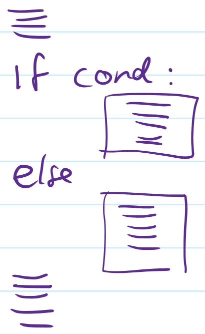
5 Loops (iteration)

Monday, February 19, 2024 2:02 PM

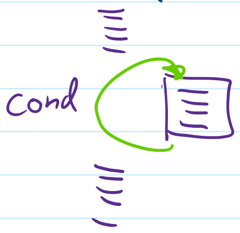
Remember:

- Conditional execution



- Conditional repetition.

"repeat something while some condition is true"



- WHILE

Syntax : while cond :
    block

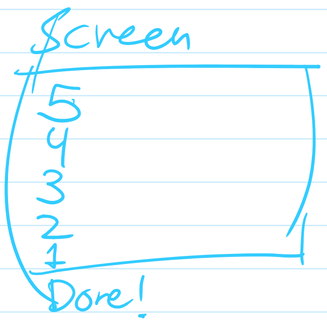
repeat block while cond is true.

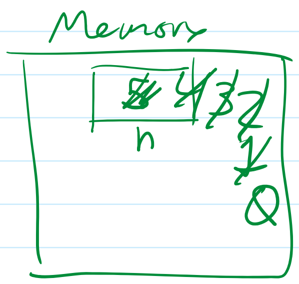
E.G. Counter loop.

```
n = int(input('Number? '))

while n > 0 :
    print(n)
    n = n+1

print('Done!')
```





Note: If the condition is never falsified, an "infinite loop" is possible

"My Program hanged"

E.G User controlled loop:
you ask the user for an input that causes
the condition to be false.

```
s = input('Name? ')

while s != 'X' :
    print('Hello', s, 'your name has',
len(s), 'letters')
    s = input('Name? ')

print('Done!')
```

E.G. Euclid's Algorithm.

- Obtain the G.C.D of two numbers.

1327 125

the biggest number that divides both 1327 and 125

- if $a=b$ then $gcd(a,b)=a$
- if $a>b$ then $gcd(a,b) = gcd(a-b,b)$
- if $a<b$ then $gcd(a,b) = \underbrace{gcd(a, b-a)}$

repeat.

notice, these numbers are smaller.

$gcd(1327, 125)$
$gcd(1102, 125)$
⋮

```
a = int(input('a? '))
b = int(input('b? '))

while a != b :
    if a > b :
        a = a - b
    else:
        b = b - a

print('The GCD ', a)
print('Done!')
```

E.G. Collatz Conjecture. 3n+1 conjecture.

$$\begin{tikzcd}
37 \arrow[r, bend right, "3n+1"'] & 112 \arrow[r, bend left, "n/2"] & 56 \arrow[r, bend left] & 28 \arrow[r, bend left] & 14 \arrow[r, bend left] & 7 \arrow[r, bend right, "3n+1"'] & 22 \arrow[r, bend left] & 11 & \cdots\cdots & \textcircled{1}
\end{tikzcd}$$

```
n = int(input('n? '))

while n > 1:
    if n%2 == 0 :
        n = n // 2
    else:
        n = n*3 + 1
    print(n)

print('Done!')
```

- FOR loop

motivation:- loops are very commonly used to iterate over the ements of a collection.
"for" loops provide a compact way to do this

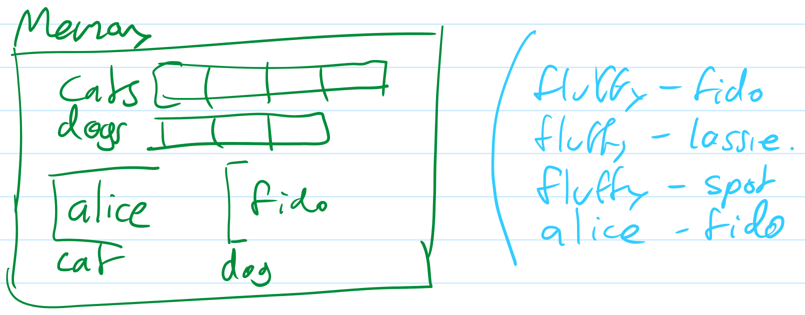
Syntax for var in container:
block.

In a for loop, var takes an element of the container, once per iteration.

Container can be any sequential type.:- lists
tuple.
string
dictionary.

- block can contain a for loop or a while loop
i.e. loops can be nested.

e.g



```
cats = ['fluffy','alice','grumpy','henry']
dogs = ['fido','lassie','spot']

for cat in cats :
    for dog in dogs :
        print(cat,'-',dog)
```

- WHILE vs FOR

while:- when you do not know how many times you need to repeat.
this is, how many iterations to perform.

for.- when you do know beforehand how many times to repeat.
this is. how many iterations to perform.

e.g Sum. vs find.

- Loops have an ELSE section.

```
while cond:                for cond :
    block 1                    block1
else:                      else:
    block2                     block2
```

block2 is executed when the repetition ends.

- GENERATORS

A "function" intended to generate elements of

- GENERATORS

A "function" intended to generate elements of a collection. elements are generated as needed one at a time.

- range (13) → [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
- range (7, 13) → [7, 8, 9, 10, 11, 12]
- range (3, 13, 2) → [3, 5, 7, 9, 11]

- enumerate(list)

  enumerate(['a', 'b', 'c']) → [(0, 'a'), (1, 'b'), (2, 'c')]

- zip(list1, list2)

  produces a list of pairs where the first element is from list1 and the second one from list2.

—o— EOF