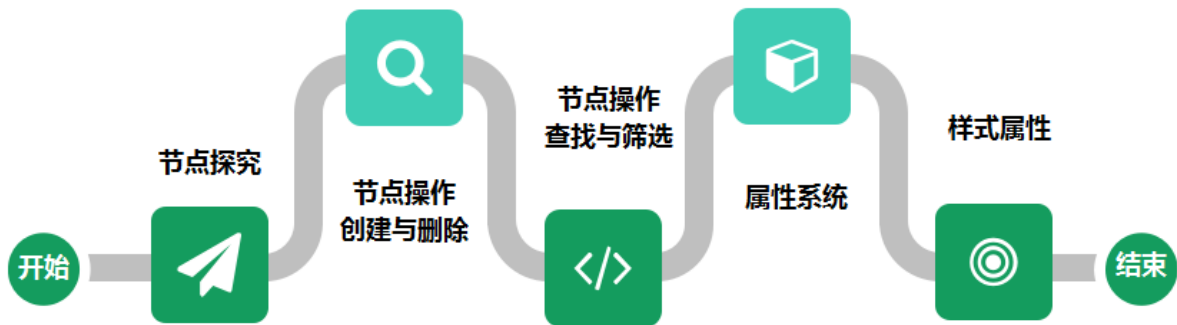


## 5.0 JS DOM 操作

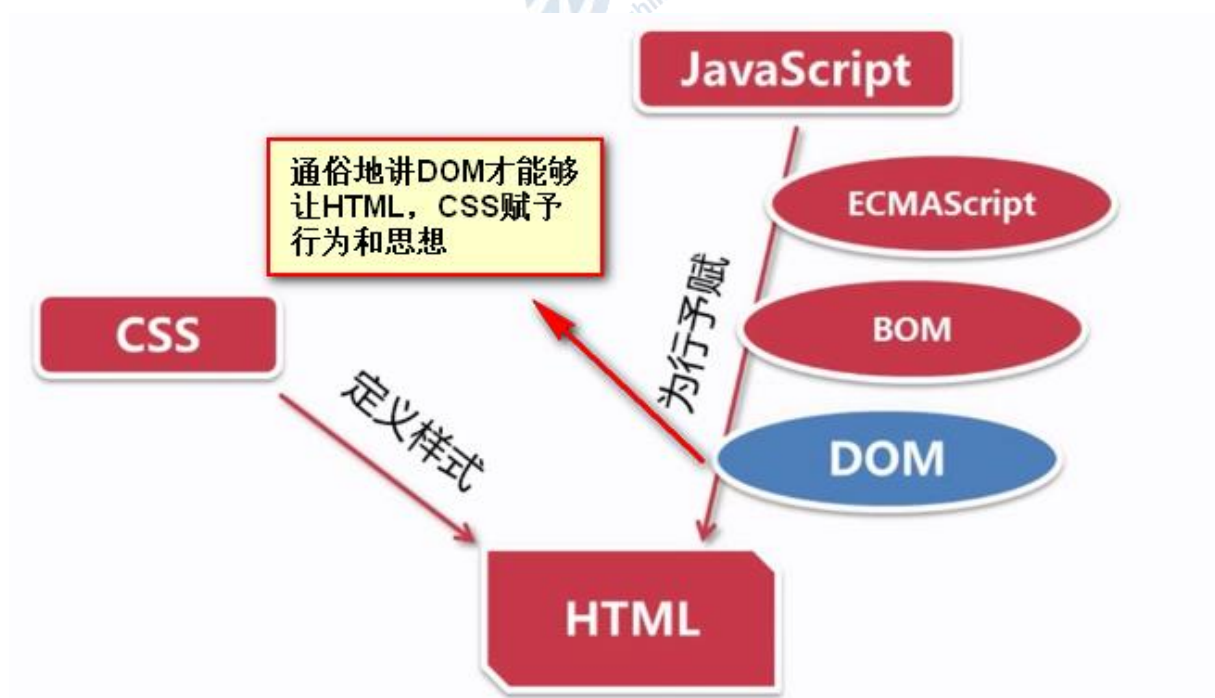
### 5.1 DOM 的学习路线



### 5.2 DOM 的学习方式

- 案例驱动学习
- 通过各种案例给大家介绍 DOM 技术的学习
- 学习效果明显，能够快速知道 DOM 技术能够用到哪里地方，它是怎么实现的
- 提高动手能力

### 5.3 DOM 的地位



## 5.4 DOM 的热身运动

### 5.4.1 案例描述

- 利用 DOM 实现随机验证码的生成，效果图如下



### 5.4.2 案例分析

- 理解函数作用和语法，学会函数的定义及使用
- 初步理解**元素节点查找的概念**，学会如何获取元素的节点
- 初步理解**如何给节点赋值**
- **再次介绍 JS 内置对象中数组（Array）的使用，以及数学函数（Math）的使用**
- 初步理解 JS 事件机制以及事件的应用

### 5.4.3 参考代码

- HTML 代码

```
<body>
<div>
  <input type="text" placeholder="请输入图片中的验证码" class="ipt">
  <a href="###" onclick="createCode()" class="huan"><span id="ma"></span>&nbsp;<span class="pan"></span>&nbsp;</a><br>
</div>
</body>
```

- CSS 代码

```
<style type="text/css">
div {
  width:300px;
  margin:100px auto;
}
.ipt {
  width:150px;
  height:30px;
}
#ma {
```

```
width:70px;
height:35px;
font-size:22px;
text-align:center;
line-height:30px;
display:inline-block;
border:1px solid #ccc;
vertical-align: middle;
}
a {
color:black;
text-decoration:none;
}
</style>
```

### ● JS 代码

// window.onload 表示等待页面中 HTML 元素加载成功后在执行下面的 JS 代码

```
window.onload = function() {
    var code; // 定义存储 4 位随机验证码

    createCode(); // 执行验证码生成过程
}

function createCode() { // 构建 4 位随机验证码
    code = "";
    var codeLength = 4; // 验证码的长度
    var checkCode = document.getElementById("ma");
    var codeChars = new Array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
        'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z',
        'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'); // 所有候选组成验证码的字符，
    当然也可以用中文的
    for (var i = 0; i < codeLength; i++) {
        var charNum = Math.floor(Math.random() * 52);
        code += codeChars[charNum];
    }
    if (checkCode) {
        checkCode.innerHTML = code;
    }
}
```

## 5.5 函数

### 5.5.1 函数的意义

- 函数又俗称方法
- 简单说**函数就是将多行代码放到一起**，并给这些多行代码去一个名字，当程序（或者用户）需要使用这些代码时候我们在去调用它。例如：前面“**DOM 热身运动**”中讲到切换验证码，**当用户点击切换验证码的时候，就相当于调用了 JS 函数**为我们重新生成一对新的验证码
- 官方概念：函数是由事件驱动的或者当它被调用时执行的可重复使用的代码块

### 5.5.2 函数的学习重点

- 函数的语法基础
- 函数的参数
- 函数的返回值

### 5.5.3 函数定义的语法

函数就是包裹在花括号中的代码块，前面使用了关键词 function：

```
function functionname()  
{  
    这里是要执行的代码  
}
```

当调用该函数时，会执行函数内的代码。

### 5.5.4 函数基础案例

- 定义一个 Button 按钮，当点击 Button 弹出一个浏览器提示框
- 通过此案例要求大家掌握函数的应用，同时理解**函数是由事件驱动的或者当它被调用时执行的可重复使用的代码块**

```
10 日 <script>
11      // 函数以function关键字来定义函数
12      // 要想执行函数一般会配合HTML元素的事件
13      // 如：按钮的点击事件...
14 日  function show() {
15          alert("触发一个按钮点击事件");
16      }
17  </script>
18  </head>
19 日  <body>
20      <button onclick="show()">触发一个按钮点击事件</button>
21  </body>
```

通过按钮onclick来执行函数show

### 5.5.6 函数参数案例

- 参数的意义：接收网页传递过来的数据，然后在将这些数据传递给函数里面的程序进行使用
- 比如：用户 **在文本框中输入的验证码**，**当函数参数接收到用户验证码的数据**，则我们就可以利用 **函数去判断** 用户输入的验证码是否正确

Jt0g 换一张  
√输入正确

// window.onload 表示等待页面中 HTML 元素加载成功后在执行下面的 JS 代码

```
window.onload = function() {
```

```
    var code; // 定义存储 4 位随机验证码
```

```
    createCode(); // 执行验证码生成过程
```

```
}
```

```
function createCode() { // 构建 4 位随机验证码
```

```
    code = "";
```

```
    var codeLength = 4; // 验证码的长度
```

```
    var checkCode = document.getElementById("ma");
```

```
    var codeChars = new Array(0, 1, 2, 3, 4, 5, 6, 7, 8, 9,
```

```
        'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z',
```

```
        'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'); // 所有候选组成验证码的字符，当
```

然也可以用中文的

```
    for (var i = 0; i < codeLength; i++) {
```

```
        var charNum = Math.floor(Math.random() * 52);
```

```
        code += codeChars[charNum];
```

```
    }
```

```
    if (checkCode) {
```

```
        checkCode.innerHTML = code;
```

```
    }
```

```
}

function inputCode(obj) {
    var pan = document.getElementById("pan");
    var inputCodeValue = obj.value;
    if(inputCodeValue.length < 0) {
        pan.innerText = "Is Not Null";
        pan.style.color = "red";
    } else if (inputCodeValue.toUpperCase() != code.toUpperCase()) {
        pan.innerText = "Input Error";
        pan.style.color = "red";
    } else {
        pan.innerText = "Input Valide";
        pan.style.color = "green";
    }
}
```

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="Generator" content="EditPlus®">
<meta name="Author" content="">
<meta name="Keywords" content="">
<meta name="Description" content="">
<title>Document</title>
<script src="03_验证码校验.js"></script>
<style type="text/css">
    div {
        width:300px;
        margin:100px auto;
    }
    .ipt {
        width:150px;
        height:30px;
    }
    #ma {
        width:70px;
        height:35px;
        font-size:22px;
        text-align:center;
        line-height:30px;
        display:inline-block;
```



```
border:1px solid #ccc;
vertical-align: middle;
}
a {
color:black;
text-decoration:none;
}
</style>
</head>
<body>
<div>
<input type="text" placeholder="请输入图片中的验证码" class="ipt" onkeyup="inputCode(this)">
<a href="###" onclick="createCode()" class="huan"><span id="ma"></span>&nbsp;换一张</a><br>
<span class="pan" id="pan"></span>
</div>
</body>
</html>
```

## 5.5.7 函数返回值语法

- 如果我们会希望函数在执行完成后返回一个运算结果的值，则需要给函数定义一个返回值
- 比如：给玩家开发一个飞机大战的游戏，需要记录此玩家游戏分数，则需要函数根据游戏规则进行分数的计算，最后将计算的结果通过函数告诉给玩家，就可以让函数返回一个值
- 函数返回值的语法定义

### 语法

```
function myFunction()
{
var x=5;
return x;
}
```

上面的函数会返回值 5。

## 5.6 ATM 机案例

- 此案例目标是将前面讲的函数进行深入的应用，当然可能会涉及到一些少量的 DOM 操作的技术点，大家不用担心，后面会详细给大家介绍。我们主要目的是希望大家能够将 ATM 案例完整的实现即可

### 5.6.1 案例描述

- 登录：要求用户输入卡号和密码。如果三次之内可以重复输入；三次输入错误，则退出系统

- 登录成功后，显示主界面。要求用户输入所需操作，然后调用相应的函数，实现操作。操作完成后，再回到主界面，继续要求用户输入所需的操作
- 存款：要求用户输入存款金额，完成存款操作
- 取款：如果超支，不允许取款，并给予提示（在原来金额基础上减上存款金额）

## 5.6.2 案例效果图



ATM机

卡号:

密码:

## 5.6.3 案例分析

- 涉及技术知识点
  - JS 基础技术，条件控制，运算符
  - 获取文本框节点的数据，使用 document
  - 页面跳转使用 **window.location.href**，了解是浏览器对象的用法
  - JS 对 DOM 节点属性操作
- 掌握函数的应用

## 5.6.4 ATM 登录程序

- 实现思路
  - 判断输入的卡号是不是数字类型
  - 判断输入的卡号和密码是否为空
  - 登录事件
- 这里**我们只给出 JS 代码，完整代码会在下面内容**中附上

```
var i = 2; // 定义密码输错的次数
```

```
// 判断输入的卡号是不是数字类型
```

```
// 1. 返回 true, 证明不是数字类型
```

```
// 2. 返回 false, 证明是数字类型
```

```
function checkNumber(account) {
```

```
    return isNaN(account);
```



```
}

// 判断输入的卡号和密码是否为空
function checkNull(account, password) {
    if(account.length > 0 && password.length > 0) {
        return true;
    }
    return false;
}

// 登录事件
function login() {
    var account = document.getElementById("account").value;
    var password = document.getElementById("password").value;

    if(!checkNull(account, password)) {
        alert("卡号和密码都不能为空");
        return;
    }

    if(checkNumber(account)) {
        alert("卡号必须是数字");
        return;
    }

    if (i > 0 && account == "123456789" && password == "123") {
        window.location.href = "http://www.zhiliaotang.com";
    } else {
        if (i == 0) {
            alert("你现有的卡号已经锁定...");
            return;
        }
        alert("你还剩下" + i + "次输入卡号和密码的机会");
        i--;
        return;
    }
}
```

### 5.6.5 ATM 操作程序

- 大家主要注意每次操作都会和余额做一次比较

```
// 存款操作
function deposit() {
```

```
var balance = parseFloat(document.getElementById("balance").value);
var deposit = document.getElementById("deposit").value;
if(!deposit.length > 0) {
    alert("请输入你要存款的金额...");
    return;
}

if(checkNumber(deposit)) {
    alert("存款操作请输入数字...");
    return;
}
balance += parseFloat(deposit);
document.getElementById("balance").value = balance;
}

// 取款操作
function withdraw() {
    var balance = parseFloat(document.getElementById("balance").value);
    var withdraw = document.getElementById("withdraw").value;

    if (!withdraw.length > 0) {
        alert("请输入你要取款的金额...");
        return;
    }

    if (checkNumber(withdraw)) {
        alert("取款操作请输入数字...");
        return;
    }

    if (parseFloat(withdraw) > balance) {
        alert("余额不足...");
        return;
    }
    balance -= parseFloat(withdraw);
    document.getElementById("balance").value = balance;
}
```

## 5.6.5 参考代码

- 登录界面代码

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

```
<head>
<meta charset="UTF-8">
<meta name="Generator" content="EditPlus®">
<meta name="Author" content="">
<meta name="Keywords" content="">
<meta name="Description" content="">
<title>Document</title>
<style>
    div {
        width: 300px;
        height: 200px;
        margin: 0 auto;
        border: 1px solid black;
        border-radius: 5px;
        text-align: center;
    }
    p {
        font-size: 20px;
    }
    button {
        border: 0px;
        padding: 5px;
        background-color: green;
        color: white;
    }
</style>
</head>
<body>
<div>
    <p>ATM 机</p>
    <p><label>帐号:</label><input type="text"></p>
    <p><label>密码:</label><input type="password"></p>
    <p><button>登录</button></p>
</div>
</body>
</html>
```

#### ● ATM 主界面代码

```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="Generator" content="EditPlus®">
<meta name="Author" content="">
```

```
<meta name="Keywords" content="">
<meta name="Description" content="">
<title>Document</title>
<script src="04_ATM.js" charset="gbk"></script>
<style>
  div {
    width: 300px;
    height: 230px;
    margin: 0 auto;
    border: 1px solid black;
    border-radius: 5px;
    text-align: center;
  }
  p {
    font-size: 20px;
    text-align: left;
  }
  button {
    border: 0px;
    padding: 5px;
    background-color: green;
    color: white;
  }
</style>
</head>
<body>
<div>
  <p>ATM 机</p>
  <p><label>余额:</label><input type="text" id="balance" value='2000.00' disabled></p>
  <p><label>存款:</label><input type="text" id="deposit">&nbsp;<button onclick="deposit()">存款</button></p>
  <p><label>取款:</label><input type="text" id="withdraw">&nbsp;<button onclick="withDraw()">取款</button></p>
</div>
</body>
</html>
```