

# Estático versus dinámico

Repasando los listados 1,2 y 3 veremos que el código de creación de los objetos está entre marcas de <script>.

Cuando Javascript está escrito de esta forma sin estar dentro de un método o función, se denomina Javascript estático.

Esto quiere decir que el código se ejecutará en algún momento antes de que se muestre la página al usuario. (No está correctamente especificado en la documentación cuando este código se ejecuta, y además los navegadores muchas veces interpretan su ejecución de diferente forma).

De todas formas tenemos garantizado que el código se ejecutará antes de que nuestros usuarios puedan interactuar con la página.

Ésta es la forma más normal de creación de un objeto XMLHttpRequest por la mayoría de los programadores de Ajax.

Dicho ésto en el listado 4 tenemos un ejemplo de creación del objeto dentro de una función.

## Listado 4. Creación de XMLHttpRequest en una función.

```
<script language="javascript">

var request;

function createRequest() {
  try {
    request = new XMLHttpRequest();
  } catch (trymicrosoft) {
    try {
      request = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (othermicrosoft) {
      try {
        request = new ActiveXObject("Microsoft.XMLHTTP");
      } catch (failed) {
        request = false;
      }
    }
  }
}

if (!request)
  alert("Error inicializando XMLHttpRequest!");
}
</script>
```

Con el código de esta forma necesitaremos llamar a esta función antes de realizar cualquier tipo de trabajo con Ajax. Véase el listado 5.

## Listado 5. Método de creación de XMLHttpRequest

```
<script language="javascript" type="text/javascript">

var request;

function createRequest() {
  try {
    request = new XMLHttpRequest();
  } catch (trymicrosoft) {
    try {
      request = new ActiveXObject("Msxml2.XMLHTTP");
    } catch (othermicrosoft) {
      try {
        request = new ActiveXObject("Microsoft.XMLHTTP");
      } catch (failed) {
        request = false;
      }
    }
  }
}
}
```

```
    }

    if (!request)
        alert("Error inicializando XMLHttpRequest!");
    }

    function getCustomerInfo() {
        createRequest();
        // Hacer algo con la solicitud recién llamada.
    }
</script>
```

En lo que respecta a este código ? y la razón por la que muchos programadores de Ajax no utilizan este método ? es que retrasa la muestra de posibles errores.

Por ejemplo: supongamos que tenemos un formulario con 10 o 15 campos, selecciones, etc.. y queremos usar Ajax cuando el usuario introduzca un valor en el campo 14 (al final del formulario). En ese momento llamaremos a la función `getCustomerInfo()` para intentar crear el objeto `XMLHttpRequest`, y supongamos que de un fallo. Entonces se mostrará una alerta al usuario indicando que no podrá usar esta aplicación y todo el tiempo que el usuario ha empleado cubriendo el formulario no habrá servido de nada, con el consiguiente cabreo por parte del cliente :o)

Si por el contrario hubiéramos empleado el código estático, habríamos evitado que nuestro cliente cubriera el formulario ya que cuando nuestro cliente entra en la aplicación ya se le habría informado del error antes de que comenzara a usar dicha aplicación y ya dependería del cliente el solucionar dicho problema (generalmente actualizando su navegador web).