JSONP 是什么？

问题：JSONP 是什么？补充如下代码，实现一个JSONP方法。

```
function jsonp(setting){
    //补充代码
}
jsonp({
  url: 'http://photo.sina.cn/aj/index',
  key: 'jsoncallback',
  data: {
    page: 1,
    cate: 'recommend'
  },
  callback: function(ret){
    console.log(ret)
  }
})
```

背景：

小谷同学在学习 ajax 后想做一个简单的天气预报应用，但找不到合适的天气接口，便向小饥求助。应小谷的要求小饥写了一个获取当前访问者所在地天气的接口发布到线上，接口URL： api.jirengu.com/weather 。小谷把 URL复制到浏览器浏览器地址栏按下回车键，页面很神奇地展示了小谷所在城市的天气数据。于是小谷立即写了个页面，使用 ajax 调用当前接口，代码如下：

```
$.get('http://api.jirengu.com/weather.php')
 .then(function(ret){
    console.log(ret)
})
```

小谷打开控制台，满心期待想看到返回的天气数据，但映入眼帘的是几行红色的警告：

「难道小饥在耍我？这人啊真不可信」，小谷愤愤的把报错截图甩给了小饥，正要开口责问，小饥说：「兄弟，这是跨域的问题，你用 JSONP 的方式调用吧，接口我都给你做了支持，直接使用 callback回调」 说完便埋入自己的代码里。

「啥 JSONP？ callback？ 跨域这个词貌似听过，我自己先查查别让小饥那小子鄙视我」小谷暗暗的想，「原来是跨域啊，你不早说，谢啦」，回到工位小谷赶紧打开谷歌。

同源策略(Same origin Policy)

浏览器出于安全方面的考虑，只允许与同域下的接口交互。

同域指的是？

- 同协议：如都是http或者https
- 同域名：如都是 jirengu.com/a 和 jirengu.com/b
- 同端口：如都是80端口

比如： 用户打开了 页面: jirengu.com/blog， 当前页面下的 js 向 jirengu.com/xxx 的接口发 ajax 请求，浏览器是允许的。但假如向: hunger-valley.com/xxx 发ajax请求则会被浏览器阻止掉，因为存在跨域调用。

「原来如此，怪不得浏览器会报错。跨域不过如此嘛！那 JSONP是什么呢？」

HTML 中 script 标签可以加载其他域下的js，比如我们经常引入一个其他域下线上cdn的jQuery。那如何利用这个特性实现从其他域下获取数据呢？

可以先这样试试：

```
<script src="http://api.jirengu.com/weather.php"></script>
```

这时候会向天气接口发送请求获取数据，获取数据后做为 js 来执行。
但这里有个问题， 数据是 JSON 格式的数据，直接作为 JS 运行的话我如何去得到这个数据来操作呢？

这样试试：

```
<script src="http://api.jirengu.com/weather.php?callback=showData"></scri
pt>
```

这个请求到达后端后，后端会去解析callback这个参数获取到字符串showData，在发送数据做如下处理：

```
之前后端返回数据： {"city": "hangzhou", "weather": "晴天"}
现在后端返回数据： showData({"city": "hangzhou", "weather": "晴天"})
```

前端script标签在加载数据后会把 「showData({"city": "hangzhou", "weather": "晴天"})」做为 js 来执行，这实际上就是调用showData这个函数，同时参数是{"city": "hangzhou", "weather": "晴天"}。
用户只需要在加载提前在页面定义好showData这个全局函数，在函数内部处理参数即可。

```
<script>
    function showData(ret){
        console.log(ret);
    }
```

```
</script>
<script src="http://api.jirengu.com/weather.php?callback=showData"></scri
pt>
```

「原来这就是 JSONP(JSON with padding)，总结一下：」
1. JSONP是通过 script 标签加载数据的方式去获取数据当做 JS 代码来执行

2. 提前在页面上声明一个函数，函数名通过接口传参的方式传给后台，后台解析到函数名后在原始数据上「包裹」这个函数名，发送给前端。换句话说，JSONP 需要对应接口的后端的配合才能实现。

「原理很简单，但用起来代码好丑陋，我做个封装让小饥看看」

```
function jsonp(setting){
  setting.data = setting.data || {}
  setting.key = setting.key||'callback'
  setting.callback = setting.callback||function(){}
  setting.data[setting.key] = '__onGetData__'

  window.__onGetData__ = function(data){
    setting.callback (data);
  }

  var script = document.createElement('script')
  var query = []
  for(var key in setting.data){
    query.push( key + '='+ encodeURIComponent(setting.data[key]) )
  }
  script.src = setting.url + '?' + query.join('&')
  document.head.appendChild(script)
  document.head.removeChild(script)

}

jsonp({
  url: 'http://api.jirengu.com/weather.php',
  callback: function(ret){
    console.log(ret)
  }
})
jsonp({
  url: 'http://photo.sina.cn/aj/index',
  key: 'jsoncallback',
  data: {
    page: 1,
    cate: 'recommend'
```

```
  },
  callback: function(ret){
    console.log(ret)
  }
})
```

查看代码：饥人谷JS Bin

小谷成就感爆棚，「小饥定会刮目相看的...」