

Aide-mémoire Javascript

Insertion de code Javascript dans une page HTML

- Comme élément de type script de la page HTML avec la balise `<script>`:

```
<script language="javascript">  
  code javascript  
</script>
```
- Dans un gestionnaire d'évènement d'une balise HTML, entre " " ou entre ' '

```

```

Variables, déclaration, types, affectation

Les variables ne sont pas typées : c'est l'affectation d'une valeur qui définit le type.

Exemples de déclarations/affectation:

```
let message="bonjour"; // message est une variable locale  
const largeur=25; // largeur est une constante locale  
var abscisse=60; // abscisse est une variable locale
```

Les valeurs peuvent être de type nombre, booléen ou chaîne de caractères (entre " " ou ' ')

Les noms de variable doivent commencer par une lettre ou _ ou \$.

JavaScript est sensible à la casse (majuscules/minuscules) dans les noms de variables

Tableaux

```
monTableau=new Array(longueur)
```

```
monTableau=new Array(element1,element2,...element N)
```

Commentaires

```
// commentaire
```

Ou

```
/*  
Bloc de commentaire  
De plusieurs  
Lignes  
*/
```

Instructions et séquences d'instructions

La fin d'une instruction est signalée par un point-virgule ;

Une séquence d'instruction est encadrée par { }

L'indentation est facultative

Principaux opérateurs

- Arithmétiques: + - * / % (modulo) ++ incrément -- décrémentation
- Logiques :
 - && (ET logique)
 - || (OU logique)
 - ! (NON logique)
- Comparaison :
 - == égalité (ou === égalité stricte incluant le type)
 - != inégalité (ou !== différence de valeur et/ou de type)
 - < > <= >=

Actions itératives (répétitions)

```
for (var i=0;i<24;i++){instructions}
for (var i=23;i>0;i--){ instructions}
for (var i of iterable) [instructions ]
```

```
while (condition){instructions}
```

Actions conditionnelles

if ou if...else ...

```
    if (condition) {
                                instructions
    }
```

Ou

```
    if (condition) {
                                instructions
    }
    else {
    instructions alternative
    }
```

Procédures et fonctions (sous-programmes)

Procédure :

```
    function  maFonction(parametres, facultatifs){
                                                    Instructions
    }
```

Fonction

```
    function  maFonction(parametres, facultatifs){
        Instructions ;
        return valeur de retour;
    }
```

Fonctions mathématiques

Ce sont des méthodes de l'objet Math : syntaxe *Math.methode(parametres)*

Math.sin(x)

abs, sin, cos, tan, ceil, floor, round, sqrt, random, min, max