

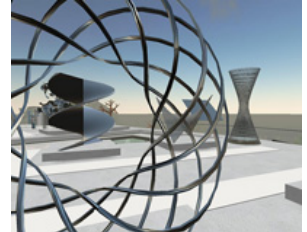


4.A. Estructuras definidas por el usuario en JavaScript

2.- Creación de funciones.

En unidades anteriores ya has visto y utilizado alguna vez funciones. **Una función es la definición de un conjunto de acciones pre-programadas. Las funciones se llaman a través de eventos o bien mediante comandos desde nuestro script.**

Las funciones son una herramienta muy potente en los lenguajes de programación, ya que le van a permitir realizar tareas de una manera mucho más organizada, y además le permitirán reutilizar un montón de código en sus aplicaciones.



Siempre que sea posible, tienes que diseñar funciones que puedas reutilizar en otras aplicaciones, de esta forma, tus funciones se convertirán en pequeños bloques constructivos que te permitirán ir más rápido en el desarrollo de nuevos programas.

Si conoces otros lenguajes de programación, quizás te suene el término de *subrutina* o *procedimiento*. En JavaScript no vamos a distinguir entre **procedimientos** (que ejecutan acciones), o **funciones** (que ejecutan acciones y devuelven valores). En JavaScript siempre se llamarán funciones.

Una función es capaz de devolver un valor a la instrucción que la invocó, pero esto no es un requisito obligatorio en JavaScript. Cuando una función devuelve un valor, la instrucción que llamó a esa función, la tratará como si fuera una expresión.

Veremos algunos ejemplos en un momento, pero antes de nada, vamos a ver la sintaxis formal de una función:

```
function nombreFunción ( [parámetro1]...[parámetroN] ) {  
  
    // Sentencias  
  
}
```

Si nuestra función va a **devolver algún valor** emplearemos la palabra reservada **return**, para hacerlo. Ejemplo:

```
function nombreFunción ( [parámetro1]...[parámetroN] ) {  
  
    // Sentencias  
  
    return valor;  
  
}
```

Los nombres que puedes asignar a una función, tendrán las mismas restricciones que tienen los elementos HTML y las variables en JavaScript. Deberías asignarle un nombre que realmente la identifique, o que indique qué tipo de acción realiza. **Puedes usar palabras compuestas como chequearMail o calcularFecha**, y fíjate que **las funciones suelen llevar un verbo**, puesto que las funciones son elementos que realizan acciones.

Una recomendación que te hacemos, es la de que las funciones sean muy específicas, es decir que no realicen tareas adicionales a las inicialmente propuestas en esa función.

Para realizar una llamada a una función lo podemos hacer con:

```
nombreFuncion( ); // Esta llamada ejecutaría las sentencias programadas dentro de la función.
```

Otro ejemplo de uso de una función en una asignación:

```
variable=nombreFuncion( ); // En este caso la función devolvería un valor que se asigna a la variable.
```

Las funciones en JavaScript también son objetos, y como tal tienen métodos y propiedades. Un método, aplicable a cualquier función puede ser **toString()**, el cuál nos devolverá el código fuente de esa función.

Créditos de la imagen.

Autoría: NMC Virtual Worlds.

Licencia: CC BY 2.0.

