

ACIT 2515

Python best practices: code structure

Conventions and best practices

- See the [PEP8](#) for recommendations about the syntax
- Use linting / formattings tools:
 - `black`
 - `flake8`
 - `autopep8`
 - etc.
- Naming conventions for variables / functions:
 - use `lowercase` for all variables and functions
 - use `UPPERCASE` for constants
 - use `lower_case_with_underscores` for variables and functions
 - `myDictValue = 10` is **not** a good practice in Python

Docstrings and documentation / comments

- Good code is commented code.
- In your functions and files, write docstrings (`""" text ... """`) to inform potential users about what your code does.
- Add comments to explain complicated steps in your program.

Example:

```
def my_func(a, b):  
    """Takes two arguments and returns their average (sum / 2)"""  
  
    output = float(a) + float(b)    # We force conversion to float  
    return output / 2
```

- Separate code into functions
- Separate functions into modules
- Organize modules into packages
- Don't forget about `__init__.py` files!
- Use `if __name__ == "__main__":` in your modules
- Create a single entrypoint in your program, and then use modules / packages to organize the logic

```
├── component_1
│   ├── __init__.py
│   └── sub_package
│       ├── __init__.py
│       ├── sub_module_1.py
│       └── sub_module_2.py
├── component_2
│   ├── __init__.py
│   └── [... files ...]
├── component_3
│   ├── __init__.py
│   └── [... files ...]
```