

变量和数据类型

一、实验介绍

1.1 实验内容

Python基本的数据类型和变量

1.2 知识点

- python 关键字
- 变量的定义与赋值
- input() 函数
- 字符串的格式化

1.3 实验环境

- python3.5
- Xfce终端
- Vim

1.4 适合人群

本课程属于初级级别课程，不仅适用于那些有其它语言基础的同学，对没有编程经验的同学也非常友好。

二、实验步骤

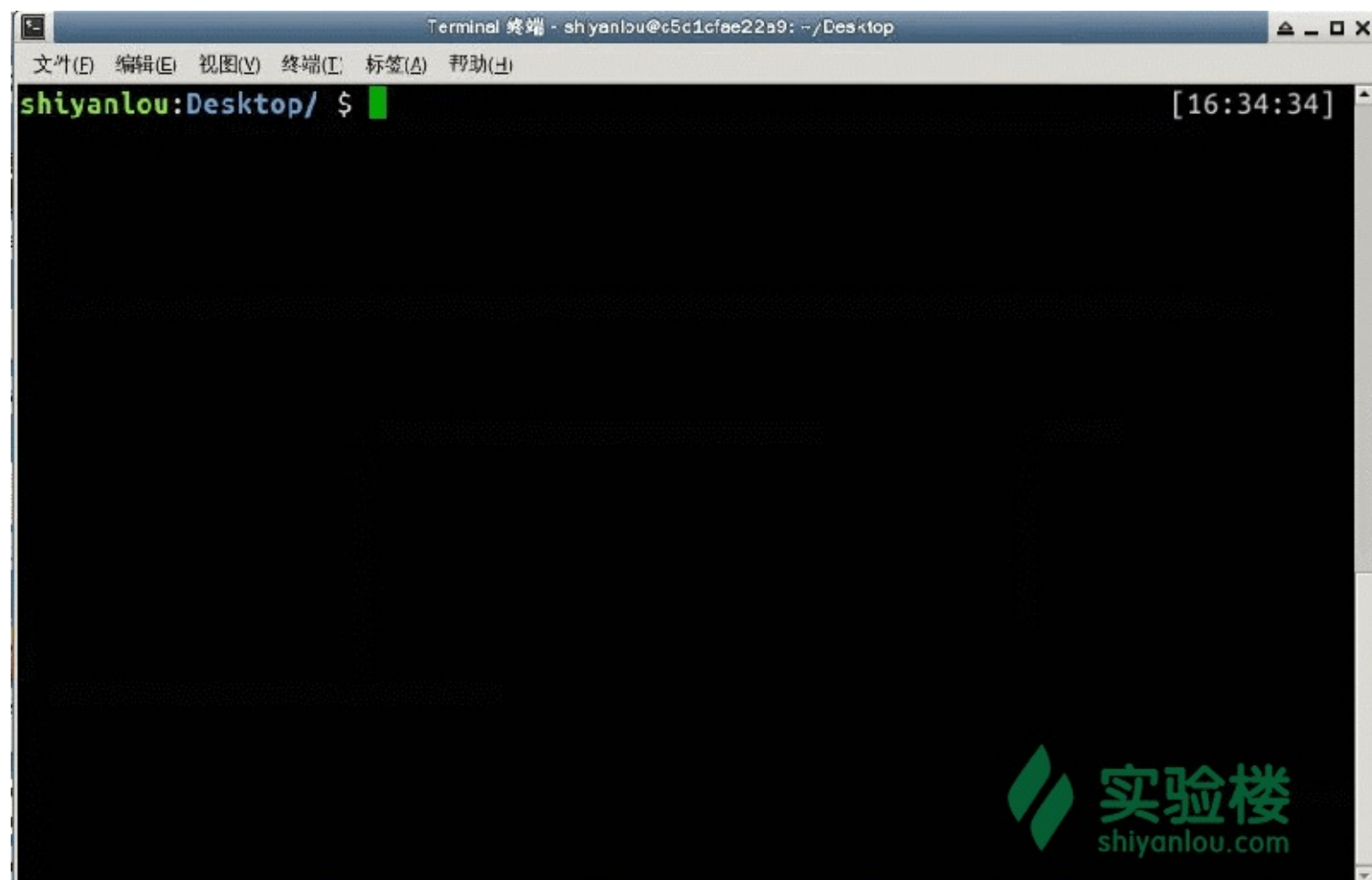
每一种编程语言都有它们自己的语法规则，就像我们所说的外语。

2.1 关键字和标识符

下列的标识符是 Python3 的关键字，并且不能用于通常的标识符。关键字必须完全按照下面拼写：

| | | | |
|----------|---------|----------|--------|
| False | def | if | raise |
| None | del | import | return |
| True | elif | in | try |
| and | else | is | while |
| as | except | lambda | with |
| assert | finally | nonlocal | yield |
| break | for | not | |
| class | from | or | |
| continue | global | pass | |

这些内容可以在 Python3 解释器中得到：



The image shows a terminal window titled "Terminal 终端 · shyanlou@c5c1cfee22a9: ~/Desktop". The window has a menu bar with options: 文件(F), 编辑(E), 视图(V), 终端(T), 标签(A), 帮助(H). The terminal prompt is "shyanlou:Desktop/ \$" with a green cursor. The time "[16:34:34]" is displayed in the top right corner. In the bottom right corner, there is a logo for "实验楼" (Shiyanlou) with the website "shyanlou.com".

在 Python 中 我们不需要为变量指定数据类型。所以你可以直接写出 `abc = 1` ，这样变量 `abc` 就是整数类型。如果你写出 `abc = 1.0` ，那么变量 `abc` 就是浮点类型。

```
>>> a = 13
>>> b = 23
>>> a + b
36
```

通过上面的例子你应该理解了如何在 Python 中定义变量，也就是只需要输入变量名和值就行了。Python 也能操作字符串，它们用单引号或双引号括起来，就像下面这样。

```
>>> 'ShiYanLou'
'ShiYanLou'
>>> 'ShiYanLou\'s best'
"ShiYanLou's best"
>>> "Hello World!"
'Hello World!'
```

2.2 从键盘读取输入

通常情况下，Python 的代码中是不需要从键盘读取输入的。不过我们还是可以在 Python 中使用函数 `input()` 来做到这一点，`input()` 有一个用于打印在屏幕上的可选字符串参数，返回用户输入的字符串。

我们来写一个程序，它将会从键盘读取一个数字并且检查这个数字是否小于 100。这个程序名称是 `testhundred.py`。还记得如何使用 `vim` 吗？忘了的话可以看看下面的动图：

shiyanlou:~/ \$

[10:33:02]



```
#!/usr/bin/env python3
number = int(input("Enter an integer: "))
if number <= 100:
    print("Your number is smaller than equal to 100")
else:
    print("Your number is greater than 100")
```

如果 number 小于 100，输出 “Your number is smaller than 100”，如果大于 100，输出 “Your number is greater than 100”。

程序运行起来就像这样:

(运行时别忘记给文件添加可执行权限，如何添加权限请回想上节实验内容，程序运行时若报错权限不够，为文件添加权限即可)

```
$ ./testhundred.py
Enter an integer: 13
Your number is smaller than 100
$ ./testhundred.py
Enter an integer: 123
Your number is greater than 100
```

下一个程序我们来计算投资：

```
#!/usr/bin/env python3
amount = float(input("Enter amount: ")) # 输入数额
inrate = float(input("Enter Interest rate: ")) # 输入利率
period = int(input("Enter period: ")) # 输入期限
value = 0
year = 1
while year <= period:
    value = amount + (inrate * amount)
    print("Year {} Rs. {:.2f}".format(year, value))
    amount = value
    year = year + 1
```

运行程序：

```
$ ./investment.py
Enter amount: 10000
Enter Interest rate: 0.14
Enter period: 5
Year 1 Rs. 11400.00
Year 2 Rs. 12996.00
Year 3 Rs. 14815.44
Year 4 Rs. 16889.60
Year 5 Rs. 19254.15
```

`while year <= period:` 的意思是，当 `year` 的值小于等于 `period` 的值时，下面的语句将会一直循环执行下去，直到 `year` 大于 `period` 时停止循环。

`Year {} Rs. {:.2f}".format(year, value)` 称为字符串格式化，大括号和其中的字符会被替换成传入 `str.format()` 的参数，也即 `year` 和 `value`。其中 `{:.2f}` 的意思是替换为 2 位精度的浮点数。

2.3 一些例子

一些关于变量和数据类型的例子。

2.3.1. 求 N 个数字的平均值

下面的程序用来求 N 个数字的平均值。

```
#!/usr/bin/env python3
N = 10
sum = 0
count = 0
print("please input 10 number:")
while count < N:
    number = float(input())
    sum = sum + number
    count = count + 1
average = sum / N
print("N = {}, Sum = {}".format(N, sum))
print("Average = {:.2f}".format(average))
```

运行程序：

```
$ ./averagen.py
1.2
3.4
3.5
33.2
2
4
6
2.4
4
5.5
N = 10, Sum = 65.2
Average = 6.52
```

2.3.2. 温度转换

在下面的程序里，我们使用公式 $C = (F - 32) / 1.8$ 将华氏温度转为摄氏温度。

```
#!/usr/bin/env python3
fahrenheit = 0
print("Fahrenheit Celsius")
while fahrenheit <= 250:
    celsius = (fahrenheit - 32) / 1.8 # 转换为摄氏度
    print("{:5d} {:.2f}".format(fahrenheit, celsius))
    fahrenheit = fahrenheit + 25
```

`{:5d}` 的意思是替换为 5 个字符宽度的整数，宽度不足则使用空格填充。

运行程序：

```
$ ./temperature.py
Fahrenheit Celsius
0    -17.78
25   -3.89
50    10.00
75    23.89
100   37.78
125   51.67
150   65.56
175   79.44
200   93.33
225  107.22
250  121.11
```

2.4 单行定义多个变量或赋值

你甚至可以在一行内将多个值赋值给多个变量。

```
>>> a , b = 45, 54
>>> a
45
>>> b
54
```

这个技巧用来交换两个数的值非常方便。

```
>>> a, b = b , a
>>> a
54
>>> b
45
```

要明白这是怎么工作的，你需要学习元组（*tuple*）这个数据类型。我们是用逗号创建元组。在赋值语句的右边我们创建了一个元组，我们称这为元组封装（*tuple packing*），赋值语句的左边我们则做的是元组拆封（*tuple unpacking*）。

下面是另一个元组拆封的例子：

```
>>> data = ("shiyancelou", "China", "Python")
>>> name, country, language = data
>>> name
'shiyanlou'
>>> country
'China'
>>> language
'Python'
```

三、总结

完成这个实验我们应该了解 python 关键字有哪些（在这里不要求全部记住），如何赋值变量，怎样从键盘读取输入，以及字符串的格式化，在这里可以了解更多有关字符串格式化的信

息：<https://docs.python.org/3/library/string.html#formatstrings> (<https://docs.python.org/3/library/string.html#formatstrings>)。最后我们接触了元组封装和元组拆封，这是一个很有用很方便的技巧，希望你能掌握它。

**本课程内容，由作者授权实验楼发布，未经允许，禁止转载、下载及非法传播。*

上一节：开始 Python 之旅 (/courses/596/labs/2036/document)

下一节：运算符和表达式 (/courses/596/labs/2038/document)