



中国科学院大学
University of Chinese Academy of Sciences

计算机科学导论实验

个人作品

zxu@ict.ac.cn
zhangjialin@ict.ac.cn

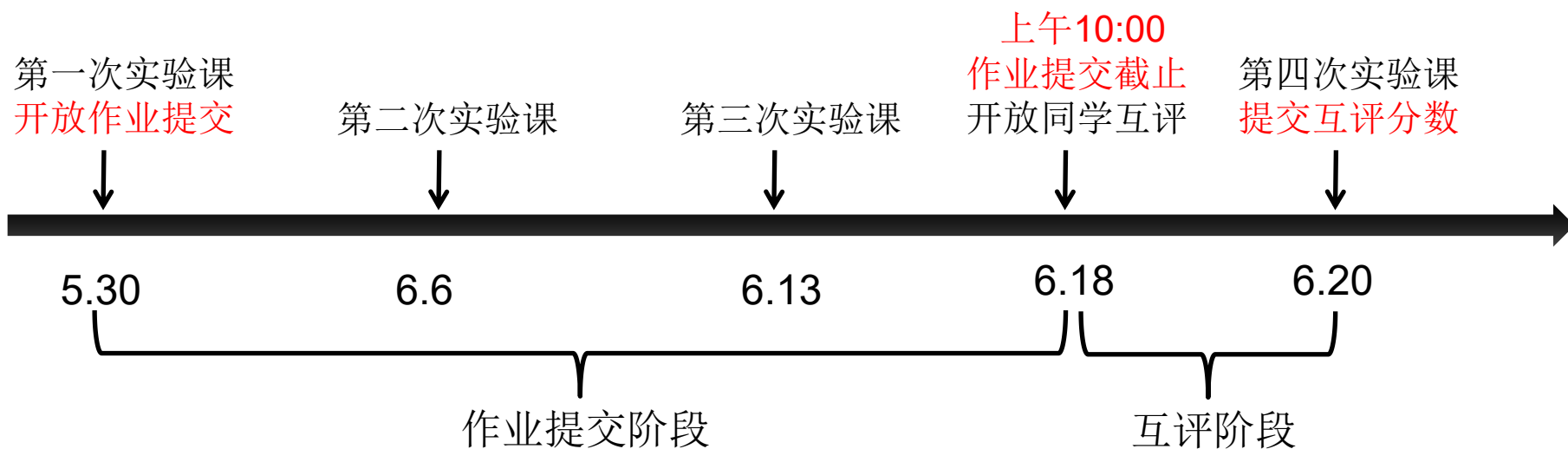
个人作品实验安排

周次	日期	实验课	课程内容	作业提交内容	截止时间
14	5月30日	制作静态网页	HTML结构 HTML语法	作品名称、 personal_artifact.zip	6月18日 10:00前
15	6月6日	制作网页版计算器	修改计算器的结果 读取操作数 执行选中的运算		
16	6月13日	制作网页版时钟	添加静态时钟 让时钟动起来 Debug方法 对比JS和Go		
17	6月20日	网页打分	小组讨论与互评	评分结果	

本课程将会以“王小二”编写一个网页博客介绍他发明了网页版时钟和计算器的伟大事迹为例，讲述相关的编程知识。

评分标准

- 6月18日 10:00 前可反复提交并预览
- 教师评分+同学互评
 - 同学互评开放时间：6月18日 17:00 – 6月20日 13:30
 - 评分维度：创造性与完成度



我们将实现网页版计算器：

计算器和时钟养成记

作者：王小二* 时间：2020年02月26日

发明前

在发明前，我也不知道自己能不能做成。

列表：王小二的动机

1. 学习CSS
2. 学习HTML
3. 学习JS
4. 写计算器和时钟

发明中

过程中，查了好多资料，好累但很快乐。下表是我这几天的艰辛历程：

表：王小二艰难历程表

事件	感悟
看CSS语法	一天没睡
看HTML语法	还是没睡
看JS语法	还是GO香

发明后

终于做好了，虽然很粗糙，那也是亲儿子

计算器：

操作数1:

操作数2:

☐ 加 ☐ 减 ☐ 乘 ☐ 除 ☐ 模

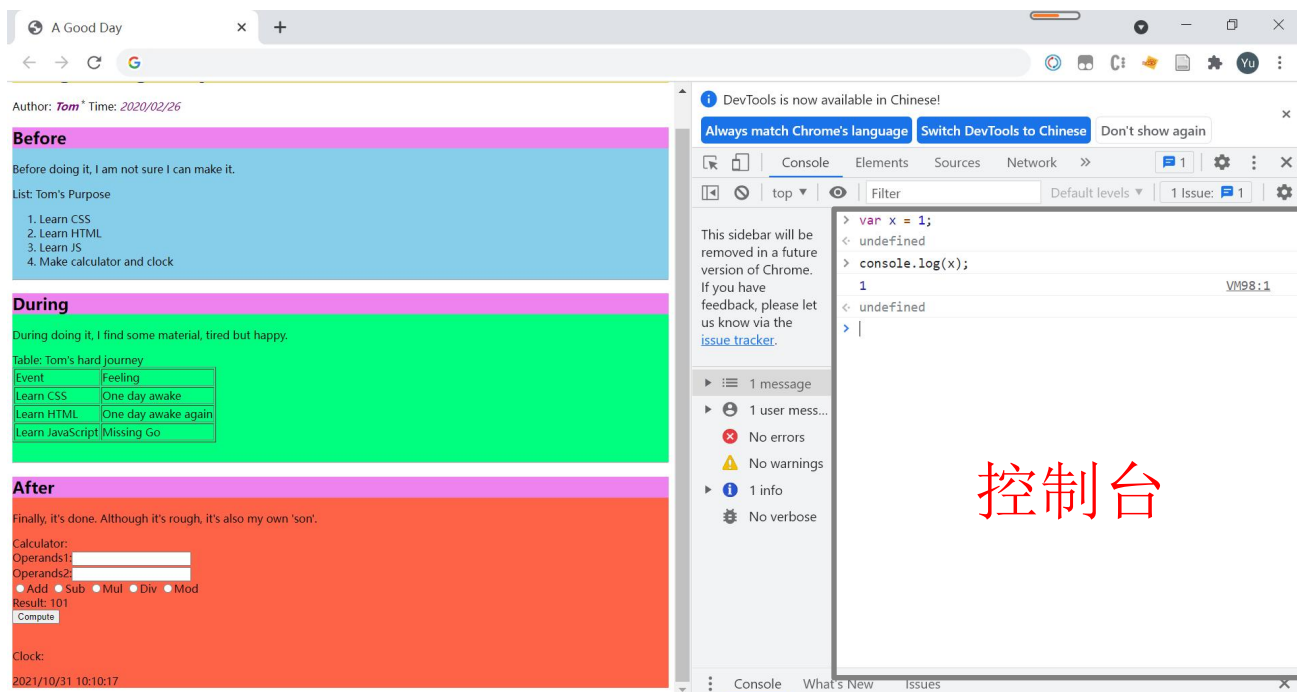
结果: XXXX

网页版
计算器

index.html

控制台 Console

- 浏览器都有一个类似于终端(**terminal**)的**控制台(console)**
 - 控制台是 debug 时的标准输入(**StdIn**)、标准输出(**StdOut**)和标准错误输出(**StdErr**)
 - 我们可以直接在控制台执行 JS 代码, 无需一个 **html** 文档
- 打开浏览器, 按 **F12** 可打开控制台



控制台 Console

- 一般来说，在控制台执行代码不改变网页的行为
 - 除非在控制台直接修改网页内容
- 在控制台输入下列代码，网页内容不改变

```
> var x=1;
< undefined
> console.log(x);
1
< undefined
> x = x + 1;
< 2
> console.log(x);
2
< undefined
> |
```

Debug: console.log 函数

- 对比 console.log(...) 函数与 fmt.Println(...) 函数

	console.log	fmt.Println
输入	字符串、整数、浮点数、对象	字符串、整数、浮点数
输出	输出到控制台	输出到终端
用途	专门用于 debug	可以用于 debug，也用于程序和用户之间的交互

插入 JavaScript 代码

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>美好的一天</title>
  </head>
  <body>
    .....(HTML 部分)
    <script>
      ...
    </script>
  </body>
</html>
```



JavaScript
部分

第二周 提纲

- 修改计算器的结果
 - 代表 HTML 元素的 JavaScript 对象
- 读取操作数
 - JavaScript "onclick" 事件
- 执行选中的运算
 - JavaScript Array 对象
 - JavaScript for-循环

JavaScript 编程知识-对象

- 对象可以认为是一种“复合”的数据类型，它是一些**属性**和一些**方法**的集合，比如：将一个人看成一个对象，它有身高、体重、年龄、性别等**属性**，以及走路、跑步、说话等**方法**
- 使用 `.`（英文句点）来访问对象的属性和方法，如下：
- 对象的属性：类似于对象包含的变量
 - `obj.property`
- 对象的方法：类似于对象包含的函数
 - `obj.method()`;

JavaScript编程知识—代表HTML元素的对象

- 为了实现计算器，王小二需要知道怎样修改计算器的结果，所以他学习了如何用 JS 访问 HTML 元素：
- JS 可以把已定义的HTML元素当作对象来访问（2 步）：

1. 通过 id 属性获取 HTML 元素

```
var x = document.getElementById("p1");
```

- document.getElementById("p1") 的返回值是代表id属性为 "p1" 的 HTML 元素的对象
- 因此，x 是代表 id 属性为 "p1" 的 HTML 元素的对象

2. 通过读写获取到的对象的属性来读写 HTML 元素的属性

```
x.innerHTML = "newContent";
```

- 设置该 HTML 元素的 innerHTML 属性为 "newContent"
- id 属性为 "p1" 的 HTML 元素的起始、终止标签中间的内容就变成了 "newContent"

- 上面 2 步可以合成一句：

```
document.getElementById("p1").innerHTML = "newContent";
```

修改计算器的结果

代码:

```
<script>  
    var z = 101;  
    var result =document.getElementById("result");  
    result.innerHTML = z;  
</script>
```

效果:

计算器:
操作数1:
操作数2:
☒ 加 ☐ 减 ☐ 乘 ☐ 除 ☐ 模
结果: 101

第二周 提纲

- 修改计算器的结果
 - 代表 HTML 元素的 JavaScript 对象
- 读取操作数
 - JavaScript "onclick" 事件
- 执行选中的运算
 - JavaScript Array 对象
 - JavaScript for-循环

JavaScript 编程知识-事件

- 王小二学会改变计算器的结果后，还需要实现：当点击按钮后，网页读取用户输入的操作数。因此，他学习了关于事件的知识：
- 3 个常见事件：
 - 用户点击按钮 (`<input type="button" onclick="JS 代码 ">`)
 - 用户的输入内容改变 (`<input type="text" oninput="JS 代码 ">`)
 - 浏览器加载完某个网页 (`<script> window.onload= 一个函数定义 </script>`)
- 当一个事件发生时，可以执行一段 JS 代码来处理该事件

添加事件响应

- onclick= “JS 代码 ” 中的 “JS 代码” 通常是一条函数调用语句，例如：

代码：

```
...  
<input type="button" value="计算" onclick="run_calculator()">  
...  
<script>  
    function run_calculator(){  
        var operand1 = document.getElementById('operand1').value - 0;  
        var operand2 = document.getElementById('operand2').value - 0;  
        var z = operand1 + operand2;  
        document.getElementById('result').innerHTML = z;  
    }  
</script>
```

效果：

计算器：

操作数1:

操作数2:

☐ 加 ☐ 减 ☐ 乘 ☐ 除 ☐ 模

结果: xxxx

点击 “计
算”



计算器：

操作数1:

操作数2:

☐ 加 ☐ 减 ☐ 乘 ☐ 除 ☐ 模

结果: 303

第二周 提纲

- 修改计算器的结果
 - 代表 HTML 元素的 JavaScript 对象
- 读取操作数
 - JavaScript "onclick" 事件
- 执行选中的运算
 - JavaScript Array 对象
 - JavaScript for-循环

JavaScript 编程知识— Array 对象

- 王小二成功读取了计算器的两个操作数后，还需要根据用户选择的运算种类进行运算，于是他将进行 5 种运算的 5 个函数放在一个数组中：

- Array 对象（数组）的创建

```
var ar = new Array(); // 变量 ar 中存一个数组（不需要指定长度，  
                        使用时直接赋值，长度自动确定）  
var ar = ["Java", "Script"]; // 可以使用 "[]" 创建数组  
var ar = [0, "ABC", [true, false]]; // 数组的各元素类型可以不同，  
                                       下标从0开始
```

- 数组元素的修改

```
ar[0]=1;
```

- 数组的属性

```
var ar=[0,1,2];  
var len = ar.length; // len=3（数组的长度）
```

使用函数数组

```
<script>
...
function add(x,y){ return x+y; } //加
function sub(x,y){ return x-y; } //减
function mul(x,y){ return x*y; } //乘
function div(x,y){ return x/y; } //除
function mod(x,y){ return x%y; } //模
function run_calculator(){
    var operand1 = document.getElementById('operand1').value - 0;
    var operand2 = document.getElementById('operand2').value - 0;
    var function_array = [add, sub, mul, div, mod];
    var z = function_array[1](operand1, operand2);
    document.getElementById("result").innerHTML=z;
}
</script>
```

效果:

计算器:

操作数1:

操作数2:

☐ 加 ☐ 减 ☐ 乘 ☐ 除 ☐ 模

结果: xxxx

点击“计算”按钮



计算器:

操作数1:

操作数2:

☐ 加 ☐ 减 ☐ 乘 ☐ 除 ☐ 模

结果: -9

第二周 提纲

- 修改计算器的结果
 - 代表 HTML 元素的 JavaScript 对象
- 读取操作数
 - JavaScript “onclick” 事件
- 执行选中的运算
 - JavaScript Array 对象
 - JavaScript for-循环

JavaScript 编程知识—for 循环

- 在王小二之前的代码里，无论用户所选运算是什​​么，都会执行减法。因此，他还需要遍历一遍网页中的五个选项，以确定用户选择了哪种运算，于是学习了下面关于 for 循环的知识
- for 循环：

```
for(statement1;expression1;statement2){  
    //some code  
}
```

- 和 Go 中的 for 循环类似，区别仅仅是 JS 中的 for 循环需要在 for 后面加一对圆括号

回顾计算器的 HTML 代码:

计算器:

<form>

操作数1: <input type="text">

操作数2: <input type="text">

<input type="radio" name="op">加

<input type="radio" name="op">减

<input type="radio" name="op">乘

<input type="radio" name="op">除

<input type="radio" name="op">模

结果: xxxx

<input type="button" value="计算">

</form>

效果:

计算器:
操作数1:
操作数2:
● 加 ● 减 ● 乘 ● 除 ● 模
结果: xxxx
计算

用 for 循环遍历数组

```
function run_calculator(){  
    ...  
    var function_array = [add, sub, mul, div, mod];  
    var options = document.getElementsByName("op");  
    var z;  
    for(var i=0; i<function_array.length; i++){  
        if(options[i].checked == true){  
            z = function_array[i](operand1, operand2);  
            break;  
        }  
    }  
    document.getElementById('result').innerHTML = z;  
}
```

效果:

计算器:

操作数1:

操作数2:

☐ 加 ☐ 减 ☒ 乘 ☐ 除 ☐ 模

结果: xxxx

点击“计算”按钮



计算器:

操作数1:

操作数2:

☐ 加 ☐ 减 ☒ 乘 ☐ 除 ☐ 模

结果: 156