基本语法

```
int scanf(const char *format, ...);
```

- **format**：格式化字符串，包含占位符（如 %d, %f 等），用于指定输入数据的类型。
- ...：可变参数列表，指向存储输入数据的变量地址。

常用占位符

占位符	含义	示例
%d	有符号十进制整数	scanf("%d", &num);
%u	无符号十进制整数	scanf("%u", &num);
%f	浮点数（float）	scanf("%f", &floatNum);
%lf	双精度浮点数（double）	scanf("%lf", &doubleNum);
%c	单个字符	scanf("%c", &charVar);
%s	字符串（不含空格）	scanf("%s", str);
%x	十六进制整数	scanf("%x", &hexNum);
%o	八进制整数	scanf("%o", &octNum);
%p	指针地址	scanf("%p", &ptr);

示例

1. 读取整数

```
#include <stdio.h>

int main() {
    int num;
    printf("请输入一个整数：");
    scanf("%d", &num);
    printf("你输入的整数是：%d\n", num);
    return 0;
}
```

2. 读取浮点数

```c
#include <stdio.h>

int main() {
    float num;
    printf("请输入一个浮点数：");
    scanf("%f", &num);
    printf("你输入的浮点数是：%.2f\n", num);
    return 0;
}
```

3. 读取字符串

```c
#include <stdio.h>

int main() {
    char name[50];
    printf("请输入你的名字：");
    scanf("%s", name);  // 注意：不能读取带空格的字符串
    printf("你好，%s！\n", name);
    return 0;
}
```

4. 读取多个值

```c
#include <stdio.h>

int main() {
    int age;
    float height;
    printf("请输入你的年龄和身高（用空格分隔）：");
    scanf("%d %f", &age, &height);
    printf("年龄：%d，身高：%.2f\n", age, height);
    return 0;
}
```

注意事项

1. **变量地址**：
    - `scanf` 需要变量的地址，因此必须使用取地址运算符 `&`。
    - 例外：读取字符串时，直接传递字符数组名（因为数组名本身就是地址）。

```c
int num;
scanf("%d", &num);  // 正确
```

```
scanf("%d", num);    // 错误

char str[50];
scanf("%s", str);    // 正确
scanf("%s", &str);   // 错误
```

2. **空格和处理空白字符**：
   - `scanf` 会跳过空白字符（空格、制表符、换行符）。
   - 如果需要读取带空格的字符串，使用 `fgets`。

```
char sentence[100];
fgets(sentence, 100, stdin);  // 正确
```

3. **返回值**：
   - `scanf` 返回成功读取的数据项数。
   - 如果返回值小于预期，说明输入格式不匹配。

```
int a, b;
int result = scanf("%d %d", &a, &b);
if (result == 2) {
    printf("成功读取两个整数：%d 和 %d\n", a, b);
} else {
    printf("输入格式错误！\n");
}
```

4. **缓冲区问题**：
   - `scanf` 可能会留下换行符或其他字符在输入缓冲区中，影响后续输入。
   - 可以使用 `getchar()` 清空缓冲区。

5. **安全性**：
   - `scanf` 不检查输入数据的长度，可能导致缓冲区溢出。
   - 对于字符串输入，建议使用 `fgets` 或设置宽度限制。

```
char buffer[10];
scanf("%9s", buffer);  // 最多读取 9 个字符，避免溢出
```

高级用法

1. 宽度限定

- 可以限制读取的字符数，防止溢出。

```c
char str[10];
scanf("%9s", str);  // 最多读取 9 个字符
```

2. 忽略特定输入

- 使用 `*` 忽略某些输入。

```c
int num;
scanf("%*d %d", &num);  // 忽略第一个整数，读取第二个
```

3. 匹配特定字符

- 使用 `%[ ]` 匹配特定字符集。

```c
char input[50];
scanf("%[0-9]", input);  // 只读取数字字符
```

完整示例

```c
#include <stdio.h>

int main() {
    int age;
    float height;
    char name[50];

    printf("请输入你的年龄：");
    scanf("%d", &age);

    printf("请输入你的身高：");
    scanf("%f", &height);

    printf("请输入你的名字：");
    scanf("%s", name);

    printf("你的年龄是：%d，身高是：%.2f，名字是：%s\n", age, height, name);

    return 0;
}
```