans la section <HEAD>..</HEAD>

JS: Associé à une balise HTML qui gère un événement

JS est généralement inséré sous forme d'un appel de fonction préalablement déclarée dans la partie HEAD

Exemple :

```
<INPUT TYPE="button" VALUE="Calculer"  
onClick="calculer()">
```

JS ASSOCIÉ AU PSEUDO-PROTOCOLE *JAVASCRIPT*

Exemple :

Pour calculer

Cette pseudo-URL permet de lancer l'exécution d'un script écrit en JS

L'opérateur VOID permet d'ignorer le paramètre de retour de la fonction

Premier programme

Pour créer ce programme, il faut :

- 1. Un navigateur Internet**
- 2. Un éditeur de texte**

Exemple :

```
<html>
<body>
<script language="javascript">
    document.write("Hello World!");
</script>
</body>
</html>
```

Les commentaires

Les commentaires sur un seul lignes :

```
// Ceci est un commentaire  
// sur plusieurs lignes
```

Les commentaires sur plusieurs lignes :

```
/* Ceci est un commentaire  
sur plusieurs lignes */
```


Les variables

5 types de variables en JS.

**Les nombres : number, les chaines : string,
les booléens : boolean, les objets**

Exemple :

```
var chaine = 'bonjour';
```

La déclaration de variables

la déclaration des variables peut se faire de deux façons :

façon explicite : en faisant précéder la variable du mot clé *var*

façon implicite : en laissant le navigateur déterminer qu'il s'agit d'une déclaration de variable

Exemple :

```
<SCRIPT language="Javascript">  
var MaVariable1 = 3;  
    MaVariable2 = 2;  
document.write(MaVariable1*MaVariable2);  
</SCRIPT>
```

Portée (visibilité) des variables

variable globale

**variable déclarée au début du script, avant toute fonction
Elle peut être utilisée n'importe où dans le script .**

variable locale :

**variable déclarée explicitement dans une fonction aura
une portée limitée à cette seule fonction,**

Les chaînes de caractères

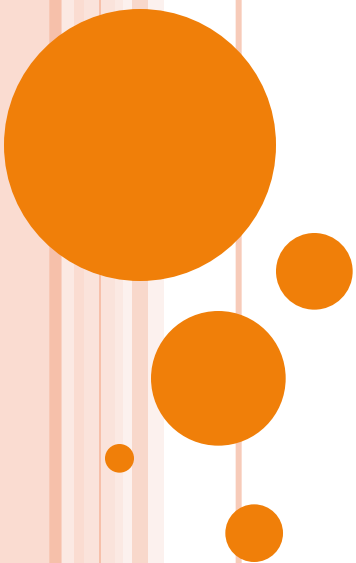
JavaScript convertit les entiers en chaînes de caractères, ce qui permet de concaténer des nombres avec des chaînes de caractères

Dans l'instruction d'écriture dans le document courant, `document.write()`, les données à afficher peuvent être séparées par des `+`

Des caractères spéciaux peuvent aussi être insérés dans les chaînes : `\b` (retour arrière), `\f` (saut de page), `\n` (nouvelle ligne), `\r` (Entrée), `&` (tabulation); `'` pour une apostrophe

On peut insérer des codes HTML sous forme de chaînes, qui seront bien interprétées à l'exécution comme de véritables balise, et non pas affichées telles quelles.

Les chaînes de caractères



```
<HTML>
  <HEAD> <TITLE> Exemple 1 </TITLE> </HEAD>
  <BODY>
    <SCRIPT LANGUAGE="JavaScript">
      var bonjour = "Bonjour !";
      var question = "Comment allez-vous ";
      var phrase = bonjour + "<BR>" + question;
      document.write (phrase, "aujourd'hui ?");
    </SCRIPT>
  </BODY>
</HTML>
```

TABLEAUX

Déclaration

```
var nom_tableau = new Array(dimension) ;
```

Exemples

```
var MonTableau = new Array(8)  
var Les4saisons = new Array("printemps",  
    "été", "automne", "hiver");  
var mois = new Array(12);
```

TABLEAUX

Propriétés

length : donne le nombre d'éléments.

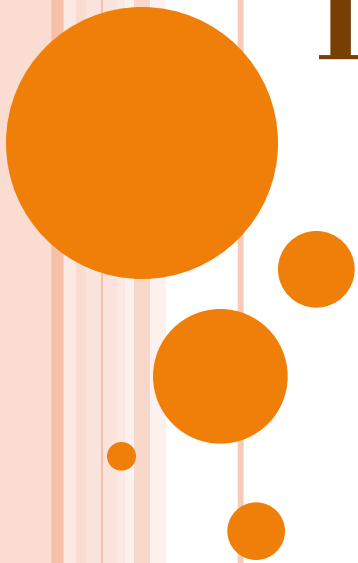
```
var mois= new Array(12);  
document.write("Il y a ", mois.length,  
"mois dans l'année");  
var NbMois = mois.length ;  
document.write(" partagés en ", NbMois / 3, "  
trimestres");
```

Méthodes

reverse() change l'ordre des éléments

sort() trie suivant l'ordre croissant, ou suivant le
modèle indiqué en paramètre

Les structures de contrôle



IF .. ELSE

```
if (condition)  
    { séquence 1 }  
else  
    { séquence 2 }
```

La condition doit être toujours entourée de ()

La séquence 1 est exécutée si la condition est vraie.

La séquence 2 (optionnelle) est exécutée si la condition est fausse.

les { } ne sont pas obligatoires qu'en cas d'instructions multiples.

L'ITÉRATION CONTRÔLÉE FOR

for (valeur initiale; condition; poursuite)
{ séquence d'instructions }

une valeur ou expression initiale portant sur une variable entière appelée compteur.

une condition : tant qu'elle est vraie, la répétition est poursuivie.

une expression de poursuite qui consiste en la mise à jour du compteur.

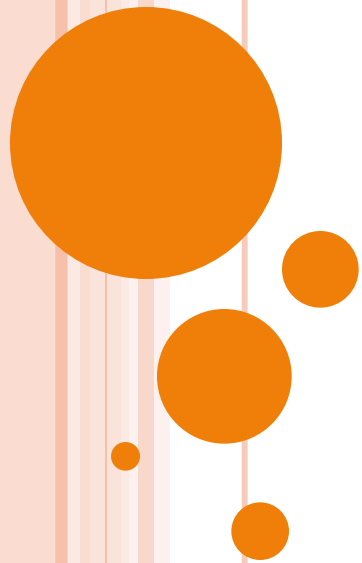
EXAMPLE

```
<SCRIPT language = "JavaScript" >  
document.write("Table des carrés<BR>");  
for (var i = 0; i <15; i++)  
{ document.write("i = "+i+"      i² = "+  
i*i+"<BR>"); }  
</SCRIPT>
```

L'itération WHILE (TANT QUE)

```
while (condition)  
{  
code qui sera exécuté à  
chaque passage  
}
```

Les classes prédéfinies



CLASSE MATH

Math.sqrt() , racine carrée.

Math.log() , **Math.exp()** , **Math.abs()** , **Math.cos()** , **Math.sin()** , **Math.tan()**

Math.floor(), **Math.ceil()** entier immédiatement inférieur / supérieur.

Math.pow(base, exposant), fct puissance, où base et exposant sont des expressions numériques quelconques évaluables.

Math.max() , **Math.min()**

Math.random(), nombre "réel" choisi au hasard dans [0 , 1[

Math.round() arrondit à l'entier le plus proche.

CLASSE STRING

parseInt(), donne un nombre entier résultant de la conversion (si possible) d'une chaîne de caractères. Si la conversion n'est pas possible, la valeur renvoyée est 0.

parseFloat(), donne un nombre décimal de la même façon.

chaîne.toUpperCase(), pour mettre chaîne en majuscule.

chaîne.substring(d, f) extrait une partie de chaîne, à partir du caractère de position d, jusqu'à f-1.

chaîne.charAt(n) donne le caractère placé en nième position (n de 0 à chaîne.length-1).

chaîne.indexOf(s_ch) donne la 1ère position du caractère de chaîne égal au 1er caractère de s_ch.

LES FONCTIONS PRÉDÉFINIES

○ eval

- Cette fonction exécute un code Javascript à partir d'une chaîne de caractères.

○ isFinite

- Détermine si le parametre est un nombre fini. Renvoie *false* si ce n'est pas un nombre ou l'infini positif ou infini négatif.

```
isFinite(240) //retourne true  
isFinite("Un nombre") //retourne false
```

○ isNaN

- détermine si le parametre n'est pas un nombre (NaN : Not a Number).

```
isNaN("un nombre") //retourne true  
isNaN(20) //retourne false
```


LES FONCTIONS PRÉDÉFINIES

○ Number

- convertit l'objet spécifié en valeur numérique

```
var jour = new Date("December 17, 1995 03:24:00");//convertit la date en  
millisecondes  
alert (Number(jour));
```

○ String

- convertit l'objet spécifié en chaîne de caractères

```
jour = new Date(430054663215);//Convertit le nombre en date Mois jour, Année  
etc.  
alert (String(jour));
```

○ Escape

- retourne la valeur hexadécimale à partir d'une chaîne de caractère codée en ISO-Latin-1.

```
escape("!&") //retourne %21%26%
```



CLASSE DATE

Date() , pour obtenir la date et l'heure courante (connue du système)

Date(month day, year hour:min:sec) pour obtenir par exemple : *December 25, 1995 13:30:00*)

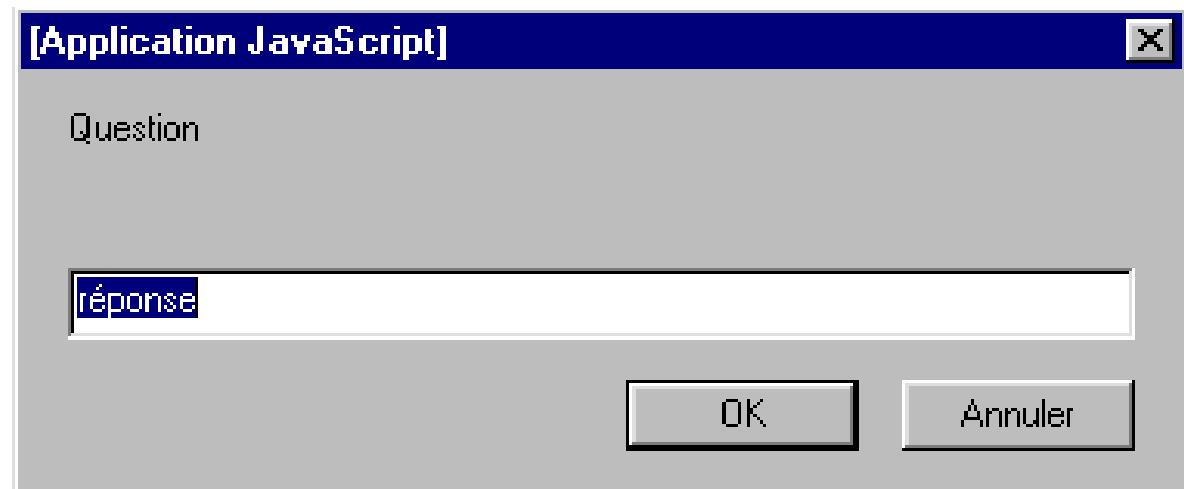
Date(année, mois, jour), une suite convenable de 3 entiers, par exemple (2000, 0, 1)

Les fenêtres de dialogue

Demande d'information

reponse = prompt("question","réponse par défaut");

document.write(reponse);



Les fenêtres de dialogue

Demande de confirmation

confirmation = confirm("demande de confirmation");

document.write(confirmation);



Les fenêtres de dialogue

Alertes

alert("message d'alerte");





Procédures et fonctions

PROCÉDURES ET FONCTIONS

<HEAD>

<SCRIPT LANGUAGE=JavaScript >

function nomProcédure(param1, param2, ...)

{ séquence d'instructions; }

function nomFonction(param1, param2, ...)

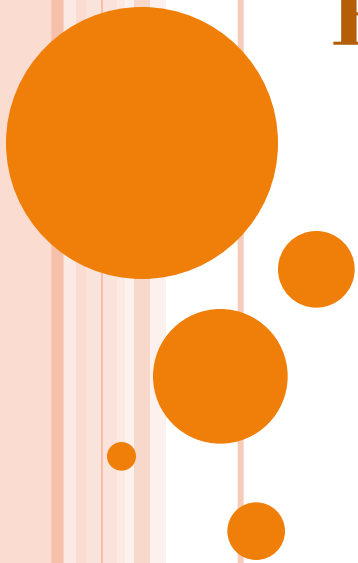
{ séquence d'instructions;

return nom_variable }

</SCRIPT>

</HEAD>

Programmation événementielle



EVÉNEMENTS

onChange : se produit quand un champ Textarea, Text ou Select est modifié par l'utilisateur.

onClick : se produit quand un composant Button, Checkbox, Radio Link, Reset ou Submit reçoit un click souris

onFocus : se produit quand un composant Textarea, Text ou Select est activé.

onLoad : se produit quand le navigateur a fini de charger une fenêtre ou toutes les frames d'un FRAMESET. L'événement *onLoad* se positionne dans la balise BODY ou dans la balise FRAMESET

onMouseOver : se produit quand la souris passe sur un Hyperlien ou une zone activable d'une image.

EVÉNEMENTS

onSelect se produit quand un composant Textarea ou Text est sélectionné.

onSubmit : se produit quand un formulaire est globalement validé appui du bouton *Submit*.

onUnload : se produit quand on quitte un document. L'événement *onUnload* se positionne dans la balise BODY ou dans la balise FRAMESET.

onError : se produit quand le chargement d'une page ou d'une image produit une erreur.

onMouseout : se produit quand la souris quitte une zone Area d'une image ou un hyperlien.

onReset : se produit quand on clique sur le bouton *reset* d'un formulaire