

浦东新区2023年Python教师培训项目

第2课-Python程序基础（一）

课程概要

本节课主要内容为学习Python中输入输出、变量、基本数据类型(number、string、list、tuple、set、dictionary)以及程序的三大执行顺序(顺序结构、分支结构、循环结构)。

知识点

输入与输出



示例用法: `print(填入内容)`

这里的`print`就是python中的输出指令，后面有一个括号，并且在括号的里面才能填入内容

课堂注意点：

`print (12345)` ✗ 间隔很大

`print(12345)` ✓ 间隔很小

所有代码都必须在英文模式下编写，括号一定要注意
中文和英文比较难区分~

`print(123456789 × 987654321)` ✗

`print(123456789 * 987654321)` ✓

很多符号都有特定的写法，比如乘号×是写成*
来进行表示的。

数字可以输出，那我们经常说的中文可以输出吗？

```
print('hello world')  
print("hello world")
```

输出中文一定要带成对的 `' '` 或 `''' '''`
为了方便，建议大家以后统一用成对的 `' '`

输出已经学会了，接下来我们来进行一个练习：

1、输出 '请问你叫什么名字？'

2、运行1之后在结果界面输入自己的名字

要想实现第2句输入，我们还是需要先认识一个英文单词：

| input 输入 | |
|----------|-------------------------|
| 英 | [¹ ɪnˈpuːt] |
| 美 | [¹ ɪnˈpuːt] |

示例用法： `input(填入内容)`

尝试以下代码：

```
print('请问你叫什么名字?')  
input()
```

```
input('请问你叫什么名字?')
```



`input`也是可以输出提示语的，相关文字也要
加成对的"

```
input('请问你叫什么名字?')  
print(input('请问你叫什么名字?'))
```

代码发现要输入两遍，而且如果输入两遍的内容不一样，那么输出的也不是第一遍的内容。那就意味着这样的代码并不能输出刚刚输入的内容。其实我们可以把这一串里面的内容放到一个容器里，然后给这个容器起个名字。下次再使用的时候，直接使用这个容器的名称就可以了。比如我们将这个容器的名称取名为a，然后用等号连接。

a = input('请问你叫什么名字?')

容器名

```
a = input('请问你叫什么名字?')  
print( a )
```

变量

程序中变量的概念来源于数学, 程序中的变量来源于数学的变量, 是计算机语言中能储存计算结果或能表示值抽象概念。变量可以通过变量名访问。

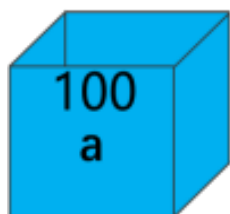
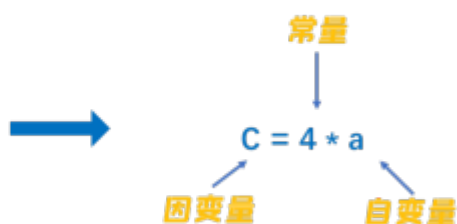
那变量具有哪些特点呢?

- 1、变量可以用来存储内容
- 2、变量必须先赋值, 再使用
- 3、变量名改变, 调用的时候变量名也必须改变
- 4、变量值可以被覆盖

变量



周长 = 4 * 边长



存 (赋值变量) **a** = 100

Labels: 变量名 (Variable Name) points to 'a', 赋值符号 (Assignment Symbol) points to '=', 变量值 (Variable Value) points to '100'.

取 (使用变量) print(a)

使用a中存放的值

变量的命名规则

- (1) 变量名由字母、数字、和下划线组成
- (2) 数字不能放在开头
- (3) 大小敏感，即A和a不是同一个变量
- (4) Python中的保留字不能作为变量名
- (5) 建议：变量命名要遵循见名知意的原则

基本数据类型

| 数据类型 | 名称 | 举例 |
|------------|-----|---|
| number | 数字 | a, b, c, d = 20, 5.5, True, 4+3j |
| string | 字符串 | str = 'String' |
| list | 列表 | list = ['abcd', 786 , 2.23, 'list', 70.2] |
| tuple | 元组 | tuple = ('abcd', 786 , 2.23, 'list', 70.2) |
| set | 集合 | student = {'Tom', 'Jim', 'Mary', 'Tom', 'Jack'} |
| dictionary | 字典 | boy= {'name': 'Tom','age':11, 'height': '150'} |

数据类型转换

| 函数 | 描述 |
|-----------------------|----------------------------------|
| int(x [,base]) | 将x转换为一个整数 |
| float(x) | 将x转换到一个浮点数 |
| complex(real [,imag]) | 创建一个复数 |
| str(x) | 将对象 x 转换为字符串 |
| repr(x) | 将对象 x 转换为表达式字符串 |
| eval(str) | 用来计算在字符串中的有效Python表达式,并返回一个对象 |
| tuple(s) | 将序列 s 转换为一个元组 |
| list(s) | 将序列 s 转换为一个列表 |
| set(s) | 转换为可变集合 |
| dict(d) | 创建一个字典。d 必须是一个 (key, value)元组序列。 |
| frozenset(s) | 转换为不可变集合 |
| chr(x) | 将一个整数转换为一个字符 |
| ord(x) | 将一个字符转换为它的整数值 |
| hex(x) | 将一个整数转换为一个十六进制字符串 |
| oct(x) | 将一个整数转换为一个八进制字符串 |

三大程序结构

顺序结构、分支结构(if语句，if...else语句，if... elif...else语句)、循环结构(for语句、while语句)

一、顺序结构

顺序结构顾名思义就是程序按照顺序来执行，类似于日记中的流水账，例如：

```
name = "张三"
```

```
age = 18
```

```
print(name)
```

```
print(age)
```

二、分支结构

分支结构又称为选择结构，程序代码根据判断条件，选择执行特定的代码。如果条件为真，程序执行一部分代码；否则执行另一部分代码。在 Python 语言中，选择结构的语法使用关键字 if、elif、else 来表示。

常用基本语法有如下几种：

if

if...else

if...elif...else

if...elif...elif.....else

if 嵌套

1、单分支结构

只有一种判断结果，语法结构：

```
if <条件1> :
```

```
    <语句1>
```

2、双分支结构

会有两种结果，语法结构：

```
if <条件1> :
```

```
    <语句1>
```

```
else :
```

```
    <语句2>
```

3、多分支结构

根据多个判断条件的结果，选择语句执行，语法结构：

```
if <条件1> :
```

```
    <语句1>
```

```
elif <条件2> :
```

```
    <语句2>
```

```
else :
```

```
    <语句N>
```

三、循环结构

循环结构是使用最多的一种结构，循环结构是指满足一定的条件下，重复执行某段代码的一种编码结构，Python的循环结构中，常见的循环结构是for循环和while循环。

1、for循环

Python for循环可以遍历任何序列的项目，经常用与字符串、列表、字典等数据结构一起使用

2、while循环

Python 编程中 while 语句用于循环执行程序，即在某条件下，循环执行某段程序，以处理需要重复处理的相同任务。其基本形式为：

while 判断条件(condition)：

 执行语句(statements).....

while循环不需要知道循环的次数，即无限循环，直到条件不满足为止。

break 与 continue 语句

两者的作用：

break：是立即退出 while 循环，不再运行循环中余下的代码，也不管条件判断的结果是否为真

continue：是结束本次循环，返回到 while 语句开始的位置，接着条件判断。如果为真，程序接着执行，否则退出。

两者的区别：

continue：跳出本次循环，只跳过本次循环 continue 后的语句

break：跳出整个循环体，循环体中未执行的循环将不会执行

相关学习网站

菜鸟教程：<https://www.runoob.com/python3/python3-tutorial.html>

鱼C课堂：http://study.163.com/course/introduction/378003.htm?utm_source=weChat&utm_medium=webShare&utm_campaign=share&utm_content=courseIntro