

L'objet Math et l'objet l'objet Date

L'objet Math

- L'objet **Math** va nous permettre de réaliser différentes opérations sur des nombres grâce à ses nombreuses méthodes.
- Dans les tableaux dessous il y'a une liste de fonctions mathématiques.

Les fonctions mathématiques

- Les fonctions de base :

<code>Math.abs(a)</code>	Retourne la valeur absolue de a
<code>Math.round(a)</code>	Retourne l'entier arrondi le plus proche de a
<code>Math.ceil(a)</code>	Retourne l'entier immédiatement supérieur (ou égal) à a
<code>Math.floor(a)</code>	Retourne l'entier immédiatement inférieur (ou égal) à a
<code>Math.sqrt(a)</code>	Retourne la racine carrée de a
<code>Math.log(a)</code>	Retourne le logarithme de a
<code>Math.ln(a)</code>	Retourne le logarithme népérien de a
<code>Math.exp(a)</code>	Retourne l'exponentielle de a
<code>Math.pow(a,b)</code>	Retourne a à la puissance b
<code>Math.min(a,b)</code>	Retourne le plus petit des paramètres a ou b
<code>Math.max(a,b)</code>	Retourne le plus grand des paramètres a ou b

Les fonctions mathématiques

- Les fonctions trigonométriques :

<code>Math.cos()</code>	Retourne le cosinus d'un angle en radians
<code>Math.sin()</code>	Retourne le sinus d'un angle en radians
<code>Math.tan()</code>	Retourne la tangente d'un angle en radians
<code>Math.acos()</code>	Retourne l'arc cosinus en radians
<code>Math.asin()</code>	Retourne l'arc sinus en radians
<code>Math.atan()</code>	Retourne l'arc tangente en radians

Exemples

```
var x=3, y=5 , z=0.4;
//retourne 3^5
var puiss=Math.pow(3,5);
//retourne la racine carrée de 3
var racine=Math.sqrt(x);
//retourne le sinus de 0.4
var sin=Math.sin(z);
//retourne le cosinus de 0.4
var cos=Math.cos(z);
//retourne la Pi la constante mathématique
var pi=Math.PI;
//retourne le le premier entier inferieur à 23.98 : 23
var val1=Math.floor(23.98);
//retourne le le premier entier supérieur à 23.98 : 24
var val2=Math.ceil(23.98);
```

La méthode random()

- La méthode random() retourne un nombre réel aléatoire compris entre 0 et 1.

```
for (var i=0; i< 10; i++) {  
  
document.write(Math.random());  
  
}
```

Va générer 10 valeurs réels aléatoires entre 0 et 1

```
var N= Math.floor(Math.random()* 20) ;  
document.write( N );
```

Va générer un nombre aléatoire entier compris entre 0 et 20

La méthode toFixed()

- Cette fonction est très utilisée, elle permet d'arrondir un nombre réel un nombre de valeur après la virgule.

- **Exemple :**

```
var x=12.908876 ;  
  
var y=x.toFixed(2) ; // 12.91  
  
var y=x.toFixed(3) ; // 12.909
```

- **Remarque:**
- La méthode toFixed() retourne un string et pas un number

L'objet Date

- l'objet Date va nous permettre de créer et de manipuler des dates grâce à ses méthodes.

Objet Date

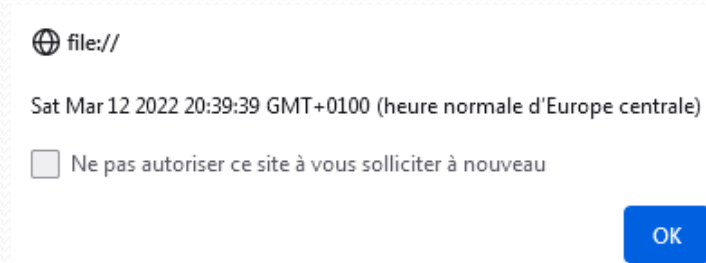
(Récupérer la date actuelle)

- On peut récupérer la date et l'heure actuelle (d'aujourd'hui) avec :

```
Var d1= new Date();
```

- Exemple :

```
var d=new Date();  
alert(d);
```



Créer un nouvelle date

- Pour créer une date spécifique on utilise:

```
Var d2= new Date(année, mois, jour, heure, minutes, secondes, millisecondes)
```

- Exemple :

```
var d=new Date(2021, 3,12);
```

Créer une date avec un format de date

- Il existe généralement 3 types de format de saisie de date javascript:

Type	Exemples
ISO date	"2021-03-20"
Short date	"03/20/2021"
Long date	"Mar 20 2021"

- Exemples :

```
var d=new Date("03/20/2021");  
var d=new Date("2021/03/20");
```

Objet Date

(Les getters de l'objet Date)

- `getDate()`: Permet de récupérer la valeur du jour du mois
- `getDay()`: permet de récupérer le jour de la semaine.
- `getFullYear()`: Permet de récupérer la valeur de l'année sur 4 chiffres
- `getMonth()` : Permet de récupérer le numéro du mois (0 pour janvier)
- `getHours()`: Permet de récupérer la valeur de l'heure
- `getMinutes()`: Permet de récupérer le nombre de minutes
- `getSeconds()`: Permet de récupérer le nombre de secondes
- `getTime()`: Permet de récupérer le nombre de millisecondes depuis le 1^{er} janvier 1970

Objet Date

(Les setters de l'objet Date)

- **setDate()**: Permet de modifier la valeur du jour du mois
- **setFullYear()**: Permet de modifier la valeur de l'année sur 4 chiffres
- **setMonth()** : Permet de modifier le numéro du mois (0 pour janvier)
- **setHours()**: Permet de modifier la valeur de l'heure
- **setMinutes()**: Permet de modifier le nombre de minutes
- **setSeconds()**: Permet de modifier le nombre de secondes

Exemple

```
<script>
//On crée un objet de type date stockant la date actuelle
var d = new Date();

/*getTime() renvoie le nombre de ms écoulées entre
 *le 1er janvier 1970 et la date contenue dans d*/
var dMs = d.getTime();

//getFullYear() renvoie l'année relative à la date
var dy = d.getFullYear();

/*getMonth() renvoie le mois sous forme de nombre
 *entre 0 (janvier) et 11(décembre)*/
var dmo = d.getMonth();

/*getDate() renvoie le jour du mois sous forme de
 *nombre (entre 1 et 31)*/
var ddate = d.getDate();

/*getDay() renvoie le jour de la semaine sous forme
 *de nombre entre 0 (dimanche) et 6 (samedi)*/
var dday = d.getDay();

var dh = d.getHours(); //Renvoie l'heure entre 0 et 23
var dmi = d.getMinutes(); //Renvoie les minutes (0-59)
var ds = d.getSeconds(); //Renvoie les secondes (0-59)
var dms = d.getMilliseconds(); //Renvoie les ms (0-999)

//On peut ensuite afficher la date complète
alert('Nous sommes le ' + ddate + '/' + dmo + '/' + dy +
      '\nIl est actuellement ' + dh + ':' + dmi + ':' + ds + '.');
</script>
```

Objet Date

(Les setters de l'objet Date)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Les dates en JavaScript</title>
    <meta charset="utf-8">
  </head>
  <body>
    <h1>Les dates</h1>
    <script>
      //On crée un objet de type date stockant la date actuelle
      var d = new Date();

      /*On modifie la date contenue dans notre objet.
      *setFullYear() va déjà servir à définir l'année. On peut
      *également l'utiliser pour définir le mois (0-11) et le jour*/
      d.setFullYear(2015,5,15);

      d.setHours(12); //Sert à définir l'heure
      d.setMinutes(30); //Sert à définir les minutes
      d.setSeconds(15); //Sert à définir les secondes

      alert('Date définie : ' +d)
    </script>
  </body>
</html>
```

Créer une date à partir d'un timestamp

- Le **timestamp** est le nombre de **millisecondes** écoulés depuis 01/01/1970 00:00:00 jusqu'à présent.
- Par exemple le timestamp de 13/03/2022 00:00:00 est: **1649714400000**
- Récupérer le timestamp actuel : **var x=Date.now()**
- ou en utilisant la méthode **getTime()**

Saisir une date

- Exemple 1:

```
var d=prompt("saisir votre date de naissance sous format Iso : yyyy-mm-dd");  
Nd=new Date(d);  
alert(Nd.getFullYear())
```

- Exemple 2:

```
var d=Date.parse(prompt("saisir votre date de naissance"));  
Nd=new Date(d);  
alert(Nd.getFullYear())
```

- **d** est un entier: est un **timestamp**. Pour appliquer les méthodes déjà vue, il faut le convertir en objet Date()

La différence entre deux dates

- La différence entre deux date est calculée en timestamp, c'est-à-dire le nombre de millisecondes entre les deux dates:

```
Var d1=new Date(2022,4,12)
```

```
Var d2=new Date(2022,2,13)
```

```
Var diff=d1 - d2
```

- On peut faire la division sur 1000 pour obtenir le nombre de secondes, diviser ensuite sur 60 pour obtenir le nombre de minutes, et ainsi de suite,
- On peut faire la différence des années des deux dates, ou bien les mois des deux dates, etc