

# JS Functions

*Una función es un bloque de código reutilizable que agrupa una secuencia de instrucciones para realizar una tarea específica. El código dentro del cuerpo de una función se ejecuta, o se ejecuta, solo cuando se llama a la función. Para llamar a una función en su código, escriba el nombre de la función seguido de paréntesis.*

## Function definitions

- Cuando definimos una función en javascript usamos la siguiente sintaxis.
- Palabra clave de función: **function**
- identificador: **NombredeFuncion**
- parámetros: **(parámetro1, parámetro2, ...)**
- cuerpo: **{código}**

EJEMPLO:

```
function saludo (name = 'Luis') {  
    console.log(`Hola, ${name}!`)  
}
```

*Figura 1. Mi autoria.*

## Parámetros de función

*Al declarar una función, podemos especificar sus parámetros. Los parámetros permiten que las funciones acepten entradas y realicen una tarea usando las entradas `saludo (nombre, apellido)` por ejemplo.*

## Invocación de funciones

*Cuando definimos nuestra función, podemos invocar la función escribiendo el identificador y los corchetes, y dentro del argumento de la función*

EJEMPLO

```
saludo('Estefania') // Output: Hola, Estefania!  
saludo() // Output: Hola, extraño!
```

*Figura 2. Mi autoria.*

## Función call

Es un método de JavaScript predefinido, que se utiliza para escribir métodos para diferentes objetos. Llama al método, tomando el objeto propietario como argumento. La palabra clave *this* se refiere al "propietario" de la función o el objeto al que pertenece.

EJEMPLO:

```
var persona = {  
  Nombre: function() {  
    return this.nombre + " " + this.apellido;  
  }  
}  
  
var persona1 = {  
  nombre: "Juan",  
  apellido: "Corrales"  
}  
  
var persona2 = {  
  nombre: "Estefania",  
  apellido: "Moreno"  
}  
  
persona.Nombre.call(persona1); // Devuelve " Juan Corrales "
```

Figura 3. [https://www.w3schools.com/js/js\\_function\\_call.asp](https://www.w3schools.com/js/js_function_call.asp).

## Función apply

Con el método *apply()*, puede escribir un método que pueda usarse en diferentes objetos.

**NOTA:** Aunque la sintaxis de esta función es casi idéntica a *call()*, la diferencia fundamental es que *call()* acepta una lista de argumentos, mientras que *apply()* acepta un simple array con los argumentos.

EJEMPLO:

```
var persona = {  
  Nombre: function(ciudad, pais) {  
    return this.nombre + " " + this.apellido + ", " + ciudad + ", " + pais;  
  }  
}  
  
var persona1 = {  
  firstName: "Juan",  
  LastName: "Corrales"  
}  
  
person.Nombre.apply(persona1, ["New York", "EEUU"]);  
  
// Devuelve "Juan Corrales, New York ,EEUU"
```

Figura 4. [https://www.w3schools.com/js/js\\_function\\_apply.aspequest\\_create.asp](https://www.w3schools.com/js/js_function_apply.aspequest_create.asp)

### Función closures

*Un cierre es la combinación de una función agrupada (encerrada) con referencias a su estado circundante (el entorno léxico). En otras palabras, un cierre le da acceso al alcance de una función externa desde una función interna. En JavaScript, los cierres se crean cada vez que se crea una función, en el momento de creación de la función.*

## JS AJAX

### AJAX Intro

*Ajax es un acrónimo de Javascript asincrónico y XML. Se utiliza para comunicarse con el servidor sin actualizar la página web y, por lo tanto, aumentar la experiencia del usuario y un mejor rendimiento.*

### AJAX - The XMLHttpRequest

*Para realizar una solicitud HTTP al servidor con JavaScript, necesita una instancia de un objeto con la funcionalidad necesaria. Su predecesor se originó en Internet Explorer como un objeto ActiveX llamado XMLHttpRequest*

**EJEMPLO:**

```
if (window.XMLHttpRequest) {  
    // code for modern browsers  
    xmlhttp = new XMLHttpRequest();  
} else {  
    // code for old IE browsers  
    xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");  
}
```

**Figura 5.** [https://www.w3schools.com/xml/ajax\\_xmlhttprequest\\_create.asp](https://www.w3schools.com/xml/ajax_xmlhttprequest_create.asp)

### AJAX – Request

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.

```
// Mozilla, Safari, IE7+ ...  
xhr = new XMLHttpRequest()
```

**Figura 6.** <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>

### AJAX – Response

La propiedad de respuesta *XMLHttpRequest* devuelve el contenido del cuerpo de la respuesta como un *ArrayBuffer*, *Blob*, *Document*, *JavaScript Object* o *DOMString*, según el valor de la propiedad *responseType* de la solicitud.

```
xhr.onreadystatechange = () => {  
    if(xhr.readyState === XMLHttpRequest.DONE) {  
        return xhr.response  
    }  
}
```

**Figura 7.** <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Functions>

## AJAX XML File

*AJAX se puede utilizar para la comunicación interactiva con un archivo XML.*

*Se puede llamar a un documento XML desde api / endpoint, internet o desde un archivo local.*

## AJAX PHP

*La respuesta php se puede llamar desde api / endpoint, internet o desde el archivo php del servidor local*

*EJEMPLO:*

```
<?php

$cd1 = array('title' => 'Empire Burlesque',
             'artist'=> 'Bob Dylan',
             'country'=> 'USA',
             'company'=> 'Columbia',
             'price'=> 10.90,
             'year'=> 1985);

$cd2 = array('title' => 'Hide your heart',
             'artist'=> 'Bonnie Tyler',
             'country'=> 'UK',
             'company'=> 'CBS Records',
             'price'=> 9.90,
             'year'=> 1988);

$cd3 = array('title' => 'Greatest Hits',
             'artist'=> 'Dolly Parton',
             'country'=> 'USA',
             'company'=> 'RCA',
             'price'=> 9.90,
             'year'=> 1982);
```

```
$catalog = array($cd1, $cd2, $cd3);  
  
echo json_encode($catalog);  
  
?>
```

**Figura 8 .**<https://github.com/segmentio/xml-parser/blob/master/examples/catalog.xml>

## AJAX ASP

```
function showHint(str) {  
    if (str.length == 0) {  
        document.getElementById("txtHint").innerHTML = "";  
        return;  
    } else {  
        var xmlhttp = new XMLHttpRequest();  
        xmlhttp.onreadystatechange = function() {  
            if (this.readyState == 4 && this.status == 200) {  
                document.getElementById("txtHint").innerHTML =  
this.responseText;  
            }  
        };  
        xmlhttp.open("GET", "gethint.asp?q=" + str, true);  
        xmlhttp.send();  
    }  
}
```

**Figura 9 .**<https://stackoverflow.com/questions/23297535/how-to-display-ajax-in-html>

## AJAX Database

*AJAX se puede utilizar para la comunicación interactiva con una base de datos. Es lo mismo que en los ejemplos anteriores, la única diferencia es que en el lenguaje que se*

*utiliza en el backend obtiene información de la base de datos, que se enviará al cliente (js), en el formato que se requiere para ser enviado.*

## Ajax Applications

*Las dos características principales de AJAX le permiten hacer lo siguiente:*

- *Hacer solicitudes al servidor sin volver a cargar la página*
- *Reciba y trabaje con datos del servidor*

# JS JSON

## JSON Intro

*JSON es una sintaxis para serializar objetos, matrices, números, cadenas, booleanos y nulos. Se basa en la sintaxis de JavaScript, pero es distinta: algunos JavaScript no son JSON.*

## JSON Syntax

*Los datos están en pares de nombre / valor*

*\* Los datos están separados por comas*

*\* Las llaves contienen objetos*

*\* Los corchetes contienen matrices*

## JSON vs XML

### SEMEJANZAS

- *Tanto JSON como XML se pueden usar para recibir datos de un servidor web*
- *Tanto JSON como XML son "autodescriptivos" (legibles por humanos)*
- *Tanto JSON como XML son jerárquicos (valores dentro de valores)*
- *Tanto JSON como XML pueden ser analizados y utilizados por muchos lenguajes de programación.*
- *Tanto JSON como XML se pueden recuperar con un XMLHttpRequest*

### DIFERENCIAS

*La mayor diferencia es: XML debe analizarse con un analizador XML. JSON puede ser analizado por una función estándar de JavaScript*

## JSON Data Types

*JSON es una sintaxis para serializar objetos, matrices, números, cadenas, booleanos y nulos. Se basa en la sintaxis de JavaScript, pero es distinta: algunos JavaScript no son JSON.*

## JSON Parse

*Analiza el texto de la cadena como JSON, opcionalmente transforma el valor producido y sus propiedades, y devuelve el valor. Cualquier violación de la sintaxis de JSON, incluidas las relacionadas con las diferencias entre JavaScript y JSON, hace que se produzca un `SyntaxError`. La opción `reviver` permite interpretar lo que el sustituto ha utilizado para reemplazar otros tipos de datos.*

```
const json = '{"result":true, "count":42}';  
  
const obj = JSON.parse(json);  
  
  
console.log(obj.count);  
  
    /// expected output: 42  
  
  
console.log(obj.result);  
  
    // expected output: true
```

**Figura 10.** [https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos\\_globales/JSON/parse](https://developer.mozilla.org/es/docs/Web/JavaScript/Referencia/Objetos_globales/JSON/parse)

## JSON Stringify

*El método `JSON.stringify()` convierte un objeto o valor de JavaScript en una cadena JSON, opcionalmente reemplaza los valores si se especifica una función de reemplazo u opcionalmente incluye solo las propiedades especificadas si se especifica una matriz de reemplazo.*

EJEMPLO:

```
console.log(JSON.stringify({ x: 5, y: 6 }));  
  
    // expected output: '{"x":5,"y":6}'
```

**Figura 11.**

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/JSON/stringify](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify)



## JSON Objects

Los nombres de propiedad deben ser cadenas entre comillas dobles; las comas finales están prohibidas.

EJEMPLO:

```
{"result":true, "count":42}
```

Figura 12.

[https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global\\_Objects/JSON/stringify](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON/stringify)

## JSON Arrays

Los nombres de propiedad deben ser cadenas entre comillas dobles; las comas finales están prohibidas.

```
{
  'libros':[
    {'titulo':'nombre1', 'autor':'autor1', 'ano:2005},
    {'titulo':'nonmre2', 'autor':'autor2', 'ano':1970}
  ]
}
```

Figura 13. Mi autoria

## JSON PHP

Un uso común de JSON es leer datos de un servidor web y mostrar los datos en una página web. PHP tiene algunas funciones integradas para manejar JSON. Los objetos en PHP se pueden convertir a JSON utilizando la función PHP `json_encode ()`:

```
<?php
$myObj->name = "John";
$myObj->age = 30;
$myObj->city = "New York";
$myJSON = json_encode($myObj);
echo $myJSON;
?>
```

Figura 14. [https://www.w3schools.com/js/js\\_json\\_php.asp](https://www.w3schools.com/js/js_json_php.asp)

## JSON HTML

*JSON se puede traducir fácilmente a JavaScript. JavaScript se puede utilizar para crear HTML en sus páginas web.*

## JSONP

*JSONP es un método para enviar datos JSON sin preocuparse por problemas entre dominios. JSONP no usa el objeto XMLHttpRequest..*

- *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.*

```
/// En html

<script src="demo_jsonp.php">

/// En php

<?php

$myJSON = '{ "name":"John", "age":30, "city":"New York" }';

echo "myFunc( ".$myJSON. " );";

?>

/// En javascript

function myFunc(myObj) {

    document.getElementById("demo").innerHTML = myObj.name;

}
```

Figura 15 .[https://www.w3schools.com/js/js\\_json\\_jsonp.asp](https://www.w3schools.com/js/js_json_jsonp.asp)

# JS vs jQuery

## jQuery

*jQuery es una biblioteca de JavaScript rápida, pequeña y rica en funciones. Hace que cosas como el desplazamiento y la manipulación de documentos HTML, el manejo de eventos, la animación y Ajax sean mucho más simples con una API fácil de usar que funciona en una multitud de navegadores.*

## jQuery Selectors

Los selectores jQuery le permiten seleccionar y manipular elementos HTML. Los selectores jQuery se utilizan para "encontrar" (o seleccionar) elementos HTML en función de su nombre, id, clases, tipos, atributos, valores de atributos y mucho más. Se basa en los selectores CSS existentes y, además, tiene algunos selectores personalizados propios. Todos los selectores en jQuery comienzan con el signo de dólar y paréntesis: \$ ().

EJEMPLO:

```
// Basics Selectors

$('div') //element

$('.class') //by class

$('#id') //by id
```

Figura 16 .<https://api.jquery.com/category/selectors>

## jQuery HTML

- *.addClass ()* Agrega las clases especificadas a cada elemento en el conjunto de elementos coincidentes.
- *.html ()* Obtenga el contenido HTML del primer elemento del conjunto de elementos coincidentes o establezca el contenido HTML de cada elemento coincidente.

## jQuery CSS

- *.height ()* Obtenga la altura calculada actual para el primer elemento del conjunto de elementos coincidentes o establezca la altura de cada elemento coincidente.
- *.css ()* Obtiene el valor de una propiedad de estilo calculada para el primer elemento del conjunto de elementos coincidentes o establece una o más propiedades CSS para cada elemento coincidente.

## jQuery DOM

- *.clone ()* Crea una copia profunda del conjunto de elementos coincidentes.
- *.empty ()* Elimina todos los nodos secundarios del conjunto de elementos coincidentes del DOM.

- *.insertAfter () Inserta cada elemento en el conjunto de elementos coincidentes después del destino.*
- *.remove () Elimina el conjunto de elementos coincidentes del DOM.*
- *.wrap () Envuelve una estructura HTML alrededor de cada elemento en el conjunto de elementos coincidentes.*

## REFERENCIAS

- *MDN Web Docs. 2020. Javascript. [online] Available at: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>> [Accessed 29 April 2020].*
- *W3schools.com. 2020. Javascript Function Definitions. [online] Available at: <[https://www.w3schools.com/js/js\\_function\\_definition.asp](https://www.w3schools.com/js/js_function_definition.asp)> [Accessed 29 April 2020].*
- *W3schools.com. 2020. Javascript Function Parameters. [online] Available at: <[https://www.w3schools.com/js/js\\_function\\_parameters.asp](https://www.w3schools.com/js/js_function_parameters.asp)> [Accessed 29 April 2020].*
- *W3schools.com. 2020. Javascript Tutorial. [online] Available at: <<https://www.w3schools.com/js/default.asp>> [Accessed 29 April 2020].*