

Lecture 15, 10 October 2024

Madhavan Mukund

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

Programming and Data Structures with Python

Exception handling

try:

≡

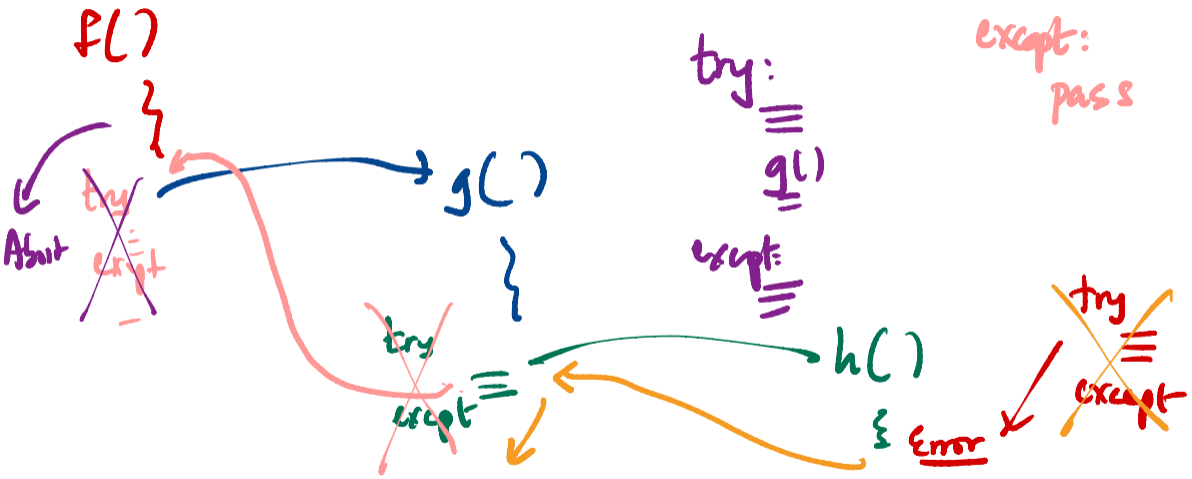
Error

Abort



except ExceptionName:

≡




Generate exception?

No negative values in List()

```
def __init__(self, initlist = []):  
    self.value = None  
    self.next = None  
    for x in initlist:  
        self.append(x)
```

```
def append(self, v):  
    if v < 0:  
        flag an error
```



"raise" an exception ("throw" an exception)

if $v < 0$:

raise ValueError ("Negative number")

Can define our own errors

NegativeListEntryError

Defining a class that "extends Exception"

How to pass the negative value with the error?

raise XYZError()
 ↑
 string

Formatting strings

print(x,y)

printf (Format spec, x, y)

printf (" --- #1 --- #2 --- ", x, y)

string with placeholders (args)

s is a string

s.format(v1, v2) → s with v1, v2 inserted
appropriately

Python for #1, #2 -- is {0}, {1}

"--- {0} --- {1} ---".format(x, y)

```
print (" --- {0} --- {1} --- ".format(x,y))
```

```
message = " --- {0} --- {1} --- "
```

```
print (message.format(x,y))
```

if $v < 0$:

raise ValueError("Negative value: {}".format(v)
+ str(v))

"Real" formatting

"Value: $\{0: \underline{3d}\}$ " . format (22)

Old C syntax $d = \text{integer}$

$3 = \text{width}$

"Value: $\{0: 6.2f\}$ " . format (33-6)

width after decimal

Input & Output

```
x = input()
```

```
y = input("Enter a number:")
```

Reads one line of input as a string

```
ynum = int(input())
```

print (a, b, c, d, e)

value(a) \searrow value(b) \searrow
 \uparrow
 separator

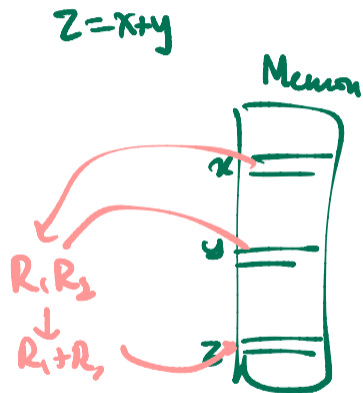
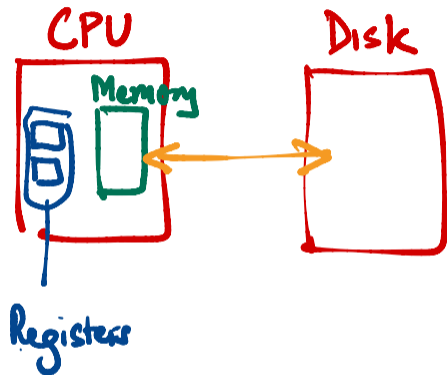
... \searrow value(c) \searrow \swarrow what happens after print

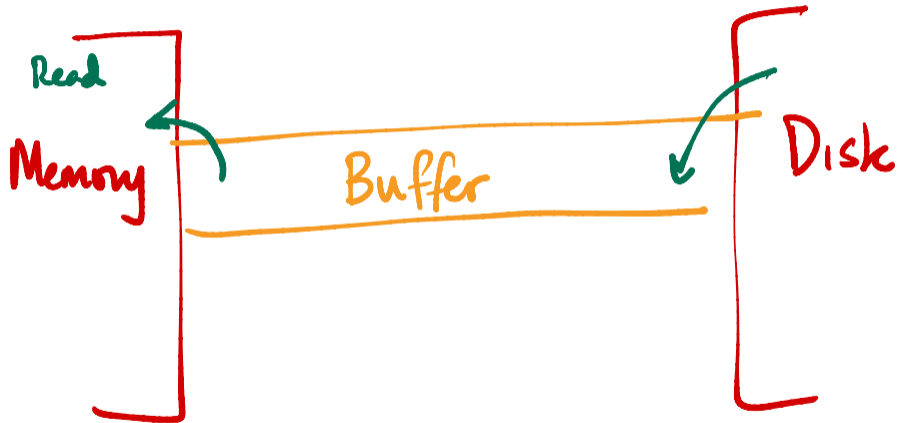
print(a, b, c, sep=",")

print(a, b, c, end="")

- use , instead of \searrow
- stay put after print

Files





Ready from file

Create a buffer linked to file

[Open a file]

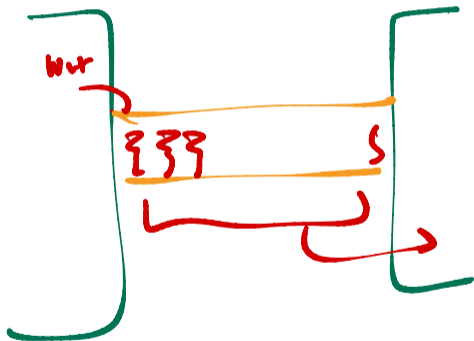
Read from buffer

Finish

Disconnect the buffer

[Close a file]

Writing



Open file

Write to buffer

Close file

= "Flushes" the
buffer