Assingment 3

February 14, 2025

0.1 RANAPANADA NETHMI MALSHA NIWANTHANI

0.1.1 2025/02/14

0.2 Exercise 3

1. Write a lambda expression to get the product of two numbers.

[4]:
```
product = lambda x, y: x * y
result = product(5, 6)
print(result)
```

```
30
```

2. Write a function to get the area of a circle from the radius. Hint: remember to import the right modul for being able to calculte the area of the circle.

[8]:
```
import math
def circle_area(radius):
    return math.pi * radius ** 2

radius = 5
area = circle_area(radius)
print(f"Area of the circle with radius {radius}: {area}")
```

```
Area of the circle with radius 5: 78.53981633974483
```

3. Build a simple calculator which can: add, subtract, multiply, divide. Hint: solve by writing a function that takes as argument two numbers and the operation and returns the desired output.

[11]:
```
def calculator(a, b, operation):
    if operation == "a":
        return a + b
    elif operation == "s":
        return a - b
    elif operation == "m":
        return a * b
    elif operation == "d":
        if b == 0:
            return "Error: Division by zero"
```

```
        return a / b
    else:
        return "Invalid operation"

result = calculator(2, 5, 'd')
print(result)
```

```
0.4
```

4. Define a class named Rectangle which can be constructed by a length and width. The Rectangle class has a method which can compute the area.

[26]:
```
class Rectangle:
    def __init__(self, length, width):
        self.length = length
        self.width = width

    def area(self):
        return self.length * self.width

r = Rectangle(5, 10)
print(r.area())
```

```
50
```

5. Define a class named Shape and its subclass Square. Shape objects can be consrtucted by name and length has an area function wich return 0

Square subclass has an init function which take a length and name as argument and has an area method and a describe method what prints the name of the Shape.

[33]:
```
class Shape:
    def __init__(self, name, length):
        self.name = name
        self.length = length

    def area(self):
        return 0

class Square(Shape):
    def __init__(self, name, length):
        super().__init__(name, length)

    def area(self):
        return self.length * self.length

    def describe(self):
        return f"This is a: {self.name}"
```

```
s = Square('square', 5)
print("The area is:\n", s.area())
print(s.describe())
```

```
The area is:
 25
This is a: square
```