# Functions

define

invoke

return values ( 0 values = None

|

>1 values = Tuples

Tuple assignment

a, b = e

# Functions are Values

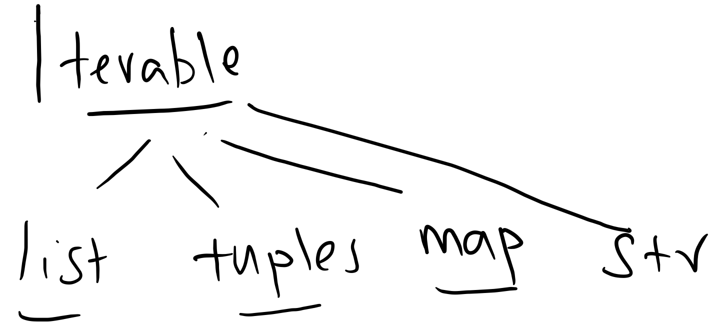
What can we do with values?

- Store values in variables
- Supply values as arguments to functions.
- Return values from functions. (5+3)

| Values | Expressions<br>Not-a-value |
|---|---|
| 1, 2 | 5+3 ? |
| 3.14 | 1/0 |
| "Hello" | [1,2,3][0] |
| [1, 2, 3] | |

mix of operations and functions.

Operations = functions



$\text{list}(\underset{\uparrow \atop \text{Iterable}}{\underline{\text{seq}}}) \rightarrow$ list that has same elements as seq in the same order.

for x in $\overbrace{\text{"abcd"}}^{\text{iterable? Yes}}$: "a" "b" "c" "d"
    Print(x)

for $\widetilde{x, y}$ in $\underbrace{[(1,2), (3,4)]}_{\text{iterable? Yes}}$:
    Print(x+y)

x, y = → (1,2)

(3,4)

(X,Y)= (1,2)

X=1

Y=2

Print(x+y) -> 3

X,Y=( 3,4 )

X = 3

Y = 4

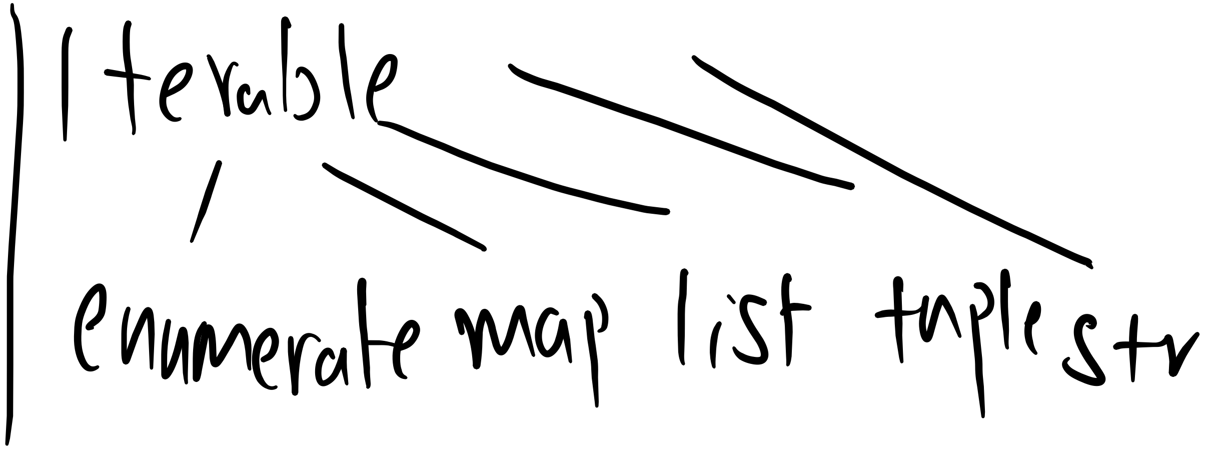
Print(X+Y) → 7

X,Y = (5,6)

X=5

Y=6

Print(X+Y)= 11



Iterable

enumerate map list tuple str

function

input

enumerate( seq ) → output

seq'

seq'[0] = (0, seq[0])

seq'[1] = (1, seq[1])

⋮

enumerate("abc")

seq 2

yes

seq[0] = "a"

seq[1] = "b"

seq[2) = "c"]

seq'[0] = (0, "a")

seq'[1] = (1, "b")

⋮