

结构语句：

- (1) 顺序结构：从前到后依次执行程序中的每条语句，且每条语句都只执行一次
- (2) 选择结构：从前向后依次执行程序中的语句，但根据满足的条件不同，执行不同的语句模块，并且不会执行所有语句
- (3) 循环结构：从前向后依次执行程序中的语句，若满足条件，则一直执行该语句，直到不满足条件为止

一、选择结构

1、if语句

(1) 基本结构

```
if(判断条件)
{
    语句块1;//如果判断条件为真，则执行语句块1;
}
else
{
    语句块2;//如果判断条件为假，则执行语句块2;
}
```

(2) 多分支结构

```
if(判断条件1)
{
    语句块1;//如果判断条件1为真，则执行语句块1;
}
else if(判断条件2)
{
    语句块2;//如果判断条件1为假，2为真，则执行语句块2;
}
else if(.....)
{
    .....
}
else
{
    语句块n;//如果之前所有判断条件都为假，则执行语句块n;
}
```

(3) 嵌套结构

```
if(判断条件1)
```

```

{
if(判断条件2)
{
语句块1;//如果判断条件1,2为真，则执行语句块1;
}
else
{
语句块2;//如果判断条件1为真,2为假，则执行语句块2;
}
}
else
{
if(判断条件3)
{
语句块3;//如果判断条件1为假,3为真，则执行语句块3;
}
else
{
语句块4;//如果判断条件1,3为假，则执行语句块4;
}
}
}

```

注：①可以只有if语句，没有else if语句或else语句；可以有if语句和else if语句，没有else语句；不能只有else if语句或else语句，没有if语句

② 多分支结构，所有条件必须依次判断，不能跳过

③ if语句与else语句或与else if语句，或else if语句与else语句之间不能出现其他语句

④ 所有语句块必须使用{.}

⑤ 语句块必须缩进

2、switch结构

经过计算已知结果进行匹配

格式：

switch(常量/具有确定值的变量或表达式)

```

{
case 选项1:
语句块1;break;
case 选项2:

```

```
语句块2;break;
```

```
....
```

```
default:
```

```
语句块n;break;
```

```
}
```

注：① **switch**后的括号中，一定是确定值

② **case** 选项可以是整型或字符型

③ 所有的**case**选项与**default**选项都是并列的，没有先后顺序之分

④ 语句块后的**break**表示当前选项已经执行完毕，结束**switch**匹配。**break**是否添加视实际情况而定

例：从终端输入数字1-7，分别输出对应的星期

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    //输入num
```

```
    int num = 0;
```

```
    printf("输入num: ");
```

```
    scanf("%d",&num);
```

```
    switch(num)
```

```
    {
```

```
        case 1:
```

```
            printf("星期一\n");break;
```

```
        case 2:
```

```
            printf("星期二\n");break;
```

```
        case 3:
```

```
            printf("星期三\n");break;
```

```
        case 4:
```

```
            printf("星期四\n");break;
```

```
        case 5:
```

```
            printf("星期五\n");break;
```

```
        case 6:
```

```
            printf("星期六\n");break;
```

```
        case 7:
```

```
            printf("星期日\n");break;
```

```
        default:
```

```
            printf("数字错误\n");break;
```

```
}
```

```
return 0;
```

```
}
```