

# 一 基本数据类型与变量

print() 输出语句

## 1.1 注释

注释：给代码打上说明 表明这段代码的作用或者实现的功能

优点：

1. 代码说明
2. 不让解释器执行注释的那句话

单行注释：python使用 # 作为单行注释的符号

#号开头 #号右边的东西都会被当做一个说明

单行注释快捷键：ctrl + ? (/)

多行注释：三组双引号 注释内容写中间 """ 内容 """

## 1.2 变量

什么是变量？可以改变的量

计算机用来存储数据的盒子，想用这个数据，直接拿盒子就好了

将数据放入变量的过程叫做赋值

变量名 = 值

```
a = 10
print(a)
```

```
a = 'ABC'
b = a    # b = 'ABC'
a = 'IG'
print('b=', b)
print('a=', a)
```

不要把赋值语句的等号等同于数学的等号

总结：

1. 变量就是可以变化的量，可以随时进行修改的
2. 程序是用来处理数据的，而变量就是用来存储数据的
3. 变量就是一个存储数据时，当前数据所在的内存地址的名字

多个变量赋值

```
a, b, c = 1, 2, 'd'
print(a)
print(b)
print(c)

a = b = c = 1
a = 1
b = 1
c = 1
print(a)
print(b)
print(c)
```

## 1.3 标识符和关键字

### 1.3.1 标识符

规则如下:

- 由数字, 字母, 下划线组成
- 不能以数字开头
- 不能使用内置的关键字 (print)
- 严格区分大小写

```
print_21 = 1
print(print_21)
```

### 1.3.2 命名规范

- 标识符命名要做到见名知意

name , age, height

驼峰命名法

- 大驼峰: 第一个单词的首字母都采用大写字母
  - FirstName , LastName
- 小驼峰: 第一个单词以小写字母开始, 第二个单词的首写字母大写
  - myName, youName
- 使用下划线来连接所有单词(关键字)

### 1.3.3 关键字

```
# 打印python所有的关键字
import keyword # 导包
print(keyword.kwlist)

['False', 'None', 'True', '__peg_parser__', 'and', 'as', 'assert', 'async',
'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except',
'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda',
'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with',
'yield']
```

### 关键字不能声明为变量名

1,2, 100 , -500 , 0

```
print(f1)
```

### 1.4.1.3 布尔值 bool

只有True和False两种值，注意大小写

True表示真(表示条件满足或者成立) == 1

False表示假(表示条件不满足或者不成立) == 0

```
print(3<2)
print(3>2)
```

### 1.4.1.4 字符串 str

字符串不属于数值类型 属于 序列类型

以单引号，双引号，三引号括起来的任意的文本

基本使用

```
相加和想乘
name = '左手'
jiaxiang = '湖南'
# age = 18
print(name+jiaxiang) # 相加就是简单的字符串拼接
print(name*5) # 想乘就是将字符串和自己相加了5次
```

两种取值顺序:

- 从左到右索引默认0开始，最大范围是字符串长度少1
- 从右到左索引默认-1开始，最大范围是字符串的开头



下标取值： 变量名[]

```
s = '左手'
print(s[1])
print(s[-1])

s = 'hello world' # 在字符串内部 空格也算作一个字符
print(s[5])
```

子字符串(变量名)[初始位置:结束位置],左闭右开  
左闭右开: 开始位置能取到, 结束位置取不到

```
s = 'abcdefg'
# print(s[1:6])
print(s[:])
# print(s[:6]) # -1 = g
# print(s[:-1])
# print(s[3:])
# print(s[-7])
```

### 1.4.1.5 查看数据类型

- 变量没有类型, 数据才有类型

type(变量的名字) ==》查看数据类型

```
a = 'ab'
b = 123
c = 1.23
print(type(a))
print(type(b))
print(type(c))
```

### 1.4.1.6 转义字符\

\n ==》换行

\t ==》制表符, 一个tab键, 也就是四个空格

\\ ==> \

转义字符必须写在引号内部, 转义就是把他变蠢了

```
# print('i 'm ok')
print("i 'm ok")
print('i \'m ok')

print('床前明月光, 疑是地上霜')
print('床前明月光\n疑是地上霜') # \n 是换行的意思
print('床前明月光\t疑是地上霜') # \t 就是四个空格

print('\\n\\')
```

## 1.5 类型转换

```
height = 175.0
height1 = int(height)
height2 = str(height)
age = 18
print(type(height))
print(type(int(height)))
print(type(height1))
# print('您的身高是:'+height)
```

```
print('您的身高是:'+height2)
```

```
ii = int('123')  
print(ii)  
123
```

```
ii = int('123个')  
print(ii)  
报错
```

## 作业

- 1.用变量给自己建立个角色信息，要输出出来(用一个print输出)

姓名  
年龄  
身高  
家乡  
兴趣  
是否有编程基础

- 2.有一个字符串变量 `s = '人生苦短，我学python'`  
通过下标索引，分别输出 苦，学 o 这几个字符

**整数（int）：**整数是没有小数部分的数字。在Python中，整数可以是正数、负数或零。 整数类型在Python 3中没有大小限制，因此可以处理非常大的整数。可以使用内置函数“`int()`”将其他类型的对象转换为整数。

**浮点数（float）：**浮点数是带有小数部分的数字。在Python中，浮点数可以是正数、负数或零。 Python使用IEEE 754标准来表示浮点数。然而，与整数不同，浮点数在进行运算时可能会遇到精度问题。可以使用内置函数“`float()`”将其他类型的对象转换为浮点数。

**布尔型（bool）：**布尔型只有两个值，`True`和`False`。它们通常用于控制流程语句中的条件。可以使用内置函数“`bool()`”将其他类型的对象转换为布尔类型。非零的数字、非空的字符串、非空的列表、元组或字典都会被转换为`True`，而其他的值都会被转换为`False`。

