Lecture 14, 23 September 2025

Madhavan Mukund

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

Programming and Data Structures with Python

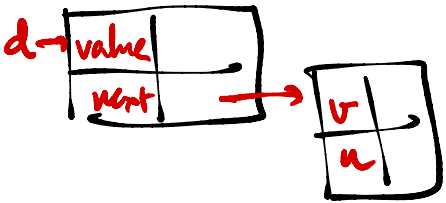
Abstract datatypes

Implementation using nested dictionaries

createlist()

insert (l, v) – Beginning

append (l, v) – End



Instead

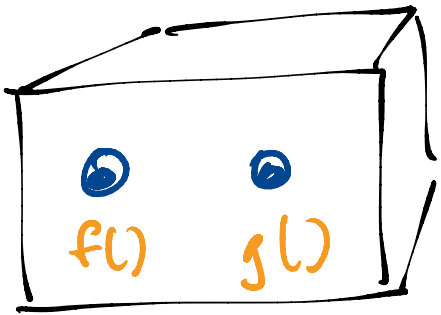
createlist → []

insert(l,v) → l.insert(0,v)

append(l,v) → l.append(v)

ADT

Interface



Implementation

Insides - "black box"

Initial brute force

⇓

Optimize

Object Oriented Programming

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

Example – 2D points

(position, attribute)

↓

(x, y) – "state"

Move a point

- translate (Δx, Δy)
- rotate

"Reporting"

distance from (0,0)

Template for ADT - "class"

```
class Point:
    def __init__(self,u,v):
        self.x = u
        self.y = v
    def translate(self, dx,dy):
        self.x = self.x + dx
        self.y = self.y + dy
```

executes some code, what fn?

"constructor"

"createpoint(a,b)" → λ

p = Point(3,7)

creates instance of Point

(3,4) → (4,6)

= object

p.translate(1,2)

```
class Point:
    ≡
    def odistance(self):
        import math
        return sqrt((self.x)**2 + (self.y)**2)
```

```
len(l)
l.length()
l.len()
p.odistance()
```

p1 = Point(3,4)

p2 = Point(9,-11)

Nice way to display an object

print(v) – calls str(v) & displays

p1.x → 3

p2.x → 9

print(p1) ⇒ "(3,4)"

⇓

call str on contents of p1

↳ calls a fixed function

```
def __str__(self):
    return "C" +
           str(self.x) +
           "," +
           str(self.y) +
           ")"
```

```
print(p1)
{
p1.__str__()
{
String
↓
display
```

Default arguments

int("72") → 72

int - 1 arg or 2 arg?

Default 2nd arg = 10

x = int("72")

y = int("AB") Error

y = int("AB", 16) → 10 x 16 + 11 x 1 = 171

default value

```
def int(s, b=10):
```

```
int("72")
```

means

```
int("72", 10)
```

```
def int(b=10, s):
```

```
int("72")?
```

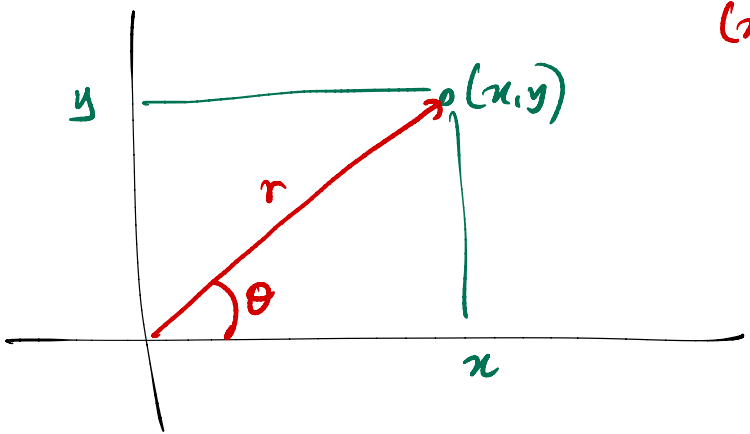
```
class Point:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y
```

```
p1 = Point(a,b)
```

but

```
p2 = Point() → (0,0)
```

Changing implementation



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

Polar