

Academia

Casos especiales

Anonymous · 08/10/2026

Tabla de contenidos

Casos especiales	3
<i>Cadenas de texto</i>	4
<i>Diccionarios</i>	5
<i>Listas</i>	6
<i>Formato de salida</i>	7

Academia » Leer datos de JSON con Selectores » Casos especiales

Casos especiales

Anonymous · 08/10/2026

Casos especiales

Última actualización el 08/10/2026

Academia » Leer datos de JSON con Selectores » Casos especiales » Cadenas de texto

Cadenas de texto

Cadenas de texto

Anonymous · 08/10/2026

Cadenas de texto

Es posible definir cadenas literales. De esta forma se puede usar un texto (string) para ser concatenado (unido) a otros selectores.

- **Formato:** "cadena literal"

Se mostrará la cadena tal cual es ingresada.

Ejemplos

Concatenación:

- Dado el JSON: {"nombre": "Empresa", "id": "123"}
- Selector: "ID de "(nombre)": "(id)"
- Resultado: "ID de Empresa: 123"

Última actualización el 08/10/2026

Academia » Leer datos de JSON con Selectores » Casos especiales » Diccionarios

Diccionarios

Diccionarios

Anonymous · 08/10/2026

Diccionarios

Los selectores pueden acceder a elementos específicos en diccionarios.

Ejemplo en Diccionario Anidado: `diccionario.clave.subclave` accede a subclave dentro de un diccionario anidado en clave.

Última actualización el 08/10/2026

Academia » Leer datos de JSON con Selectores » Casos especiales » Listas

Listas

Listas

Anonymous · 08/10/2026

Listas

Los selectores pueden acceder a elementos específicos en listas o arreglos

Ejemplo en Lista: lista[2] accede al tercer elemento de una lista.

Ejemplos

Selector de Arreglos:

- **Dado el JSON:** {"array": [1, 2, 3]}
- **Selector:** array[1]
- **Resultado:** 2

Selector de Arreglo Anidado:

- **Dado el JSON:** {"nested_array": {"array": [1, 2, 3]}}
- **Selector:** nested_array.array[2]
- **Resultado:** 3

Selector de Arreglo de Diccionarios:

- **Dado el JSON:** {"mixed": [{"key": 10, "value": "hola"}, {"key": 20, "value": "mundo"}]}
- **Selector:** mixed[key=20:value]
- **Resultado:** "mundo"

Nota

Este último caso es interesante porque permite obtener, de un objeto que está en una lista de objetos, el valor de un índice a partir del valor de otro índice del mismo objeto.

Última actualización el 08/10/2026

Academia » Leer datos de JSON con Selectores » Casos especiales » Formato de salida

Formato de salida

Formato de salida

Anonymous · 08/10/2026

Formato de salida

Los selectores adecuarán el formato de la salida según el tipo de datos de los elementos que participen en la selección del selector.

Respetar tipo de datos original

Si existe un solo selector como resultado, ya sea porque:

- Se pidió solo un selector
- O, el selector está formado por varios selectores unidos mediante OR y se obtiene el primer valor no vacío.

El tipo de dato del resultado del selector será el tipo de dato del elemento seleccionado del JSON.

Tipo de datos como cadena de texto (string)

Si el selector está concatenando resultados de varios selectores o cadenas literales, y el resultado no es vacío, el tipo de datos del resultado será siempre una cadena de texto o string.

Tipo de datos null

Si el elemento seleccionado mediante el selector tiene los valores en el JSON:

- null.
- "" (cadena de texto vacía).

El resultado del selector será null.

Selector no encontrado en el JSON

Si se escribe un selector de manera errónea, o que en el JSON que estamos usando para extraer el dato no existe lo buscado, el resultado será **null**, un caso no existente.

Por ejemplo, si tenemos el siguiente JSON:

```
{  
  "array": [1, 2, 3]  
}
```

Y usamos el siguiente selector:

```
array2[2]
```

Obtendremos **null** como resultado, porque se buscó el índice **array2** y este índice no existe en el JSON.

Última actualización el 08/10/2026