Basic Types

integers

floats

-4 -3 -2 -1 0 1 2 3 4

booleans: True
           False

NoneType: None

Variables
Python:

```
x = 1          ← variable
               ← value that variable is
                 equal to
y = "hello world"
x = 3.14
x = x + 1   →   x += 1
                x = x + 1
```

$y += "!"$ → $y = y + "!"$

Virtual computer world

x ← 4.14
y ← "hello world!"

Conditionals
if statement

```
x = -1

if x < 0:
    x = -x

print (x)
```

x < 0

True

x = -x

False
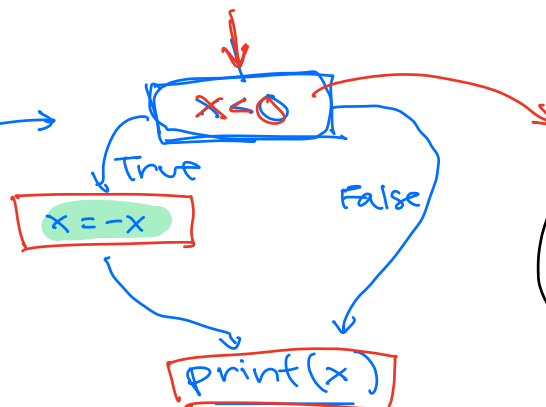
print(x)

Output:
1

Virtual computer world

x ← 1

if-else statement

```
x = -1

if x < 0:
    print ("x is negative")
else:
    print ("x is not negative")
```

True

x < 0

False

rest of
code
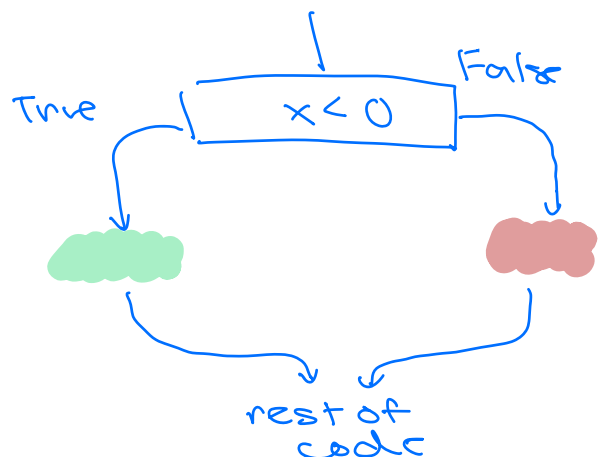
if - elif - else statement:

```
x = 1
y = 2
if x == y:
    print("x equals y")
if x > y:
    print("x greater than y")
else:
    print("x less than y")
```

Virtual comp. world

$x \leftarrow 1$
$y \leftarrow 2$

True   False
x == y

True   False
x > y

Continue w/ program
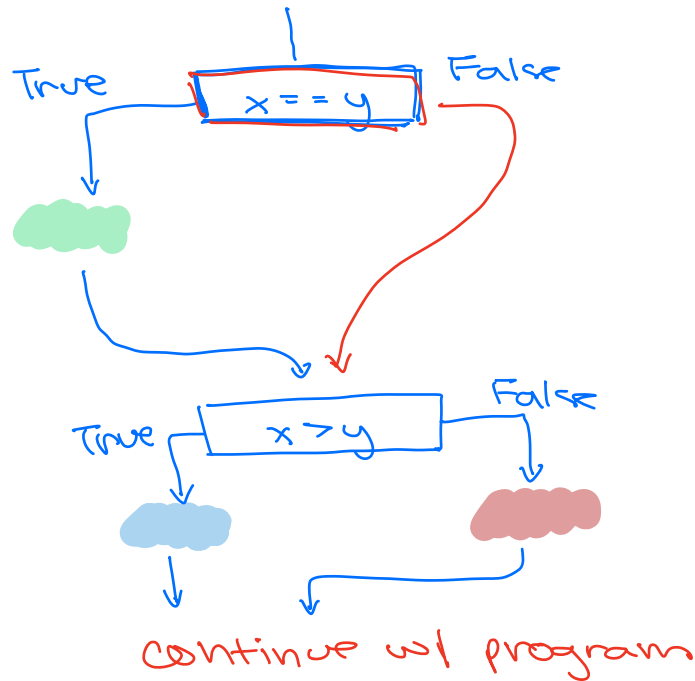
Strings

x = "Hello" →

" H e l l o "
  0 1 2 3 4
 -5 -4 -3 -2 -1

indexing

indices

print(x[0]) → "H"
print(x[4]) → "o"

print(x[-4]) → "e"

(slicing)   x = "Hello"   "Hello"
                              0 1 2 3 4
                                   ↑
                                  end
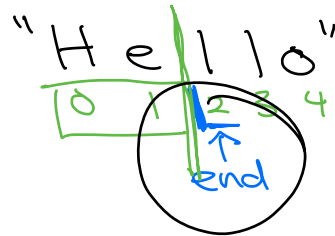
print (x [1 : -1]) → "ell"
print (x [1:]) → "ello"
print (x [: 2]) → "He"

Lists

| 1 | 2 | "blue" | -7.8 | None |
| 0 | 1 | 2 | 3 | 4 | indices
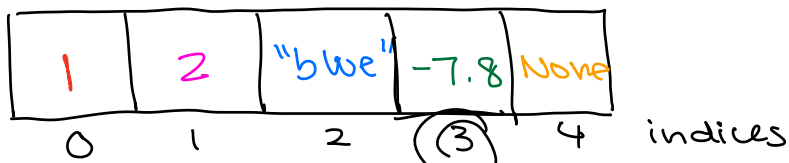
x = [1, 2, "blue", -7.8, None]
print (x[0])   1
print (x[3])   -7.8

Virtual comp world

x = [ ]
x.append (1)   #[1]
x.append (2)
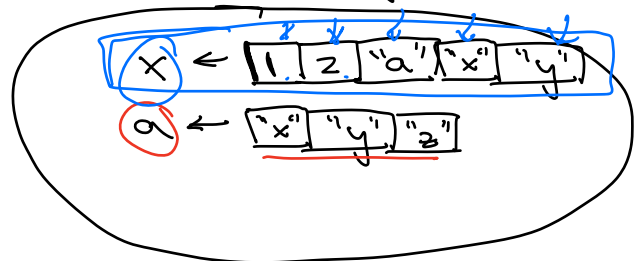x.append ("a")

a = ["x", "y", "z"]

x. extend (a)
print (x)   →   [1, 2, "a", "x", "y", "z"]
x.pop()
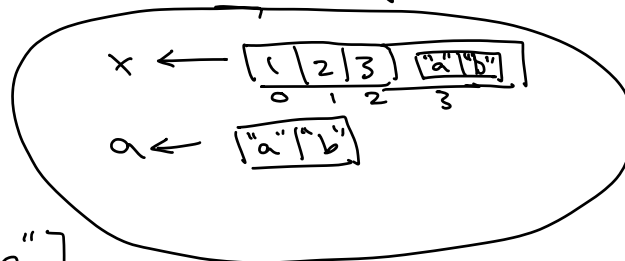print (x)   →   [1, 2, "a", "x", "y"]

x = [1, 2, 3]
a = ["a", "b"]

x.append(a)

print(x[3]) → ["a", "b"]

Virtual comp world

x ← | 1 | 2 | 3 | "a""b" |
      0   1   2   3

a ← | "a" | "b" |

Loops   (iteration) ⤷ run a block of code repeatedly

⤷ 2 kinds: for loops. while loops

For loops - iterate over sequence

x = ["c", "a", "t"]

VCP

x ← | "c" | "a" | "t" | → letter

for letter in x:
    print(letter)
print("done")
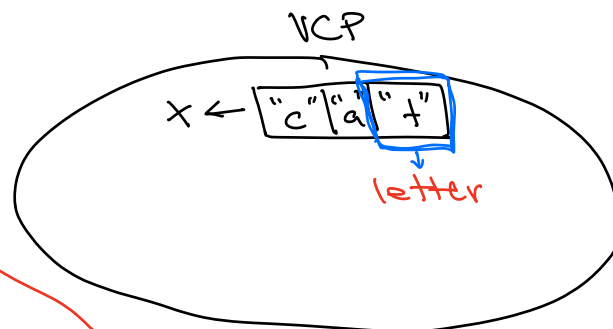
for *variable name* (in) *sequence*:
    code that we want to repeat

Output
c
a
t
done

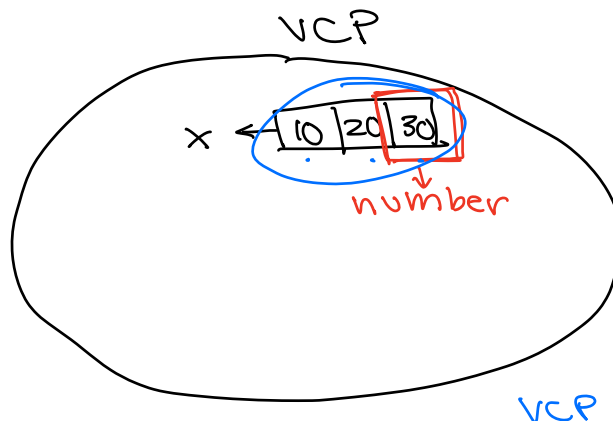x = [10, 20, 30]

VCP

x ← | 10 | 20 | 30 | number

for number in x:
    print(number+1)
print("done")

VCP

_____

x = 10
x -= 5 → x ⟵ 10 5

x = x - 5
10 - 5 = 5
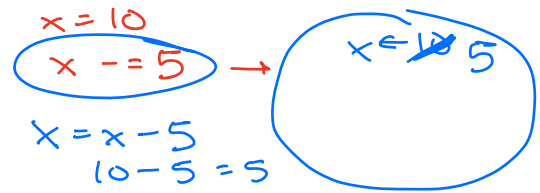
x = [10, 11, 30]
doubles_x = [ ]

for number in x:
    doubles_x.append
        (2*number)

print(x)
print(doubles_x)

Output
_____

[10, 11, 30]
[20, 22, 60]

VCW
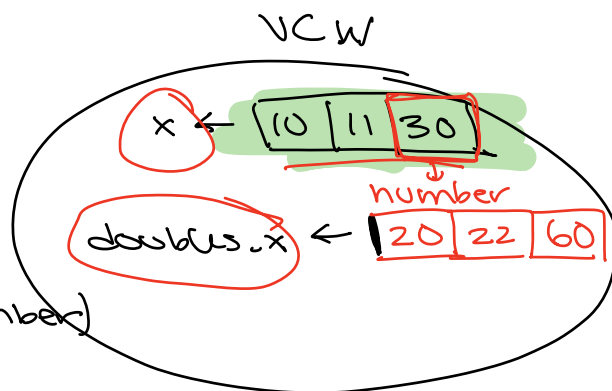
x ⟵ | 10 | 11 | 30 |
              number
doubles_x ⟵ | 20 | 22 | 60 |

Loop over integers
    for i in range(2, 25, 2): ← step.   :
        print(i)
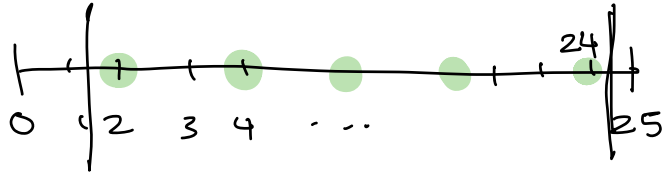
Output

    2 | | 1 | | 1 | | | | 24| |
    4   0 |2   3 4   · · · |25
    ⋮
    24

            [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

while loop:
                                                    VCW

    while *condition*:   x ← 1̶ 2̶ 3̶ 4̶ 5
        repeated code

                                                                    Output
    x = 0   5<5? NO   1
    while x<5:   True ┌──────────┐   2
        x += 1 False   │ a == 1 │   3
        print(x)   └──────────┘   4
                                         x += 1   5
    print("yeet")   print(x)

                                              continue w/ program?
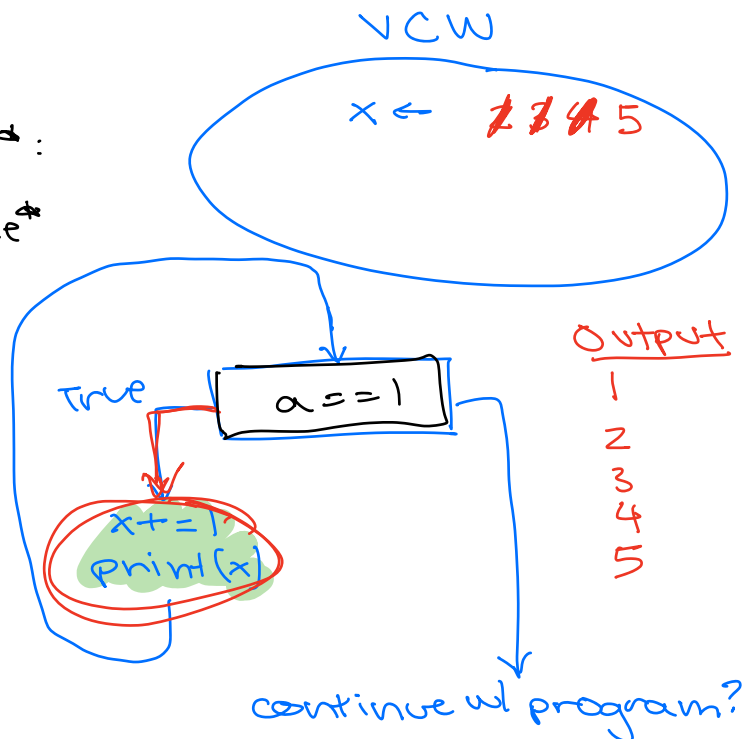
    a = 1   infinite
    while a == 1 :   loop
        print(a)

```
computer_guess ← 5
guess ← 1

while [condition]:
    [code]
```

True   [condition]   False

code

continue