

1初识C语言

课前准备

博客

CSDN – 推荐 <https://blog.csdn.net/>

富文本 Markdown

github – gitee

<https://github.com/>

<https://gitee.com/>

思维导图

xmind – 整理框架

学习点

1. 语言
2. 数据结构和算法
3. 操作系统
4. 计算机网络
5. 实战项目

一. C语言（面向过程）

计算机语言，人和计算机交流的一种语言
擅长底层应用开发 电脑硬件

ANSI C – C国际标准

1. 机器语言 – 2进制 0和1

2. 汇编语言 – 助记符

例：0101 – ADD

0000 – JMP

3. B语言

4. C语言

二. 第一个C语言程序

Hello World Ctrl+F5 开始执行

```
1 #include <stdio.h>
2 // stdio - 标准输入输出
3
4 // main函数是程序的入口
5 // 一个工程中main函数有且仅有一个
6 int main()
7 {
8     // 库函数 - 打印函数 - 输出函数
9     printf("Hello World!\n");
10    return 0;
11 }
```



三. 数据类型

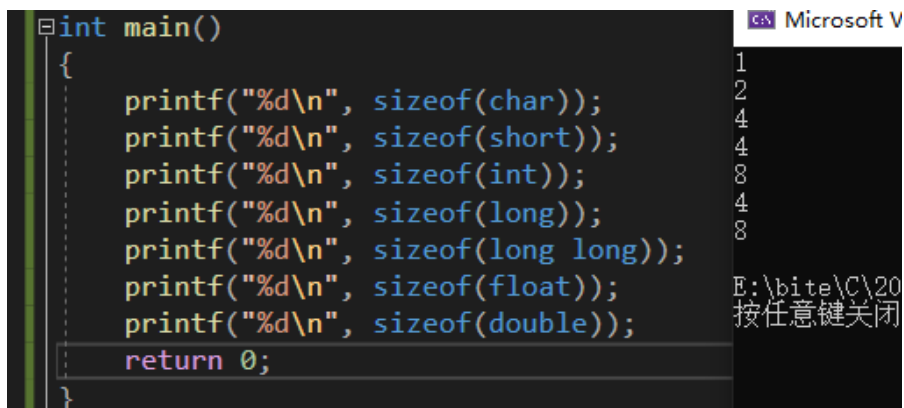
1. 类型

char	//字符数据类型
short	//短整型
int	//整形
long	//长整型

long long //更长的整形
float //单精度浮点数
double //双精度浮点数 --> 科学计数法, 小数点可以浮动

2. 每种类型的大小

```
1 int main()  
2 {  
3     printf("%d\n", sizeof(char));  
4     printf("%d\n", sizeof(short));  
5     printf("%d\n", sizeof(int));  
6     printf("%d\n", sizeof(long));  
7     printf("%d\n", sizeof(long long));  
8     printf("%d\n", sizeof(float));  
9     printf("%d\n", sizeof(double));  
10    return 0;  
11 }
```



```
int main()  
{  
    printf("%d\n", sizeof(char));  
    printf("%d\n", sizeof(short));  
    printf("%d\n", sizeof(int));  
    printf("%d\n", sizeof(long));  
    printf("%d\n", sizeof(long long));  
    printf("%d\n", sizeof(float));  
    printf("%d\n", sizeof(double));  
    return 0;  
}
```

1
2
4
4
8
4
8
E:\bite\C\20
按任意键关闭

%d – 整数

%c – 字符

计算机中的单位:

bit – 比特

byte – 字节 = 8bit

kb = 1024byte

mb = 1024kb

gb = 1024mb

tb = 1024gb

pb = 1024tb

为什么出现这么多类型:

整数: 为了选择适合范围的整形类型; 浮点数: 选择更加符合精度的类型。为了丰富地表达生活中的各种值。

sizeof(long)>=sizeof(int) long可能是4或8

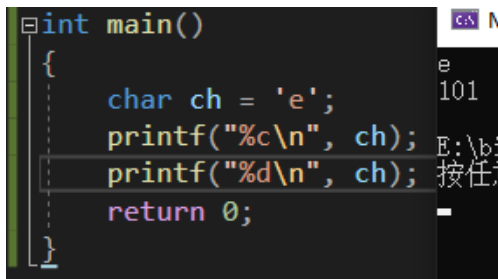
3. 使用

```
1 int main()
```

```

2 {
3     char ch = 'e';
4     printf("%c\n", ch);
5     printf("%d\n", ch);
6     return 0;
7 }

```



4. ASCII编码

每个字符对应一个值 转化为二进制存进

如 a – 97

b – 98

A – 65

B – 66

ASCII表																									
(American Standard Code for Information Interchange 美国标准信息交换代码)																									
高四位	ASCII控制字符													ASCII打印字符											
	0000						0001							0010	0011	0100	0101	0110	0111						
	十进制	字符	Ctrl	代码	转义	字符解释	十进制	字符	Ctrl	代码	转义	字符解释	十进制	十进制	十进制	十进制	十进制	十进制	十进制	十进制	十进制	十进制	十进制	十进制	十进制
低四位	0	0	^@	NUL	\0	空字符	16	▶	^P	DLE		数据链路转义	32		48	0	64	@	80	P	96	`	112	p	
	0001	1	1	^A	SOH	标题开始	17	◀	^Q	DC1		设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q	
	0010	2	2	^B	STX	正文开始	18	↑	^R	DC2		设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r	
	0011	3	3	^C	ETX	正文结束	19	!!	^S	DC3		设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s	
	0100	4	4	^D	EOT	传输结束	20	¶	^T	DC4		设备控制 4	36	\$	52	4	68	D	84	T	100	d	116	t	
	0101	5	5	^E	ENQ	查询	21	§	^U	NAK		否定应答	37	%	53	5	69	E	85	U	101	e	117	u	
	0110	6	6	^F	ACK	肯定应答	22	—	^V	SYN		同步空闲	38	&	54	6	70	F	86	V	102	f	118	v	
	0111	7	7	^G	BEL	\a 响铃	23	↕	^W	ETB		传输块结束	39	'	55	7	71	G	87	W	103	g	119	w	
	1000	8	8	^H	BS	\b 退格	24	↑	^X	CAN		取消	40	(56	8	72	H	88	X	104	h	120	x	
	1001	9	9	^I	HT	\t 横向制表	25	↓	^Y	EM		介质结束	41)	57	9	73	I	89	Y	105	i	121	y	
	1010	A	10	^J	LF	\n 换行	26	→	^Z	SUB		替代	42	*	58	:	74	J	90	Z	106	j	122	z	
	1011	B	11	^K	VT	\v 纵向制表	27	←	^[_	ESC	\e	溢出	43	+	59	;	75	K	91	[107	k	123	{	
	1100	C	12	^L	FF	\f 换页	28	└	^[_	FS		文件分隔符	44	,	60	<	76	L	92	\	108	l	124		
	1101	D	13	^M	CR	\r 回车	29	↔	^J	GS		组分隔符	45	-	61	=	77	M	93]	109	m	125	}	
	1110	E	14	^N	SO	\s 移出	30	▲	^_	RS		记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~	
	1111	F	15	^O	SI	\s 移入	31	▼	^_	US		单元分隔符	47	/	63	?	79	O	95	_	111	o	127	␣	^Backspace 代码: DEL

注：表中的ASCII字符可以用“Alt + 小键盘上的数字键”方法输入。

<https://blog.csdn.net/2013/03/08/1081919>

四. 变量 常量

1. 定义变量

age是变量 可以被改变

```

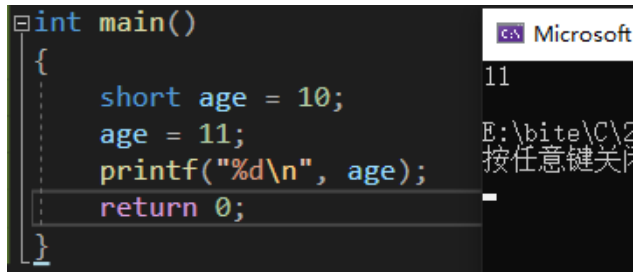
1 int main()
2 {
3     short age = 10;
4     age = 11;

```

```

5     printf("%d\n", age);
6     return 0;
7 }

```



```

int main()
{
    short age = 10;
    age = 11;
    printf("%d\n", age);
    return 0;
}

```

Microsoft
11
E:\bite\C\2
按任意键关闭

2. 变量的分类

局部变量

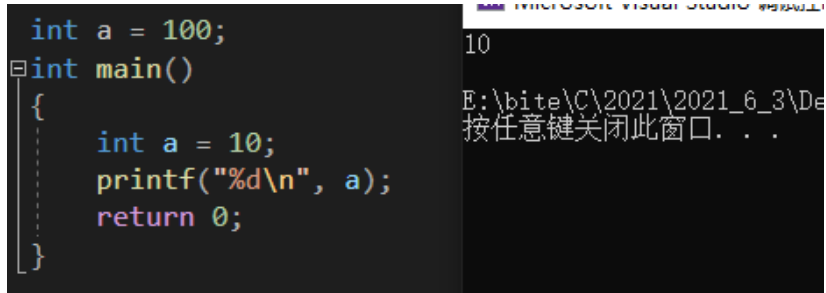
全局变量

```

1  int g = 100; // 全局变量
2
3  int main()
4  {
5      int a = 10; // 局部变量 大括号内部
6      return 0;
7  }

```

局部变量和全局变量的名字可以相同，相同时，局部优先，例：



```

int a = 100;
int main()
{
    int a = 10;
    printf("%d\n", a);
    return 0;
}

```

Microsoft Visual Studio
10
E:\bite\C\2021\2021_6_3\De
按任意键关闭此窗口...

3. 使用 2个整数相加

```

1  int main()
2  {
3      // 2个整数相加
4      int num1 = 0;
5      int num2 = 0;
6      // 输入
7      scanf_s("%d%d", &num1, &num2);
8      int sum = num1 + num2;
9      // 输出
10     printf("%d\n", sum);
11     return 0;
12 }

```

```
int main()
{
    int num1 = 0;
    int num2 = 0;
    scanf_s("%d%d", &num1, &num2);
    int sum = num1 + num2;
    printf("%d\n", sum);
    return 0;
}
```

Microsoft Visual Studio
10 20
30
E:\bite\C\2021\test_6
按任意键关闭此窗口...

4. 变量的作用域和生命周期

作用域：

局部变量的作用域：局部变量所在的局部范围。

```
1 #define _CRT_SECURE_NO_WARNINGS 1
2
3 #include <stdio.h>
4
5 int main()
6 {
7     int b = 100;
8     {
9         int a = 10;
10        printf("a = %d\n", a); // 10
11        printf("b = %d\n", b); // 100
12    }
13    printf("b = %d\n", b); // 100
14    return 0;
15 }
```

```
#include <stdio.h>

int main()
{
    int b = 100;
    {
        int a = 10;
        printf("a = %d\n", a);
        printf("b = %d\n", b);
    }
    printf("b = %d\n", b);
    return 0;
}
```

Microsoft Visual Studio
a = 10
b = 100
b = 100
E:\bite\C\2021\test_6
按任意键关闭此窗口...

全局变量的作用域：整个工程都可使用

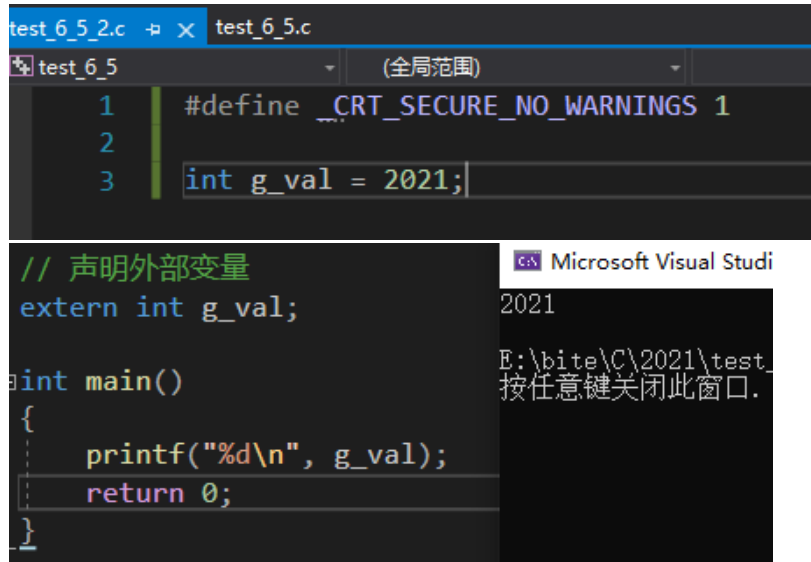
```
1 int g_val = 2021;
```

```
1 // 声明外部变量
2 extern int g_val;
```

```

3
4 int main()
5 {
6     printf("%d\n", g_val);
7     return 0;
8 }

```



生命周期：

局部变量的生命周期：进入局部变量的范围，生命周期开始，出局部变量生命周期结束。

全局变量的生命周期：程序的生命周期，

五. 常量

1. 字面常量

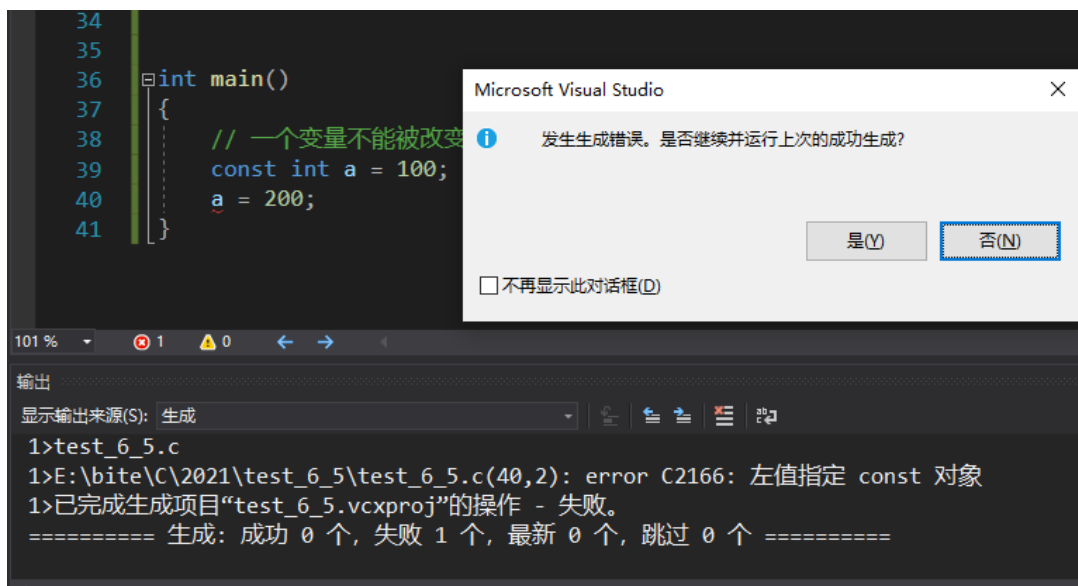
```

1 int main()
2 {
3     // 字面常量
4     3.14; // 浮点型字面常量
5
6 }

```

2. const修饰的的常变量

一个变量不能被改变我们就说具有常属性 – 不能被改变的属性



n不是常量，只是具有常属性，不能被修改，n还是变量

```
int main()
{
    const int n = 100; // n不是常量，只是具有常属性
    int arr[n] = { 0 }; // ERROR
    return 0;
}
```

3. #define 定义的标识符常量

```
// #define 定义的标识符常量
#define MAX 100
int main()
{
    int a = MAX;
    printf("a = %d\n", a); // 100
    return 0;
}
```

可用于数组

```
1 int arr[MAX] = {0};
```

4. 枚举常量

枚举：——列举

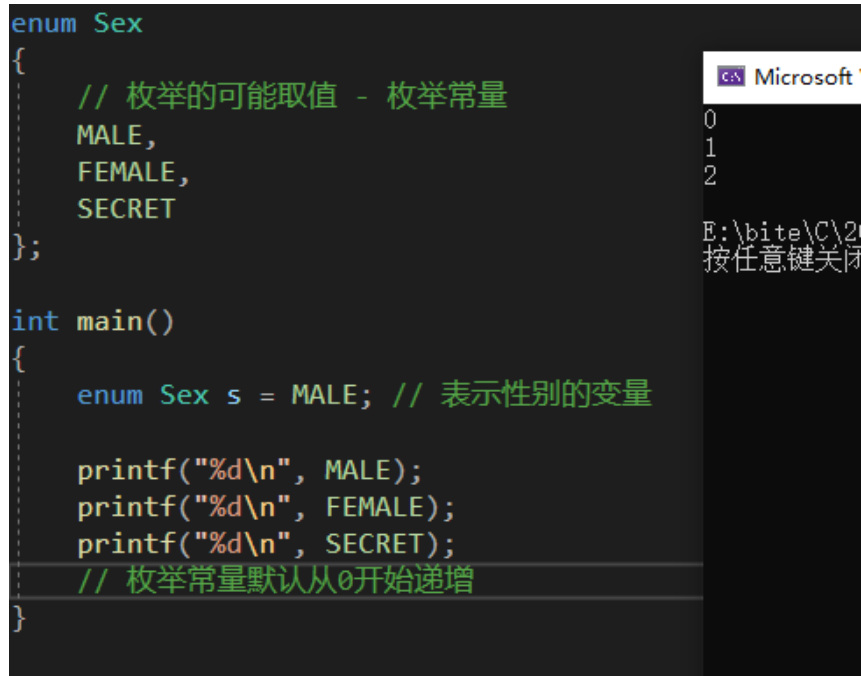
```
1 // enum - 枚举关键字
2 // enum Sex - 枚举类型
3 enum Sex
4 {
5     // 枚举的可能取值 - 枚举常量
6     MALE,
7     FEMALE,
8     SECRET
9 };
```



```

10
11 int main()
12 {
13     enum Sex s = MALE; // 表示性别的变量
14
15     printf("%d\n", MALE);
16     printf("%d\n", FEMALE);
17     printf("%d\n", SECRET);
18     // 枚举常量默认从0开始递增 0 1 2
19 }

```



```

enum Sex
{
    // 枚举的可能取值 - 枚举常量
    MALE,
    FEMALE,
    SECRET
};

int main()
{
    enum Sex s = MALE; // 表示性别的变量

    printf("%d\n", MALE);
    printf("%d\n", FEMALE);
    printf("%d\n", SECRET);
    // 枚举常量默认从0开始递增
}

```

0
1
2
E:\bite\C\2
按任意键关闭

赋初值可以更改，枚举常量不能改

```

1 enum Sex
2 {
3     MALE = 3,
4     FEMALE = 7,
5     SECRET
6 };
7
8 int main()
9 {
10     MALE = 4; // ERROR
11 }

```

5. C99中引入一个：变长数组

若编译器支持C99标准变长允许使用变量

```

1 int main()
2 {
3     const int n = 3;
4     int arr[n] = { 0, 1, 2 };

```

```

5 printf("%d", arr[0]);
6 printf("%d", arr[1]);
7 printf("%d", arr[2]);
8 return 0;
9 }

```

六. 字符串 转义字符 注释

1. 字符串

char – 字符类型

'a' – 字符:有单引号引起是

"abcdef" – 字符串:由双引号引起的一串字符

"abcdef"; // 字符串字面值

```

1 char ch[10] = "abcdef"; // 用abcdef初始化数组

```

// 其中有七个字符abcdef\0, 剩下三个字符默认放\0 -- abcdef\0\0\0\0

打印字符串用%s

```

1 printf("%s\n", ch);

```

\0 – 字符串的结束标志

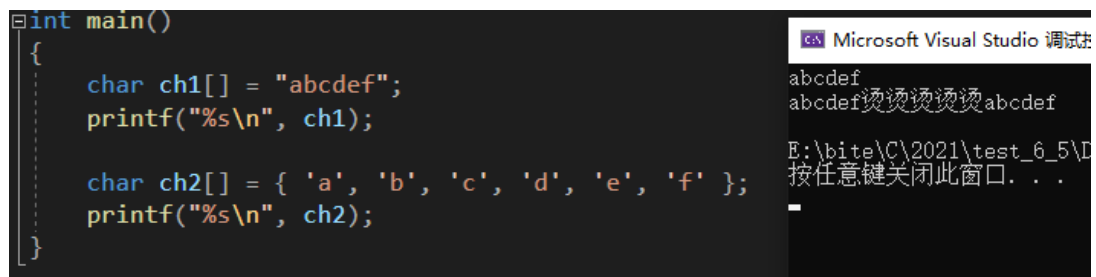
两种形式, 用字符串打印时, f后隐藏\0;

而单个字符打印时, 没有\0, 此时要添加'\0', 否则会出现乱码

```

1 int main()
2 {
3     char ch1[] = "abcdef";
4     printf("%s\n", ch1);
5
6     char ch2[] = { 'a', 'b', 'c', 'd', 'e', 'f', '\0' };
7     printf("%s\n", ch2);
8 }

```



```
int main()
{
    char ch1[] = "abcdef";
    printf("%s\n", ch1);

    char ch2[] = { 'a', 'b', 'c', 'd', 'e', 'f', '\0' };
    printf("%s\n", ch2);
}
```

Microsoft Visual Studio

abcdef
abcdef

E:\bite\C\2021\test_...
按任意键关闭此窗口.

求字符串长度

```
1 #include <string.h>
2 int main()
3 {
4     char ch1[] = "abcdef";
5     printf("%d\n", strlen(ch1)); // 6
6     // strlen 库函数 - 求字符串长度的 \0是字符串结束标志 不算长度
7
8     char ch2[] = { 'a', 'b', 'c', 'd', 'e', 'f', '\0' };
9     printf("%d\n", strlen(ch2)); // 6
10    return 0;
11 }
```

2. 转义字符