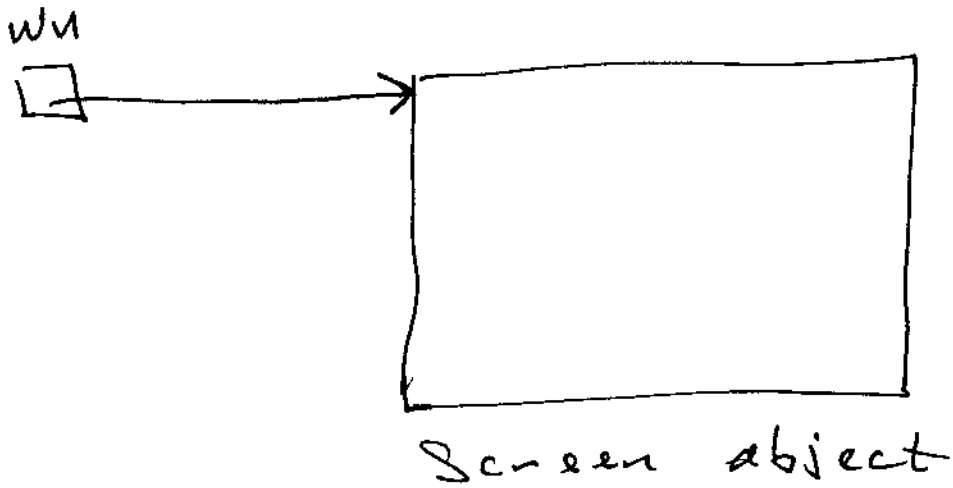


css 20 4-16-20

1

`wn = turtle.Screen()`



Ex
`Turtle3.py`

Iterative control structures (loops)

often have a block of instructions that should be repeated some # of times.

Ex. Turtle 3.py

lines 16-21 : draw Δ

" 26-33 : draw $\square$

new control structure: for loop

for variable in list :

 → do something

 ⋮

 → do something

 # end for

loop control variable
 loop body

- Always use spaces, not tabs, to indent
- after first indentation, cannot change width
- recommend 3 spaces

Ex Loop1.py

more in future about lists (or list objects.)

for time being:

['one', 'two', 'three']

list of
strings

[1, 2, 3]

ints

[1.0, 2.0, 3.0]

floats

[1, 'two', 3.0]

mixed objects

Ex. Loop2.py

use built-in `range()` to construct a list.

Ex. Loop3.py

Ex. Loop4.py

- `range(a, b)` creates a list from `a` up to `b-1`, (`a` and `b` are ints)

$[a, a+1, a+2, \dots, b-2, b-1]$

- `range(n)` creates a list from 0 to `n-1`.

$[0, 1, 2, \dots, n-1]$

note $+=$ is a compound assignment

Operator!
shorthand

$$a += b$$

same as

$$a = a + b$$

memory:

	a	b
before	2	3
after	5	3

$$a -= b$$

$$a = a - b$$

$$a *= b$$

$$a = a * b$$

$$a /= b$$

$$a = a / b$$

$$a // = b$$

$$a = a // b$$

$$a \% = b$$

$$a = a \% b$$

Loop 4. Py Computer sum

$$\frac{1}{1} + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots + \frac{1}{1000}$$

$$= \sum_{k=1}^{1000} \frac{1}{k}$$

$$\text{sum} = 0.0$$

$$\text{sum} = 0.0 + 1.0 = 1.0$$

$$\text{sum} = 1.0 + 0.5 = 1.5$$

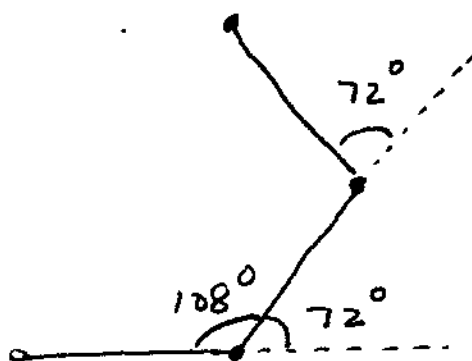
$$\text{sum} = 1.5 + 0.33\dots = 1.83\dots$$

⋮

$$\text{sum} = \dots$$

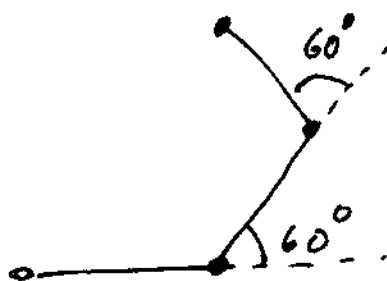
Ex. TurtleLoop1.py

Pentagon:



$$\frac{360}{5} = 72$$

Hexagon:



$$\frac{360}{6} = 60$$