# C语言介绍

## 语言详细介绍：

|  |
| --- |
| **计算机底层数据的保存格式都是二进制(0/1)** |

## 开发工具介绍:

|  |
| --- |
|  |

# C语言的应用场景：跟硬件打交道的程序都可以使用c语言开发，比如设备的驱动、操作系统等

# 位、字节、排列组合

## 什么是位(bit)？

|  |
| --- |
|  |

## 什么是字节？

|  |
| --- |
| **1个字节=8位** |
| **字节与位的关系：1Byte(B)=8bit，总共有256中排列组合** |
| **字节与KB的关系：1KB=1024Byte** |
| **1M=1024KB** |
| **1G=1024M** |
| **1T=1024G** |

## 排列组合：

|  |
| --- |
| **n个位共有：2n种不同排列组合** |
| **题目：**  **1G=?bit---->1024\*1024\*1024\*8bit**  **20M=?Byte---->20\*1024\*1024B**  **2T=?bit---->2\*1024\*1024\*1024\*1024\*8bit**  **4Byte总共有多少种排列组合？--->232**  **2Byte总共有多少中排列组合？--->216** |
|  |

# 第一个C语言程序(\*)

|  |
| --- |
|  |

# 数据类型与变量的声明(\*)

## 常见的数据类型：

|  |
| --- |
| **什么是数据类型？常见的数据格式有：视频、音频、图片、数字、字符** |
| **通过程序来查看C语言中的数据类型长度：sizeof(数据类型)--->返回类型长度** |

## 变量的声明语法：

|  |
| --- |
| **(1)数据类型 变量名 = 变量值;** |
| 1. **数据类型 变量名;**   **变量名=变量值;** |
|  |
| **常量：** |

## 变量命名的语法规则(\*)：

|  |
| --- |
| **变量与常量在命名时，可以包含数字、字母、下划线，但是不能够以数字开头；同时变量在取名字时需要避开系统的关键字。** |

# 打印之占位符

|  |
| --- |
|  |

# 算术运算符与算术表达式

## 常用的算术运算符

|  |
| --- |
| **+（加法）、-(减法)、\*(乘法)、/(除法)、取余(%)、自增(++)、自减(--)** |
|  |

## 算术表达式：通过算术运算符连接起来的式子称之为算术表达式

# 关系运算符(比较运算符)与关系表达式

|  |
| --- |
| **>(大于)、>=(大于等于)、<(小于)、<=(小于等于)、==(等于)、不等于(!=)** |
| **语法规则：数据1 关系运算符 数据2---->返回值要么为1要么为0** |

# 逻辑运算符与逻辑表达式

|  |
| --- |
| **常用的逻辑运算符：与(&&)、或(||)、非(!)**  **语法规则：**   1. **条件1 && 条件2：当条件1与条件2同时为真(1)，最终结果才为真；否则为0(假)** 2. **条件1 || 条件2：当条件1与条件2同时为假(0)，最终结果才为假;条件1与条件2只要有一个为真，最终结果就为真** 3. **!条件：如果条件为1，则结果为0；如果条件为0，则结果为1** |

# 键盘录入

|  |
| --- |
|  |

# 选择结构(\*)

|  |
| --- |
| **第一种：if选择结构,语法规则：if(条件){代码块}，如果条件为真(1)则执行代码块，否则不执行** |
| **第二种：if...else选择结构，语法规则:if(条件){代码块1}else{代码2}，如果条件为1则执行代码块1，否则执行代码块2** |
| **第三种结构：else ...if** |
| **第四种：switch结构** |

# 循环结构(\*)

|  |
| --- |
| **第一种循环结构：while**  **语法规则:while(条件){代码块;循环结束条件;},永远是先判断条件是否为真，如果为真则执行代码块。**  **不可达代码：在死循环后面的代码** |
| **带终止条件的while循环(\*)：** |
| **带break的while循环：break可以提前终止while循环，不能够终止整个程序** |
| **带continue的while循环：跳过本次循环** |
| **第二种循环结构：do...while**  **语法规则：do{代码块;终止条件}while(条件);** |
| **第三种循环结构：for循环(\*)**  **语法规则：for(初识化表达式;循环条件;条件终止表达式){代码块;}**    **其他说明：for循环中可以使用break与continue**  **for循环中可以同时有多个初始化表达式：**    **for循环中的死循环：**   1. **：3个表达式都不写**      1. **省略终止条件表达式**      1. **省略掉条件表达式** |

# 变量的作用范围(\*)：

|  |
| --- |
|  |

# 打印与录入字符

|  |
| --- |
| **打印字符：**    **录入字符：** |

# 嵌套for循环(\*)

# 数组（\*）

|  |
| --- |
| **数组的语法规则(先声明，再赋值)：数据类型 数组名[数组长度];**  **int attr1[3];**  **attr1[0]=20;**  **attr1[1]=30;**  **attr1[2]=40;**  **printf("attr1[0]=%d\n",attr1[2]);**  **数组的语法规则(边声明边赋值)：数据类型 数组名[数组长度]={数值1,数值2,数值3...};**  **int attr2[3]={1,2,3};**  **for(int i=0;i<=2;i++){**  **printf("attr2[%d]=%d\n",i,attr2[i]);**  **}** |
| **计算数据类型的长度：sizeOf(数据类型/数组名):返回的是字节长度** |
| **找出数组中的最大值：打擂台**  **int attr1[4];**  **printf("请输入4家店的价格：\n");**  **for(int i=1;i<=4;i++){**  **printf("第%d家店的价格:",i);**  **scanf("%d",&attr1[i-1]);**  **}**  **//找最大值**  **int max=attr1[0];**  **for(int j=1;j<=3;j++){**  **if(max<attr1[j]){**  **max=attr1[j];**  **}**  **}**  **printf("最大值：%d\n",max);** |
| **找最小值：**  **//最小值**  **int min=attr1[0];**  **for(int k=1;k<=3;k++){**  **if(min>attr1[k]){**  **min=attr1[k];**  **}**  **}**  **printf("最小值：%d",min);** |

# 键盘录入字符串

|  |
| --- |
| char ch[10];  printf("请输入一个字符串:");  scanf("%s",&ch);  printf("录入的字符串为：%s\n",ch);  //遍历char类型的数组  for(int i=1;i<=10;i++){//乱码  printf("%c\t",ch[i-1]);  } |

# 排序(冒泡排序)(\*)

|  |
| --- |
|  |

# 函数:

|  |
| --- |
| **定义函数的基本语法规则：返回值类型 函数名(参数类型1 参数名1,参数类型2 参数名2...){函数体;}** |