

Lecture 17, 14 October 2025

Madhavan Mukund

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

Programming and Data Structures with Python

Exception handling

try:

≡

except:

≡

Exception



Context: User input

Not always about errors

Count occurrences of
a value in a list

Dictionary of counters

: get a city c

if c in count:
 count[c] += 1

else:

 count[c] = 1

try:

 count[c] += 1

except KeyError:

 count[c] = 1

Input & output

In Python - via str

```
x = input()
```

↖
string

More about strings

```
raise ValueError(msg)
```

```
msg = "Negative input" +  
      str(v)
```

Typical situation

```
"  
  _____"  
           ↑  
       place to  
       insert value
```

String formatting

"Negative input: {0}" .format(v)



str(v)

Formatting — specify output "style"

Syntax is inherited from C

Formatting

`%d` - "digit" - i.e. integer

`%f` float

`%s` string

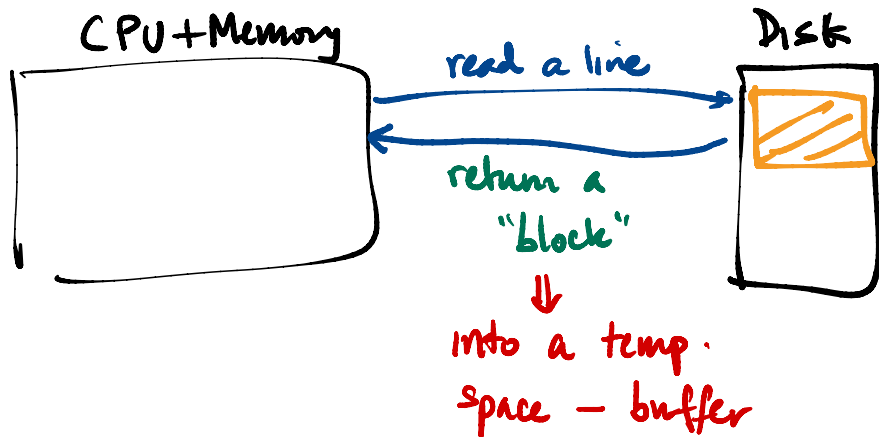
`%6d` - use six characters
↑
width
won't truncate

% 7.2 f

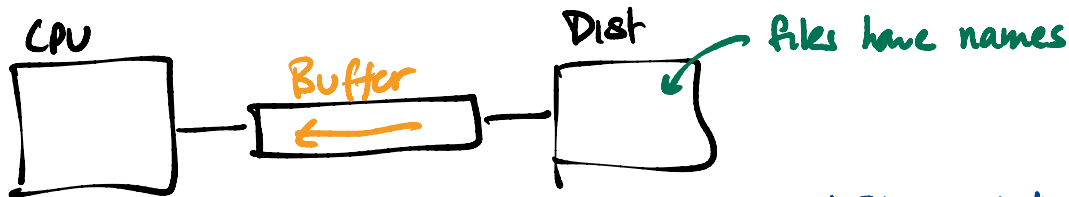
width after decimal pt

Files

Memory is much faster than disk



Reading



open a file — connects buffer — "File handle" — associates a name to buffer
read (from buffer) — from file handle
close a file — disconnects buffer

Writing

Open (attach file handle)

Write (to file handle) → "flush" the buffer

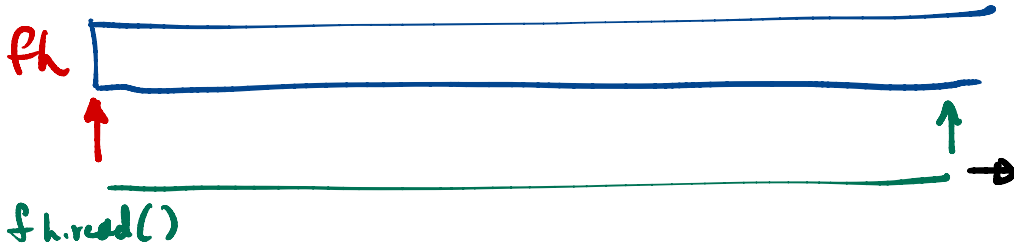
Close

↳ otherwise buffer may not flush

Overwrite vs append

Reading

```
fh = open(filename, "r")
```



seek()

