VACATION CLASS

ALGORITHM AND DATA STRUCTURE

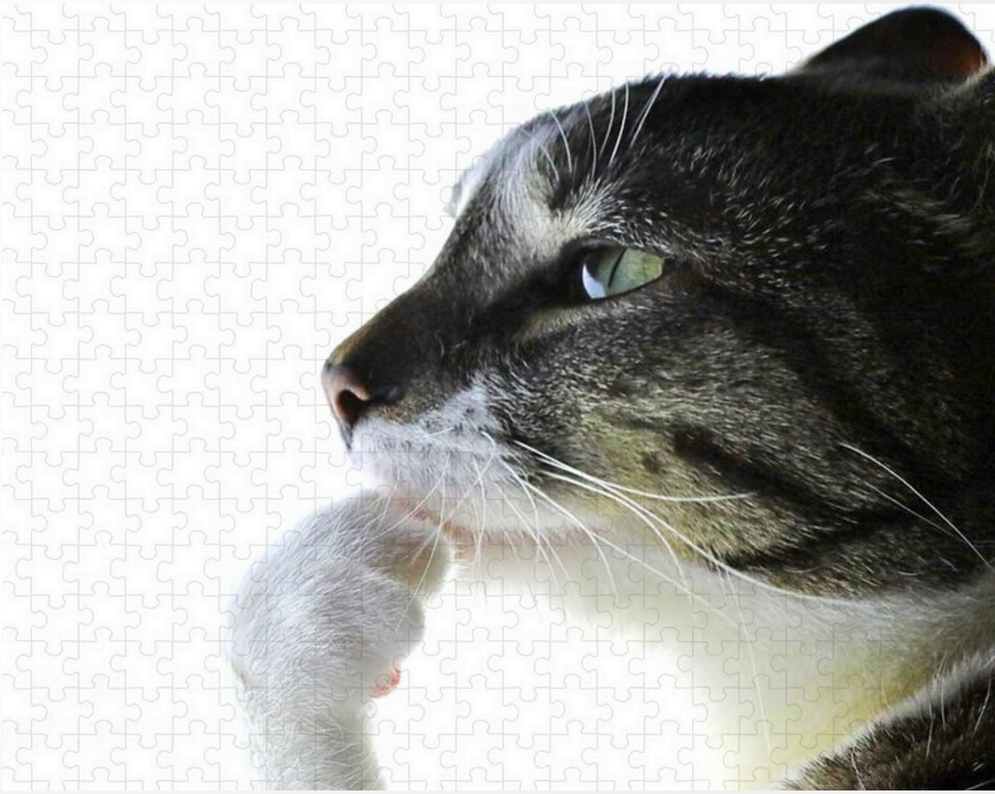
Week 1

04 September 2025

YEM DARO

CONCEPT REVIEW

- **Data Types**
- **Basic Operations**
- **Variable**
- **Function**
- **Condition**
- **Loop**
- **Collection**



ALGORITHM

1) sequence of simple **steps**
2) **flow of control** process that specifies when each step is executed
3) a means of determining **when to stop**

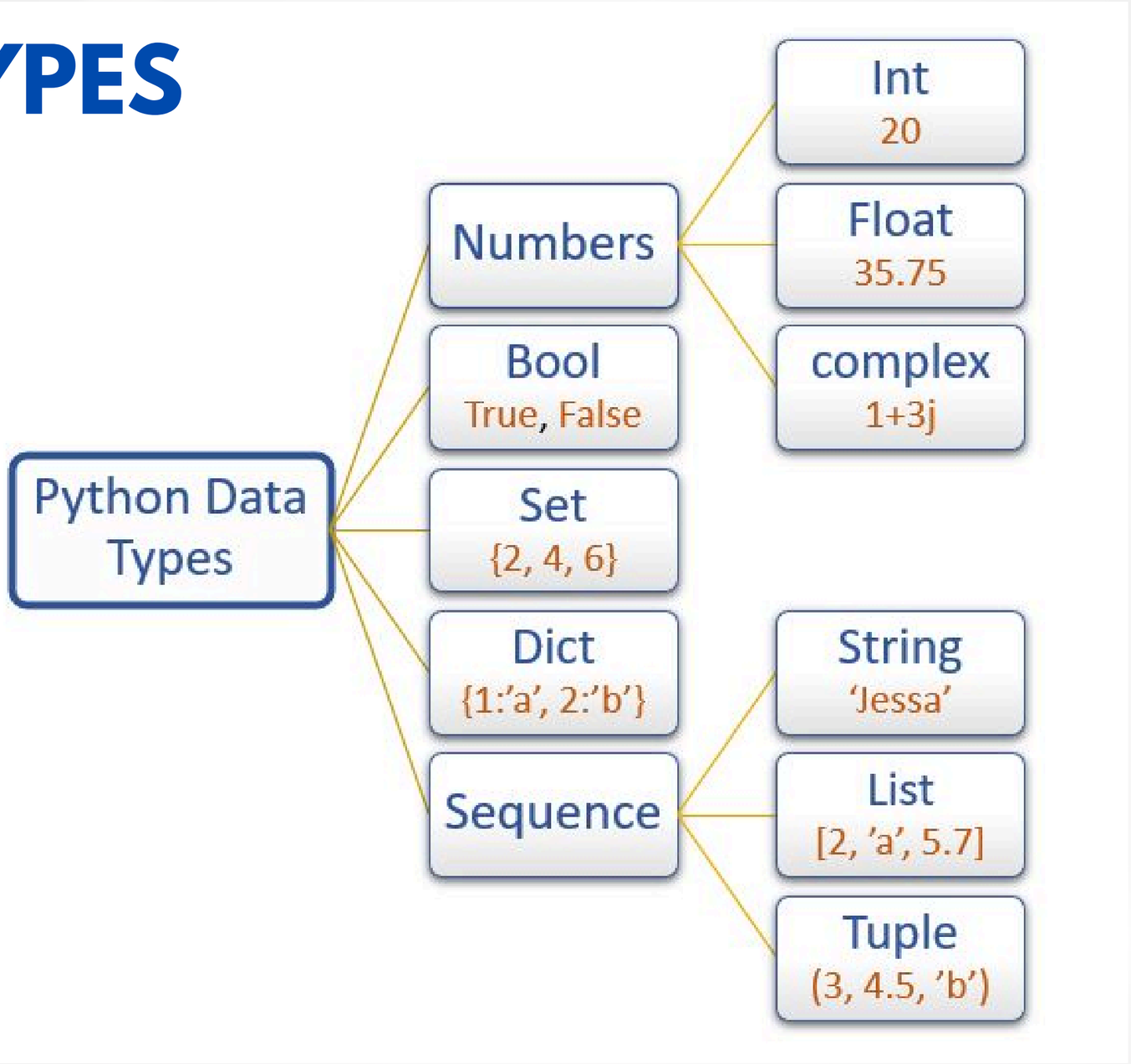
1+2+3 = an **algorithm**!

ALGORITHM

IT’S LIKE A RECIPE

1. **Wash the rice.**
2. **Cook the rice.**
3. **Chop vegetables.**
4. **Heat oil in a pan.**
5. **Add vegetables and stir-fry.**
6. **Add cooked rice.**
7. **Season with soy sauce and salt.**
8. **Stir well and serve.**

DATA TYPES


Python Data Types
Numbers
Int
20
Float
35.75
complex
1+3j
Bool
True, False
Set
{2, 4, 6}
Dict
{1:'a', 2:'b'}
Sequence
String
'Jessa'
List
[2, 'a', 5.7]
Tuple
(3, 4.5, 'b')

DATA TYPES

Example Code

```
x = 10                  # Integer
y = 10.5                # Float
is_active = True        # Boolean
name = "Alice"          # String

print(type(x))
print(type(y))
print(type(is_active))
print(type(name))
```

Output:

OPERATION

OPERATORS ON ints and floats

- `i+j` → the **sum**
- `i-j` → the **difference**
- `i*j` → the **product**

if both are ints, result is int
if either or both are floats, result is float

- `i/j` → **division** → result is float

- `i%j` → the **remainder** when `i` is divided by `j`
- `i**j` → `i` to the **power** of `j`

OPERATION

COMPARISON

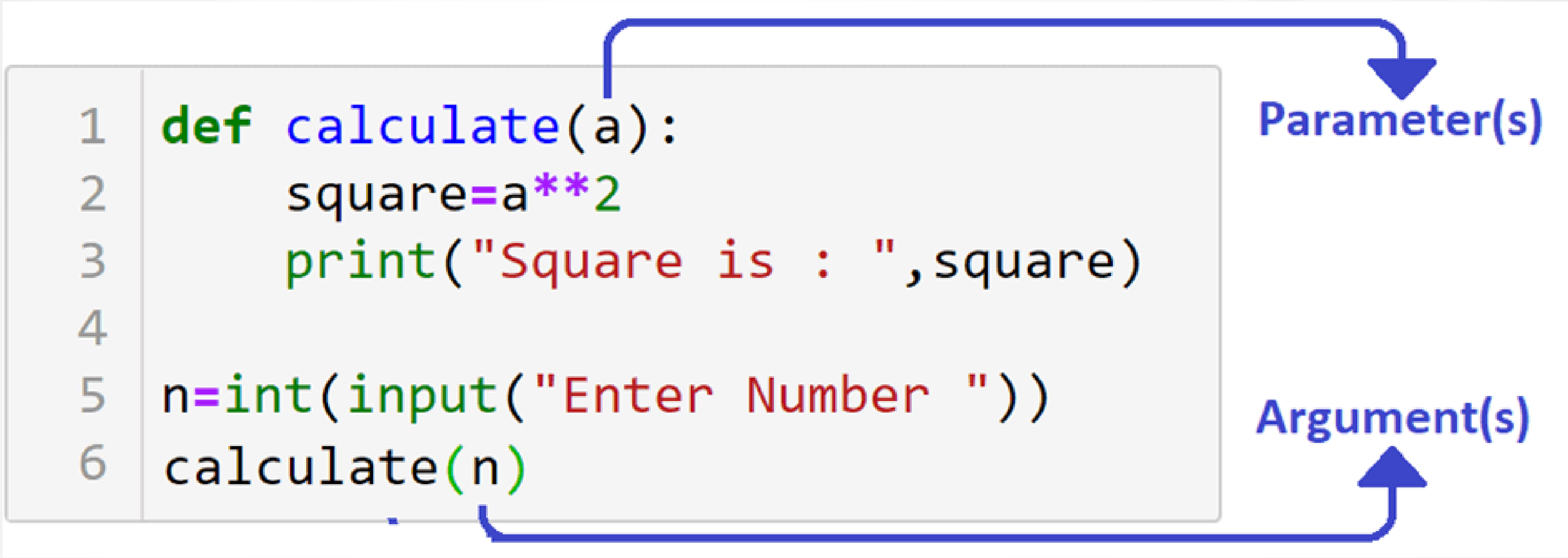
Operator	Name	Example
==	Equal	x == y
!=	Not equal	x != y
>	Greater than	x > y
<	Less than	x < y
>=	Greater than or equal to	x >= y
<=	Less than or equal to	x <= y

OPERATION

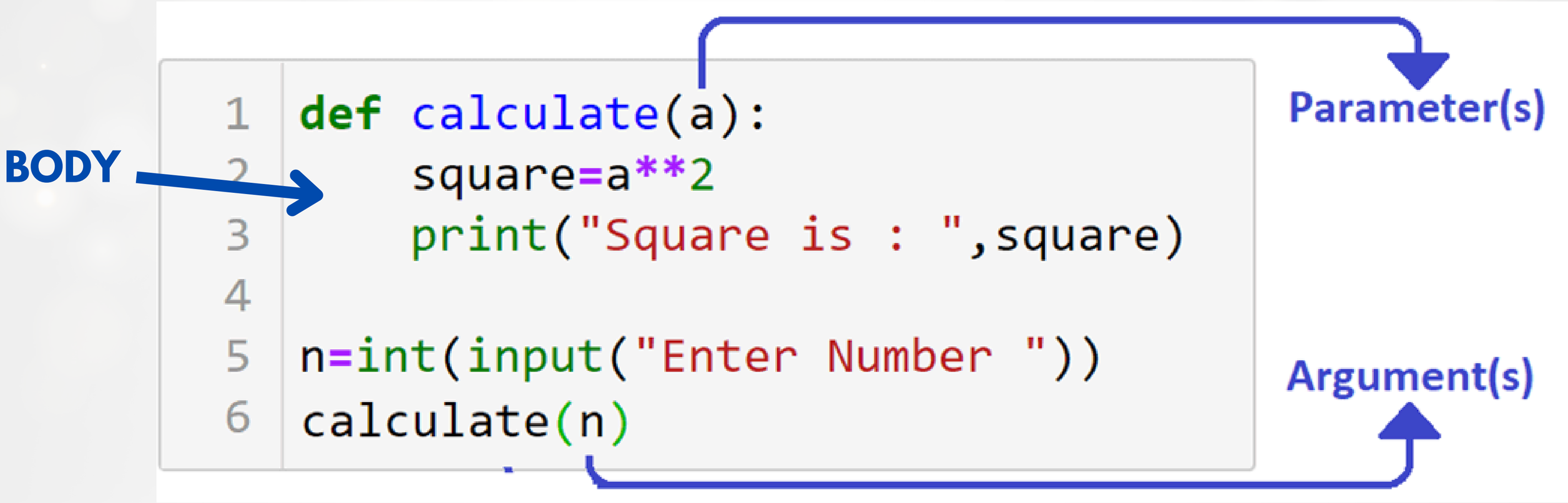
LOGICAL OPERATOR

Operator	Description	Example
and	Returns True if both statements are true	x < 5 and x < 10
or	Returns True if one of the statements is true	x < 5 or x < 4
not	Reverse the result, returns False if the result is true	not(x < 5 and x < 10)

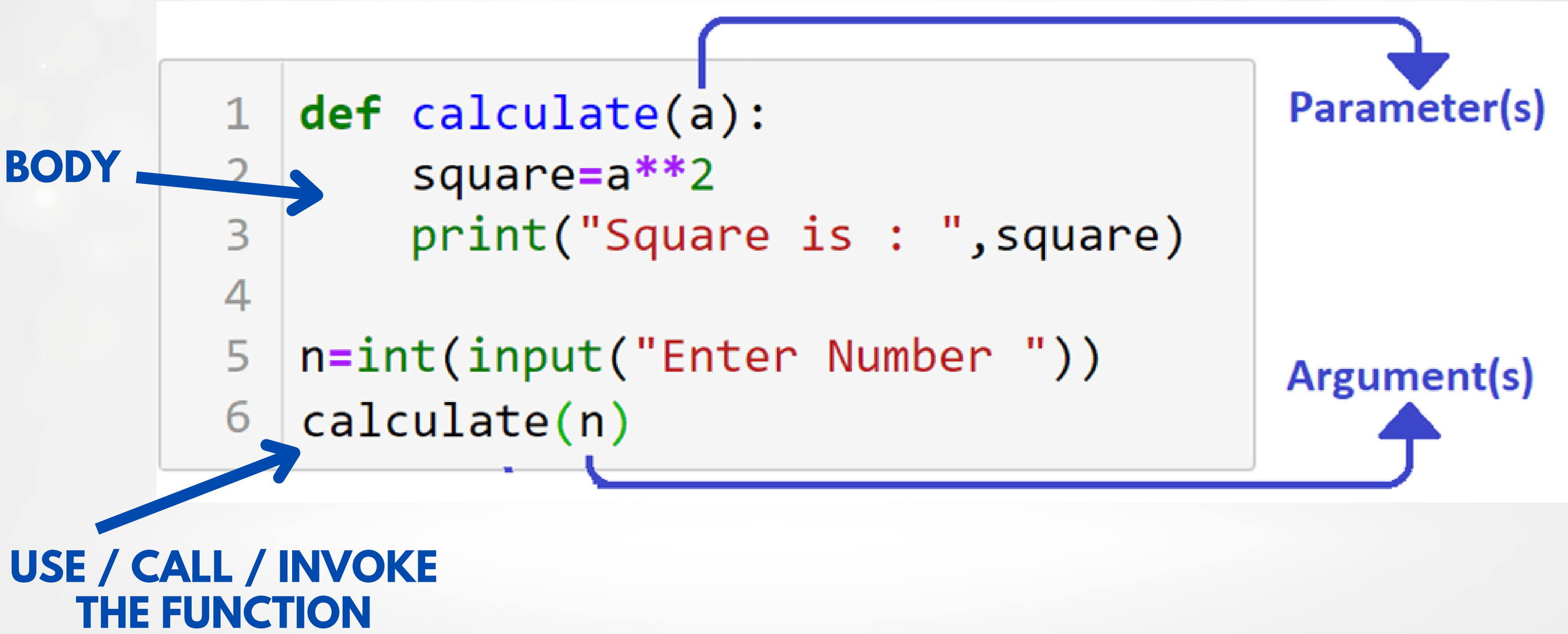
FUNCTION


1 def calculate(a):
2 square=a**2
3 print("Square is : ",square)
4
5 n=int(input("Enter Number "))
6 calculate(n)
Parameter(s)
Argument(s)

FUNCTION


BODY
1 def calculate(a):
2 square=a**2
3 print("Square is : ",square)
4
5 n=int(input("Enter Number "))
6 calculate(n)
Parameter(s)
Argument(s)

FUNCTION


def calculate(a):
square=a**2
print("Square is : ",square)
n=int(input("Enter Number "))
calculate(n)
BODY
Parameter(s)
Argument(s)
USE / CALL / INVOKE
THE FUNCTION

FUNCTION

INDENTATION MATTERS

```
1  def calculate(a):
2      square=a**2
3      print("Square is : ",square)
4
5  n=int(input("Enter Number "))
6  calculate(n)
```

Pa

Ar

FUNCTION

SOME USEFUL FUNCTIONS IN PYTHON

- range
- len

CONDITIONAL

```
x = float(input("Enter a number for x: "))
y = float(input("Enter a number for y: "))
if x == y:
    print("x and y are equal")
    if y != 0:
        print("therefore, x / y is", x/y)
elif x < y:
    print("x is smaller")
else:
    print("y is smaller")
print("thanks!")
```

CONDITIONAL

INDENTATION MATTERS

```
x = float(input("Enter a number for x: "))
y = float(input("Enter a number for y: "))
if x == y:
    print("x and y are equal")
    if y != 0:
        print("therefore, x / y is", x/y)
elif x < y:
    print("x is smaller")
else:
    print("y is smaller")
print("thanks!")
```

LOOP

- iterate through numbers in a sequence

WHILE

```
# more complicated with while loop
n = 0
while n < 5:
    print(n)
    n = n+1
```

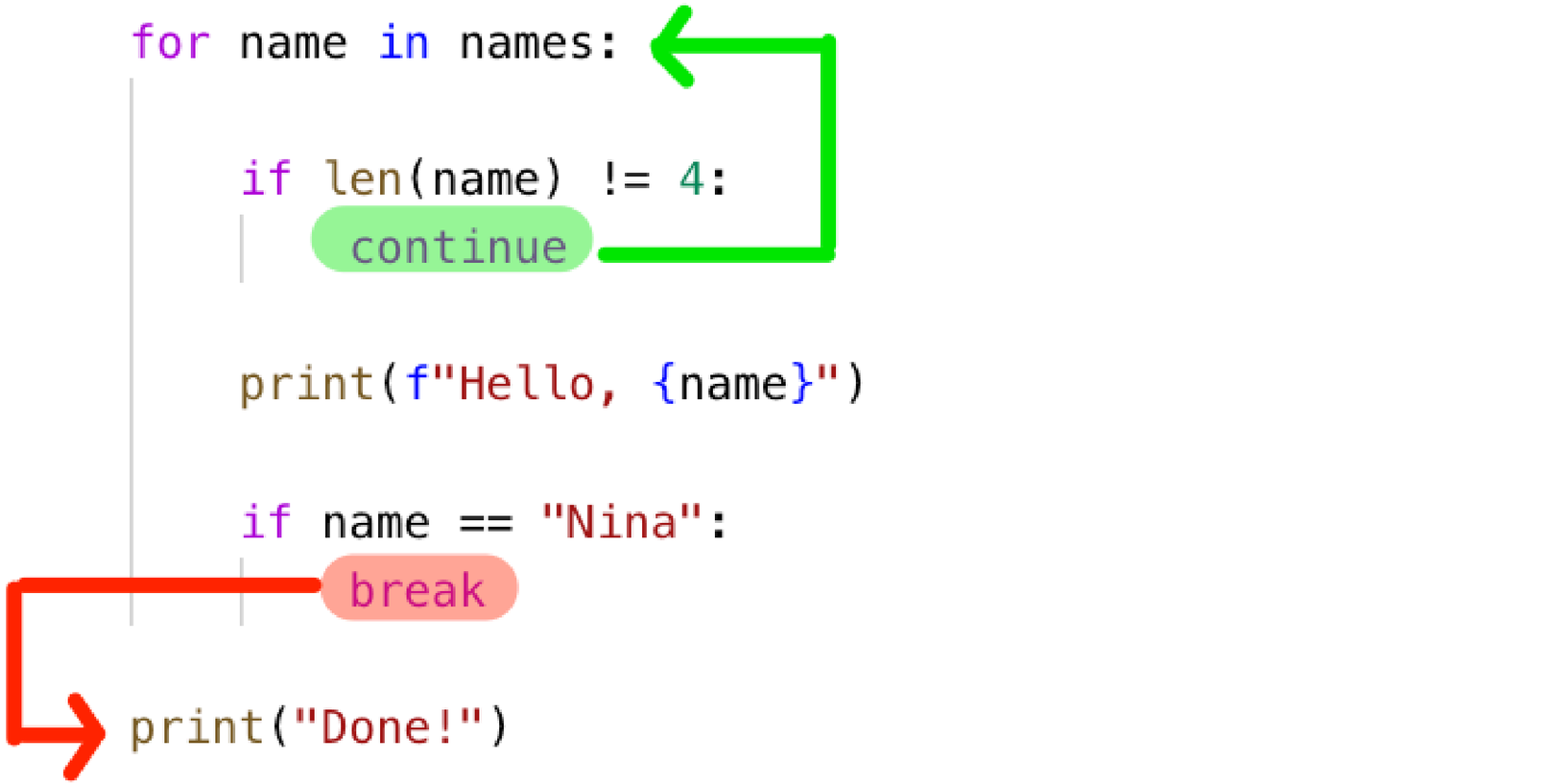
FOR

```
# shortcut with for loop
for n in range(5):
    print(n)
```

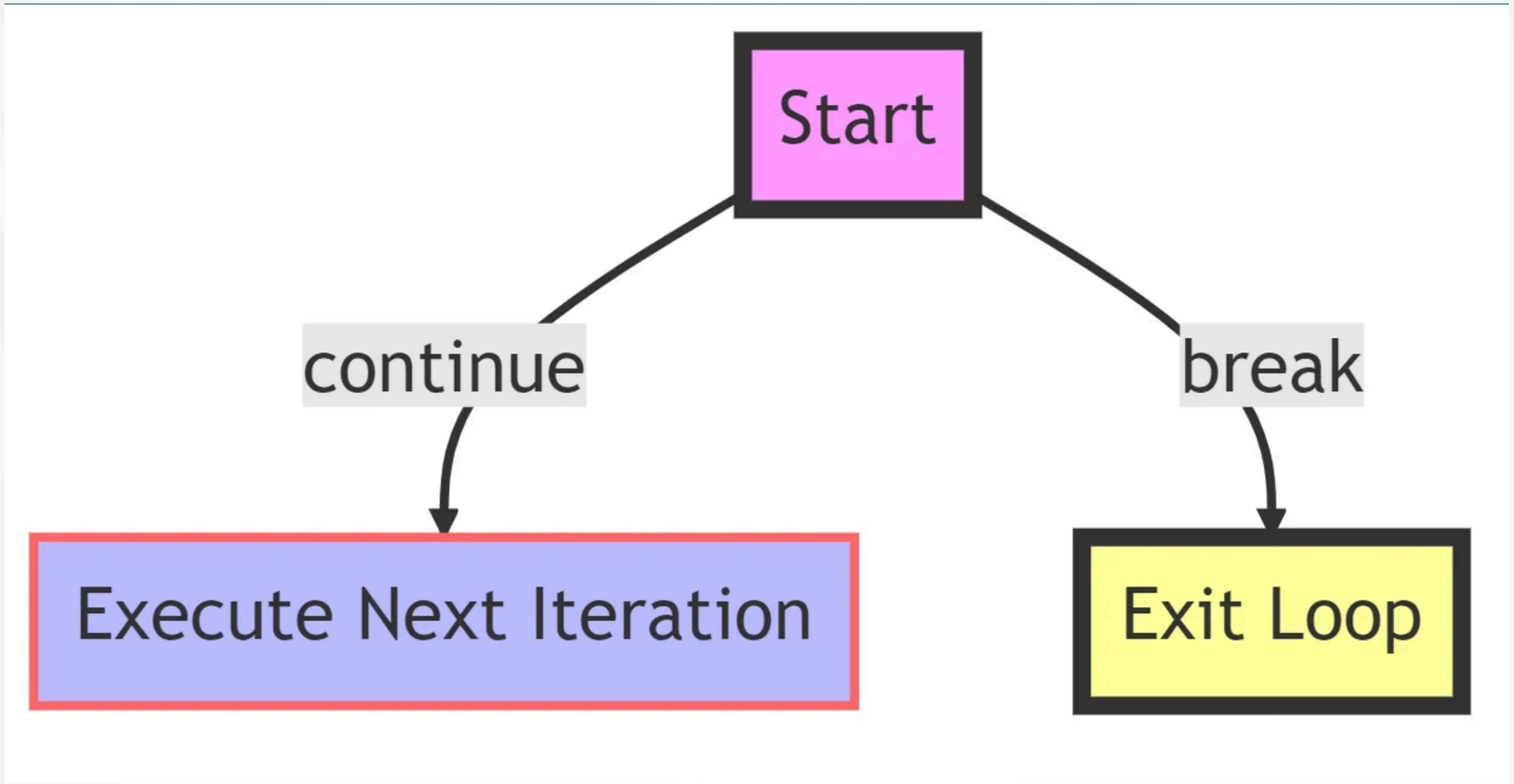
LOOP

Continue

Break



LOOP



ZZZZ

LET’S PRACTICE TOGETHER

PRACTICE

A NUMBER GUESSING GAME

If the user guesses:

- too low: prints “go higher”
- too high: prints “go lower”
- correctly: prints “YOU WIN”

PRACTICE

FOR LOOP

CALCULATE PI

In mathematics, the **Leibniz formula for π**, named after Gottfried Wilhelm Leibniz, states that

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \cdots = \sum_{k=0}^{\infty} \frac{(-1)^k}{2k+1},$$

PRACTICE

GOLANG CODE

```
func Sqrt(x float64) float64 {
    z := 1.0
    for i := 0; i < 10; i++ {
        z -= (z*z - x) / (2*z)
    }

    return z
}

func main() {
    fmt.Println(Sqrt(2))
}
```

CAN YOU WRITE THE FUNCTION IN PYTHON?

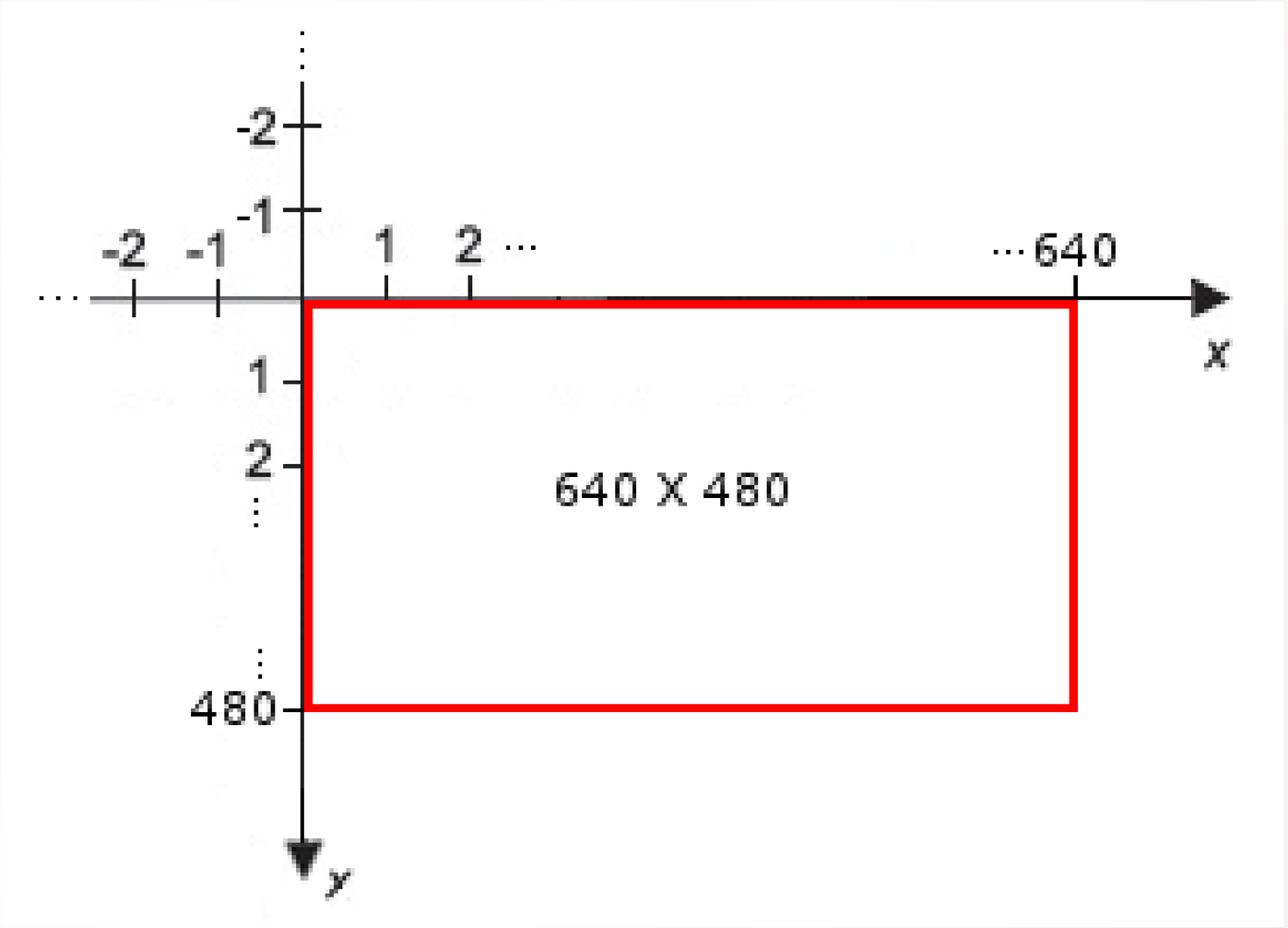
PYTHON VIRTUAL ENVIRONMENT

- python -m venv venv
- venv\Scripts\activate

GAME LIBRARY IN PYTHON

- pip install pygame-ce

PYGAME



PYGAME

LET'S MAKE SOMETHING MOVE

REFERENCE

MIT PYTHON COURSE

https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/download/