

KIT100

Programming

Preparation

Tutorial Three – Week 4

Today's Flow

- Tutorial Tasks
 - To become familiar with storing data
 - To be able to define essential Python types
 - To get experience using the Date and Time module
- Walk through Portfolio Tasks
- Take Attendance
- Your time to work on the tutorial tasks

Assigning Values

- Create a variable and assign it an integer value. Create a statement that prints the **data type** of your named variable, and a statement that prints the **value** of your variable.

```
Python 3.7.4 (default, Aug 13 2019, 15:17:50)
[Clang 4.0.1 (tags/RELEASE_401/final)] :: Anaconda, Inc. on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> a=1
>>> b='1'
>>> a_type=type(a)
>>> b_type=type(b)
>>> print(a)
1
>>> print(a_type)
<class 'int'>
>>> print(b)
1
>>> print(b_type)
<class 'str'>
>>> █
```

Assigning Values

- Write a program that will output a **Boolean** (True/False) value. An example could be to see if one number is greater than another number.
- Boolean is useful to do the decision making.

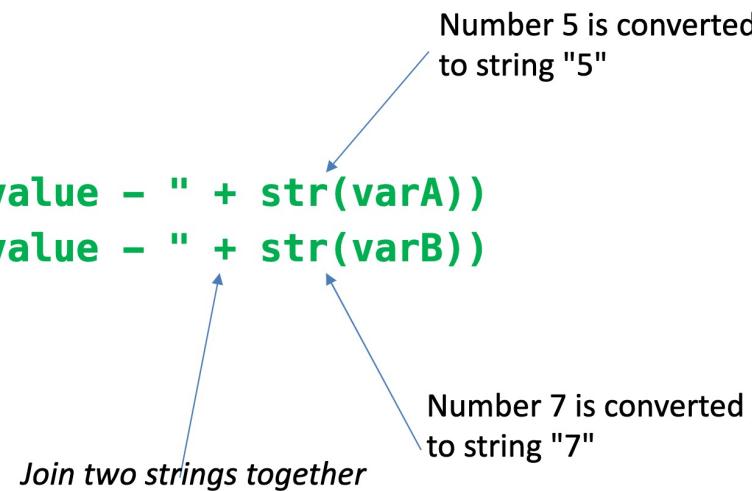
```
[>>> print(100<200)
True
[>>> print(100>200)
False
[>>> print(100==200)
False
>>> ]
```

Assigning Values

- Work through the **assigning variables** example in **Lecture 3**. Create varA and assign it the value '5'. Create varB and assign it the value '7'. Display the name and value of varA. Display the name and value of varB.

```
varA = 5
varB = 7
print ("varA current value - " + str(varA))
print ("varB current value - " + str(varB))

...
>>> varA = 5
>>> varB = 7
>>> print ("varA current value - " + varA)
Traceback (most recent call last):
  File "<pyshell#20>", line 1, in <module>
    print ("varA current value - " + varA)
TypeError: can only concatenate str (not "int") to str
>>> print ("varA current value - " + str (varA))
varA current value - 5
>>> print ("varA current value - " + str (varB))
varA current value - 7
>>>
```



Assigning Values

- Write a program that will assign the same integer value to three different variables.
Print the value of each variable.

```
>>> a = 10  
>>> b = 10  
>>> c = 10  
>>> print(a,b,c)  
10 10 10  
>>> █
```

```
>>> a = b = c = 10  
>>> print(a,b,c)  
10 10 10  
>>> █
```

Assigning Values

- Work through the example in Lecture 3 that demonstrated **simultaneous assignment** of variables.

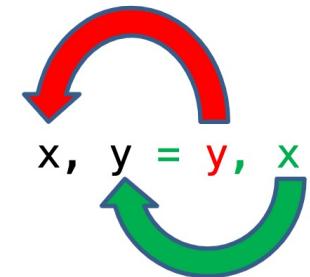
Reference: Week 3 Lecture slide pp36-37

We can then swap the values of two variables quite easily in Python!

e.g. use `x, y = y, x`

```
>>> x = 3  
>>> y = 4  
>>> print (x, y)  
3 4  
>>> x, y = y, x  
>>> print (x, y)  
4 3
```

```
>>> x = 3  
>>> y = 4  
>>> print (x, y)  
3 4  
>>> x, y = y, x  
>>> print (x, y)  
4 3  
>>>
```



Assigning Values

- Look through the follow table. Which variable names are allowed in Python?

Reference: Week 3 Lecture slide pp19

Variable Name	Allowed?
units_per_day	
dayOfWeek	
3dGraph	
June1997	
Mixture#3	

The name of a variable is

- Called an **IDENTIFIER**
- This is chosen by the programmer

Rules apply (syntax errors if broken)

- You **cannot** use one of Python's **keywords** (e.g., **if**) as a variable name.
- A variable name **cannot** contain spaces.
- The **first** character **must** be one of the letters a-z, A-Z, or an underscore character (_).
- After the first character, you **may** use the letters a-z or A-Z, the digits 0-9, or underscores.
- Uppercase and lowercase characters are distinct. This means the variable name **Items0rdered** is not the same as **itemsordered**.

```
>>> if = 5
SyntaxError: invalid syntax
>>> Jimmy Cao = 30
SyntaxError: invalid syntax
>>> 2020year = "today"
SyntaxError: invalid syntax
>>> year2020 = "today"
>>> |
```

Assigning Values

- Create a string variable called `first_name`, and a string variable called `last_name`. Assign the variables with your first name and last name, respectively. Write a statement that will display the `first_name` and `last_name` variable on the same line.

```
>>> first_name="Kai"
>>> last_name="Wong"
>>> print(first_name, last_name)
Kai Wong
>>> print(first_name + last_name)
KaiWong
>>> █
```

Assigning Values

- Create an int, a float, a Boolean, and a string variable. Determine the type of each type by using the:

type(variable)

```
>>> a=1
>>> b=1.0
>>> c='1'
>>> type(a)
<class 'int'>
>>> type(b)
<class 'float'>
>>> type(c)
<class 'str'>
>>> █
```

Assigning Values

- You are given the following program fragment:

```
[>>> a=1  
[>>> b=1.0  
[>>> result=a+b  
[>>> type(result)
```

What will be the type of variable **result**?

- a) int
- b) float
- c) string
- d) boolean

Write an algorithm that uses arithmetic operators

- Write a program that will **assign the same integer value to three different variables**. Add the first two variables together and **multiply** that result by the third variable.

Which one is correct?

1. $a = b = c = 10$
 $a + b * c$
2. $a = b = c = 10$
 $(a + b) * c$

Write an algorithm that uses arithmetic operators

- Write a program that displays the area of a rectangle with the width of 4.5 and height of 7.9 using the following formula:

$$\text{area} = \text{width} \times \text{height}$$

```
>>> width=4.5
>>> height=7.9
>>> area=width*height
>>> print(area)
35.55000000000004
>>> |
```

Creating Multiline Strings

- Write a statement that displays your name, address and email over multiple lines.
- Hints: Use triple single quote ("") to help you

```
>>> print(''  
... Kai Chiu Wong  
... Address: 11 King Street  
... Email: kaichiwong@utas.edu.au  
... '')
```

```
Kai Chiu Wong  
Address: 11 King Street  
Email: kaichiwong@utas.edu.au
```

```
>>> █
```

Python datetime module

- Remember from the lecture we can import a datetime command to get the current date and time from the computer's internal clock. To do this we first must import the datetime module.

```
import datetime  
  
print(datetime.datetime.now())
```

Supported operations:

Operation	Result
<code>t1 = t2 + t3</code>	Sum of $t2$ and $t3$. Afterwards $t1 - t2 == t3$ and $t1 - t3 == t2$ are true. (1)
<code>t1 = t2 - t3</code>	Difference of $t2$ and $t3$. Afterwards $t1 == t2 - t3$ and $t2 == t1 + t3$ are true. (1)(6)
<code>t1 = t2 * i or t1 = i * t2</code>	Delta multiplied by an integer. Afterwards $t1 // i == t2$ is true, provided <code>i != 0</code> .
	In general, $t1 * i == t1 * (i-1) + t1$ is true. (1)
<code>t1 = t2 * f or t1 = f * t2</code>	Delta multiplied by a float. The result is rounded to the nearest multiple of <code>timedelta.resolution</code> using round-half-to-even.
<code>f = t2 / t3</code>	Division (3) of overall duration $t2$ by interval unit $t3$. Returns a <code>float</code> object.
<code>t1 = t2 / f or t1 = t2 / i</code>	Delta divided by a float or an int. The result is rounded to the nearest multiple of <code>timedelta.resolution</code> using round-half-to-even.
<code>t1 = t2 // i or t1 = t2 // t3</code>	The floor is computed and the remainder (if any) is thrown away. In the second case, an integer is returned. (3)
<code>t1 = t2 % t3</code>	The remainder is computed as a <code>timedelta</code> object. (3)

Review and Practise

- Open **KIT100 Tutorial Three.pdf** on MyLO
- The items in this section are from tutorials from previous years. Feel free to test your comprehension of the unit so far and answer questions.

School of Engineering & ICT

KIT001 Programming Preparation

Tutorial 3

Aims

- to become familiar with storing data;
- to be able to define the essential Python types; and
- to get experience using the Date and Time module.

1. Assigning values

➤ Create a variable and assign it an integer value. Create a statement that prints the data type of your named variable, and a statement that prints the value of your variable.

➤ Write a program that will output a Boolean value. An example could be to see if one number is greater than another number. Review this link if you need revision on which symbol to use:
<https://www.mathsisfun.com/equal-less-greater.html>

Decision Making

Program flow

```
Run only when  
conditionA is True  
  
if (conditionA):  
    program statement  
elif (conditionB):  
    program statement  
else:  
    program statement  
  
4 space characters  
  
Run only when  
conditionB is True
```

Program flow

```
1 weight = float(input("Please input your weight in kg: "))  
2 height = float(input("Please input your height in meters: "))  
3 bmi = weight / ( height * height )  
4  
5 print("Your BMI is:", bmi)  
6 if (bmi < 18.5):  
7     print("You are underweight.")  
8 elif (bmi > 25):  
9     print("You are overweight.")  
10 else:  
11     print("You are in normal weight.")
```

```
Please input your weight in kg: 75  
Please input your height in meters: 1.7  
Your BMI is: 25.95155709342561  
You are overweight.
```

Decision Making

- You are given the following program fragment:

```
1 score = float(input("Please enter your score: "))
2
3 if (score > 0):
4     print("NN")
5 elif (score > 50):
6     print("PP")
7 elif (score > 60):
8     print("CR")
9 elif (score > 70):
10    print("DN")
11 elif (score > 80):
12    print("HD")
13 else:
14    print("HD+")
```

Please enter your score: 75

What will be the output?

1. NN
2. PP
3. CR
4. DN
5. HD

Portfolio Tasks

- **4.1PP Decision Making**

```
if (condition) :  
    program statement  
elif (condition) :  
    program statement  
else:  
    program statement
```

Purpose: simple decision making

Learning outcomes: 1, 2 and 3

Time: submit before 5pm Friday Week 4.

Resources: MyLo: lecture notes and tutorial materials

Task:

Scientists measure an object's mass in kilograms and its corresponding weight on Earth in Newtons. If you know the weight an object has in Newtons, you can calculate its mass in kilograms by using the following formula:

$$\text{mass} = \text{weight} / 9.8$$

(where 9.8 is the acceleration a mass feels due to the force of gravity)

Write a Python program that **asks the user** to enter an object's weight in Newtons, and then calculate and display its mass in kilograms. If the object's calculated mass is more than 500 kilograms, display a message indicating that it is **too heavy**. If the object's mass is less than 100 kilograms, display a message indicating that it is **too light**.

Submission Details

Upload the following to the MyLO submission folder for this task:

1. The source file (i.e. the text file containing your code)
2. A screenshot of the Python shell window that shows the **execution** results of the source code.

Assessment Criteria & Hints

A completed submission **must**:

1. Include comments about the program purpose and the author of the program (your name)
2. Declare variables and assign initial values at the start of the program
3. Use meaningful names for variables, starting with a lower case
4. Use a constant variable (all UPPERCASE) to store the value of acceleration due to gravity
5. Ask the user to enter the weight (consider what *sort of data* will be entered)
6. Calculate the mass correctly
7. Display the resulting information properly
8. Submit both the source file and the screenshot

Portfolio Tasks

- **4.2PP Decision Making with Ranges**

```
if (condition) :  
    program statement  
elif (condition) :  
    program statement  
else:  
    program statement
```

Purpose: simple decision making with data between a range of values

Learning outcomes: 1, 2 and 3

Time: submit before 5pm Friday Week 4.

Resources: MyLo: lecture notes and tutorial materials

Task:

A restaurant has controversially decided to charge customers a surcharge multiplier on their meal based on their age:

Age	Meal Surcharge
10 or less	0.95
More than 10 but less than or equal to 20	1.5
More than 20 but less than or equal to 40	2.5
More than 40	3.5

Your program should ask the user their age and then their pre-surcharge meal cost. It should then display the correct total meal cost for the user after the surcharge multiplier has been applied. For example, if the user is 27 and their meal cost was \$10.50, then the output should indicate their **total meal cost** is \$26.25 (meal cost 10.50 * meal surcharge 2.5).

Note: You will need to use appropriate formatting with the *print* function to ensure monetary amounts are shown properly (2 digits after the decimal point) - this will be discussed in lectures.

Submission Details

Upload the following to the MyLO submission folder for this task:

1. The source file (i.e. the text file containing your code)
2. A screenshot of the Python shell window that shows the **execution** results of the source code.