

Explicación y Ejemplos de JSON (JavaScript Object Notation)

JSON (JavaScript Object Notation) es un formato de texto ligero utilizado para el intercambio de datos. Aunque nació como una derivación de la sintaxis de objetos de JavaScript, actualmente es completamente independiente del lenguaje y es compatible con prácticamente cualquier lenguaje de programación moderno (Python, Java, PHP, C#, entre otros). Su popularidad radica en que es fácil de leer y escribir para los humanos, y al mismo tiempo, fácil de interpretar y generar para las computadoras.

Usos Principales de JSON

JSON se ha convertido en el estándar de facto en el desarrollo de software y la ciencia de datos gracias a su versatilidad. Sus usos principales incluyen:

- **Comunicación con APIs (Servicios Web):** Es el formato estándar para enviar y recibir datos entre un servidor y una aplicación web o móvil (arquitecturas REST y GraphQL).
- **Archivos de Configuración:** Muchos entornos de desarrollo y aplicaciones (como VS Code o Node.js con su package.json) utilizan JSON para almacenar la configuración del sistema.
- **Almacenamiento en Bases de Datos NoSQL:** Bases de datos orientadas a documentos (como MongoDB) almacenan la información en formatos directamente derivados de JSON (BSON).
- **Intercambio de Datos Estructurados:** Es el sustituto moderno del formato XML debido a su menor peso y mayor velocidad de procesamiento.

Estructura y Reglas Sintácticas de JSON

Un archivo o cadena JSON se construye principalmente sobre dos estructuras:

1. **Colección de pares de nombre/valor:** En la mayoría de los lenguajes, esto se traduce como un objeto, diccionario, registro o tabla hash.
2. **Lista ordenada de valores:** Lo que comúnmente se conoce como un arreglo (array), vector o lista.

Reglas de Oro en JSON:

- Los datos siempre se representan en pares de "clave": valor.
- Las claves (nombres) deben ir siempre entre comillas dobles (" ").
- Los datos están separados por comas (,).

- Las llaves { } envuelven a los objetos.
- Los corchetes [] envuelven a los arreglos.

Tipos de Datos Soportados

Dentro de un archivo JSON, los valores pueden ser únicamente los siguientes tipos de datos:

Tipo de Datos	Descripción / Ejemplo
Cadena (String)	Texto envuelto en comillas dobles. Ej: "Hola Mundo"
Número (Number)	Enteros o de punto flotante. Ej: 25 o 1050.75
Objeto (Object)	Una estructura anidada de llaves {}.
Arreglo (Array)	Una lista de valores indexados [].
Booleano (Boolean)	Valores lógicos: true o false.
Nulo (Null)	Representa un valor vacío o inexistente: null.

Ejemplos de JSON

1. Ejemplo Básico: Un Objeto Simple

Este ejemplo representa los datos básicos de un usuario:

```
{
  "id": 101,
  "nombre": "Carlos Mendoza",
  "edad": 29,
  "es_administrador": false,
  "correo": null
}
```

2. Ejemplo Avanzado: Estructura Completa y Anidada

En este escenario del mundo real, simulamos los datos de una empresa que cuenta con un objeto anidado para la dirección y un arreglo de objetos para listar a sus empleados:

```
{
  "empresa": "Tech Solutions",
  "año_fundacion": 2018,
  "activa": true,
  "direccion": {
    "calle": "Avenida Reforma 405",
    "ciudad": "Puebla",
    "pais": "México"
  },
  "departamentos": [
    "Sistemas",
    "Análisis de Datos",
    "Recursos Humanos"
  ],
  "empleados": [
    {
      "id": 1,
      "nombre": "Ana Gómez",
      "puesto": "Científica de Datos",
      "salario": 65000
    },
    {
      "id": 2,
      "nombre": "Luis Pineda",
      "puesto": "Desarrollador Backend",
      "salario": 58000
    }
  ]
}
```