

目标

- 格式化输出
 - 格式化符号
 - f-字符串
- print的结束符

输出

作用：程序输出内容给用户

```
1 print('hello Python')
2
3 age = 18
4 print(age)
5
6 # 需求：输出“今年我的年龄是18岁”
```

一. 格式化输出

所谓的格式化输出即按照一定的格式输出内容。

1.1 格式化符号

格式符号	转换
%s	字符串
%d	有符号的十进制整数
%f	浮点数
%c	字符
%u	无符号十进制整数
%o	八进制整数
%x	十六进制整数（小写ox）
%X	十六进制整数（大写OX）
%e	科学计数法（小写'e'）
%E	科学计数法（大写'E'）
%g	%f和%e的简写
%G	%f和%E的简写

技巧

- %06d，表示输出的整数显示位数，不足以0补全，超出当前位数则原样输出
- %.2f，表示小数点后显示的小数位数。

1.2 体验

格式化字符串除了%s，还可以写为 f'{表达式}'

```
1 age = 18
2 name = 'TOM'
3 weight = 75.5
4 student_id = 1
5
6 # 我的名字是TOM
7 print('我的名字是%s' % name)
8
9 # 我的学号是0001
10 print('我的学号是%d' % student_id)
11
12 # 我的体重是75.50公斤
13 print('我的体重是%.2f公斤' % weight)
14
15 # 我的名字是TOM，今年18岁了
16 print('我的名字是%s，今年%d岁了' % (name, age))
17
```

```
18 # 我的名字是TOM, 明年19岁了
19 print('我的名字是%s, 明年%d岁了' % (name, age + 1))
20
21 # 我的名字是TOM, 明年19岁了
22 print(f'我的名字是{name}, 明年{age + 1}岁了')
```

f-格式化字符串是Python3.6中新增的格式化方法，该方法更简单易读。

1.3 转义字符

- `\n`：换行。
- `\t`：制表符，一个tab键（4个空格）的距离。

1.4 结束符

想一想，为什么两个print会换行输出？

```
1 print('输出的内容', end="\n")
```

在Python中，`print()`，默认自带 `end="\n"` 这个换行结束符，所以导致每两个 `print` 直接会换行展示，用户可以按需求更改结束符。

总结

- 格式化符号
 - `%s`：格式化输出字符串
 - `%d`：格式化输出整数
 - `%f`：格式化输出浮点数
- f-字符串
 - `f'{表达式}'`
- 转义字符
 - `\n`：换行
 - `\t`：制表符
- print结束符

```
1 print('内容', end="")
```