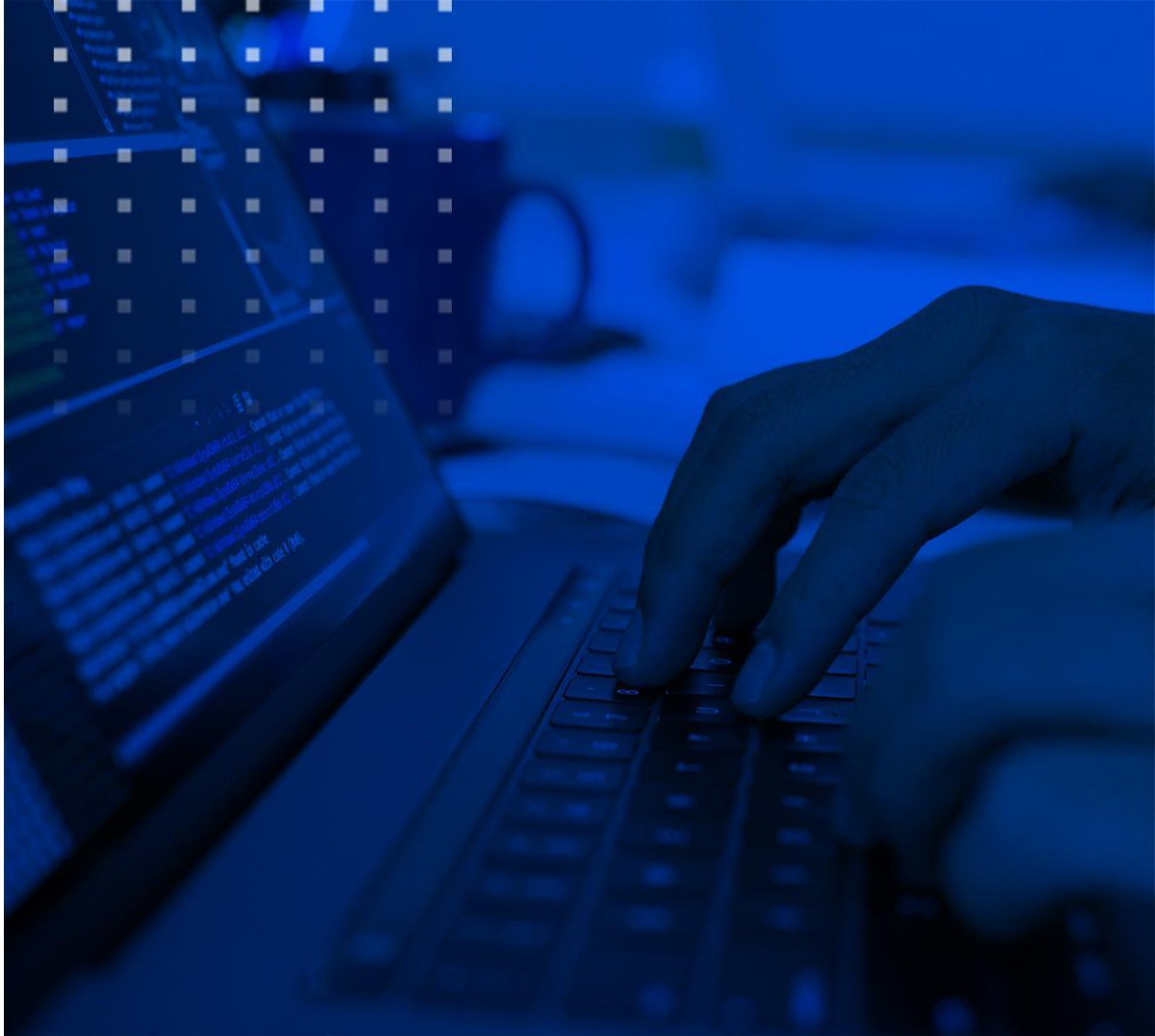
incluyeme.com

UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES

CURSO

FULL STACK DEVELOPER
NIVEL INICIAL



UNIDAD 10
JavaScript: Datos y repaso JS

Full Stack Developer Inicial

JavaScript: Datos y repaso JS

Datos

Hasta ahora trabajamos con variables/datos que creamos en nuestros archivos javaScript...

Pero en la práctica, generalmente los datos sobre los que tenemos que trabajar van a venir de otro lado...

JSON

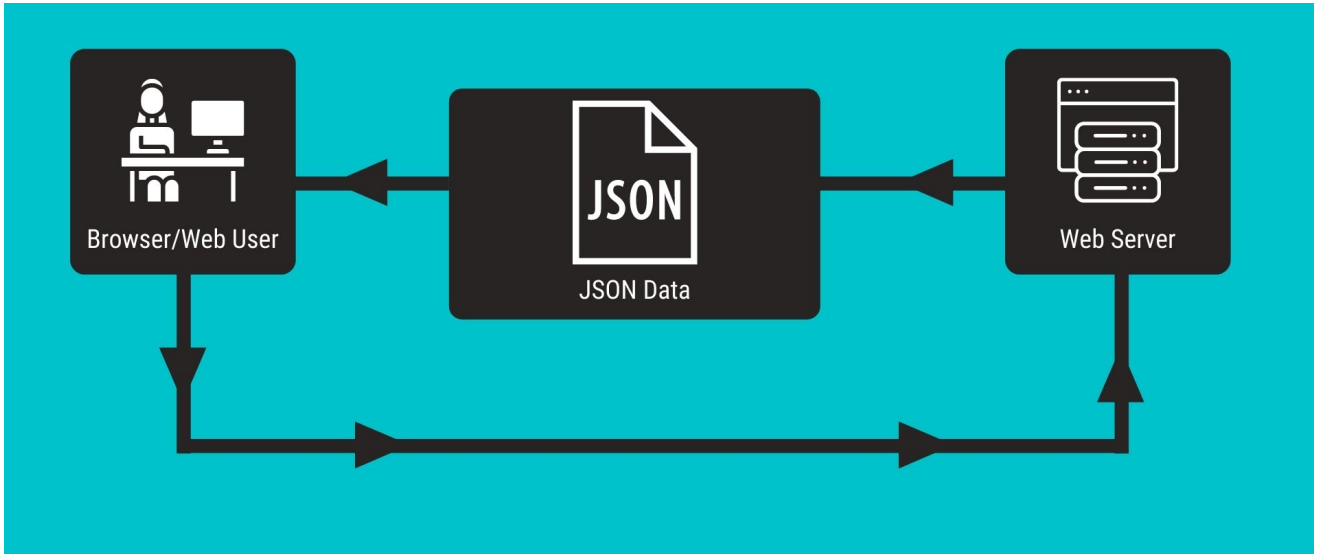
JSON (**JavaScript Object Notation**, «notación de objeto de JavaScript») es un **formato de texto** sencillo para el intercambio de **datos**.
Es muy **parecido a la sintaxis de objeto literal** de JavaScript, pero puede ser utilizado independientemente de JavaScript. Y muchos lenguajes de programación poseen la capacidad de leer y generar JSON.

JSON

Es comúnmente utilizado para **transmitir datos en aplicaciones web**.

Por ejemplo: enviar algunos datos desde el servidor al cliente, así estos datos pueden ser mostrados en páginas web, o vice versa.

JSON



JSON

Estructura

Como dijimos, un JSON es una cadena cuyo formato *recuerda* al de los objetos literales JavaScript.
Es posible incluir los mismos **tipos de datos básicos** dentro de un JSON que en un objeto estándar de JavaScript - cadenas, números, arrays, booleanos, y otros objetos literales.
Esto permite construir una jerarquía de datos.

```
"team": "Star Wars Characters",
"characters": [
{
     "name": "Leia Organa",
     "height": "150",
     "mass": "49",
     "hair_color": "brown",
     "skin_color": "light",
     "eye_color": "brown",
     "birth_year": "19BBY",
     "species": [],
          "starships": [],
 }

]
```

Accediendo a datos...

JSON

Si se carga este objeto en un programa JavaScript, convertido (parseado) en una **variable** llamada **personajesStarWars** por ejemplo, se podría acceder a los datos que contiene.

JSON

```
personajesStarwars.team
personajesStarwars['team’]
```

Acá **personajesStarWars** es nuestra variable que contiene el json y puedo ir accediendo a los distintos elementos.

JSON

```
personajesStarwars.team //Star Wars Characters
personajesStarwars.[team] //Star Wars Characters
```

JSON

```
personajesStarwars.characters[0].name //Leia Organa
```

Podemos acceder a todos los elementos del JSON siguiendo la notación de objetos y arrays que conocemos.

Fetch

Fetch

Proporciona una interfaz JavaScript para **acceder y manipular partes del canal HTTP**, tales como peticiones y respuestas.

Es una alternativa moderna (a XMLHttpRequest) que nos permite **interactuar con APIs** y **obtener los datos a nuestra aplicación**.

Fetch

```
fetch('https://hp-api.herokuapp.com/api/characters');
```

Utilizar el Fetch es muy sencillo. El **único parámetro requerido** de fetch() es una **url**. El método por defecto en este caso es GET.

Fetch

```
fetch('https://hp-api.herokuapp.com/api/characters', {
    method: 'GET/POST/PUT/DELETE'
});
```

El segundo parámetro es **opcional**: es un objeto que recibe el tipo de petición que vamos hacer, si queremos enviar datos, etc.

Fetch / Get

```
fetch('https://hp-api.herokuapp.com/api/characters')
      .then(response => response.json())
      .then(json => console.log(json))
```

El método **GET** se usa para recuperar datos del servidor. Este es un **método de solo lectura**, por lo que no tiene riesgo de mutar o corromper los datos.

Fetch / Get

```
fetch('https://hp-api.herokuapp.com/api/characters')
        .then(response => response.json())
        .then(json => console.log(json))
```

Cuando el valor solicitado con fetch está disponible, usamos el método **then()** para ejecutar una función con esos datos.

API Rest

¿Qué es una API?

Es una abreviatura de **Application Programming Interfaces** (interfaz de programación de aplicaciones). Se trata de un conjunto de **definiciones** y **protocolos** que se utiliza para desarrollar e integrar el software de las aplicaciones, permitiendo la **comunicación entre dos aplicaciones de software a través de un conjunto de reglas**.

¿Qué es una API?

En otras palabras, es una característica de **algunos servicios** que permite a los/as desarrolladores **interactuar** con ellos desde **aplicaciones externas**.

Son **mediadoras** entre los **usuarios** o clientes y los recursos o **servicios web** que quieren obtener.

API Rest

REST es un estilo de arquitectura que se usa para describir una interfaz que pueda ser utilizada **entre sistemas que usen HTTP** para la transferencia de información.

Es decir, describe una **forma de interactuar con un servidor HTTP**.

API Rest

La información, o representación, se entrega por medio de HTTP en uno de estos formatos: HTML, XLT, Python, PHP, texto sin formato o JSON.
JSON es uno de los más populares, ya que tanto las máquinas como las personas lo pueden comprender y no depende de ningún otro lenguaje.

Protocolo HTTP

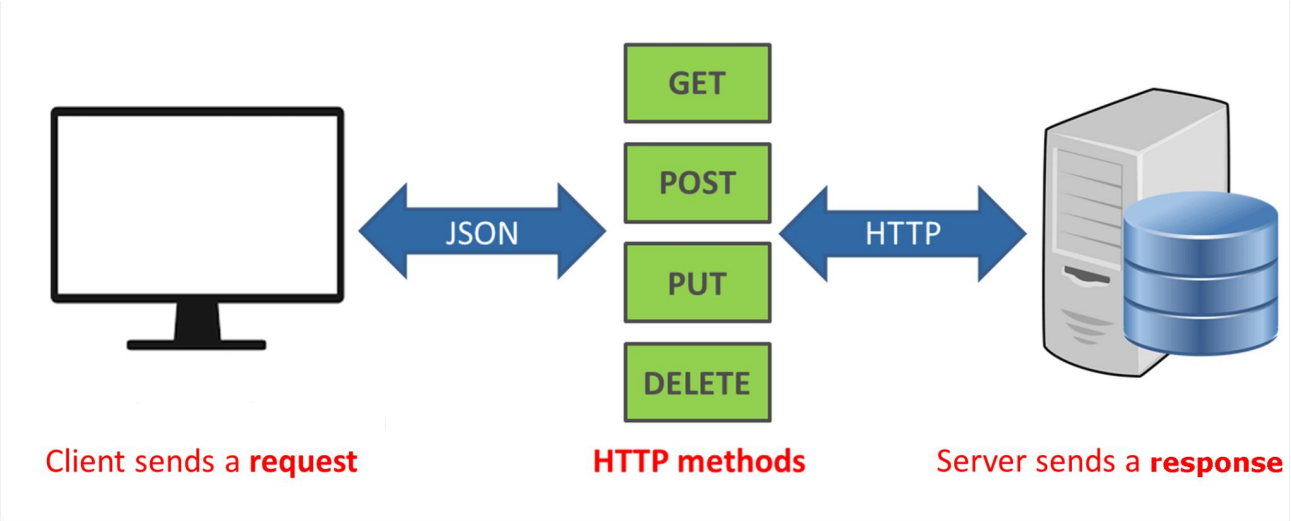
En la web, los clientes (como un navegador) se comunican con los distintos servidores web con ayuda del **protocolo HTTP**, el cual regula cómo formula sus peticiones el cliente y cómo ha de responder el servidor. El protocolo HTTP emplea **varios métodos** de petición diferentes.

Protocolo HTTP

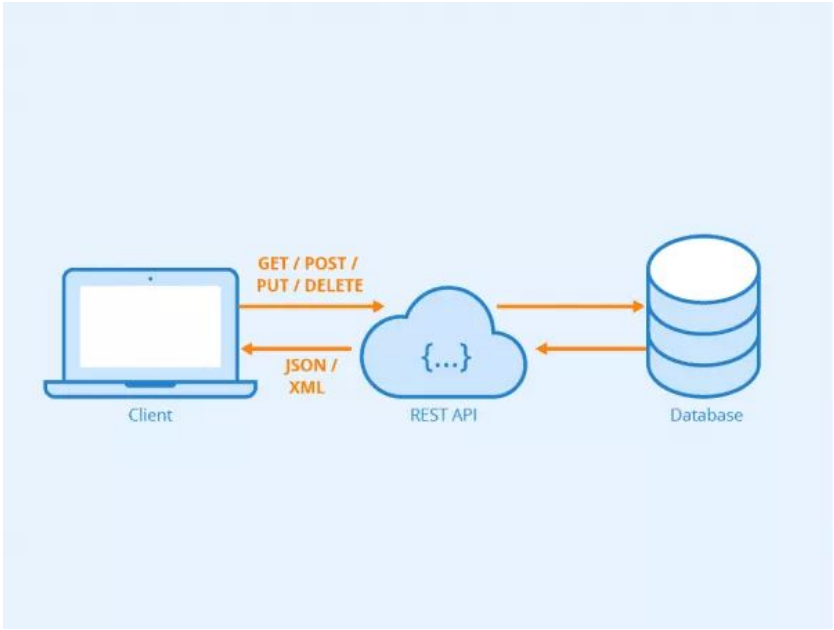
Los métodos de petición más importantes son:

- ★ **GET** es usado para recuperar un recurso.
- ★ **POST** se usa la mayoría de las veces para crear un nuevo recurso.
- ★ **PUT** es útil para crear o editar un recurso. En caso de existir, se reemplaza, de lo contrario se crea el nuevo recurso.
- ★ **DELETE** se usa para eliminar un recurso.

Protocolo HTTP



Ejemplos



- Harry Potter API
- Star Wars API
- JSON Placeholder
- Pokeapi
- Ghibli Api

Repaso JavaScript

Variables básicas

Una variable es un espacio en la memoria de la computadora que nos permite almacenar información para luego recuperarla y operarla.

Sus tipos básicos son:

- ★ Números
- ★ Strings
- ★ Booleanos

Arrays

Un array es un objeto especial que permite crear **grupos ordenados de datos** y nos provee herramientas para trabajar con ellos.

Los arrays pueden agrupar cualquier tipo de dato.

Tienen sus **propios método**s como: forEach, map o filter.

Arrays

Un array es un objeto especial que permite crear **grupos ordenados de datos** y nos provee herramientas para trabajar con ellos.

Los arrays pueden agrupar cualquier tipo de dato.

Tienen sus **propios método**s como: forEach, map o filter.

Objetos literales

Son objetos literales cuyas propiedades están **declaradas textualmente** en el código.
Los objetos en JavaScript nos ayudan agrupar información. Un objeto es un conjunto de propiedades en donde cada propiedad está compuesta de una llave y un valor.

Funciones

Una función un **conjunto de instrucciones** que realiza una tarea y se define con la palabra function y luego el nombre que le queramos poner.
Las funciones pueden **devolver un resultado**. La sentencia **return** finaliza la ejecución de la función y especifica un valor para ser devuelto.

DOM

Es la **estructura de objetos** que genera el **navegador** cuando se carga un documento y se puede alterar mediante Javascript para cambiar dinámicamente los contenidos y aspecto de la página.

Podríamos decir que es la forma en que JavaScript interpreta nuestro código HTML.

Condicionales

Las **instrucciones condicionales** se usan para realizar las diferentes acciones en el código según una condición que se evalúa como verdadera o falsa:

- ★ If / if else / if else if
- ★ Switch

Operadores

Los operadores nos ayudan a armar nuestras condiciones. Tenemos de dos tipos:

- **lógicos:** and, or, not
- **comparación:**mayor, menor, mayor igual, menor igual, igual, estrictamente igual. desigual, estrictamente desigual.

Bucles

Los bucles (loops) son utilizados para realizar **tareas repetitivas** con **base en una condición**. Las condiciones típicamente devuelven true o false al ser evaluados. El bucle continuará ejecutándose hasta que la condición devuelva false. Conocemos tres tipos de bucles:

- ★ While
- ★ Do...while
- ★ For

¡A practicar!

1. Crear una página que reciba datos de una API y los muestre, como en el ejemplo visto en clase. Hay que usar fetch para traer los datos. Ponerlos en un array de objetos una vez convertidos del formato JSON al formato de objetos nativos JavaScript y luego iterar ese arreglo para ir extrayendo los datos que nos interesan de cada elemento. En cada pasada, con esos datos generamos nuestra “card” de personajes y la agregamos al DOM.

2. Crear una hoja CSS y añadirle **estilo** a la lista de personajes.


[i]ncluyeme
.com


UTN.BA
UNIVERSIDAD TECNOLÓGICA NACIONAL
FACULTAD REGIONAL BUENOS AIRES