

JSON

# JSON - Definición

- JavaScript Object Notation → Notación de objetos de Javascript
- Según la definición de la web de JSON ([json.org](https://json.org)) es un formato ligero de intercambio de datos.
- Podemos decir que es una sintaxis para serializar objetos.
- Serializar: Cuando representamos datos (p.e. de variables que se encuentran en memoria) como una secuencia de bytes listas para su almacenamiento o transmisión.
- JSON es independiente del lenguaje de programación, de hecho es un formato soportado por todos los lenguajes de programación actuales.

# JSON – Tipos de datos

- Los tipos de datos que se pueden utilizar en JSON son:
  - Números enteros y reales: 7, -4, 22.5
  - Cadenas de texto: “una cadena de texto”
  - Booleanos: true y false
  - Valores nulos: null
  - Arrays: representados como una lista de valores entre corchetes.
    - [ 1, 2, “abc”, [ 3, 4], true ]
  - Objetos: Colecciones de pares del tipo <nombre>:<valor>, separados por comas y puestas entre llaves.
    - El <nombre> debe aparecer entre comillas
    - El <valor> puede ser de cualquiera de los tipos de datos admitidos por JSON.
    - { “n1” : 6 , “n2”: true, “n3”: [ 6 ,7, 8], “n4”: { “a”: “serranito” } , “n5” : null }

# JSON – Uso en JS

- En Javascript podemos tener variables con cadenas de texto en formato JSON.

```
let cadenaJSON = '{ "n1" : "prueba" , "n2" : 7 }';
```

- Podemos convertir una cadena JSON en un objeto:

```
// De cadena en formato JSON a objeto  
let objetoNuevo = JSON.parse(cadenaJSON);  
console.log(objetoNuevo.n1); // "prueba"
```

# JSON – Uso en JS II

- En Javascript es común la definición de objetos literales, que no debemos confundir con una cadena en formato JSON:

```
let objetoLiteral = { x: 9 , y: 7 };
```

- Podemos transformar las variables a formato JSON:

```
// De objeto a cadena en formato JSON
```

```
let nuevaCadenaJSON = JSON.stringify(objetoLiteral);  
console.log(nuevaCadenaJSON); // '{ "x" : 9, "y" : 7 }'
```

# JSON – Envío y recepción de datos

- La utilización de JSON se ha extendido mucho debido a que es soportado por muchos lenguajes de programación, que de esta forma pueden intercambiar datos de una forma sencilla.
- Por ejemplo, es común en programación web enviar y recibir datos en formato JSON, ejecutándose en el lado del cliente (navegador web) lenguaje JS y en el lado del servidor otro lenguaje distinto (PHP, Java, C#,...)

