

模板字符串：（``）

``里面放的是字符串，如果有变量，那么使用\${}

例：‘我叫’+str === `我叫\${str}`

\${}除了可以放变量还可以运算 \${num*10} -> 10 使用了模板字符串之后可以随意换行。

上传

浏览器中的js，不能操作本地文件（增删改.....）

上传：传统的上传

最大的优点就是我们可以异步上传一个二进制文件。

```
var f = new FormData();  
f.append(key,val);  
ajax.send(f);
```

```
btn.onclick = function(){  
    var ajax = new XMLHttpRequest;  
    ajax.open('post','post_file.php');  
    var formData = new FormData();  
    formData.append('file',f.files[0]);  
    ajax.send(formData);  
    ajax.onload = function(){  
        console.log(ajax.responseText);  
    }  
}
```

file空间中的数据是在<input type="file" id="f">的files[0]

ajax.upload.onprogress(进行监听数据上传进度的，只要后端收到一次数据就会调用这个函数，每次数据的细节信息都放在事件对象上)。

ev.loaded 当前传输进度

ev.total 总大小

```
btn.onclick = function(){
    var ajax = new XMLHttpRequest;
    ajax.open('post','post_file.php');
    var formData = new FormData();
    formData.append('file',f.files[0]);
    ajax.upload.onprogress = function(ev){
        let scale = Math.floor((ev.loaded / ev.total)*100);
        loading.style.cssText =
`transition:.5s;width:${scale}%;opacity:${(100-
scale)/100}`;
        if(scale == 100){
            loading.style.transition = 0;
            setTimeout(function(){
                loading.style.width = '0';
                setTimeout(function(){
                    loading.style.opacity = '1';
                    box.style.opacity = 0;
                    document.body.style.transition = '.5s';
                    document.body.style.background = 'yellow';
                });
            });
        }
    };
}
```

```

        },1000)
    }
}
ajax.send(formData);
}

```

跨域

同源策略： 一种约定，它是浏览器最核心也是最基本的安全功能，如果缺少了同源策略，则浏览器的正常功能可能会受到影响。可以说Web是构建在同源策略基础之上的，浏览器只针对同源策略的一种实现。

源： 协议、端口、域名

同源： 同协议、同端口、同域名

协议： http、https、file(本地的)、ftp.....

<http://localhost/2017-10->

[25/5_%E4%B8%8A%E4%BC%A02.html](http://localhost/2017-10-25/5_%E4%B8%8A%E4%BC%A02.html)

<file:///C:/xampp/php/www/2017-10->

[25/6_%E8%B7%A8%E5%9F%9F.html](file:///C:/xampp/php/www/2017-10-25/6_%E8%B7%A8%E5%9F%9F.html)

端口： 专门提供某项服务的“窗口”

域名： IP的别名（方便记）

www.baidu.com

www.miaov.com

跨域（跨源）： 不同协议、不同端口、不同域名

解决跨域： 通过新版本的XMLHttpRequest + 服务器开权限

。 'Access-Control-Allow-Origin: *'

服务器代理： 服务器文件能访问别的域下的资源，如果服务

器文件与自己的页面是同源，我就能访问别的域下的资源了。

Jsonp

jsonp = JSON + Padding

函数名 + 括号 括号中带有数据fn([]) : fn({})

ajax的数据是没有函数名的

jsonp的原理：首先，jsonp数据格式是函数名+括号的、
然后，*全局*需要定义一个与后端函数名一致的函数、

最后，当需要数据的时候，创建一个script标签，进行请求，获取数据（按需加载）。

需要执行数据的时候执行一个函数，函数内有想要的书籍，提前约定好这个函数名，后端会把数据传到函数中，实现按需加载。

```
function fn(arr){  
    alert(arr);  
}  
function fn2(arr){  
    console.log('我是div2')  
}  
div1.onclick = function(){  
    let os = document.createElement('script');  
    or.src = 'file:///C:/xampp/php/www/2017-10-25/php/qq.js';  
    document.getElementsByTagName('head')[0].appendChild(os);
```

```
        os.remove();
    }
    div2.onclick = function(){
        let os = document.createElement('script');
        os.src = 'file:///C:/xampp/php/www/2017-10-25/php/qq.js?callback=fn2';
        document.getElementsByTagName('head')[0].appendChild(os);
        os.remove();
    }
```

缓存：

当前一次请求的地址和后一次请求的地址是一样的，那么有可能出现缓存（特别是get请求在IE下特别容易被缓存）