

Cours de programmation

Les classes prédéfinies

Introduction :

Afin de gérer facilement les types de données complexes les plus courantes (dates, images...), JavaScript dispose de toute une série de classes prédéfinies. Pour chacune, il existe un constructeur qui permet de créer des objets de la classe en question.

La classe *Date()* :

Constructeur :

Pour manipuler une date avec JavaScript, il faut créer un objet de la classe *Date*. La création d'un tel objet de façon classique, en associant l'instruction *new* et le constructeur prédéfini *Date()*.

Méthodes :

Les méthodes ci-dessous s'appliquent aux objets de la classe *Date()*.

Méthodes	Description
<i>getDate()</i>	Retourne le jour du mois, compris entre 1 et 31.
<i>getDay()</i>	Retourne le jour de la semaine, compris entre 0 et 6 (0 = Dimanche).
<i>getHours()</i>	Retourne l'heure, comprise entre 0 et 23.
<i>getMinutes()</i>	Retourne les minutes, comprises entre 0 et 59.
<i>getMonth()</i>	Retourne le mois compris entre 0 et 11 (0 = Janvier).
<i>getSeconds()</i>	Retourne les secondes, comprises entre 0 et 59.
<i>getTime()</i>	Retourne le nombre de millisecondes écoulées depuis le premier Janvier 1970.
<i>getTimeZoneoffset()</i>	Retourne le nombre de minutes de décalage entre l'heure locale et l'heure GMT.
<i>getYear()</i>	Retourne le nombre d'années depuis 1900.
<i>setDate()</i>	Affecte le jour du mois, compris entre 1 et 31.
<i>setHours()</i>	Affecte l'heure, comprise entre 0 et 23.
<i>setMinutes()</i>	Affecte les minutes, comprises entre 0 et 59.
<i>setMonth()</i>	Affecte le mois compris entre 0 et 11 (0 = Janvier).
<i>setSeconds()</i>	Affecte les secondes, comprises entre 0 et 59.
<i>setTime()</i>	Affecte le nombre de millisecondes écoulées depuis le premier Janvier 1970.
<i>setYear()</i>	Affecte le nombre d'années depuis 1900.

toGMTString()	Retourne une chaîne en indiquant la date en utilisant les conventions GMT.
toLocaleString()	Retourne une chaîne en indiquant la date en utilisant les conventions locales.

La classe *Image()* :

Constructeur :

La création d'un objet *Image* se fait de façon classique, en associant l'instruction *new* et le constructeur prédéfini *Image()*.

Une fois que l'on a créé un objet *Image*, il suffit d'indiquer l'url référençant l'image, de façon qu'elle soit chargée à travers le réseau et stockée dans cet objet. Cela se fait en utilisant la propriété *src*.

Propriétés :

La classe *Image* a quatre propriétés. Ces propriétés ont déjà été vues.

Propriétés	Description
src	Chaîne de caractère contenant l'url du fichier image à charger.
height	Hauteur de l'image en pixels.
width	Largeur de l'image en pixels.
complete	Booléen indiquant si le chargement de l'image est terminé.

Pour modifier une image, on se contente de modifier le contenu *src*. Cette méthode est souvent utilisée lors de conception de menu.

Les événements :

Bien sûr, on peut utiliser les événements avec les objets *Image*. Consultez le cours sur les événements pour en connaître davantage.

La classe *Math* :

Propriétés :

Il s'agit en effet d'une classe un peu particulière. Elle ne dispose pas de constructeur.

Les propriétés de l'objet *Math* sont les principales constantes mathématiques.

Propriétés	Description
E	Constante d'Euler.
LN10	Logarithme népérien de 10.
LN2	Logarithme népérien de 2.

PI	C'est le rapport de la circonférence d'un cercle sur son rayon.
SQRT1_2	Racine carrée de 1/2.
SQRT2	Racine carrée de 2.

Méthodes :

Les méthodes de l'objet *Math* sont les principales fonctions mathématiques.

Méthodes	Description
abs(x)	Retourne la valeur absolue de "x".
acos(x)	Retourne l'arc cosinus de "x".
asin(x)	Retourne l'arc sinus de "x".
atan(x)	Retourne l'arc tangente de "x".
atan2(x,y)	Retourne l'angle associé aux coordonnées polaires du point de coordonnées cartésiennes "(x,y)".
ceil(x)	Retourne le plus petit entier supérieur ou égal à "x".
cos(x)	Retourne le cosinus de "x".
exp(x)	Retourne l'exponentielle de "x".
floor(x)	Retourne le plus grand entier inférieur à "x".
log(x)	Retourne le logarithme de "x".
max(x,y)	Retourne le plus grand des deux nombres "x" et "y".
min(x,y)	Retourne le plus petit des deux nombres "x" et "y".
random()	Retourne un nombre pseudo-aléatoire entre 0 et 1.
round(x)	Retourne l'entier le plus proche de "x".
sin(x)	Retourne le sinus de "x".
sqrt(x)	Retourne la racine carrée de "x".
tan(x)	Retourne la tangente de "x".

La classe String :

Comme nous l'avons déjà vu, il n'est pas nécessaire d'utiliser un constructeur pour créer une chaîne de caractères. Pour une question de syntaxe, il est tout de même conseillé d'utiliser le constructeur *String()*.

Constructeur :

La création d'un objet chaîne de caractères se fait de façon classique, en associant l'instruction *new* et le constructeur prédéfini *String()*.

• Exemple :

```
var pays = new String('France');
// Cette syntaxe est équivalente à :
var pays = 'France';
```

Propriété :

Les objets chaînes de caractères possèdent une seule propriété.

Propriétés	Description
length	Nombre de caractères dans la chaîne de caractères.

Méthodes :

Par contre, les objets chaînes de caractères possèdent de nombreuses méthodes, dont une grande partie est destinée à formater la chaîne selon une balise HTML.

Méthodes	Description
charAt(x)	Retourne le caractère présent à la position "x"
charCodeAt(x)	Retourne l'entier correspondant au code du caractère présent à la position "x".
concat(x)	Permet de concaténée deux chaînes de caractères. "x.concat(y)" est équivalent à "x+y".
indexOf(x,y)	Retourne la position de la première chaîne "x" situé après l'entier "y".
lastIndexOf(x,y)	Retourne la position de la dernière chaîne "x" situé avant l'entier "y".
slice(x,y)	Retourne la sous-chaîne débutant à la position "x" et terminant à la position "y".
split(x,y)	Retourne un tableau contenant les sous-chaînes obtenues en découpant la chaîne selon le séparateur "x". "y" est un paramètre optionnel qui permet d'indiquer le nombre d'élément maximum.
substr(x,y)	Retourne la sous-chaîne débutant à la position "x" et de longueur "y".
substring(x,y)	Retourne la sous-chaîne comprise entre les positions "x" et "y".
toLowerCase()	Retourne la chaîne mise en minuscule.
toUpperCase()	Retourne la chaîne mise en majuscule.

Exemple :

```
var pays = new String('France et Allemagne');
pays.indexOf('an'); // Retourne 3
pays.indexOf('a',5); // Retourne 16
pays.lastIndexOf('e',7) // Retourne 6
```

Méthodes de formatage HTML :

Il est possible d'utiliser des balises HTML au sein d'une chaîne de caractères, par exemple pour mettre du texte en gras ou autre.

Eh bien, certaines fonctions JavaScript permettent de les insérer. Dans le tableau suivant, les points de

suspension sont à remplacer par la chaîne de caractères.

Mthodes	Résultat
anchor(x)	...
big()	<BIG>...</BIG>
blink()	<BLINK>...</BLINK>
bold()	<BOLD>...</BOLD>
fixed()	<TT>...</TT>
fontcolor(x)	...
fontSize(x)	...
italics()	<I>...</I>
link(x)	...
small()	<SMALL>...</SMALL>
strike()	<STRIKE>...</STRIKE>
sub()	_{...}
sup()	^{...}

• Exemple :

```
var titre = new String('Langage de programmation');
document.write(titre.bold()+'<BR>');
document.write(titre.fontcolor("#3333FF")+'<BR>');
document.write(titre.fontSize(1));
```

• Ce que cela donne :

```
var titre = new String('Langage de programmation'); document.write(titre.blink()+'
'); document.write(titre.fontcolor("#3333FF")+'
'); document.write(titre.fontSize(1));
```

