Hello Python!

INTRODUCTION TO PYTHON

Hoai Thuan TRAN
Gia Dinh Univerrsity

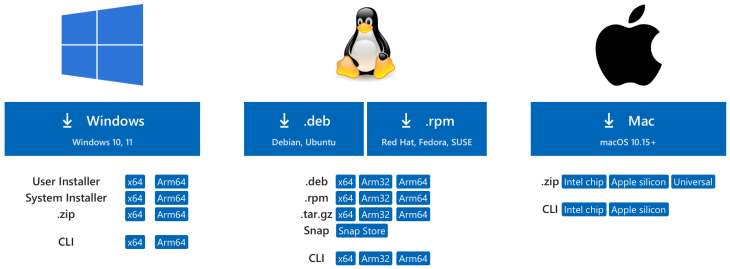
Python

- General purpose: build anything
- Open source! Free!
- Python packages, also for data science
  Many applications and fields
- Version 3.x - https://www.python.org/downloads/


```
$ python --version
```

Visual Studio Code

- Download the last version at:
  - https://code.visualstudio.com/download



- Install Python Extension

First program with Python

- Select **File -> New File** and create a file with the extension **.py**
  For example,

- Check the run result

```
Trans-MacBook-Pro:~ tranthuan$ python -u "/Users/tranthuan/Desktop/helloword.py"
Hello Word!
```

Let's practice!

INTRODUCTION TO PYTHON

Python as a calculator

- Python is perfectly suited to do basic calculations. It can do addition, subtraction, multiplication and division.

Instructions

- Print the sum of `4 + 5`.
- Print the result of subtracting `5` from `5`.
- Print the result of multiplying `3` by `5`.
- Print the result of dividing `10` by `2`.
- Above the `print( 4 + 5 )`, add the comment `# Addition`

Variables and Types

INTRODUCTION TO PYTHON

Hoai Thuan TRAN
Gia Dinh Univerrsity

Variable

- Specific, case-sensitive name
- Call up value through variable name
- **1.79** m - **68.7** kg

```
height = 1.79
weight = 68.7
height
```

```
1.79
```

Calculate BMI

```
height = 1.79
weight = 68.7
height
```

```
1.79
```

$$\text{BMI} = \frac{\text{weight}}{\text{height}^2}$$

```
68.7 / 1.79 ** 2
```

```
21.4413
```

```
weight / height ** 2
```

```
21.4413
```

```
bmi = weight / height ** 2
bmi
```

```
21.4413
```

Reproducibility

```
height = 1.79
weight = 68.7
bmi = weight / height ** 2
print(bmi)
```

```
21.4413
```

Reproducibility

```
height = 1.79
weight = 74.2 # <-
bmi = weight / height ** 2
print(bmi)
```

```
23.1578
```

Python Types

```
type(bmi)
```

```
float
```

```
day_of_week = 5
type(day_of_week)
```

```
int
```

Python Types (2)

```
x = "body mass index"
y = 'this works too'
type(y)
```

```
str
```

```
z = True
type(z)
```

```
bool
```

Python Types (3)

```
2 + 3
```

```
5
```

```
'ab' + 'cd'
```

```
'abcd'
```

- Different type = different behavior!

Let's practice!

INTRODUCTION TO PYTHON

Calculations with variables

- Instead of calculating with the actual values, you can use variables instead.
- In Python, a variable allows you to refer to a value with a name. To create a variable x with a value of 5, you use =, like this example

```
x = 5
```

Instructions

- Create a variable savings with the value of 100.
- Check out this variable by typing print(savings) in the script.
- Create a variable monthly_savings, equal to 10 and num_months, equal to 4.
- Multiply monthly_savings by num_months and save it to new_savings.
- Add new_savings to savings, saving the sum as total_savings.
- Print the value of total_savings.

Variable types

In the previous exercise, you worked with the integer Python data type:

- int, or integer: a number without a fractional part. savings, with the value `100`, is an example of an integer.
- float, or floating point: a number that has both an integer and fractional part, separated by a point. `1.1`, is an example of a float
- str, or string: a type to represent text. You can use single or double quotes to build a string.
- bool, or boolean: a type to represent logical values. It can only be True or False (the capitalization is important!).

Instructions

- Create a new float, half, with the value `0.5`.
- Create a new string, intro, with the value "Hello! How are you?"
- Create a new boolean, is_good, with the value True.

Operations with other types

In Python, different types behave differently.

- When you sum two strings, for example, you'll get different behavior than when you sum two integers or two booleans.

Instructions

- Calculate the product of monthly_savings and num_months. Store the result in year_savings.
- What do you think the resulting type will be? Find out by printing out the type of year_savings.
- Calculate the sum of intro and intro and store the result in a new variable doubleintro.
- Print out doubleintro. Did you expect this?

Type conversion

Using the + operator to paste together two strings can be very useful in building custom messages. Suppose, for example, that you've calculated your savings want to summarize the results in a string

To do this, you'll need to explicitly convert the types of your variables. More specifically, you'll need str(), to convert a value into a string. str(savings), for example, will convert the integer savings to a string.

Similar functions such as int(), float() and bool() will help you convert Python values into any type.

Instructions

Run this code try to understand the error and fix the code.

```
# Definition of savings and total_savings
savings = 100
total_savings = 150

# Fix the printout
print("I started with $" + savings + " and now have $"
+ str(total_savings) + ". Awesome!")
```