



JSON

- JavaScript Object Notation -

¿Qué es JSON?

“

Formato de representación de datos

”

¿Para qué sirve?. Características

- **Usado para APIs y ficheros de configuración**
- **Ligero y fácil de leer/escribir**
- **Intercambiable entre diferentes sistemas**
- **Se integra fácilmente en diferentes lenguajes de programación**

Tipos en JSON

Cadenas	“Gallinas entrantes, gallinas salientes”
Números	10 1.5 -10 1.2e10
Booleanos	true false
null	null
Arrays	[1, 2, 3] [“Periquito”, “Tortuga”]
Objetos	{“clave” : “valor”} {“edad”:30} {12}

ejemplo.json

```
{  
  "nombre": "Chema",  
  "numeroFavorito": 12,  
  "EsProgramador": true,  
  "hobbies": [ "Criar caracoles", "Sembrar nísperos", "Jugar al CoD" ],  
  "amigos": [ {  
    "nombre": "Hassan II",  
    "numeroFavorito": 7,  
    "EsProgramador": false,  
    "hobbies": [ "Enfoscarse mezquitas", "Ungir pies de pecadores", "Fórmula 1" ]  
  } ]  
}
```



JSON vs. XML

XML

Ventajas:

- Tiene un formato muy estructurado y fácil de comprender.
- Puede ser validado fácilmente mediante Schemas (XSD).
- Se pueden definir estructuras complejas y re utilizables.

Desventajas:

- Es más lioso de leer por un humano.
- El formato es sumamente estricto.
- Lleva más tiempo procesarlo.
- Un error con los *namespace* puede hacer que todo el documento sea inválido.

JSON vs. XML

JSON

Ventajas:

- Formato sumamente simple.
- Velocidad de procesamiento alta.
- Archivos de menor tamaño.

Desventajas:

- No puede verse con facilidad la estructura que representa.
- La estructura no se puede validar (al menos no en la versión sin Schema)

JSON vs. XML

```
<empleados>
  <empleado>
    <Nombre>Juan</Nombre>
```

```
<Apellidos>García</Apellidos>
```

```
</empleado>
```

```
<empleado>
```

```
<Nombre>Marisa</Nombre>
```

```
<Apellidos>Pérez</Apellidos>
```

```
</empleado>
```

```
<empleado>
```

```
p<Nombre>Carmen</Nombre>
```

```
{ "empleados": [
  { "Nombre": "Juan", "Apellidos": "García" },
  { "Nombre": "Marisa", "Apellidos": "Pérez" },
  { "Nombre": "Carmen", "Apellidos": "Martín" }
]}
```

↑
JSON

←
XML



Pero...

No hay librerías nativas en Java que procesen (en el argot se dice “parsear”, del inglés “parse”) el fichero o la cadena en formato Json.

Así que tenemos que utilizar una solución de terceros. Algunos ejemplos son:

- Jackson (<https://github.com/FasterXML/jackson>)
- Moshi (<https://github.com/square/moshi>)

Nosotros usaremos una de Google:

- Gson (<https://github.com/google/gson>)

Fin



prof.jduran@iesalixar.org