

Ex. NestedLoop.py { for loop
while loop

template for while loop:

```

while cond:
    statement
    statement
#end while
statement
    
```

bool expression
loop repetition condition
(LRC)

while loop body

common form

initialize loop control var (LCV)

while (test LRC):

start.

increment LCV

Ex. Sum.py

$$S = 1 + 2 + 3 + \dots + n$$

$$sum = 0$$

$$sum = sum + 1$$

$$sum = sum + 2$$

$$sum = sum + 3$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$sum = sum + 10$$

sum

0

1

3

6

10

15

21

28

36

45

55

$$S = n + (n-1) + (n-2) + \dots + 3 + 2 + 1$$

$$S + S = (n+1) + (n+1) + (n+1) + \dots + (n+1) + (n+1) + (n+1)$$

$$2S = n(n+1)$$

$$S = \frac{n(n+1)}{2}$$

$$n = 10 \therefore S = \frac{10 \cdot 11}{2} = 5 \cdot 11 = 55$$

4th way

row										+ =	1
1	1										2
2	1	1									3
3	1	1	1								4
4	1	1	1	1							5
	:	:	:	:	:	:	:	:	:		
n	1	1	1	-	-	-	-	-	-	1	$\frac{n(n+1)}{2}$

Ex. Fibonacci.py

Fibonacci sequence

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2} \quad (\text{for } n \geq 2)$$

ind. 0 1 2 3 4 5 6 7 8 9

$$F = (0, 1, 1, 2, 3, 5, 8, 13, 21, 34, \dots)$$

.. next time - - -

Fibonacci2.py $\leftarrow$ recursive function

Euclid.py $\leftarrow$ finds GCD