

Abstract Classes

```
sadfas-
dfa-      from abc import ABC,
sdfas      abstr actm ethod
asdfas-
dfsfdsd-  class MyClas s(ABC):
eeeeeee-  @abstr act method
eeeeeee-  def some_m eth od( self):
eeeeeee-
eeeeeee-  pass
eeeeeee-
eeeeeee-
eeeeeee-
eeeeeee-
eeeeeee-
eeeeeee-
eeeeeee-
eeeeeee-
```

Data Classes (cont)

```
> # and store it in the object
self.area: int = width * height

def draw(self) -> str:
    return f'Draw a {self.color} \
rectangle with area {self.area}'
rect = Rectangle(width=10, height=20,
color='red')
print(rect.draw()) # Draw a red rectangle
with area 200
# AttributeError: 'Rectangle' object has no
attribute 'width'
print(rect.width)
# AttributeError: 'Rectangle' object has no
attribute 'height'
print(rect.height)
```

gdsfgsdfg

```
from abc import ABC,
    abstr actm ethod

class MyClas s(ABC):
    @abstr act method

    def some_m eth od( self):
        pass
```

das

```
fasdfasdf
```

Data Classes

```
from dataclasses import
dataclass,
InitVar
@dataclass
class Rectangle:
    # We don't want to store
    # width in the object
    width: InitVa r[int]
    # We don't want to store
    height in the object
    height: InitVa r[int]
    color: str
    def __post__in it__(self,
width: int, height: int):
        # Create a new attribute
        called area
```

Unit Testing with Pytest

1. Write tests early and often
2. Keep tests small and focused:
3. Use descriptive test names
4. Avoid testing implementation details
5. Use test fixtures
6. Use mocking and stubbing when necessary
7. Run tests automatically
8. Maintain code coverage

```
pip install pytest
def test_addition():
    assert 2 + 2 == 4
```

```
pytest test_m ath.py
```

```
pip install coverage
pytest --cov= my_ package
```



By **ilbaks01**
cheatography.com/ilbaks01/

Not published yet.
Last updated 22nd January, 2024.
Page 1 of 1.

Sponsored by **ApolloPad.com**
Everyone has a novel in them. Finish
Yours!
<https://apollopad.com>