

Mit JSON (**J**ava**S**cript **O**bject **N**otation) verfügt man über einen Syntax um Daten zu speichern, bzw. Daten auszutauschen. Diese Notation wandelt Objekte, Variablen und Arrays in einen reinen Text um, der dann z. B. mit `localStorage` gespeichert oder an ein PHP Skript übergeben werden kann. JSON ist ähnlich aufgebaut wie ein OOP Literal, nur werden hier die Keys immer unter doppelte Anführungszeichen gestellt.

```
{ "Name" : "Karl König", "Alter" : 33 }
```



Folgende Datentypen werden durch JSON unterstützt.

String	{ "wohntort": "Graz" }
Zahlen	{ "PLZ": 8010 }
JSON Objekt	{ "Person": { "name": "Peter", "alter": 25 } }
Arrays	{ "Besucher": ["Karl", "Eva", "Friedrich"] }
Boolean	{ "zugriff": true }
JSON null	{ "akademischerGrad": null }



Gegenwärtig ist es nicht möglich eine Funktion oder ein Datum in eine JSON Notation zu wandeln. Möglich ist es aber, das Datum oder die Funktion als String-Typ anzugeben.

JS



JSON.stringify(objekt) ;

Die Methode wandelt ein JavaScript Objekt (oder Array bzw. Variable) in die JSON Notation. Das Beispiel hat eine Konstrukturfunktion `mitglied()`. Ein `xxlGruppe` Objekt wird vereinbart – die Eigenschaften `.name` und `.pincode` wurden verändert. Im Anschluss wird das `xxlGruppe` mit `stringify` in einen JSON Text gewandelt.

```
function mitglied() {
    this.name = "",
    this.frei = [3, 6, 4, 22],
    this.pincode = 9999
}
var xxlGruppe = new mitglied;
    xxlGruppe.name = "Herbert Trinker";
    xxlGruppe.pincode = 3532;
var speichern = JSON.stringify(xxlGruppe);
console.log(speichern);
```

JS



JSON.parse(objekt) ;

In die andere Richtung geht es mit `JSON.parse()` ;
Dieses konvertiert den JSON Text in ein JavaScript Objekt.

```
var eineGruppe = JSON.parse(speichern);
console.log(eineGruppe);
```