

JavaScript 第五天

对象

学习目标

Learning Objectives

1. 掌握对象相关知识，具备对象数组数据渲染页面的能力



目录

Contents

- ◆ 对象
- ◆ 综合案例



对象

- 什么是对象
- 对象使用
- 操作对象
- 遍历对象
- 内置对象



思考

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
 - ...

1.1 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
- 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



总结

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
- 操作对象
- 遍历对象
- 内置对象

1.2 对象使用

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 = {}
```

1.2 对象使用

目标：掌握对象语法，用它保存多个数据

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  
}
```



目标：掌握对象语法，用它保存多个数据

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 person = {  
  uname: 'andy',  
  age: 18,  
  sex: '男'  
}
```

- 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
- 属性就是依附在对象上的变量（外面是变量，对象内是属性）
- 属性名可以使用 `"` 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



总结

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

课堂单独案例

请声明一个产品对象，里面包如下信息：

要求：

1. 对象是一个产品信息可以命名为： goods
2. 商品名称命名为： name
3. 商品编号： num
4. 商品毛重： weight
5. 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

1.2 对象使用

目标：掌握对象语法，用它保存多个数据

4. 属性访问

声明对象，并添加了若干属性后，可以使用 `.` 或 `[]` 获得对象中属性对应的值，我称之为属性访问。

简单理解就是获得对象里面的属性值。

```
let person = {  
  name: 'andy',  
  age: 18,  
  sex: '男'  
}  
console.log(person.name)  
console.log(person.age)
```

或者

```
let person = {  
  name: 'andy',  
  age: 18,  
  sex: '男'  
}  
console.log(person['name'])  
console.log(person['age'])
```



总结

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再加引号
- [] 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

课堂练习：5分钟

把刚才写的商品的对象，打印输出其中的属性

目标：掌握对象语法，用它保存多个数据

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

1.2 对象使用

目标：掌握对象语法，用它保存多个数据

6. 对象中的方法访问

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。

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

```
person.sayHi()
```

注意：千万别忘了给方法名后面加小括号



总结

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



对象

- 什么是对象
- 对象使用
- 操作对象
- 遍历对象
- 内置对象

1.3 操作对象

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 或者 对象['属性']
对象.方法()

查

重新赋值

对象.属性 = 值
对象.方法 = function() {}

改

对象添加新的数据

对象名.新属性名 = 新值

增

删除对象中属性

delete 对象名.属性名

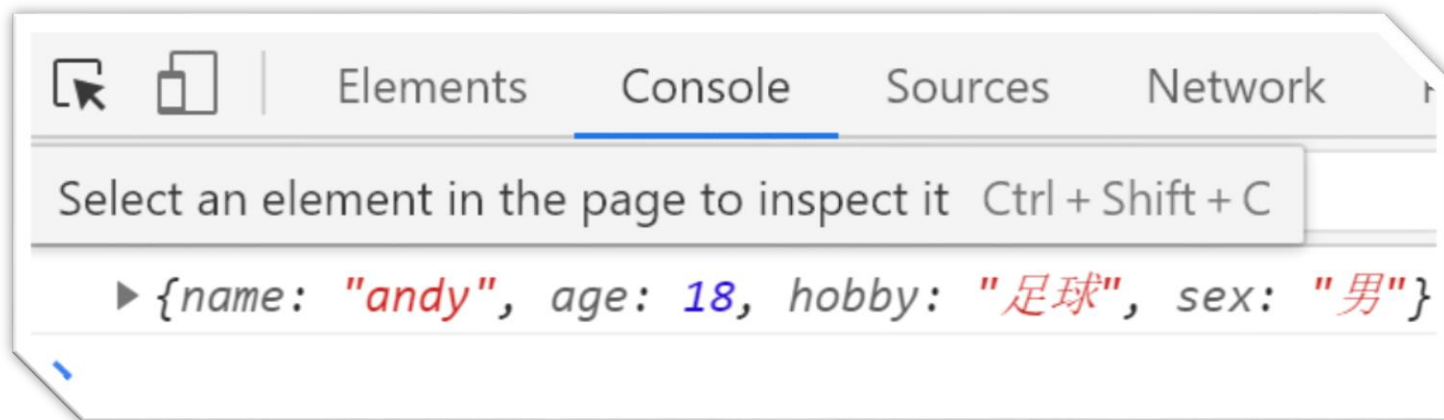
删

1.3 操作对象

增加属性

也可以动态为对象添加属性，动态添加与直接定义是一样的，只是语法上更灵活。

```
let person = {  
  name: 'andy',  
  age: 18,  
}  
person.hobby = '足球'  
person['sex'] = '男'  
console.log(person)
```



课堂练习：2分钟

把刚才写的商品的对象，追加一个 price 属性 加个为 3999

一. 对象的介绍和语法

新增对象中的方法

也可以动态为对象添加方法，动态添加与直接定义是一样的，只是语法上更灵活

```
person.move = function() {  
    document.write('移动一点点')  
}
```

注：无论是属性或是方法，同一个对象中出现名称一样的，后面的会覆盖前面的。

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



总结

1. 对象是什么？

- 一种数据类型
- 可以理解为一堆数据的集合
- 用来表示某个事物

2. 对象的是如何声明和使用的？

- 用 {} 把一堆数据包起来
- 多个属性和方法之间用逗号隔开
- console.log 输出能方便查看对象里有哪些属性和方法

3. 如何动态添加属性和方法？

- 对象如果有这个属性相当于重新赋值
- 对象如果没有这个属性相当于动态添加一个属性



对象

- 什么是对象
- 对象使用
- 操作对象
- 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

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

目标：能够遍历输出对象里面的元素

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
- 一定记住：**k** 是获得对象的**属性名**，**对象名[k]** 是获得 **属性值**



总结

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'}  
]
```

案例

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
- 操作对象
- 遍历对象
- 内置对象

1.5 内置对象

目标：掌握内置对象，调用JavaScript为我们准备好的功能

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

2.1 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()`



2.2 内置对象Math

- Math对象是JavaScript提供的一个“数学高手”对象
- 提供了一系列做数学运算的方法
- 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](#)

2.3 生成任意范围随机数

- 如何生成0-10的随机数呢？

```
Math.floor(Math.random() * (10 + 1))
```

- 如何生成5-10的随机数？

```
Math.floor(Math.random() * (5 + 1)) + 5
```

- 如何生成N-M之间的随机数

```
Math.floor(Math.random() * (M - N + 1)) + N
```

案例

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

案例

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



案例

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

案例

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



目录

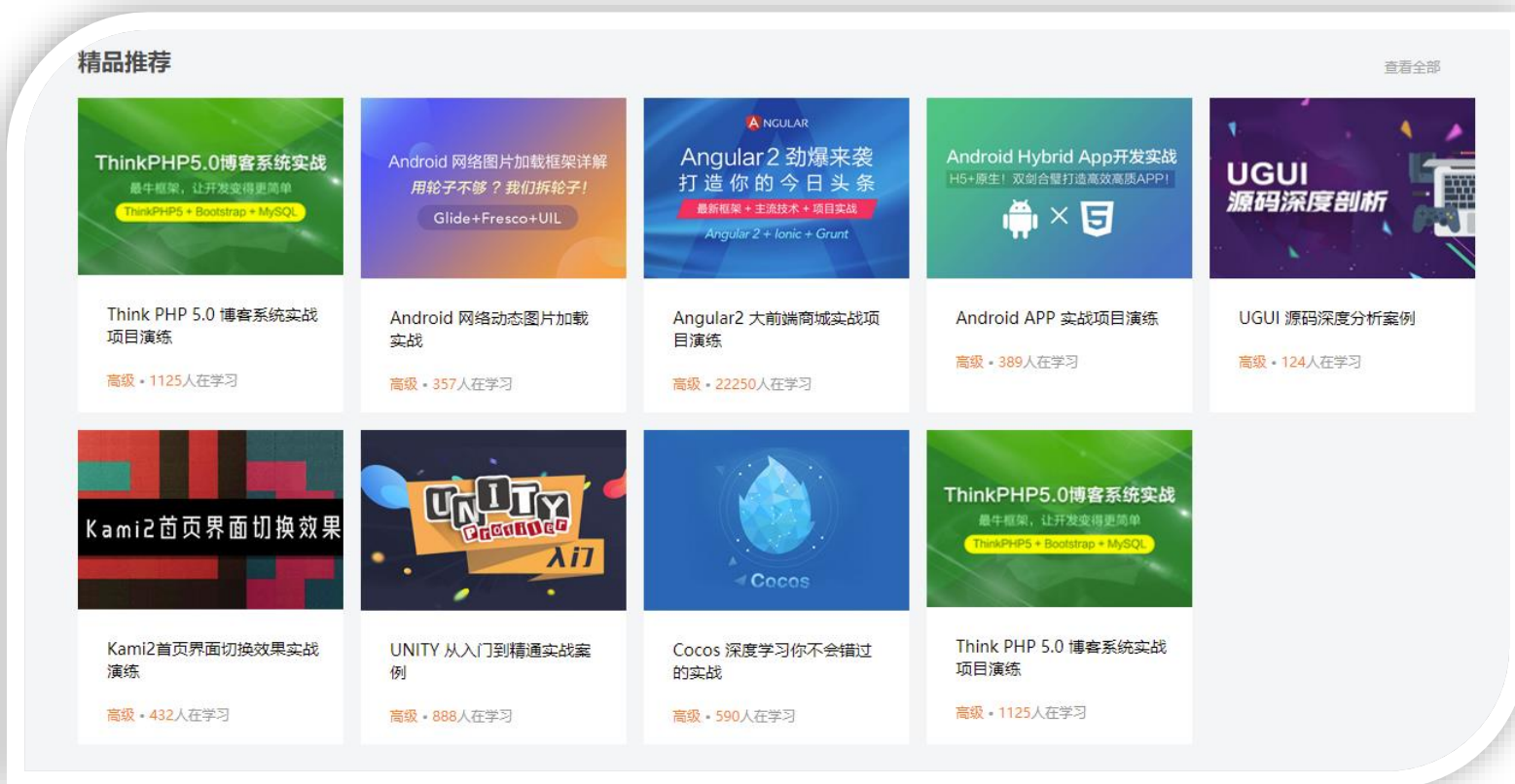
Contents

- ◆ 对象
- ◆ 综合案例

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

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	一句代码也称之为一条语句，一般按用途还会分类：输出语句、声明语句、分支语句	无

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

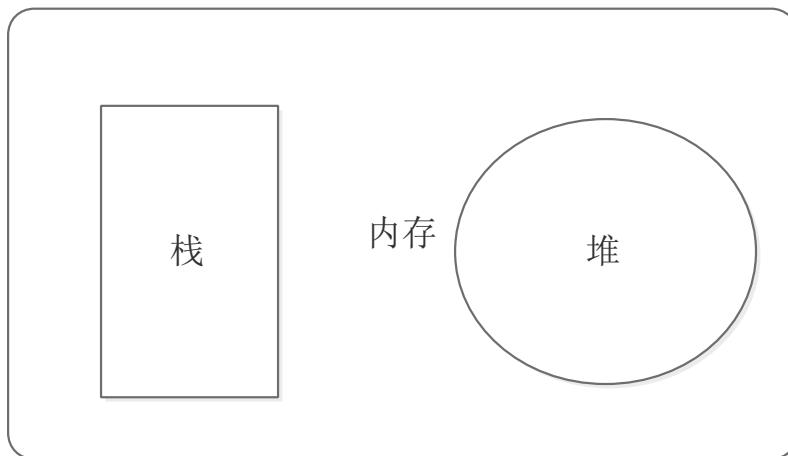
堆栈空间分配区别：

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；

简单数据类型存放在栈里面

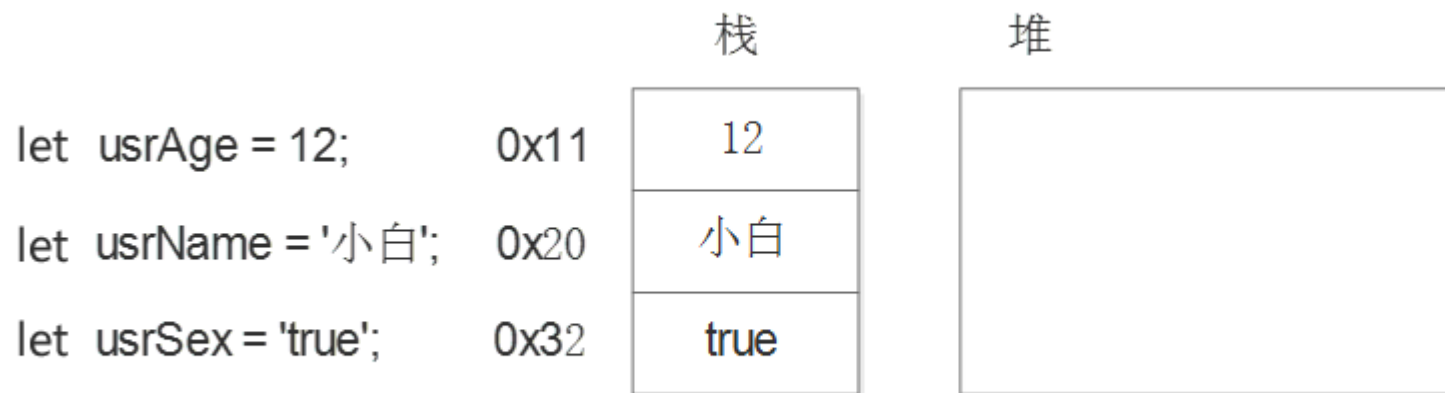
2、堆（操作系统）：存储复杂类型(对象)，一般由程序员分配释放，若程序员不释放，由垃圾回收机制回收。

引用数据类型存放在堆里面



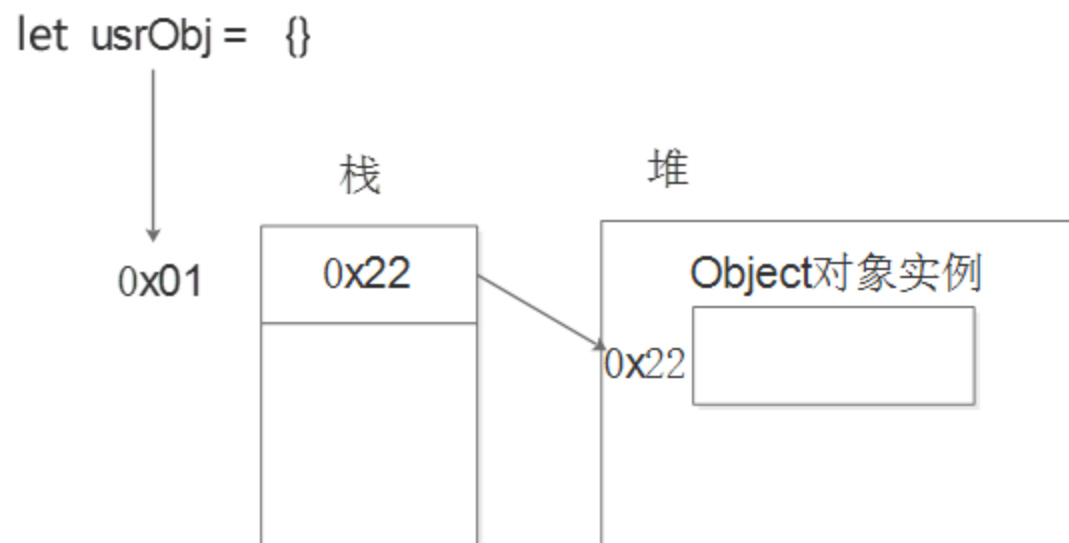
1. 简单类型的内存分配

- 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



2. 复杂类型的内存分配

- 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），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



思考

```
let num = 10
let num2 = num
num = 20
console.log(num2) // ?
```

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

```
console.log(obj2) // ?
obj2.age = 20
let obj2 = obj1
```


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2遍学生信息表案例
3. 写2遍学成在线案例
4. 复习总结今天学习笔记
5. 总结js基础阶段内容

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！



再见 祝你们幸福



传智教育旗下高端IT教育品牌