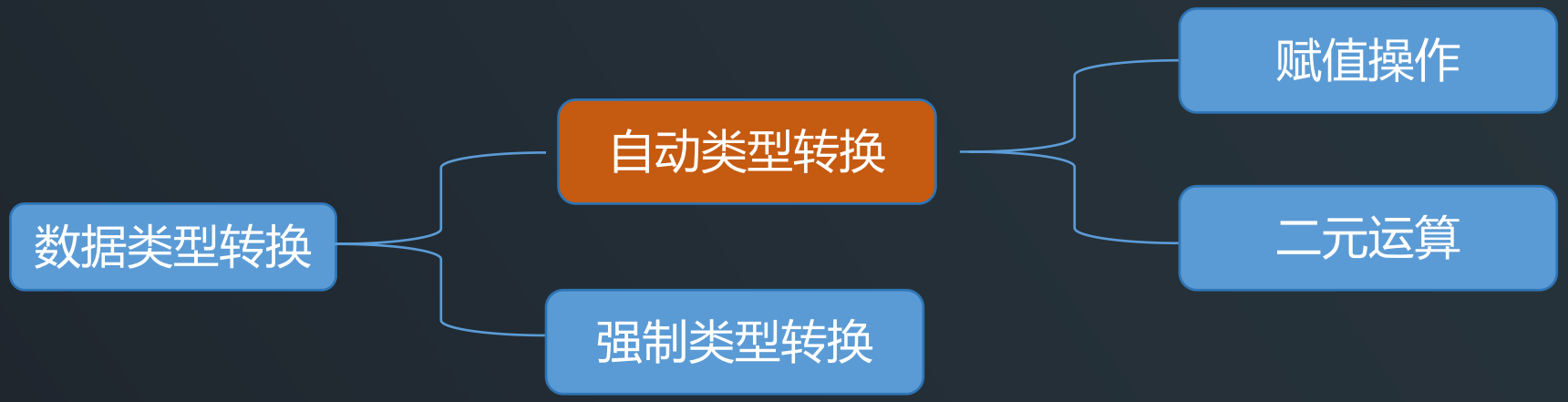
强制类型转换

华清远见 | 创客学院 小美老师

数据类型转换




```
int   pi  =  3.14
```

3

整数　　小数

double 8字节
int 4字节

在对变量赋值时，如果等号右边的表达式值与左边的变量类型不同，右边的类型将转换为左边的类型

并且如果右边的数据类型长度比左边的长，就会丢失一部分数据，导致精度降低

自动类型转换

```
#include <stdio.h>
int main()
{
  double a = 100;//100是整数，会先转换为double后赋值给变量a;
  int b = 3.14;//3.14会自动舍弃小数部分再赋值给b
  printf("a=%lf\n", a);
  printf("b=%d\n", b);
  return 0;
}
```

```
$ gcc type_convertion1.c -Wall
$ ./a.out
a=100.000000
b=3
```

a没有损失精度

b的小数部分被舍弃了

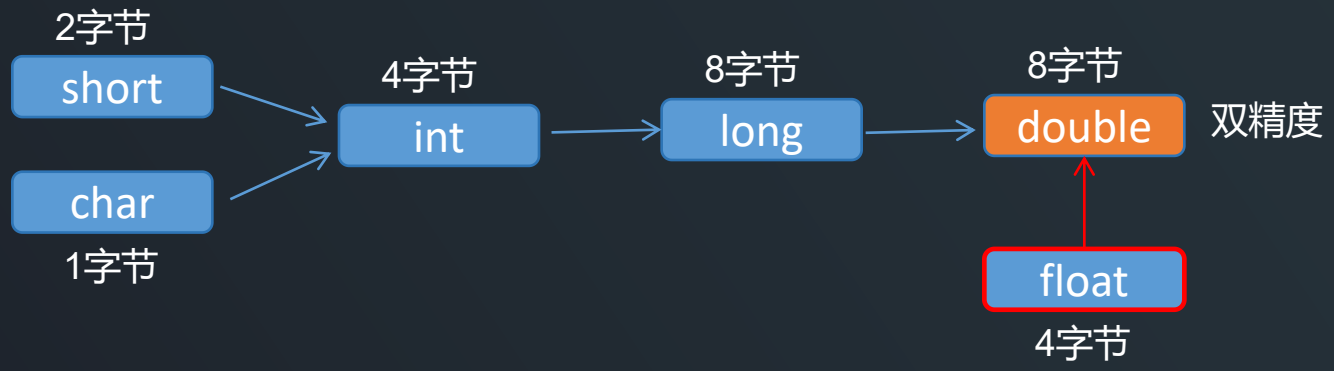
自动类型转换

- 在不同数据类型之间进行二元运算时，也会发生隐士的自动类型转换
- 如果参与运算的变量类型不同，会先转换成同一类型再进行计算

```
#include <stdio.h>
int main()
{
  char a = 'a';     //-->97
  int b = 2000;
  printf("%d\n", a + b);   //97+2000=2097
  return 0;
}
```

自动类型转换

- 如果运算时发生类型转换，就会按数据长度增加的方向进行，从而保证精度不降低
- 并且所有浮点数参与的运算都以双精度进行，即使表达式中只有float，也会先转为double，再进行计算



隐士自动类型转换

- 隐士自动类型转换是编译器根据代码上下文自行判断的结果

```
#include <stdio.h>
int main(int argc, const char *argv[])
{
int a = 50;
//由于0.3是double型，a会转为double型，再和0.3相乘，将结果赋给b
double b = a * 0.3;
printf("result=%lf\n", b);
return 0;
}
```

```
$ gcc type_convertion3.c -Wall
$ ./a.out
result=15.000000
```

强制类型转换

- 为了确保类型转换按照开发者期望的方式进行，可以明确设置类型转换的方式，这就是强制类型转换
- 强制类型转换是一种运算符，在需要转换的表达式前添加小括号括起来的新类型名称

```
(type_name)expression
```

新类型名称　　表达式

强制类型转换

```
#include <stdio.h>
int main(int argc, const char *argv[])
{
        int a = 100;
        double x = 1.23;
        double y = 2.99;
        //将变量a转换为float类型
        printf("(float)a=%f\n", (float)a);
        //将表达式x+y的结果转为int类型
        //1.23+2.99=4.22,强制转为int类型,小数部分被丢弃了
        printf("(int)(x+y)=%d\n", (int)(x+y));
        return 0;
}
```

强制类型转换

- 有时，为了得到正确的结果，必须要使用强制类型转换

- 示例：要求变量c可以保留2个小数位

```
int a = 100;
int b = 3;
double c;
c = a / b;
```

整形数转浮点数时要显示的转换类型

```
#include <stdio.h>
int main()
{
        int a = 100;
        int b = 3;
        double c;
        c = a / b;
        printf("c = %.2lf\n", c);
        return 0;
}
```

```
$ gcc type_convert.c -Wall
$ ./a.out
c = 33.00
```

错误

整形数转浮点数时要显示的转换类型

```c
#include <stdio.h>
int main()
{
   int a = 100;
   int  b = 3;
   double c;
   //计算a除以b的值，精确到2位小数，将其中一个变量转换为浮点型再计算
    c = (double)a / b;
    printf("a/b = %.2lf\n", c);
   return 0;
}
```

```
$ gcc type_convertion5.c -Wall
$ ./a.out
a/b = 33.33
```

扫一扫，获取更多信息

THANK YOU