Rappel Classe et Objet

Une **classe** est un type qui est la **description d'un ensemble** de :

- **propriétés** (les données) et de
- **méthodes** (les fonctions).

Un **objet** est une **instance de classe** (c'est à dire une variable de type classe). On écrira :

- o.p pour accéder à une propriété p d'un objet o
- o.m() pour appeler une méthode m d'un objet o

Les objets Javascript

Les objets en JavaScript

Les objets en JavaScript sont des collections de **propriétés** et de **méthodes**. Une **méthode** est une **fonction membre d'un objet**. Une propriété est une valeur ou un ensemble de valeurs (sous la forme d'un tableau ou d'un objet), membre d'un objet. JavaScript prend en charge différents types d'objets :

- Les objets intrinsèques, tels que **Array String** et **Math**.
- Les objets hôtes, tels que **window** et **document**.
- Les objets que vous créez vous même.

Propriétés et méthodes exemple

```
var contact = new Object();

contact.name = "Philippe";
contact.age    = 48;

contact.getAge =

   function () {
        return this.age;
    };

document.write(contact.name);
document.write("<br/>");
document.write(contact.age);
document.write("<br/>");
document.write(contact.getAge());
```

Les tableaux

- Dans JavaScript, les tableaux et les objets sont gérés quasiment de la même manière, car les tableaux sont simplement un type particulier d'objet. Les tableaux peuvent avoir des propriétés et des méthodes. Les tableaux ont une propriété **length.** Pour créer un tableau, utilisez l'opérateur **new** et le **constructeur Array()**, comme dans l'exemple suivant.

```
// un tableau avec trois éléments
var myArray = new Array(3);

// Add some data
myArray[0] = "Hello";
myArray[1] = 42;
myArray[2] = new Date(2000, 1, 1);

document.write("original length is: " + myArray.length);
```

Les tableaux à plusieurs dimensions

- JavaScript ne prend pas en charge les tableaux multidimensionnels, mais vous pouvez obtenir leur comportement en enregistrant des tableaux dans les éléments d'un autre tableau. C'est un tableau de tableaux. Exemple

```
// The size of the table.
var iMaxNum = 5;
var i, j;

// Set the length of the array to iMaxNum + 1.
// The first array index is zero, not 1.

var MultiplicationTable = new Array(iMaxNum + 1);

for (i = 1; i <= iMaxNum; i++) {
    MultiplicationTable[i] = new Array(iMaxNum + 1);
     for (j = 1; j <= iMaxNum; j++)    {
        MultiplicationTable[i][j] = i * j;

    }
}
```

Objet String

L'objet **String** possède certaines méthodes intégrées que vous pouvez utiliser avec vos chaînes. L'une d'entre elles est la **méthode substring**, qui retourne une partie de la chaîne. Elle utilise deux nombres comme arguments.

```
var aString = "0123456789";


// This code sets aChunk to "456".
var aChunk = aString.substring(4, 7);
// This code sets anotherChunk to "456"
```

Une autre propriété de l'objet **String** est la propriété **length**. Cette propriété contient le nombre de caractères de la chaîne.

Objet Math

L'objet **Math** a plusieurs constantes et fonctions prédéfinies. Quelque soit la méthode ou la propriété utilisée, il est indispensable de la préfixer avec Math car il s'agit de **méthodes et propriétés statiques**,

```
var rayon = 5;
var circleArea = Math.PI * rayon * rayon;
```

L'une des fonctions intégrées de l'objet **Math** est la méthode d'élévation, ou **Math.pow ()**, qui élève un nombre à une puissance spécifiée.

```
var volume = (4/3)*(Math.PI*Math.pow(radius,3));
```

Objet Math (suite)

Les méthodes de l'objet Math sont :

- abs()
- ceil() floor() round() trunc()
- max(Nombre1, Nombre2) min(Nombre1, Nombre2)
- pow(Valeur1, Valeur2)
- random()
- sqrt(Valeur)
- sin(valeur) cos(valeur) tan(valeur)
- asin(valeur) acos(valeur) atan(valeur)

Objet Date

- L'objet **Date** peut être utilisé pour représenter des dates et heures arbitraires, pour obtenir la date système actuelle et pour calculer les différences entre les dates. Il comporte plusieurs propriétés et méthodes prédéfinies.
- Généralement, l'objet Date fournit le jour de la semaine ; le mois, le jour et l'année ; et l'heure en heures, minutes et secondes.
- Ces informations sont basées sur le nombre de millisecondes écoulées depuis le 1er janvier 1970, 00:00:000 GMT.

L'objet date (exemple)

L'exemple suivant montre comment instancier une date sans utiliser aucun paramètre et l'afficher dans la console au format mm-dd-yy.

```
var dt = new Date();
// Display the month, day, and year.
// getMonth() returns a 0-based number.
var month = dt.getMonth()+1;
var day = dt.getDate();
var year = dt.getFullYear();
console.log(month + '-' + day + '-' + year);
```

Objet date (suite)

Vous pouvez également spécifier une heure. L'exemple suivant montre une façon de spécifier une date et une heure au format ISO. Le « Z » à la fin indique l'heure UTC.

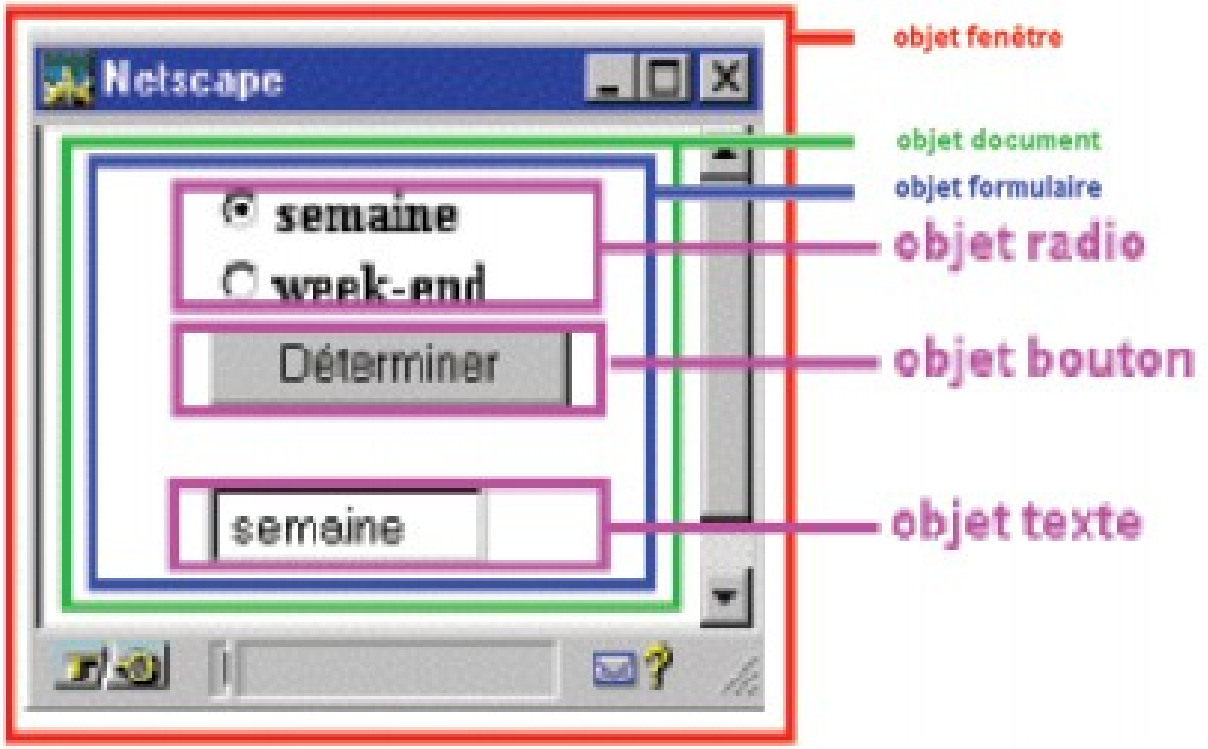
```
var dt = new Date('2017-06-03T15:20:00Z');
document.write( dt.toISOString() );
```

- Pour en savoir plus sur l'objet date consulter la documentation

  https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Obje

Document Object Model

- Lorsqu'un document HTML est chargé dans un navigateur, celui-ci fournit une interface DOM (Document Object Model) pour accéder aux objets le composant :
- Ces objets sont classés de manière hiérarchique (notion d'arbre). L'objet le plus haut dans la hiérarchie est l'objet de la classe window.
-



Document Object Model (suite)

Dans ce document HTML, on pourrait donc accéder aux objets de la manière suivante :

```
// Accès à une propriété de l'objet button
window.document.form.button.value = "Déterminer";
// Accès à une méthode de l'objet button
window.document.form.button.focus();
// Accès à une méthode de l'objet window
window.alert("Hello world");
```

Document Object Model (suite 2)

On recommande d'utiliser la méthode getElementById() pour accéder aux objets par leur identifiant (attribut id de l'élément HTML) :

```
var bouton = document.getElementById('id_button');
```

Création d'objets (JavaScript)

- En JavaScript, vous pouvez créer vos propres objets de différentes manières. Vous pouvez instancier directement un Object, objet (JavaScript), puis ajouter vos propres propriétés et méthodes.

```
var pasta = new Object();
pasta.grain = "wheat";
pasta.width = 0.5;
pasta.shape = "round";

pasta.getShape = function() {
    return this.shape;
};

document.write(pasta.getShape());
```

Création d'objets (json)

- Vous pouvez également utiliser la notation de littéral d'objet quand vous souhaitez créer une seule instance d'un objet. Le code suivant montre comment instancier un objet en utilisant la notation littérale d'objet (json).

```
var pasta = {
    grain: "wheat",
    width: 0.5,
    shape: "round"
};
```

Création d'objets (json)

L'exemple suivant illustre l'utilisation de la syntaxe abrégée pour définir des méthodes dans les littéraux d'objet.

```
var obj = {
    method1() {},
    method2() {}
};
```

Exercices

- Exo 1 : Créer un objet date pour afficher la date courante dans sur la page HTML.

  Pour l'affichage utiliser la méthode innerHTML() de document.

- Exo 2 : Utiliser la méthode toDateString() pour afficher la date avec le format suivant :

  Thu Mar 23 2017

Exercices suite

- Créez un formulaire permettant de faire la multiplication de deux valeurs (le résultat sera affiché lors d'un clic sur le bouton "=").

- Complétez le formulaire pour pouvoir également faire l'addition de deux valeurs (le résultat sera affiché lors d'un clic sur le bouton "=").