Programación orientada a objetos

Funciones

Francisco Arce
www.pacoarce.com

¿Qué es una función?

Francisco Arce
www.pacoarce.com

¿Qué es una función?

Una función en un conjunto de instrucciones que se llaman o invocan bajo un nombre y tienen un propósito definido y que puede regresar un valor.

Francisco Arce
www.pacoarce.com

¿Qué es una función?

```
function suma(a, b) {
   var c = a + b;
   return c;
}
```

Francisco Arce
www.pacoarce.com

¿Qué es una función?

- función (function)
- nombre
- propiedades
- instrucciones
- return

Francisco Arce
www.pacoarce.com

¿Qué es una función?

Una función tiene 2 parámetros:

- arguments
- prototype

Y tiene dos métodos:

- call()
- apply()

Francisco Arce
www.pacoarce.com

¿Qué es una función?

```
function sumaNumeros() {
  var i, tot = 0;
  var numeros = arguments.length;
  for (i = 0; i < numeros; i++) {
      tot += arguments[i];
  }
  return tot;
}
```

Francisco Arce
www.pacoarce.com

Funciones predefinidas

Francisco Arce
www.pacoarce.com

Funciones predefinidas

En JavaScript tenemos funciones predefinidas que no necesitan ninguna instancia para ser utilizadas:

- parseInt()
- parseFloat()
- isNaN()

Francisco Arce
www.pacoarce.com

Funciones predefinidas

- isFinite()
- encodeURI()
- decodeURI()
- encodeURIComponent()
- decodeURIComponent()
- eval()

Francisco Arce
www.pacoarce.com

Funciones anónimas

Francisco Arce
www.pacoarce.com

Función anónima

No tiene nombre y por lo general es ejecutada en el momento. La podemos asignar como una variable.

Francisco Arce
www.pacoarce.com

Función anónima

```
// asignamos la función a la variable saludo
 var saludo = function(hora)
{
  if (hora >= 22 || hora <= 5)
     document.write("Buenas noches");
  else
     document.write("Buenos días");
}
// llama a la función
saludo(10);
```

Francisco Arce
www.pacoarce.com

Funciones de callback

Francisco Arce
www.pacoarce.com

Función callback

Las funciones no son otra cosa que datos asignados a una variable, por lo que pueden ser copiados, borrados y llamados como parámetros.

Francisco Arce
www.pacoarce.com

Funciones de callback

Una función puede ser pasada como un parámetro.

Francisco Arce
www.pacoarce.com

Funciones que se autoinvocan

Francisco Arce
www.pacoarce.com

Funciones que se auto invocan

Por medio de los paréntesis, podemos autoinvocar a una función, generalmente es una función anónima.

Podemos pasarle parámetros por medio de paréntesis.

Francisco Arce
www.pacoarce.com

Funciones que se autoinvocan

```
(
   function(){
      alert('Hola, cara de bola');
   }
)()
```

Francisco Arce
www.pacoarce.com

Funciones que se autoinvocan

```
(
   function(nombre){
      alert('Hola ' + nombre + '!');
  }
)('Crayola')
```

Francisco Arce
www.pacoarce.com

Funciones dentro de funciones

Francisco Arce
www.pacoarce.com

Funciones dentro de funciones

Se puede crear una función dentro de otra función, pero sólo será visible dentro de ésta.

Francisco Arce
www.pacoarce.com

Funciones dentro de funciones

```
function a(numero) {
  function b(entrada) {
    return entrada * 2;
  };
  return 'Resultado ' + b(numero);
};
```

Francisco Arce
www.pacoarce.com

Funciones dentro de funciones

```
var a = function(numero) {
   var b = function(entrada) {
      return entrada * 2;
   };
   return 'Resultado ' + b(numero);
};
```

Francisco Arce
www.pacoarce.com

Funciones que regresan funciones

Francisco Arce
www.pacoarce.com

Funciones que regresan funciones

Dentro de la sentencia return puedes ejecutar una función anónima.

Si una función no tiene una sentencia “return”, regresará una valor “undefined”.

Francisco Arce
www.pacoarce.com

Funciones que regresan funciones

```
function a() {
  alert('Hola ');
  return function(){
              alert('cara de bola');
          };
}
```

Francisco Arce
www.pacoarce.com

Redefinir una función

Francisco Arce
www.pacoarce.com

Redefinir una función

Ya que una función puede regresar otra función, podemos utilizar la segunda función para redefinir a la primera.

Francisco Arce
www.pacoarce.com

Redefinir una función

Ya que una función puede regresar otra función, podemos utilizar la segunda función para redefinir a la primera.

Francisco Arce
www.pacoarce.com

call y apply

Francisco Arce
www.pacoarce.com

call y apply

Ambas funciones sirven para llamar a otra función.

En ambas, el primer parámetro debe ser el objeto propietario.

Francisco Arce
www.pacoarce.com

call y apply

En el método ***call***, los parámetros se pasan separados por comas.

En el método ***apply***, los parámetros se pasan como un arreglo.

Francisco Arce
www.pacoarce.com

call y apply

```
function producto(a, b) {
   return a * b;
}
var objeto = producto.call(objeto, 7, 3);
alert(objeto);
```

Francisco Arce
www.pacoarce.com

call y apply

```
function producto(a, b) {
   return a * b;
}
var objeto = producto.apply(objeto, [7, 3]);
alert(objeto);
```

Francisco Arce
www.pacoarce.com