

# JS基础与实践-上篇

《WEB实训课程》



# 目录

- ◆ JS简介
- ◆ JS基础用法
- ◆ JS语句（教学生如何自学，不展开）
- ◆ JS语法（教学生如何自学，不展开）
- ◆ 事件
- ◆ DOM操作
- ◆ jQuery介绍
- ◆ jQuery简单实例
- ◆ 作业：弹幕墙



# JS简介

- ◆ JS是一个非常混乱的语言，直到现在才逐步走向规范
- ◆ 传统的JS是为了解决HTML的交互问题
- ◆ 现在的JS是为了解决构建复杂应用的问题
- ◆ 目前少有的横跨三界的技术：服务端/WEB端/APP端。
- ◆ 面向过程、面向对象、范式编程三种编程思想并存



# JS基础用法

## 示例说明

1. 增加<span>, id设置为demo

2. 增加<script>标签

3. 在<script>中写脚本, 其中

3.1. document.getElementById用于获取指定ID的元素

3.2. el.innerHTML = ...用于设置内容

```
train2 > <> js1.html > html > body > script
9   </head>
10
11  <body>
12    <h1>基础示例</h1>
13    <span id="demo">测试内容</span>
14    <script>
15      var el = document.getElementById('demo');
16      el.innerHTML="Hello JavaScript";
17    </script>
18  </body>
19
20  </html>
```

## 基础示例

Hello JavaScript



# JS基础用法

## 示例说明

```
<> js2.html > html > head > style > body
<body>
  
  <script>
    function changeImage() {
      var el = document.getElementById("myimage");
      if (el.src.match('bulbon')) {
        el.src = 'https://www.runoob.com/images/
        pic_bulboff.gif';
      } else {
        el.src = 'https://www.runoob.com/images/
        pic_bulbon.gif';
      }
    }
  </script>
</body>
```



1. 创建一个id为myimage的<img>
2. 增加<script>标签
3. 在<script>增加changeImage函数，根据匹配情况替换图片
4. 在<img>增加onclick="changeImage();"表示点击时执行函数



# JS基础用法

- ◆ 直接在<script></script>插入JS语句；
- ◆ 或者通过src属性引入外部文件，比如：  
`<script src="jquery.js"></script>`

JS教程： <https://www.runoob.com/js/js-tutorial.html>



# JS语句

- ◆ JS语句，讲白了是指三大基础控制结构（顺序、分支、循环）怎么写。比如JS的分支是`if (...) {}`,但Bash的分支是`if [ ... ] ; then...endif`。
- ◆ 顺序：参考菜鸟教程的JS语句
- ◆ 分支：参考菜鸟教程的`if...else if...else`, `switch`
- ◆ 循环：参考菜鸟教程的`for`, `while`以及中断`break`和`continue`



# JS语法

- ◆ 数据类型：数值、字符串、布尔值、数组、对象
- ◆ 运算符：赋值运算、算术运算、字符运算、比较运算、逻辑运算、类型运算、位运算
- ◆ 变量声明、定义与赋值
- ◆ 函数定义与调用
- ◆ 对象基础
- ◆ 注释

重点：字符串的基础、拼接、各种处理函数、正则比较



# 事件

- ◆ 事件是发生在HTML元素上的“事情”，JS可以捕捉这些事件并做对应处理。可以说，事件是交互的基础。
- ◆ 事件既可以由浏览器自己产生，也可以由用户产生。

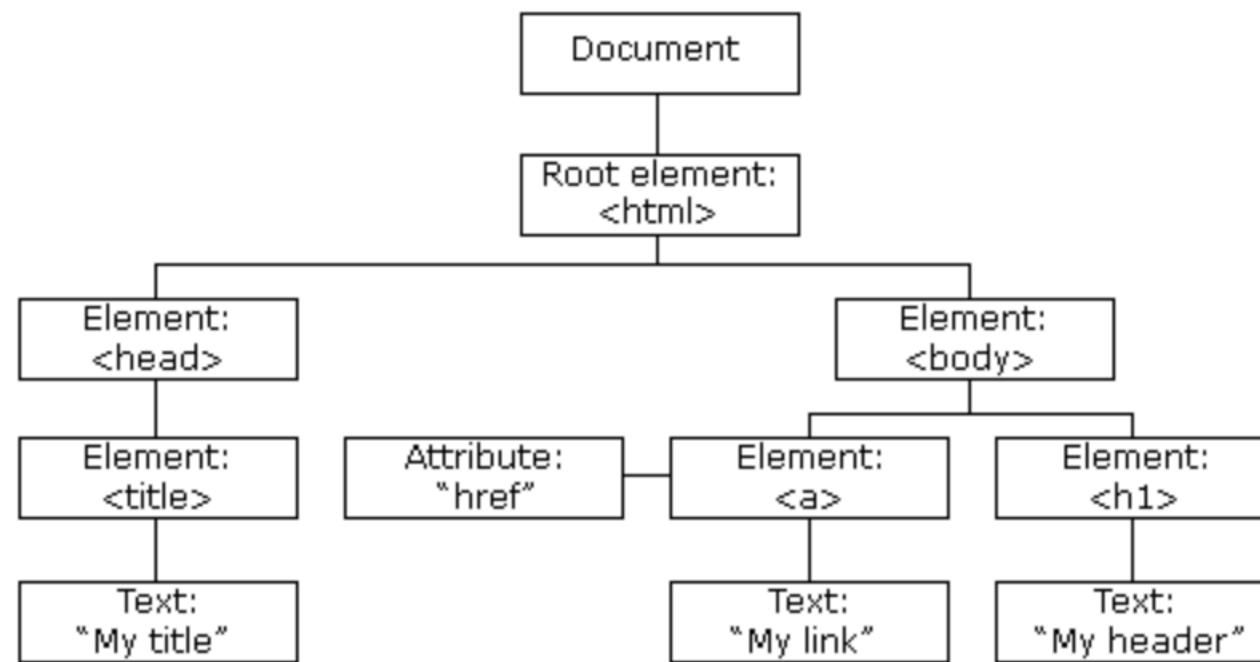


# 事件~常见事件

事件	描述
onchange	HTML 元素已被改变
onclick	用户点击了 HTML 元素 <b>重点</b>
onmouseover	用户把鼠标移动到 HTML 元素上
onmouseout	用户把鼠标移开 HTML 元素
onkeydown	用户按下键盘按键
onload	浏览器已经完成页面加载



# DOM操作



- DOM规定了JS如何获取、更改、添加和删除HTML元素，因此JS对HTML元素的操作就称为DOM操作
- DOM（文档对象模型）将HTML看成一颗如上图所示的DOM树，于是对HTML元素的修改就是选择DOM树上的某个节点，并对其操作操作。
- 通过ID、名称、类名查找HTML元素



# DOM操作~主要操作

- ◆ 修改元素内容和属性(`el.innerHTML`, `el.src`)
- ◆ CSS处理(`el.style.fontSize = '14px'`)
- ◆ 处理事件(`el.onclick = function () {}`)



# DOM操作-计数器示例

```
train2 > <> js3.html > html > head > style > #counter
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width,
6          initial-scale=1.0">
7      <meta http-equiv="X-UA-Compatible" content="ie=edge">
8      <title>JS example3</title>
9      <style>
10         #counter {
11             display:inline-block;
12             padding:8px;
13             border:1px solid black;
14         }
15     </style>
16 </head>
17 <body>
18     <h3>计数器</h3>
19     <button onclick="minus()">-</button>
20     <span id="counter">0</span>
21     <button onclick="plus()">+</button>
22     <script>
23         function minus() {
24             var el = document.getElementById('counter');
25             if (el.innerHTML > 0) {
26                 el.innerHTML--;
27             }
28         }
29         function plus() {
30             var el = document.getElementById('counter');
31             el.innerHTML++;
32         }
33     </script>
34 </body>
</html>
```

计数器

- 0 +



# jQuery介绍

- ◆ jQuery不是一个新的语言，而是JS的一个库，目标是简化HTML DOM操作、事件处理、动画和Ajax。
- ◆ 标准的JavaScript增加了许多支持，以替代jQuery，但写出的代码仍然非常难用，如果有较多DOM操作时，仍然应该使用jQuery。
- ◆ jQuery是传统WEB开发影响最大的一个库，推动WEB的发展功不可没，因此有大量丰富的jQuery配套插件。但是现代WEB开发已经不是直接对DOM操作了，因此现代WEB开发不会引入jQuery，我们对jQuery的理解，也仅需要了解它在DOM操作和事件处理上的高效，其他的已经没必要了解。



# jQuery介绍-用法

- ◆ 引入方法:

```
<script src="https://cdn.bootcss.com/  
jquery/3.4.1/jquery.min.js"></script>
```

- ◆

```
$(document).ready(function(){  
  
    // 开始写 jQuery 代码...  
  
});
```

jQuery教程: <https://www.runoob.com/jquery/jquery-tutorial.html>



# jQuery介绍~主要操作

- ◆ jQuery选择器(使用CSS选择器语法)
- ◆ jQuery事件
- ◆ jQuery修改HTML内容、输入框的值
- ◆ jQuery添加/删除HTML元素
- ◆ jQuery修改样式



# jQuery介绍-简单实例

- ◆ 原来的导航栏例子：点击项应该变成active样式，其他的不变
- ◆ 原咖啡实例：侧边栏，通过JS和position:fixed实现与position:sticky一样的效果



# 商品添加和删除实例

点击添加时

## 商品列表添加和删除

添加了新商品:葡萄

- 苹果
- 梨
- 葡萄

删除所有商品

名称: 葡萄

添加

点击梨，删除时

## 商品列表添加和删除

删除了商品:梨

- 苹果
- 葡萄

删除所有商品

名称: 葡萄

添加

提示:

1.val()方法获取输入值

2.console.log辅助调试

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>商品列表添加和删除</title>
  <style>
    .danger {
      color: red;
    }
  </style>
</head>

<body>
  <h2>商品列表添加和删除</h2>
  <div id="message"></div>
  <ul id="goods">
    <li>苹果</li>
  </ul>
  <div>
    <button id="delAllBtn">删除所有商品</button>
  </div>
  <div>
    <span>名称:</span>
    <input type="text" id="name">
    <button id="add">添加</button>
  </div>

  <script src="https://cdn.bootcss.com/jquery/3.4.1/jquery.min.js"></script>
  <script>
    $(document).ready(function () {
      $('#add').click(function () {
        console.log('value:', $('#name').val());
        $('#goods').append('<li>' + $('#name').val() + '</li>');
        $('#message').html('添加了新商品:' + $('#name').val());
        $('#message').removeClass('danger');
        // 会发现新增的li无法删除，需要重新给li增加事件
        $('#goods li').click(function () {
          $('#message').html('删除了商品:' + $(this).text());
          $('#message').addClass('danger');
          $(this).remove();
        });
      });
      $('#goods li').click(function () {
        $('#message').html('删除了商品:' + $(this).text());
        $('#message').addClass('danger');
        $(this).remove();
      });
      $('#delAllBtn').click(function () {
        $('#goods li').remove();
      })
    });
  </script>
</body>

</html>
```



# 作业：弹幕墙

A word cloud visualization of the phrase "小哥哥加油" (Xiao Ge Ge Jia You). The text is repeated multiple times in various colors (blue, green, yellow, orange, red, purple, pink) and sizes, arranged in a circular pattern. The largest instances of the text are in the center, while smaller instances are scattered towards the edges. The colors are vibrant and varied, creating a dynamic and energetic visual effect. The overall shape is roughly circular, with the text elements filling the space.

小哥哥加油

发射

清屏