

JavaScript 基础第一天

直播课



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



目录

Contents

- ◆ 理解什么是数据并知道数据的分类
- ◆ 理解变量是存储数据的“容器”
- ◆ 知道 JavaScript 数据类型转换的特征

学习目标

Learning Objectives

1. JavaScript介绍
2. 变量
3. 数据类型
4. 类型转换
5. 实战案例



JavaScript 介绍

- JavaScript 是什么
- JavaScript 书写位置
- JavaScript 的注释
- JavaScript 的结束符
- 输入和输出语句

1.1 JavaScript 是什么

目标：初步了解 JavaScript

1. Web 标准的构成 ☆

标准	内容	说明
结构	HTML	网页元素的 结构 和 内容
表现	CSS	网页元素的 外观 和 位置 ，包括：版式、颜色、大小等
行为	JavaScript	网页模型的 定义 与 交互

2. JavaScript 是一种运行在**客户端（浏览器）**的编程语言，可以用来创建动态更新的内容，控制多媒体，制作图像动画等等实现**人机交互效果**。

3. 作用

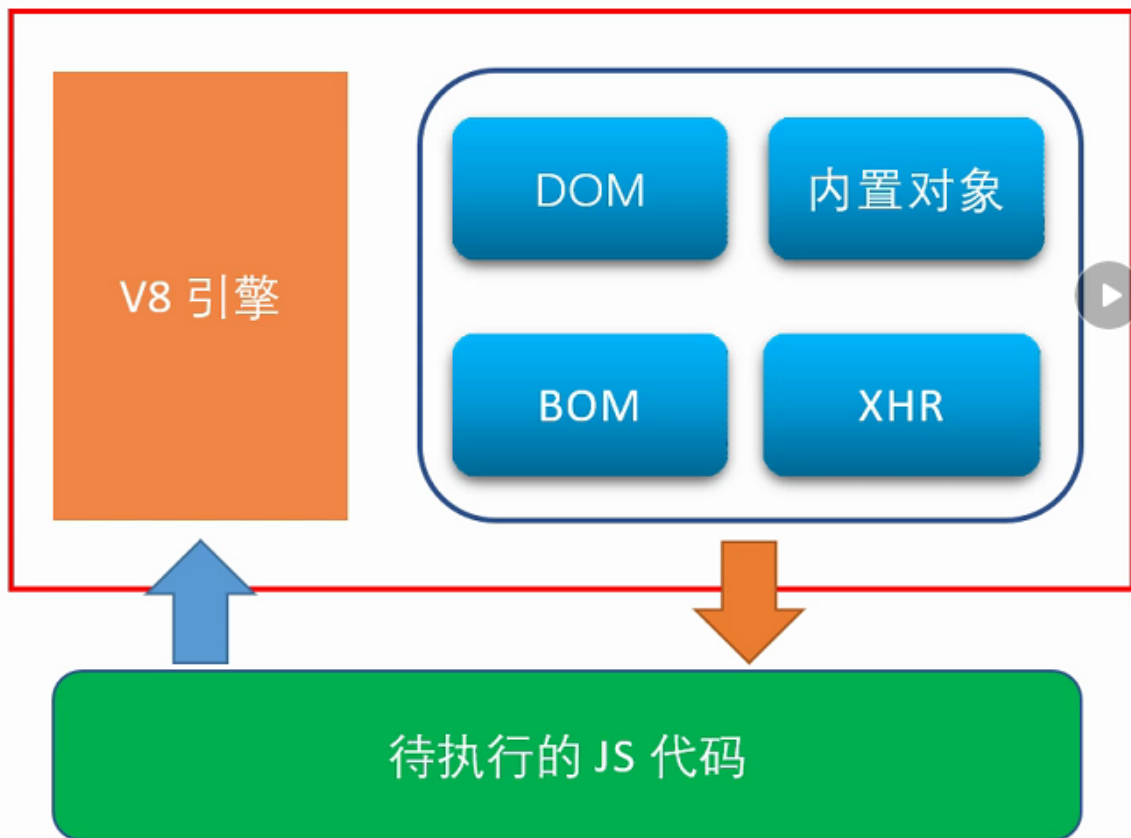
- 网页特效 (监听用户的一些行为让网页作出对应的反馈)
- 表单验证 (针对表单数据的合法性进行判断)
- 数据交互 (获取后台的数据, 渲染到前端)
- 服务端编程 (node.js)



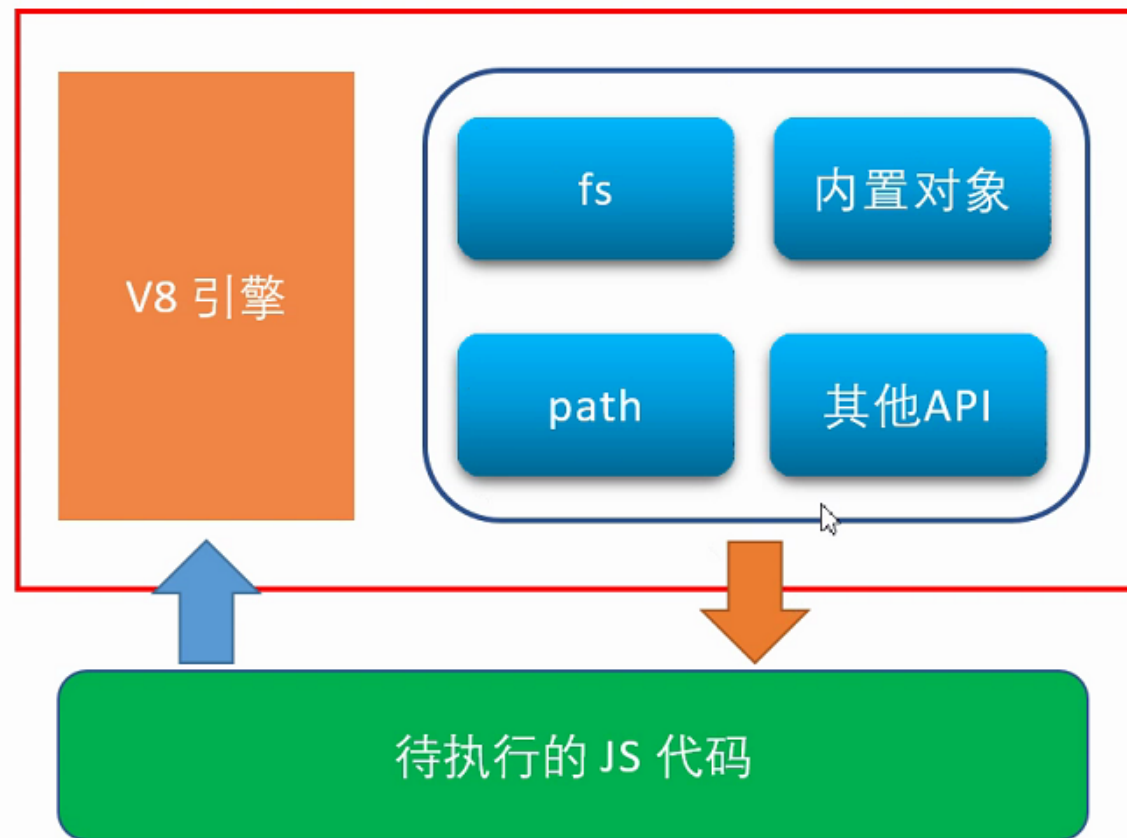
1.1 JavaScript 是什么

目标：初步了解 JavaScript

Chrome 浏览器运行环境



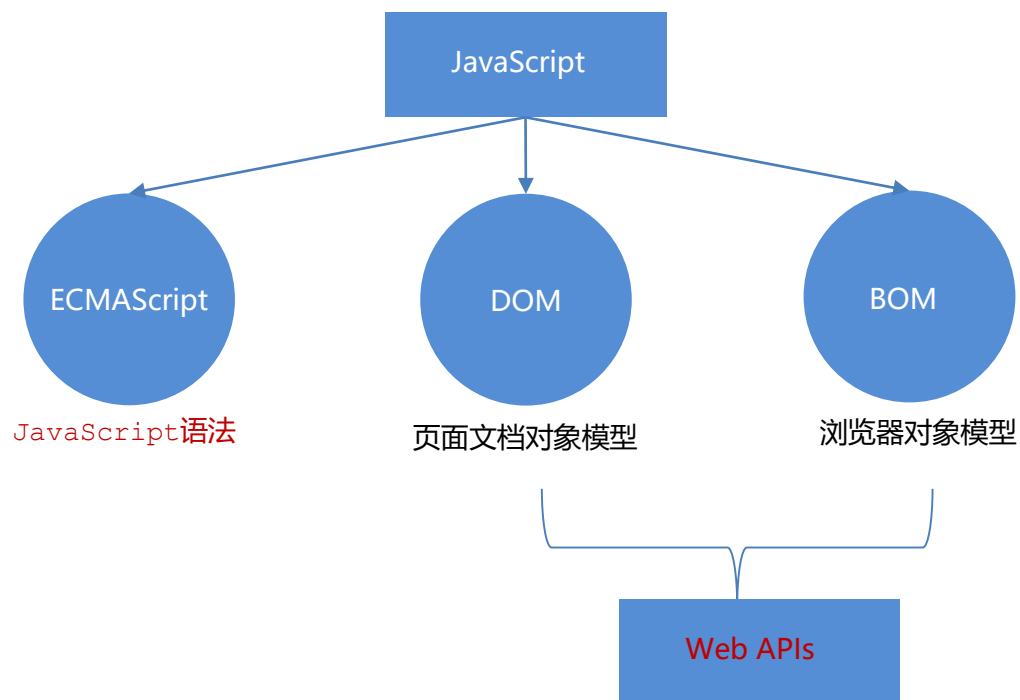
Node.js 运行环境



1.1 JavaScript 是什么

目标：初步了解 JavaScript

JavaScript的组成



JavaScript权威网站: <https://developer.mozilla.org/zh-CN/docs/Web/JavaScript>



总结

1. JavaScript是什么?

- JavaScript 是一种运行在客户端（浏览器）的编程语言

2. JavaScript组成是什么?

- ECMAScript、web APIs (DOM、BOM)



JavaScript 介绍

- JavaScript 是什么
- JavaScript 书写位置
- JavaScript 的注释
- JavaScript 的结束符
- 输入和输出语句

1.2 JavaScript 书写位置

目标：知道如何向页面添加 JavaScript

CSS 书写位置：

- 行内样式表
- 内部样式表
- 外部样式表



JavaScript 书写位置：

- 内嵌 JavaScript
- 内部 JavaScript
- 外部 JavaScript

1. 内部 JavaScript

直接写在html文件里，用script标签包住

规范： script标签写在</body>上面

拓展： alert('你好，js') 页面弹出警告对话框

```
<body>

  <!-- 内联形式：通过 script 标签包裹 JavaScript 代码 -->
  <script>
    alert('嗨，欢迎来传智播学习前端技术! ');
  </script>
</body>
```

1.2 JavaScript 书写位置

目标：知道如何向页面添加 JavaScript

CSS 书写位置：

- 行内样式表
- 内部样式表
- 外部样式表



JavaScript 书写位置：

- 内嵌 JavaScript
- 内部 JavaScript
- 外部 JavaScript

2. 外部 JavaScript

代码写在以.js结尾的文件里

语法：通过script标签，引入到html页面中

注意：script标签中间无需写代码，否则会被忽略！

```
<body>
  <!-- 外联形式：通过 script 的 src 属性引入独立的 .js 文件 -->
  <script src="demo.js"></script>
</body>
```

```
<\poql>
```

1.2 JavaScript 书写位置

目标：知道如何向页面添加 JavaScript

CSS 书写位置：

- 行内样式表
- 内部样式表
- 外部样式表



JavaScript 书写位置：

- 内嵌 JavaScript
- 内部 JavaScript
- 外部 JavaScript

3. 内嵌 JavaScript

代码写在标签内部

语法：

```
<body>  
  <button onclick="alert('逗你玩~~~')">点击我月薪过万</button>  
</body>
```

注意：此处作为了解即可，但是后面vue框架会用这种模式

课堂练习：5 分钟，实现 两种 JavaScript 书写方式，页面弹出：努力，奋斗



总结

1. JavaScript三种书写位置?

- 外部
- 内部
- 内嵌

2. JavaScript三种书写位置注意?

- 开发中最常用外部JavaScript
- 也会用下内部JavaScript
- 内嵌式比较少用，但是要了解一下



JavaScript 介绍

- JavaScript 是什么
- JavaScript 书写位置
- JavaScript 的注释
- JavaScript 的结束符
- 输入和输出语句

1.3 JavaScript 注释

目标：会使用两种JavaScript注释方法

1. 单行注释

- 符号：//
- 作用：//右边这一行的代码会被忽略
- 快捷键：ctrl + /

```
<script>  
    // 这种是单行注释的语法  
    // 一次只能注释一行  
    // 可以重复注释  
</script>
```

2. 块注释

- 符号：/* */
- 作用：在/* 和 */ 之间的所有内容都会被忽略
- 快捷键：shift + alt + A

```
<script>  
    /* 这种的是多行注释的语法 */  
    /*  
        更常见的多行注释是这种写法  
        在些可以任意换行  
        多少行都可以  
    */  
</script>
```



JavaScript 介绍

- JavaScript 是什么
- JavaScript 书写位置
- JavaScript 的注释
- JavaScript 的结束符
- 输入和输出语句

1.4 JavaScript 结束符

目标：了解JavaScript结束符

1. 结束符

- 代表语句结束
- 英文分号；
- 可写可不写（现在不写结束符的程序员越来越多）
- 换行符（回车）会被识别成结束符，所以一个完整的语句，不要手动换行
- 因此在实际开发中有许多人主张书写 JavaScript 代码时省略结束符
- 但为了风格统一，要写结束符就每句都写，要么每句都不写（按照团队要求.）

```
<script>
    alert(1);
    alert(2);
</script>
```



```
<script>
    alert(1)
    alert(2)
</script>
```



JavaScript 介绍

- JavaScript 是什么
- JavaScript 书写位置
- JavaScript 的注释
- JavaScript 的结束符
- 输入和输出语句

1.5 JavaScript 输入输出语句

目标：能写出常见 JavaScript 输入输出语句

输出和输入也可理解为人和计算机的交互，用户通过键盘、鼠标等向计算机输入信息，计算机处理后再展示结果给用户，这便是一次输入和输出的过程。

1. 输出语句：

```
document.write('要输出的内容');
```

- 向body内输出内容
- 如果输出的内容写的是标签，也会被解析成网页元素

```
alert('要输出的内容');
```

- 页面弹出警告对话框

```
console.log('控制台打印')
```

- 控制台输出语句，程序员调试使用

1.5 JavaScript 输入输出语句

目标：能写出常见 JavaScript 输入输出语句

输出和输入也可理解为人和计算机的交互，用户通过键盘、鼠标等向计算机输入信息，计算机处理后再展示结果给用户，这便是一次输入和输出的过程。

2. 输入语句：

```
prompt('请输入您的姓名:');
```

- 显示一个对话框，对话框中包含一条文字信息，用来提示用户输入文字

课堂练习：5 分钟：

浏览器中弹出对话框：你做好准备了吗？

页面中输出：JavaScript 依然悄然来临！

1. JavaScript是什么？

JavaScript是一门编程语言，可以实现很多的网页交互效果。

2. JavaScript 书写位置？

- 行内 JavaScript
- 内部 JavaScript - 写到 `</body>` 标签上方
- 外部 JavaScript - 通过 `src` 引入html页面中，但是 `<script>` 标签不要写内容，否则会被忽略

3. JavaScript 的注释？

- 单行注释 `//`
- 多行注释 `/* */`

4. JavaScript 的结束符？

- 分号； 可以加也可以不加，可以按照团队约定
- 注意换行默认为结束符

5. JavaScript 输入输出语句？

- 输入： `prompt()`
- 输出： `alert()` `document.write()` `console.log()`

学习目标

Learning Objectives

1. JavaScript介绍
2. 变量
3. 数据类型
4. 类型转换
5. 实战案例



变量

- 变量是什么
- 变量基本使用 ☆
- 变量的本质
- 变量命名规则与规范

2.1 变量是什么？

目标：理解变量是计算机存储数据的“容器”

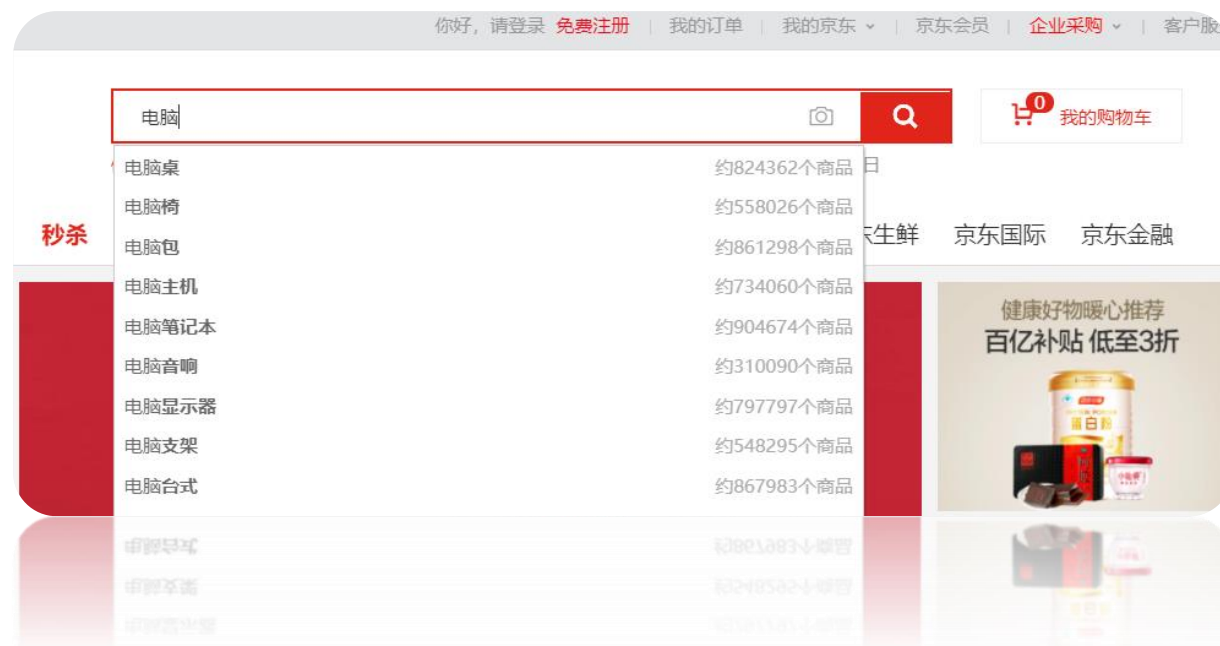
问题1：用户输入的数据我们如何存储起来？

此网页显示

请输入用户名：

确定 取消

答案1：变量。

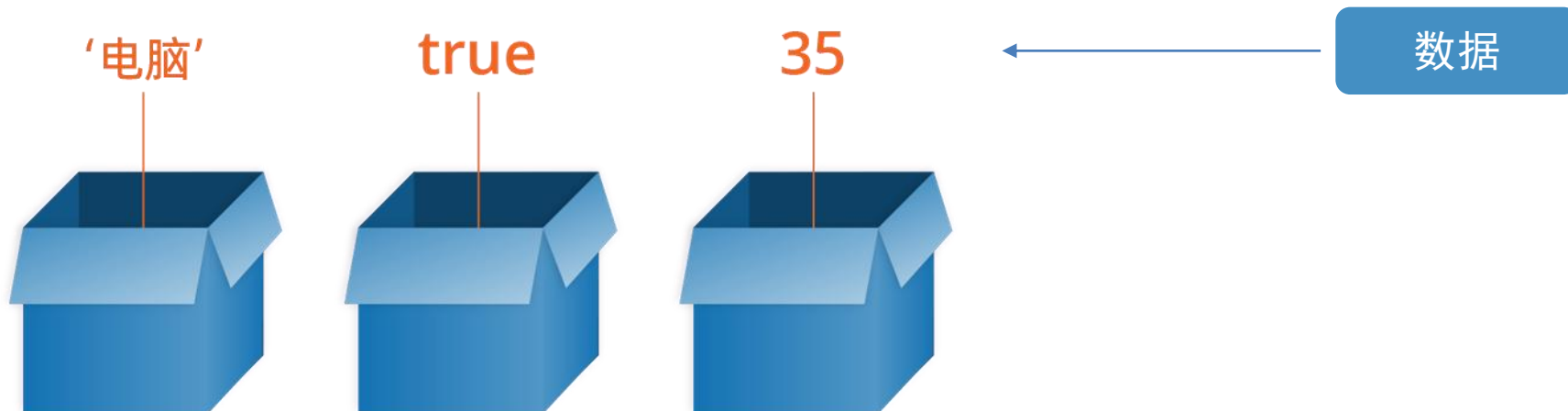


2.1 变量是什么？

目标：理解变量是计算机存储数据的“容器”

1. 变量：

1. 白话：变量就是一个装东西的盒子。
2. 通俗：变量是计算机中用来存储数据的“容器”，它可以让计算机变得有记忆。



3. 注意：变量不是数据本身，它们仅仅是一个用于存储数值的容器。可以理解为是一个个用来装东西的纸箱子。

2.1 变量是什么？

目标：理解变量是计算机存储数据的“容器”

总结：

1. 变量是计算机中用来存储数据的“容器”，简单理解是一个个的盒子。
2. 变量是用来存放数据的。注意变量指的是容器而不是数据。
3. 你还能想到那些生活中的变量？
 - HTML标签
 - 教室
 - 宿舍
 - ...



总结

1. 变量是怎么理解？

- 计算机中用来存储数据的“容器”，简单理解是一个个的盒子。

2. 变量有什么作用呢？

- 用来存放数据的。注意变量指的是容器而不是数据。

3. 你还能想到那些生活中的变量？

- HTML标签
- 教室
- 宿舍
- ...



变量

- 变量是什么
- 变量基本使用 ☆
- 变量的本质
- 变量命名规则与规范

2.2 变量的基本使用

目标：掌握变量的声明方式

1. 声明变量：

要想使用变量，首先需要创建变量（专业说法：声明变量）

语法：

```
let 变量名
```

- 声明变量有两部分构成：声明关键字、变量名（标识）
- let 即关键字 (let: 允许、许可、让、要)，所谓关键字是系统提供的专门用来声明（定义）变量的词语

举例：

```
let age;
```

- age 即变量的名称，也叫标识符

2.2 变量的基本使用

目标：掌握变量的声明方式

2. 变量赋值：

定义了一个变量后，你就能够初始化它（赋值）。在变量名之后跟上一个“=”，然后是数值。

```
let age;  
// 赋值，将 18 这个数据存入了 age 这个“容器”中  
age = 18;  
// 这样 age 的值就成了 18  
document.write(age);  
qocnwεuf·MLTfε(sεε)?
```

注意：是通过变量名来获得变量里面的数据。

2.2 变量的基本使用

目标：掌握变量的声明方式

2. 变量赋值：

也可以声明变量的时候同时给变量初始化。

```
<script>  
    // 声明了一个age变量，同时里面存放了 18 这个数据  
    let age = 18  
</script>  
<\script>
```

2.2 变量的基本使用

目标：掌握变量的声明方式

3. 更新变量：

变量赋值后，还可以通过简单地给它一个不同的值来更新它。

```
// 声明了一个age变量，同时里面存放了 18 这个数据
let age = 18
// 变量里面的数据发生变化更改为 19
age = 19
// 页面输出的结果为 19
document.write(age)
```

```
qocnw6uf*MLTf6(9B6)
```

```
// 声明了一个age变量，同时里面存放了 18 这个数据
let age = 18
// 这里不允许多次声明一个变量，没有理由重新声明变量
let age = 19
// 输出会报错
document.write(age)
```

```
qocnw6uf*MLTf6(9B6)
```

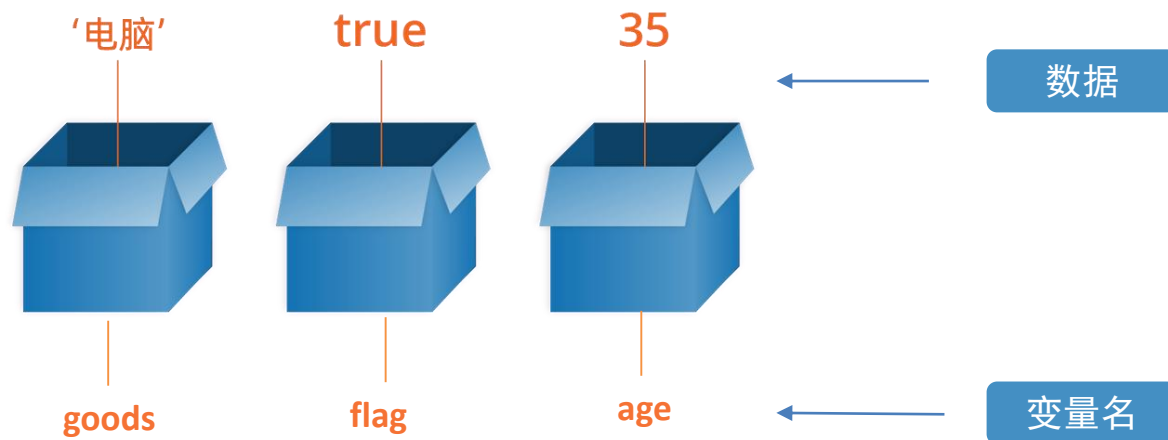
注意：let 不允许多次声明一个变量。

2.2 变量的基本使用

目标：掌握变量的声明方式

总结：

1. 变量就是一个容器，用来存放数据的。
2. 变量通过 let 来声明 通过 = 实现赋值。
3. 变量通过变量名来获得里面的数据。
4. 变量不允许多次声明。
5. 最早变量声明使用 var 关键字，后期我们讲解 var 和 let 区别



课堂练习：5 分钟：

浏览器中弹出对话框：请输入姓名

页面中输出：刚才输入的姓名





变量

- 变量是什么
- 变量基本使用 ☆
- 变量的本质
- 变量命名规则与规范

2.3 变量的本质

目标：能够说出变量的本质是什么

内存：计算机中存储数据的地方，相当于一个空间

变量：是程序在内存中申请的一块用来存放数据的小空间

```
let age = 18
```

```
let goods = '电脑'
```

18

电脑

← 内存



变量

- 变量是什么
- 变量基本使用 ☆
- 变量的本质
- 变量命名规则与规范

2.4 变量命名规则与规范

目标：能写出符合规范的变量名

规则：必须遵守，不遵守报错

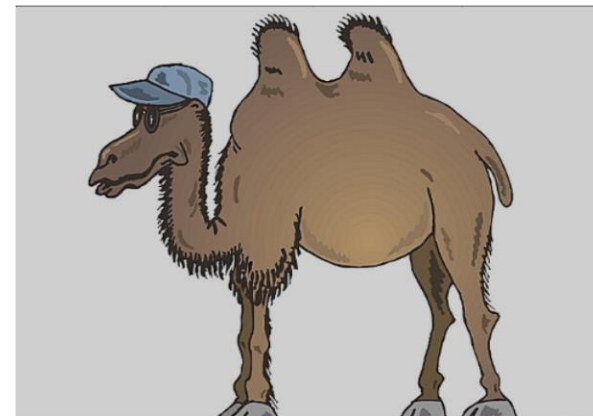
规范：建议，不遵守不会报错，但不符合业内通识

1. 规则：

- 不能用关键字
 - ✓ 关键字：有特殊含义的字符，JavaScript 内置的一些英语词汇。例如：let、var、if、for等
- 只能用下划线、字母、数字、\$组成，且数字不能开头
- 字母严格区分大小写，如 Age 和 age 是不同的变量

2. 规范：

- 起名要有意义
- 遵守小驼峰命名法
 - ✓ 第一个单词首字母小写，后面每个单词首字母大写。例：userName



2.4 变量命名规则与规范

目标：能写出符合规范的变量名

以下哪些是合法的变量名？

变量名	是否报错	是否符合规范
21age		
_age		
user-name		
username		
userName		
let		
na@me		
\$age		

练习

变量练习-输出用户信息

需求：让用户输入自己的名字、年龄、性别，再输出到网页

分析：

- ①：弹出 **输入** 框（prompt）： 请输入您的姓名： 用变量保存起来。
- ②：弹出输入框（prompt）： 请输入您的年龄： 用变量保存起来。
- ③：弹出输入框（prompt）： 请输入您的性别： 用变量保存起来。
- ④：页面分别 **输出** (document.write) 刚才的 3 个变量。



总结

1. 为什么需要变量？

- 因为我们一些数据需要保存，所以需要变量

2. 变量是什么？

- 变量就是一个容器，用来存放数据的。方便我们以后使用里面的数据

3. 变量的本质是什么？

- 变量是内存里的一块空间，用来存储数据。

4. 变量怎么使用的？

- 我们使用变量的时候，一定要声明变量，然后赋值
- 声明变量本质是去内存申请空间。

总结

4.1. 变量的声明

- let 变量名
- 也可以把let换成var, 建议用let

4.2. 变量的赋值

- 变量名 = 数据

4.3. 变量声明同时赋值

- let 变量名 = 数据

```
<script>  
    // 声明了一个age变量, 同时里面存放了 18 这个数据  
    let age = 18  
</script>  
<\script>
```

使用 let 时的注意事项:

- 允许声明和赋值同时进行
- 不允许重复声明
- JavaScript 中内置的一些关键字不能被当做变量名
- 允许同时声明多个变量并赋值，同时，输出也可以输出多个变量

```
// 声明多个变量  
let goods = '电脑', age = 18  
// 输出多个变量  
document.write(goods, age)
```

```
document.write('电脑', 18)
```

学习目标

Learning Objectives

1. JavaScript介绍
2. 变量
3. 数据类型
4. 类型转换
5. 实战案例



数据类型

- 数据类型 ☆
- 检测数据类型

3.1 数据类型

目标：能说出JS中四种数据类型

计算机世界中的万事万物都是数据。

计算机程序可以处理大量的数据，为什么要给数据分类？

- 1. 更加充分和高效的利用内存
- 2. 也更加方便程序员的使用数据

数据分成了不同的类型：



3.1 数据类型

目标：能说出JS中四种数据类型

JS 数据类型整体分为两大类：

- 基本数据类型
- 引用数据类型



3.1 数据类型

目标：能说出JS中四种数据类型

先学习基本数据类型：

数据类型	作用	表示形式	例
string（字符串）	表示文字内容	单引号或者双引号包起来	'传智'"黑马"
number（数字）	表示数字内容	直接写数字	10,11,12,10.32
boolean（布尔）	表示对立的两种状态：真和假、对和错、开和关等	true和false	true, false
undefined（未定义）	表示未定义	undefined	undefined

3.1 数据类型 - 数字类型 (number)

即我们数学中学习到的数字，可以是整数、小数、正数、负数。

```
let score = 100; // 正整数  
let price = 12.345; // 小数  
let temperature = -40; // 负数
```

JavaScript 中的数值类型与数学中的数字是一样的，分为正数、负数、小数等。

3.1 数据类型 - 字符串类型 (string)

通过单引号 (') 、双引号 (") 或反引号 (`) 包裹的数据都叫字符串，单引号和双引号没有本质上的区别，推荐使用单引号。

```
let user_name = '小明'; // 使用单引号
let gender = "男"; // 使用双引号
let str = '123'; // 看上去是数字，但是用引号包裹了就成了字符串了
let str1 = ''; // 这种情况叫空字符串
```

```
let str2 = '\n'; // 这种写法叫转义字符串
```

注意事项：

1. 无论单引号或是双引号必须成对使用
2. 单引号/双引号可以互相嵌套，但是不以自己嵌套自己（口诀：外双内单，或者外单内双）
3. 必要时可以使用转义符 \，输出单引号或双引号

3.1 数据类型 - 字符串类型 (string)

字符串拼接:

```
document.write('我叫' + '刘德华')    // 我叫刘德华
```

```
let uname = '刘德华'
```

```
let song = '忘情水'
```

```
document.write(uname + song)    // 刘德华忘情水
```

```
document.write(uname + song)    // 刘德华忘情水
```

1. 作用

- 拼接字符串和变量
- 在没有它之前，要拼接变量比较麻烦

```
document.write('大家好，我叫' + name + '，今年' + age + '岁')
```

2. 符号

- ``
- 在英文输入模式下按键盘的tab键上方那个键（1左边那个键）
- 内容拼接变量时，用 `${}` 包住变量

```
document.write(`大家好，我叫${name}，今年${age}岁`)
```

3.1 数据类型 - 布尔类型 (boolean)

表示肯定或否定时在计算机中对应的是布尔类型数据。

它有两个固定的值 true 和 false，表示肯定的数据用 true（真），表示否定的数据用 false（假）。

```
// pink老师帅不帅? 回答 是 或 否  
let isCool = true; // 是的, 摔死了!  
isCool = false; // 不, 套马杆的汉子!
```

3.1 数据类型 - 未定义类型 (undefined)

未定义是比较特殊的类型，只有一个值 undefined。

什么情况出现未定义类型？

只声明变量，不赋值的情况下，变量的默认值为 undefined，一般很少【直接】为某个变量赋值为 undefined。

```
let age    // 声明变量但是未赋值
document.write(age) // 输出 undefined
```

情况	说明	结果
let age ; console.log (age)	只声明 不赋值	undefined
console.log(age)	不声明 不赋值 直接使用	报错
age = 10; console.log (age)	不声明 只赋值	10（不提倡）

3.1 数据类型

总结：

计算机世界中的万事万物都是数据。

计算机程序可以处理大量的数据，为了方便数据的管理（用途不同），将数据分成了不同的类型：

数据类型	作用	表示形式	例
string（字符串）	表示文字内容	单引号或者双引号包起来	'传智' "黑马"
number（数字）	表示数字内容	直接写数字	10,11,12,10.32
boolean（布尔）	表示对立的两种状态：真和假、对和错、开和关等	true和false	true, false
undefined（未定义）	表示未定义	undefined	undefined



数据类型

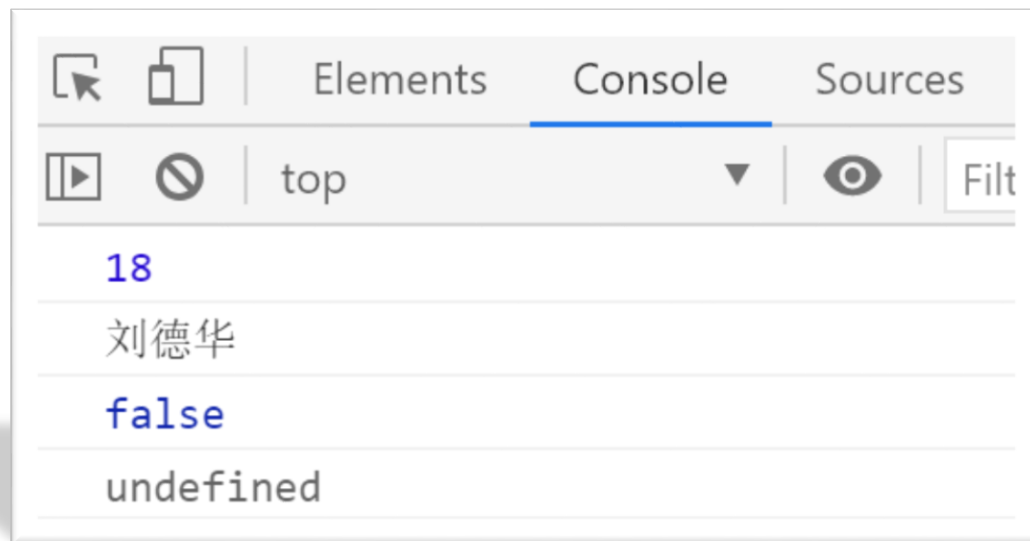
- 数据类型 ☆
- 检测数据类型

3.2 控制台输出语句和检测数据类型

1. 控制台输出语句:

```
let age = 18
let uname = '刘德华'
let flag = false
let buy
console.log(age)
console.log(uname)
console.log(flag)
console.log(buy)
```

```
console.log(18)
console.log('刘德华')
```



- 控制台语句经常用于测试结果来使用。
- 可以看出数字型和布尔型颜色为蓝色，字符串和undefined颜色为灰色

3.2 控制台输出语句和检测数据类型

2. 通过 typeof 关键字检测数据类型

```
let age = 18
let uname = '刘德华'
let flag = false
let buy
console.log(typeof age)    // number
console.log(typeof uname) // string
console.log(typeof flag)  // boolean
console.log(typeof buy)   // undefined
```

```
console.log(typeof 18)    // number
console.log(typeof '刘德华') // string
```

课堂练习

交换2个变量的值

要求：

1. 有2个变量： num1 里面放的是 10， num2 里面放的是20
2. 最后变为 num1 里面放的是 20 ， num2 里面放的是 10

课堂练习

交换2个变量的值

思路：

1. 核心思路：使用一个 临时变量 用来做中间存储

num1

10

num2

20



临时变量

课堂练习

交换2个变量的值

思路：

1. 核心思路：使用一个 临时变量 用来做中间存储

num1

20

num2

10

临时变量

课堂练习

交换2个变量的值

步骤：

1. 声明一个临时变量 temp
2. 把num1的值赋值给 temp
3. 把num2的值赋值给num1
4. 把temp的值给num2
5. 没了~~~~

五. 类型转换

目标：理解弱类型语言的特征，掌握数据类型转换的方法

学习路径

1. 为什么要类型转换
2. 隐式转换
3. 显式转换

5.1 为什么需要类型转换

JavaScript是弱数据类型：JavaScript也不知道变量到底属于那种数据类型，只有赋值了才清楚。

坑：使用表单、prompt 获取过来的数据默认是字符串类型的，此时就不能直接简单的进行加法运算。

```
console.log('10000' + '2000') // 输出结果 100002000
```

此时需要转换变量的数据类型。

通俗来说，就是把一种数据类型的变量转换成我们需要的数据类型。

学习目标

Learning Objectives

1. JavaScript介绍
2. 变量
3. 数据类型
4. 类型转换
5. 实战案例



类型转换

- 为什么要类型转换
- 隐式转换
- 显式转换

4.1 为什么需要类型转换

prompt() 虽然用户输入的是数字，但是得到的是字符串，同理比如input文本框得到的也是如此

- 而字符串和数字相加会变成相连
- 此时想做加法运算出现： `'10' + '10' = '1010'` 而不是 20
- 所以，需要讲类型转换为我们想要的类型



类型转换

- 为什么要类型转换
- 隐式转换
- 显式转换

5.2 隐式转换

某些运算符被执行时，系统内部自动将数据类型进行转换，这种转换称为隐式转换。

规则：

- + 号两边只要有一个是字符串，都会把另外一个转成字符串
- 除了+以外的算术运算符 比如 - * / 等都会把数据转成数字类型

缺点：

- 转换类型不明确，靠经验才能总结

小技巧：

- +号作为正号解析可以转换成Number

```
<script>
  console.log(11 + 11)
  console.log('11' + 11)
  console.log(11 - 11)
  console.log('11' - 11)
  console.log(1 * 1)
  console.log('1' * 1)
  console.log(typeof '123')
  console.log(typeof +'123')
  console.log(+'11' + 11)
</script>
```

22

1111

0

0

1

1

string

number

22

编写程序时过度依靠系统内部的隐式转换是不严禁的，因为隐式转换规律并不清晰，大多是靠经验总结的规律。

为了避免因隐式转换带来的问题，通常根逻辑需要对数据进行显示转换。

概念：

自己写代码告诉系统该转成什么类型

转换为数字型

➤ **Number(数据)**

- ✓ 转成数字类型
- ✓ 如果字符串内容里有非数字，转换失败时结果为 NaN（Not a Number）即不是一个数字
- ✓ NaN也是number类型的数据，代表非数字

➤ **parseInt(数据)**

➤ 只保留整数

➤ **parseFloat(数据)**

➤ 可以保留小数

编写程序时过度依靠系统内部的隐式转换是不严禁的，因为隐式转换规律并不清晰，大多是靠经验总结的规律。

为了避免因隐式转换带来的问题，通常根逻辑需要对数据进行显示转换。

概念：

自己写代码告诉系统该转成什么类型

转换为字符型：

- `String(数据)`
- `变量.toString()`

课堂单独练习

输入2个数，计算两者的和，打印到页面中



总结

1. 类型转换的概念

一种数据类型转成别的类型，JavaScript是弱数据类型，很多情况计算的时候，需要转换数据类型

2. 隐式转换

- 系统自动做转换

3. 显式转换

自己写代码告诉系统转成什么类型

- Number

字符串内容里有非数字得到NaN

- String

学习目标

Learning Objectives

1. JavaScript介绍
2. 变量
3. 数据类型
4. 类型转换
5. 实战案例

案例

用户订单信息案例

需求：用户输入商品价格和商品数量，以及收货地址，可以自动打印订单信息

TLIAS管理后台 面试列表 | ShowM... 前端三人组 课程汇总 传智门户 顺义校区听课记录...

此网页显示

请输入商品价格:

确定 取消

案例

用户订单信息案例

需求：用户输入商品价格和商品数量，以及收货地址，可以自动打印订单信息

分析：

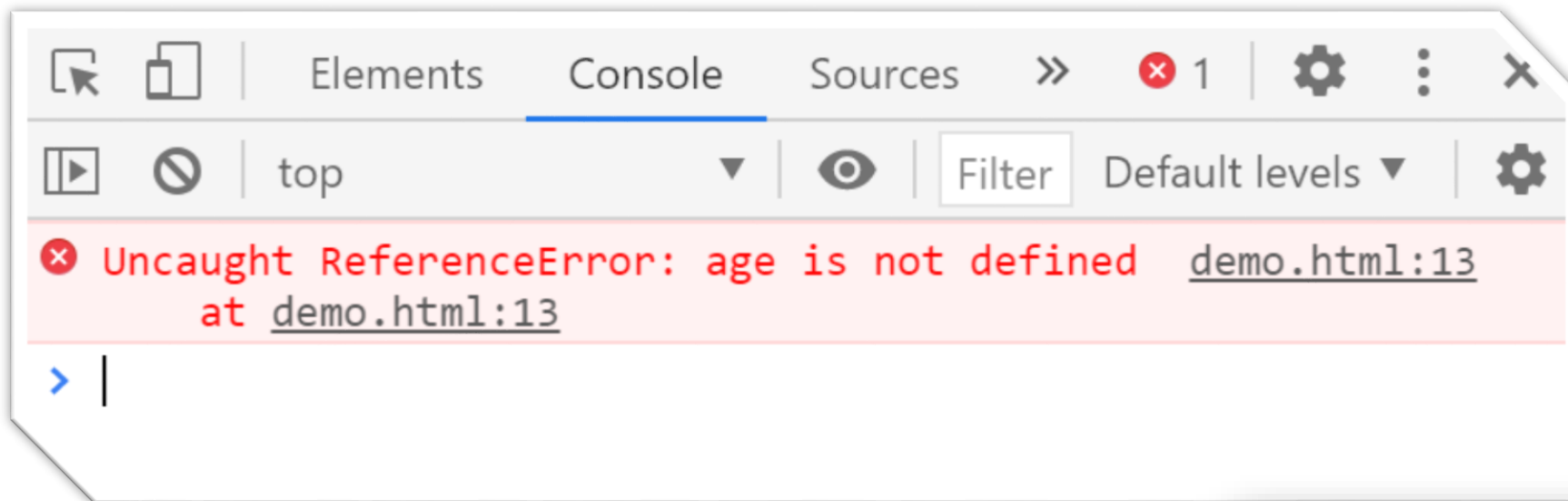
- ①：需要输入3个数据，所以需要3个变量来存储 price num address
- ②：需要计算总的价格 total
- ③：页面打印生成表格，里面填充数据即可
- ④：记得最好使用模板字符串

订单付款确认页面

商品名称	商品价格	商品数量	总价	收货地址
小米手机青春PLUS	1999元	3	5997元	北京顺义黑马程序员

六. 常见错误

1. 下面可能出现的原因是什么？



分析：

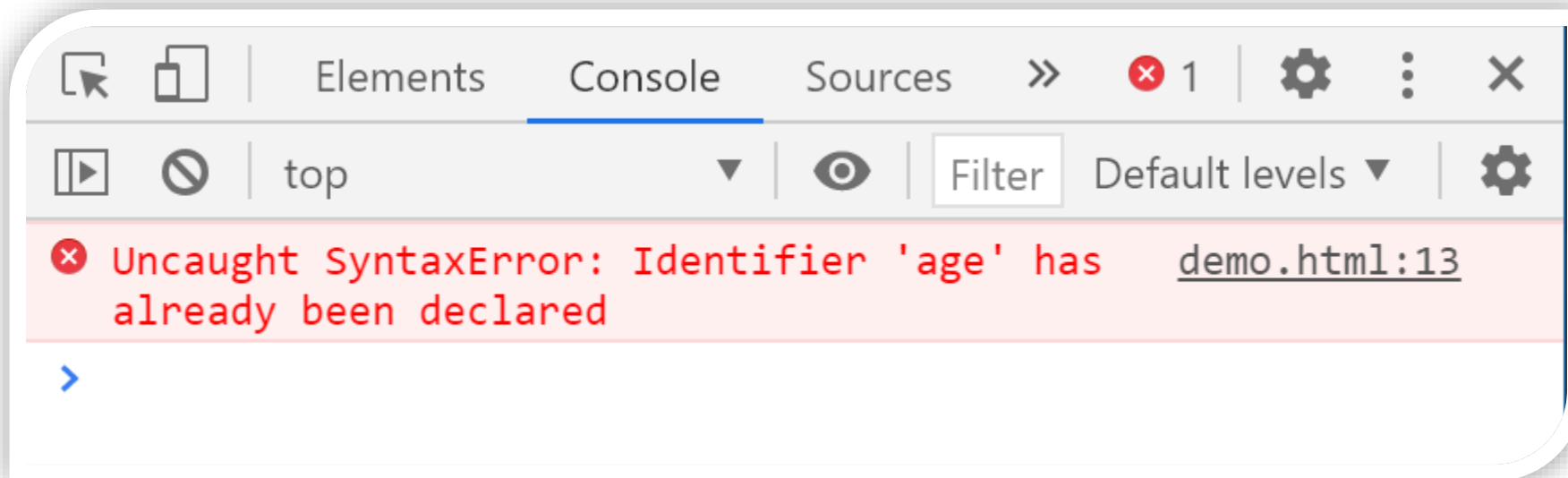
1. 提示 age 变量没有定义过
2. 很可能 age 变量没有声明和赋值
3. 或者我们输出变量名和声明的变量不一致引起的。

```
10 <body>
11   <script>
12     let aeg = 10
13     console.log(age);
14   </script>
15 </body>
```

Diagram illustrating the error: The variable 'age' is used in the console.log statement (line 13), but the variable declared is 'aeg' (line 12). The arrows point to the mismatch between the variable name in the log statement and the variable name in the declaration.

六. 常见错误

2. 下面可能出现的原因是什么？



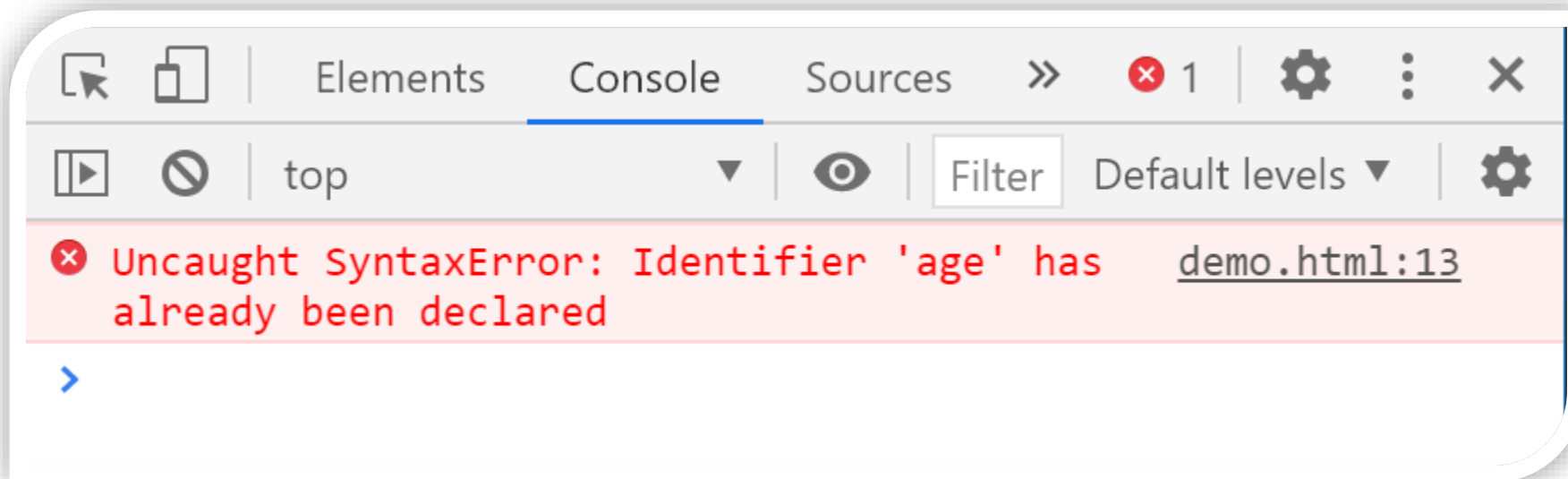
分析：

1. 提示 “age” 已经声明
2. 很大概率是因为使用let 重复声明了一个变量。
3. 注意let 变量不允许多次声明同一个变量

```
10 <body>
11   <script>
12     let age = 10
13     let age = 20
14   </script>
15 </body>
```

六. 常见错误

2. 下面可能出现的原因是什么？



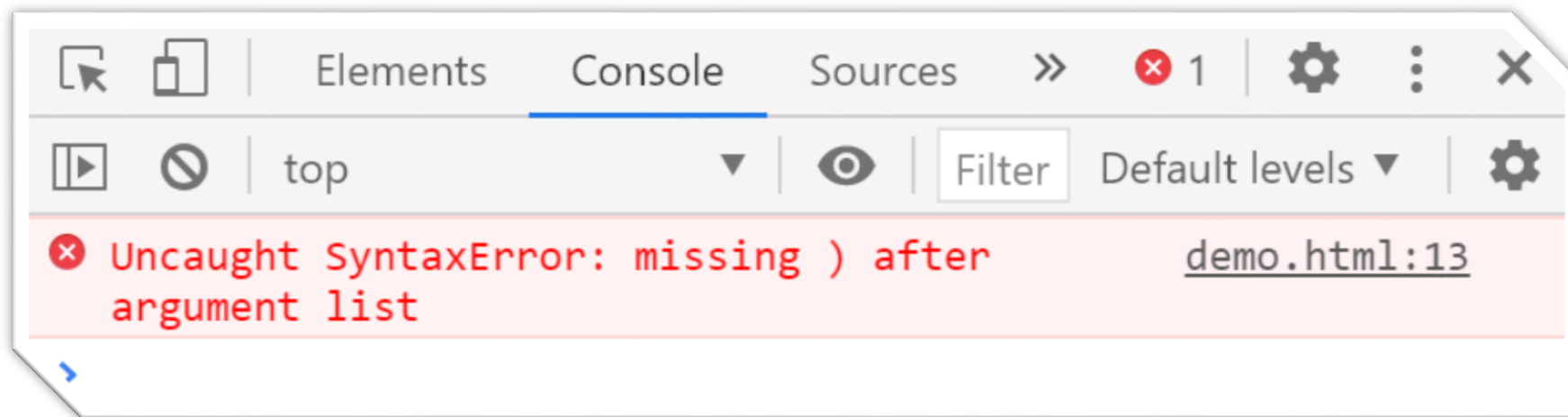
分析：

1. 提示 “age” 已经声明
2. 很大概率是因为使用let 重复声明了一个变量。
3. 注意let 变量不允许多次声明同一个变量

```
10 <body>
11   <script>
12     let age = 10
13     let age = 20
14   </script>
15 </body>
```

六. 常见错误

3. 下面可能出现的原因是什么?



分析:

1. 提示 参数少了)
2. 很大概率是小括号不匹配, 掉了一半

```
10 <body>
11   <script>
12     let age = 18
13     document.write(`我今年已经${age}岁啦`
14   </script>
15 </body>
```

六. 常见错误

4. 下面可能出现什么问题？怎么办？

```
0  <body>
1      <script>
2          let num1 = prompt('请输入第一个数')
3          let num2 = prompt('请输入第二个数')
4          alert(num1 + num2)
5      </script>
</body>
```

分析：

1. 出现字符相加的问题
2. prompt 如果出现相加 记得要转为数字型

七. 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. 梳理完毕再次写2遍今日综合案例（订单信息案例案例）记得先写伪代码思路，然后里面填写代码。
3. 晚上8点是做测试题（测试时间最多30分钟）
4. 做完考题的同学开始：独立书写今日作业。



images



JavaScript基础...



作业答案参考.html

5. 明天自习：上午相互提问今日单词和案例思路
6. 明天自习：下午预习第二天笔记
7. 每日一句鼓励自己的话：

如果你的面前有阴影，别怕，那是因为你的背后有阳光

8. 战歌响起来~~~ 明月天涯.mp3



传智教育旗下高端IT教育品牌