

# JSON

## 一、概念

1. 概念：JavaScript 对象表示法
2. 作用：用来存储和交换文本信息
3. 优势：相较于 xml 轻便小巧

## 二、基本规则

### 1. 基本规则

#### 1. 基本规则

- \* 数据在名称/值对中：json数据是由键值对构成的
- \* 键用引号(单双都行)引起来，也可以不使用引号
- \* 值得取值类型：
  1. 数字（整数或浮点数）
  2. 字符串（在双引号中）
  3. 逻辑值（true 或 false）
  4. 数组（在方括号中） {"persons":[{"},{}]}
  5. 对象（在花括号中） {"address":{"province": "陕西"...}}
  6. null
- \* 数据由逗号分隔：多个键值对由逗号分隔
- \* 花括号保存对象：使用{}定义json 格式
- \* 方括号保存数组：[]

### 2. 示例

```
//1.定义基本格式
var person = {"name": "张三", age: 23, 'gender': true};

//alert(person);
//2.嵌套格式 {}——> []
var persons = {
  "persons": [
    {"name": "张三", "age": 23, "gender": true},
    {"name": "李四", "age": 24, "gender": true},
    {"name": "王五", "age": 25, "gender": false}
  ]
};
// alert(persons);
```

## 三、获取数据

### 1. 基本方式

1. json对象.键名
2. json对象["键名"]
3. 数组对象[索引]

### 2. 示例

```
// alert(persons);
//获取王五值
var name1 = persons.persons[2].name;
// alert(name1);

//2.嵌套格式 []——> {}
var ps = [{"name": "张三", "age": 23, "gender": true},
  {"name": "李四", "age": 24, "gender": true},
  {"name": "王五", "age": 25, "gender": false}];
//获取李四值
//alert(ps);
alert(ps[1].name);
```

## 四、JSON 解析器-Jackson

### (一) 基本介绍

1. 概念：Spring 内置的 JSON 解析器
2. 使用步骤
  - (1) 导入 jackson 相关 jar 包
  - (2) 创建 jackson 核心对象

## (3) 对象转 JSON-调用对象中的方法进行转换

```
//3.转换
/*
    转换方法:
    writeValue(参数1, obj):
        参数1:
            File: 将obj对象转换为JSON字符串,并保存到指定的文件中
            Writer: 将obj对象转换为JSON字符串,并将json数据填充到字符输出流中
            OutputStream: 将obj对象转换为JSON字符串,并将json数据填充到字节输出流中
            writeValueAsString(obj): 将对象转为json字符串
*/
String json = mapper.writeValueAsString(p);
//{"name":"张三","age":23,"gender":"男"}
System.out.println(json);
```

## (4) JSON 转对象-调用对象中的方法进行转换

```
//演示 JSON字符串转为Java对象
@Test
public void test5() throws Exception {
    //1.初始化JSON字符串
    String json = "{\"gender\":\"男\",\"name\":\"张三\",\"age\":23}";

    //2.创建ObjectMapper对象
    ObjectMapper mapper = new ObjectMapper();
    //3.转换为Java对象 Person对象
    Person person = mapper.readValue(json, Person.class);

    System.out.println(person);
}
```

## (二) 注解

1. @JsonIgnore: 排除属性

2. @JsonFormat: 对展示的数据进行格式化

```
@JsonFormat(pattern = "yyyy-MM-dd")
private Date birthday;
```