Desarrollo Web en Entorno Cliente (2ª parte)

JavaScript

Funciones

Elementos de una función

Un función consta de:

- palabra reservada **function**
- identificador o nombre de la función
- parámetros
- cuerpo o instrucciones
- valor devuelto

Las funciones tienen que declararse antes de usarlas.

Una función se puede llamar desde cualquier bloque de código, esto incluye a las propias funciones.

Sintaxis de una función

Sintaxis:

function nombre(*parámetros*) {
    ... código o instrucciones ...
    return valor_devuelto;
}

No son obligatorias todas las partes. Por ejemplo, esta función no tiene bloque de instrucciones:

```
function porTres(a){
    return 3*a;
}
```

Función anónima

Es otra forma de crear funciones.

Es un tipo de función que no tiene nombre y se asigna a una variable.

Ejemplo:

```
var multPorTres = function(a){
        return 3*a;
}
```

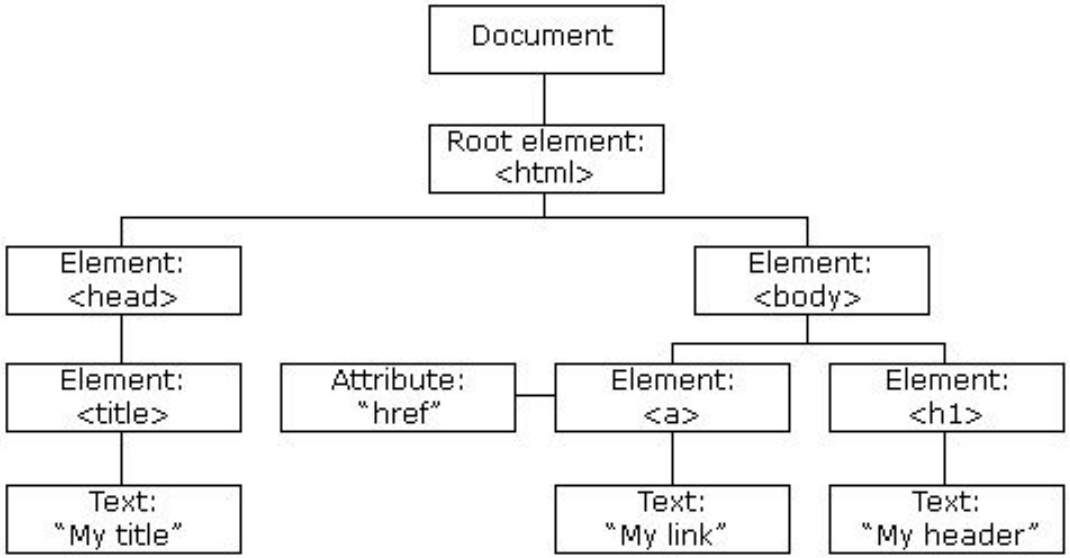
Se usa como si fuera una función:

```
console.log(multPorTres(5)); //saca 15 por consola
```

DOM (Document Object Model)

Definición

El DOM es la estructura que crea un navegador al descargar una página HTML. Esta estructura se puede manipular desde Javascript.



Acceso a elementos

document es un objeto que tiene varios métodos para acceder a los elementos de la página HTML.

Cada elemento de la página HTML es un objeto con propiedades y métodos.

Las propiedades pueden modificarse con un valor que sea aceptado.

Acceso a elementos

Los métodos de document:

document.getElementById -> permite recuperar los elementos que tienen un id declarado, este id se pasa como parámetro al método. Devuelve el objeto si lo encuentra, sino null.

document.getElementsByTagName -> devuelve un array de todos los elementos con la misma etiqueta.

document.getElementsByClassName -> devuelve un array de todos los elementos con el mismo atributo class.

Cambiar elementos

element.innerHTML -> es el contenido de un elemento, se puede leer o modificar.

element.attribute -> attribute es el nombre del atributo de un elemento, se puede leer o modificar.

element.style.property -> se refiere a las propiedades de los estilos css de los elementos.

Eventos

Definición

Los eventos son acciones dentro del navegador que se pueden controlar. Por ejemplo:

Cuando una página se carga totalmente
Cuando una imagen se carga totalmente
Cuando un usuario hace clic con el ratón en un elemento
Cuando el ratón pasa por encima de un elemento
Cuando un campo de entrada cambia
Cuando un usuario pulsa una tecla

Eventos que llaman a funciones

La habitual es que los eventos ejecuten funciones de javascript.

Por ejemplo:

window.onload=mifuncion;

Este evento se dispara cuando se carga totalmente la página y produce la ejecución de la función **mifuncion**.

Eventos de elementos

Los elementos pueden disparar eventos, por ejemplo, este evento se dispara cuando se hace click en el botón.

```
<button onclick="myFunction()">Click me</button>
```

No tiene porqué ser un botón, puede ser otro elemento:

```
<h1 onclick="myFunction()">Click me</h1>
```

Ajax

Definición

Es la tecnología que nos permite acceder a recursos de un servidor desde javascript, es decir, desde código javascript se pueden hacer peticiones recursos de un servidor y manejar la respuesta recibida desde el servidor.

Esto permite, junto con el DOM, pedir información al servidor y añadirla a la página sin necesidad de hacer recargas.

Estructura AJAX

```
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML = this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
```

En color azul es el código que hace la petición al servidor.
En color rojo es el código que maneja la respuesta del servidor.
En este ejemplo se pide al servidor un fichero de texto y lo que devuelve el servidor se introduce dentro de un elemento que se identifica por “demo”.

Peticiones GET

Las peticiones GET se construyen sobre la misma petición como una cadena de texto. Por ejemplo, si quiero enviar como parámetros el nombre y la edad a un recurso php se construye así:

```
$nombre=”pepe”;
$edad=”37”;
xhttp.open("GET", "inserta.php?nom=”.$nombre.”&eda=”.$edad, true);
xhttp.send();
```

Peticiones POST

Las peticiones POST se construyen sobre el método send. Por ejemplo, si quiero enviar como parámetros el nombre y la edad a un recurso php se construye asi:

```
$nombre=”pepe”;
$edad=”37”;
xhttp.open("POST", "inserta.php”, true);
xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xhttp.send(nom=”.$nombre.”&eda=”.$edad);
```