Created by: ifm CSR
www.ifm3d.com

Python basics (on Windows)

Scripts

Definitions

- What is a script: a program, written with a scripting language.
- What is a scripting language = interpreted language: a language that is not compiled. It is translated into “machine talk” at runtime. A compiled language is first translated into machine language and then executed.
- What goes in a script: code. The exact same code you’ve been using in a command line interface.
- Why would you use scripts: you can write the script once and execute it any number of time. You can have multiple scripts for multiple tasks and run them at the same time, or choose the one to use depending on the task to accomplish.

In practice

- How to execute a script called script.py:

```
python.exe script.py
OR
./python.exe script.py
```

- How to handle errors

Read the error message, it contains valuable information.

```
(o3r-venv) PS C:\Users\usmasslo\Desktop\winpython\python-3.7.4.amd64>
.\python.exe ..\py-scripts\script.py
  File "..\py-scripts\script.py", line 3
    device =Device(“192.168.0.69”, 50012)
                   ^
SyntaxError: invalid character in identifier
```

In the example above, the wrong quotation marks have been used around the IP address: “ instead of ".

An error, if not handled inside the script, will interrupt the execution: you need to correct the source of the error and launch the script again.

Tip:

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

Add the line above at the beginning of your script. This will "force" your operating system to use Python 3.

Example:

```
from ifmO3r.ifm3rTiny import Device

device2 = Device(“192.168.0.69”, 50012)
device3 = Device(“192.168.0.69”, 50013)

device2.config_from_json_file(“config.json”)
device3.config_from_json_file(“config.json”)
```

In the example above, we create two `Device` objects for the 3D ports 2 and 3, and configure them from a file called `config.json` holding both configurations and located in the current working directory.

Basic python syntax

- Tab/spaces serve as a "level" separator: every line starting with the same spacing constitutes a logical block.

WARNING: do not mix spaces and tabs!

Examples: see while and for sections.

- Parenthesis when calling functions/classes: function (for instance `config_from_json_file()`) and classes (for instance `Device()`) require parenthesis when called. You can think of these as placeholders for passing in variables (the IP address for the `Device` for instance). Some function or classes do not require any variable input, but you still have to use the parenthesis.
- Column at the beginning of a "block": don't forget the column when starting a loop!

While True loop

A `while True:` loop is a feature that executes code *forever*, or until a stop signal is sent (CTRL+C in most cases).

Example 1 :

```
while True:
    print(“The O3R is great!”)
```

Try it!

Example 2:

```
...
i = 0           # initializing a counter
while True:
    im_log.get_current_frame()
    im_log.write_png({"distance": "distance_frame"+str(i)})
    i += 1      # incrementing the counter
```

Explanations:

- `str(i)` → converts the variable `i` to a string (multiple characters, letters, special characters or numbers).
- `{"distance": "distance_frame"+str(i)}` → the name of the saved file will have the number of the iteration appended at the end of it, for instance `distance_frame1` for the first saved frame.

Watch out for the amount of data you are saving → it will grow

Remember: stop the loop with `CTRL+C`.

Let's put this in a script file, save_loop.py:

```
from ifmO3r.ifm3dTiny import Device, FrameGrabber, ImageBuffer, ImageLogger
device =Device("192.168.0.69", 50012)
fg = FrameGrabber(device)
im = ImageBuffer()
im_log = ImageLogger(fg, im)
i = 0  # initializing a counter

# Let's set the framerate to 8Hz to keep the amount of data saved under
control:
device.config_from_json_str('{"Port2":{"mode":{"modeParam":{"framerate":
8}}}}')

while True:
    im_log.get_current_frame()
    im_log.write_png({"distance": "distance_frame"+str(i)})
    print("Saved frame "+str(i))
    i += 1  # incrementing the counter
```

Launch the script:

```
python while_loop.py
```

Stop the script after a few seconds:

```
CTRL+C
```

For loop

A `for loop` executes code a certain amount of iterations.

Example: save 10 frames

```
...
i = 0
for i in range(0,10):        #loop 10 times
    imgLogger.get_current_frame()
    imgLogger.write_png({"distance": "distance_frame"+str(i)})
    i += 1
```

Use the code from the previous section and adapt it to use the `for` loop instead of the `while` loop. The script is finishing after saving the 10 frames, but if you want to stop it earlier, you can still use `CTRL+C`.

Resources and tools

Code editor:

A good code editor can be really useful and help you detect syntax errors. We like Atom and VSCode (a much more advanced tool), but there are a ton of them ou there. Try one or two out and see what you think.

Documentation and online classes:

- RealPython
- w3schools
- Python docs
- Python Cheat Sheets
- Facing an unknown problem? Want to learn more about a specific feature? Try Googling it. This is what we do! But as always, don't hesitate to reach out to us with questions.

What is next?

Next steps to gain a deeper understanding of the O3R:

- Understand how the data is structured: multi-dimensional arrays, different formats (hdf5, etc), ...
- Understand how to analyze the data: display saved data, get statistics of one pixel over time, ...

Interested? Let us know and we will setup some support around these topics.