

INTRODUCCION A LOS OBJETOS DE USUARIO EN JAVASCRIPT:

1.- CONCEPTOS BÁSICOS:

Desde el punto de vista del código podemos definir como **objeto** una entidad compuesta por un conjunto de variables y funciones que actúan sobre dichas variables.

- Las **variables** son las **propiedades** que definen su estado.
- las **funciones** son los **métodos** que actúan sobre esas propiedades.

La forma de acceder a una propiedad de un objeto

`nombreObjeto.propiedad`

`nombreObjeto.método([parámetros opcionales])`

La forma de acceder a un método de un objeto

2.- CREACION DE OBJETOS:

En JavaScript existen múltiples maneras de realizar las mismas acciones, aquí veremos la estructura básica y estándar para trabajar con objetos.

Antes de crear un objeto nuevo, es necesario **definir a ese objeto** mediante la descripción de sus propiedades y métodos, es decir, debemos **crear primero el molde en el que especificaremos todo lo que se puede hacer con los objetos que utilicen ese molde.**

Este molde es la **clase o tipo**, y los objetos que creemos a partir del mismo son **instancias de esa clase**.

Para definir el nombre y propiedades de la clase se utiliza una función que se conoce como **constructor**:

Su sintaxis es la siguiente:

```
function mi_objeto (valor_1, valor_2, valor_x){  
  this.propiedad_1 = valor_1;  
  this.propiedad_2 = valor_2;  
  this.propiedad_x = valor_x;  
}
```

vamos a verla con un ejemplo:

```
function CALumno (nombre, nota, curso)  
{  
  this.nombre = nombre;  
  this.calificacion = nota;  
  this.curso = curso;  
}
```

← Hemos definido la clase CALumno

CAumno es el nombre de la clase y los valores que tendrá como **propiedades** van a continuación, entre paréntesis.

Notar que la palabra reservada **this** se utiliza para referenciar internamente al propio objeto, de esta forma se puede asignar valores a las propiedades de los objetos a partir de los valores pasados a la función constructora.

Una vez definido el nuevo objeto, podemos crear instancias, para ello se utiliza la palabra **new**:

```
alumno1 = new CAumno ( "Ana", 7 , "ASI2V" );
```

Hemos creado una instancia u objeto llamado Alumno1 con los valores de las tres propiedades. De esta manera podemos crear tantas instancias u objetos como deseemos.

El acceso a las propiedades de un objeto se hace mediante la siguiente sintaxis:

```
alumno1.nombre;  
alumno1.calificacion;  
alumno1.curso;
```

() Notar que esos nombres son los definidos en el this.nombre de la función, no son los parámetros que actúan como variables internas.*

3.- OBJETOS ANIDADOS:

Los objetos pueden anidarse de manera que un objeto sea propiedad de otro objeto:

```
// Clase CInstituto  
  
function CInstituto(denom,local)  
{  
    this.denominacion=denom;  
    this.ciudad=local;  
}  
  
//Clase CAumno  
  
function CAumno (nombre, nota, curso, CInstituto)  
{  
    this.nombre = nombre;  
    this.calificacion = nota;  
    this.curso = curso;  
    this.centro=CInstituto;  
}
```

Ahora puedo crear instancias del objeto CInstituto:

```
micentro = new CInstituto("IES Pio Baroja", "Orcasitas");
```

Y también instancias del objeto CAumno:

```
alumno1= new CAumno("Ana",6,"2ASI1", micentro);
```

Podemos referirnos a las propiedades de la siguiente forma:

```
alumno1.nombre  
alumno1.calificacion  
alumno1.curso
```

pero en el caso de querer ver el centro, este pertenece a otro objeto, entonces.

alumno1.centro.denominacion
alumno1.centro.ciudad

Un ejemplo completo es así:

```
<html>

<head>
<title> O b j e t o s </title>

<script>

    //Creo el objeto CInstituto

    function CInstituto(denom,local)
    {
        this.denominacion=denom;
        this.ciudad=local;
    }

    //Creo el objeto CAlumno, pero uno de sus propiedades va a ser el objeto CInstituto

    function CAlumno (nombre, nota, curso, CInstituto)
    {
        this.nombre = nombre;
        this.calificacion = nota;
        this.curso = curso;
        this.centro=CInstituto;
    }

</script>
</head>

<body>

<script>

    //Creo instancias de la clase CInstituto

    micentro   = new CInstituto("IES Pio Baroja", "Orcasitas");
    otrocentro  = new CInstituto("IES Felipe Trigo", "Mostoles");

    //Creo objetos (Instancio) de esa clase CAlumno

    alumno1= new CAlumno("Ana",6,"2ASIR1", micentro);
    alumno2= new CAlumno("Pepe",5,"1ASIR2", micentro);
    alumno3= new CAlumno("Julian",6,"1DAW2",otrocentro);

    //Accedo al objeto en formato objeto.propiedad

    document.write (micentro.denominacion + " " + micentro.ciudad + "<br>");

    //Accedo al objeto CAlumno en formato objeto.propiedad

    document.write (alumno1.nombre + "<br>");

    //Accedo al objeto CAlumno en formato array asociativo

    document.write (alumno2["nombre"] + "<br>");

    // document.write (alumno3.nota + "<br>");
    // ES UN ERROR ya que nota es una variable interna del objeto, está encapsulada y no se puede acceder a ella.

    document.write (alumno3.calificacion + "<br>"); //El acceso es mediante calificacion

    //Cuando me refiero al centro tengo que especificar qué quiero ver del centro:

    document.write (alumno2.centro.denominacion+"<br>");
    document.write (alumno3.centro.ciudad+"<br>");

</script>

</body>
```

</html>

4.- METODOS DE LOS OBJETOS:

Dentro de la definición de un Objeto se pueden incluir funciones que accedan a las propiedades del mismo. Estas funciones son los llamados **métodos**.

Para añadir métodos debemos primero crear la función que lo representa y una vez creada será necesario incluirla en la definición de la clase de manera similar a las propiedades:

```
function CAlumno (nombre, nota, curso) // Esta es la Clase
{
    this.nombre = nombre;
    this.nota = edad;
    this.curso = curso;
    this.verdatos=verdatos;
}

// Este es el método:

function verdatos() //Esto es un metodo que se incluirá en la clase
{
    Var curso=char
    document.write("Nombre: " + this.nombre + "<br>");
    document.write("Nota: " + this.nota + "<br>")
    document.write("Estudia: " + this.curso + "<br>")
}
```

Después de crear un objeto del tipo CAlumno:

```
alumno1 = new CAlumno ( "Pedro", 6 , "DAW" );
```

El acceso a los métodos se haría del siguiente modo:

```
alumno1.verdatos();
```

Avanzado: Los métodos pueden llevar parámetros, por ejemplo:

```
Tortuga.avanza(5,8)
```

Ejemplo completo con métodos:

```
<html>
<head>
<title> Ejemplo de creación objeto de usuario </title>
<script>

function CInstituto(denom,local) {
    this.denominacion=denom;
    this.ciudad=local;
}

function CAlumno(nombre,nota,curso, CInstituto) {
    this.nombre = nombre;
    this.calificacion = nota;
    this.curso = curso;
    this.centro=CInstituto; //Aquí incluyo otra clase anidada
    this.estado=MEstado; //Aquí incluyo un método
}
```

//Esto es un método que nos va a dar informacion del alumno si es de 1º o de 2º y de qué área.

```
function MEstado() {
    var miGrado=this.curso;           //Se usan variables internas normales para hacer mas sencillo el código
    var miEspecialidad=this.curso;

    var numCurso, estudios;

    var textoSalida="";

    //Usamos funciones de cadena para extraer el curso y la especialidad

    miGrado=miGrado.charAt(0); //Saco el numero del curso 1 o 2.

    miEspecialidad=miEspecialidad.substr(1,3) //Saco la especialidad "ASIR" o "DAW"

    //Se usa el if abreviado ya que se supone que solo hay dos casos, en otra situacion usaria un switch

    numCurso=(miGrado=="1")?"Primero":"Segundo"
    estudios=(miEspecialidad=="ASIR")?"Sistemas":"Desarrollo"

    textoSalida+="

```

5.- ARRAYS Y OBJETOS (Arrays asociativos):

En JavaScript las propiedades de los objetos y los Arrays están relacionados:

```
var alumno1 = new Array("Repulles",5,"2ASIR")           //      ← Array

alumno1 = new CAlumno ( "Repulles", 5 , "2ASIR" );      //      ← Objeto
```

Son los ejemplos que hemos creado como Array alumno1 y un objeto de la clase Alumno.

Se observa con claridad que, en realidad, un objeto de JavaScript es básicamente un Array, y sus propiedades son realmente nombres de casillas o posiciones del Array según su orden de creación, así en nuestro objeto Alumno1:

```
alumno1[0] = alumno1.nombre

alumno1[1] = alumno1.nota
```

`alumno1[2] = alumno1.curso`

Se puede acceder a las propiedades de un objeto utilizando la sintaxis de los Arrays, es decir, el nombre del objeto seguido de corchetes dentro de los cuales en lugar del índice de la posición, pondremos, entre comillas la propiedad a la que queremos acceder:

| | | |
|---------------------------------|----------------|-----------------------------|
| <code>alumno1 ["nombre"]</code> | <code>↔</code> | <code>alumno1.nombre</code> |
| <code>alumno1 ["nota"]</code> | <code>↔</code> | <code>alumno1.mota</code> |
| <code>alumno1 ["curso"]</code> | <code>↔</code> | <code>alumno1.curso</code> |

Enlaces:

https://www.w3schools.com/js/js_objects.asp

https://developer.mozilla.org/es/docs/Web/JavaScript/Introducci%C3%B3n_a_JavaScript_orientado_a_objetos

http://es.wikibooks.org/wiki/Programaci%C3%B3n_en_JavaScript/OOP

<http://www.desarrolloweb.com>