```
i = 0
b = (i < 5)
print(b)

i = i+1
```

```
b = (i<5)
print(b)

i = i+1
b = (i<5)
print(b)

i = i+1
b = (i<5)
print(b)

i = i+1
b = (i<5)
print(b)

i = i+1
b = (i<5)
print(b)
```

```
print('START')                    (1)

i = 0                             (2)

while i<5:                        (3)
        print('i:', i)
        i = i+1

print('END')                      (4)
```

O/P:

```
START
i: 0
i: 1
i: 2
i: 3
i: 4
END
```

i = 0 | val(i) is set to 0

i < 5 → value(i) < 5 → 0 < 5 → True

∴ execute the body.

print(i) → print(value(i)) → print(0)

i = i+1

RHS. i+1 → value(i) +1 → 0+1 → 1

i = 1 → value i is set to 1

Go back to while cond.

i < 5 | value(i) < 5 | 1 < 5 | True

∴ execute the body.

print(i) → print(value(i)) → print(1)

i = i+1

RHS: i+1 ∴ value(i)+1 ∴ 1+1 : 2

i = 2 value of (i) is set to 2

Go back to while cond

i < 5 ≡ value(i) < 5 ≡ 2 < 5 ≡ True.

∴ execute the body.

print(i) ≡ print(value(i)) ≡ print(2)

i = i+1

RHS: i+1 ≡ value(i)+1 ≡ 2+1 ≡ 3.

i = 3 set value of i to 3.

Go back to while condition.

Cond ≡ i<5 ≡ value (i) <5 ≡ 3<5 ≡ True.

∴ execute the body.

print (i) ≡ print (value (i)) ≡ print (3)

i = i+1

RHS ≡ i+1 ≡ value (i)+1 ≡ 3+1 ≡ 4.

i = 4 ≡ value of i is set to 4.

Go back to while condition

Condition: i<5 ≡ value (i) <5 ≡ 4<5 ≡ True.

∴ execute the body

print (i) ≡ print (value (i)) ≡ print (4)

i = i+1

RHS. ≡ i+1 ≡ value (i)+1 ≡ 4+1 ≡ 5

i = 5 ≡ value of (i) is set to 5

Go back to while condition.

Condition ≡ i<5 ≡ value (i) <5 ≡ 5<5 ≡ False.

∴ Go to the next step.

Anto Gowdy ∴ God lies in the details.

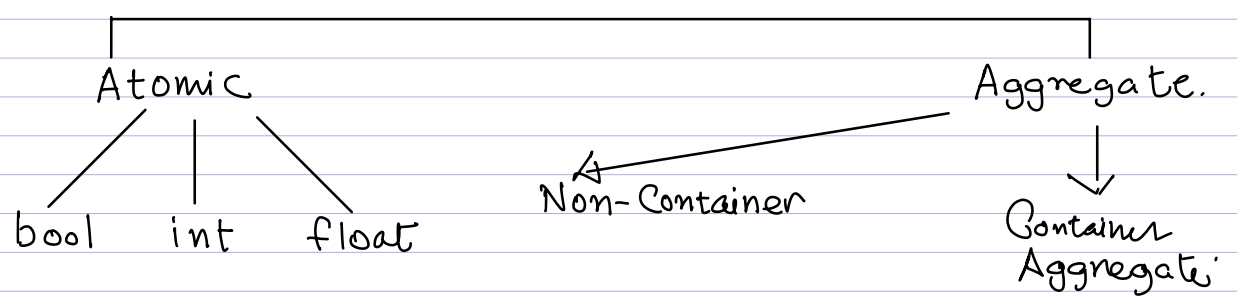
5, 98, 17, 152, 47

9899123482, 9899134218O, 98991 4423,
0x1124FFB49, 0o17723461, 9899442189S

① take a variable named max, assign it the first number in the list

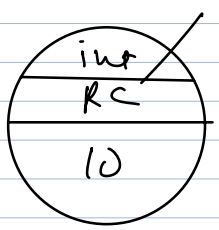
② for each of the following number in list
compare current number with max
if greater then update max with the current number.

Data Types



Atomic = Indivisible into more basic parts.

Atom



Atomic X Aggregate

↓ indivisible, fundamental, elemental. not made up of more basic parts

↓ Something which is made by combining more fundamental level entities.