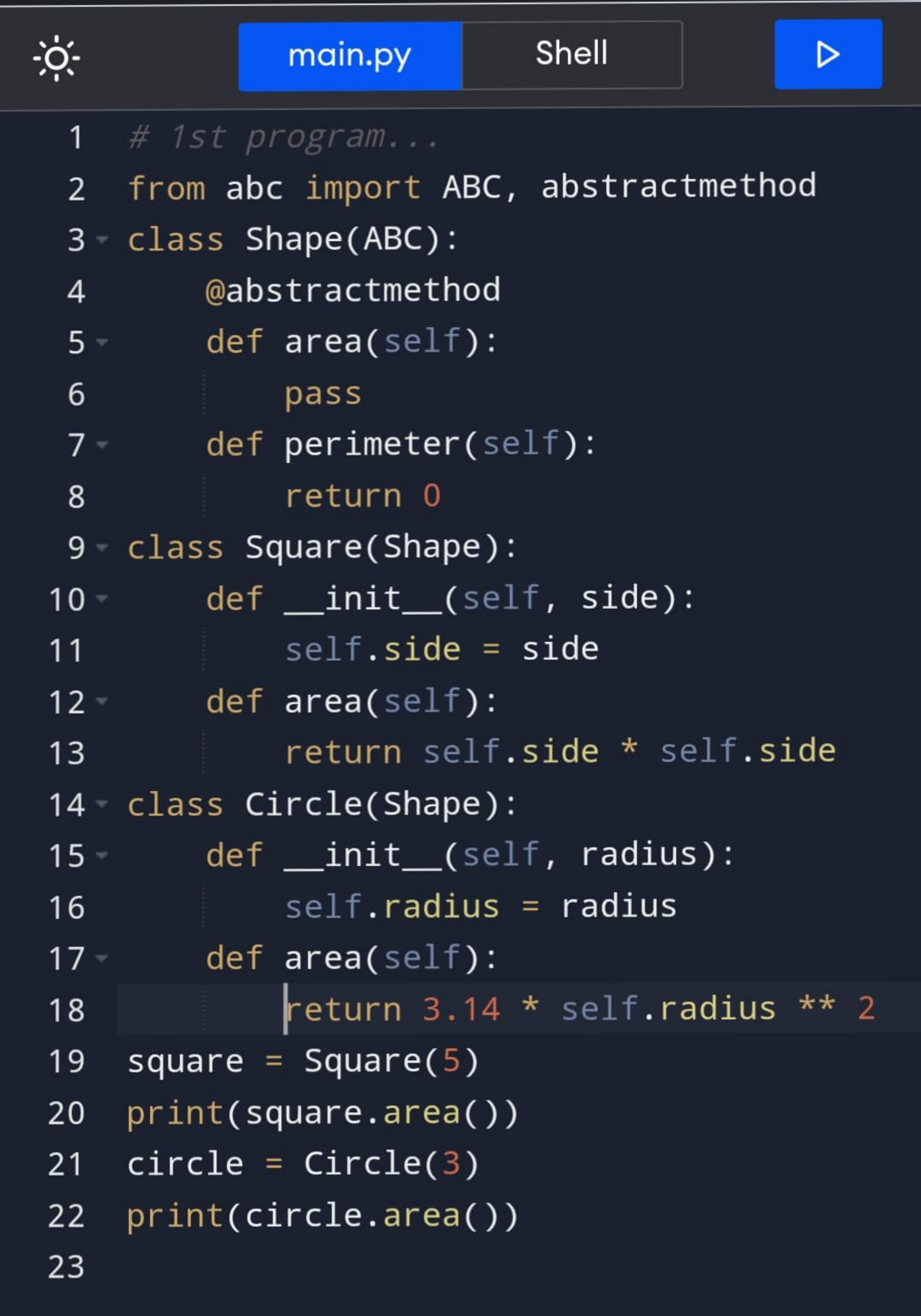
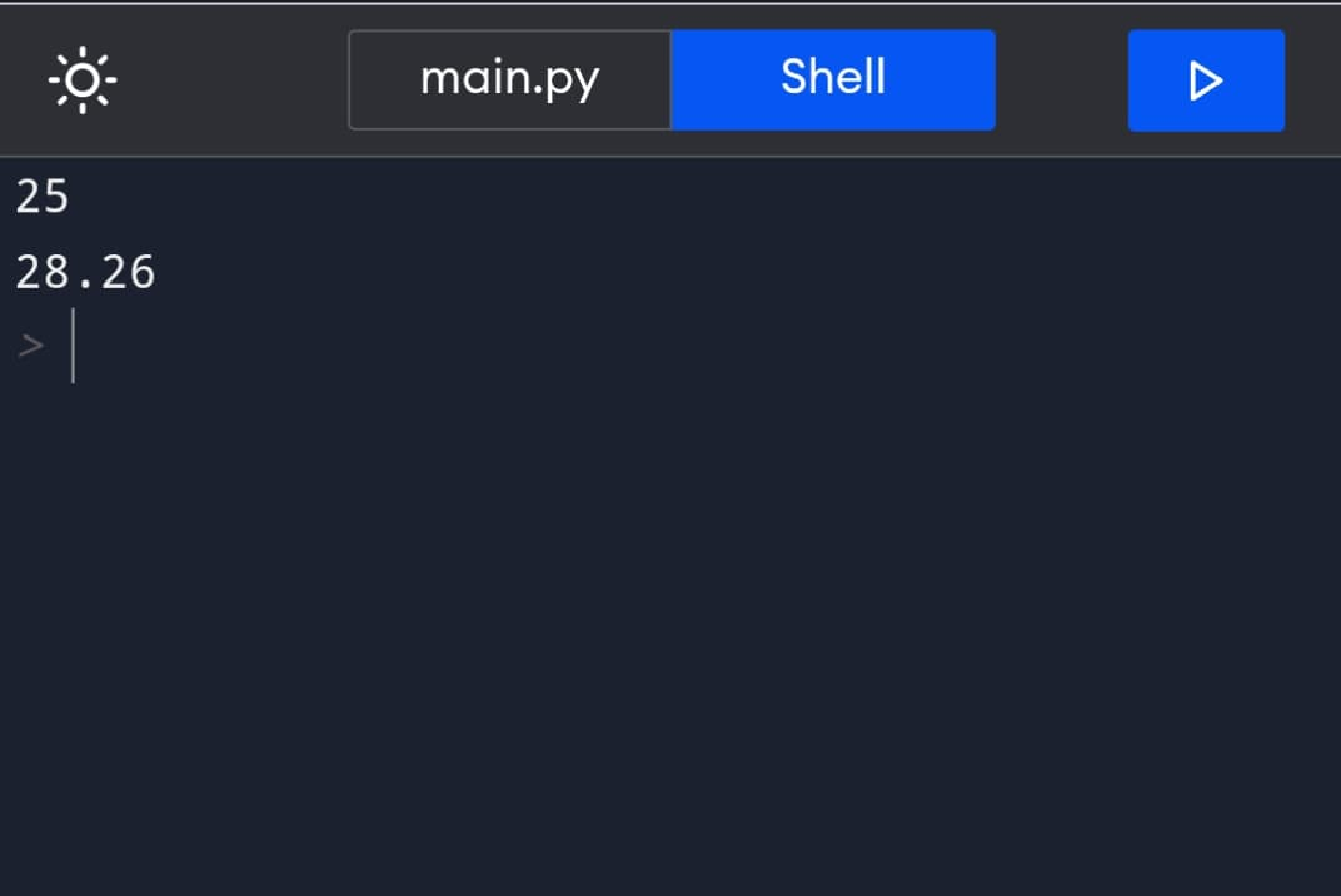
9.Abstract Class

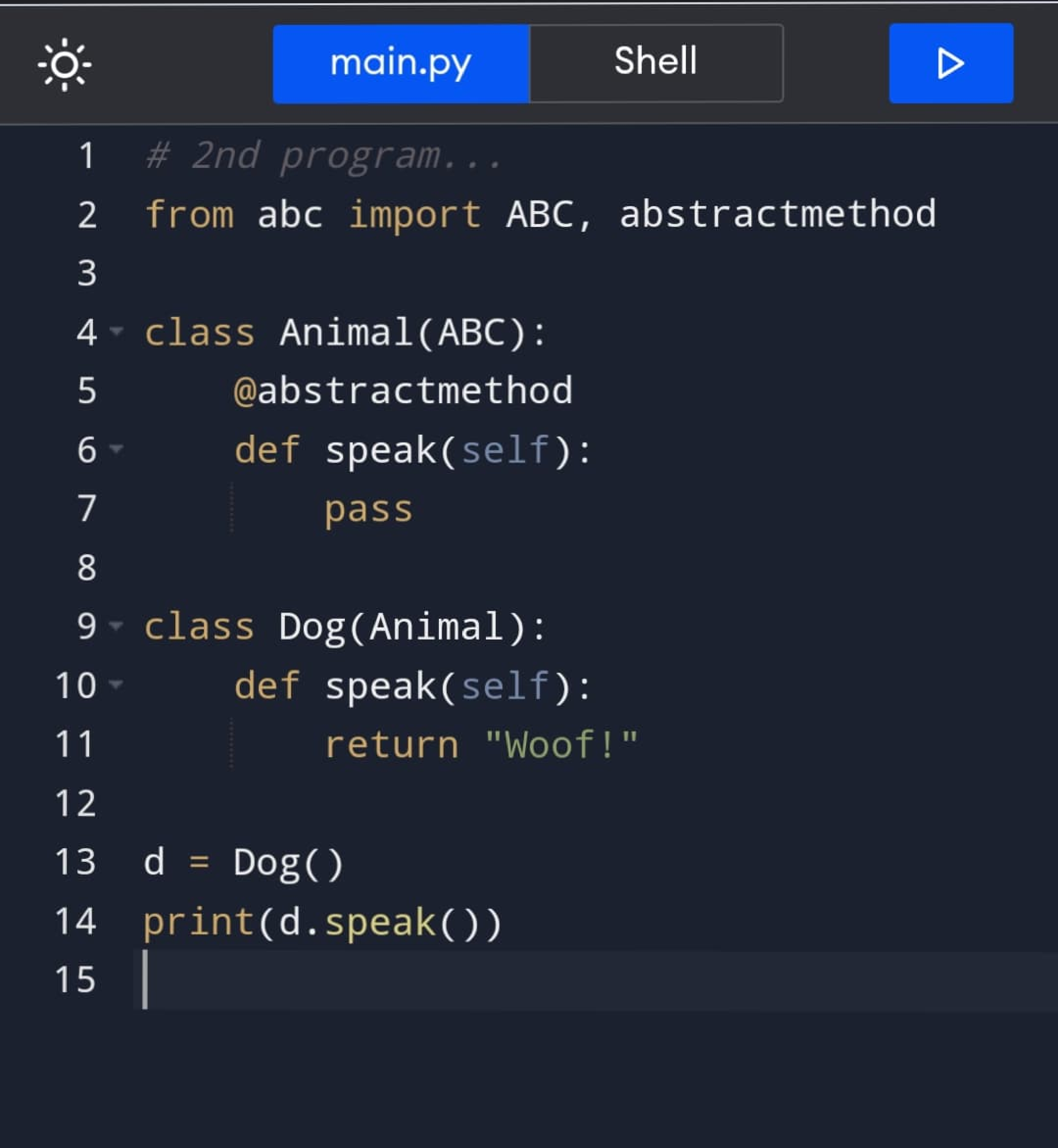
1. Create an abstract class with abstract and non-abstract methods.
2. Create a sub class for an abstract class. Create an object in the child class for the abstract class and access the non-abstract methods
3. Create an instance for the child class in child class and call abstract methods
4. Create an instance for the child class in child class and call non-abstract methods



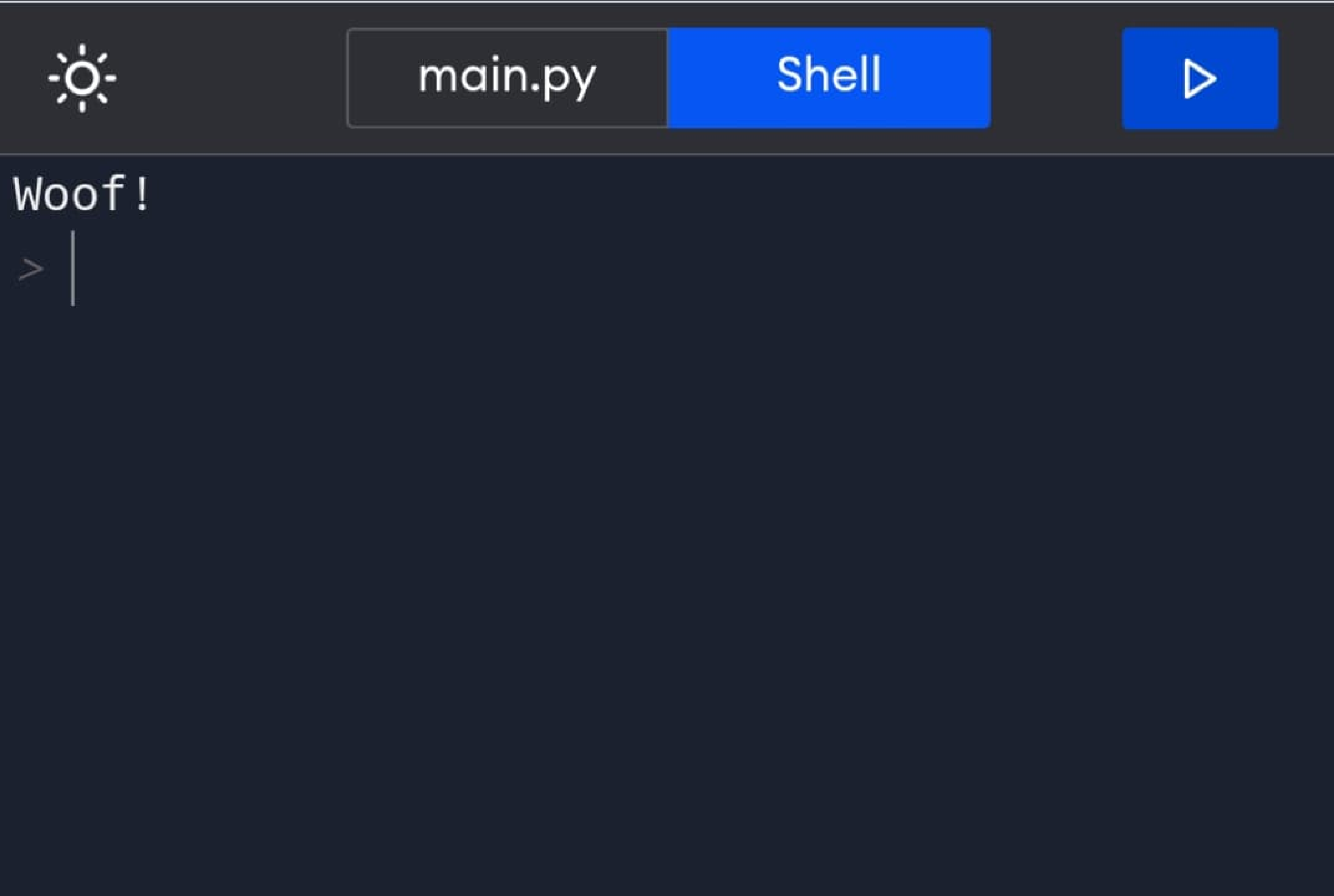
```python
# 1st program...
from abc import ABC, abstractmethod
class Shape(ABC):
    @abstractmethod
    def area(self):
        pass
    def perimeter(self):
        return 0
class Square(Shape):
    def __init__(self, side):
        self.side = side
    def area(self):
        return self.side * self.side
class Circle(Shape):
    def __init__(self, radius):
        self.radius = radius
    def area(self):
        return 3.14 * self.radius ** 2
square = Square(5)
print(square.area())
circle = Circle(3)
print(circle.area())
```



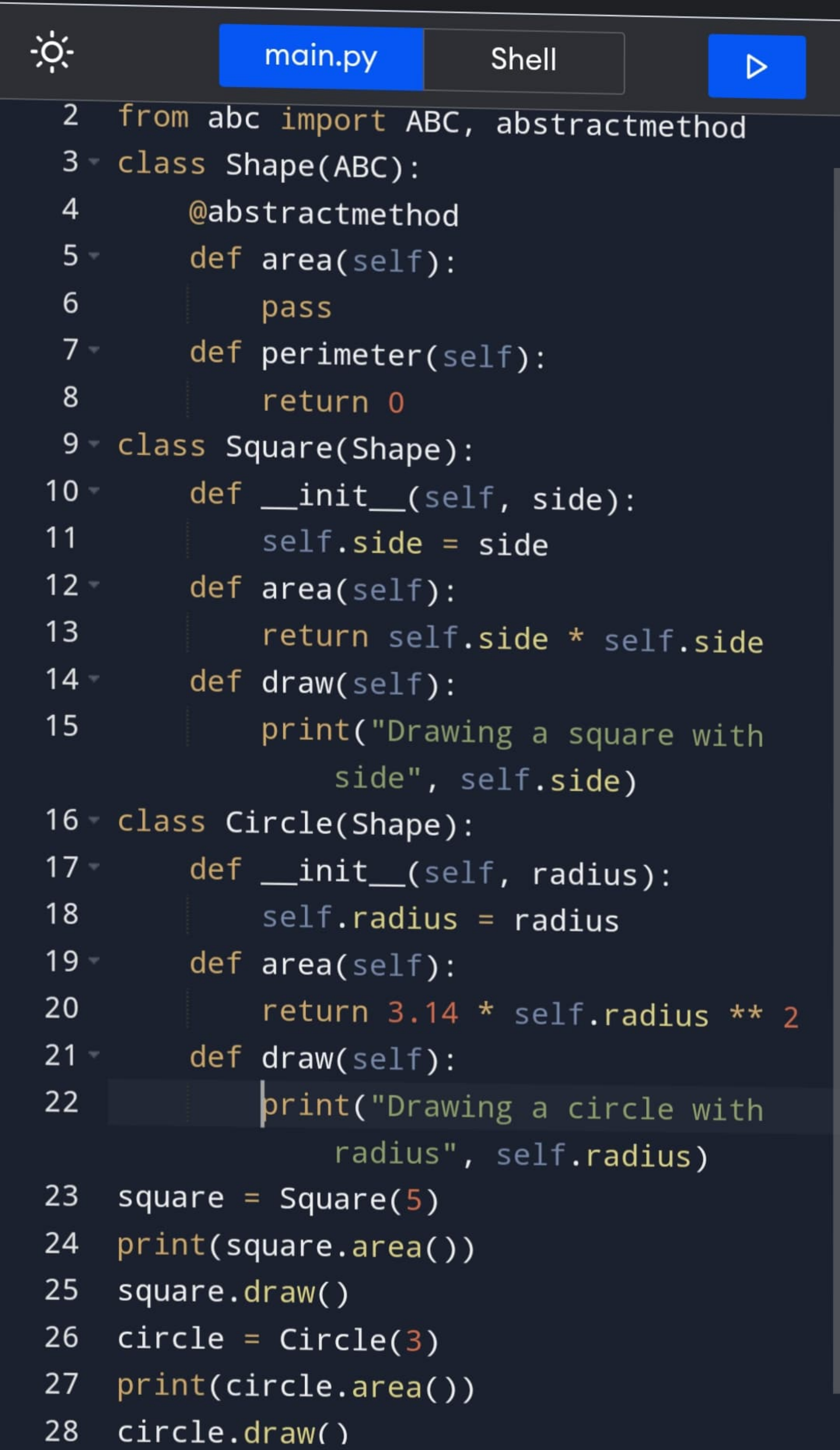
```
25
28.26
>
```



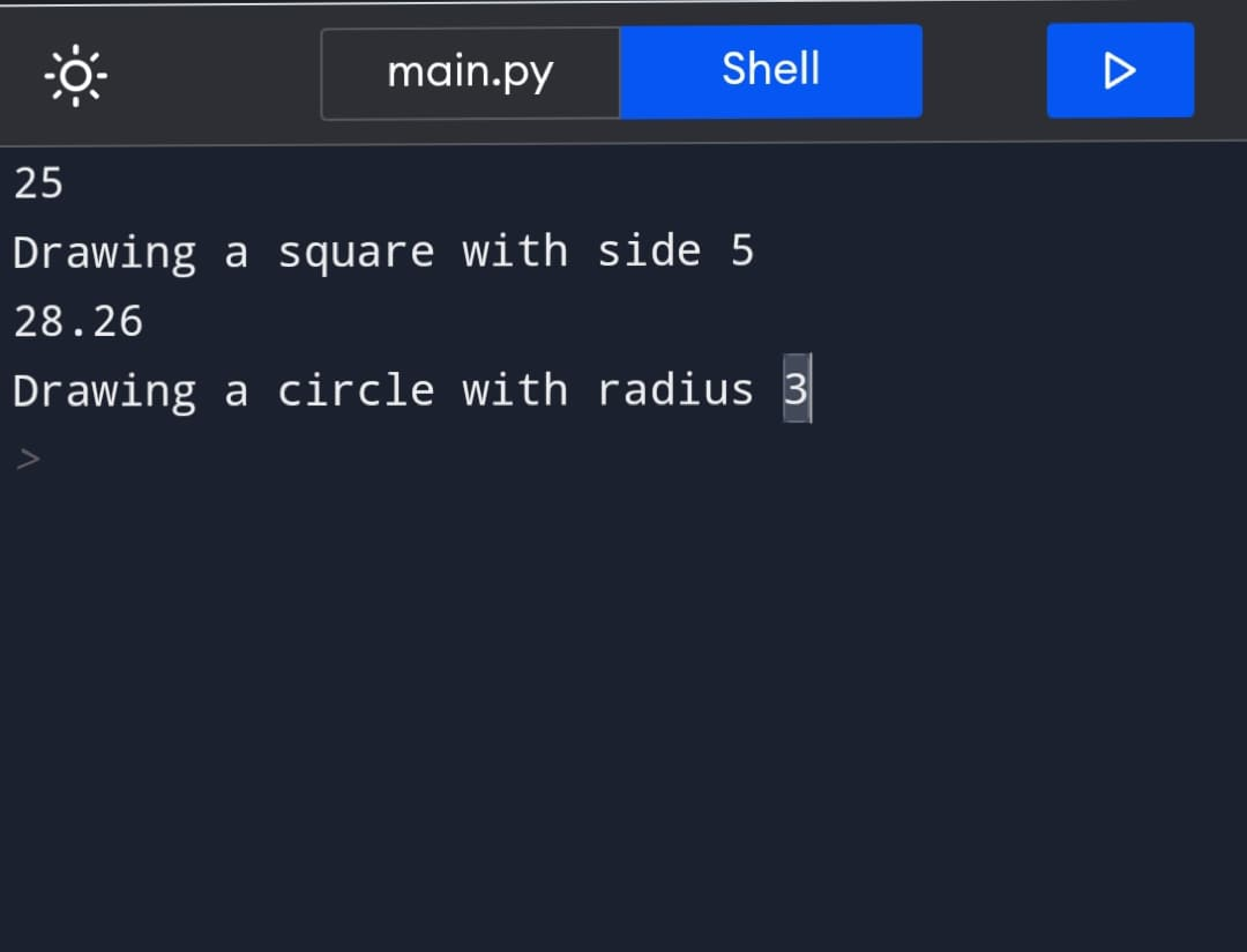
```python
# 2nd program...
from abc import ABC, abstractmethod

class Animal(ABC):
    @abstractmethod
    def speak(self):
        pass

class Dog(Animal):
    def speak(self):
        return "Woof!"

d = Dog()
print(d.speak())

```



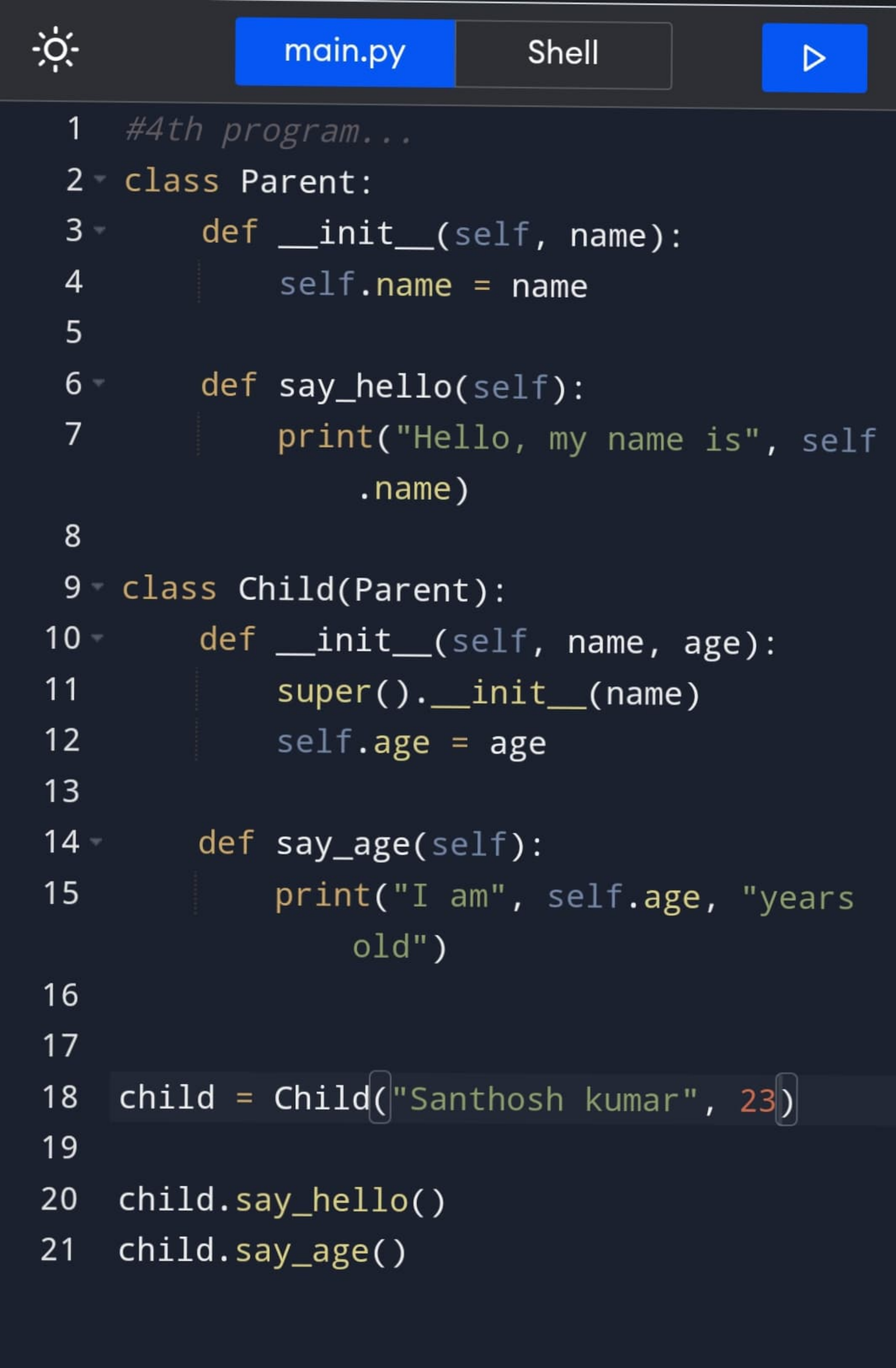
```
Woof!
>
```



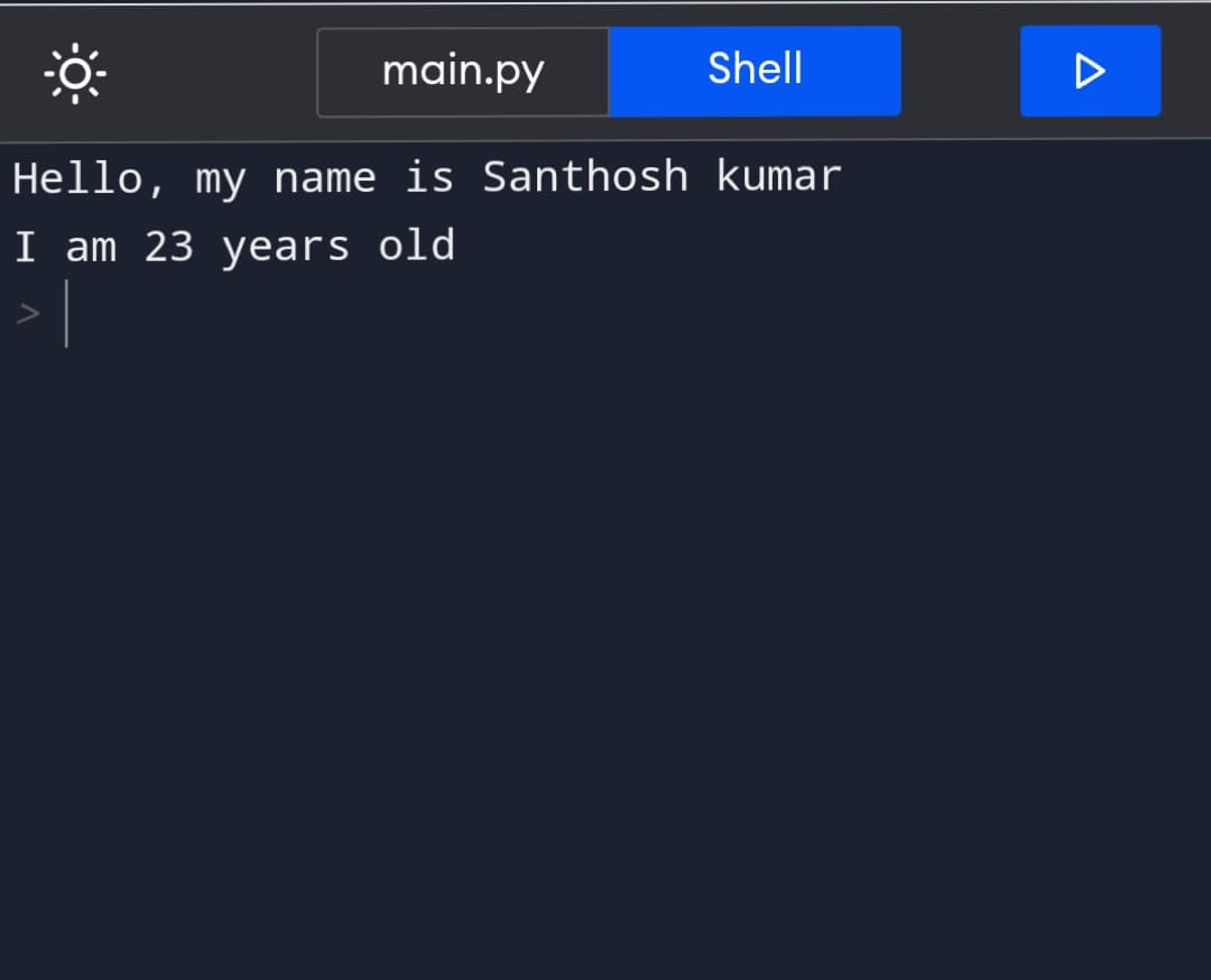
```python
from abc import ABC, abstractmethod
class Shape(ABC):
    @abstractmethod
    def area(self):
        pass
    def perimeter(self):
        return 0
class Square(Shape):
    def __init__(self, side):
        self.side = side
    def area(self):
        return self.side * self.side
    def draw(self):
        print("Drawing a square with side", self.side)
class Circle(Shape):
    def __init__(self, radius):
        self.radius = radius
    def area(self):
        return 3.14 * self.radius ** 2
    def draw(self):
        print("Drawing a circle with radius", self.radius)
square = Square(5)
print(square.area())
square.draw()
circle = Circle(3)
print(circle.area())
circle.draw()
```



```
25
Drawing a square with side 5
28.26
Drawing a circle with radius 3
>
```



```python
#4th program...
class Parent:
    def __init__(self, name):
        self.name = name

    def say_hello(self):
        print("Hello, my name is", self.name)

class Child(Parent):
    def __init__(self, name, age):
        super().__init__(name)
        self.age = age

    def say_age(self):
        print("I am", self.age, "years old")


child = Child("Santhosh kumar", 23)

child.say_hello()
child.say_age()
```



```
Hello, my name is Santhosh kumar
I am 23 years old
>
```