

上节课回顾：

1、HTML常见标签 展示页面界面

2、javascript脚本 (js脚本) 用户交互

知识点：变量定义, js函数定义、常用弹窗处理（三大弹窗）、js流程控制语句

js流程控制语句：if/else 嵌套if/else switch语句

布置作业：

支付方式：

Q币单价

Q币数量：

结算金额

```
1 <html>
2 <head>
3 <title>网页的标题</title>
4 <script language="javascript">
5 document.write("执行外部js脚本"+"<br/>")
6 <!--变量-->
7 var a=10;
8 var name="xingyao";
9 var c= true;
10 document.write(a+"<br/>");
11 document.write(name+"<br/>");
12 document.write(c+"<br/>");
13 <!--javascript函数-->
14 function zt1(){
15 <!--函数体语句-->
16 var value="正在学习<<Python基础>>";
17 document.getElementById("result").value=name;
18
19 }
20 function zt2(){
21 <!--函数体语句-->
22 var value2="正在学习<<接口自动化测试>>";
23 document.getElementById("result").value=value2;
24
25 }
26 function zt3(){
27 <!--函数体语句-->
28 var value3="正在学习<<web自动化测试>>";
```

```
29 document.getElementById("result").value=value3;
30
31 }
32 function abc(){
33 alert("操作成功!!");
34 }
35 function con(){
36 var res=confirm("请问需要删除此商品?");
37 if (res==true){
38 alert("删除");
39 }
40 }
41 function pro(){
42 var res=prompt("请输入需要删除的商品的编号");
43 alert(res);
44 }
45
46 function jiesuan(){
47 var paymoney;
48 <!--int类型转化 parseInt-->
49 var price=parseInt(document.getElementById('danjian').value);
50 var count=parseInt(document.getElementById('shuliang').value);
51 var paytype=document.getElementById('fangshi').value;
52
53 <!-- isNaN 表示不是一个数字 !isNaN()判断参数是数字-->
54 if (!isNaN(price) && !isNaN(count)){
55 <!--计算支付金额-->
56 <!--选择不同的支付方式，计算方式不一样-->
57 switch(paytype){
58 case "9":
59 case "8":
60 case "7":
61 case "6":
62 paymoney=price*count*parseInt(paytype)/10;
63 break;
64 default:
65 alert("请选择支付方式!!!");
66
67 }
68 <!--计算支付金额显示在结算金额输入框 -->
```

```
69 document.getElementById('jsje').value=paymoney+"元";
70
71
72 }
73 else{
74 alert("Q币单价及Q币的数量必须是数字!!!");
75 }
76
77
78
79
80 }
81
82
83 </script>
84 </head>
85 <body>
86 <hr color="red" size="2" width="100%">
87 <form>
88 <input type="button" value="专题一" onclick="zt1()"></input><br/>
89 <input type="button" value="专题二" onclick="zt2()" ></input><br/>
90 <input type="button" value="专题三" onclick="zt3()"></input><br/>
91 .....<br/>
92 <input type="button" value="专题八"></input><br/>
93 当前学习阶段:<input type="text" id="result" value="Python基础"></input><br/>
94 <hr color="red" size="2" width="100%">
95
96 <input type="button" value="alert弹窗" onclick="abc()"></input><br/>
97 <input type="button" value="confirm弹窗" onclick="con()"></input><br/>
98 <input type="button" value="prompt弹窗" onclick="pro()"></input><br/>
99 </form>
100 <hr color="red" size="2" width="100%">
101 <form>
102 支付方式:
103 <select id="fangshi">
104 <option value="9">现金支付(9折)</option>
105 <option value="8">支付宝支付(8折)</option>
106 <option value="7">微信支付(7折)</option>
107 <option value="6">京东支付(6折)</option>
108 </select><br/>
```

```
109  Q币单价: <input type="text" id="danjian"></input><br/>
110  购买数量: <input type="text" id="shuliang"></input><input type="button"
value="结算" onclick="jiesuan()"></input><br/>
111  结算金额: <input type="text" id="jsje" ></input>
112  </form>
113  </body>
114  </html>
```

一、web自动化测试学习说明

进阶web自动化测试学习：掌握python编程基础

二、什么web自动化测试

web端测试：对网页的测试

UI测试：web测试+app测试

web自动化测试：通过代码对网页进行测试

UI测试=web自动化测试+APP自动化测试

三、selenium简介

selenium是企业主流应用广泛web自动化测试框架

selenium的三大组件：

- 1、selenium IDE 浏览器插件：实现脚本录制
- 2、WebDriver 实现对浏览器的各种操作（API包）
- 3、Grid 分布式执行，用例同时在多个浏览器同时执行，提搞测试效率

四、web自动化测试环境搭建

环境搭建步骤：

- 1、安装selenium(第三方库)

cmd命令：pip install selenium

- 2、安装浏览器（谷歌浏览器）

常见浏览器：chrome浏览器、IE浏览器、Firefox浏览器

- 3、安装浏览器驱动

谷歌浏览器驱动：chromedriver.exe

IE浏览器驱动：ieserverdriver.exe

Firefox浏览器驱动：geckodriver.exe

特别注意事项：下载驱动版本必须与浏览器的版本一致

下载地址：<http://npm.taobao.org/mirrors/chromedriver/>

chromedriver.exe文件放置到python安装路径（python.exe所在的路径）

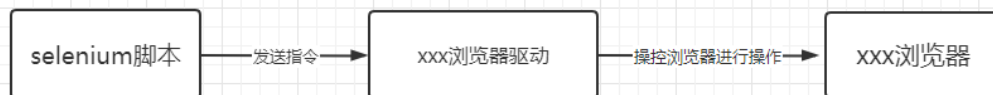
五、web自动化测试第一个脚本

打开浏览器，加载项目地址

```
1 from selenium import webdriver
2
3 # 指令1：打开浏览器
4 driver=webdriver.Chrome()
5 # 指令2：加载项目地址
6 driver.get("https://www.baidu.com")
7 # 不断发送其他指令：不同指令操控浏览器做不同的事情
8
```

六、selenium原理及源码讲解

selenium脚本如何操控浏览器进行对应的操作？



三者如何通信，实现web自动化测试：

结合selenium源码讲解selenium的原理：

- 1、启动浏览器驱动（chromedriver.exe）服务
- 2、selenium脚本与浏览器驱动建立连接，再selenium脚本发送指令（基于http通信）
（发送指令给浏览器驱动，浏览器驱动控制浏览器操作）

不同的指令对浏览器进行不同的操作

指令包（API包）：JsonWireProtocol(JWP)

[JsonWireProtocol · SeleniumHQ/selenium Wiki · GitHub](#)

打开浏览器，执行的命令1：

```
response = self.execute(Command.NEW_SESSION, parameters)
```

指定请求地址http://ip: 端口号/路径+请求方式+请求参数

```
response = self.command_executor.execute(driver_command, params)
```

