Parameters → func-name → return

reuse → organize → debug

Code

str.isalpha() →→ True
len(—) →→ 5
eval(—) →→ 7
max(5,2,3) →→ 5

5 3

def func-name (P1, P2, ...) :

func_name(5, 3)

---

weight   height

$$bmi = \frac{weight}{(height/100)^2}$$

→ get_bmi(w, h)

< 18.5 Underweight
< 25   Normal
< 30   Overweight
        Obese

→ get_status(bmi)

---

mark   full_mark

$$Pct = \frac{mark}{full\_mark} * 100$$

→ get_Percent(m, fm)

> 85   Excellent
> 75   V. Good
> 65   Good
>= 50 Pass
       Fail

→ get_grade(Pct)

Collection Datatype

* List ✓
* tuple
* set
* dict ✓

$x = [5, 17, 13, 7]$

print(x)

print($x[0]$) ⟹ 5

print($x[-1]$) ⟹ 7

print(len(x)) ⟹ 4

$x.$ ⟵ methods

index	value	neg index
0	5	-4
1	17	-3
2	13	-2
3	7	-1

Enter num of std.: 5

Enter std. mark: 66

~ ~ ~ : 35

~ ~ ~ : 70

~ ~ ~ : 44

~ ~ ~ : 58

best = max(—)

marks

value	grade
66	A
35	D
70	A
44	C
58	B

A ⟹ m >= best - 10   (60)

B ⟹ m >= best - 20   (50)

C ⟹ m >= best - 30   (40)

D ⟹ m >= best - 40   (30)

F ⟹ else

Enter num of emps: 4

Enter emp. salary: 7000

~ ~ ~ : 12000

~ ~ ~ : 5000

~ ~ ~ : 15000

$avg = \dfrac{sum(—)}{len(—)}$ ⟹ 10000

Salaries

salary	count
7000	0
12000	1
5000	2
15000	

$x = 5$

$y = x$

$y += 3$

x	y
5	5
	8

copy Value

$x = [10, 20, 30]$

$y = x$

$y[2] += 3$

copy Reference

index	value
0	10
1	20
2	30 / 33

dob = "22/6/2002"

x = dob.split("/")

y = "-".join(x)

x

0	22
1	6
2	2002

y

22-6-2002

split()

Str → list

join()

---
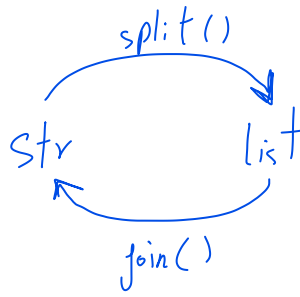
y ⟹ list

0	Saudi Arabia
1	Arab Emirates
2	Egypt
.	
.	
.	

print(y[1])

print(x["ae"])

x ⟹ dict

966 sa	Saudi Arabia
971 ae	Arab Emirates
20 eg	Egypt

71
52
13
50
48

A ⟹ m >= best-10   61

B ⟹ m >= best-20   51

C ⟹ m >= best-30   41

D ⟹ m >= best-40   31

f ⟹ else

A	0 1
B	0 1
C	0 1 2
D	0
f	0 1

Salaries

7000
13000
5000
15000

S > 12000

S > 6000

high	0 1 2
Normal	0 1
low	0 1