

1.JSON

1. 前端传递JSON数据给服务器端

1. 前端: 将数据封装成JS对象/数组

```
var params = {  
    uname : "wangwc",  
    uage : 30,  
    ugender : "male"  
}
```

2. 前端: 将JS对象转换成JSON格式的字符串

方法 : `JSON.stringify()`
作用 : 将JS对象转换成JSON串
示例 :

```
var jsonStr = JSON.stringify(params)
```

3. 服务器端: 接收响应数据

```
jsonStr = request.GET['xxxx']
```

4. 服务器端: 将JSON串转换为Python字典/列表

方法: `json.loads()`
作用: 将JSON串转换成Python字典/列表
示例:

```
dic = json.loads(jsonStr)  
通过 dic['key'] 取值
```

2.JQ对AJAX的支持

1. `$obj.load()`

作用: 载入远程的HTML文件到指定的元素中

语法:

```
$obj.load(url, data, callback)
```

`$obj`: 显示响应内容的jq元素

`url`: 请求地址

`data`: 请求参数(可省略)

方式1: 字符串传参

```
"key1=value1&key2=value2"
```

注: 此种传参会使用 `get` 方式发送请求

方式2: 使用JS对象传参

```
{  
    key1: "value1",  
    key2: "value2"  
}
```

注: 此种传参会使用 `post` 方式发送请求

`callback`: 响应成功后的回调函数(可省略)

2. `$.get()`

作用: 通过`get`方式异步的向远程地址发送请求

语法: `$.get(url, data, callback, type)`

`url`: 请求地址

`data`: 参数

`callback`: 响应成功后的回调函数

type: 响应回来的数据的格式

取值如下:

- 1.html : 响应回来的文本是html文本
- 2.text : 响应回来的文本是text文本
- 3.script : 响应回来的文本是js执行脚本
- 4.json : 响应回来的文本是json格式的文本

3.\$.post()

4.\$.ajax()

作用: 自定义所有的ajax操作

语法:

\$.ajax({AJAX的参数对象});

AJAX的参数对象:

- 1.url: 异步请求地址
- 2.data: 请求到服务器端的参数
 1. 字符串
 2. JS对象
- 3.type: 请求方式, 'get' 或 'post'
- 4.dataType: 服务器端响应回来的数据的格式
html, text, script, json
- 5.async: 是否采用异步的方式发送请求
true: 使用异步 (默认值)
false: 使用同步
- 6.success: 响应成功后的回调函数
function(data){
 data表示的是响应回来的数据
}
- 7.error: 请求或响应失败时的回调函数

3. 跨域(Cross Domain)

1. 什么是跨域

HTTP协议中的一个策略 - "同源策略"

同源: 多个地址中, 具备相同协议, 相同域名, 相同端口的地址称之为"同源地址"

在HTTP中, 只有同源地址才能发送ajax请求, 非同源地址是被拒绝的(<script> 和 除外)

http://www.tedu.cn/a.html

http://www.tedu.cn/server

以上地址是"同源地址"

http://www.tedu.cn/a.html

https://www.tedu.cn/server

以上地址是"非同源地址"

http://localhost:8000/a.html

http://127.0.0.1:8000/server

以上地址是"非同源"

