JSON

JSON是JavaScript Object Notation的缩写，它是一种数据交换格式。

JSON数据类型

- number：和JavaScript的number完全一致；
- boolean：就是JavaScript的true或false；
- string：就是JavaScript的string；
- null：就是JavaScript的null；
- array：就是JavaScript的Array表示方式——[]；
- object：就是JavaScript的{ ... }表示方式。

以及上面的任意组合。

并且，JSON还定死了字符集必须是UTF-8，SON的字符串规定必须用双引号""，Object的键也必须用双引号""。

把任何JavaScript对象变成JSON，就是把这个对象序列化成一个JSON格式的字符串，这样才能够通过网络传递给其他计算机。

如果我们收到一个JSON格式的字符串，只需要把它反序列化成一个JavaScript对象，就可以在JavaScript中直接使用这个对象了。

序列化

`JSON.stringify(object);`

```
var xiaoming = {
    name: '小明',
    age: 14,
    gender: true,
    height: 1.65,
    grade: null,
    'middle-school': '\"W3C\" Middle School',
    skills: ['JavaScript', 'Java', 'Python', 'Lisp']
};
JSON.stringify(xiaoming); // '{"name":"小明","age":14,"gender":true,"height":1.65,"grade":null,"middle-school":"\"W3C\" Middle School","skills":["JavaScript","Java","Python","Lisp"]}'
```

`JSON.stringify(object, null, '  ');`

加上参数，按缩进输出

```
JSON.stringify(xiaoming, null, '  ');
```

结果：

```
{
  "name": "小明",
  "age": 14,
  "gender": true,
  "height": 1.65,
  "grade": null,
  "middle-school": "\"W3C\" Middle School",
  "skills": [
    "JavaScript",
```

```
    "Java",
    "Python",
    "Lisp"
  ]
}
```

JSON.stringify(object, ['name1', 'name2'], ' ');
第二个参数用于控制如何筛选对象的键值，如果我们只想输出指定的属性，可以传入Array

```
JSON.stringify(xiaoming, ['name', 'skills'], '  ');
```

结果：

```
{
  "name": "小明",
  "skills": [
    "JavaScript",
    "Java",
    "Python",
    "Lisp"
  ]
}
```

JSON.stringify(object, fuc, ' ');
还可以传入一个函数，这样对象的每个键值对都会被函数先处理：

```
function convert(key, value) {
    if (typeof value === 'string') {
        return value.toUpperCase();
    }
    return value;
}
JSON.stringify(xiaoming, convert, '  ');
```

上面的代码把所有属性值都变成大写：

```
{
  "name": "小明",
  "age": 14,
  "gender": true,
  "height": 1.65,
  "grade": null,
  "middle-school": "\"W3C\" MIDDLE SCHOOL",
  "skills": [
    "JAVASCRIPT",
    "JAVA",
    "PYTHON",
    "LISP"
  ]
}
```

如果我们还想要精确控制如何序列化小明，可以给xiaoming定义一个toJSON()的方法，直接返回JSON应该序列化的数据：

```
var xiaoming = {
    name: '小明',
    age: 14,
    gender: true,
    height: 1.65,
    grade: null,
    'middle-school': '\"W3C\" Middle School',
    skills: ['JavaScript', 'Java', 'Python', 'Lisp'],
```

```
    toJSON: function () {
        return { // 只输出name和age，并且改变了key：
            'Name': this.name,
            'Age': this.age
        };
    }
};
JSON.stringify(xiaoming); // '{"Name":"小明","Age":14}'
```

反序列化

JSON.parse()

把一个JSON格式的字符串变成一个JavaScript对象

```
JSON.parse('[1,2,3,true]'); // [1, 2, 3, true]
JSON.parse('{"name":"小明","age":14}'); // Object {name: '小明', age: 14}
JSON.parse('true'); // true
JSON.parse('123.45'); // 123.45
```

JSON.parse()还可以接收一个函数，用来转换解析出的属性：

```
JSON.parse('{"name":"小明","age":14}', function (key, value) {
    // 把number * 2:
    if (key === 'name') {
        return value + '同学';
    }
    return value;
}); // Object {name: '小明同学', age: 14}
```