



Argentina  
programa  
4.0

**.UBA**fiuba   
FACULTAD DE INGENIERÍA

# **JavaScript: Clase 5 – JSON**

Bianchi – Zulaica

Febrero 2023

# Contenido

---

- Qué es JSON
- Para que se usa
- `JSON.stringify`
- `JSON.parse`
- `localStorage`

# Qué es JSON

JavaScript **Object Notation** es un formato basado en texto plano, para representar **datos estructurados** en la sintaxis de objetos de **JavaScript**

**EJEMPLO**

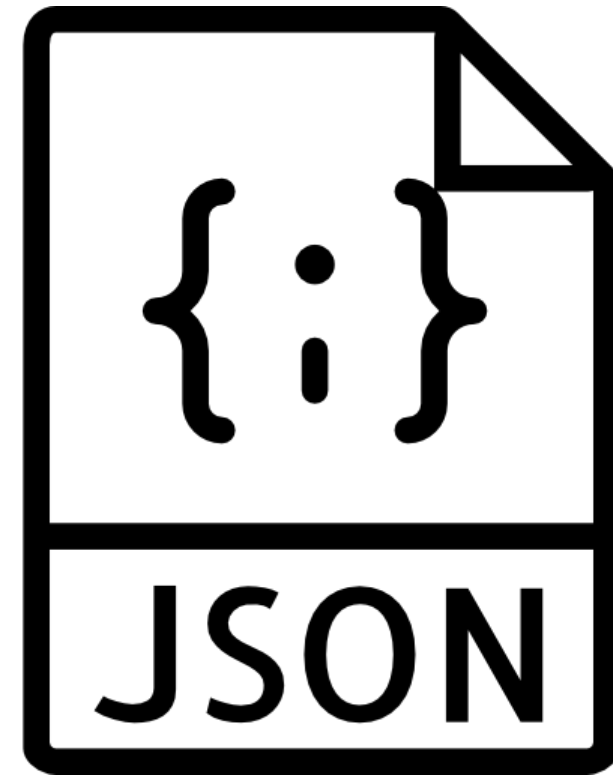
```
{ } elements.json •
C: > Users > NICOLAS > Desktop > text-to-periodic-symbols > data > { } elements.json
You, 1 minute ago | 1 author (You)
1  [
2  {
3    "number": 1,
4    "symbol": "H",
5    "name": "Hydrogen",
6    "atomic_mass": 1.008,
7    "category": "diatomic nonmetal"
8  },
9  {
10   "number": 2,
11   "symbol": "He",
12   "name": "Helium",
13   "atomic_mass": 4.0026022,
14   "category": "noble gas"
15 },
16 {
17   "number": 3,
18   "symbol": "Li",
19   "name": "Lithium",
20   "atomic_mass": 6.94,
21   "category": "alkali metal"
22 },
23 ]
```

- Usamos la extensión de archivo .json
- El elemento principal puede ser una lista u objeto. En el ejemplo tenemos una lista de objetos.
- Las claves deben ir entre comillas.
- No debemos poner coma luego del último elemento.

# Para que se usa JSON

---

- Intercambiar datos
  - Entre backend y frontend
  - Entre servidores
- Almacenamiento de datos
  - Datos de aplicación
  - Configuraciones
- Serialización y Deserialización



# JSON.stringify

Convierte un objeto o valor de JavaScript en una cadena de texto JSON

```
<script>  
  let objeto = {id: 1, color: "#256", nombre: "box"};  
  let texto = JSON.stringify(objeto);  
  console.log(texto);  
</script>
```

Console

```
"{'id':1,'color':'#256','nombre':'box'}"
```

# JSON.parse

Convierte una cadena de texto JSON en un objeto o valor de JavaScript

```
<script>  
  let texto = '{ "id": 1, "color": "#256", "nombre": "box" }';  
  let objeto = JSON.parse(texto);  
  console.log(objeto);  
</script>
```

Console

```
▼ {  
  "id": 1,  
  "color": "#256",  
  "nombre": "boxs"  
}
```

# localStorage

Objeto global que **almacena** datos de manera **persistente**.  
Es decir, se mantendrá aunque cerremos ó recargamos la página

```
localStorage.getItem(nombre);
```

- Accedemos a un dato con **getItem**
- Nos devolverá texto **JSON**
- Obtenemos un objeto con **JSON.parse**

```
localStorage.setItem(nombre, item);
```

- Establecemos un dato con **setItem**
- Debemos pasar el ítem como texto **JSON**
- Obtenemos texto con **JSON.stringify**



# localStorage - Ejemplo

```
<script>
  let numero_fav = JSON.parse(localStorage.getItem("numero_fav"));

  function pedir_numero_fav() {
    numero_fav = prompt("Ingrese su número favorito");
    localStorage.setItem("numero_fav", JSON.stringify(numero_fav));
  }

  if (!numero_fav) {
    pedir_numero_fav();
  } else {
    let correcto = confirm(`Su número favorito es ${numero_fav}`);
    if (!correcto) pedir_numero_fav();
  }

  alert("Recarga la página");
</script>
```

# Resumen

---

- ✓ Usamos JSON para representar **datos estructurados** en la sintaxis de objetos de **JavaScript**
- ✓ **JSON.stringify** convierte un objeto o valor de JavaScript en una cadena de texto JSON
- ✓ **JSON.parse** convierte una cadena de texto JSON en un objeto o valor de JavaScript
- ✓ **localStorage** almacena datos de manera **persistente**

# Ejercicios

---

- 1) La siguiente aplicacion: <https://codepen.io/nicolasezequielzulaicarivera/pen/OJowPYY> nos permite agregar tareas a una lista de tareas; pero estas se pierden al recargar la página. Modificar su código para guardar las tareas usando localStorage
- 2) Hacer una página que muestre el “nombre del usuario” y un botón para cambiarlo. Al clicar el botón se le pedirá el nombre por prompt, este es el nombre que se mostrará. El nombre debe persistir incluso si se recarga la página.

**[www.ingenieria.uba.ar](http://www.ingenieria.uba.ar)**

**f**    /ingenieriauba

 /FIUBAoficial