

# JavaScript 基础第五天

对象







1. 知道对象数据类型的特征,具备对象数组数据渲染页面的能力。





- ◆ 对象
- ◆ 综合案例





# 对象

- 什么是对象
- 对象使用
- 遍历对象
- 内置对象





// *这3个180是啥意思? 你猜*~~ let arr = [180, 180, 180]

- 1. 保存网站用户信息,比如姓名,年龄,电话号码…用以前学的数据类型方便吗?
  - ▶ 不方便,很难区分
- 2. 我们是不是需要学习一种新的数据类型,可以详细的描述某个事物?
  - ▶ 姓名
  - ▶ 年龄
  - ▶ 电话
  - > ...



#### 1. 对象是什么

- 对象(object): JavaScript里的一种数据类型
- 可以理解为是一种无序的数据集合, 注意数组是有序的数据集合
- 用来描述某个事物,例如描述一个人
  - ▶ 人有姓名、年龄、性别等信息、还有吃饭睡觉打代码等功能
  - ▶ 如果用多个变量保存则比较散,用对象比较统一
- 比如描述 班主任 信息:
  - ▶ 静态特征 (姓名, 年龄, 身高, 性别, 爱好) => 可以使用数字, 字符串, 数组, 布尔类型等表示
  - ▶ 动态行为 (点名, 唱, 跳, rap) => 使用函数表示

```
let obj = {
  uname: 'pink老师',
  age: 18,
  gender: '女'
}
```

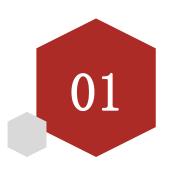




#### 1. 对象是什么?

- ▶ 对象是一种数据类型
- > 无序的数据的集合
- 2. 对象有什么特点?
  - > 无序的数据的集合
  - ▶ 可以详细的描述描述某个事物





# 对象

- 什么是对象
- 对象使用
- 遍历对象
- 内置对象



目标:掌握对象语法,用它保存多个数据

● 1. 对象声明语法

let 对象名 = {}

例如:

// 声明了一个person的对象 let person = {}

实际开发中,我们多用花括号。 {} 是对象字面量

let 对象名 = new Object()



• 2. 对象有属性和方法组成

▶ 属性:信息或叫特征(名词)。比如 手机尺寸、颜色、重量等...

▶ 方法:功能或叫行为(动词)。比如 手机打电话、发短信、玩游戏...

▶ 你还能举例吗?

```
let 对象名 = {
属性名: 属性值,
方法名: 函数
}
```



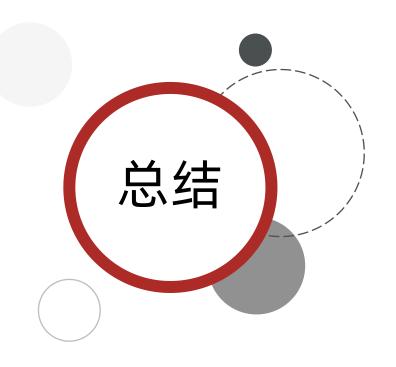


- 3. 属性
- 数据描述性的信息称为属性,如人的姓名、身高、年龄、性别等,一般是名词性的。

```
let obj = {
  uname: 'pink老师',
  age: 18,
  gender: '女'
}
```

- ▶ 属性都是成 对出现的,包括属性名和值,它们之间使用英文: 分隔
- ▶ 多个属性之间使用英文 , 分隔 最后-个不用写
- ▶ 属性就是依附在对象上的变量(外面是变量,对象内是属性)
- ▶ <mark>属性名可以使用 "" 或 '', 一般情况下省略</mark>, 除非名称遇到特殊符号如空格、中横线等





- 1. 对象属性有顺序吗?
  - ▶ 没有
- 2. 属性和值用什么符号隔开? 多个属性用什么隔开?
  - ▶ 属性和值用;隔开
  - ▶ 多个属性用, 逗号隔开

```
let obj = {
  uname: 'pink老师',
  age: 18,
  gender: '女'
}
```





### 请声明一个产品对象,里面包如下信息:

要求:

商品对象名字: goods

商品名称命名为: name

商品编号: num

商品毛重: weight

商品产地: address

品牌: 小米 (MI)

商品名称: 小米小米10 青春版 商品编号: 100012816024 商品毛重: 0.55kg 商品产地: 中国大陆



■ 对象本质是无序的数据集合,操作数据无非就是 增 删 改 查 语法:



查询对象

对象.属性

查

改

重新赋值

对象.属性 = 值

对象添加新的数据对象名.新属性名 = 新值





删除对象中属性

delete 对象名.属性名



- 3. 属性-查
- 声明对象,并添加了若干属性后,可以使用. 获得对象中属性对应的值,我称之为属性访问。
- 语法:对象名.属性
- 简单理解就是获得对象里面的属性值。

```
let person = {
   uname: 'pink老师',
   age: 18,
   gender: '女'
}
console.log(person.uname)
console.log(person.age)
console.log(person.gender)
```



- 3. 属性-改
- 语法:对象名.属性 = 新值

```
let person = {
    uname: 'pink老师',
    age: 18,
    gender: '女'
}

person.gender = '男'
console.log(person.gender) // 修改了为男
console.log(person)
```



- 3. 属性-增
- 语法:对象名.新属性 = 新值

```
let person = {
    uname: 'pink老师',
    age: 18,
    gender: '女'
}

person.address = '武汉黑马'
console.log(person)

couzoje:jog(belzou)
belzou:genderse = belzous
```

```
▼{uname: 'pink老师', age: 18, gender: '女', address: '武汉黑马']
address: "武汉黑马"
age: 18
gender: "女"
uname: "pink老师"
```



- 3. 属性-删 (了解)
- 语法: delete 对象名. 属性

```
let person = {
  uname: 'pink老师',
  age: 18,
  gender: '女'
}
delete person.gender // 删除 gender属性
console.log(person)
couzoje:jog(berson)
```

```
▼{uname: 'pink老师', age: 18}
age: 18
uname: "pink老师"
```





- 1. 对象查语法如何写?
  - ▶ 对象名.属性
- 2. 对象改语法如何写:
  - ▶ 对象名.属性 = 新值
- 3. 对象增语法如何写:
  - ▶ 对象名.新属性名 = 新值

```
let person = {
    uname: 'pink老师',
    age: 18,
    gender: '女'
}
console.log(person.age) // 查语法
person.gender = '男' // 改语法
person.address = '武汉黑马' // 增语法

betsou'seques = "弘汉黑马' // 增语法
```

改和增语法一样,判断标准就是对象有没有这个属性,没有就是新增,有就是改





#### • 请对产品对象,做如下操作:

#### 要求:

1. 请将商品名称里面的值修改为: 小米10 PLUS

2. 新增一个属性颜色 color 为 '粉色'

3. 请依次页面打印输出所有的属性值

原先:

商品对象名字: goods

商品名称命名为: name

商品编号: num

商品毛重: weight

商品产地: address

品牌: 小米 (MI)

商品名称: 小米小米10 青春版

商品编号: 100012816024

商品毛重: 0.55kg

商品产地: 中国大陆



目标: 掌握查找属性的另外写法

- 3. 属性-查的另外一种写法
- 对于多词属性或则 等属性,点操作就不能用了。
- 我们可以采取: 对象['属性'] 方式, 单引号和双引号都阔以

```
let person = {
  'user-name': 'pink老师',
  age: 18,
  gender: '女',

}
console.log(person.user-name) // NaN
cousoje.jog(berson.user-name) // NaN
```

```
let person = {
  'user-name': 'pink老师',
  age: 18,
  gender: '女',
}
console.log(person['user-name']) // pink老师
console.log(person['user-name']) // pink老师
```



目标: 掌握查找属性的另外写法

- 3. 属性-查的另外一种写法
- 对象['属性'] 方式, 单引号和双引号都阔以
- 也可以用于其他正常属性,比如:
- []语法里面的值如果不添加引号 默认会当成变量解析

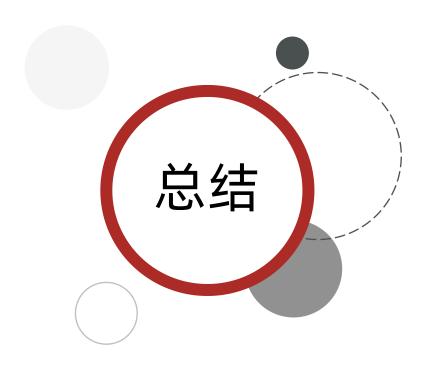
#### 总结:

没有必要的时候直接使用点语法,在需要解析变量的时候使用[]语法

```
let person = {
    'user-name': 'pink老师',
    age: 18,
    gender: '女',
}
console.log(person['user-name']) // pink老师
console.log(person['age']) // 18

couzoje:jog(berson['age']) // 18
```





- 1. 对象访问属性有哪两种方式?
  - ▶ 点形式 对象.属性
  - ▶ []形式 对象['属性']
- 2. 两种方式有什么区别?
  - ▶ 点后面的属性名一定不要加引号
  - ▶ [] 里面的属性名一定加引号
  - ▶ 后期不同使用场景会用到不同的写法

练习:请利用[]语法把对象里面的属性依次打印出来



- 5. 对象中的方法
- 数据行为性的信息称为方法,如跑步、唱歌等,一般是动词性的,其本质是函数。

```
let person = {
    name: 'andy',
    sayHi: function() {
        document.write('hi~~')
    }
}
```

- 1. 方法是由方法名和函数两部分构成,它们之间使用:分隔
- 2. 多个属性之间使用英文 , 分隔
- 3. 方法是依附在对象中的函数
- 4. 方法名可以使用 "" 或 '', 一般情况下省略, 除非名称遇到特殊符号如空格、中横线等



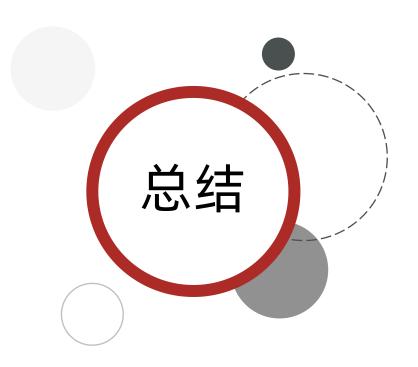
- 6. 对象中的方法 可以在对象外直接增加方法!
- 声明对象,并添加了若干方法后,可以使用. 调用对象中函数,我称之为方法调用。
- 也可以添加形参和实参

```
let person = {
    name: 'andy',
    sayHi: function () {
        document.write('hi~~')
    }
}
// 对象名.方法名()
person.sayHi()

实参
```

课堂练习: 5分钟 给对象增加唱歌和跳舞的方法,并打印输出

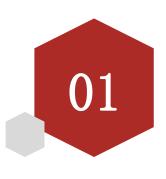




- 1. 对象访问方法是如何实现的?
  - ▶ 对象.方法()
  - person.sayHi()
- 2. 对象方法可以传递参数吗?
  - ▶ 可以,跟函数使用方法基本一致

```
let person = {
    name: 'andy',
    sayHi: function () {
        document.write('hi~~')
    }
}
// 对象名.方法名()
person.sayHi()
```





# 对象

- 什么是对象
- 对象使用
- 遍历对象
- 内置对象



#### 1. 4 遍历对象

目标: 能够遍历输出对象里面的元素

#### ● 遍历对象

for 遍历对象的问题:

- ▶ 对象没有像数组一样的length属性,所以无法确定长度
- ▶ 对象里面是无序的键值对,没有规律.不像数组里面有规律的下标



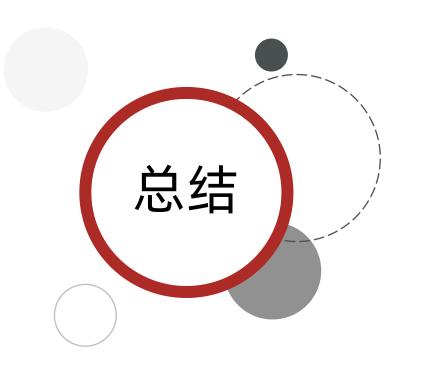
#### 1. 4 遍历对象

#### ● 遍历对象

```
let obj = {
    uname: 'andy',
    age: 18,
    sex: '男'
}
for (let k in obj) {
    console.log(k) // 打印属性名
    console.log(obj[k]) // 打印属性值
}
```

- 一般不用这种方式遍历数组、主要是用来遍历对象
- for in语法中的 k 是一个变量, 在循环的过程中依次代表对象的属性名 k是字符串类型的键值
- 由于 k 是变量, 所以必须使用 [] 语法解析
- 一定记住: k 是获得对象的属性名, 对象名[k] 是获得属性值





- 1. 遍历对象用那个语句?
  - >for in
- 2. 遍历对象中, for k in obj,获得对象属性是那个, 获得值是那个?
  - ➤ 获得对象属性是 k
  - ➤ 获得对象值是 obj[k]





### 遍历数组对象

需求: 请把下面数据中的对象打印出来:

```
// 定义一个存储了若干学生信息的数组
let students = [
    {name: '小明', age: 18, gender: '男', hometown: '河北省'},
    {name: '小红', age: 19, gender: '女', hometown: '河南省'},
    {name: '小刚', age: 17, gender: '男', hometown: '山西省'},
    {name: '小丽', age: 18, gender: '女', hometown: '山东省'}
]
```





# 遍历数组对象

需求: 根据以上数据渲染生成表格

#### 学生信息

将数组中存储的学生信息,以表格的形式输出到网页中

#### 学生列表

序号	姓名	年龄	性别	家乡
1	小明	18	男	河北省
2	小红	19	女	河南省
3	小刚	17	男	山西省
4	小丽	18	女	山东省





# 遍历数组对象

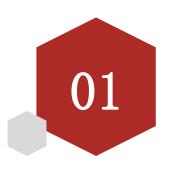
需求: 根据以上数据渲染生成表格

分析:

1. 打印表格 头部和尾部

2. 中间的行遍历数组,然后填充对象数据





# 对象

- 什么是对象
- 对象使用
- 遍历对象
- 内置对象



# 1.5 内置对象

目标: 学会调用JavaScript为我们准备好的内置对象

#### 学习路径

- 1. 内置对象是什么
- 2. 内置对象Math
- 3. 生成任意范围随机数



### 1.5 内置对象-内置对象是什么?

- JavaScript内部提供的对象,包含各种属性和方法给开发者调用
- 思考:我们之前用过内置对象吗?
  - > document.write()
  - > console.log()





#### 1.5 内置对象-Math

- 介绍: Math对象是JavaScript提供的一个"数学"对象
- 作用:提供了一系列做数学运算的方法
- Math对象包含的方法有:
  - ➤ random: 生成0-1之间的随机数(包含0不包括1)
  - > ceil: 向上取整
  - ➤ floor: 向下取整
  - ▶ max: 找最大数
  - ▶ min: 找最小数
  - ▶ pow: 幂运算
  - ➤ abs: 绝对值
  - ▶ Math对象在线文档



#### 1.5 内置对象-生成任意范围随机数

- Math. random() 随机数函数, 返回一个0 1之间,并且包括0不包括1的随机小数 [0, 1)
- 如何生成0-10的随机数呢?

● 如何生成5-10的随机数?

● 如何生成N-M之间的随机数





### 随机点名案例

需求:请把 ['赵云','黄忠','关羽','张飞','马超','刘备','曹操'] 随机显示一个名字到页面中

分析:

①:利用随机函数随机生成一个数字作为索引号

②: 数组[随机数] 生成到页面中





#### 随机点名案例改进

需求:请把 ['赵云','黄忠','关羽','张飞','马超','刘备','曹操'] 随机显示一个名字到页面中,但是不允许重复显示

分析:

①:利用随机函数随机生成一个数字作为索引号

②:数组[随机数]生成到页面中

③:数组中删除刚才抽中的索引号





## 猜数字游戏

需求:程序随机生成 1~10 之间的一个数字,用户输入一个数字

①: 如果大于该数字,就提示,数字猜大了,继续猜

②:如果小于该数字,就提示,数字猜小了,继续猜

③:如果猜对了,就提示猜对了,程序结束





#### 猜数字游戏

需求:程序随机生成 1~10 之间的一个数字,用户输入一个数字

分析:

①:利用随机数生成一个数字

②:需要一直猜,所以需要不断的循环

③:因为条件是结果猜对了,就是判断条件退出,用while循环合适

④:内部判断可以用多分支语句



# 1 案例

#### 生成随机颜色

需求:该函数接收一个布尔类型参数,表示颜色的格式是十六进制还是rgb格式。

①:如果参数传递的是true或者无参数,则输出一个随机十六进制的颜色

②:如果参数传递的是false,则输出一个随机rgb的颜色

③: 格式:

function getRandomColor(flag){

}

console.log(getRandomColor(true)) //#ffffff

console.log(getRandomColor(false)) //rgb(255,255,255)

分析:

提示: 16进制颜色格式为: '#ffffff' 其中f可以是任意 0-f之间的字符

提示: rgb颜色格式为: 'rgb(255,255,255)' 其中255可以是任意0-255之间的数字



# 1 案例

### 生成随机颜色

需求:该函数接收一个布尔类型参数,表示颜色的格式是十六进制还是rgb格式。

分析:

提示: 16进制颜色格式为: '#ffffff' 其中f可以是任意 0-f之间的字符,需要用到数组,

let arr = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'a', 'b', 'c', 'd', 'e', 'f']

提示: rgb颜色格式为: 'rgb(255,255,255)' 其中255可以是任意0-255之间的数字

步骤:

①:如果参数为true或者无参数,则处理16进制颜色,核心思想是循环6次,生成随机的6个数字(取值范围0~15),根据这个数字去找数组的值,然后和#拼接起来,并且返回值。

②:如果参数为false,随机生成一个0~255的数给三个变量,分别作为 r g b 三个颜色,之后拼接字符串rgb(255,255,255)格式





- ◆ 对象
- ◆ 综合案例





#### 学成在线页面渲染案例

需求:根据数据渲染列表页面







## 学成在线页面渲染案例

需求: 根据数据渲染列表页面

分析:

①:根据数据来渲染页面



## 拓展-术语解释

目标:知道一些术语,让自己更专业

术语	解释	举例	
关键字	在JavaScript中有特殊意义的词汇	let、var、function、if、else、 switch、case、break	
保留字	在目前的JavaScript中没意义,但未 来可能会具有特殊意义的词汇	int、short、long、char	
标识(标识符)	变量名、函数名的另一种叫法	无	
表达式	能产生值的代码,一般配合运算符出 现	10 + 3、age >= 18	
语句	一段可执行的代码	If () for()	



#### 拓展-基本数据类型和引用数据类型

目标:了解基本数据类型和引用数据类型的存储方式

简单类型又叫做基本数据类型或者值类型,复杂类型又叫做引用类型。

- 值类型:简单数据类型/基本数据类型,在存储时变量中存储的是值本身,因此叫做值类型 string, number, boolean, undefined, null
- 引用类型:复杂数据类型,在存储时变量中存储的仅仅是地址(引用),因此叫做引用数据类型通过 new 关键字创建的对象(系统对象、自定义对象),如 Object、Array、Date等



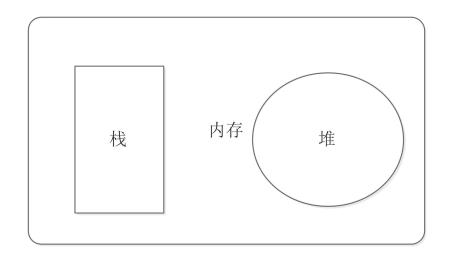
#### 堆栈空间分配区别:

1、栈(操作系统):由操作系统自动分配释放存放函数的参数值、局部变量的值等。其操作方式类似于数据结构中的 栈;

#### 简单数据类型存放到栈里面

2、堆(操作系统):存储复杂类型(对象),一般由程序员分配释放,若程序员不释放,由垃圾回收机制回收。

#### 引用数据类型存放到堆里面





## 1. 简单类型的内存分配

● 值类型 (简单数据类型): string , number, boolean, undefined, null

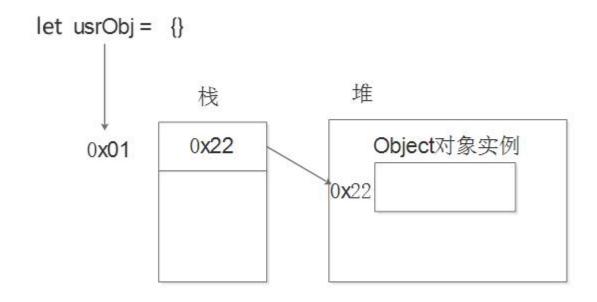
● 值类型变量的数据直接存放在变量(栈空间)中

		栈	堆	
let usrAge = 12;	0x11	12		
let usrName = '小白';	<b>0x</b> 20	小白		
let usrSex = 'true';	<b>0x3</b> 2	true		



## 2. 复杂类型的内存分配

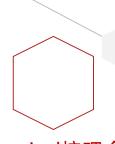
- ullet 引用类型(复杂数据类型):通过 new 关键字创建的对象(系统对象、自定义对象) $_{n}$  如 Object、Array、Date等
- 引用类型变量(栈空间)里存放的是地址,真正的对象实例存放在堆空间中







```
let num = 10
let num2 = num
num = 20
console.log(num2) // ?
let obj1 = {
  age: 18
let obj2 = obj1
obj1.age = 20
console.log(obj2) // ?
```



# 今日复习路线

- 1. 晚自习回来每个同学先必须xmind梳理今日知识点 (md 笔记也行)
- 2. 写遍学成在线案例
- 3. 写3遍返回随机颜色案例
- 4. 开始做检测题: PC端地址: https://ks.wjx.top/vj/tHu7X7y.aspx
- 5. 开始做作业 见课后作业 06-作业
- 6. 总结js基础阶段内容-开始准备 js基础实战内容,可以先准备笔试题







我们什么都没有, 唯一的本钱就是青春。梦想让我与众不同, 奋斗让我改变命运!



传智教育旗下高端IT教育品牌