

python

December 28, 2021

1 复习知识点

- 1. Python 程序需要描述数据和操作数据。
- 2. Python 程序区分大小写
- 3. 标识符的基本要求 - 驼峰法则 (camelCase) / 匈牙利法则 (Hungarian notation)
 - iPhone
 - eBay
 - johnSmith
- 4. 变量赋值及相应的类型 (主要的基本数据类型)

类型	典型值
布尔型 (bool)	True, False
整形 (int)	1, 100
浮点型 (float)	3.1415926
复数 (complex)	1+2j
字符串 (string)	'hello'

- 5. 注意 = 与 == 的区别
 - 赋值运算符 (=) - a = 10 把 10 赋给变量 a。
 - 比较运算符 (==) - a == 10 比较 a 和 10 是否相等, 相等返回 True, 不相等返回 False。
- 6. 特殊字符换行符\n

```
[1]: print("hello\nworld")
```

```
hello
world
```

```
[2]: print("hello", end="\n\n")
```

```
hello
```

```
[3]: print('a', 'b', 'c', sep='$$$')
```

```
a$$$b$$$c
```

```
[4]: print(True, 1, 3.14, 1+2j, 'hello', sep=', ')
```

True, 1, 3.14, (1+2j), hello

- 7. Python 语法的缩进格式 (严格的逻辑关系、语句块)

Guido van Rossum 认为使用缩进进行分组非常优雅, 并且对提高普通 Python 程序的清晰度有很大贡献。大多数人会在一段时间后学会喜欢这个功能。

```
[5]: a, b = 0, 1
      while a < 10:
          print(a)
          a, b = b, a+b
```

0
1
1
2
3
5
8

- 8. 输入内置函数 input() 的使用、返回值。

```
[6]: s = input('输入你的年龄:')
      print(s)
```

20

- 9. 格式化输出 print() 的应用, 包括宽度、小数点后位数等。

```
[7]: print('1      10      20      30')
      print('----+----|----+----|----+----|')
      print(f'{3.1415926535897932384626:30.8}')
      print(f'{3.1415926535897932384626:<30.8}')
      print('      1      10      20      30')
      print('      ----+----|----+----|----+----|')
      print('Pi = { :30.8}'.format(3.1415926535897932384626))
      print('Pi = { :<30.8}'.format(3.1415926535897932384626))
```

```
1      10      20      30
----+----|----+----|----+----|
                        3.1415927
3.1415927
      1      10      20      30
      ----+----|----+----|----+----|
Pi =                        3.1415927
Pi = 3.1415927
```

- 10. 各种运算符
 - + - 数值的运算、字符串、列表等的拼接
 - * - 数值的运算、字符串、列表等的重复
 - / - 除法
 - // - 整除

- % - 取余
- in - 字符串、列表、元组、集合、字典等成员资格判断
- +=, -=, *=, /=, //=, %= - 扩展的赋值运算符

```
[8]: # `+` - 数值的运算、字符串、列表等的拼接
print(1231+999999)      # 整数相加
print('Hello' + ' ' + 'World!') # 字符串拼接
print([1, 3, 7] + [5, 7, 9, 11]) # 列表拼接
print((1, 3, 7) + (5, 7, 9, 11)) # 元组拼接
```

```
1001230
Hello World!
[1, 3, 7, 5, 7, 9, 11]
(1, 3, 7, 5, 7, 9, 11)
```

```
[9]: # `*` - 数值的运算、字符串、列表等的重复
print(123321 * 3)      # 整数相乘
print('哈' * 7)        # 字符串重复
print([1, 2, 3] * 5)    # 列表重复
print([[1, 2, 3]] * 5)  # 嵌套列表重复
print((1, True, "Good") * 3) # 元组重复
print(((1, True, "Good"),) * 3) # 嵌套的元组重复
print(('One',) * 3)     # 一个元素的元组重复
```

```
369963
哈哈哈哈哈
[1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3, 1, 2, 3]
[[1, 2, 3], [1, 2, 3], [1, 2, 3], [1, 2, 3], [1, 2, 3]]
(1, True, 'Good', 1, True, 'Good', 1, True, 'Good')
((1, True, 'Good'), (1, True, 'Good'), (1, True, 'Good'))
('One', 'One', 'One')
```

```
[10]: # `/` - 除法
# `//` - 整除
# `%` - 取余
# 100 / 7 = 14 ... 2
print(100 / 7)
print(100 // 7)
print(100 % 7)
```

```
14.285714285714286
14
2
```

```
[11]: # `in` - 字符串、列表、元组、集合、字典等成员资格判断
print('a' in 'abc')    # 字符串
print('lo' in 'hello') # 字符串
print(1 in [1, 2, 3])  # 列表
print(1 in (1, 2, 3))  # 元组
```

```
print(1 in {1, 2, 3}) # 集合
print(1 in {1: 'one', 2: 'two', 3: 'three'}) # 字典键
print('one' in {1: 'one', 2: 'two', 3: 'three'}.values()) # 字典值
```

```
True
True
True
True
True
True
True
```

[12]: # `+=`, `-=`, `*=`, `/=`, `//=`, `%=` - 扩展的赋值运算符

```
x = y = 100
x = x + 10
y += 10
print('100 + 10 ->', x, y)

m = n = 100
m = m // 7
n //= 7
print('100 // 7 ->', m, n)
```

```
100 + 10 -> 110 110
100 // 7 -> 14 14
```