1. Use meaningful names for variables, functions, and classes

```python
width  = 10
height = 20

def calculate_area(length, width):
    return length * width
```

2. Write small and focused functions

```python
def clean_data(data = []):
    # function that cleans the data
    return 0
```

3. Use comments sparingly and only when necessary

+ Code \+ Text

```python
for element in my_list:
    #print(element)
```

4. Follow the PEP 8 style guide

```python
def my_function(argument1, argument2):
    result = argument1 + argument2
    return result
```

5. Use docstrings to document functions and classes

```python
def calculate_area(length, width):
    """Calculate the area of a rectangle.

    Args:
        length (float): The length of the rectangle.
        width (float): The width of the rectangle.

    Returns:
        float: The area of the rectangle.
    """
    return length * width
```

6. Write tests for your code

```python
import unittest

class TestCalculation(unittest.TestCase):
    def test_area_calculation(self):
        self.assertEqual(calculate_area(10, 20), 200)
```

7. Use exceptions to handle errors

```python
try:
    result = 5 / 0
except ZeroDivisionError:
    result = None
```

8. Use constants or configuration files instead of hardcoding values

```python
PRICE_PER_ITEM = 5.0

```

```python
def calculate_price(quantity):
    return quantity * PRICE_PER_ITEM
```

9. Write simple and readable code

```python
a, b, c, d, e = 1, 1, 1, 1, 1
term1 = b * c
term2 = d ** e
result = a + term1 / term2
```

10. Use built-in functions and libraries

```python
my_list = [1, 2, 3, 4, 5]
result = list(map(lambda x: x ** 2, my_list))
```