# 目录

[目录 1](#_Toc1400435)

[第3章 基本语句 2](#_Toc1400436)

[3.1 基本语句概述 2](#_Toc1400437)

[3.2 顺序语句 2](#_Toc1400438)

[3.3 选择语句 2](#_Toc1400439)

[3.3.1 if结构 2](#_Toc1400440)

[3.3.2 switch结构 2](#_Toc1400441)

[3.4 循环语句 2](#_Toc1400442)

[3.4.1 for结构 2](#_Toc1400443)

[3.4.2 while结构 2](#_Toc1400444)

[3.4.3 do……while结构 2](#_Toc1400445)

[3.5 跳转语句 2](#_Toc1400446)

[3.5.1 break结构 2](#_Toc1400447)

[3.5.2 continue结构 2](#_Toc1400448)

[3.5.3 return结构 2](#_Toc1400449)

[3.5.4 goto结构 2](#_Toc1400450)

# 基本语句

## 基本语句概述

基本语句：C语言中算法的基础 冒泡排序(for语句嵌套)

结构化编程：

三种结构+4条跳转语句 🡪 9条语句

顺序结构：所有的语句都是顺序结构

选择结构：2条

1. 单种选择：if语句
2. 两种选择：if…else语句
3. 多种选择：if…else if…else语句

swtich语句

循环结构：3条

1. for语句 循环次数已知
2. while()语句 循环次数未知
3. do…while()语句 基本不用

跳转语句：4条

1. break语句：跳出循当前环体和跳出switch语句
2. continue语句：结束本次循环继续下一次循环
3. return语句：结束函数和返回一个值
4. go to语句：无条件转移(同一个函数中) 少用

## 顺序结构

## 选择结构

### if语句

也称为单分支结构

有三种形式

第一种

|  |
| --- |
| if(表达式)  {  代码段;//可以是多行代码  }  原理：当表达式非零值时，就执行代码段，否则跳过if语句  表达式：常量、变量、表达式  注意：可以不执行if语句  如果代码段的语句只有一条，可以省略大括号  省略了大括号，默认作用的范围最近的一条 |

第二种

|  |
| --- |
| if(表达式)  {  代码段1;  }else  {  代码段2;  }  原理：当表达式成立，执行代码段1，否则执行代码段2；  注意：必须选择一个代码段执行 |

第三种

|  |
| --- |
| if(表达式1)  {  代码段1;  }else if(表达式2)  {  代码段2;  }else  {  代码段3;  }  原理：当表达式1是非零值时，执行代码段1，否则判断表达式2是否非零值  如果表达式2是非零值时，执行代码段2，否则执行代码段3  注意：三种选择必须选一个执行  if else满足就近原则 |

课堂练习：

100制分数等级

小于60 输出 E

60到70 输出D

70到80 输出C

80 到 90 输出 B

90 到100 输出 A

|  |
| --- |
| int main(void)  {  int score = 0;  printf("请输入一个分数:");  scanf("%d", &score);  if(score >= 90)  {  printf("A\r\n");  }else if(score >= 80)  {  printf("B\r\n");  }else if(score >= 70)  {  printf("C\r\n");  }else if(score >= 60)  {  printf("D\r\n");  }else  {  printf("E\r\n");  }  return 0;  } |

### switch语句

也称为多分支结构

标准形式

|  |
| --- |
| switch(条件)  {  case 条件表达式1：代码段1；break;  case 条件表达式2：代码段2；break;  case 条件表达式3：代码段3；break;  case 条件表达式4：代码段4；break;  default：代码段5；break;通常用来提示错误 前面的条件与条件表达式都不匹配，才会执行这条代码  }  原理：当条件与条件表达式1匹配，执行代码段1，接着执行break语句跳出switch语句。  当条件与条件表达式2匹配，执行代码段2，接着执行break语句跳出switch语句。  当条件与条件表达式3匹配，执行代码段3，接着执行break语句跳出switch语句。  当条件与条件表达式4匹配，执行代码段4，接着执行break语句跳出switch语句。  当条件与条件表达式都不匹配，执行代码段5，接着执行break语句跳出switch语句。  条件：整型常量、整型变量、字符型常量、字符型变量 （不能浮点型）  条件表达式：字符型常量、整型常量 |

不标准形式

|  |
| --- |
| switch(条件)  {  case 条件表达式1：代码段1；  case 条件表达式2：代码段2；  case 条件表达式3：代码段3；  case 条件表达式4：代码段4；  default：代码段5；通常用来提示错误  }  原理：当条件与条件表达式1匹配时，执行代码段1，接着执行代码段2、代码段3、代码段4、代码段5  当条件与条件表达式2匹配时，执行代码段2，接着代码段3、代码段4、代码段5  当条件与条件表达式3匹配时，执行代码段3，接着代码段4、代码段5  当条件与条件表达式4匹配时，执行代码段4，接着代码段5  当条件与条件表达式都不匹配时，执行代码段5 |

什么情况下用if语句什么情况下用switch语句？

条件简单：switch语句

条件复杂：if语句

课堂练习：

今天是今年的星期几？

打印出今年日历？

## 循环结构

重复性的工作，减少工程师的代码量

### for语句

标准结构

|  |
| --- |
| for(循环变量初始化; 循环条件; 改变循环变量)  {  循环体;  }  原理：1. 先执行循环变量初始化(只会执行一次)  2. 判断循环条件是否成立  3. 如果成立，接着循环体，回到改变循环变量，接着回到步骤2  4. 如果不成立，跳出for语句 |

不标准结构

|  |
| --- |
| for(; 循环条件; 改变循环变量)//省略了循环变量初始化  {  循环体;  }  注意：循环变量要有初值  for(; 循环条件;)//省略了循环变量初始化  {  循环体;  改变循环变量;  }  for(;;)  {  循环体;  }  死循环 |

循环嵌套：循环里面有循环

课堂练习：

1. 99乘法表
2. 打印星号

|  |
| --- |
| \*\*\*\*\*\*\* 7  \*\*\*\*\* 5  \*\*\* 3  \* 1 |

### while语句

格式

|  |
| --- |
| while(循环条件)  {  循环体;  改变循环变量；  }  原理：1. 判断循环条件是否成立  2. 如果成立，执行循环体，接着执行改变循环变量，接着回到第一步  3. 如果不成立，跳过while语句  While(1)  {  循环体  }  //标准死循环 |

### do……while语句

格式

|  |
| --- |
| do  {  循环体;  改变循环变量；  }while(循环条件);  原理：1. 先执行循环体、改变循环变量  2. 判断循环条件是否成立  3. 成立，回到1  4.不成立，跳出do…while语句 |

while语句和do…while语句

while语句：可以一次都不执行

do…while语句：至少执行一次

for语句和while语句

for语句：循环次数已知

while语句：循环次数未知

输入一个数，判断位数？

## 跳转语句

### break语句

作用：跳出当前的循环体和跳出switch语句

|  |
| --- |
| int main(void)  {  int i = 0;  while(1)  {  if(i > 5)  {  while(1)  {  break;  }  }  printf("%d\r\n", i);//0 1 2 3 4 5  i++;//1 2 3 4 5 6  }  return 0;  } |

### continue语句

作用：结束本次循环继续下一次循环

注意：不能跳出循环

|  |
| --- |
| int main(void)  {  int i = 0;  while(i < 10)  {  i++;//1 2 3 4 5 10  if(i < 5)  {  continue;  }  printf("%d\r\n", i);//5 6 7 8 9 10  }  return 0;  } |

### return语句

作用：结束函数和返回一个值

|  |
| --- |
| int main(void)  {  int i = 0;  while(i < 10)  {  i++;//1 2 3 4 5 10  if(i < 5)  {  return 5;  }  printf("%d\r\n", i);//5 6 7 8 9 10  }  printf("1\r\n");  return 0;  } |

### goto语句

作用：无条件转移

注意：只能在同一个函数中使用

|  |
| --- |
| int main(void)  {  char str[10] = {0};  printf("请输入密码:");  scanf("%s", str);  if(strcmp( str, "666666" ) == 0)  {  printf("密码正确\r\n");  }else  {  printf("密码错误\r\n");  goto nx;  }  printf("6666666");  return 0;  }  void add(void)  {  int a = 5;  int b = 10;  nx: //错误  printf("%d\r\n", a+b);  } |