---------------------------------------------------

计划课程安排：

FUNDAMENTAL(7天)         该阶段已结束

OOP(11天)           该阶段已结束

JAVASE01(10天) 该阶段已结束

JAVASE02(12天) 该阶段已结束

WEBBASIC(7天) ←当前为该阶段第7天

JQUERY(2天)

DATABASE(6天)

JDBC(3天)

SERVLETJSP(12天)

SPRINGMYBATIS01(8天)

AJAX(2天)

SPRINGMYBATIS02(12天)

LINUX(2天)

NGINX(2天)

REDIS(2天)

T-STORE(5天)

预计合计：102天

预计剩余：56天

注：上述信息随时可能发生更改，仅供参考。

---------------------------------------------------

**2019年5月7日**

**Java就业班-第56天**

**8:00实训室2-早自习**

早到早起看书。《Java编程思想》

摘录①：

Alan Kay曾经总结了第一个成功的面向对象语言、同时也是Java所基于的语言之一的Smalltalk的5个基本特性，这些特性表现了一种纯粹的面向对象程序设计方式：

1. **万物皆为对象**。将对象视为奇特的变量，它可以存储数据，除此之外，你还可以要求它在自身上执行操作。理论上讲，你可以抽取待求解问题的任何概念化构件（狗、建筑物、服务等），将其表示为程序中的对象。
2. **程序是对象的集合，他们通过发送消息来告知彼此所要做的**。要想请求一个对象，就必须对该对象发送一条消息。更具体地说，可以把消息想象为对某个特定对象的方法的调用请求。
3. **每个对象都有自己的由其他对象所构成的存储**。换句话说，可以通过创建包含现有对象的包的方式来创建新类型的对象。因此，可以在程序中构建复杂的体系，同时将其复杂性隐藏在对象的简单性背后。
4. **每个对象都拥有其类型**。按照通用的说法，“每个对象都是某个类（class）的一个实例（instance）”，这里“类”就是“类型”的同义词。每个类最重要的区别于其他类的特性就是“可以发送什么样的消息给它”。
5. **某一特定类型的所有对象都可以接收同样的消息。**这是一句意味深长的表述，你在稍后便会看到。因为“圆形”类型的对象同时也是“几何形”类型的对象，所以一个“圆形”对象必定能够接受发送给“几何形”对象的消息。这意味着可以编写与“几何形”交互并自动处理所有与几何形性质相关的事物的代码。这种可替代性（substitutability）是OOP中最强有力的概念之一。

Booch对对象提出了一个更加简洁的描述：**对象具有状态、行为和标识**。这意味着每一个对象都可以拥有内部数据（它们给出了该对象的状态）和方法（他们产生的行为），并且每一个对象都可以唯一地与其他对象区分开来，具体说来，就是每一个对象在内存中都有唯一的地址。

摘录②：

**访问控制**的第一个存在原因就是让客户端程序员无法触及他们不应该触及的部分——这些部分对数据类型的内部操作来说是必需的，但不是用户解决特定问题所需的接口的一部分。这对客户端程序员来说其实是一项服务，因为他们可以很容易地看出哪些东西对他们来说很重要，而哪些东西可以忽略。

访问控制的第二个存在原因就是允许库设计者可以改变类内部的工作方式而不用担心会影响到客户端程序员。例如，你可能为了减轻开发任务而以某种简单的方式实现了某个特定类，但稍后发现你必须改写它才能使其运行得更快。如果接口和实现可以清晰地分离并得以保护，那么你就可以轻而易举地完成这项工作。

（up主：访问控制也是“高内聚低耦合”中低耦合的实现方式之一。）

**8:30实训室2-早自习**

点完名开始晨讲了。今日晨讲主题——TreeSet排序

**9:03实训室2-正课**

开始上课。

从今天开始学习JavaScript。(总算是TMD开始有新的进展了。)

初识JS：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>初次了解JavaScript</title>

<!-- JS语言的引入方式 -->

<!-- 内部 -->

<script type="text/javascript" src="first.js">

alert("内部引入方式");

</script>

</head>

<body>

<!-- 内联 -->

<input type="button" value="天下漫友是一家！" onclick="alert('AC在，爱一直在！')">

</body>

</html>

first.js：

/\*\*

\* JS语言的引入方式-外部

\*/

alert("外部引入方式");

**9:50实训室2-正课**

JavaScript的方法与调用：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>JavaScript的方法与调用</title>

<script type="text/javascript">

//无参无返回值

function fn1(){

alert("fn1执行了！");

}

/\* 调用方法和java一样 \*/

fn1();

//有参无返回值

function fn2(name , age){

alert(name + ":" + age);

}

//调用

fn2("Kamui",25);

//无参有返回值

function fn3(){

return "Subaru";

}

var s = fn3();

alert(s);

//有参有返回值

function fn4(x,y){

return x\*y;

}

var result = fn4(2,8);

alert(result);

/\* 第二种格式： \*/

var fn5 = function(){

alert("fn5执行！");

}

//调用方法一样

fn5();

/\* 第三种格式： \*/

var fn6 = new Function("x","y","z","return x\*y\*z;");

//调用

var result = fn6(3,4,5);

alert(result);

</script>

</head>

<body>

<input type="button" value="按钮" onclick="fn1()">←这是通过按钮启动方法

</body>

</html>

**10:26实训室2-正课**

获取、修改页面元素：

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>获取、修改页面元素</title>

<script type="text/javascript">

function myfn(){

var d = document.getElementById("d1");

d.innerText = "なにを食べよう～";s

}

</script>

</head>

<body>

<input type="button" value="按钮" onclick="myfn()">

<div id = "d1">Time for lunch</div>

<!-- 可通过 .getElementById("d1")的方法获取页面元素 -->

</body>

</html>

**11:10实训室2-正课**

课程笔记：

###JavaScript

- 作用：给页面添加动态效果和动态数据 ， html负责搭建页面结构和内容，css美化页面

###JS历史

1995年由网景（NetScape）公司发布，LiveScript，同年更名为JavaScript，和Java语言没啥关系，为了蹭热度

###JS特点

1. 属于脚本语言，不需要编译，由浏览器解析执行

2. js语言可以嵌入到html页面中

3. js语言基于面向对象

4. js属于弱类型的语言

5. 交互性高

6. 安全性高 js语言只能访问浏览器中的各种数据，不能访问浏览器以外的数据

###JS语言的引入方式

- 内联：在元素的事件属性中添加js代码，当事件触发时js代码执行

- 内部：在html页面中添加<script>标签，当页面加载时执行

- 外部：在单独的js文件中写js代码，在页面中通过script标签的src属性引入，如果script标签引入文件的话则不能在标签体内继续写js代码

###JS语法

####数据类型

- JavaScript中只有对象类型

- 常见的对象类型

1. 数值类型 number

2. 字符串类型 string

3. 布尔值类型 boolean

4. 未定义类型 undefined

5. 自定义类型 object

####变量的声明和赋值

- js属于弱类型语言

- java： int x = 10;

- js: var x = 10; var s = "abc"; var b = true/false;

####常见类型介绍

1. 数值类型

- js中数值类型的底层都是浮点数，在使用过程中会自动转换整数或小数

java int x =5 int y= 2 int z = x/y; z=2

js var x = 5 var y = 2 var z = x/y; z=2.5

- NaN: Not a Number 不是一个数， 用于判断变量是否是数值类型 ， 使用 isNaN(变量) 进行判断 isNaN(x) 返回值是true或false ，true代表不是数，false代表是数

2. 布尔值类型

var b = true/false;

3. 字符串

var s = "abc";

var s = 'abc';

4. 未定义类型undefined

- 当一个变量只声明不赋值的情况下，则此变量为未定义类型

###各种类型间的隐式转换

1. 数值类型

- 转字符串： 直接转 24-> "24"

- 转布尔值： 0和NaN转成false 其它转成true

2. 布尔值类型

- 转字符串： 直接转 true->"true" "abc"+true

- 转数值： true->1 false->0 18+true=19

3. 字符串类型

- 转数值： 能转直接转 不能转转成NaN

加法会把数值转字符串 "20"+5 = "205"

减乘除会把字符串转成数值 "20"-5 = 15

\*\*任何数值和NaN进行任何运算 结果都是NaN\*\*

- 转布尔值： 空字符串转false 其它true

4. 未定义undefined

- 转数值： NaN

- 转字符串： 直接转 "abc"+undefined "abcundefined"

- 转布尔值： false if(undefined){alert("aaa")}

5. null

- 转数值： 0

- 转布尔值： false

- 转字符串： 直接转 "abc"+null = "abcnull"

####运算符 + - \* / % = > < >= <= == !=

- 和Java大体相同

- ==和=== ： == 先统一类型再比较值 ===先比较类型 如果类型相等再比较值

- 除法运算 会自动转换整数和小数

- typeof x; typeof 66 + 6 = "number6"

####语句 if else while do while for switch case

- 和Java大体相同

- 不同点：

1. if 和 while 括号里面的内容 如果不是布尔值会自动隐式转换成布尔值

2. for循环中int i改成var i 不支持增强（新）for循环

####方法声明

- java： public void 方法名(参数列表){方法体}

- js： function 方法名(参数列表){方法体}

- 常见的四种方法：

1. 无参无返回值

2. 无参有返回值

3. 有参无返回值

4. 有参又返回值

- 三种声明方法的格式：

1. function 方法名(参数列表){方法体}

2. var 方法名 = function(参数列表){方法体};

3. var 方法名 = new Function("参数1","参数2","方法体");

###和页面元素相关的方法

1. 通过id获取页面元素的方法

var d = document.getElementById("d1");

2. 获取和修改元素的文本内容

d.innerText

d.innerText = "abc";

**14:05实训室2-正课**

练习代码demo04:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>获取、修改页面元素2</title>

<script type="text/javascript">

function myfn() {

//得到文本框和div

var i = document.getElementById("i1");

var d = document.getElementById("d1");

//取出文本框的文本给到div

d.innerText = i.value;

}

</script>

</head>

<body>

<input type="text" id="i1">

<input type="button" value="按钮" onclick="myfn()">

<div id="d1"></div>

</body>

</html>

**14:35实训室2-正课**

练习代码demo05:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>练习：实现求任意数的平方的功能</title>

<script type="text/javascript">

function squared() {

var x = document.getElementById("x1");

var result = document.getElementById("r1");

if (isNaN(x.value)) {

result.innerText = "输入有误，请重新输入！";

} else {

result.innerText = parseFloat(x.value)\*parseFloat(x.value);

}

}

</script>

</head>

<body>

<input type="text" id="x1">

<input type="button" value="输出平方值" onclick="squared()">

<div id="r1"></div>

</body>

</html>

**15:21实训室2-正课**

练习代码demo06:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>练习：实现低级计算器</title>

<script type="text/javascript">

function math(x){

var num1 = document.getElementById("num1");

var num2 = document.getElementById("num2");

var result = document.getElementById("result");

switch(x){

case 1:

result.innerText = parseFloat(num1.value) + parseFloat(num2.value);

break;

case 2:

result.innerText = parseFloat(num1.value) - parseFloat(num2.value);

break;

case 3:

result.innerText = parseFloat(num1.value) \* parseFloat(num2.value);

break;

case 4:

result.innerText = parseFloat(num1.value) / parseFloat(num2.value);

break;

}

}

</script>

</head>

<body>

<!-- 只能输入包含2个小数的数字 -->

<input type="text" id="num1" oninput="value=value.replace(/^(\-)\*(\d+)\.(\d\d).\*$/,'$1$2.$3')">

<input type="text" id="num2" oninput="value=value.replace(/^(\-)\*(\d+)\.(\d\d).\*$/,'$1$2.$3')">

<input type="button" value=" + " onclick="math(1)">

<input type="button" value=" - " onclick="math(2)">

<input type="button" value=" \* " onclick="math(3)">

<input type="button" value=" / " onclick="math(4)">

<div id="result"></div>

</body>

</html>

**15:42实训室2-正课**

练习代码demo07:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>练习：实现连续添加元素功能</title>

<script type="text/javascript">

function upload(){

var up = document.getElementById("up");

var ul = document.getElementById("myul");

ul.innerHTML += "<li>"+ up.value +"</li>" ;

}

</script>

</head>

<body>

<input type="text" id="up">

<input type="button" value="添加" onclick="upload()">

<ul id="myul"></ul>

</body>

</html>

**16:02实训室2-正课**

练习代码demo08:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>练习：猜数字游戏(0-100以内)</title>

<script type="text/javascript">

var number = parseInt(Math.random()\*100);

function check(){

var guess = document.getElementById("guess");

var d = document.getElementById("d1");

if(number<guess.value){

d.innerText = "猜大了";

}else if(number>guess.value){

d.innerText = "猜小了";

}else{

d.innerText = "不容易，中了！";

}

}

</script>

</head>

<body>

<input type="text" id="guess">

<input type="button" value="猜" onclick="check()">

<div id="d1"></div>

</body>

</html>

**16:38实训室2-正课**

练习代码demo09:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>练习：实现增加元素标签功能</title>

<script type="text/javascript">

function fn(){

var i = document.getElementById("i1");

//往body里面添加

document.body.innerHTML += "<div>"+ i.value +"</div>";

}

</script>

<style type="text/css">

div{

width: 100px;

height: 100px;

border: 1px solid red;

}

</style>

</head>

<body>

<input type="text" id="i1">

<input type="button" value="添加" onclick="fn()">

</body>

</html>

**17:23实训室2-正课**

练习代码demo10:(←不知道为什么这段从讲师那抄来的代码在我这不能正确执行)

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>练习：实现通过按钮添加指定颜色div元素</title>

<script type="text/javascript">

function select(color){

document.body.innerHTML += "<div class='"+ color +"'></div>"

}

</script>

<style type="text/css">

.red {

width: 50px;

height: 50px;

background-color: red;

}

.blue {

width: 50px;

height: 50px;

background-color: blue;

}

</style>

</head>

<body>

<input type="button" value="红色" onclick="select('red')">

<input type="button" value="蓝色" onclick="select('blue')">

</body>

</html>

最后进行了一下课程回顾，便下课了。

**19:01宿舍-休息**

总算是开始讲js的内容了，我是真的已经受不了这个讲师讲的课了，听他的课是真的无聊，讲课风格也极度催眠，简直是煎熬。具体现象为只要一下课休息，所有学员倒头就睡。

另外，大概从今天起吧，没有什么必要的情况，往后的晚自习准备全部都翘掉了，毕竟坐在教室里空气又难受，坐一天屁股也疼，把痔疮养出来了得不偿失。最主要的是，我想早些睡觉……回到宿舍里好歹能够躺床上看会儿书，如果困了也正好可以直接睡，这样白天上课的时候也不至于精神那么疲惫。

其实我现在就已经很疲惫了，连吃晚饭的胃口都没有，于是我决定现在先小睡一会，等到晚一点睡醒了再起来看书。

由于今天的课程是已经讲完了，以免我等会一觉睡到天亮，所以我觉得今天的日记也差不多可以投稿了，至于看书的笔记如果我有觉得不错的地方还是会摘录下来贴到日记里。虽然现在才晚上7点，但是人累了就该休息嘛。各位老哥也是，早点休息，狗命要紧。

今日感想总结：HTML、CSS、JS应该用什么开发工具才好？eclipse用起来感觉很不方便，也几乎没有错误代码提示，一套代码写完跑起来简直就是不成功便成仁了……

今日评分：5/10分

今天就到这里，明天继续投稿。

感谢支持！

================================================================

**Acer程序员群：790482850**

**up主的java学习日记word文件及全部代码（随日记同步更新）：**

Adress：github.com/wt62635