

JavaScript

Introducción, Operadores básicos y Programación orientada a objetos

Jhon Gesell Villanueva Portella¹

¹Toulouse Lautrec.
Diseño.
Diseño y Desarrollo para Medios Digitales.

08 de julio de 2020

Contenido

1 Introducción

2 Operaciones Básicas

3 POO

Contenido

1 Introducción

2 Operaciones Básicas

3 POO

Introducción

Con JavaScript, puede hacer que sus páginas web cobren vida haciendo que interactúen con el usuario. JavaScript es un lenguaje de programación completo y, como tal, todo es posible. Web Programming with html5, css, and javascript de John Dean (2019)

Historia de JavaScript 01/04

La primera versión de HTML, diseñada por Tim Berners-Lee de 1989 a 1991, era de naturaleza bastante estática. Excepto por los saltos de enlace con el elemento `a`, las páginas web simplemente mostraban contenido, y el contenido se corrigió. En 1995, el fabricante dominante de navegadores era Netscape, y uno de sus empleados, Brendan Eich, pensó que sería útil agregar funcionalidad dinámica a las páginas web. Así que diseñó el lenguaje de programación JavaScript, que agrega funcionalidad dinámica a las páginas web cuando se usa junto con HTML. Por ejemplo, JavaScript proporciona la capacidad de actualizar el contenido de una página web cuando ocurre un evento, como cuando un usuario hace clic en un botón. También proporciona la capacidad de recuperar la entrada de un usuario y procesar esa entrada. Web Programming with html5, css, and javascript de John Dean (2019)

Historia de JavaScript 02/04

Eich tardó solo 10 días en mayo de 1995 en implementar el lenguaje de programación JavaScript, una hazaña realmente notable. Marc Andreessen, uno de los fundadores de Netscape, originalmente nombró el nuevo lenguaje Mocha y luego LiveScript. Pero para fines de marketing, Andreessen realmente quería el nombre JavaScript. En ese momento, la industria del software estaba entusiasmada con el nuevo lenguaje de programación, Java. Andreessen pensó que todos los devotos del carro de Java gravitarían a su nuevo lenguaje de programación de navegador si tuviera el nombre Java. En diciembre de 1995, Andreessen consiguió su deseo cuando Netscape obtuvo una licencia de marca registrada del fabricante de Java, Sun Microsystems, y el nombre de LiveScript se cambió a JavaScript. Desafortunadamente, muchas, muchas personas a lo largo de los años han cometido el error de suponer que JavaScript es igual a Java o muy cercano a él. No se deje engañar por el nombre: JavaScript no es tan similar a Java. En realidad, C ++ y otros lenguajes de programación populares están más cerca de Java que JavaScript.

Web Programming with html5, css, and javascript de John Dean (2019)

Historia de JavaScript 03/04

En 1996, Netscape envió JavaScript a la organización de estándares Ecma International para promover la influencia de JavaScript en todos los navegadores (no solo en el navegador de Netscape). Ecma International utilizó JavaScript como base para crear el estándar ECMAScript. Como se esperaba, ECMAScript ahora sirve como el estándar para los lenguajes de programación interactivos integrados en todos los navegadores populares de hoy. En el momento de la impresión de este libro, la versión más reciente de ECMAScript es la versión 7, publicada en 2016. Encontrar el nombre de ECMAScript fue un proceso difícil, con diferentes fabricantes de navegadores que tenían opiniones muy opuestas. El creador de JavaScript, Brendan Eich, ha declarado que el resultado, ECMAScript, es un nombre comercial no deseado que suena como una enfermedad de la piel". Web Programming with html5, css, and javascript de John Dean (2019)

Historia de JavaScript 04/04

En 1998, Netscape formó la comunidad de software libre Mozilla, que finalmente implementó Firefox, uno de los principales navegadores de hoy. Brendan Eich se mudó a Mozilla, donde él y otros continuaron actualizando JavaScript a lo largo de los años, siguiendo el estándar ECMAScript establecido por Ecma International.

Otros fabricantes de navegadores admiten sus propias versiones de JavaScript. Para sus navegadores Internet Explorer y Edge, Microsoft usa JScript. Para su navegador Chrome, Google usa el motor V8 JavaScript. Afortunadamente, todos los fabricantes de navegadores intentan seguir el estándar ECMAScript, por lo que para la mayoría de las tareas, los programadores pueden escribir una versión de su código y funcionará para todos los navegadores diferentes. En este libro, nos atenemos al código ECMAScript estándar que funciona igual en todos los navegadores. Como con casi todos en la comunidad de programación web, nos referimos a nuestro código como JavaScript, a pesar de que JavaScript es solo una de varias implementaciones de ECMAScript (JavaScript es la implementación utilizada en Firefox de Mozilla). Web Programming with html5, css, and javascript

de John Dean (2019)

Hola mundo en mi primera página web 01/02



FIGURE 8.1 Initial display and what happens after the user clicks the button on the Hello web page

Figura: Pantalla del 'hola mundo'.

Hola mundo en mi primera página web 02/02

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="author" content="John Dean">
<title>Hello</title>
<script>
  function displayHello() {
    var msg;
    msg = document.getElementById("message");
    msg.innerHTML = "<h1>Hello, world!</h1>";
  }
</script>
</head>

<body>
<h3 id="message">
  To see the traditional first-program greeting, click below.
</h3>
<input type="button" value="Click Me!" onclick="displayHello();">
</body>
</html>
```



FIGURE 8.2 Source code for Hello web page

Figura: Código del 'hola mundo'.

Botones 01/03

Existen diferentes tipos de botones, cada uno con su propia sintaxis. Para simplificar las cosas, comenzaremos con un solo tipo de botón, y aquí está su sintaxis:

```
<input type="button"  
  value="button-label"  
  onclick="click-event-handler" >
```

Figura: Sintaxis.

Tenga en cuenta que el código es un elemento vacío que utiliza la etiqueta de entrada. Como su nombre lo indica, la etiqueta de entrada implementa elementos que manejan la entrada del usuario. Es posible que no piense en un botón como una entrada del usuario, pero lo es: el usuario elige hacer algo haciendo clic en un botón. Más adelante, presentaremos otros elementos de entrada del usuario (por ejemplo, controles de texto y casillas de verificación) que también usan la etiqueta de entrada.

Botones 02/03

Como lo hemos hecho a lo largo del libro al introducir nuevas construcciones, el fragmento de código anterior muestra solo los detalles de sintaxis más importantes, para que no se sienta abrumado con demasiado para recordar. Mostramos los atributos más comunes del elemento de entrada: tipo, valor y onclick. Observe cómo el elemento de entrada en la parte inferior de la Figura 8.2 sigue este patrón de sintaxis e incluye esos tres atributos.

El elemento de entrada se usa para diferentes tipos de entrada de usuario, y su atributo de tipo especifica qué tipo de entrada de usuario. Más formalmente, el atributo type especifica el tipo de control que se está implementando, donde un control es una entidad de entrada del usuario, como un botón, control de texto o casilla de verificación. En el código fuente de la página web Hello, tenga en cuenta que el atributo type obtiene el botón de valor, que le indica al navegador que muestre un botón. Si no proporciona un atributo de tipo, el navegador mostrará un control de texto, porque ese es el tipo de control predeterminado para el elemento de entrada. Describiremos los controles de texto más adelante en este capítulo. Por ahora, solo sepa que un control de texto es un cuadro en el que un usuario puede ingresar texto, y así es como se ve un control de texto (completado) (con un mensaje a su izquierda):

Web Programming with html5, css, and javascript de John Dean (2019)

Botones 03/03

First Name:

Figura: El usuario puede ingresar texto.

Debido a que utiliza un cuadro, muchos desarrolladores web se refieren a los controles de texto como “cuadros de texto”. Usamos el término “control de texto” porque es el término utilizado con mayor frecuencia por las organizaciones de estándares HTML. El atributo de valor del elemento de entrada especifica la etiqueta del botón. Si no proporciona un atributo de valor, el botón no tendrá etiqueta. Si desea un botón sin etiqueta, en lugar de simplemente omitir el atributo de valor, le recomendamos que especifique `value = .` Esa es una forma de autodocumentación y hace que su código sea más comprensible.

Web Programming with html5, css, and javascript de John Dean (2019)

Contenido

1 Introducción

2 Operaciones Básicas

3 POO

Introducción

Comenzamos el capítulo con una discusión sobre el objeto de la ventana, que permite al programador recuperar información sobre la ventana del navegador: la URL de la página web actualmente activa, el tipo de navegador, etc. Luego pasamos a los cuadros de diálogo, que proporcionan un medio crudo, pero efectivo, para obligar al usuario a responder una pregunta. Introducimos la declaración `if` para hacer cosas diferentes, dependiendo de la respuesta del usuario a una pregunta. Luego hablamos de cadenas y números. Para obligar a los usuarios a proporcionar información que tenga sentido, describimos técnicas de validación de restricción de entrada. Finalmente, hablamos de varios operadores que pueden usarse con la declaración `if` para distinguir entre diferentes situaciones.

Web Programming with html5, css, and javascript de John Dean (2019)

Objeto Ventana 01/02

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<meta name="author" content="John Dean">
<title>Window Information</title>
<script>
  // This function displays information about the window.

  function displayWindowInfo() {
    var msg1, msg2, msg3; // they hold window information
    msg1 = document.getElementById("location-info");
    msg1.innerHTML =
      "This page's full URL: " + window.location.href + "<br>" +
      "This page's protocol: " + window.location.protocol;
    msg2 = document.getElementById("browser-info");
    msg2.innerHTML =
      "Browser name: " + window.navigator.userAgent;
    msg3 = document.getElementById("screen-info");
    msg3.innerHTML =
      "Screen width: " + window.screen.availWidth +
      " pixels<br>" +
      "Screen height: " + window.screen.availHeight + " pixels";
  } // end displayWindowInfo
</script>
</head>

<body onload="displayWindowInfo();">
<h2>Getting Information From the Window Object</h2>
<p id="location-info"></p>
<p id="browser-info"></p>
<p id="screen-info"></p>
</body>
</html>
```

Retrieves an object with information about web pages URL.

Retrieves an object with information about user's monitor.

Retrieves an object with information about user's browser.

FIGURE 9.2 Source code for Window Information web page

Figura: Código fuente de una ventana.

Objeto Ventana 02/02

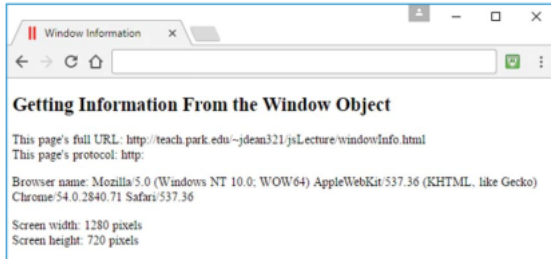


FIGURE 9.3 Window Information web page

Figura: Información de una ventana.

Web Programming with html5, css, and javascript de John Dean (2019)

Contenido

1 Introducción

2 Operaciones Básicas

3 POO

Introducción

Se definirá estructuras para nuevos tipos de objetos y usará esas estructuras para crear los objetos. Las estructuras que defina y los objetos que cree le ayudarán a organizar los datos dentro del código JavaScript de una página web. El uso de objetos para organizar datos se conoce como programación orientada a objetos (OOP). Deberá usar OOP cuando haya una cantidad significativa de datos que deba organizarse. En este capítulo, las páginas web usan más datos que en capítulos anteriores, y OOP ayudará a que el código sea más fácil de entender.

Web Programming with html5, css, and javascript de John Dean (2019)

Generalidades de la POO

Como puede imaginar, si tiene una gran tarea de programación, existe un gran potencial para crear un nido de códigos intrincados. 1 En la década de 1970, los diseñadores del lenguaje de programación SmallTalk abordaron este problema de código complicado al acuñar el término programación orientada a objetos y convirtiéndolo en una parte central de su nuevo lenguaje de programación. El objetivo del paradigma OOP es que los programas modelen cómo las personas normales piensan sobre un problema y su solución. Al pensar en un problema, las personas tienden a centrarse en las cosas que lo componen. Con OOP, esas cosas se llaman objetos. Por lo general, son entidades físicas, pero también pueden ser entidades conceptuales. Como programador de OOP, una vez que ha identificado las cosas que desea modelar, identifica sus propiedades y comportamientos básicos. Se agrupan las propiedades y los comportamientos de cada cosa en una estructura coherente llamada objeto. Al escribir un programa OOP, usted define objetos, los crea y hace que interactúen entre sí. Por ejemplo, si está escribiendo un programa para asignar cursos a las aulas, probablemente necesitará objetos del curso y objetos de la sala. Los objetos del curso contendrían propiedades del nombre del curso y del número de estudiantes, y los objetos de la sala contendrían propiedades de ubicación y capacidad. Los comportamientos de un objeto se refieren a las actividades asociadas con el objeto. Para un objeto del curso, probablemente necesitará un comportamiento que ajuste la propiedad del número de estudiantes del curso, ya que los estudiantes agregan y abandonan los cursos antes del primer día de clase. Con JavaScript, implementa los comportamientos de un objeto como métodos.

Web Programming with html5, css, and javascript de John Dean (2019)

Generalidades de la POO

Una de las características fundamentales de OOP es la encapsulación. En general, la encapsulación es cuando algo está envuelto dentro de una cubierta protectora. Cuando se aplica a objetos, la encapsulación significa que los datos de un objeto están protegidos al estar ocultos dentro del objeto. Con datos ocultos, el resto del programa no puede acceder a los datos de un objeto directamente; el resto del programa se basa en los métodos del objeto para acceder a los datos. Acceder a los datos de un objeto se refiere a leer los datos o modificarlos. Suponiendo que los métodos de un objeto están bien escritos, los métodos aseguran que se acceda a los datos de manera adecuada. Al limitar el acceso al acceso "apropiado", eso hace que sea más difícil desordenar los datos de un programa, y eso genera menos errores. ¡Hurra! Ver FIGURA 11.1. Muestra cómo los métodos de un objeto forman la interfaz entre los datos del objeto y el mundo exterior.

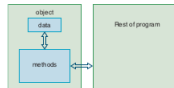


FIGURE 11.1 Methods provide an interface between data and the rest of the program

Figura: tex1.

Web Programming with html5, css, and javascript de John Dean (2019)

Classes, Constructores, Propiedades, nuevos Operadores y Metodos

```
class Point {  
  constructor(x, y) {  
    this.x = x;  
    this.y = y;  
  } // end constructor  
  
  value() {  
    return "(" + this.x + ", " + this.y + ")";  
  } // end value  
} // end class Point
```

Diagram labels and arrows:

- class heading (points to `class Point {`)
- constructor heading (points to `constructor(x, y) {`)
- Use "this" to define x and y properties for the class. (points to `this.x = x;` and `this.y = y;`)
- A method that returns a Point object's value. (points to `value() {`)

FIGURE 11.2 Bare-bones class definition for a Point class

Figura: tex1.

Web Programming with html5, css, and javascript de John Dean (2019)

Contenido

4 Referencias

Referencias



Web Programming with html5, css, and javascript

Dean, John.

Jones & Bartlett Learning (2019)

Referencias



Learn to Code HTML and CSS: Develop and Style Websites

Howe, Shay.

New Riders, 2014