



Tema

Segundo Parcial

Presenta

Jhon Gabriel Silva Tibaduiza

Docente

ALONSO GUEVARA PEREZ

Asignatura

Programación web
NRC: 7478

Colombia, Bogotá

Abril 28 de 2020.

JS Functions

Function Definitions:

Una función es una porción de código reusable que puede ser llamada en cualquier momento desde nuestro programa. De esta manera se evita la necesidad de estar escribiendo el mismo código una y otra vez. Esto nos ayudará a escribir código modular en JavaScript.

Antes de poder usar una función necesitamos definirla. La forma más común de definir una función es usando la palabra reservada "function" seguido de un único nombre de función, y de los parámetros deseados (puede ser sin parámetros), y un bloque de instrucciones dentro de llaves.

Function Parameters:

Hay dos tipos de parámetros que se pasan a la función, los parámetros de tipo de datos primitivos y los parámetros de tipo de datos no primitivos. Los tipos de datos primitivos son tipos de datos predefinidos, por ejemplo, entero, carácter y cadena son todos tipos de datos primitivos. Los tipos de datos no primitivos son creados por el usuario. A veces se denominan "variables de referencia" o "referencias de objeto", ya que hacen referencia a una ubicación de memoria, que almacena los datos. Por ejemplo, los objetos definidos por el usuario, las matrices son tipos de datos no primitivos.

Function Invocation:

El operador de invocación es un par de paréntesis que sigue a cualquier expresión que produce un valor de función. Los paréntesis pueden contener cero o más expresiones, separadas por comas. Cada expresión produce un valor de argumento. Cada uno de los valores del argumento se asignará a los nombres de los parámetros de la función. No hay error de tiempo de ejecución cuando el número de argumentos y el número de parámetros no coinciden. Si hay demasiados valores de argumento, se ignorarán los valores de argumento adicionales. Si hay muy pocos valores de argumento, el valor indefinido se sustituirá por los valores faltantes. No hay verificación de tipo en los valores del argumento: cualquier tipo de valor se puede pasar a cualquier parámetro.

Function Call:

La definición de una función simplemente nombra la función y especifica qué hacer cuando se llama a la función. Llamar a la función realmente ejecuta las instrucciones escritas dentro de la función con los parámetros indicados. Los argumentos de una función no se limitan a cadenas y números, puede pasar objetos completos a una función.

Function Apply:

`apply ()` invoca la función especificada como si fuera un método de `thisobj`, pasándole los argumentos contenidos en la matriz `args`. Devuelve el valor devuelto por la invocación de la función. Dentro del cuerpo de la función, la palabra clave `this` se refiere al objeto `thisobj`.

Function Closures:

Un closure es una función que es libre de variables, esto quiere decir que las variables de la función padre funcionan. La función padre() crea una variable local y una función closure(). Esta función interna es un closure y solo está disponible dentro de padre(). A diferencia de padre() esta función no tiene variables locales y usa las declaradas dentro de padre().

JS Ajax

Ajax Intro:

AJAX permite que las páginas web se actualicen de forma asíncrona mediante el intercambio de datos con un servidor web detrás de escena. Esto significa que es posible actualizar partes de una página web, sin volver a cargar toda la página.

Con Ajax podemos:

- Leer datos de un servidor web: después de que la página se haya cargado
- Actualizar una página web sin volver a cargar la página
- Enviar datos a un servidor web: en segundo plano

Ajax XMLHttpRequest:

Todos los navegadores modernos admiten el objeto XMLHttpRequest.

El objeto XMLHttpRequest se puede usar para intercambiar datos con un servidor web detrás de escena. Esto significa que es posible actualizar partes de una página web, sin volver a cargar toda la página.

XMLHttpRequest Object Methods

| Method | Description |
|--|--|
| new XMLHttpRequest() | Creates a new XMLHttpRequest object |
| abort() | Cancels the current request |
| getAllResponseHeaders() | Returns header information |
| getResponseHeader() | Returns specific header information |
| open(<i>method</i> , <i>url</i> , <i>async</i> , <i>user</i> , <i>psw</i>) | Specifies the request <i>method</i> : the request type GET or POST <i>url</i> : the file location <i>async</i> : true (asynchronous) or false (synchronous) <i>user</i> : optional user name <i>psw</i> : optional password |
| send() | Sends the request to the server Used for GET requests |
| send(<i>string</i>) | Sends the request to the server. Used for POST requests |
| setRequestHeader() | Adds a label/value pair to the header to be sent |

Ajax Request:

| Method | Description |
|---------------------------------------|---|
| <code>open(method, url, async)</code> | Specifies the type of request <i>method</i> : the type of request: GET or POST <i>url</i> : the server (file) location <i>async</i> : true (asynchronous) or false (synchronous) |
| <code>send()</code> | Sends the request to the server (used for GET) |
| <code>send(string)</code> | Sends the request to the server (used for POST) |

Ajax Response

Una función de devolución de llamada es una función que se pasa como parámetro a otra función.

Si tiene más de una tarea AJAX en un sitio web, debe crear una función para ejecutar el objeto XMLHttpRequest y una función de devolución de llamada para cada tarea AJAX.

La llamada a la función debe contener la URL y qué función llamar cuando la respuesta esté lista.

Ajax Xml File:

XML es el acrónimo de Extensible Markup Language, es decir, es un lenguaje de marcado que define un conjunto de reglas para la codificación de documentos. ¿Te parece complicado? Entonces, vamos a simplificarlo.

El lenguaje de marcado es un conjunto de códigos que se pueden aplicar en el análisis de datos o la lectura de textos creados por computadoras o personas. El lenguaje XML proporciona una plataforma para definir elementos para crear un formato y generar un lenguaje personalizado.

Ajax PHP:

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores. Podemos saber algo más sobre la programación del servidor y del cliente en el artículo qué es DHTML.

Ajax ASP:

ASP (Active Server Pages) es la tecnología desarrollada por Microsoft para la creación de páginas dinámicas del servidor. ASP se escribe en la misma página web, utilizando el lenguaje Visual Basic Script o Jscript (Javascript de Microsoft).

Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la página ASP. Como la página resultante contiene únicamente código HTML, es compatible con todos los navegadores.

Ajax Database:

Como función, la opción ajax.data se puede utilizar para modificar los datos que DataTables envía al servidor tras una solicitud de Ajax, manipulando el objeto de datos original DataTables construye internamente o sustituyéndolo por completo.

Esto proporciona la capacidad de enviar información adicional al servidor en una solicitud de Ajax, con la función que se ejecuta en cada solicitud de Ajax, lo que permite que los valores se calculen dinámicamente. Por ejemplo, un valor podría leerse de un campo de entrada de texto para actuar como una opción de búsqueda adicional.

Ajax Applications:

Con esta función podemos realizar distintas aplicaciones de la función unas de ellas las mencionaremos a continuación:

- Solicitar datos de un servidor: después de que la página se haya cargado
- Recibir datos de un servidor: después de que la página se haya cargado
- Enviar datos a un servidor - en segundo plano

JS JSON

JSON Syntax:

La sintaxis JSON se deriva de la sintaxis de notación de objetos JavaScript:

- Los datos están separados por comas.
- Las llaves contienen objetos.
- Los corchetes contienen matrices.

JSON vs XML:

Tanto JSON como XML se pueden usar para recibir datos de un servidor web.

Los siguientes ejemplos JSON y XML definen un objeto de empleados, con una matriz de 3 empleados.

Para aplicaciones AJAX, JSON es más rápido y fácil que XML:

Usando XML

- Obtener un documento XML
- Use el DOM XML para recorrer el documento
- Extraer valores y almacenar en variables

Usando JSON

- Obtener una cadena JSON
- JSON.Parse la cadena JSON

JSON Data Types:

En JSON, los valores deben ser uno de los siguientes tipos de datos:

- Una cadena
- Números
- Un objeto (Objetos JSON)
- Una matriz
- Un booleano
- Nulo

JSON Parse:

Un uso común de JSON es intercambiar datos hacia / desde un servidor web.

Al recibir datos de un servidor web, los datos siempre son una cadena.

Analice los datos con `JSON.parse ()`, y los datos se convierten en un objeto JavaScript. Siempre que la respuesta del servidor se escriba en formato JSON, puede analizar la cadena en un objeto JavaScript.

JSON Stringify:

Un uso común de JSON es intercambiar datos hacia / desde un servidor web.

Al enviar datos a un servidor web, los datos deben ser una cadena.

Convierta un objeto JavaScript en una cadena con `JSON.stringify ()`. En JSON, los objetos de fecha no están permitidos. La función `JSON.stringify ()` convertirá cualquier fecha en cadenas.

JSON Objects:

Los objetos JSON están rodeados de llaves {}.

Los objetos JSON se escriben en pares clave / valor.

Las claves deben ser cadenas y los valores deben ser un tipo de datos JSON válido (cadena, número, objeto, matriz, booleano o nulo).

Las claves y los valores están separados por dos puntos.

Cada par clave / valor está separado por una coma.

JSON Arrays:

Las matrices en JSON son casi lo mismo que las matrices en JavaScript.

En JSON, los valores de la matriz deben ser de tipo cadena, número, objeto, matriz, booleano o nulo.

En JavaScript, los valores de la matriz pueden ser todos los anteriores, más cualquier otra expresión válida de JavaScript, incluidas las funciones, las fechas y las indefinidas.

JSON PHP:

Un uso común de JSON es leer datos de un servidor web y mostrar los datos en una página web.

Este capítulo le enseñará cómo intercambiar datos JSON entre el cliente y un servidor PHP.

PHP tiene algunas funciones integradas para manejar JSON.

Los objetos en PHP se pueden convertir a JSON utilizando la función PHP `json_encode()`

JSON HTML:

JSON se puede traducir fácilmente a JavaScript.

JavaScript se puede utilizar para crear HTML en sus páginas web.

JSON JSONP:

JSONP es un método para enviar datos JSON sin preocuparse por problemas entre dominios.

JSONP no usa el objeto `XMLHttpRequest`.

JSONP utiliza la etiqueta `<script>` en su lugar.

Solicitar un archivo de otro dominio puede causar problemas, debido a la política entre dominios.

Solicitar un script externo de otro dominio no tiene este problema.

JSONP usa esta ventaja y solicita archivos usando la etiqueta de script en lugar del objeto `XMLHttpRequest`.

JS vs jQuery

jQuery Selectors:

jQuery fue creado en 2006 por John Resig. Fue diseñado para manejar las incompatibilidades del navegador y para simplificar la manipulación del DOM HTML, el manejo de eventos, las animaciones y Ajax.

Durante más de 10 años, jQuery ha sido la biblioteca de JavaScript más popular del mundo.

Sin embargo, después de JavaScript Versión 5 (2009), la mayoría de las utilidades de jQuery se pueden resolver con unas pocas líneas de JavaScript estándar.

jQuery HTML:

HTML es un lenguaje de marcado que se utiliza para el desarrollo de páginas de Internet. Se trata de la sigla que corresponden a HyperText Markup Language, es decir, Lenguaje de Marcas de Hipertexto. Existen ciertas excepciones con el fin de las etiquetas, ya que algunas usualmente sólo se utilizan con el inicio de la etiqueta, ejemplos de esto son las etiquetas BR (salto de línea), IMG (etiqueta para poner una imagen), entre otras.

jQuery CSS:

CSS es un lenguaje utilizado en la presentación de documentos HTML. Un documento HTML viene siendo coloquialmente “una página web”. Entonces podemos decir que el lenguaje CSS sirve para organizar la presentación y aspecto de una página web. Este lenguaje es principalmente utilizado por parte de los navegadores web de internet y por los programadores web informáticos para elegir multitud de opciones de presentación como colores, tipos y tamaños de letra, etc.

jQuery DOM:

Es un acrónimo del término Document Object Model. Su trabajo es describir la estructura de un documento HTML y las relaciones entre los diferentes elementos como etiquetas, atributos, y textos de la página. Al agregar, eliminar o modificar elementos existentes en tu sitio web, estás creando estructuras que el navegador interpreta como el DOM. Entonces, si agrego una nueva etiqueta dentro de una etiqueta en mi página HTML, estoy añadiendo un nodo al DOM. Este nodo al ser agregado al DOM tiene diferentes relaciones, como hermanos y padres.

Referencias

<https://www.oreilly.com>

https://www.w3schools.com/js/js_jquery_dom.asp

Gao, Y., Shi, Z. J., Gao, S., & Mazda, A. (2005). *U.S. Patent No. 6,941,562*. Washington, DC: U.S. Patent and Trademark Office.

FARD, Amin Milani; MESBAH, Ali. Jsnoise: Detecting javascript code smells. En *2013 IEEE 13th International Working Conference on Source Code Analysis and Manipulation (SCAM)*. IEEE, 2013. p. 116-125.

Stefanov, S. (2008). *Object-Oriented JavaScript*. Packt Publishing Ltd.

Vinoski, S. (2006). Scripting JAX-WS [JavaScript]. *IEEE internet computing*, 10(3), 91-94.