

第一周js免费课第一天

第一周js免费课第一天

javascript是什么？

js的组成

ECMAScript：js的规范文档，里面规定了一些js的数据类型、基本操作语句、命名规范、变量、关键字、保留字...

DOM：文档对象模型 (Document Object Model)

DOM操作的是什么？

BOM：浏览器对象模型(Browser Object Model)

BOM是操作什么的？

js的引入方式？

js的输出方式

js的命名规范

js中的变量

创建变量的语法

js中的数据类型

数字数据类型 number

Number(value) 方法

parseFloat(value)

parseInt(value)

isNaN(value) -> 返回值为布尔类型值

字符串 string

null

undefined

布尔数据类型 boolean ['buliən]

Boolean(value)

! 一个叹号 -> 取反

!! 两个叹号 -> 将其他数据类型转为布尔类型（取反再取反）

对象数据类型 object

对象类（Object）

访问对象的方式

增加属性的方式

删除属性的方式

修改属性值的方式

javascript是什么？

js是一门弱类型的脚本语言，是一门**解释型**语言

【html和css是文本语言，用来描述】

js是运行在用户端（客户端）浏览器上的语言

js运行在客户端，容易被篡改，所以安全性较低

课后思考：防篡改，负载均衡（独立的产品），API

js的组成

ECMAScript：js的规范文档，里面规定了一些js的数据类型、基本操作语句、命名规范、变量、关键

字、保留字...

ECMAScript的版本：ES5和ES6

DOM：文档对象模型（ Document Object Model ）

DOM操作的是什么？

文档对象（ document ）【html文件中的所有元素内容】

标签：由尖括号和关键字组成

元素：成对的标签和内容组成（ 内容在元素中可以为空 ）

1. `<!-- 标签 -->`
2. `<tagName>`
3. `<!-- 元素 -->`
4. `<tagName></tagName>`
5. `<tagName />`

Root Element(根元素)

根目录

```
<link rel="shortcut icon"
href="//9.url.cn/edu/lego_modules/edu-
ui/0.0.1/img/nohash/favicon.ico"/>
```

BOM：浏览器对象模型(Browser Object Model)

BOM是操作什么的？

window对象（浏览器的顶层对象）

js的引入方式？

- 行内式：将js代码块直接写在标签内

```
<input type="button" onclick="alert('欢迎来到珠峰培训')">
```

- 内嵌式：将js代码块写在 `<script>代码块</script>` 中，放在html文件里
- 外联式：通过 `<script src='url'></script>`，引入一个外部js文件

1、js引入的时候一般放在html所有内容之后

2、外联式元素内写的代码块不会生效

onclick 绑定点击事件

alert 弹出对话框

document.documentElement 获取到html元素

```
1. //单行注释
2. /*多行注释*/
3. //给html元素绑定点击事件，点击这个元素的时候，触发一个方法，弹出一段文字的方法
4.     document.documentElement.onclick=function () {
5.         alert('点不到')
6.     }
```

```
1. //写在外联式里面的代码块不会生效
2. <script src="link.js" type="text/javascript">
3.     document.write('<h1>张晓斐 ❤️</h1>');
4. </script>
```

js的输出方式

- document.write('content') – 向文档内输出，不会覆盖原文档的内容
- console 在控制台输出 【开发者工具】
 - console.log() 向控制台输出一条消息
 - console.dir() 输出一个对象的全部属性（输出结果类似于 DOM 面板中的样式）
 - console.error() 向控制台抛出异常
 - console.clear() 清空控制台
 - ctrl+L
 - 右键 clear console

- 点击禁止图标直接清空控制台
- alert() 方法用于显示带有一条指定消息（弹出）
- innerHTML/innerText 获取或者向指定元素内添加内容
 - innerHTML() 获取元素内所有的内容（包括元素）
 - innerText() 获取元素内的文本内容这种方法会将元素内原有的内容覆盖掉

js的命名规范

- 严格区分大小写
- 命名时名称中可以出现字母、数字、下划线、\$，但是数字不能为首位
- 推荐使用驼峰命名法：由多个有意义单词组成名称的时候，第一个单词首字母小写，其余单词首字母大写
- 不能使用ECMAScript中规定的关键字和保留字
- 变量命名时推荐使用匈牙利命名法【使用数据类型为名称前缀】

js中的变量

变量是用来储存信息的容器，由于js是松散数据类型，所以变量中可以放置任何的数据类型

创建变量的语法

声明 变量名; 此时这个变量中什么都没有 获取变量中的值时，返回值undefined

如何赋值 一个等号

var obj={}; 声明一个变量叫做obj将{}赋值给这个变量

“=” 在js中是赋值的意义 将等号右侧的值给到等号左侧

“+=” 保留原有值，在原有值上累加

```
num+=2;
```

```
num*=4;
```

```
num/=2;
```

```
num-=2;
```

```
num%=2; 取余数
```

一条声明可以有多个变量，每一个变量用逗号隔开

p.s：一般会相同数据类型使用一个声明

```
1.      var ary=[];
```

```
2.      var obj={};
```

```
3.      var num=1;
```

```
4.      var str='';
```

```
5.
```

```
6.      var ary=[],obj={},num=1,str='';
```

js中的数据类型

- 基本数据类型：

包括数字（number），字符串（string），布尔

(boolean) , null , undefined

- 引用数据类型：包括对象数据类型和函数数据类型
 - 对象数据类型：包括数组类 (Array) , 对象类 (Object) , 正则类 (RegExp)
 - 函数数据类型：function

数字数据类型 number

包括 整数 (正整数 0 负整数) 小数 (浮点数 Float)

NaN (not a number)

Number(value) 方法

这种方法是**强制**转化，只要转化的对象中有一个内容不是数字，那么就会返回NaN

作用 (应用场景)：将其他数据类型转为数字数据类型

返回值 (最终结果)：数字或者NaN

如果转化的是字符串，直接调用Number() 方法将字符串转为数字，如果字符串中有一项不是数字返回NaN

如果转化的是对象数据类型，先调用.toString() 方法，将数组中的内容转为字符串，再用上一条规则转为数字

如果转化布尔类型，true → 1 false → 0

null → 0

undefined → NaN

parseFloat(value)

作用：将**字符串**转化为数字（可转化为浮点数）

机制：将值从左到右依次查找，遇到第一个非有效数字停止查找，将之前的值返回

返回值：数字或NaN

```
1. parseFloat('.56px');  
2. .56  
3. 0.56  ✓  
4. NaN   ×  
5. parseFloat(.23.6);  
6. // Uncaught SyntaxError: missing ) after  
   argument list
```

parseInt(value)

作用：将**字符串**转化为一个整数

机制：将值从左到右查找，找到第一个非有效整数停止，将查找到的值返回

返回值：数字或NaN

```
1. parseInt('.36px')
2. NaN
3.
4. parseInt(.36px)
5. 报错 Uncaught SyntaxError: Invalid or unexpected token
6.
7. parseInt(.36)
8. 0
9.
```

isNaN(value) → 返回值为布尔类型值

is not a number 这不是一个数字

作用：判断一个值不是数字

不是数字的时候返回 true

是数字的时候返回 false

机制：isNaN在判断之前会先将非数字数据类型的值，调用Number()方法，将其转化为数字数据类型，如果Number()转化结构为NaN的时候，isNaN()返回true，反之为false

返回值：true或false

```
1. isNaN(36) --> false
```

“==”在js中是比较的意思，返回一个布尔类型值

相等返回true 不相等返回false

NaN==NaN -> false

Number()

parseFloat()

parseInt()

字符串 string

是用来储存字符的变量，字符串在js中没有任何意义

null

现在预留位置为空，之后再添加值

undefined

现在没有，将来也没有

布尔数据类型 boolean ['buliən]

布尔有两个值 true 和 false

判断一个值是真是假

在js中 0、NaN、”、null、undefined 为假，其余都为真

Boolean(value)

作用：判断一个值是真是假

返回值：true或false

判断一个条件是否成立

! 一个叹号 -> 取反

机制：先将!后面的值调用Boolean()方法转为布尔类型后取反

返回值：true或false

```
1. console.log(!0) --> true
2. 1.Boolean(0)--> false
3. 2.!false-->true
```

!! 两个叹号 -> 将其他数据类型转为布尔类型 (取反再取反)

和Boolean()方法得到的值相同

返回值：true或false

```
1. console.log (!!0) --> false
2. 1.Boolean(0) --> false
3. 2.!false-->true
4. 3.!true-->false
```

对象数据类型 object

包括数组类（Array），对象类（Object），正则类（RegExp）.....

对象类（Object）

由键值对组成，这些键值对是用来描述这个对象的，每一对键值对之间用逗号隔开

键值对由属性名和属性值组成 key:value

创建对象的实例

```
1. //字面量创建方式
2. var obj1={
3.     key:'value',
4.     key1:'value1'
5. }
6. //实例创建方式
7. var obj2=new Object();
```

对象的属性：

- 内置属性：对象天生自带的
 - 自定义属性：人为定义的
- 对象的属性是用来描述对象的，还可以储存值

js中用来储存值的方式

- 变量

- 属性

对象的属性名可以为数字

访问对象的方式

- 对象名.属性名
- 对象名['属性名']

增加属性的方式

自定义属性的过程，需要增加属性名和属性值

- 对象名.新的属性名=属性值
- 对象名['新的属性名']=属性值

删除属性的方式

delete 对象名.属性名

delete 对象名[属性名]

修改属性值的方式

就是将之前有的属性名里面的属性值重新赋值

对象名.属性名=新的属性值

对象名['属性名']=新的属性值

属性名为数字的时候，获取属性的方式为：对象名['属性名']，这个方法也适用其他属性名，属性名为数字的时候可以省略引号，其他属性名省略引号，系统默认这个属性名为变量

明天的内容：

- 1、基本数据类型和引用数据类型的区别
- 2、数据类型之间的比较
- 3、js的基本语句 判断和循环
- 4、数据类型的检测
- 5、字符串的拼接

晚上7-9点补课函数