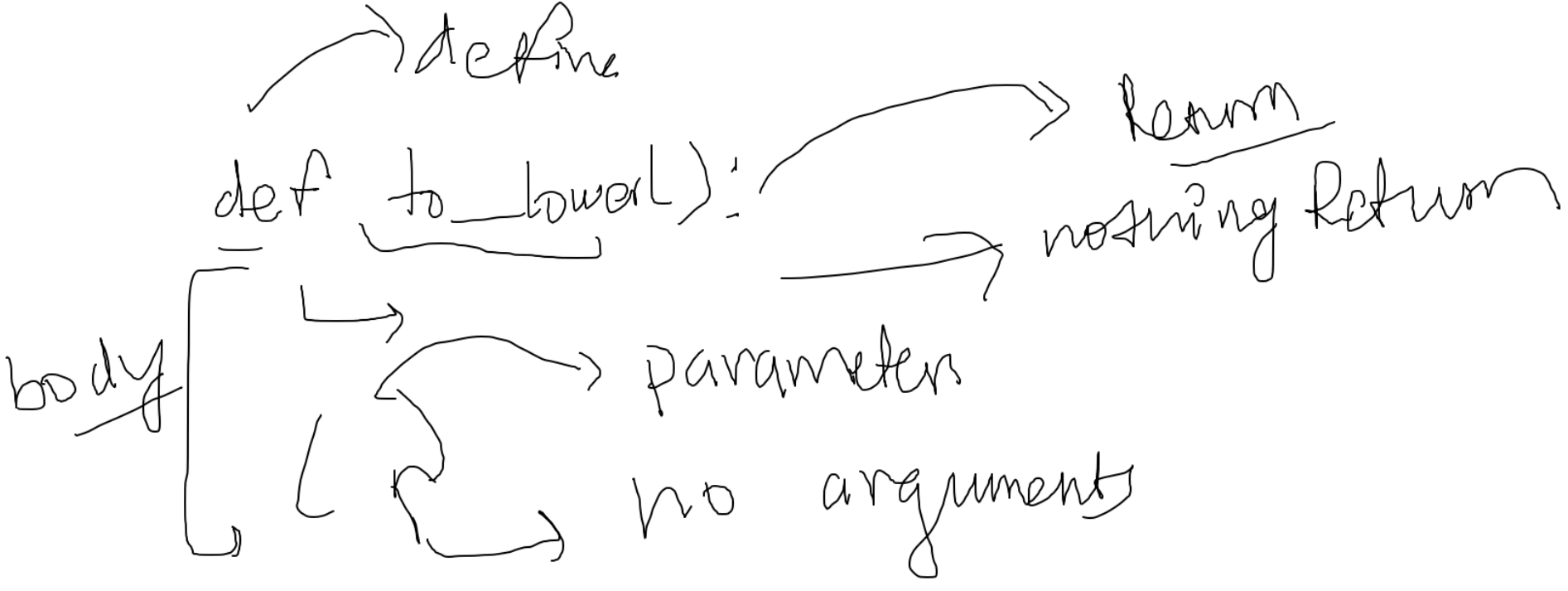
* Function : len(), input(), upper() lower()



* Function : len(), input(), upper() lower()

parameters.

Aer... 1)

define

def to_lower():

Return

nothing Return

body

parameter

no arguments

arguments

* Function : len(), input(), upper() lower()

parameters.

Aer... 1)

define

def to_lower():

Return

nothing Return

body

parameter

no arguments

arguments

```
def some_fun(a, b):
    sum = a + b
    return sum
```

multiple

* some_fun(2, 3) → positional

* function call, (2, 3, 4)

(3, 2, 4) → X

L)

(a) → input / information

(a, b, c) d)

(a, b, c)

(a, b, c)

* Function : len(), input(), upper() lower()

parameters.

Aer... 1)

define

def to_lower():

Return

nothing Return

body

parameter

no arguments

arguments

```
def some_fun(a, b):
    sum = a + b
    return sum
```

multiple

```
some_fun(2, 3)
```

→ positional

function call, (2, 3, 4)

(3, 2, 4) → X

L)

(a) → input / information

(a, b, c) d)

(a, b, c)

(a, b, c)

+ Function
+ Return
+ Default argument
+ Optional Argument

⊕ function → loop; if Problem Solve

⊕

⊕

* Function : len(), input(), upper() lower()

parameters.

Aer... 1)

define

def to_lower():

Return

nothing Return

body

parameter

no arguments

arguments

```
def some_fun(a, b):
    sum = a + b
    return sum
```

multiple

```
some_fun(2, 3)
```

→ positional

function call, (2, 3, 4)

(3, 2, 4) → X

L)

(a) → input / information

(a, b, c) d)

(a, b, c)

(a, b, c)

+ Function
+ Return
+ Default argument
+ Optional Argument

⊕ function → loop; if Problem Solve

⊕

⊕

+ pass → dict{
list[]

↳ function Argument.

- Dict
- List
- Built

* Function : len(), input(), upper() lower()

parameters.

Aer... 1)

define

def to_lower():

Return

nothing Return

body

parameter

no arguments

arguments

```
def some_fun(a, b):
    sum = a + b
    return sum
```

multiple

```
some_fun(2, 3)
```

→ positional

function call, (2, 3, 4)

(3, 2, 4) → X

L)

(a) → input / information

(a, b, c) d)

(a, b, c)

(a, b, c)

+ Function
+ Return
+ Default argument
+ Optional Argument

⊕ function → loop; if Problem Solve

⊕

⊕

+ pass → dict{}
list[]

↳ function Argument.

- Dict
- List
- Set

Basic Syntax

```
def function_name():
    do something here

def main():
    call the function from here

if __name__ == "__main__":
    main()
```

indentation

under double underscore

- modify list in function
- Continue Next week
- complete 2 exercise file

+ Correction
+ Youtube Video (full watch)