Python IT Automation: Intro to Python, Regex, and Bash Scripting

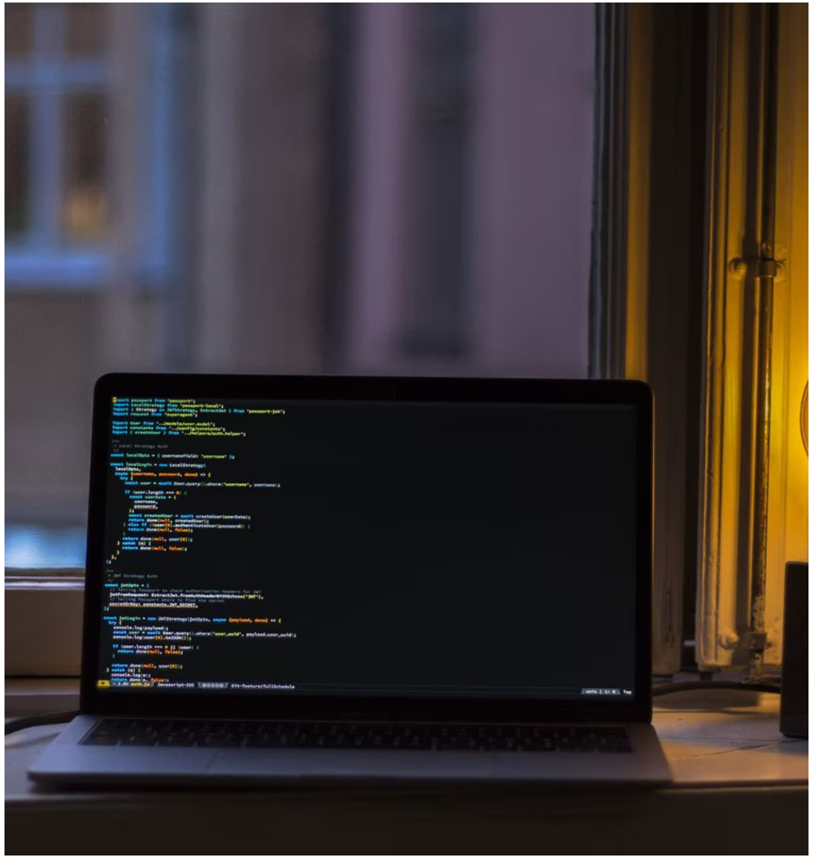
Ground Rules

Observe the following rules to ensure a supportive, inclusive, and engaging classes

- Give full attention in class
- Mute your microphone when you're not talking
- Keep your camera on
- Turn on the CC Feature on Meet
- Use raise hand or chat to ask questions
- Make this room a safe place to learn and share

Outline Session

- Hello Python
- Python Basic Syntax
- Python Data Structure
- Regular Expressions
- Managing Files, Data & Processes with Python
- Bash Scripting



Hello Python

Hello Python

What is **Python**?

Python is a dynamic, interpreted (bytecode-compiled) language.

Why programming with **Python**?

- Easy syntax
- Most chosen language for IT
- Omnipresent

Hello in Python Programming Language

```
print('hello, bangkit!')

# this is comment,
# won't be interpreted
```

Getting Ready for Python

- To check Python installed
  - open a terminal or command prompt
  - execute Python command
  - passing --version as a parameter.
- Result similar to "unrecognized command" means no Python installed.

Check on terminal (OS Linux or MacOS)

shell prompt, no need to type $

```
$ python --version
Python 3.7.9

$ python3 --version
Python 3.7.9
```

as alternative, also check python3

Python installed, version 3.7.9

Check on Command Prompt or PowerShell (OS Windows)

command prompt, no need to retype

```
C:\Users\bangkit> python
--version
Python 3.7.9
```

Python installed, version 3.7.9

Basic Syntax

Basic Python Syntax: Data Types

- String (str): text.
- Integer (int): numbers, without fraction.
- Float: numbers with fraction.
- Boolean (bool): data type which only has 2 values

We can convert from one data type to others by committing to implicit conversion or defining an explicit conversion.

Basic Python Syntax: Variables

- Name to certain values
- The values can be any data type
- The process of storing a value inside a variable is called an assignment.
- Can only be made up of letters, numbers, and underscore.
- Can't be Python **reserved keywords**.

Do This

```
length = 10
width = 2
area = length * width
name = ‘Saturnus’
print(area)
print(type(width))
print(type(str(area)))
print(name)
```

Don't Do This

Cannot be used, will expected error

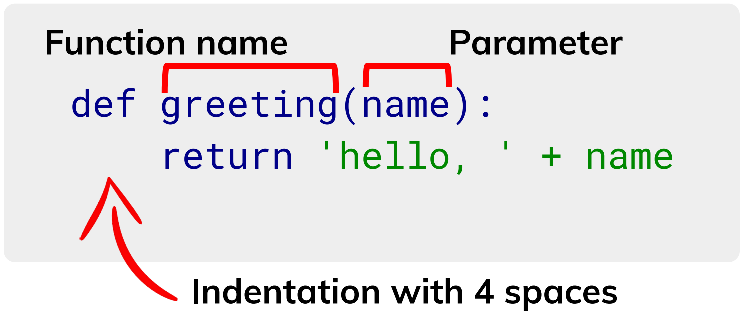
```
def = ‘Function’
class = ‘Class’
print(and)
print(or)
```

bangk!t

Basic Python Syntax: Functions

- Define function with **def** keyword.
- Function has body, written as a block after colon in function definition. The block has indented to the right.
- To get value from a function use the **return** keyword.

Define Function, to Return String



```
def greeting(name):
    return 'hello, ' + name
```

Call Function

```
print(greeting('bangkit'))
```

Output

```
hello, bangkit
```

bangkit

Conditional & If Statements

- The ability of a program to alter its execution sequence is called branching.
- The if block will be executed **only if the condition is True**.
- Use elif & else statement to handle multiple conditions.

Condition Evaluations

The Condition of Comparison

```
if hour < 12:
    print("Good morning!")
```

Indentation with 4 spaces

```
def check(number):
    if number > 0:
        return "Positive"
    elif number == 0:
        return "Zero"
    else:
        return "Negative"
```

Loops: while & for

- while loop instruct computer to continuously execute code based on the value of a condition.
- for loop iterates over a sequence of values.

while loop

Initialization of variable

```
x = 7  # also try with x = 0
```

The conditions

```
while x > 0:
    print("positive x=" + str(x))
    x = x - 1
print("now x=" + str(x))
```

for loop

The sequence

```
for x in range(3):  # 0, 1, 2
    print("x=" + str(x))
```

Loops: break & continue

- Both while and for loops can be interrupted using the break keyword.
- Use the continue keyword to skip the current iteration and continue with the next one.

break from loop

```python
for x in range(3):
    print("x=" + str(x))
    if x == 1:
        break  # quit from loop
```

continue inside loop

```python
for x in range(3, 0, -1):
    if x % 2 == 0:
        continue  # skip even
    print(x)
```

bangkit

Python Data Structure

Data Strings

- Represent a piece of text.
- To access substring, use index or slicing.
- Strings in Python are immutable
- Provide a bunch of methods for working with text.

Strings

```
name = 'bangkit'
program_year = "it's the 2nd"
multi_line = """hello,
email test. Signature."""
                  Index
print(name[1])  # a
                                      Slicing
print(name[4:len(name)-1])  # ki

year = "it's 2021"
year[-1] = "0"  # TypeError

print(name.upper())  # BANGKIT
```

* Complete string methods in Python docs https://docs.python.org/3/library/stdtypes.html#string-methods

List

- In Python list can contain a different value.
- Python use square brackets [] to indicate where the list starts and ends.
- List in Python are mutable.

List

list starts

list ends

```python
paths = ['ML', 'Cloud']
for path in paths:
    print(path)  # element per line

paths.append('Android')
paths.remove('Cloud')
paths.insert(1, 'Mobile')
paths.pop(-1) # remove 'Android'

# change 'ML' to 'Machine Learning'
paths[0] = 'Machine Learning'

# list comprehensions
even = [x*2 for x in range(1,5)]
print(even)  # [2, 4, 6, 8]
```

Tuples & Dictionary

- Tuples can contain elements of any data type. But, unlike lists, tuples are immutable.
- Dictionary in Python contain **pairs of keys and values**.
- To get a dictionary value, use its corresponding key.
- Dictionary in Python are mutable.

Tuple

tuples starts

tuples ends

```
paths = ('ML', 'Cloud')
for path in paths:
    print(path)  # element per line
```

Dictionary

dictionary key

dictionary value

```
students = {'ml': 500,'mobile': 700,
'cloud': 900}
print(students['cloud'])  # 900

students['ml'] = 1000
```

Regular Expression

Regular Expressions

- Regex is a search query for text that’s expressed by string pattern.
- Regular expressions in Python uses raw string (r"")
- Circumflex (^) pattern matches the beginning of the line.
- Dot (.) matches any character.

Simple Matching in Python re

```
import re

result = re.search(r"aza", "plaza")
print(result)
<re.Match object; span=(2, 5),
match='aza'>
print(re.search(r"aza", "maze"))
None

print(re.search(r"^x", "xenon"))
<re.Match object; span=(0, 1),
match='x'>

print(re.search(r"p.ng", "sponge"))
<re.Match object; span=(1, 5),
match='pong'>
```

bangkit

Regular Expressions

- To matched a range of characters, use another feature of regexes called character classes ([ ]).
- Use the pipe symbol (|) to match one expression or another.
- Dollar sign ($) pattern match the end of the line.

```
import re

print(re.search(r"cloud[a-zA-Z0-9]",
"cloud9"))
<re.Match object; span=(0, 6),
match='cloud9'>

print(re.search(r"[^a-zA-Z]", "This is a
sentence."))
<re.Match object; span=(4, 5), match=' '>

print(re.search(r"cat|dog", "I like cats."))
<re.Match object; span=(7, 10), match='cat'>

print(re.search(r"cats$", "I like cats"))
<re.Match object; span=(7, 11),
match='cats'>
```

bangkit

Regular Expressions

Repeated matches is another regex concept.

- The star (*) **takes as many character as possible.**
- The plus (+) character **matches one or more occurrences of the character before it**.
- The question (?) **mark symbol means either zero or one occurrence of the character** before it.

```
import re

print(re.search(r"Py[a-z]*n",
"Python Programming"))
<re.Match object; span=(0, 6),
match='Python'>

print(re.search(r"o+l+", "woolly"))
<re.Match object; span=(1, 5),
match='ooll'>

print(re.search(r"p?each", "I like
peaches"))
<re.Match object; span=(7, 12),
match='peach'>
```

bangkit

* More about regex introduction visit https://www.regex101.com

Managing Files Using Python

bangk!t

Managing Files with Python

- Function open will start to open the file.
- To read file, use the readline & read function.
- To ensure that all open files are always closed, use an alternative method to write it as a block of code using the with keyword.

Read Existing File

```
file = open("spider.txt")
print(file.readline())
file.close()

The itsy bitsy spider climbed
up the waterspout.
```

open mode

```
with open("spider.txt", “r”) as
file:
    print(file.read())
The itsy bitsy spider climbed
up the waterspout.
Down came the rain
and washed the spider out.
```

bangkit

* Complete modes for function open in Python docs https://docs.python.org/3/library/functions.html#open

Managing Files with Python

- Use **os** module to interact with operating system in Python

- To read and writing tabular data in CSV format, use an **csv** module.

Working with Directory

```
import os
os.mkdir("new_dir")
```

Reading CSV Files

```
import csv
file = open("data.csv")
csv_f = csv.reader(file)
for row in csv_f:
    name, phone, role = row
    print(name+': '+role)
file.close()
```

```
Sabrina Green: System Administrator
Eli Jones: IT specialist
```

bangkit

* Python Documentation for module os & csv
https://docs.python.org/3/library/os.html, https://docs.python.org/3/library/csv.html

Managing Data & Processes

Three different I/O streams by default:

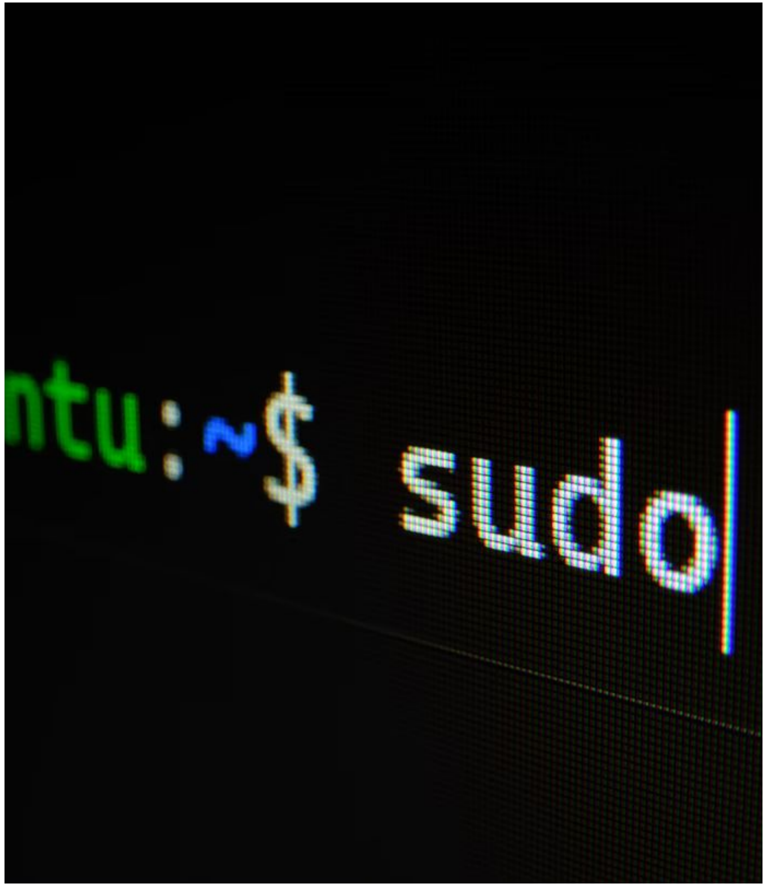
- Standard input stream (**STDIN**): the input function.
- Standard output stream (**STDOUT**): the print function.
- Standard error stream (**STDERR**): specifically as a channel to show error messages and diagnostic from the program.

Bash Scripting

bangkit

Why Bash Scripting?

- Flexible
- Less Resource
- Used in cloud environment
- Automate command



Most Commonly Used Bash Command

- Linux commands:
  - **echo**: print information (like environment variable) to standard output
  - **cat** file: shows the content of the file through standard output
  - **ls**: lists the contents of the current directory
  - **cd** directory: change current working directory to the specified one
  - **rm**: remove file or directory (with specific arguments)
  - **chmod** modifiers files: change permissions for the files according to the provided modifiers
  - **man:** show command documentation

Sharing Session

bangkit

Demo Link

Demo basic python + regex:
https://colab.research.google.com/drive/1k3R9xx_-cTIJvU9bqu9Vf2IlgpC1bwcV?usp=sharing

Discussions

Quiz

Thank You

bangkit