APRENDERAPROGRAMAR.COM

JERARQUÍA DE OBJETOS JAVASCRIPT. FORMS, ELEMENTS, IMAGES, LINKS. NAVIGATOR: USERAGENT, GEOLOCATION, ONLINE. (CU01170E)

Sección: Cursos
Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº70 del Tutorial básico "JavaScript desde cero".

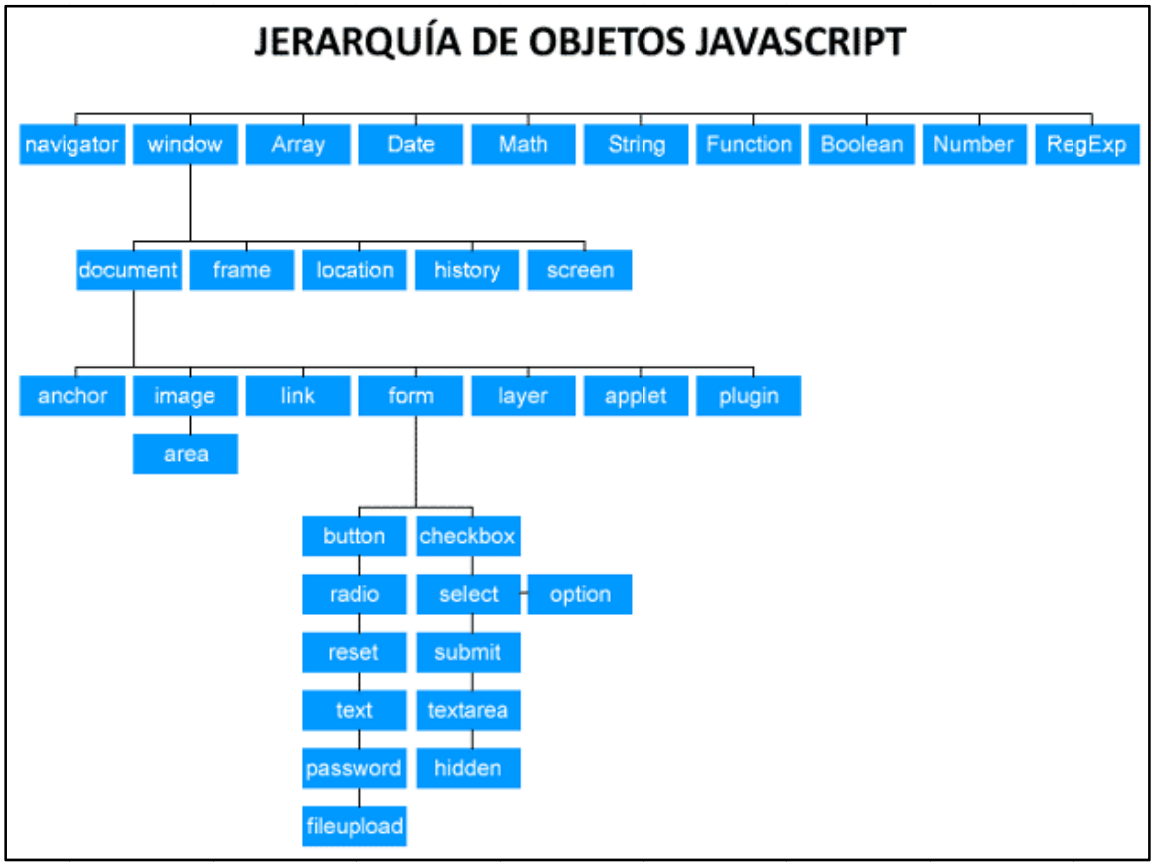
Autor: César Krall

JERARQUÍA DE OBJETOS JAVASCRIPT

Recordemos que window, el objeto global en JavaScript, tiene sus propiedades y métodos. La función que venimos usando para mostrar mensajes por pantalla, alert, es un método de window. Cuando escribimos alert('Hola'); en realidad estamos invocando window.alert('Hola');

Es decir, alert y window.alert son lo mismo para el navegador. El hecho de que no sea necesario escribir window cuando escribimos alert obedece a que si se invoca una función que no ha sido definida de otra manera, se considera que es un método del objeto global, es decir, un método de window.

JavaScript define una jerarquía de objetos que podemos reflejar de forma aproximada en un esquema similar al siguiente:



En estas jerarquía podemos pensar en objetos predefinidos de JavaScript (como Math, Date, String, RegExp, etc., incluido el objeto window).

También podemos pensar en el objeto document que contiene toda la información relativa a la estructura del documento HTML, lo que hemos llamado el DOM. No obstante, debemos diferenciar

entre la jerarquía del DOM, teniendo en cuenta que el DOM existe de forma independiente a JavaScript, y la jerarquía de objetos JavaScript, aunque guarden cierta similitud organizativa.

Podemos ver una página web como una colección de objetos. Por ejemplo, para JavaScript un formulario es un objeto, una imagen es un objeto, etc.

Los objetos tienen propiedades, métodos y eventos asociados.

Los objetos se organizan conforme a una jerarquía de forma que heredan métodos o propiedades de sus objetos padre, e incluso el nombre de un objeto se crea a partir de sus objetos padre.

Todo documento HTML dispone de los siguientes objetos en la jerarquía de objetos JavaScript:

navigator: tiene propiedades relacionadas con el nombre y la versión del navegador, protocolos de transferencia permitidos por el navegador (mime types) y sobre plugins instalados.

window: considerado habitualmente el objeto global o de máximo nivel. Tiene propiedades relacionadas con la ventana del navegador. En caso de uso de frames ("subventanas") hay un objeto window por cada "ventana hija" que exista.

document: tiene propiedades relacionadas con el documento como título, links, formularios, etc.

location: tiene propiedades relacionadas con la URL actual.

history: tiene propiedades relacionadas con URLs previamente visitadas.

Además de estos objetos en el documento HTML existirán más objetos según su contenido: objetos imágenes, objetos link, objetos formulario, etc.

CÓMO SE NOMBRAN LOS OBJETOS EN LA JERARQUÍA JAVASCRIPT

Para nombrar los objetos en la jerarquía de objetos JavaScript debemos tener en cuenta las siguientes reglas:

1) El nombre de un objeto que desciende de otro en la jerarquía JavaScript incluye el nombre de los objetos padre de la jerarquía. Por ejemplo el objeto document podemos nombrarlo como window.document. Un objeto hijo es a su vez un objeto y una propiedad del objeto padre. document es a su vez un objeto y una propiedad de window.

2) Dado que todos los objetos que podemos usar en el código descienden de window, normalmente omitiremos el uso de window a la hora de nombrar un objeto. Por eso escribiremos por ejemplo document.body en lugar de window.document.body, aunque ambas formas son válidas.

3) JavaScript organiza de forma automática ciertos objetos de naturaleza "múltiple" en arrays. Por ejemplo, una página web puede contener varios formularios. JavaScript, de forma automática, crea un array de objetos cuyo nombre es forms, siendo el primer formulario el de índice 0 y sucesivamente el 1, 2, etc. representan los siguientes formularios que aparezcan en el documento HTML por orden de aparición. Nos podemos referir a un formulario como window.document.forms[0], o más frecuentemente: document.forms[0]

Entre los arrays de objetos que crea JavaScript automáticamente tenemos:

forms: array con todos los formularios existentes en el documento HTML.

elements: array para cada formulario con los objetos que conforman dicho formulario (esto comprende objetos text, button, checkbox, hidden, radio, textarea, etc.). Si el primer formulario de una web tiene 3 input de tipo texto nos podemos referir al último de ellos como forms[0].elements[2]

images: array con todas las imágenes existentes en el documento HTML.

links: array con todos los links (tag HTML a) existentes en el documento HTML.

Para comprobar cómo podemos acceder a los objetos en la jerarquía de JavaScript escribe este código y comprueba los resultados (hemos incluido la ruta de dos imágenes, cámbiala si es necesario):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html><head><title>Ejemplo aprenderaprogramar.com</title><meta charset="utf-8">
<script type="text/javascript">
function ejemplo() {
window.document.images[0].style.border = 'solid blue 10px';
document.images[1].style.border = 'solid red 10px';
document.links[0].style.color = 'grey';
}
</script>
</head>
<body><div id="cabecera"><h2>Cursos aprenderaprogramar.com</h2><h3>Ejemplos JavaScript</h3></div>
<div style="color:blue; margin:20px;" id ="pulsador" onclick="ejemplo()"><a href="#"> Probar</a> </div>
<img src="http://i.imgur.com/afC0L.jpg" alt="Notepad++" title="Notepad++, un útil editor de texto" />
<img src="http://www.crimsoneditor.com/images/logo.jpg" alt="Crimson" title="Crimson, un editor de texto alternativo"
/>
</body>
</html>
```

El resultado esperado es que al pulsar sobre el texto “Probar”, la primera imagen adquiere un borde azul de 10 píxeles de grosor, la segunda imagen un borde rojo del mismo grosor y el texto del link queda de color gris. La sintaxis window.document.images[1] o simplemente document.images[1] nos da acceso a la segunda imagen existente en el documento HTML.

A través de la jerarquía de objetos de JavaScript podemos acceder a las propiedades de cada objeto por separado. En el ejemplo anterior hemos visto cómo cambiar propiedades de estilo CSS, pero también podríamos acceder y modificar otras propiedades. Por ejemplo, introduce estas líneas y comprueba el resultado:

```
var rut = 'http://aprenderaprogramar.com/images/stories/Libros/portada_libro_aprender_programar_java_lowres.jpg';
document.images[1].src = rut;
```

El resultado esperado es que al hacer click en el texto <<Probar>> una de las imágenes existentes deje de mostrarse y pase a mostrarse otra, debido a que hemos accedido al objeto que representa la imagen y hemos modificado su propiedad src (que define la ruta de la imagen).

El acceso a formularios y elementos de formularios suele ser de gran importancia con JavaScript, por lo que lo estudiaremos por separado.

EL OBJETO NAVIGATOR

Podemos recuperar un objeto de tipo Navigator haciendo la invocación objetoNav = window.navigator; o dado que window no es necesario, simplemente escribiendo navigator.

Ese objeto Navigator dispondrá de propiedades y métodos que nos pueden ser útiles, aunque de momento muchas de las propiedades y métodos son experimentales o no responden de la misma manera en los distintos navegadores. Vamos a citar aquí algunas de las propiedades:

PROPIEDAD	UTILIDAD	EJEMPLOS aprenderaprogramar.com
userAgent	Devuelve una cadena de texto representando el agente que se está empleando en la navegación. Puede identificar el navegador empleado, pero no es seguro porque puede configurarse para falsearlo.	alert(window.navigator.userAgent); // Por ejemplo Mozilla/5.0 (Windows NT 6.0; rv:31.0) Gecko/20100101 Firefox/31.0
battery	Devuelve un objeto BatteryManager que puede usarse para obtener información sobre el estado de la batería. No disponible para todos los navegadores.	alert('¿Cargando?: '+window.navigator.battery.charging); // ¿Cargando?: true si está cargando
geolocation	Devuelve un objeto Geolocation que puede usarse para obtener información sobre la ubicación (latitud, longitud) desde donde el usuario navega. Algunos navegadores no responden bien. Otros restringen o piden permiso al usuario por considerar que puede atentar contra su privacidad.	navigator.geolocation.getCurrentPosition (funcionSiExito, funcionSiFallo, arrayDeOpciones);

language	Devuelve un string representativo del lenguaje del navegador. Algunos navegadores no disponen de language pero en su lugar tienen disponible userLanguage. Otros navegadores no reconocen ni una ni otra forma.	var lenguaje = navigator.language || navigator.userLanguage;
		alert ('Código lenguaje navegador: '+lenguaje);
		//Código lenguaje navegador: es-ES por ejemplo

online	Devuelve un valor booleano (true o falso) que indica si se está o no con conexión a internet. Combinado con eventos (window.ononline y window.onoffline), puede servir para mostrar un mensaje de alerta si se pierde o recupera la conexión.	alert ('¿Estamos online?: '+window.navigator.onLine); // ¿Estamos online?: true si estamos conectados a internet

Entre los métodos de los objetos Navigator únicamente citaremos a Navigator.vibrate(), que genera la vibración de aquellos dispositivos (como smartphones) que admiten la vibración.

EJERCICIO

Usando la propiedad userAgent de los objetos Navigator, determina el navegador que está usando el usuario y muestra un mensaje por pantalla informando de ello. El resultado debe ser del tipo: <<Estás usando: nombreNavegador>>, donde nombreNavegador será Google Chrome, Apple Safari, Opera, Mozilla Firefox, Microsoft Internet Explorer ó Desconocido. Resuélvelo de dos maneras distintas:

a) Usando expresiones regulares.

b) Usando el método indexOf de los objetos tipo String de JavaScript.

Para comprobar si tus respuestas y código son correctos puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01171E

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente: http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206