

# Asynchronous JavaScript and XML

Ajax 介紹



# Json (Javascript Object Notation)

☕ JSON 是個以純文字為基底去儲存和傳送簡單結構資料

☕ JSON 的優點如下:

1. 相容性高

2. 格式容易瞭解，閱讀及修改方便

3. 支援許多資料格式

(number,string,booleans,nulls,array,associative array)

4. 許多程式都支援函式庫讀取或修改 JSON 資料

# Json 資料格式：名稱：值 or Key : Value

## 範例

```
{  
  "number": "1020501",  
  "name": "小傑",  
  "age": 32,  
  "sex": "M",  
  "interest": [  
    "網頁設計",  
    "撰寫文章"  
  ]  
}
```

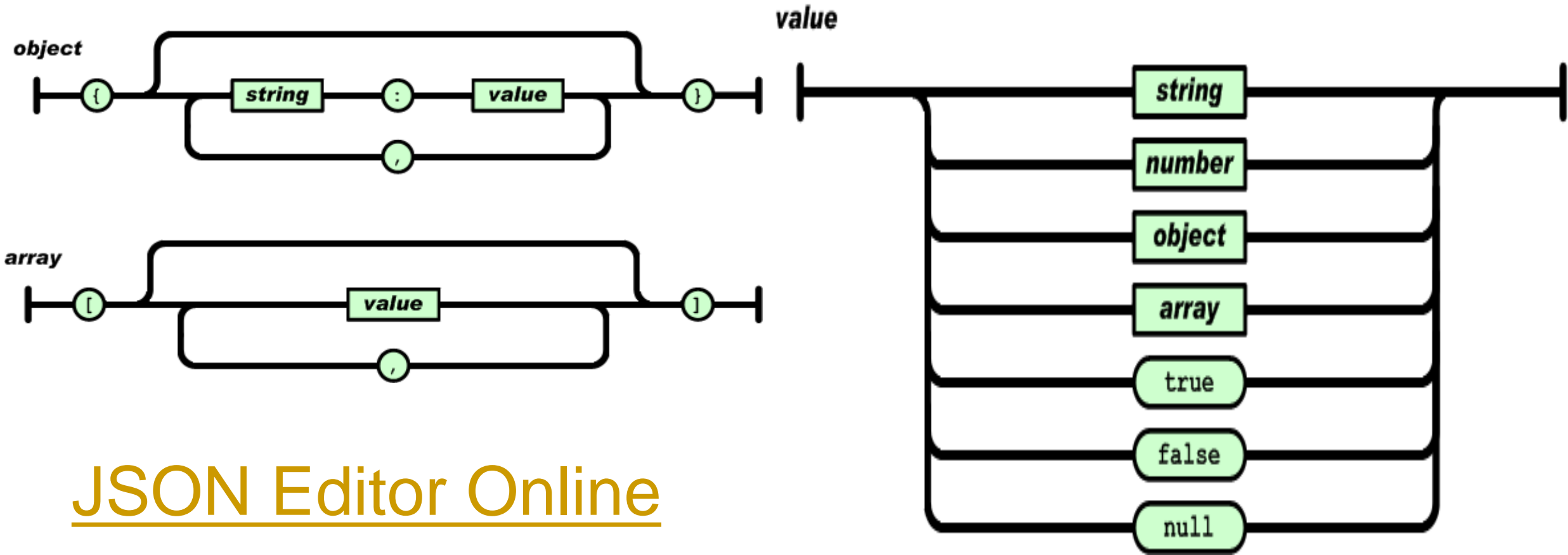
物件: { "name": value, "name": value }

1. 以 { 開始，並以 } 結束
2. name 為 string，必須以 "" 括起來
3. name 與 value 之間使用 : 分隔
4. 多個 name:value 之間使用 , 分隔

陣列: [value, value]

1. 以 [ 開始，並以 ] 結束
2. 多個 value 之間使用 , 分隔

# JSON 示意圖



[JSON Editor Online](#)

實作練習

```
[{"name": "Tom", "lastname": "Chen", "report": [{"subject": "Math", "score": 80}, {"subject": "English", "score": 90}], {"name": "Amy", "last name": "Lin", "report": [{"subject": "Math", "score": 86}, {"subject": "English", "score": 88}]}
```

# Json 資料互轉

將物件轉成JSON : `JSON.stringify()`

```
var me = { name: 'emma' };  
console.log(JSON.stringify(me));    {"name":"emma"}
```

將JSON轉成物件 : `JSON.parse()`

```
var meStr = JSON.parse('{"name":"emma"}')  
console.log(meStr);    { name: 'emma' }
```

# 表單資料轉換成JSON資料(一) : stringify

## 1. 建立物件

```
let obj={}; or let obj=new Object();
```

## 2. 新增物件元素與屬性值

```
obj.name="蘋果";  
obj.color="紅色";  
obj.price=20;
```

## 3. 建立一個陣列

```
let array=[];
```

## 4. 將物件push 給陣列

```
array.push[obj];
```

## 5. 把陣列物件轉成JSON

```
JSON.stringify(array)
```



## 表單資料轉換成JSON資料(二): 用變數串接組合

1. 設定變數, 利用 `$("#元件id").val()` 指定給變數  
`let cname=$("#className").val()`
2. 設定一個變數, 將所有變數串接成json格式  
`let result='班級:' + cname + ', 學生1:' + st1 + .....`

### HTML

```
<div>  
  班級: <input id='className' type='text' />  
  學生1 : <input id='student1' type='text' />  
  學生2 : <input id='student2' type='text' />  
</div>
```

## Ajax 簡寫用法: \$.getJSON()

```
jQuery.getJSON(url, data, success(data,status,xhr))
```

//url 必需。規定將請求傳送的哪個 URL。

//data 可選。規定連同請求傳送到伺服器的資料。

//success(data,status,xhr) 可選。規定當請求成功時執行的函式

```
$.ajax({  
    url: url,  
    data: data,  
    success: callback,  
    dataType: "json"  
});
```





# Ajax 語法結構

```
$.ajax({  
  url:'地址',  
  data:{  
    引數:xxx,  
    引數:xxx  
  },  
  type:'POST/GET',  
  success:function(str){ //成功回撥函式  
    alert(str);  
  },  
  error:function (err){ //失敗回撥函式  
    alert(err);  
  }  
});
```

# Ajax 讀取文字檔

## Test.txt 內容

1 jane,  
2 tina,  
3 Bob,  
4 Raj,  
5 RuRu,  
6 james,  
7 grace,

## Ajax 程式

```
$.ajax({  
    url: "test.txt",  
    success: function (e) {  
        let array = e.split(",");  
        $("#txt_div").html(array[0]);  
    }  
});
```

## Html 程式

```
<div id='txt_div'></div>
```

## 執行結果

1 jane

# Ajax 讀取 json 檔案

## Javascript

```
$.ajax({
  url: "json/test.json",
  contentType: 'application/json; charset=utf-8',
  type: "get",
  dataType: "json",
  success: function (data) {
    let items = [];

    $.each(data, function (index, val) {
      items.push('<li id=' + val.id + '>' + val.spot_name + ' - ' + val.address + '</li>');
      $("#area").append(items);
    });
  }
});
```

HTML: <ul id="area"></ul>

# Convert JSON to C#

## Convert Json to C# Classes Online

Convert any JSON object to a C# class online. Check out the help panel below to view details on how to use this converter.

The interface features a top navigation bar with several conversion options: JSON to C#, C# to C#, JSON to JAVA, XML to C#, and XML to JAVA. Below this is a two-pane editor. The left pane is labeled 'JSON' and contains a large icon representing a JSON file. The right pane is labeled 'C#' and contains a large icon representing a C# file. At the bottom, there are three buttons: 'Settings' with a gear icon, a green 'Convert' button with a lightning bolt icon, and a dark 'Copy To Clipboard' button with a document icon.

# 實作題: 輸入表單資料 轉換成Json 透過Ajax 傳回後端

班級:

學生1:

學生2:

- 1.設計表單
- 2.設定按鈕事件
- 3.撰寫script 事件function
  - A. 建立一個物件
  - B. 將表單欄位指定給物件
  - C. 建立一個陣列
  - D. 利用陣列.push 將物件存入
  - E. 利用JSON.stringify 將陣列轉成 json 字串
4. 用\$.ajax({ }); 將json字串傳到後端



## 實作題: C# 後端接收Json 字串, 轉換成List<T>物件

1. 透過request.InputStream 接收來自ajax送來的內容
2. 用StreamReader.ReadToEnd()讀取Request.InputStream內容
3. 利用JsonConvert.DeserializeObject<List<T>>() 將json 反序列化成為物件.
4. 取出所需要的物件, 進行處理
5. 回傳給前端訊息

註: 須引用 System.IO & Newtonsoft.Json