

# C语言基础

BASIC-C

DAY01

# 内容

上午	09:00 ~ 09:30	开学典礼
	09:30 ~ 10:20	
	10:30 ~ 11:20	行业概述
	11:30 ~ 12:20	课程体系介绍
下午	14:00 ~ 14:50	IOS开发环境
	15:00 ~ 15:50	
	16:00 ~ 17:00	



# 行业概述

---



# 什么是IOS



# 什么是IOS

- IOS是一款由苹果公司开发的操作系统(OS是Operating System的简称)，就像平时在电脑上用的Windows一样，都是操作系统。
- 操作系统：操作系统是一种软件，用来管理计算机软件资源和硬件资源。
- IOS与Windows的区别
  - IOS运行在移动便携式设备
  - Windows运行在PC(个人电脑)上。



# 主流手机操作系统

- 移动便携式设备
  - Android
  - IOS
  - Windows Phone
  - Symbian
  - Black Berry OS
- PC个人电脑
  - Windows
  - MacOS
  - Unix
  - Linux



# 什么是IOS开发



# 什么是IOS开发

- IOS开发就是开发运行在IOS系统上的应用或者游戏软件，比如：QQ、微博、愤怒的小鸟等...
- 2010年国内掀起移动开发热潮，但由于 IOS开发门槛要比android较高，所以相对来讲开发者并不多
- 从事IOS开发的人员称为IOS程序员或IOS软件工程师
- IOS开发可以归类到移动开发范畴





# IOS方向的优势

	iOS	android
系统架构	基于UNIX	基于Linux的虚拟机
系统安全性	安全、稳定	安全性略低、漏洞较多
开发难易度	强大的SDK、开发工具	SDK也不差
是否开源	不开源	开源
设备的种类	iPhone\iPad等	数不清的设备类型
用户人群	消费能力强的高端用户	各种用户人群
背后的BOSS	Apple	Google
应用商店	AppStore（08年上线）	GooglePlay



# 课程体系介绍

---

课程体系介绍

课程体系介绍

语言基础

iOS核心编程

iOS高级编程

游戏开发

# 课程体系介绍



# 语言基础

- C语言：了解计算机语言，掌握基本的编程方法，熟悉绝大多数计算机语言的基础知识。学会结构化的编程方式，掌握C语言的高级特性，特别是指针的使用。
- OC面向对象编程：理解什么是面向对象编程，以及这种编程思想的好处。理解什么是类，什么是对象。学会定义类和创建对象。理解和掌握面向对象的三大特征，以及OC语言对面向对象的支持。掌握OC语言的内存管理方式。
- 熟悉基础类库Foundation API，熟悉掌握使用集合对象进行对象管理，能正确处理日期数据。



# iOS核心编程

- 控制器层和控件：理解iOS的核心框架模式MVC，熟练使用各种视图控制器和一些控件进行界面编程。理解xib文件原理。
- 视图层：掌握视图中的绘制，图像处理，视图手势，视图动画，布局，以及故事板编程。
- 模型层：掌握如何处理构造应用程序模型，如何处理普通文件，属性文件等数据。



# iOS高级编程

- 数据持久与模式：了解几个必要的设计模式，理解KVC和KVO，充分理解MVC模式中的各模板之间的通信。掌握持久层通知CoreData。
- 系统服务：掌握IOS系统提供的各种系统服务，在应用开发中熟练使用这些功能。
- 多线程与多任务：理解和掌握多线程的概念，掌握使用多任务处理各种情况。
- 网络编程与云端技术：理解网络通信原理全面掌握各层级别的网络通信技术了解和使用云端技术。

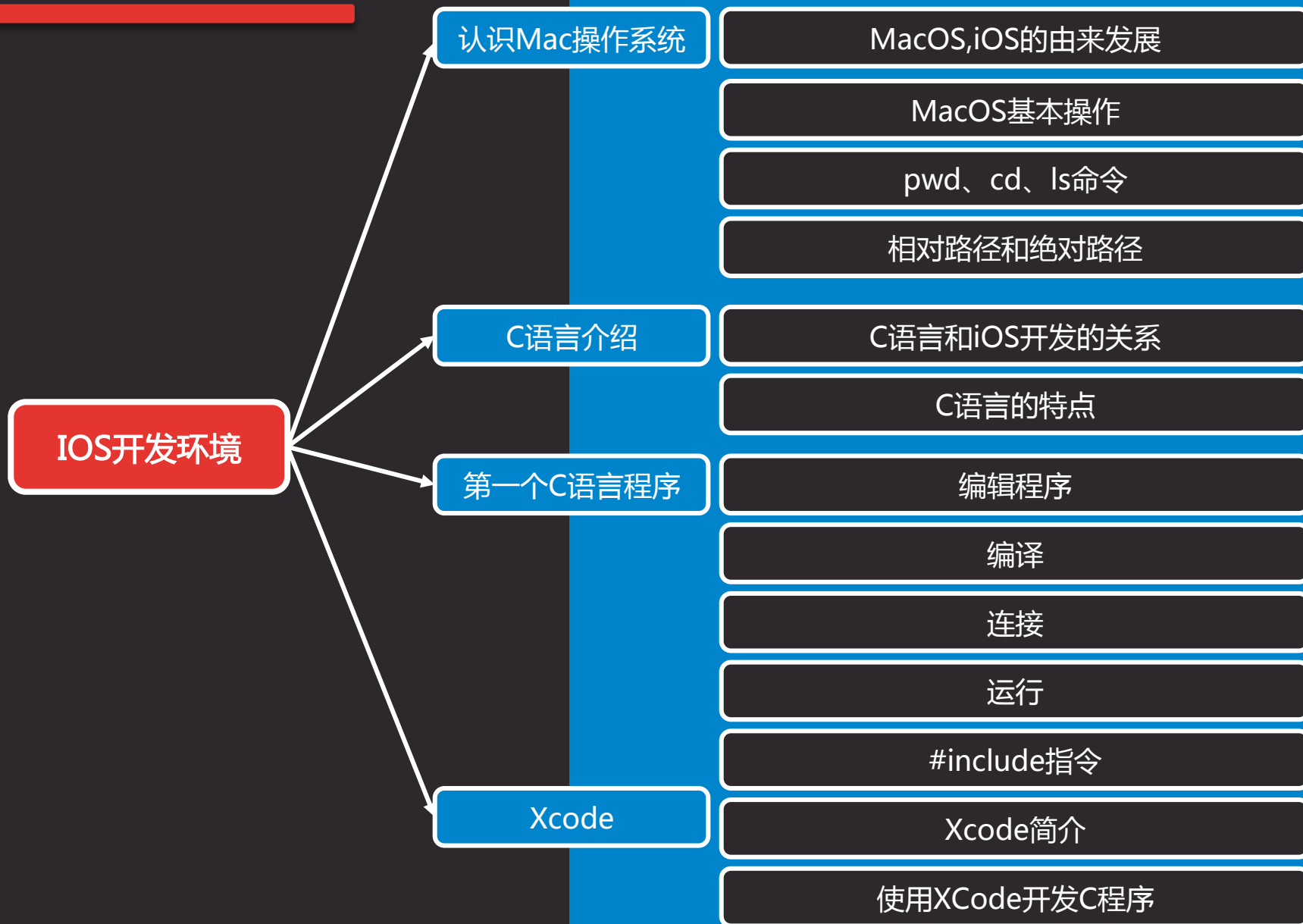


# 游戏开发

- C++：深入理解C++语言特性，掌握C++的核心语法，熟练使用C++标准库，为游戏编程打下基础。
- COCOS2D-x：全面掌握流行的游戏引擎cocos2d-x，并基于游戏引擎开发几款游戏项目。打造真正的游戏开发能力。



# IOS开发环境





# 认识Mac操作系统



# MacOS,iOS的由来及发展

- 操作系统OS：管理硬件资源和软件资源。
- MACOS:Apple公司，自我封闭的，一切都在苹果公司掌控、监督下。MACOS源于Unix系统演变而来，安全性、稳定性、性能、美观。
- IOS:苹果硬件上的一个操作系统，苹果公司便携设备上的操作系统。在2007年1月9日，MacWorld大会上公布的系统，最初设计给iphone使用的，后来陆续用到ipod touch、ipad、apple TV等苹果产品。



# MacOS基本操作

- 布局
  - a.上部 菜单栏 软件的基本设置
  - b.中部 桌面
  - c.下部 Dock 软件的快捷方式
- 查看系统的硬件配置
  - 小苹果->关于本机
- 查找文件 Finder文件管理器(我的电脑)
  - 只有一个盘符，且只有一个根。
- 软件安装
  - \*.dmg(setup.exe、install.exe)(虚拟光驱)
  - \*.app(绿色软件)
  - 直接从app store下载

# MacOS基本操作(续1)

- 输入法的切换 command+space
- Launchpad(开始->程序)
- 文本编辑(记事本) 保存一些文字信息
- 系统偏好设置(控制面板)
- MissionControl 任务控制
  - 手势控制
  - F3、control+<-或->
- \*.zip压缩文件，双击该文件就可以解决压缩
- ftp服务器->Filezilla 上传下载ftp上的文件



# pwd、cd、ls命令

- 操作计算机两种方式
  - 图形化操作方式(操作直观)
  - 命令行操作方式(操作便捷、节约资源)
- 常用命令
  - ls命令 查看当前文件夹下的文件或者子目录
  - pwd查看当前操作目录
  - cd <子目录名称> 切换子目录
  - cd.. 退回上一级目录
  - clear 清除屏幕信息
  - mkdir <文件名> 创建一个目录



# 相对路径和绝对路径

- 路径
  - 用于指明一个文件（或目录）在文件系统中的位置，路径有两种表示方式：相对路径和绝对路径。
- 相对路径
  - 相对于当前操作状态下的文件（或目录）的位置
- 绝对路径
  - 根据硬盘根目录开始，文件（或目录）的位置



# 文件夹的创建与查看

- 创建一个文件夹Test。
  - 通过图形化操作界面创建一次，通过命令行操作方式找到。
  - 通过命令行操作方式创建一次，通过图形化界面找到。



# C语言介绍





# C语言和iOS开发的关系

- C语言
  - C语言是任何编程语言的基础语言
- IOS开发
  - IOS编程开发所使用的语言是OC语言
  - OC语言是在C语言基础之上进行了加强和扩展
- 两者之间的关系
  - 学好C语言再学习OC语言，对其工作原理就会有更深的体会
  - OC语言的基础打好了，将来再来学习IOS开发就会容易的多了



# C语言的特点

- 语言背景
  - 1946年，美国 冯.诺依曼 第一台计算机  
四大部分：中央处理器(控制器、运算器)、存储器、输入设备、输出设备
- 计算机语言分类
  - 机器语言 0、1 低级语言 偏向硬件本身表达方式 效率高、难理解
  - 汇编语言 介于机器语言与高级语言之间
  - 高级语言 C、Java、OC、C# 偏向人表达方式 效率一般、易理解
- 编程：自己有一个想法，通过计算机语言使计算机理解，并帮我们实现。



# C语言的特点(续1)

- C语言：就是告诉计算机如果去操作的一种方式。
- C语言及标准化：
  - 最早由Denis Richchie于1973年为Unix设计并实现，从贝尔实验室。
  - C89 ANSI美国国家标准89年12月正式通过C99 1999年IOS通过了新版本，C语言的标准，包含了一些重要改变。



# C语言的特点(续2)

- C语言的优点
  - 高效
  - 功能强大
  - 灵活 从嵌入式至商用开发
  - 标准库(工具)
  - Unix、Linux、MAC系统集成C语言的开发环境
- C语言的缺点
  - 有时难以理解
  - 有时易隐藏错误
  - 有时会难修改



# 第一个C语言程序



# 编辑程序

- 编写C语言的代码，保存在扩展名.c文件中，源文件。
- 编写代码有三种方式：
  - vi命令行方式 系统自带
  - UltraEdit 网络下载
  - xcode 网络下载



# 编译

- 将C语言代码转换成计算机识别的代码(二进制代码) 目标文件（可执行文件）
- GNU C Compiler，GNU自由软件基金会的项目，C语言，Compiler编译。
- 编译C语言、C++、Java、ObjectiveC



# 连接

- 经过编译后生成的目标文件是不能直接执行的，它需要经过连接之后才能生成可执行的代码。连接后得到的可执行文件都是以“ .out” 为其文件的扩展名。





# 运行

- 经过编译、连接之后，源程序文件就可以生成可执行的文件。
- 默认情况下，编译后的文件名叫做a.out文件
- 通过命令"./" 运行编译后的文件(a.out)。



# 原样显示以下内容到屏幕中

a.       \*\*\*\*\*  
              \*\*\*  
              \*

b.       \*  
          \*\*\*  
         \*\*\*\*\*

c.       \*\*\*\*\*       \*  
          \*\*\*       \*\*\*  
             \*       \*\*\*\*\*



# 原样显示以下内容到屏幕中(续1)

## 输出个人信息

姓名: zhangsan

性别: M

年龄: 18

期望薪资: 6000

## 模拟登录屏幕信息

=====

系统登录

用户名: zhangsan

密        码: \*\*\*

=====



# #include指令

- 在编译之前做的事情
- 相当于复制操作 不推荐在一个文件中包含另一个.c文件
- 加载系统提供的标准函数库
  - #include <stdio.h>
  - stdio.h 标准输入输出功能
  - standard input output (printf)



# #include指令 ( 续1 )

- gcc -E \*.c 查看编译前文件内容
- 在使用#include指令的时候，可以有两种方式
  - a.<>和 "" 两种格式
  - b.<>是加载系统功能，可以在usr/include中得到相应文件
  - c.""是加载自定义功能，优先查找当前文件夹下的文件，再去查看C函数库中的文件



# Xcode



# Xcode简介

- Xcode是苹果公司向开发人员提供的集成开发环境（非开源），用于开发Mac OS X,iOS的应用程序。其运行于苹果公司的Mac操作系统下。
- 不管你用C、C++、Objective-C或Java编写程序，在AppleScript里编写脚本，还是试图从另一个奇妙的工具中转移编码，你会发现 Xcode 编译速度极快。每次操作都很快速和轻松。
- 苹果公司为用户提供了全套免费的Cocoa程序开发工具(Xcode)。Mac OS X一起发行，在苹果公司官方的网站下载。



# 使用XCode开发C程序

- 操作步骤：
  - 创建一个新的project
  - 选择项目类型  
osx->application->Command Line Tool
  - 配置项目相关属性
    - Product Name 项目名称
    - Organization Name 组织名称
    - Company Identifier 公司标识
    - type 编写代码方式





# 使用Xcode开发C语言应用程序

将前面中的练习题，使用命令行的方式编写C语言程序，换成使用Xcode来写C语言程序。



# 总结和答疑

