PYTHON - INTRODUCTION

Bendik Egenes Dyrli

UiA University of Agder

OVERVIEW

1. What is Python
2. Glossary
3. Syntax
4. Syntax Errors
5. Libiaries

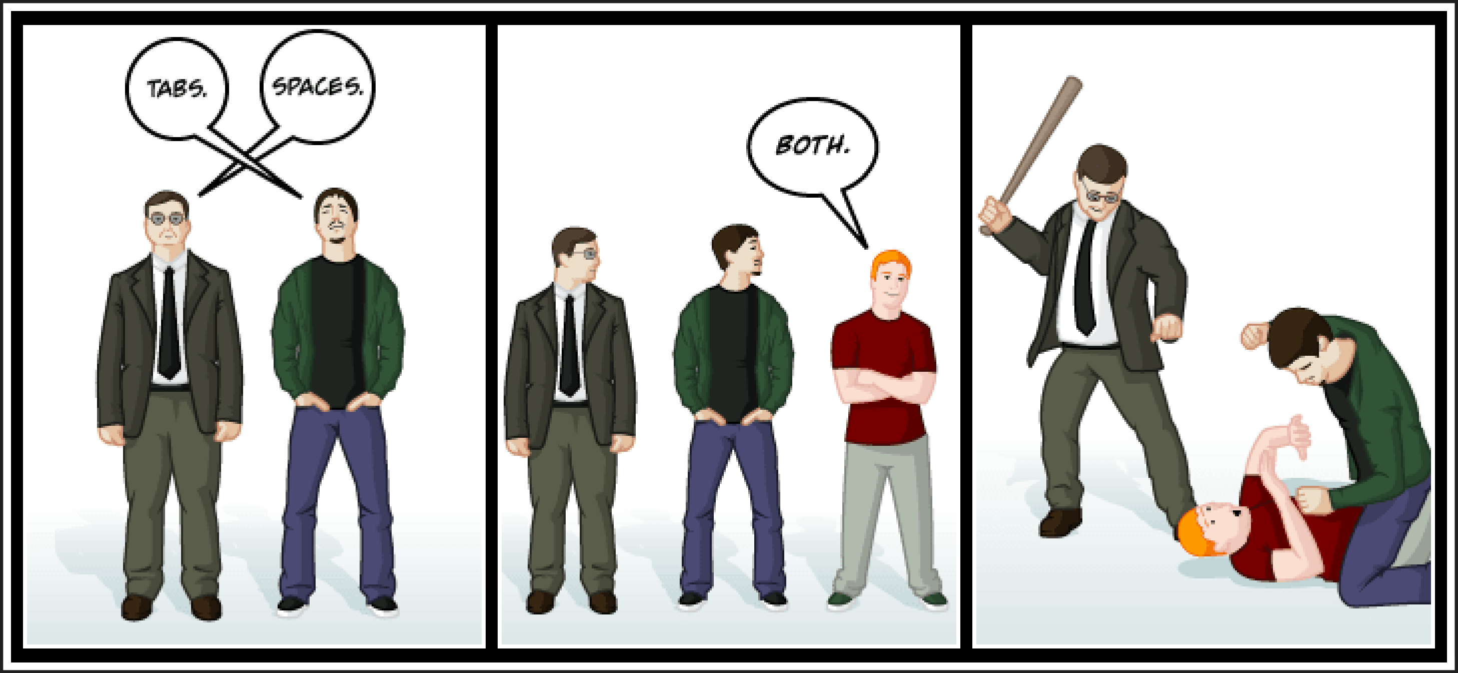
WHAT IS PYTHON?

- Interpreted
- Object Oriented
- High-level
- Garbage Collected
- Guido van Rossum
- Spaces vs Tabs

WHO USES PYTHON

- Instagram
  - Web
  - Django, Python
- Spotify, Netflix
  - Data analysis
- Dropbox
  - Desktop Client

TABS VS SPACES



- Spaces are the preferred indentation method
- Python 3 disallows mixing the use of tabs and spaces for indentation.

GLOSSARY

IDLE

An Integrated Development Environment for Python. IDLE is a basic editor and interpreter environment which ships with the standard distribution of Python.

2TO3

A tool that tried to convert python 2.x to python 3.x code.

ARGUMENTS

- Keyword Arguemnt

```
complex(real=3, imag=5)
```

- Postitional Argument

```
complex(3,5)
```

DUCK-TYPING

If it looks like a duck and quacks like a duck, it must be a duck.

__FUTURE__

A Pseudo-module which programmers can use to enable new language features which are not compatible with the current interpreter

```
from __future__ import braces

print (braces)
```

GARBAGE COLLECTION

The process of freeing memory when it is not used anymore.

GIL

Global Intrepreter Lock

is a mutex that allows for only one thread to hold the control of the Python interpreter.

PEP

Python Enhancement Proposal.

A PEP is a design document providing information to the Python community, or describing a new feature for Python or its processes or environment.

PEP20

THE ZEN OF PYTHON

1. *Beautiful is better than ugly.*
2. *Explicit is better than implicit.*
3. *Simple is better than complex.*
4. *Complex is better than complicated.*
5. *Flat is better than nested.*
6. *Sparse is better than dense.*
7. *Readability counts.*
8. *Special cases aren’t special enough to break the rules.*
9. *Although practicality beats purity.*
10. *Errors should never pass silently.*
11. *Unless explicitly silenced.*
12. *In the face of ambiguity, refuse the temptation to guess.*
13. *There should be one—and preferably only one—obvious way to do it.*
14. *Although that way may not be obvious at first unless you’re Dutch.*
15. *Now is better than never.*
16. *Although never is often better than right now.*
17. *If the implementation is hard to explain, it’s a bad idea.*
18. *If the implementation is easy to explain, it may be a good idea.*
19. *Namespaces are one honking great idea—let’s do more of those!*

SYNTAX

SHA-BANG!

```
#!/usr/bin/env python3

#!/bin/python3
```

VARIABLES

```
name="Rick"

age=42

occupation="Time travler"
```

VARIABLES

```
str_age = "42"
age = 42
```

TYPES

```
age = 42

aage = 42.0

aaage = "42.0"

aaaage = True
```

TYPES

```
string = str(string)

integer = int(integer)

flyttall = float(flyttall)

boolean = bool(boolean)
```

TYPES

```
text = "Hello my pincode is "
pincode = 3301

print (text + pincode)
```

TYPES

```python
text = "Hello my pincode is "
pincode = 3301

print (text + pincode)
```

```
Traceback (most recent call last):
  File "/home/skandix/types.py", line 4, in <module>
    print (text + pincode)
TypeError: can only concatenate str (not "int") to str
```

TYPES

```python
text = "Hello my pincode is "
pincode = 3301

print (text + str(pincode))
```

```
Hello my pincode is 3301
```

IF

```
RickSober = False
MortyLikesJessica = True
JessicaLikesMorty = None

if (RickSober != True):
    print("Rick is not sober")
    if (MortyLikesJessica and JessicaLikesMorty):
        print("Morty and Jessica Likes each other")
    else:
        print("Jessica doesn't like Morty :( ")

elif (RickSober == True):
    print("WHAT, Rick is sober... what dimension is this ?")
```

INPUTS

```
take_a_guess = input('-->')

if input == 9:
    print ("Congrats... you knew the magic number")
else:
    print("sorry, no prize for you.")
```

STRINGS

```
name = "Rick"
age = 42

# String interpolation
print ("My name is " +name+ " and i'm " +age+ " years old")

# str.format
print ("My name is {name}, and i'm {age} years old".
                           format(name=name,age=age))
# F(ormat)-strings
print (f"My name is {name}, and i'm {age} years old.")
```

FOR

```python
languages = ['Lua', 'Bash', 'Python']
for lang in languages:
    print(lang)
```

FOR

```
for number in range(10):
    print (number)
```

FOR

```
string = "Can we iterate over strings"
for word in string:
    print (word)
```

FOR

```python
counter = 0
for number in range(10):
    counter += 1
    print ("on ", counter, "and counting")
```

LEN

```
state = "mississippi"
print(len(state))
```

HELP

```
help(len)
```

SYNTAX ERROR

```
Morty = "HEEEYY, Rick can i iterate over string?"
for word in Morty
    print (word)
```

SYNTAX ERROR

```
Rick = "YES MORTY YOU CAN!"
for word in Rick:
    print (word<Paste>
```

SYNTAX ERROR

```
print("SHIT, MORTY WHAT DID YOU DO!?!?!")
```

LIBARIES

IMPORT RANDOM

Generate pseudo-random numbers

```python
from random import randint

print(randint(1,10))
```

5 MINUTE TASK

Make a program that prints out random 4length pincodes

5 MINUTE TASK

Make a program that prints out random 4length pincodes

```
from random import randint

pincode = ""
for _ in range(4):
    pincode += str(randint(1,9))

print (pincode)
```

IMPORT ARGPARSE

```
import argparse

parser = argparse.ArguemntParser()
parser.add_argument("--name", "-n", type=str, help="What is yo
parser.add_argument("--age", "-a", type=int, help="What is you

print (f"Hello my name is {args.name}, and i'm {args.age} year
```

RESOURCES

https://docs.python.org/3/glossary.html#glossary
https://instagram-engineering.com/web-service-efficiency-at-instagram-with-python-4976d078e366
https://www.youtube.com/watch?v=66XoCk79kjM
https://docs.python.org/3/tutorial/appetite.html
https://docs.python.org/3/tutorial/interpreter.html
https://docs.python.org/3/tutorial/introduction.html