

cin

cin 语句共有两个功能

功能 1: 接受单个字符, 数字, 不可接受空白字符

功能 2: 接受一个字符数组, 遇“空格”、“Tab”、“回车”结束, 后面的字符不再读取。

原理:

cin 读取数据是从缓冲区中获取数据, 缓冲区为空时, cin 的成员函数会阻塞等待数据的到来, 一旦缓冲区中有数据, 就触发 cin 的成员函数去读取数据。

当 cin>>从缓冲区中读取数据时, 若缓冲区中第一个字符是空格、tab 或换行这些分隔符时, cin>>会将其忽略并清除, 继续读取下一个字符, 若缓冲区为空, 则继续等待。

但是如果读取成功, 字符后面的分隔符是残留在缓冲区的, cin>>不做处理。
使用 cin 读取输入 100 的过程如下:

我们先通过键盘输入 100, 缓冲区如下:

缓冲区			
1	0	0	\n

之后, 我们调用 cin>>a, 将数据赋给 a, 此时缓冲区如下

缓冲区			
\n			

示例 1:

```
#include <iostream>
using namespace std;
int main() {
    int a, b;
    char c, d;
    cin >> a >> b >> c >> d;
    cout << "a:" << a << endl;
    cout << "b:" << b << endl;
    cout << "c:" << c << endl;
    cout << "d:" << d << endl;
}
```

运行结果 1:

输入:

1 2 a b

输出:

a:1

b:2

c:a

d:b

运行结果 2:

输入:

1 2

输出:

无

不输入其他字符，只输入空格和回车，程序不能终止

示例 2:

```
#include <iostream>
using namespace std;
int main() {
    char name[50];
    cout << "请输入您的名称: ";
    cin >> name;
    cout << "您的名称是: " << name << endl;
}
```

运行结果:

请输入您的名称: addffd 121d

您的名称是: addffd

注意: 空格后的数字没有被获取

示例 3:

```
#include <iostream>
#include<stdio.h>
#include<string>
using namespace std;
int main() {
    char c;
    cin >> c;
    cout << c;
    cout << endl << "RETURN END" << endl;
    return 0;
}
```

运行结果:

输入:

(一大段空白)aadfsac

输出:

a

RETURN END

注意: cin 自动忽略了前面的空格

多变量输入

C++ 编译器根据要输入值的数据类型, 选择合适的流提取运算符来提取值, 并把它存储在给定的变量中。流提取运算符 **>>** 在一个语句中可以多次使用, 如果要求输入多个数据, 可以使用如下语句:

```
cin >> name >> age;
```

这相当于下面两个语句:

```
cin >> name;
cin >> age;
```

cin.get()

功能 1: 接受单个字符，数字，空白字符

功能 2: cin.get(字符数组名, 接收字符数)用来接收一行字符串，可以接收除回车外的字符。
该函数以换行符结束，但之后会丢弃换行符并以'\0'代替

示例 1:

```
#include <iostream>

using namespace std;

int main() {
    int a, b;
    char c, d;
    a = cin.get();
    b = cin.get();
    c = cin.get();
    d = cin.get();
    cout << "a:" << a << endl;
    cout << "b:" << b << endl;
    cout << "c:" << c << endl;
    cout << "d:" << d << endl;
}
```

运行结果:

输入:

1 2 a

输出:

a:49

b:32

c:2

d:

在这里 b 并没有按照预期输出 2 的 ASCII 码 50，而是输出了空格的 ASCII 码 32。

示例 2:

```
#include <iostream>

using namespace std;

int main() {
    char c[5];
    cin.get(c, 5); //数组被填满，或者遇回车结束
    for (int i = 0; i < 5; i++) {
        cout << c[i];
    }
}
```

运行结果:

输入:

12aabedf

输出:

12aa

有关数组的更多内容，我们将在后续课程进行讲解。

辑航线培优教育，信息学奥赛培训专家。

扫码添加作者获取更多内容。

