Lecture 11, 11 September 2025

Madhavan Mukund

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

Programming and Data Structures with Python

Variables & Values

Mutable vs Immutable

⇓ Call by reference

⇓ Call by value

"Aliasing"

Scope

```
def setx():
    x = 5
```

Scope of a variable/name

⇒ where it is available

setx() defines a "local" x

```
setx()
```

x? print(x)?

Scope vs lifetime

```
def setx():
    x=5
```

2 different x:

"Global" x ← 0

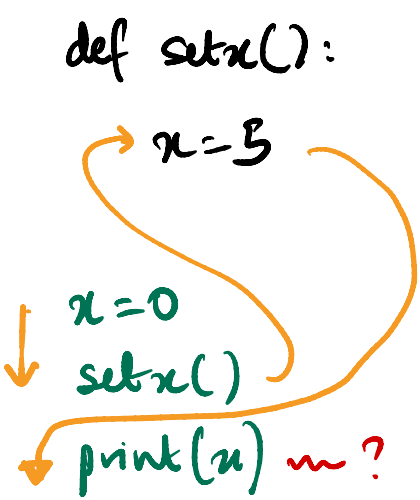
"Local" x in setx() ← 5

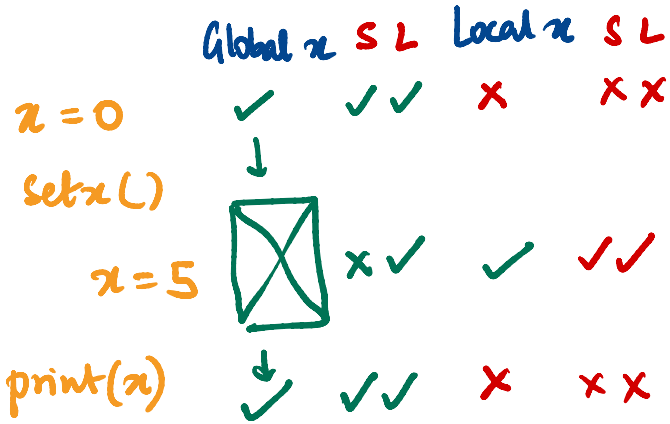
```
x=0
setx()
print(x)
```

→ ?

"Name space"

Scope Lifetime





Function calls & Memory

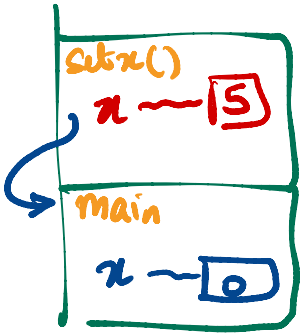
Stack

Last In
First Out

```
def setx():
    x=5
```

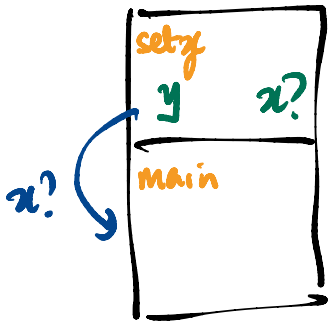
"main"

```
x=0
setx()
print(x) ~ ?
```



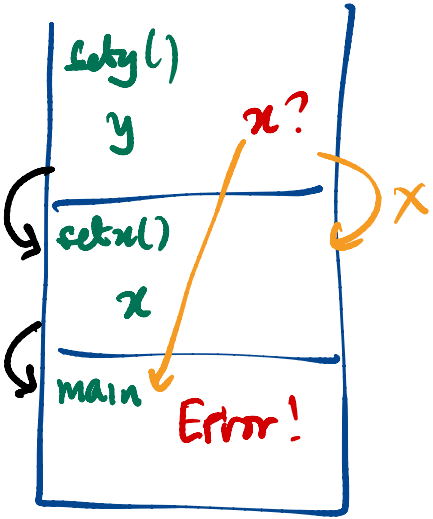
```
def setx():
    y = x+5 ?
    print(y) ?
x = 3
setx()
```

Global x is available in setx() if it is not redefined



Static Scope

```
def setx():
    y = x+5
    print(y)
def setx():
    x = 0
    setx()
setx()
```

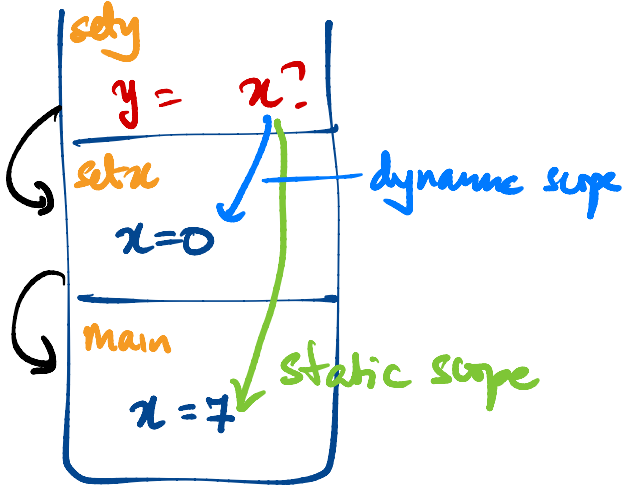


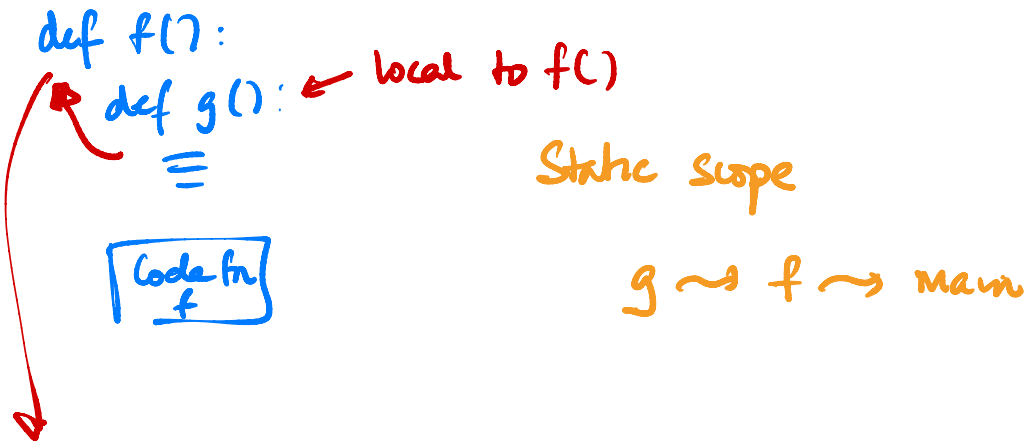
Static vs dynamic scope

```
def sety():
    y = x+5
    print(y)

def setx():
    x = 0
    sety()

x = 7
setx()
```




def f():
def g():
local to f()
≡
Code fn
f
Static scope
g ⇝ f ⇝ main

```
def scty(a):
    if a<0:
        y = 7
    else:
        x = g(a)
        y = x + 9
```

print(x) — Error

x = 22

```
if False:
    x = g(a)
y = x + 7
```

Never executed

```
def increment():
    global count
    count = count+1
```

How would you write something like this

Tell Python that both "count" are same

```
count=0
increment()
```

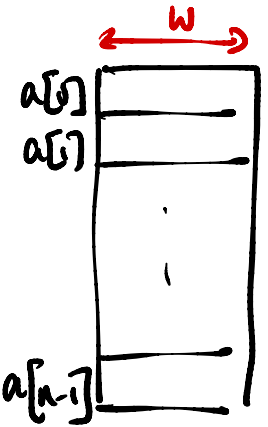
Arrays, Lists, Dictionaries

✓ ✓

Array

Contiguous block of memory

Each entry of same size

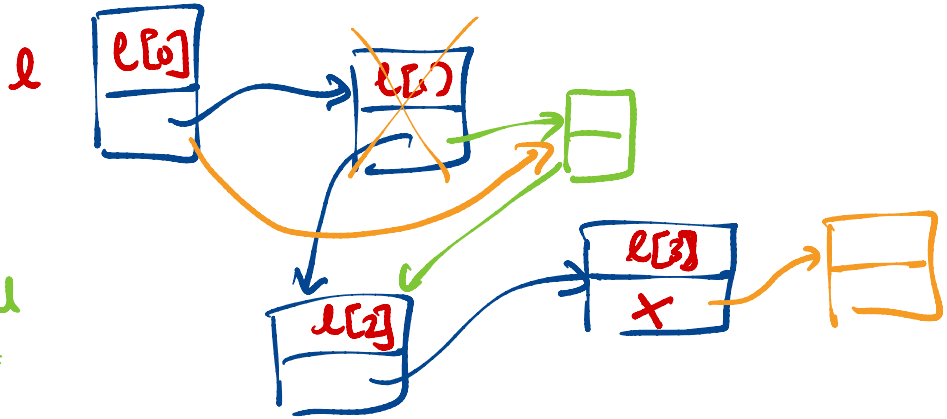


"Random access"

a[i] ? a[0] + i * w

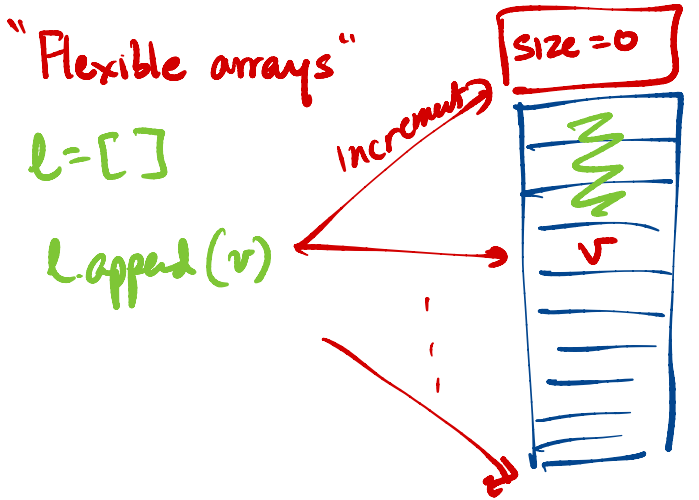
Index lookup is "constant time"

List



"Sequential access"

Python lists ?



Run out space

Allocate double the space

Copy — Not unit cost

$a[i] = v$ — Original unit cost

— One unit of copy cost

"Amortised" cost

In array

del(a[i]) Undefined a[i]

a[0..i-1] [i+1 .. n-1]

shift left

a.insert(0,v)