Lecture 15, 25 September 2025

Madhavan Mukund

https://www.cmi.ac.in/~madhavan

Programming and Data Structures with Python

ADTs, Classes, Objects

2D Point, (x,y) coordinates

Constructor `__init__`

translate(dx,dy)

odistance()

```
p = Point(3,4)
q = Point()
```

$\Rightarrow (0,0)$

Special functions

`__str__`

`__add__`

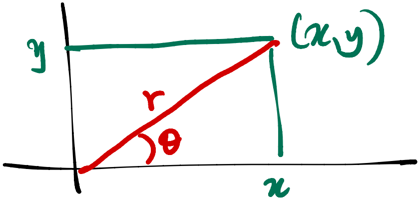
`__lt__`

```
p <= q   ≡   p < q   or   p == q
```

```
p.x < q.x          p.x == q.x
   and       or       and
p.y < q.y          p.y == q.y
```

Change implementation

$(x,y) \rightarrow (r,\theta)$



$r = \sqrt{x^2+y^2}$

$\theta = \tan^{-1}\left(\frac{y}{x}\right)$

$x = r\cos\theta$

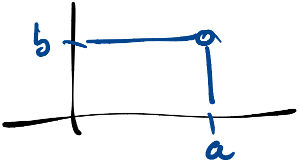
$y = r\sin\theta$

Interface remains (x,y)

$p = \text{Point}(a,b)$

$\Downarrow$

__init__ should compute and store r, θ



odistance $\longrightarrow$ report r

translate $\longrightarrow$ $(r,\theta) \rightarrow (x,y) \rightarrow (x+dx, y+dx) \rightarrow (r',\theta')$

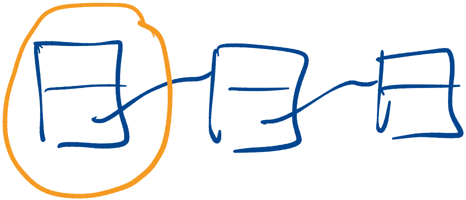
l.append(v) vs append(l,v)

self

List

List.append(l,v)

List



Dictionaries, nested

List → First Node

↓

Next Node

↓

⋮

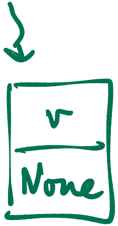
class Node

value

next

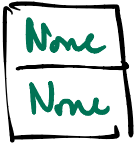
```
class list:
    def __init__(self, a):
        self.value = a
        self.next = None
```

l = list(v)



What does an empty list look like?

l = None



No holes — no other value in a list is None

```
class List:
    def __init__(self):
        self.value = None
        self.next = None
```

Begin with empty list

```
    def is_empty(self):
        return (self.value == None)
```

l = List(v)

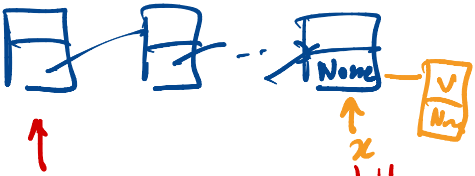
⇓

l = List()

l.insert(v)

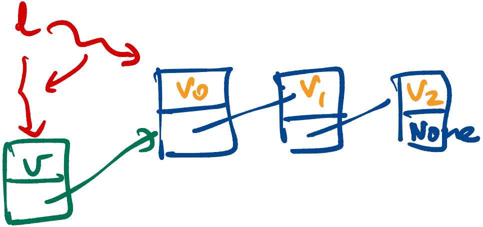
append

```
def append(self,v)
    if self.is_empty():
        self.value = v
        return
    temp = self
    while temp.next != None:
        temp = temp.next
    temp.next = List()
    temp.next.value = v
```



$x \rightarrow x.\text{Next}$ - - till $x.\text{Next}$ is None

```
def insert(self,v):
    if self.is_empty():
        self.value = v
        return
    newnode = List()
    newnode.value = self.value
    newnode.next = self.next
    self.value = v
    self.next = newnode
```




```
newnode.
  value = v
newnode.
  next = self
```