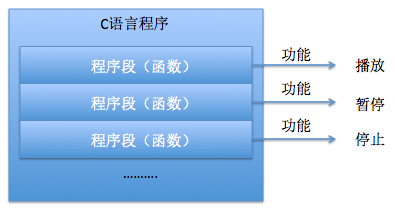
# 什么是函数

* 任何一个C语言程序都是由一个或者多个程序段（小程序）构成的，每个程序段都有自己的功能，我们一般称这些程序段为“函数”。所以，你可以说C语言程序是由函数构成的。
* 比如你用C语言编写了一个MP3播放器程序，那么它的程序结构如下图所示：



# 函数的定义

## 定义函数的目的

* 将一个常用的功能封装起来，方便以后调用

## 定义函数的步骤

* 函数名：函数叫什么名字
* 函数体：函数是干啥的，里面包含了什么代码

## 格式

* 固定格式（很多语言的函数都是这么写的）

返回值类型 函数名(形式参数列表)

{

函数体

}

* 举例

定义一个函数，计算两个整数的和

sum(int a, int b)

{

int c = a + b;

}

# 函数调用

* sum(10, 11); 引出返回值
* 说明函数的调用过程
* 简述return的作用

# 函数的参数

1. 形参和实参的基本概念
2. 形参个数和实参一致：sum(10, 11, 12)
3. 参数的传递是值传递
4. 参数名不能跟函数内的局部变量同名
5. 函数可以没有参数：设计一个函数返回PI

# 函数的返回值

1. 返回值的基本概念、return的作用
2. void
3. return

* void可以省略return
* 可以多次使用return
* return后面不能有其他语句

1. 函数的弱语法

* 如果没有写返回值类型，默认是int
* 如果写了返回值，可以不返回
* 调用一个没有定义过的函数

# 定义函数的步骤

1. 明确函数作用，起一个有意义的函数名称
2. 明确函数的参数和返回值
3. 举例：

* 求两个整数的和
* 打印一条横线
* 打印N条横线

# 函数注意

* 不能嵌套定义函数
* 死循环调用，自己调用自己
* 不能重复定义、可以重复声明

# 函数的补充

## main函数

* 返回值：0，正常退出；1，异常退出

## printf函数

* #include
* 返回值：字符串的长度

# 习题

1. 编写一个函数double avr(int a, int b)，计算a和b的平均值
2. 编写一个函数int pows(int base, int n)返回base的n次方
3. 编写一个函数int pieAdd(int n)，计算1+2+3+……+n的值并返回。比如pieAdd(3)的返回值是1+2+3=6

# 函数的声明

1. 函数的定义顺序
2. 函数的声明作用：声明和定义，类似于身份证和人的关系，编译买机票、链接登机
3. 只有函数声明、没有定义，编译警告、链接报错

# .h文件和.c文件的分工

1. 单文件的坏处

* 一个文件的内容太多，不利于阅读、调试
* 多人修改同一个文件出问题
* 公司里面是团队合作

1. 将sum函数抽取到另一个.c文件中

* 先直接调用sum函数，编译警告、链接main.c错误
* #include “sum.c” ， 编译链接main.c，运行成功（画图分析.o文件中的东西）
* 如果avr.c中使用sum.c，编译链接main.c，运行程序（画图分析出错原因）

1. 在其他文件中声明sum函数

* int sum(int,int);
* 编译链接main.c
* 编译链接sum.c
* 编译链接main.c sum.c，运行成功
* avr.c利用一样的方法

1. 将int sum(int,int)抽取到另外一个文件中

* 不抽取声明的坏处：增加新的函数了
* 抽取到.c文件中？开发工具会将所有的.c进行编译链接
* 抽取到.h文件中

1. 抽取步骤总结

* .c文件写函数的定义
* .h文件写函数的声明
* 要想用我的函数，请包含我的.h文件
* 好处分析

# #include

## 预处理指令简介

## #include的作用

纯粹是文件拷贝

## #include <stdio.h>

* stdio.h中有什么？链接的时候做了什么事情
* <>和””的区别

## #include的路径问题

* 默认是同路径，其他问题以后再讨论