

JavaScript及Web网页前端开发

# 第7讲 JavaScript进阶

唐大仕

# 课程进度

12

综合应用

13

复习

10

web api

11

ajax/服务端

9

Bootstrap

8

Jquery

5

HTML5

6

CSS3

7

JavaScript进阶

2

HTML

3

CSS

4

JavaScript基础

1 序

# 学习资料

## ■ 文档

- <https://www.w3school.com.cn/jsref/index.asp> 参考文档
- <https://www.w3school.com.cn/js/index.asp> 教程
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference>
- <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

## ■ ES官网

- <https://www.ecma-international.org/>
  - <https://www.ecma-international.org/publications-and-standards/standards/>
- 其中 Standards – ECMAScript (如 ECMA-262)
- <https://github.com/tc39/ecma262>

# 本讲内容

- ES中的对象
- ES中的类
- ES中的函数
- ES内置对象
- 环境对象（ DOM、BOM ）
- ES6-12新语法

# JavaScript基础回顾

## ■ JS书写

- `<script>... </script>`

## ■ 单词、语句、段落

- 单词：var 区分大小写

- 语句：带分号（可不带）

- 段落：function xxx() { ... }

## ■ 与HTML交互

- 元素：获得DOM对象 `document.getElementById('..')`

- 对象：属性、方法、事件

- 事件：onclick onmouseover

- 周期执行：`setInterval(fun, 100);`

# ECMAScript版本

## ■ ECMA-262

- 1997 (最初的ES标准)
- 2009,2011 (ES5, 5.1)
- ES2015 ( ES6 )
- 以后每年发布一个 ECMAScript 版本，以年份命名，如ES2016,2017,.....
- 另外有相关的标准
  - ECMA 414 标准集合、409 嵌入式系统API、402 国际化、404 JSON等

## ■ 最新文档

- 正式 <https://262.ecma-international.org/>
- 草案 <https://github.com/tc39/ecma262> 或 <https://tc39.es/ecma262/>
- 更多标准 <https://www.ecma-international.org/publications-and-standards/standards/>
- 历史参见 [https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Language\\_Resources](https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Language_Resources)

# ES6中的对象

# JSON

## ■ JSON(JavaScript Object Notation)

- 是一种轻量级的数据交换格式， ECMA404标准
- 方便编写、阅读
- 用途：表示对象、数据

## ■ 例

```
var p = {  
  "name": "zhang",  
  "age": 18,  
  "email": ["a@aa.com", "b@bb.com"]  
}
```

## ■ { }表示对象； [ ] 表示数组； key与value之间用冒号

- 访问其中的属性： p.name p.age p.email[0] p["age"] p["email"][0]



# JSON相关网站

## ■ 网站

- <https://www.json.org>

## ■ 格式化工具

- <http://www.bejson.com/>

- <http://kjson.com>

- <http://json.cn/>

# 名称/值

## ■ json是“名称/值” 对的集合

- ( A collection of name/value pairs ) (key-value pairs)
- 名称是属性，值就是属性值
- 它可以理解为对象（ object ），记录（ record ），结构（ struct ），字典（ dictionary ），哈希表（ hash table ）

## ■ 名称（ 键 ）

- 在js中可以不加双引号；但要与后台或其他语言交互时，一定要加双引号

## ■ 值

- 值可以为数值、字符串、布尔值等
- 还可以为数组、对象（ 包括json对象，即对象的嵌套 ）
- 在JavaScript中，值还可以为函数（ 这可以理解为对象的“方法” method ）

# 对象的属性

## ■ `p.age` 或 `p["age"]`

- 前一种写法更直观
- 后一种写法更灵活：属性名中间可以有空格；可以为变量（字符串或Symbol）

## ■ 例：

- `document.forms["form1"]["username"].value`
- 或写成 `document.forms.form1.username.value`

# 对象属性写法很灵活

- 可以用属性，可以用下标
- 下标可以用文本（名字），也可以用数字（顺序号）
  - `document.forms[0]`
  - `document.forms["myform"]`
  - `document.forms.myform`
- `document.forms[0]["age"]`
- `document.forms[0][0]`

# 对属性的遍历

```
■ for( var key in obj ){  
■   console.log( key + "=" + obj[key] );  
■ }
```

## ■ 注：

- 对于数组，可以用in, 还可以用of

- for( var i in ary ){ }

- for( let v of ary ){ }

# JSON用途广泛

- 用于函数传递参数(如有很多选项 )
- 用于前后端数据的传递
- 与其他语言交换数据

# ES中的JSON

■ JSON用于与其他语言时，属性名要用双引号引起来

■ 但在es中比较灵活

□ 属性名可以不用引号

- `p={age:18, email:["a", "b"]}`

□ ES6以上：如果属性值是一个变量，且与属性同名，可以直接写变量名

- `age=19`

- `p2 = {age, email:[] }` 这里age相当于 `age:age`

□ 属性甚至可以是函数

- `p3={name:name, age:age, sayHello:function(){console.log(this.age);} }`

- 或`p3={name, age, sayHello(){console.log(this.age);} }`

- `p3.sayHello()`

# ES中的函数



# 函数是一等公民

## ■ 定义方式

- ❑ `function f(a, b){ let s = a+b; return s;}`
- ❑ 对象中：`{ x: 5, add(a,b){ return a+b; } }`
- ❑ 变量式赋值：`let f = function(a,b){ return a+b;}`
- ❑ lambda表达式(ES6箭头函数) `btn.onclick = ()=>{ 语句}`
  - 注意变量式赋值与function定义有区别

# 函数的参数

■ arguments

# call, apply, bind

# 闭包closure

**this指向问题**

# ES中的类

# 使用class定义类

- 使用class来定义类 ( ES6)
  - 变量, 方法, constructor
- 使用new来生成一个对象
- 使用extends来继承类
  - super

# 使用function来定义类

- class其实是一个function的语法糖
- 使用function来定义类



# 原型链

## ■ 原型链

- `p.__proto__ === Person.prototype`
- `Person.prototype.constructor === Person`

## ■ 使用原型链来实现继承

# JS内置对象

# 数组

## ■ 数组(array)

- 一组变量，用名称及下标来访问

## ■ 定义数组

- `var ages = [ 18, 19, 18, 22 ];` //用方括号
- `var ages=[ ];`

# 使用数组

## ■ 数组的长度

□ ages.length

## ■ 遍历数组

```
for(var i=0; i<ages.length; i++){           //注意.length是数组的长度
    console.log( ages[i] );                 //数组名及方括号，注意下标从0开始
}
```

另一种写法

```
for(var i in ages){
    console.log( ages[i] );
}
for(var age of ages){ //ES5
    console.log( age );
}
```

# 数组的常用方法

- `.push(a)` 在尾部放入一个元素

- `ages[ages.length]=a;`

- `.join(';')` 连接成字符串

- `.concat(a)` 连接另一数组

- `.slice(start, end)` 取数组的一部分

- 从start开始，到end结束（不包含end）

# 应用示例

- 使用数组的push及join

# ES新版中的数组

- .forEach
- 延展运算符
- of遍历及in遍历

# 内部数组

- 为了方便内部对象的操作, 可以使用一些属性或方法, 它们的结果实际上是数组 ( 准确地说类似于数组 )

- `xxx.length`   `xxx[i]`

- 常用的有 :

- `document.getElementsByName("age")` 按name属性进行选择
  - `document.getElementsByTagName("div")` 按tag进行选择
  - `document.getElementsByClassName("wrapper")` 按css类名进行选择
  - `document.forms`, 所有的表单
  - `document.links`: 所有的链接   `<a href=""></a>`
  - `document.images`: 所有的图片



# 应用示例

- 访问多个元素

# JS内置对象

- JavaScript 提供了常用的内部对象

- 如

  - Math ( 数学函数 )

  - String ( 字符串 )

  - Date ( 日期 )

- 参考

  - <http://www.w3school.com.cn/jsref/>

# Math对象

- 功能：提供除一些数学运算。如对数，平方根等。
- 主要属性
  - 常数 E 写法 Math.E
  - 3.14159的PI 写法 Math.PI
  -

# Math的方法

## ■主要方法:

- 绝对值 : `.abs()`  
正弦余弦值 : `.sin()`, `.cos()` 以弧度做单位  
四舍五入 : `.round()` `.floor` `.ceil`  
平方根 : `.sqrt()`  
几方次 : `.pow(base,exponent)`
- 随机数
  - `Math.random()`
  - ...

# 应用示例

- 使用Math.sin ( ) 函数

# 字符串对象

- String：字符串（文本）
- 其内部是用unicode来表示的

# String对象的属性

## ■ String对象的length属性

□ 它表明了字符串中的字符个数。

例：

```
mytest="This"
```

```
len=mytest.length 为4
```

注： `".length` 则为0

# String对象的方法

## ■ String对象的方法

- ❑ `.toLowerCase()` 小写转换
- ❑ `.toUpperCase()` 大写转换
- ❑ `.indexOf(charactor, fromIndex)`
  - 从指定formIndtx位置开始搜索charactor第一次出现的位置
  - `"abcd".indexOf("bc")` 为 1
- ❑ `.substring(start,end)` 返回字串的一部分
  - 从start开始到end ( 不含end ) 的字符 ( 下标都是从0开始 )
- ❑ `.charCodeAt(i)` 某个字符的ascii ( unicode ) 编码
- ❑ `String.fromCharCode(97)` 对应于编码97的字符



# 应用示例

## ■ 字符串的加密

# 日期及时间对象

- 功能：提供一个有关日期和时间的对象。
- 使用new运算符创建一个实例。例：

MyDate=new Date()

□ 注：Date内部是用从1970年1月1日00:00:00以来的毫秒数表示的。

# 获取日期的时间方法

- `getFullYear()`: 返回年数
- `getMonth()`: 返回当月号数, 注意月份是以0开始的
- `getDate()`: 返回当日号数
- `getDay()`: 返回星期几
- `getHours()`: 返回小时数
- `getMinutes()`: 返回分钟数
- `getSeconds()`: 返回秒数
- `getTime()`: 返回毫秒数

# 设置日期和时间

- `setYear()`:设置年
- `setDate()`:设置当月号数
- `setMonth()`:设置当月份数
- `setHours()`:设置小时数
- `setMinutes()`:设置分钟数
- `setSeconds()`:设置秒数
- `setTime ()`:设置毫秒数

# 示例

- 显示当前日期、时间

# JavaScript中的系统函数

- JavaScript中的系统函数又称内部方法。
- 它提供了与任何对象无关的系统函数，使用这些函数不需创建任何实例,可直接用。
- (也可以认为是window对象的方法)



**eval** 求字符串表达式中的值：

```
test=eval("8+9+5/2");
```

**parseFloat** 返回实数：

```
parseFloat("3.14");
```

**parseInt** 返回不同进制的整数：

```
parseInt("254", 10)
```

❑ `parseInt("312.78")`

❑ `parseInt("200px")`

# ES新版中的新对象

- Map (映射、字典)
- WeakMap
- Set (集合)

# 环境对象



# DOM对象

- HTMLElement
- Node
- 属性、方法、事件

# BOM对象

## ■ Browser Object Model

### □ window

- window.innerWidth

### □ screen

- screen.width

### □ navigator

- navigator.userAgent, navigator.platform

### □ history

- history.back()

### □ web api相关的 : localStorage, geolocation .....

# 其他环境对象

## ■ node.js

### □ File等

- `var fs = require("fs");`
- `var data = fs.readFileSync('input.txt');`
- `console.log("读取的内容为 " + data.toString());`

## ■ 关于全局对象

### □ globalThis

# ES的新语法

# 从ES6到ES12

- 增加了不少的语法糖

- 参考ES规范

- 参考文章

- JS语法ES6、ES7、ES8、ES9、ES10、ES11、ES12新特性

- <https://mp.weixin.qq.com/s/fWm8RaBMrcX7z0O33BOE9g>



# 本讲内容

## ■ ES中的对象

- JSON对象

## ■ ES中的类

- class对象，继承

## ■ ES中的函数

- 函数高级用法：this, 原型链，闭包，call, apply

## ■ 异步

- 事件：冒泡及两阶段；一些特殊事件，拖拽
- Promise, async, await

## ■ ES6新语法

## ■ JS对象：ES6新对象，数组，Math, Str, Date, Set, Maps

## ■ DOM与BOM对象