cct | College Dublin
Computing • IT • Business

PROGRAMMING: OBJECT-ORIENTED APPROACH 

PYTHON MODULES

- Press `Space` to navigate through the slides
- Use `Shift+Space` to go back
- Save as **PDF**:
  - Open **Chrome** then **click here**
  - Press **Ctrl+P**/**Cmd+P** to print
    - Destination: **Save as PDF**
    - Layout: **Landscape**
  - Press **Save** button

1

MODULES IN PYTHON

- Modules are a way of packaging and segmenting code to help with organization.
- Python comes with a huge selection of modules already built-in called the python standard library.
- This includes everything from ways to access the computers operating system (the os module) to random number and choice generators (the random module), to even an etch-a-sketch style drawing module called turtle.

MODULE IMPORTS

- In python you can import modules in a few different ways.
- For example, in python there is a math module for doing math operations:

```
import math # Brings the whole math module in

print(math.sqrt(4)) # Prints the square root of 4, which is 2
```

- One thing you will notice with this is that I had to type `math.sqrt()` to call the function.
- This is because I imported the whole module, which means I need to tell python what to run, and from which module.
- A better way of doing this example would be to use the *from module_name import function*.

MODULE IMPORTS

- Using the previous math example from before:

```
from math import sqrt # brings in JUST the sqrt() function from the math module

print(sqrt(4)) # Prints the square root of 4, which is 2
```

- Another neat trick is to import a function with an alias (another name). Using the previous example we could also write:

```
# brings in JUST the sqrt() function from the math module and aliases it to square
from math import sqrt as square_root

print(square_root(4)) # Prints the square root of 4, which is 2
```

MODULE IMPORTS

- Modules are essential to building larger applications, but now that you know how to use them let's look at what they actually are.
- Modules are just plain old python files, buried deep inside the guts of your python installation there is a file called math.py with a function called sqrt().

WRITING YOUR OWN MODULES

- There are actually many ways to write your own modules, but the simplest is just to link separate python files together inside the same folder.
- Let's assume you have a project that is structured like this:

```
├── /project
|    ├── my_program.py
|    └── my_module.py
```

- Now lets assume your files look like this: `my_module.py`

```
def my_module_function():
    print("Hello from my_module.py")
```

`my_program.py`

```
from my_module import my_module_function # import my_module_function() from the my

if __name__ == "__main__":
    my_module_function() # Run my_module_function() from my_module.py
```

- As you can see you can import 'modules' by using the filename without the .py.

6

WRITING YOUR OWN MODULES

- The following explanation of modules is simplified.
- If you are looking at writing your own module to put online please read a more in-depth guide about packaging: https://packaging.python.org/tutorials/packaging-projects/

PIP; THE PYTHON PACKAGE MANAGER

- In python the most popular way of managing modules (sometimes called packages) is with a utility called pip.
- If you have followed my instructions for installing python you should already have pip installed.
- You can find out if it's installed by typing `pip`(windows) or `pip3` (linux/mac) into your command line.
- If you do not have pip installed, please go back and look at the installation instructions in module 0.

PIP; THE PYTHON PACKAGE MANAGER

- pip allows you to download and run modules that other people have created on pypi (The python package index), or install modules that you have created specifically for pip.
- Using pip you can get access to thousands of modules that do everything from machine learning, to web development, to interfacing with other services.
- The simplest way to use pip is to just type `pip install <project name>` on windows or `pip3 install <project name>` on mac/linux.

REQUIREMENTS.TXT

- It is common in python projects to store a list of dependencies necessary to run a project in a file called `requirements.txt`.
- For example lets say you have a project that requires you to install requests, tqdm and Flask you can write a requirements.txt file that would look like this:

```
requests
tqdm
Flask
```

- Once you have the file you can get pip to install everything listed by using the -r (read from file) flag. i.e. `pip install -r requirements.txt`.

PYPI. (THE PYTHON PACKAGE INDEX)

- PyPI. is a website that hosts python modules (sometimes called packages).
- In every previous example of using a module that is not part of the Python Standard Library, we have been installing packages from PyPI.
- For example Flask is hosted by PyPI. which is why when we type pip install flask, what actually happens is that pip looks up flask on PyPI. and installs it from there.
- Typically most packages can be installed through pip using PyPI.