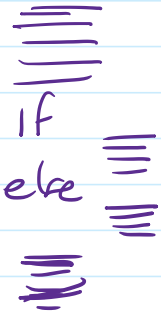
Ch5 Loops

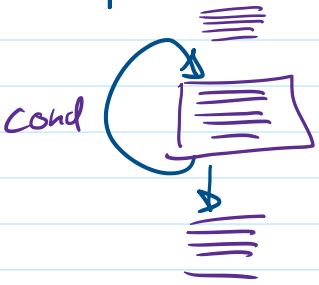
Monday, October 2, 2023 8:27 AM

1- Conditional exec.



2- Repetition:-

"repeat something until some condition is true"



- The python "while" loop.

Syntax `while cond :`
`    block`

repeat block while condition is true.

NOTE: infinite loop is possible.

e.g Counter loops.

1 2 3 4 9 10

X

e.g. user loop.

eg. Euclid's Algorithm. G.C.D.

1237 152

a b

The biggest number that divides both a and b evenly

a). if $a = b$ then $gcd(a,b) = a$

b). if $a > b$ then $gcd(a,b) = gcd(a-b, b)$

$= gcd(a \bmod b, b)$

c). if $b > a$ then $gcd(a,b) = gcd(b-a, a)$

$gcd(b \bmod a, a)$

E.g. Collatz conjecture: by Lothar Collatz.

$3n+1$

37 112 56 28 14 7 22 ... 1

$3n+1$ $n/2$ $3n+1$

If you follow this procedure the chain stops at 1

- The python "for" Loop

Note: "for" loops in python are different than in other languages.

```
for variable in container :
    [block
```

← variable takes the value of an element in the container once per iteration.

eg. a string.

per iteration.

eg. a string.
eg. a list.

- range(N)

  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(7, 13, 2) ⇒ [7, 9, 11]

eg: nested for loop.

- 'while' vs 'for.'

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

  for :- when you do know beforehand how many times to repeat, how many iterations to perform.

e.g: sum vs. find.

- Loop 'else'

```
while cond:
    block 1
else:
    block 2
```

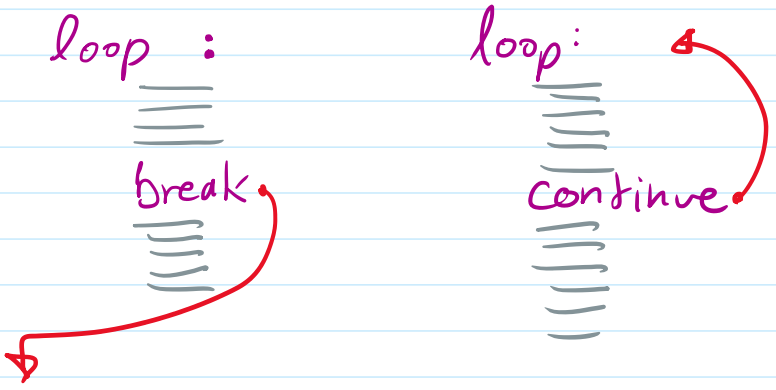
```
for var in container:
    block 1
else:
    block 2.
```

block2: is executed when the loop ends.

- 'break' 'continue'

  break :- exits a loop.

  continue :- skips to the next iteration.



- generator range(n) ≈ [0, 1, 2, 3, ..., n-1]

  enumerate(list)

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