**传智播客前端和移动开发学院**

**http://web.itcast.cn**



复习和进阶

第0天课堂笔记（本课程共0天）

**目录**

[一、 函数复习 3](#_Toc434566406)

[1.1. 函数的定义 3](#_Toc434566407)

[1.2. 函数相关术语：形参 实参 3](#_Toc434566408)

[1.3. 如何查看实参 arguments 3](#_Toc434566409)

[1.4. 函数的作用 封装性 3](#_Toc434566410)

[二、 闭包复习（可选） 3](#_Toc434566411)

[三、 原型对象复习 3](#_Toc434566412)

[3.1. 函数和对象的共同点 – 封装多个功能 3](#_Toc434566415)

[3.2. 面向对象的重要性 3](#_Toc434566416)

[3.3. 什么是对象 4](#_Toc434566417)

[3.4. 抽象和具体 4](#_Toc434566418)

[3.5. 如何将现实世界的实体抽象成编程世界的对象 5](#_Toc434566419)

[四、 Json对象复习 5](#_Toc434566420)

[4.1. 如何使用json对象描述新闻 5](#_Toc434566422)

[4.2. 如何使用json对象描述新闻列表 5](#_Toc434566423)

[4.3. 如何使用json对象描述产品 5](#_Toc434566424)

[4.4. 如何使用json对象描述产品列表 5](#_Toc434566425)

[五、 字面量 5](#_Toc434566426)

[5.1. 对象的字面量形式 5](#_Toc434566428)

[5.2. 数组的字面量形式 5](#_Toc434566429)

[5.3. 函数的字面量形式 5](#_Toc434566430)

[5.4. 正则表达式的字面量形式 5](#_Toc434566431)

# 面向对象的重要性

我们工作当中可能会遇到的项目：

PC web

移动web

移动app – 混合app

Html5 canvas 游戏

相关知识点我们再前面已经学习了。每个知识体系需要的技术是不一样的。但是有些又是公共的。

比如PC web的知识基本可以运用在其他项目开发中。

移动web开发：需要学习移动端布局，响应式布局等。

混合app开发：前面的技术完全是混合app开发的基础，同时需要学习其他一些技术。后面会详细讲解。

但是面向对象编程确是通用的，也是所有平台都需要学习的通用基础知识。

# 前后台认识整个开发过程 1

# 函数复习和进阶 30

再讲面向对象之前我们先来复习下函数

## 函数的定义

## 函数相关术语：形参 实参

## 如何查看实参 arguments

## 常用数据结构

数据结构是我们再编程的时候，将需要的数据临时保存在一个变量中。

比如对于简单的场景，我们用字符串，数字等保存数据。

对于集合类，我们一般使用数组保存。

对于较为复杂的我们一般使用json对象保存数据。

下面我们学习如何使用json对象保存现实世界的数据。

最简单： var s = ‘12222’ var num = 1

集合类： var 星座 = 【】；

复杂类： var json = {}；//既可以包含简单的变量，又可以包含数组

简单和集合类，我们前面学习了很多，以及各种使用场景。下面我们复习如何使用json描述数据

# 对象复习和进阶 30

## 什么是对象

生活中，任何实体都可以看做一个对象，比如人，动物，汽车，飞机，苹果，香蕉。

对象又分成抽象和抽象。

比如 人是抽象的，具体的某个人是具体的。

学生是抽象的，田佩瑶同学是具体的。

水果是抽象的，具体的某个水果是具体的。

## 抽象和具体

|  |
| --- |
| 什**么是抽象呢？什么是具体呢**？  比如：水果是对水果 抽象，具体包含：梨子，苹果，葡萄  比如：学生是 抽象，具体的某个学生是具体，张三 李四  定义一个学生抽象类：  var Student = function（）{  this.name =’’;  this.score = ‘’;  this.age=’’;  this.sex=’’;  } |
| Student.prototype = function(){  This.    } |

|  |
| --- |
|  |

## 如何将现实世界的实体抽象成编程世界的对象

比如开发京东商城产品详细信息，我们可以定义一个产品对象

属性：名称，描述，价格

方法：就是行为—比如添加到购物车，查看产品详细信息

比如开发京东商城购物车，我们可以定义一个购物车对象

属性：产品列表，总价格

方法：获取产品列表，计算产品总价格（当修改产品个数需要重新计算）

## 高级语言对象的定义方式

## JS中对象的定义方式 - 双对象形式

构造对象：放置属性

原型对象：放置方法

# 函数和对象的共同之处- 封装性 60

## 函数的封装性 – 单一功能

内置函数封装性

自定义对象封装性

演示tab封装性

总结：只能封装一个功能

## 对象的封装性 – 多个功能 多个函数

**函数和对象虽然彼此似乎没什么关系，但是他们之间是有关系的：**

1. 他们的作用都是封装性
2. 函数只能封装一个功能，而对象可以封装多个功能。

解析：

函数的封装性：封装一些功能，我们无需了解函数代码如何编写的，可以直接使用

对象的封装性：比如document封装了和DOM操作相关的方法

封装性 –内置对象： document.getElementById

有了工作经验后需要领悟的：封装变化 – 可以将一些容易变化的代码从依托代码中提取出来，专门封装起来，好处，一旦需求变化，我们只需要修改这个函数即可

封装多个功能案例演示

我们可以将很多函数封装在一个对象中，以方便以后使用

我们可以封装一些常用的字符串，数组，数字，Ajax，正则等方便我们以后使用，我们以后在编程的时候会需要编写很多算法，很多算法都是经过几代人千锤百炼写出来的，通过测试，分析，可能花了半年的时间才到达完美，如果我们每次编程都花半年的时间研究这些算法，那么我相信没有一个项目会成功。所以我们可以将别人写好的封装起来，以备日后使用。

比如：字符串，虽然系统内部提供了很多关于字符操作的函数，但是不够用：

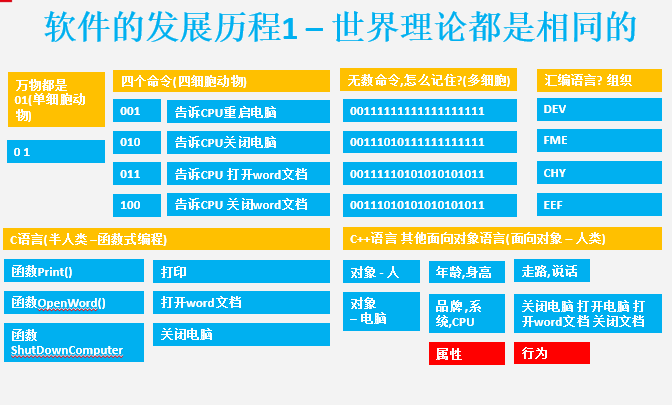
比如： trim，去除字符串前后的空白是非常常用的。

下面我们使用对象封装一些常用方法，以便我们以后使用

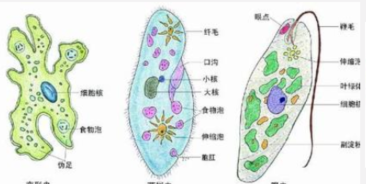
Function Itcast（）{}

# 对象的作用 - 从人类发展历程理解面向对象语言

一节课时间讲解大学计算机原理，编译原理相关课程



单细胞动物就类似计算机中0 1



多细胞 比如用用001 010 011表示 123 来描述世界



## 二进制时代

## 汇编时代

## C语言时代 面向过程时代

## 面向对象编程时代

C++语言：面向对象的鼻祖

目前流行语言排行榜 – C C++遥遥领先

其他流行的面向对象语言：java C# JavaScript

## 未来时代

## 面向对象思维

* 面向对象的语法不重要，重要的是面向对象思维。
* 如何用面向对象分析问题。

面向过程思维：函数为主题

面向对象思维：宏观思考，比如淘宝

# 对象三大使用地方 - 概述 15



封装一些常用的功能 –前面已经讲过

描述数据

面向对象编程

# 对象三大使用地方 - 描述数据 60

## 使用构造 原型方式描述数据

## 使用json描述数据 – 更常用

### 如何使用json对象描述新闻

### 如何使用json对象描述新闻列表

### 如何使用json对象描述产品

### 如何使用json对象描述产品列表

## 字面量定义和总结



### 对象的字面量形式

### 数组的字面量形式

### 函数的字面量形式

### 正则表达式的字面量形式

# 对象三大使用地方 – 面向对象编程

## 一般页面开发方式



### 一般页面编写方法

var name = 'iphone8'  
 var description = '手机中的战斗机 '  
 var price = 0;  
 var domName = **document**.getElementById('pname')  
 var domPrice = **document**.getElementById('pprice')  
 var domad = **document**.getElementById('p-ad')  
 var domadd= **document**.getElementById('add')  
 var dombuy= **document**.getElementById('buy')  
  
 **window**.onload=function(){  
 domName.**innerHTML** = name  
 domPrice.**innerHTML** = price  
 domad.**innerHTML** = description  
 domadd.onclick=function(){  
 price = 7000;  
 domPrice.**innerHTML**=price;  
 }  
 dombuy.onclick=function(){  
 price = 7000;  
 domPrice.**innerHTML**=price;  
 }  
 }

### 数据绑定概念

数据绑定概念：就是将数据和前端的dom进行绑定，换句话说就是将原来的造的假数据变成真数据  
什么是假数据：就是静态页面开发人员随便造的一些假数据，这些数据只用于演示页面大概什么样的，后期会将这些数据替换成后台传递过来的真正的数据

什么是真数据：

真数据的来源：后台开发人员传递给你的

数据绑定有两种方式：我们编写好静态页面之后交给后台开发人员，后台开发人员使用某种后台技术进行数据绑定，比如php，asp，jsp，最后我们会稍微讲解下后台的开发技术。

domName.**innerHTML** = name  
 domPrice.**innerHTML** = price  
 domad.**innerHTML** = **description**

### 注释的重要性

无规矩不成方圆，高手的代码除了能解决各种疑难杂症，还能够让其他员工能够读懂代码，在世界500强企业中，对代码可读性，条理性，清晰性的要求比能力还重要。

我们一般通过注释来对模块注释，下满我们通过代码带讲解如何注释：

// 定义变量保存数据  
 var name = 'iphone8'  
 //定义变量保存dom元素  
 var domName = **document**.getElementById('pname')  
 var domPrice = **document**.getElementById('pprice')  
 var domad = **document**.getElementById('p-ad')  
  
 **window**.onload=function(){  
// 数据绑定  
 domName.**innerHTML** = name  
 domPrice.**innerHTML** = price  
 domad.**innerHTML** = description  
  
// 页面事件  
// 将产品添加到购物车事件  
 domadd.onclick=function(){  
 price = 7000;  
 domPrice.**innerHTML**=price;  
 }  
// 立即购买事件  
 dombuy.onclick=function(){  
 price = 7000;  
 domPrice.**innerHTML**=price;  
 }  
 }

### 上述代码存在的缺点

**可读性差：**

放在一起，如果别人拿着你的代码二次开发，读的费劲

**可维护性差：**

比较难维护，一旦需求变更，需要读懂整个代码，然后再修改。

如果你辞职了，别人接收你的代码，更改就更难了。

呢么怎么办呢，

**下面我们讲解如何使用函数来实现 – 面向过程编程**

## 面向过程编程

### 什么是面向过程编程

什么是面向过程编程：

面向过程就是使用函数将一个大的功能分成很多小功能，每个功能称之为一个过程，这就是面向过程编程。

我们可以考虑使用函数将原来写在一起的一坨的代码拆分成一个一个的独立的小功能



举例：

比如购买方法：

Function init()

{

GetData（）；

BindData();

BindEvent();

}



### 面向过程编写原则 – 单一功能原则 – 条理性编程法则

单一功能原则就是将类似功能放在一个函数里面，

好处

### 面向过程编程常用的三个封装函数和其他过程提炼方法

一般一个函数只包含某种类似的功能

我们在开发页面的时候：

一般将**数据绑定相关的代码**放在一个函数里面，取名:bindData（）

一般将**事件相关代码**放在一个函数里面，取名：bindEvents（）

一般定义一个**init函数**保存页面在初次加载的时候需要编写的代码：

我们还可以将某个功能专门用一个函数来保存，比如可以将购买产品这个功能专门放在一个函数里面，比如代码：

**window**.onload=function(){  
 init()  
}  
  
//我们一般定义一个函数init来保存页面初始化加载的时候代码  
function init() {  
 //其他可能的代码  
 bindData()  
 //其他可能的代码  
 bindEvent()  
  
}  
  
 //我们一般定义一个函数保存数据绑定相关代码  
 function bindData(){  
 domName.**innerHTML** = name  
 domPrice.**innerHTML** = price  
 domad.**innerHTML** = description  
 }  
// 我们一般定义函数封装页面或者某个模块的 事件  
function bindEvent(){  
 //将产品添加到购物车事件  
 domadd.onclick=function(){  
 AddProductToCart()  
 }  
 //立即购买事件  
 dombuy.onclick=function(){  
 buyProduct()  
 }  
}  
  
//购买产品  
function buyProduct(){  
}  
  
//将产品添加到购物车  
function AddProductToCart(){  
  
}

### 总结：单一功能原则的好处

当你们有了工作经验才能很好领悟的知识点：

灵活的应对需求变化

这里的单一功能：其实指的是一个函数只有一个引起他变化的原因。

初级开发工程师一般将所有的代码都放在一起，

Windows。Onload = function(){

XXXX

XXXX

X

X

X

X

XX

}

这样的坏处是可读性差

## 面向对象编程

### 面向对象编程定义



### 原型对象版本实现京东商城

原型对象一般用于封装组件，开发一个网页模块的模板

我们一般将我们需要开发的页面或者某个功能相关的代码都放在一个对象中统一管理

我们在开发网页的时候一般定义一个bindevent方法专门放置绑定事件的代码

我们在开发网页时候一般定义一个bindDOM或者bindData专门负责将后台的数据和前端的html进行绑定

我们一般定义一个init方法，用于某模块初次加载的时候做的事情

比较常用，我们经常需要将后台传递的字符串转换成json对象，然后绑定到前端

### Json字面量对象复习

### Json对象模式改造商城

### 原型对象和json对象总结

原型对象一般用于封装组件，封装一个功能点的属性和方法。

比如开发京东产品详细信息模块:我们定义一个产品对象，包含名称，描述，添加到购物车等属性，

## 实战 百度

## 实战 京东 pc

## 实战 移动商城