



DocStrings in Python

- DocStrings are strings that appear after the definition of a function, module, or class.
- They are used to document the purpose and usage of the function/module/class.

```
LEMENT_ARRAY_BUFFER, indexBufferObj);
LEMENT_ARRAY_BUFFER, 6 * sizeof(unsigned int), indices

der = createShader(readFile("res/shaders/vertex.shader",
s/fragment.shader"));
der);
glGetUniformLocation(shader, "u_color");

owShouldClose(window)) {
e buffer
COLOR_BUFFER_BIT);

e background color
or(1.0f, 1.0f, 1.0f, 1.0f);

am(shader);
4f(location1, 85.0f*INV_255, 184.0f*INV_255, 237
texArray(vertexArrayObj);
ments(GL_TRIANGLES, 6, GL_UNSIGNED_INT, NULL);

ay bound buffer
Buffers(window);
```

DIFFERENCE BETWEEN DOCSTRINGS & COMMENTS

DOCSTRINGS

Used for documenting functions, classes, and modules.

Written using triple quotes `""" """`.

Can be accessed at runtime using `__doc__`.

COMMENTS

Used for explaining code and making it readable.

Written using `#` at the beginning.

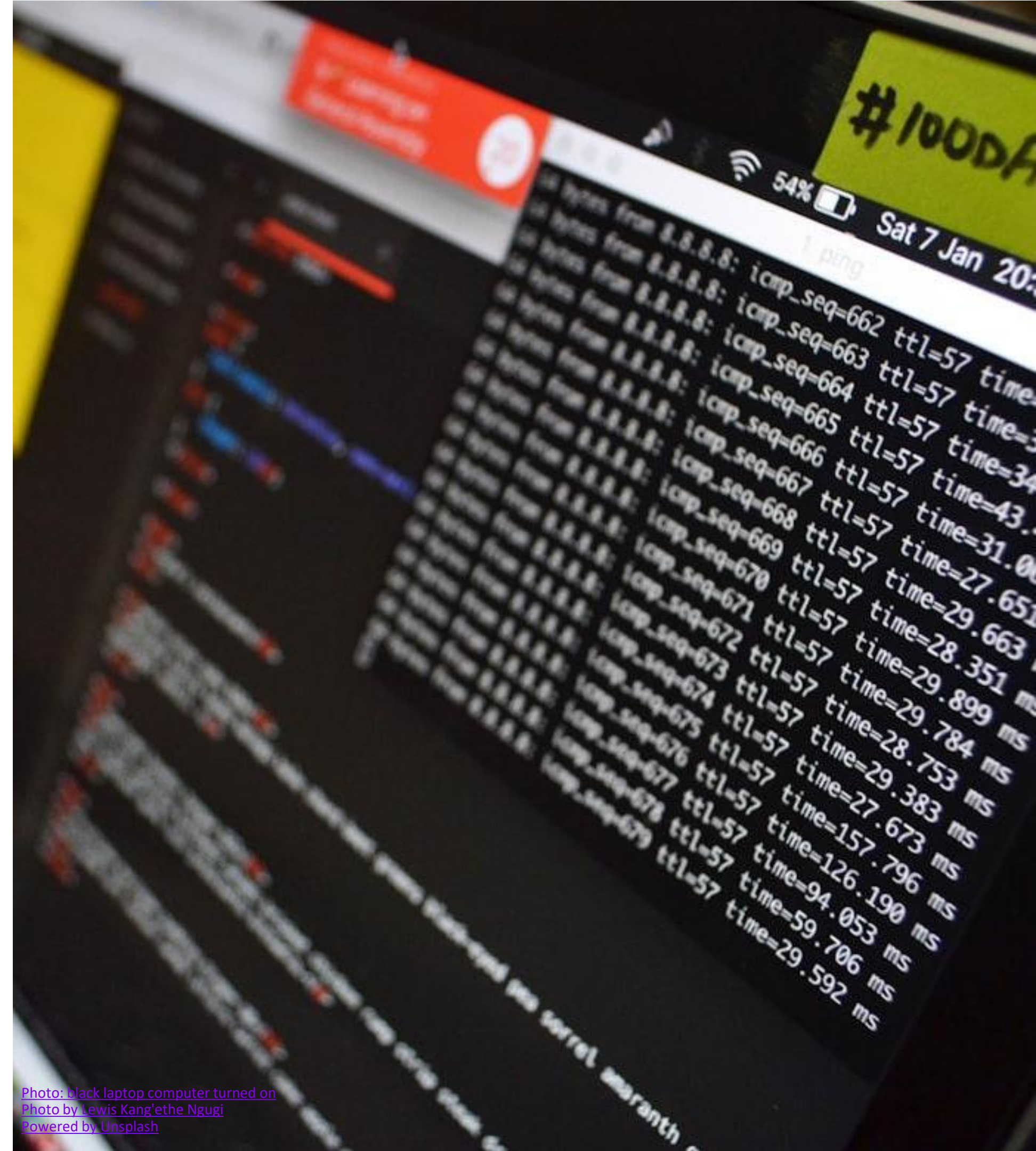
Not accessible at runtime.



Example 1 - Correct Docstring Usage

Example:

```
def abc(a):  
    """This is a docstring"""  
    print(a)  
abc(4)  
print(abc.__doc__)
```





Example 2 - Incorrect Docstring Usage

- Example:

```
def abc(a):  
    print(a)  
    '''This is not a valid docstring'''
```
- The docstring should be placed immediately after the function definition.





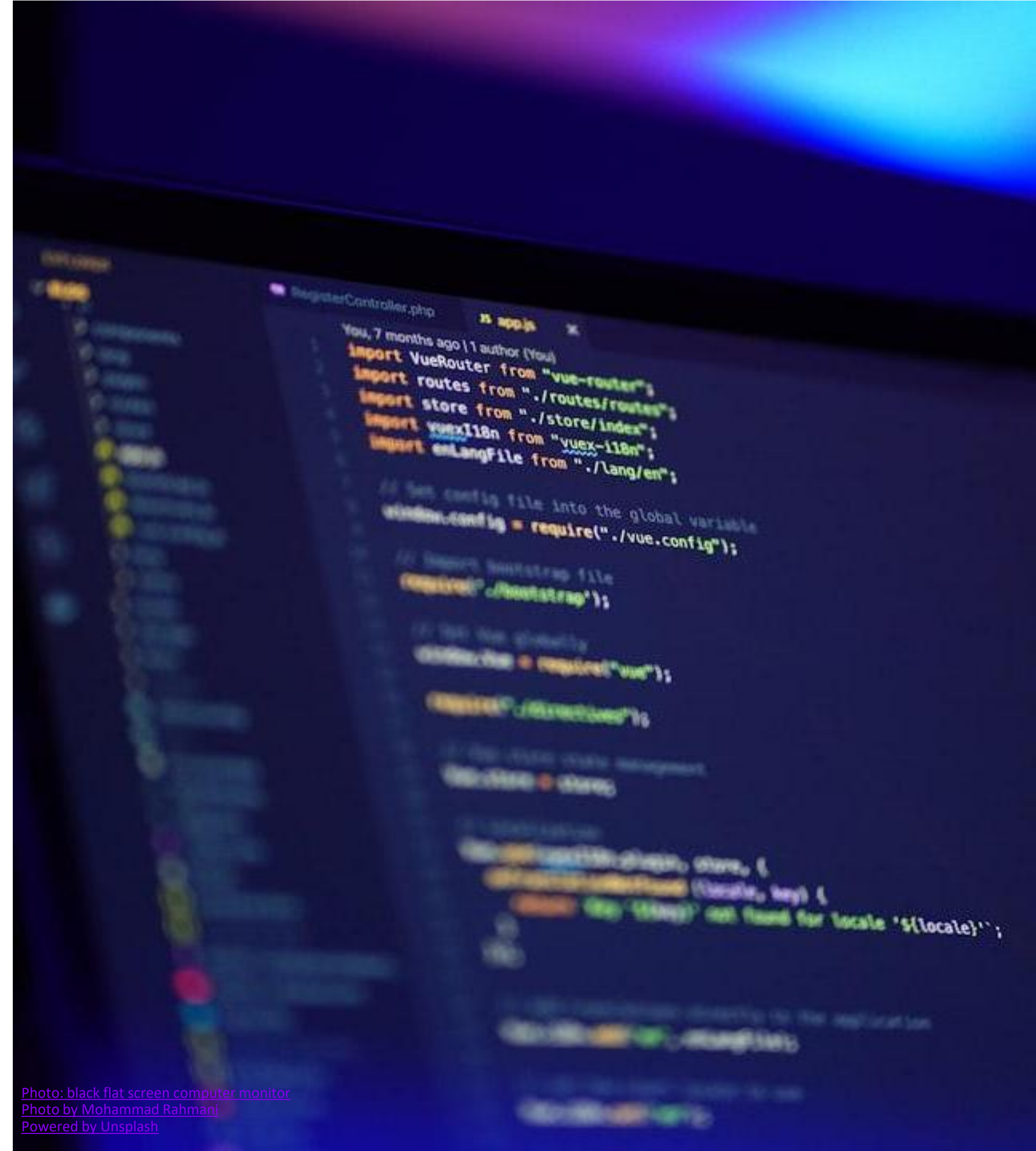
PEP 8 - Python Enhancement Proposal

- PEP 8 is the official style guide for Python code.
- Key principles include:
 - - Consistent indentation
 - - Readable naming conventions
 - - Proper use of whitespace



Zen of Python

- A collection of guiding principles for writing Pythonic code.
- Famous rules include:
 - - Simple is better than complex.
 - - Readability counts.



ZEN OF PYTHON

RUNNING ZEN IN CMD

```
py  
import this
```

COMMENTS

The Zen of Python, by Tim
Peters

Beautiful is better than ugly.

Explicit is better than implicit.