

```
[45]: class Bird:
      def __init__(self, name):
          self.name = name
      class FlyingBird(Bird):
          def fly(self):
              print('I am flying')
      class NonFlyingBird(Bird):
          def fly(self):
              print('I can not fly')
      peacock = NonFlyingBird("Morni")
      peacock.fly()
```

I can not fly

```
[46]: sparrow = FlyingBird("Chidiya")
      sparrow.fly()
```

I am flying

```
[ ]:
```

Parent Class

```
[47]: class Bird:
      def __init__(self, name):
          self.name = name

      class FlyingBird(Bird):
          def fly(self):
              print(self.name, 'I am flying')

      class NonFlyingBird(Bird):
          def fly(self):
              print(self.name, 'I can not fly')

      peacock = NonFlyingBird("Morni")

      peacock.fly()

      Morni I can not fly
```

```
[48]: sparrow = FlyingBird("Chidiya")
      sparrow.fly()

      Chidiya I am flying
```



PEP 8, which stands for Python Enhancement Proposal 8, is the official style guide for writing Python code. It provides a set of conventions and recommendations for code layout, formatting, naming, and structure, aiming to enhance the readability and consistency of Python code. M Medium

Key aspects covered by PEP 8 include:

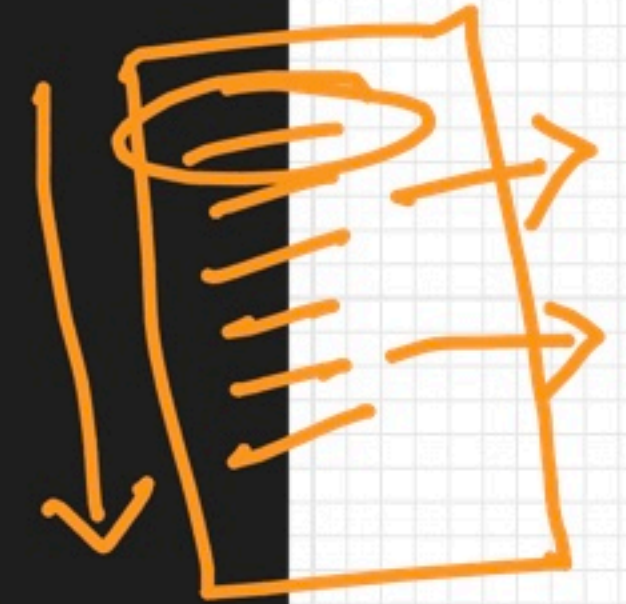
- ✓ **Indentation:** Recommends using 4 spaces per indentation level. ✓
- **Line Length:** Suggests limiting lines to a maximum of 79 characters for improved readability. ✓
- ✓ **Naming Conventions:** Provides guidelines for naming variables, functions, classes, modules, and constants (e.g., snake\\_case for variables and functions, CamelCase for classes, ALL\\_CAPS for constants). ✓
- **Whitespace:** Defines rules for using whitespace around operators, in expressions, and within various code structures. ✓
- **Comments:** Offers guidance on writing clear and concise comments. ✓
- **Imports:** Specifies how imports should be organized and ordered. ✓



$$x = a + b \quad \checkmark$$

$$PI = 3.14$$

~~Python~~ Python



isort