Creación de Objetos

Tema 5

¿Cómo?

- Crear un nuevo objeto consiste básicamente en declararlo en una función, y aplicarle nuevas propiedades y métodos.

Crear mediante un constructor

- Un constructor es una función que inicializa el objeto, para ello utilizamos una función normal, y en el paréntesis le pasamos algunos parámetros que nos servirán para definir las propiedades que tendrá el objeto.

Ejemplo:

- Crear un modelo de objeto Alumnos, para recoger los datos de cada alumno cuando se matricula (por ejemplo que recoja el nombre, curso y número de asignaturas)

La función que la crea será la siguiente:

- **function Alumno(nombrep,cursop,numMateriasp) { ... }**

Crear Propiedades

- Los parámetros que enviamos vamos a convertirles en propiedades del objeto Alumno.
- Creamos una referencia al parámetro dentro de la función con la palabra reservada **this**, de la siguiente manera:

```
function Alumno(nombrep,cursop,numMateriasp) {
this.alumno = nombrep;
this.curso = cursop;
this.materias = numMateriasp;
 }
```

Crear un objeto a partir de una definición

- Crearemos mediante las funciones constructor:

alumno1 = new Alumno("Juan Pérez","tecnología",5)

- De está forma tenemos un objeto de tipo Alumno y podremos acceder a sus propiedades:

nombre1 = alumno1.alumno

Métodos

Los métodos son funciones, por lo tanto el método debe igualarse al nombre de una función que indique qué es lo que queremos hacer con el objeto.

En la función constructor pondremos lo siguiente para crear un método:

this.precioCurso = matricula (Estará incluida junto con las propiedades)

Donde precioCurso es el nombre que le daremos al método, y matricula es el nombre que le daremos a la función.

El siguiente paso es crear la función a la que se refiere el método, en nuestro ejemplo, el método precioCurso() establecerá el precio en base al número de asignaturas, de manera que serán 100€ por asignatura. la función que controla este método es la siguiente:

la función puede hacer referencia a propiedades del objeto mediante **la palabra reservada `this`,** y debe tener siempre un **valor de retorno con `return`** que será el que es devuelto por el método.

Estará incluida junto con las propiedades

```
function matricula() {
precio = 100*this.materias
return precio
}
```

Metodos con argumentos

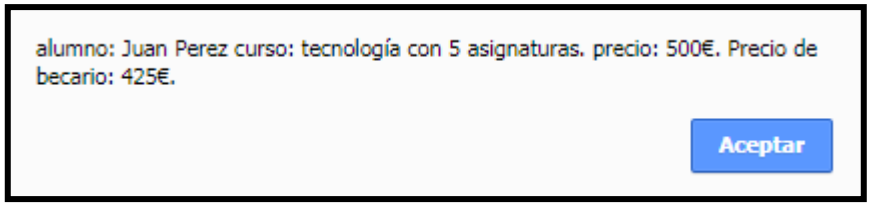
- Cuando los métodos necesitan argumentos, crearemos la función asociada, normalmente con sus parámetros: (En el parámetro de la función pasaremos el tanto por ciento de reducción del precio)

```
function beca(num)  {
precio = 100*this.materias;
descuento = precio*num/100;
precioFinal = precio - descuento;
return precioFinal;
 }
```

La definición en el objeto es igual y la llamada desde un objeto nuevo debe de ser completada con el parámetro.(Como siempre).

Probando

- Definir el objeto Alumno (igual que en el ejemplo), crear un objeto alumno1 a partir de el, y visualizar sus datos:



De la misma manera podemos cambiar el valor de cualquier propiedad, simplemente con asignarle un nuevo valor.

Cambiar el número de asignaturas y visualizar otra vez los datos.

La propiedad prototype

- **Definición**

La propiedad prototype es una propiedad del objeto Object, y por tanto es heredada por el resto de los objetos. Por lo tanto todos los objetos poseen la propiedad prototype.

Mediante esta propiedad podemos asignar nuevos metodos y propiedades a cualquier clase de objetos.

Asignar nuevas propiedades

Ejemplo:

Function Calcular() { }

Tenemos estos valores:

a = 8; b = 10;

Después mediante la propiedad prototype asignamos estos valores como propiedades de la función anterior.

```
Calcular.prototype.n1 = a;
Calcular.prototype.n2 = b;
```

Probando

- Crear un objeto **calculo,** de tipo calcular y visualizar sus propiedades.

Asignar nuevos métodos

Asignar nuevos métodos a un objeto mediante la propiedad **prototype** se hace de la misma forma que para asignar propiedades.

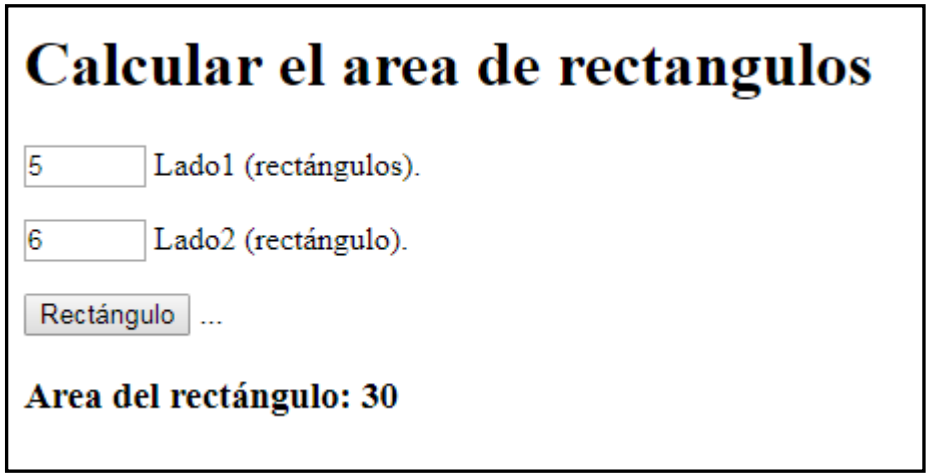
La única diferencia es que en lugar de asignarle como valor por defecto una variable, se le asigna como valor una función.

Probando

- Siguiendo con nuestro ejemplo, crearemos un método que calcule el área de un rectángulo. las propiedades n1 y n2 serán la longitud de los lados. Debemos crear primero la función que calcula el área, en ella la referencia a las propiedades anteriores se hace mediante la palabra reservada this.
- Después asignarsela a nuestro objeto Calcular.

Integrando

- Ahora recomponer la aplicación creando un formulario para introducir los valores y también con una caja para visualizar el resultado.



Completar

Calcular el area de circulos, rectangulos y triángulos.

Radio (circulos), lado1 (rectángulos), o base(triángulo)

Lado2 (rectángulo), altura (triángulo)

Rectángulo ... Círculo ... Triángulo ...

Práctica nº1

- Crear un modelo de objetos llamado ***cd_musica*** que contenga las propiedades: ***titulo, grupo y fecha*** **y un método** que nos devuelve una cadena con esta información.
- Hacer un programa que **cree tres objetos** con la información que se solicita por pantalla y después los muestre utilizando el método creado para ello.
- Una vez que nos funcione vamos a **crear un array de objetos música.**
- **Visualizamos la información de todos.**
- **Añadimos** una nueva **propiedad** al modelo que será su **precio.**
- **Les** asignamos **sus precios a todos.**
- Caculamos **el precio total del lote.**

Práctica nº2

- 1-Crear el modelo de objeto **Cuentacorriente**.
- Propiedades: **Datos Personales-Nºcuenta-Saldo**
- Datos personales será una propiedad que recibirá el objetos **datosP**, que consta de estas propiedades **: nombre-dni-dirección y teléfono**
- Métodos:
- **Visualizar datos del cliente**
- **Visualizar saldo**
- **Abono y Cargo**(estos dos métodos servirán para aumentar el saldo o disminuirle según el caso)
- 2-Creamos un script que nos cree **N objetos Cuentacorriente** y nos **visualice los datos** de todos los clientes, así como el **saldo** de ese momento.
- 3-Nos aumente el saldo en 1000 euros a todos los clientes.
- 4-Nos cargue una retención del 10% a todos los clientes.
- 5-Nos de la posibilidad de hacer cargos o abonos a un cliente en particular con la cantidad que queramos y visualizar su saldo.

La propiedad prototype en objetos predefinidos.

- Podemos **definir nuevos métodos y propiedades** para los arrays, cadenas de texto, números, valores boleanos...
- Sin embargo el objeto **Math no admite** la propiedad prototype , ya que su función no es crear nuevos objetos, sino resolver ciertas operaciones matemáticas

Propiedades

- Vamos a añadir la propiedad nombre al objeto String:
  String.prototype.nombre = "nombre por defecto";

Si definimos así :

var texto = "Esta es una cadena de texto.“

- La propiedad no se puede modificar.

- Sin embargo si la definimos mediante el objeto String de la siguiente manera:

var otroTexto = new String("Esta es otra cadena de texto");

- Podremos modificar la propiedad:

texto.nombre = "Párrafo primero";

otroTexto.nombre = "Párrafo segundo."

Probando

- Añadir una propiedad nueva al objeto Array,
- Para probarlo crear dos objetos de distinta forma, uno metiéndole directamente los valores y otro con el constructor de Array.
- Probar.
- ¿Qué diferencia hay con la clase String?

http://aprende-web.net/javascript/js12_4.php

Métodos

- Los objetos predefinidos sí que tienen un contenido en sí mismos, y es a ese contenido al que se quiere acceder para modificarlo.
- La forma de acceder a ese contenido al construir la función que genera el método es mediante la palabra reservada this.

Ejemplo

- Creamos un método (o sea una función) para el objeto String que ponga los elementos de un array entre guiones, y los pase a mayúsculas:

```
function guionMayus() {
 var nuevoArray = this;
    for (i in nuevoArray) {
          nuevoArray[i] = "-"+nuevoArray[i]+"-";
          nuevoArray[i] = nuevoArray[i].toUpperCase();
    }
return nuevoArray;
 }
```

Ejemplo

- Le añadimos el método al objeto:

Array.prototype.guionMayus = guionMayus;

Para probar declarar un array estaciones con datos.

- Después lo probamos:

- alert("array original: \n"+estaciones.join(", "));
  alert("array modificado:
  \n"+estaciones.guionMayus().join(", "));

Ejercicio

- Añadir nuevo método al objeto number que ponga dos decimales a un número y lo redondee si tiene más, y pone el símbolo del euro detrás (€).