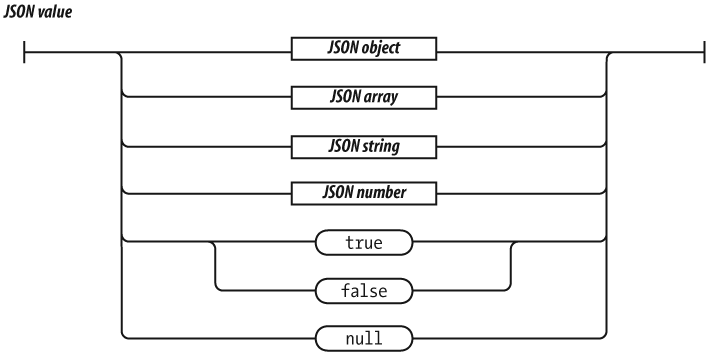
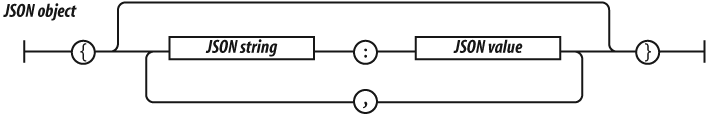
Appendix E. JSON

# Sintaxis JSON

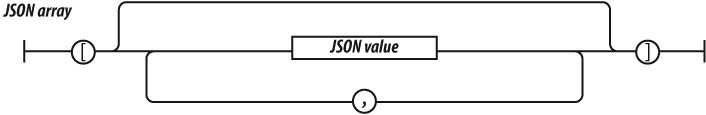
JSON tiene seis tipos de valores: objetos, matrices, cadenas, números, booleanos (true y false), y el valor especial null. Los espacios en blanco (espacios, tabuladores, retornos de carro y caracteres de nueva línea) pueden insertarse antes o después de cualquier valor. Esto puede hacer que los textos JSON sean más fáciles de leer para los humanos. Los espacios en blanco pueden omitirse para reducir los costes de transmisión o almacenamiento.



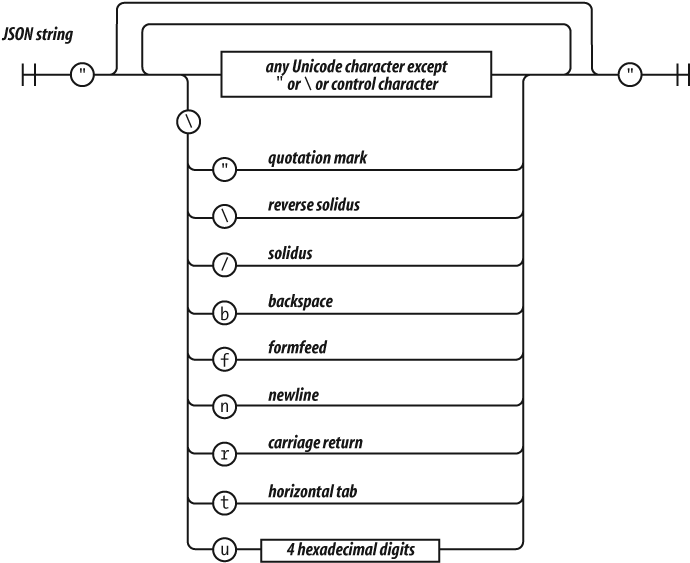
Un objeto JSON es un contenedor desordenado de pares nombre/valor. Un nombre puede ser cualquier cadena. Un valor puede ser cualquier valor JSON, incluidas matrices y objetos. Los objetos JSON pueden anidarse a cualquier profundidad, pero generalmente es más eficaz mantenerlos relativamente planos. La mayoría de los lenguajes tienen una característica que mapea fácilmente a los objetos JSON, como un objeto, struct, registro, diccionario, tabla hash, lista de propiedades o matriz asociativa.



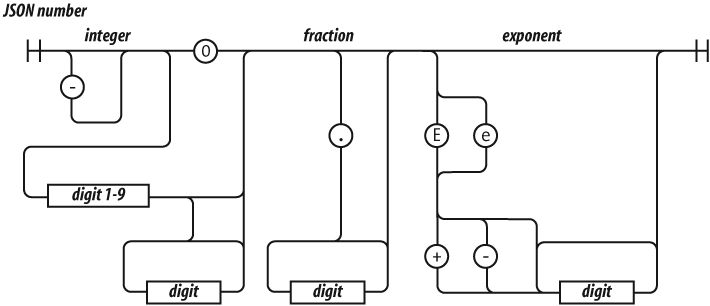
La matriz JSON es una secuencia ordenada de valores. Un valor puede ser cualquier valor JSON, incluidas las matrices y los objetos. La mayoría de los lenguajes tienen una característica que se mapea fácilmente en las matrices JSON, como una matriz, un vector, una lista o una secuencia.



Una cadena JSON va entre comillas dobles. El carácter \ se utiliza para el escape. JSON permite escapar el carácter / para que JSON pueda incrustarse en etiquetas HTML . HTML no permite la secuencia </ excepto para iniciar la etiqueta </script>. JSON permite <\/, que produce el mismo resultado pero no confunde a HTML.



Los números JSON son como los números JavaScript. No se permite un cero a la izquierda en los números enteros porque algunos lenguajes lo utilizan para indicar el octal. Ese tipo de confusión radix no es deseable en un formato de intercambio de datos. Un número puede ser entero, real o científico.



Eso es todo. Eso es todo en JSON. Los objetivos de diseño de JSON eran ser mínimo, portátil, textual y un subconjunto de JavaScript. Cuanto menos tengamos que ponernos de acuerdo para interoperar, más fácilmente podremos interoperar.

[  
 {  
 "first": "Jerome",  
 "middle": "Lester",  
 "last": "Howard",  
 "nick-name": "Curly",  
 "born": 1903,  
 "died": 1952,  
 "quote": "nyuk-nyuk-nyuk!"  
 },  
 {  
 "first": "Harry",  
 "middle": "Moses",  
 "last": "Howard",  
 "nick-name": "Moe",  
 "born": 1897,  
 "died": 1975,  
 "quote": "Why, you!"  
 },  
 {  
 "first": "Louis",  
 "last": "Feinberg",  
 "nick-name": "Larry",  
 "born": 1902,  
 "died": 1975,  
 "quote": "I'm sorry. Moe, it was an accident!"  
 }  
]

[anterior](ape_1.html)[Subtema 2 de 2: (Ver todo)](ape.html)