

C++语法基础(1)

——程序的基本结构

第1课

基本介绍

- 洛谷平台注册、登录
- Dev C++软件介绍
- C++程序基本语法结构
- 顺序结构：变量、数据类型、格式化输入输出……
- 分支结构：if-else, switch, 关系运算、逻辑运算
-

- 洛谷平台: <https://www.luogu.com.cn/>
- 注册: 电子邮件/手机号

第一部分

C++基本语法

——顺序结构

- 知识点1：C++程序基本结构
 - 知识点2：变量与数据类型
 - 知识点3：赋值语句与变量自增
 - 知识点4：数学运算
 - 知识点5：输入输出语句
-
- 练习1：字符三角形（B2005）
 - 练习2：数字反转（P5705）
 - 练习3：大象喝水（B1019）

C++程序基本结构

- 引入头文件
 - #include<iostream>
 - #include<bits/stdc++.h>
- 标准命名空间
 - using namespace std;
- 主函数
 - int main()
{
 程序主体;
 //return 0;
}

注释:

- // 行注释
- /*
 段落注释
*/

```
//C++程序结构

#include<iostream>           //声明头文件



using namespace std;        //命名空间, 注意用英文分号“ ; ” 结尾

int main(){                 //声明主函数 , 用“ {} ” 将程序主体包括起来

    ... .. ;                //程序每句结尾用英文分号“ ; ” 结尾 , 如果输入正确, 分号是红色的
    //编程的程序主体
    ... .. ;

    return 0;               //程序主函数结束
}                            //“ } ” 在 程序最后一句之后
```

Dev C++基本操作

- 新建源代码
 - 快捷键: ctrl+N
- 编写程序
- 保存
 - 快捷键: ctrl+S
- 编译
 - 
- 运行
 - 
- 编译+运行
 - 

变量

- 变量是在程序执行过程中，其值可以改变的量。
- 变量其实是某个有符号名（也成为变量名）的存储位置（内存地址），该位置上存放着描述某种信息的量（称为变量的值）。
- 声明变量：
 - 数据类型 变量名；
 - 例如：int a;
- 变量名：
 - 变量名中只能出现字母、数字或下划线；
 - 不能以数字开头，例如2Server不是一个合法的C++变量；
 - 不能是C++的关键字。所谓关键字，就是C++中已经定义好的有特殊含义的单词；
 - 区分大小写，例如，A1和a1是两个不同的变量。

数据类型

数据类型	含义	占用空间	取值范围
int	整型	4字节 (32位)	$-2^{31} \sim 2^{31} - 1$ -2147483648 ~ 2147483647
long long	长整型	8字节 (64位)	$-2^{63} \sim 2^{63} - 1$
float	单精度浮点数	4字节 (32位)	6-7位有效数字
double	双精度浮点数	8字节 (64位)	15-16位有效数字
char	单字符	1字节 (8位)	ASCII码范围 (0-127)
bool	布尔类型	1字节 (8位)	0(false) 或 1(true)

字符型与整型的转换

- 根据ASCII码, 'A'-65, 'a'-97, '0'-48。
- 如何把小写字母转换成大写字母?
- 如何把字符数字转换成对应的数值?

The ASCII code

American Standard Code for Information Interchange

ASCII control characters			
DEC	HEX	Simbolo ASCII	
00	00h	NULL	(carácter nulo)
01	01h	SOH	(inicio encabezado)
02	02h	STX	(inicio texto)
03	03h	ETX	(fin de texto)
04	04h	EOT	(fin transmisión)
05	05h	ENQ	(enquiry)
06	06h	ACK	(acknowledgement)
07	07h	BEL	(timbre)
08	08h	BS	(retroceso)
09	09h	HT	(tab horizontal)
10	0Ah	LF	(salto de línea)
11	0Bh	VT	(tab vertical)
12	0Ch	FF	(form feed)
13	0Dh	CR	(retorno de carro)
14	0Eh	SO	(shift Out)
15	0Fh	SI	(shift In)
16	10h	DLE	(data link escape)
17	11h	DC1	(device control 1)
18	12h	DC2	(device control 2)
19	13h	DC3	(device control 3)
20	14h	DC4	(device control 4)
21	15h	NAK	(negative acknowle.)
22	16h	SYN	(synchronous idle)
23	17h	ETB	(end of trans. block)
24	18h	CAN	(cancel)
25	19h	EM	(end of medium)
26	1Ah	SUB	(substitute)
27	1Bh	ESC	(escape)
28	1Ch	FS	(file separator)
29	1Dh	GS	(group separator)
30	1Eh	RS	(record separator)
31	1Fh	US	(unit separator)
127	20h	DEL	(delete)

ASCII printable characters											
DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
32	20h	espacio	64	40h	@	96	60h	`			
33	21h	!	65	41h	A	97	61h	a			
34	22h	"	66	42h	B	98	62h	b			
35	23h	#	67	43h	C	99	63h	c			
36	24h	\$	68	44h	D	100	64h	d			
37	25h	%	69	45h	E	101	65h	e			
38	26h	&	70	46h	F	102	66h	f			
39	27h	'	71	47h	G	103	67h	g			
40	28h	(72	48h	H	104	68h	h			
41	29h)	73	49h	I	105	69h	i			
42	2Ah	*	74	4Ah	J	106	6Ah	j			
43	2Bh	+	75	4Bh	K	107	6Bh	k			
44	2Ch	,	76	4Ch	L	108	6Ch	l			
45	2Dh	-	77	4Dh	M	109	6Dh	m			
46	2Eh	.	78	4Eh	N	110	6Eh	n			
47	2Fh	/	79	4Fh	O	111	6Fh	o			
48	30h	0	80	50h	P	112	70h	p			
49	31h	1	81	51h	Q	113	71h	q			
50	32h	2	82	52h	R	114	72h	r			
51	33h	3	83	53h	S	115	73h	s			
52	34h	4	84	54h	T	116	74h	t			
53	35h	5	85	55h	U	117	75h	u			
54	36h	6	86	56h	V	118	76h	v			
55	37h	7	87	57h	W	119	77h	w			
56	38h	8	88	58h	X	120	78h	x			
57	39h	9	89	59h	Y	121	79h	y			
58	3Ah	:	90	5Ah	Z	122	7Ah	z			
59	3Bh	;	91	5Bh	[123	7Bh	{			
60	3Ch	<	92	5Ch	\	124	7Ch				
61	3Dh	=	93	5Dh]	125	7Dh	}			
62	3Eh	>	94	5Eh	^	126	7Eh	~			
63	3Fh	?	95	5Fh	_						

Extended ASCII characters											
DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo	DEC	HEX	Simbolo
128	80h	Ç	160	A0h	á	192	C0h	Ł	224	E0h	Ó
129	81h	ü	161	A1h	í	193	C1h	ł	225	E1h	ô
130	82h	é	162	A2h	ó	194	C2h	Ł	226	E2h	ö
131	83h	â	163	A3h	ú	195	C3h	ł	227	E3h	õ
132	84h	ä	164	A4h	ñ	196	C4h	Ł	228	E4h	ö
133	85h	à	165	A5h	Ñ	197	C5h	ł	229	E5h	Ö
134	86h	á	166	A6h	ª	198	C6h	Ł	230	E6h	µ
135	87h	ç	167	A7h	º	199	C7h	Ł	231	E7h	þ
136	88h	ê	168	A8h	¿	200	C8h	Ł	232	E8h	ß
137	89h	ë	169	A9h	®	201	C9h	ł	233	E9h	Û
138	8Ah	è	170	AAh	¬	202	CAh	Ł	234	EAh	Ü
139	8Bh	ï	171	ABh	½	203	CBh	ł	235	EBh	Ù
140	8Ch	ì	172	ACH	¼	204	CCh	Ł	236	ECh	Ý
141	8Dh	í	173	ADh	¾	205	CDh	ł	237	EDh	Ý
142	8Eh	Ä	174	Aeh	«	206	CEh	Ł	238	Eeh	·
143	8Fh	Å	175	Afh	»	207	CFh	ł	239	Efh	·
144	90h	É	176	B0h	⋮	208	D0h	Ł	240	F0h	±
145	91h	æ	177	B1h	⋮	209	D1h	ł	241	F1h	±
146	92h	Æ	178	B2h	⋮	210	D2h	Ł	242	F2h	¼
147	93h	ô	179	B3h	⋮	211	D3h	ł	243	F3h	¼
148	94h	ò	180	B4h	⋮	212	D4h	Ł	244	F4h	½
149	95h	ò	181	B5h	⋮	213	D5h	ł	245	F5h	¾
150	96h	û	182	B6h	⋮	214	D6h	Ł	246	F6h	÷
151	97h	ù	183	B7h	⋮	215	D7h	ł	247	F7h	÷
152	98h	ÿ	184	B8h	⋮	216	D8h	Ł	248	F8h	°
153	99h	Ö	185	B9h	⋮	217	D9h	ł	249	F9h	°
154	9Ah	Ü	186	BAh	⋮	218	DAh	Ł	250	FAh	°
155	9Bh	ø	187	Bbh	⋮	219	DBh	ł	251	FBh	°
156	9Ch	£	188	BCh	⋮	220	DCh	Ł	252	FCh	°
157	9Dh	Ø	189	BDh	⋮	221	DDh	ł	253	FDh	°
158	9Eh	×	190	BEh	⋮	222	DEh	Ł	254	FEh	°
159	9Fh	f	191	Bfh	⋮	223	DFh	ł	255	FFh	°

赋值语句

- 赋值符号：‘=’
- 形如 $a = b$ 的语句是基本赋值语句，其作用是将b的值赋值给a，b可以是常量、变量或表达式。
- 赋值的运算顺序是**从右到左**，将右边的运算结果存入左边的变量中。
- 赋值具有**覆盖性**，新存入的数据会覆盖原有的数据。

```
int x,y,a,t  
x=5;  
y=8;  
a=x+y;  
x=x+y;  
y=x-y;
```

变量自增与自减

- 自增运算符“++”用于将整数型或浮点型变量的值增加1
 - `a++`与`a=a+1`的效果相同
- 自减运算符“--”用于将整数型或浮点型变量的值减1
- “++”出现在变量前面和后面是有区别的
 - `a++;` 和 `++a;` 没有区别
 - `b=a++;` 和 `b=++a` 有区别
 - ++在变量前，先进行自增操作，再执行本语句；
 - ++在变量后，先执行本语句，再做自增操作。

数学运算

- C++中的数学符号
- /的注意事项
 - 如果除法前后都是int数据类型，结果也是int型，不会出现小数。
 - 通过乘上小数，可以使结果变成小数，例如 $1.0*5/3$
- %的用法
 - 取余
 - 例如： $5\%3$ 结果是2

符号	对应的含义
=	赋值
+	加
-	减
*	乘
/	除
%	取余

输出语句

- cout
 - `cout<<"hello world"<<endl;`
 - `cout<<"hello"<<" "<<"world";`
 - `cout<<7+8;`
 - `cout<<"7+8";`
 - `cout<<a;`
- printf标准输出语句
 - `printf("%d # %c",a,b);`
 - 输出什么?
 - `printf("%.3f\n",a);`
 - 输出什么?

输入语句

- cin
 - cin>>a;
 - cin>>a>>b;
- scanf
 - scanf("%c%c.%c",&a,&b,&c);

- 练习1：字符三角形（洛谷）

- 题号：B2005

- 难度：入门

- 考点：输入输出练习

- 练习2：反向输出一个三位数
 - 题号：B2028
 - 难度：入门
 - 考点：/、%的用法

- 练习3：浮点数反转

- 题号： P5705

- 难度： 入门

- 考点： 格式化输入、输出函数- scanf(),printf()

- 练习4：大象喝水

- 题号：B2029

- 难度：入门

- 考点：变量、数据类型、取整函数

- 使用函数：ceil() 向上取整，floor() 向下取整

- 利用int()显式数据类型转换

第二部分

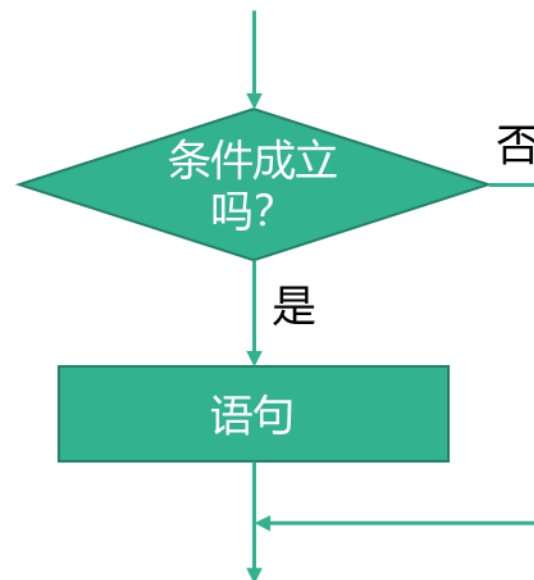
C++基本语法

——分支结构

- 知识点1：if-else语句
- 知识点2：关系运算
- 知识点3：逻辑运算
- 知识点4：运算符优先级
- 知识点5：switch语句
- 练习1：闰年判断
- 练习2：输出星期几

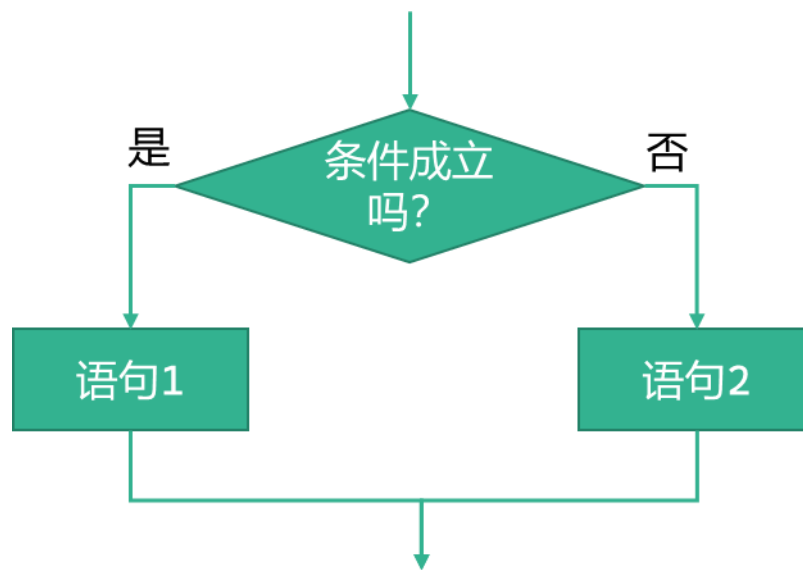
if 语句

- 单分支结构：满足“判断条件”则执行语句，否则跳过
- if（判断条件）
 - {
 - 执行语句；
 - }



if-else语句

- 如果满足条件1，执行语句1；否则（不满足条件1），执行语句2
- ```
if(条件1)
 {语句1;}
else
 {语句2;}
```



## if-else嵌套

- 在if...else分支中还存在if...else语句。
- 分别满足什么条件时，
  - 执行语句3、语句4、语句5、语句6、语句7？

```
if(判断条件1)
{
 语句1... ;
 语句2... ;
 if(判断条件2)
 {
 语句3... ;
 }
 else
 {
 语句4... ;
 }
}
else
{
 语句5... ;
 if(判断条件3)
 {
 语句6... ;
 }
 else
 {
 语句7... ;
 }
}
```

## 关系运算

- 关系运算符用于表达式之间的大小比较，共有6种。
- 关系表达式判断结果成立，返回值为真值；不成立，返回值为假值。
- 关系运算符的优先级低于算数运算符，高于赋值运算符。
  - $\leq$ 、 $\geq$ 、 $<$ 、 $>$  的优先级高于  $==$  和  $!=$

| 关系运算符              | 术语   | 示例                   | 结果 | 计算机返回结果 |
|--------------------|------|----------------------|----|---------|
| <code>==</code>    | 相等于  | <code>4==3</code>    | 假  | 0       |
| <code>!=</code>    | 不等于  | <code>4!=3</code>    | 真  | 1       |
| <code>&lt;</code>  | 小于   | <code>4&lt;3</code>  | 假  | 0       |
| <code>&gt;</code>  | 大于   | <code>4&gt;3</code>  | 真  | 1       |
| <code>&lt;=</code> | 小于等于 | <code>4&lt;=3</code> | 假  | 0       |
| <code>&gt;=</code> | 大于等于 | <code>4&gt;=3</code> | 真  | 1       |

```
if((x>3)==0)
{
 x=x+1;
}
```



## 逻辑运算

- 逻辑运算符，对表达式或者变量进行逻辑判断，有3种。
- 运算结果为真或假，在表达式中分别用1和0来代表。

| 逻辑运算符 | 含义         | 形式     | 描述                 |
|-------|------------|--------|--------------------|
| !     | 非（理解为“相反”） | ! a    | 如果原表达式为真，则“!表达式”为假 |
| &&    | 与（理解为“并且”） | a && b | &&左右两边都是真时，结果才为真   |
|       | 或（理解为“或者”） | a    b | 左右两边都为假时，结果才为假     |

## 运算符优先级

|                                              |       |     |           |       |     |     |       |
|----------------------------------------------|-------|-----|-----------|-------|-----|-----|-------|
| !                                            | * / % | + - | < > <= >= | == != | &&  |     | =     |
| 逻辑非                                          | 算术运算符 |     | 关系运算符     |       | 逻辑与 | 逻辑或 | 赋值运算符 |
| 优先级高 <span style="float:right">→ 优先级低</span> |       |     |           |       |     |     |       |
| 注意：（）优先级最高，可用（）调整运算顺序                        |       |     |           |       |     |     |       |

- 表达式从左到右运算，从右向左赋值

## 经典练习1：闰年判断

- 闰年的年份能被4整除但不能被100整除，或者能被400整除。
- 如何写判断条件？

$(year \% 4 == 0 \&\& year \% 100 != 0 || year \% 400 == 0)$

## 分支结构补充——switch语句

- **switch()** 中应该填写要判断的内容，可以是某个变量，也可以是某个表达式。
  - 需要注意的是，变量或者表达式的结果只能是int型、char型、bool型
- **case后跟常量**，用于匹配switch（）中的内容。
  - case后的常量类型应该和switch（）中的数据类型相同，可以为int型、char型、bool型
- **break;**
  - break的作用是跳出switch条件语句，如果缺少，则会执行case匹配后的所有语句。
- **default:**
  - 如果前面所有case都不满足，则执行default后的语句。
  - default并不是必须的，可以根据实际情况省略。

```
switch(变量或表达式)
{
 case 常量1 :
 语句块... ..
 break ;

 case 常量2 :
 语句块... ..
 break ;

 case 常量3 :
 语句块... ..
 break ;

 default :
 语句块... ..
}
```

## 经典练习2：输出星期几

- 根据输入的数值1~7，输出对应的星期一.....星期日

```
int day;
cin>>day;
switch(day)
{
 case 1:
 cout<<"星期一";
 break;
 case 2:
 cout<<"星期二";
 break;
 case 3:
 cout<<"星期三";
 break;
 case 4:
 cout<<"星期四";
```

## 第三部分： C++基础语法 ——综合练习

- 练习5：月份天数（P5716）
- 练习6：小王家的电费（P1422）
- 练习7：买铅笔（P1909）
- 练习8：三角形分类（P5717）
- 练习9：ABC输出（P4414）
- 练习10：上学迟到（P5707）
- 练习11：ISBN编码（P1055）

- 练习5：月份天数（洛谷）
  - 题号：P5716
  - 难度：入门
  - 考点：判断条件设置、逻辑表达式

- 练习6：小王家的电费（洛谷）
  - 题号：P1422
  - 难度：入门
  - 考点：条件判断，分段计算，格式化输出



- 练习7：买铅笔（洛谷）
  - 题号：P1909
  - 难度：入门
  - 背景：2016普及组T1
  - 考点：输入输出，条件判断，向上取整

- 练习8：三角形分类
  - 题号：P5717
  - 难度：普及-
  - 考点：条件判断
- 三目运算(if-else的简化)

- 练习9：ABC输出
  - 题号：P4414
  - 难度：入门
  - 考点：条件判断，字符变量，输入输出，三目运算

- 练习10：上学迟到（洛谷）
  - 题号：P5707
  - 难度：普及-
  - 考点：顺序计算、情况判断、按要求输出、头文件

- 练习11：ISBN编码（洛谷）
  - 题号：P1055
  - 难度：普及-
  - 考点：格式化输入输出，字符数值转换，条件判断
  - 背景：2008普及组T1