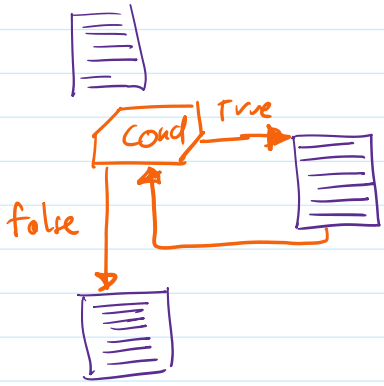
5 Loops (iteration)

Friday, September 27, 2024 8:02 AM

- CONDITIONAL REPETITION

Repeat something while some condition is true



- WHILE loop

```
while cond :
    block
```

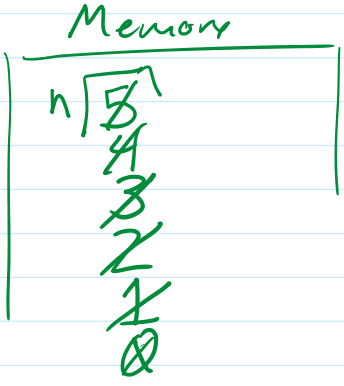
repeat block while cond is true.

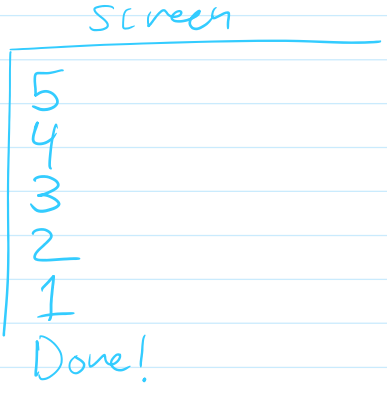
Demo: A counting loop.

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

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

print('Done!')
```





Note: if the condition is never falsified your program will get stuck in an **infinite loop.**

the code block should have a way to falsify the condition.

Demo: User controlled loop:
ask the user input and stop the loop on a specific input.

ask the user input and stop the loop on a specific input.

Repeatedly: ask for a name, say "hello"
finish when the name is "oops"

```
name = input('-Name? ')

while name != 'oops' :
    print('Hello', name,'!')
    name = input('=Name? ')

print('Done!')
```

Demo: Euclids Algorithm.

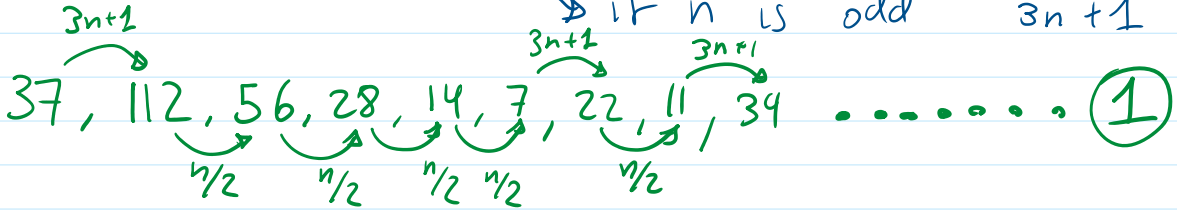
```
a = int(input('a? '))
b = int(input('b? '))

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

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

Demo: The 3n+1 game (Collatz) conjecture.

Given a number n: if n is even, n/2; if n is odd, 3n+1




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

while n != 1 :
    if n%2 == 0 :
```

```
            n = n // 2
        else:
            n = 3*n + 1
        print(n)

    print('Done!')
```

- FOR loop

motivation:- the most common use of loops is to do something for every item in a collection.

Syntax:

```
for var in collection :
    block
```

i.e. var will take the value of each item in collection, and then block is executed.

collection can be any sequencial type of
- list
- tuple
- set
- dictionary
- string

loops can be nested.
A for loop can have a while loop inside or a for loop, or a while loop .....

Demo.-

```
cats = ['chunky','grumpy','henry','bill']
dogs = ['fido','lassie','spot','snoopy']

for c in cats :
    for d in dogs :
        print(c ,'vs',d)
```

```
chunky vs fido
chunky vs lassie
chunky vs spot
chunky vs snoopy
grumpy vs fido
grumpy vs lassie
grumpy vs spot
grumpy vs snoopy
henry vs fido
henry vs lassie
henry vs spot
henry vs snoopy
bill vs fido
bill vs lassie
```

bill vs spot
bill vs snoopy

- WHILE vs FOR

while :- when it is not known beforehand how many times something is going to repeat.

for :- when it is known beforehand how many times something is going to repeat.

e.g.
sum → for.
find → while.

- GENERATORS

A "function" intended to generate a collection of items. items 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]

Demo:

how many numbers between 0 and 10,000 are divisible by 7 and 11 ?

consider all numbers in 0 to 10000 and test.
Keeping a counter.

```
counter = 0
for i in range(10000) :
    if i%7 == 0 and i%11 == 0 :
        counter = counter + 1
```

- 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, the second element from list2.

EOF