Python语言程序设计

【将数值转换成字符串】

Python Language Programming

[converting numerical values into strings]

知识点【将数值转换成字符串】

（Python2的用法）利用一对反撇（反撇和单引号不一样）

【例】 Python2用法

```
a = 123
b = 456
c = a+b
d= 'a与b作为字符连接：' + `a` +`b`
print(c) # 579
print(d) # a与b作为字符连接：123456
```

【例】 Python2和3共同用法(str())

```
a = 123
b = 456
c = a+b
d= 'a与b作为字符连接：' + str(a)
+str(b)
print(c) # a与b作为字符连接：123456
print(d)
```

Knowledge Points

[Convert a numeric value into a string]

(Usage of Python 2) Use a pair of backticks (backticks are different from single quotes)

[Example] Usage of Python 2

```
a = 123
b = 456
c = a+b
D='a and b are connected as characters:
'+` a `+` b`
print(c)
print(d)
```

579

#A and b are connected as characters: 123456

[Example] Common usage of Python 2 and 3 (str())

```
a = 123
b = 456
c = a+b
D='a and b are connected as characters:
'+str (a)+str (b)
print(c)
print(d)
```

#A and b are connected as characters: 123456

Python语言程序设计

【字符串的连接与倍增】

Python Language Programming

[String concatenation and multiplication]

知识点【字符串的连接与倍增】

字符串连接方式：

【例】字符串连接方式1　直接用 “+” 来连接两个字符串，str1 + str2

```
print (‘Deris’ + ‘Weng’ )  #  DerisWeng
```

【例】字符串连接方式2　两个字符串用“逗号”隔开，但字符串之间会多出一个空格

```
print (‘Deris’，‘Weng’ )  #  Deris Weng
```

Knowledge Points

[String concatenation and multiplication]

String connection method:

[Example] String connection mode 1

Connect two strings directly with "+", str1+str2

```
print ('Deris' + 'Weng' )   #  DerisWeng
```

[Example] String connection mode 2

The two strings are separated by commas, but there is an extra space between the strings.

```
print ('Deris',  'Weng' )   #  Deris Weng
```

知识点【字符串的连接与倍增】

字符串连接方式：

【例】字符串连接方式3

两个字符串放在一起，中间有空格或者没有空格

```
print (‘Deris’ ‘Weng’ )    #  DerisWeng
print (‘Deris’   ‘Weng’ )  #  DerisWeng
```

【例】字符串连接方式4

用符号“%”连接一个字符串和一组变量

```
print ('%s, %s' % ('Deris', 'Weng'))   #  Deris,Weng
```

Knowledge Points

[String concatenation and multiplication]

String connection method:

[Example] String Connection Mode 3

Two strings are put together with or without spaces in the middle.

```
print (‘Deris’ ‘Weng’ )    #  DerisWeng
print (‘Deris’   ‘Weng’ )  #  DerisWeng
```

[Example] String Connection Mode 4

Connect a string and a set of variables with the symbol "%"

```
print ('%s, %s' % ('Deris', 'Weng'))   #  Deris,Weng
```

知识点【字符串的连接与倍增】

字符串连接方式：

【例】字符串连接方式5

利用字符串的函数 join

```
var_list = ['Deris', 'Weng ', 'Female']
a = '***'
print(a.join(var_list))   # Deris ***Weng ***Female
```

Knowledge Points

[String concatenation and multiplication]

String connection method:

[Example] String connection mode 5

Use string function join

```
var_list = ['Deris', 'Weng ', 'Female']

a = '***'

print(a.join(var_list))   # Deris ***Weng ***Female
```

知识点【字符串的连接与倍增】

字符串连接方式：

【例】字符串连接方式6　　字符串乘法，即字符串的**倍增**

```
a = 'Weng'
print(a * 3 )   # WengWengWeng
```

Knowledge Points

[String concatenation and multiplication]

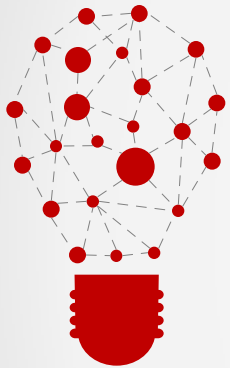
String connection method:

[Example] String Connection Mode 6

String multiplication, that is, the multiplication of strings

```
a = 'Weng'
print(a * 3 )   # WengWengWeng
```

练习 【字符串的连接与倍增】



问题:

1.若定义a = ‘A’，执行print(a*10)后，结果为 _____。

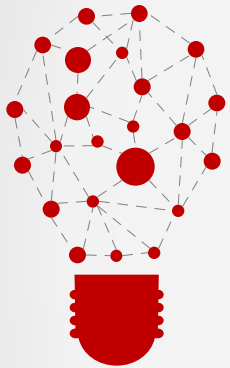
2.若定义a = ‘A’，b = ‘B’，执行print(a + b)后，结果为 _____。

3.若定义a = ‘A’，b = ‘B’，执行print(a，b)后，结果为 _____。

4.若定义a = ‘A’，b = ‘B’，执行print(ab)后，结果为 _____。

5.若定义a = ‘A’，b = ‘B’，执行print(a b)后，结果为 _____。

practise [String concatenation and multiplication]



Question:

1. If a='A' is defined, the result is after executing print (a * 10)_____.
2. If a='A' and b='B' are defined, the result is after executing print (a+b)_____.
3. If a='A' and b='B' are defined, the result is after executing print (a, b)_____.
4. If a='A' and b='B' are defined, the result is after executing print (ab)_____.
5. If a='A' and b='B' are defined, the result is after executing print (a b)_____.

Python语言程序设计

【值与类型】

Python Language Programming

[Value and Type]

知识点【值与类型】

1. 数字型：将数字分为：整数和带小数点的数。

Python3 支持 int、float、complex（复数）。

int	float	complex
9	0.0	123.45j
99	9.99	1234.j
-99	-999.9	0.12e-34j
0x123AB123EF	99.9+e99	12e+345j

Knowledge Points

[Value and Type]

1. Numerical type: Divide numbers into integers and numbers with decimals.

Python3 support int、float、complex.

int	float	complex
9	0.0	123.45j
99	9.99	1234.j
-99	-999.9	0.12e-34j
0x123AB123EF	99.9+e99	12e+345j

知识点【值与类型】

观察数据的类型

Python还提供了一个“内置函数”：

type()用来观察数据的类型：

若有定义**a=3**，执行print（ **type(a)** ）之后，结果是：**int**。

Knowledge Points

[Value and Type]

Observe the type of data

Python also provides a 'built-in function':

Type() is used to observe the type of data:

If a=3 is defined, after executing print (type (a)), the result is: int.

知识点【值与类型】

2. 字符串

【例】print（“这是一个‘单、双引号混合使用’的字符串示例”）

【例】转义字符\的使用

```
print('I\'m \"OK\"!')
```

作用：如果字符串内部既包含单引号'又包含双引号"，可以用转义字符\来标识。

[Value and Type]

2.character string

[Example] print ("This is an example of a string that uses a combination of single and double quotation marks")

[Example] Use of escape character "\"

```
print('I\'m \"OK\"!')
```

Function: If the string contains both single and double quotation marks inside, the escape character "\" can be used to identify it.

知识点【值与类型】

3. 布尔值（True、False）

```
1 print(True)
2 print(False)
3 print(3 > 2)
4 print(3 > 5)
```

输出结果：

```
True
False
True
False
```

```
1 print(True and True))
2 print(True and False)
3 print(False and
4 False)
  print(5 > 3 and 3 > 1)
```

输出结果：

```
True
False
False
True
```

Knowledge Points

[Value and Type]

3. Boolean values (True, False)

```
1 print(True)
2 print(False)
3 print(3 > 2)
4 print(3 > 5)
```

Output result:

```
True
False
True
False
```

```
1 print(True and True))
2 print(True and False)
3 print(False and
4 False)
  print(5 > 3 and 3 > 1)
```

Output result:

```
True
False
False
True
```

知识点【值与类型】

3. 布尔值（True、False）

```
1 print(True or True)
2 print(True or False)
3 print(False or False)
4 print(5 > 3 or 1 > 3)
```

输出结果：

```
True
True
False
True
```

```
1 print(not True)
2 print(not False)
3 print(not 1 > 2)
```

输出结果：

```
False
True
True
```

Knowledge Points

[Value and Type]

3. Boolean values (True, False)

```
1 print(True or True)
2 print(True or False)
3 print(False or False)
4 print(5 > 3 or 1 > 3)
```

Output result:

```
True
True
False
True
```

```
1 print(not True)
2 print(not False)
3 print(not 1 > 2)
```

Output result:

```
False
True
True
```

知识点【值与类型】

3. 布尔值

【例】布尔在条件判断中使用

```
1  if age >= 18:
2      print('成年')
3  else:
4      print('未成年')
```

[Value and Type]

3. Boolean values

[Example] Boolean is used in conditional judgment

```
1  if age >= 18:
2      print('Adult')
3  else:
4      print('Minor')
```

知识点【值与类型】

4. 空值

空值是Python里一个特殊的值，用**None**表示。

None不能理解为0，因为***0***是有意义的，而**None**是一个***特殊的空值***

5. 其他数据类型

Python还支持其他常用的数据类型，如：**List（列表）**、**Tuple（元组）**、**Sets（集合）**、**Dictionary（字典）**。

Knowledge Points

[Value and Type]

4. Empty value

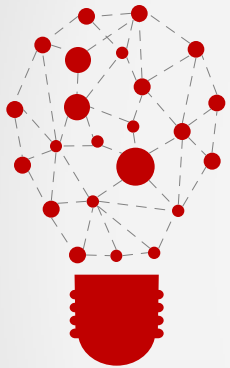
A null value is a special value in Python, represented by None.

None cannot be understood as 0 because 0 is meaningful, while None is *a special null value*.

5. Other data types

Python also supports other commonly used data types, such as **List, Tuple, Sets, Dictionary.**

课后练习【值与类型】

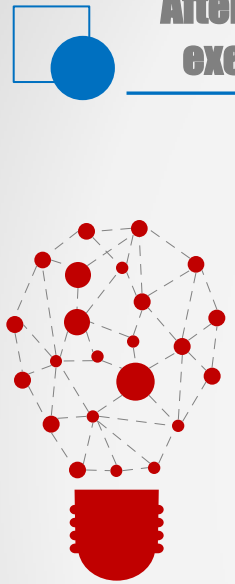


问题:

1. Python3的**数字类型**分为__int__、__float__、__complex__等子类型。

2. Python不支持的数据类型有（ **A** ）

A. char　　B. int　　C. float　　D. list



After-school exercises

[Value and Type]

Question:

1. The number types in Python 3 are divided into subtypes such as _____, ____**int**____, ____**float**____ etc. **complex**

2. The data types that Python does not support is (**A**)

A. char　　B. int　　C. float　　D. list

Python语言程序设计

【输入与输出】

Python Language Programming

[Input and output]

知识点【输入】

变量 = input（ “提示信息” ）

从**键盘读取字符串**是从用户处获取信息的一种最基本方式。

【例】input()输入

```
1  print('你叫什么名字')
2  name = input("我的名字是： ")
3  print ("你好！ "+ name.capitalize())
```