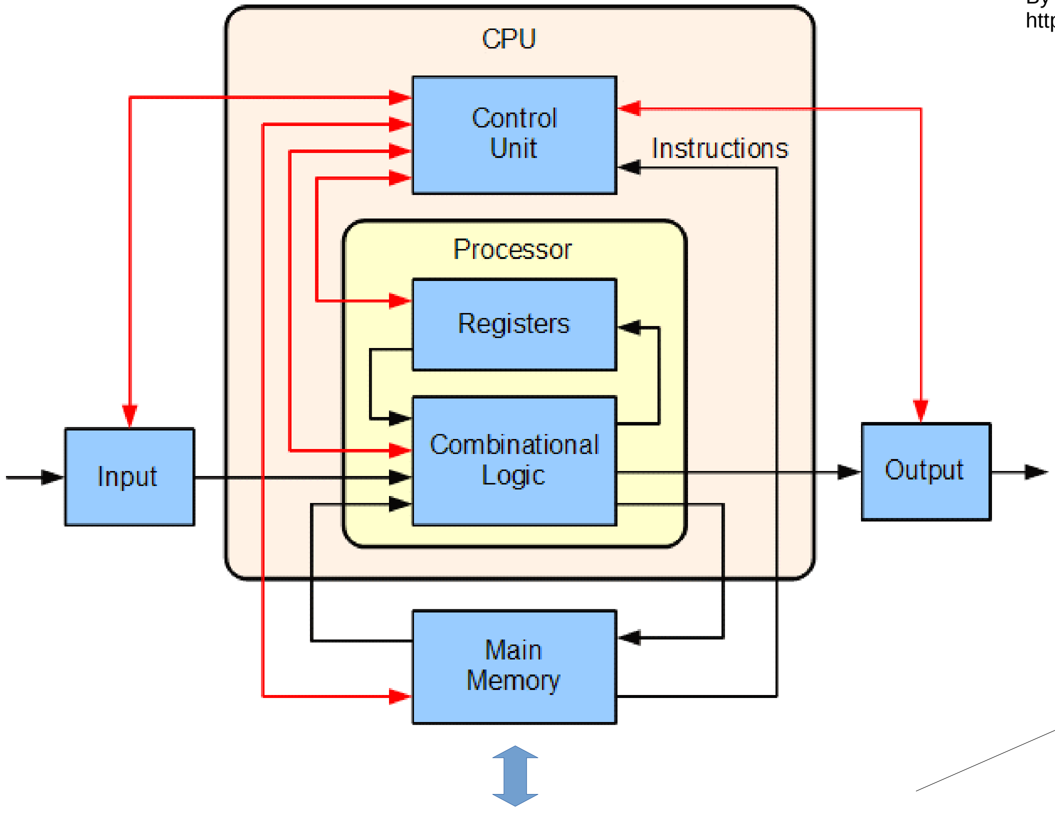
Input and output

IC152 Feb 2021

Parts of the computer

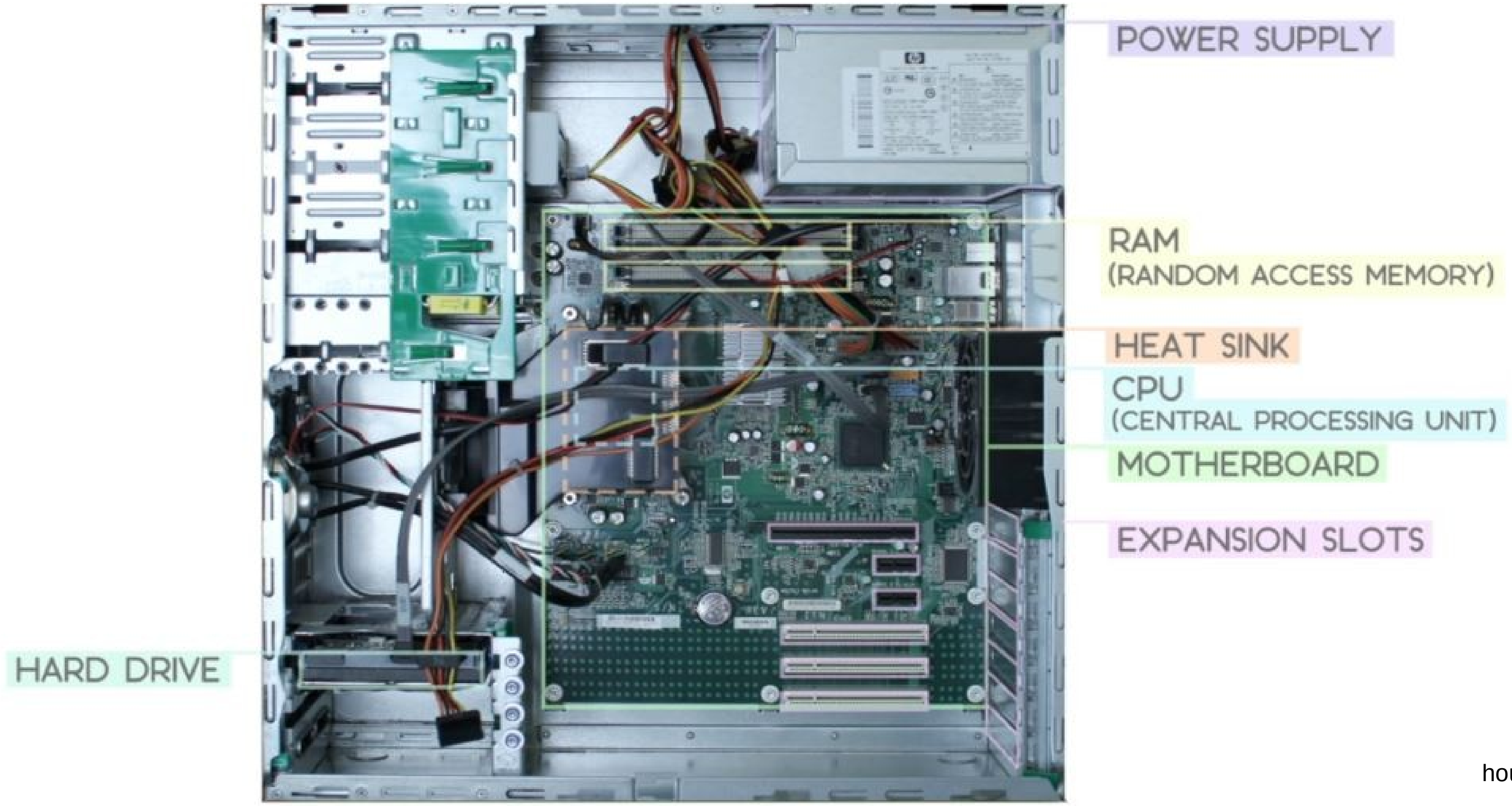


Secondary memory (non-volatile)

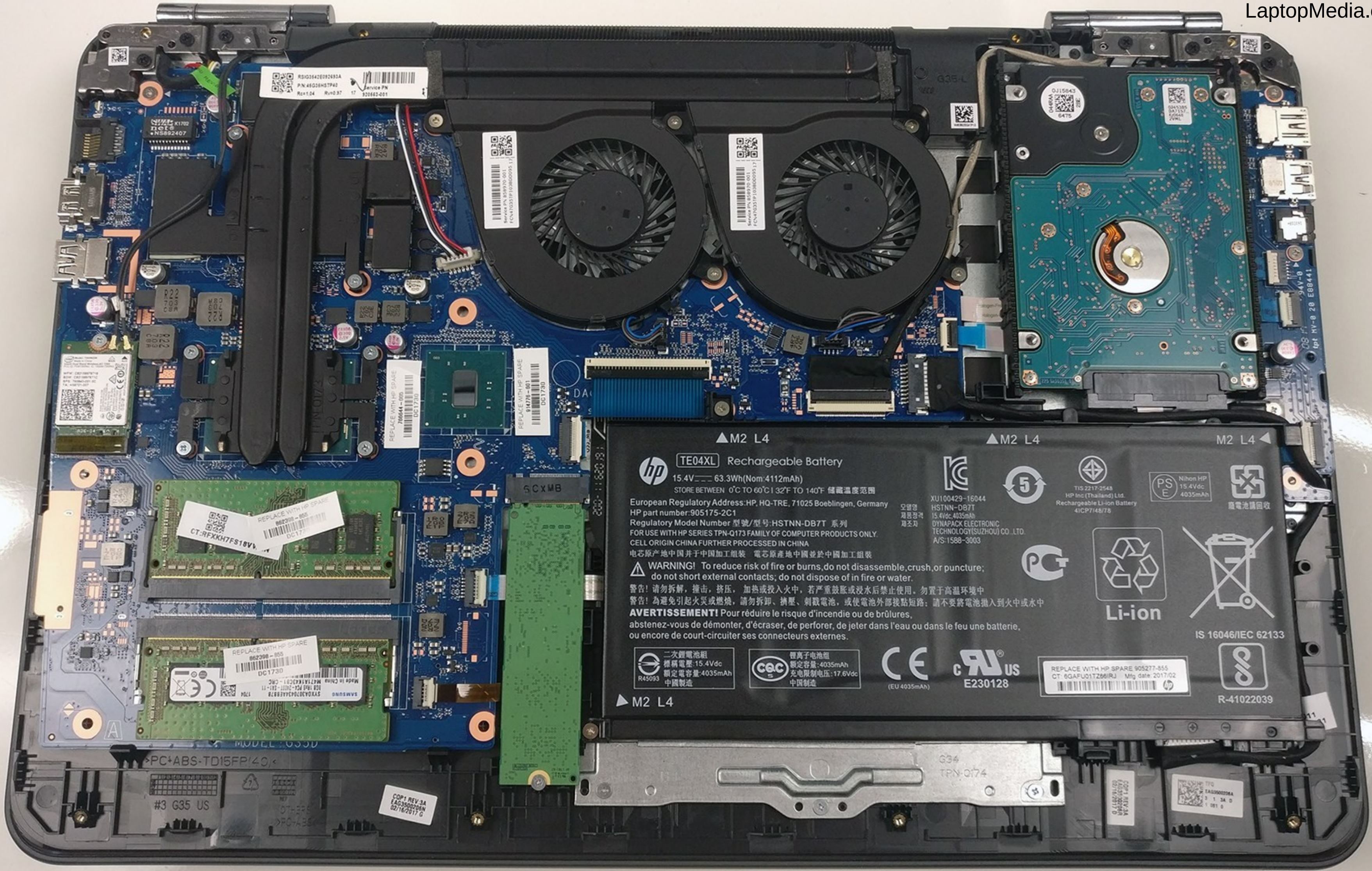
Usually, main memory is volatile

To retain content, we usually use non-volatile memory

This is usually hard disks, flash memory, optical disks, tape

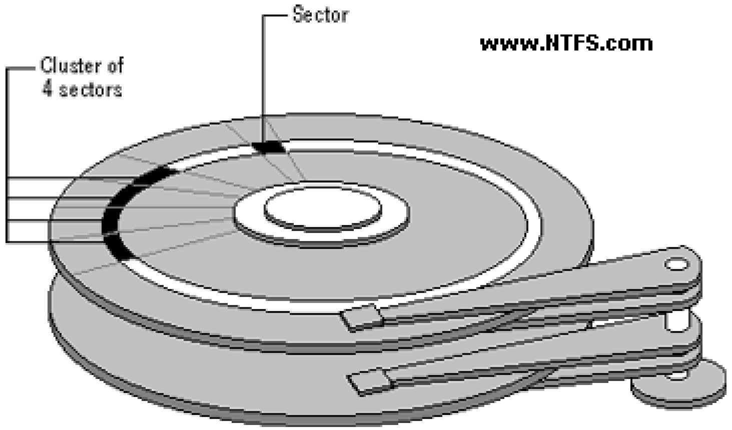


houkconsulting.com


TE04XL Rechargeable Battery
15.4V 63.3Wh(Nom:4112mAh)
HP part number:905175-2C1
Regulatory Model Number HSTNN-DB7T
Li-ion
M2 L4
REPLACE WITH HP SPARE
SAMSUNG

Inside a hard disk drive (HDD)

https://www.nextofwindows.com/



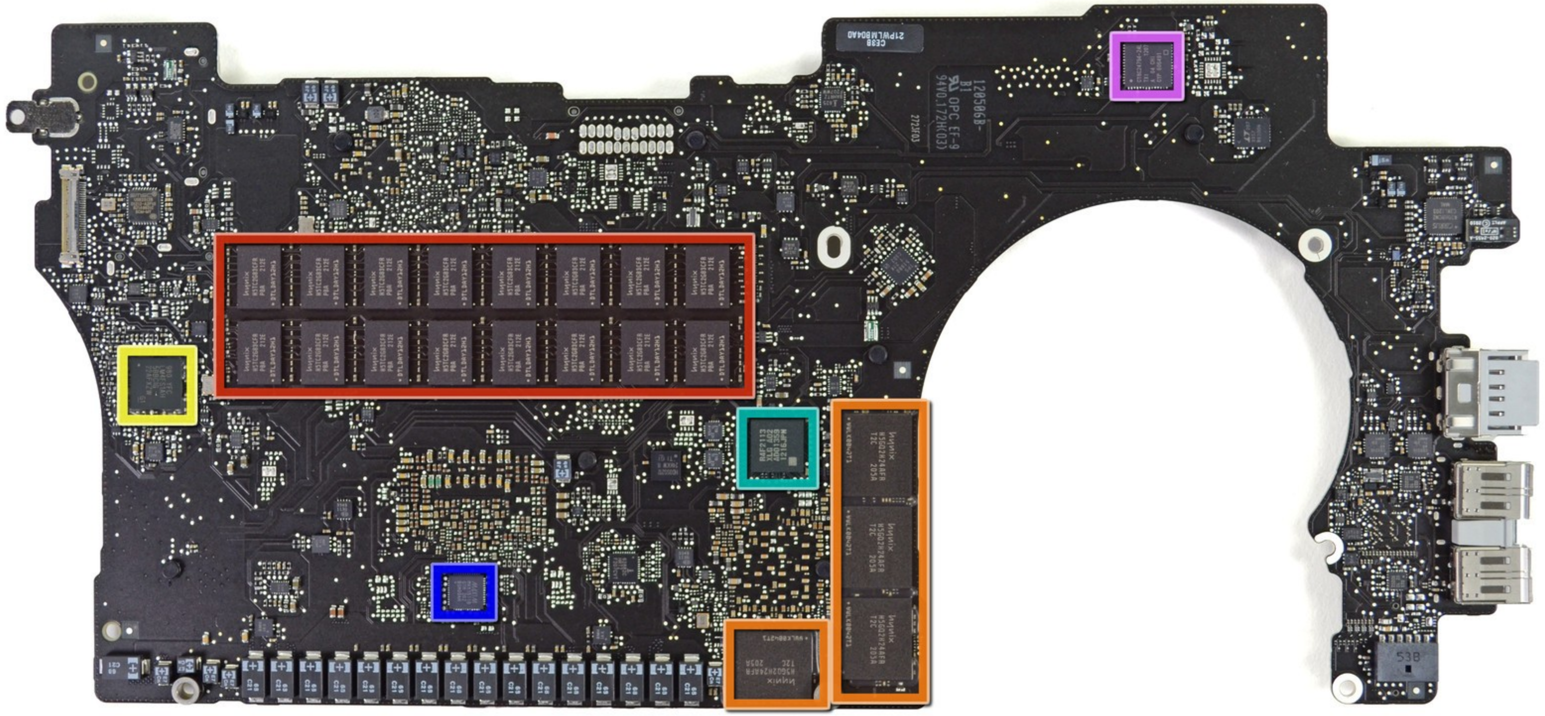
Made up of tracks and sectors

Watch video:
https://youtu.be/NtPc0jI21i0

https://commons.wikimedia.org/wiki/File:Samsung_SSD_840_120GB_MZ-7TD120--4_LID_REMOVED.JPG

Solid-state drives are more reliable than HDDs
SSDs are becoming mainstream in many computers

https://commons.wikimedia.org/wiki/File:Super_Talent_2.5in_SATA_SSD_SAM64GM25S.jpg ifixit.com



Sometimes SSDs are directly soldered onto the motherboard

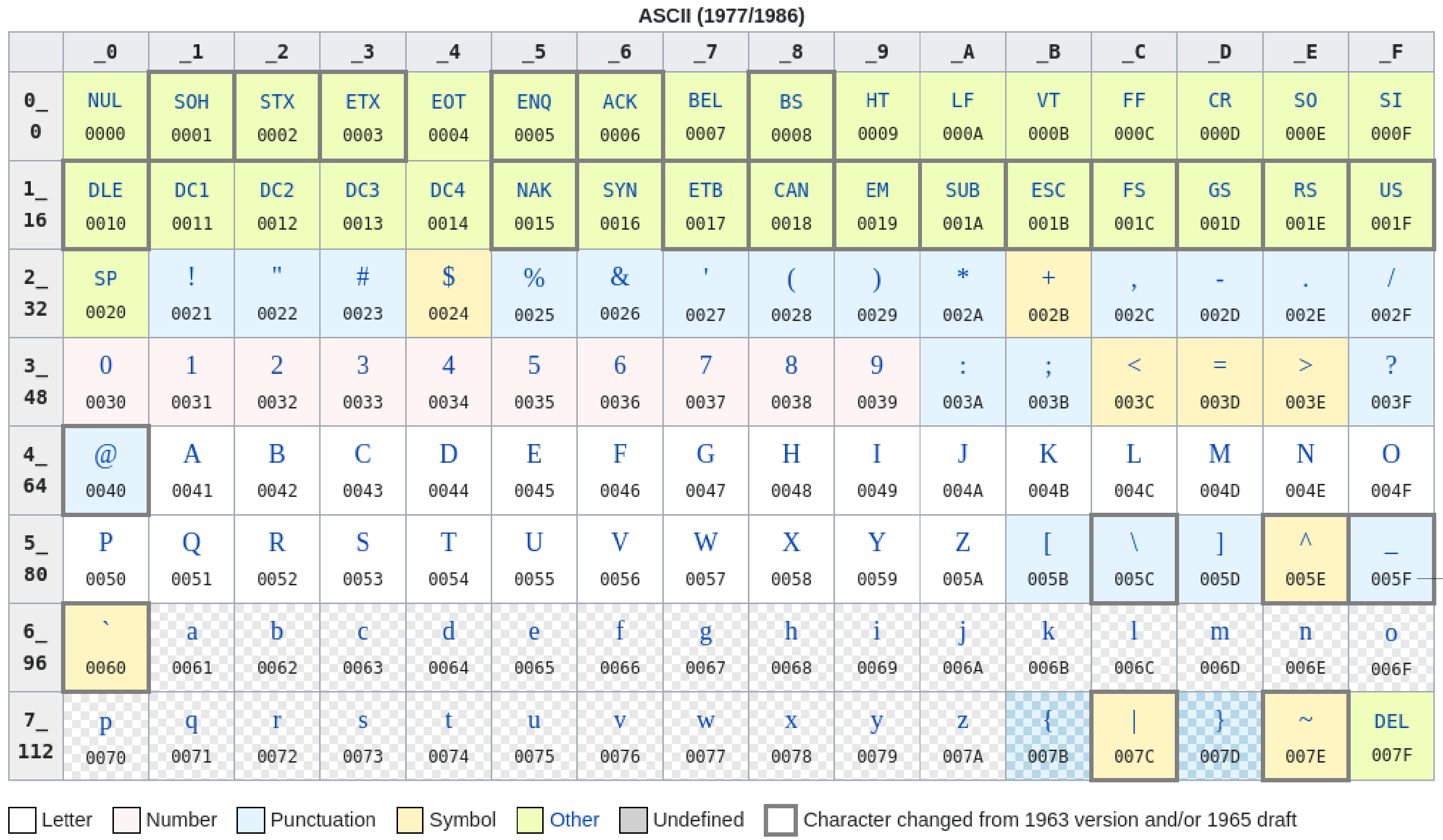
Why is I/O challenging?

Vast speed difference between CPU+memory and electromechanical I/O devices

DVD	Laser	1 or 2 sided disk	600-1600 rpm	1-10 GB	ms-sec
Hard disk	Magnetic	2 to 30 sided disk	3600-15000 rpm	1 GB- 10 TB	ms
Solid state drive	Flash memory	Electronic	No mechanical movement	1 GB- 100 GB	µs-ms
Tape	Magnetic	500-900m reel	10 meters/sec	1 GB-300 PB	secs-hours
Cloud storage	Behind the network				ms-hours

- The operating system hides these differences
- A file is provided as an abstraction for the physical device
- All I/O is performed with streams
  - Sequence of bytes in the OS memory

- Various types of files:
  - Binary file: named sequence of bytes
  - Text file (ASCII file):
    - Each byte is a character (see ASCII code)
    - Each line ends with ‘\n’
  - Formatted file
    - eg. tabular data (rows and columns)
    - .csv file: comma separated values
    - .ods, .xls: binary spreadsheet files
    - Semi-structured file: eg. a web page (HTML file)



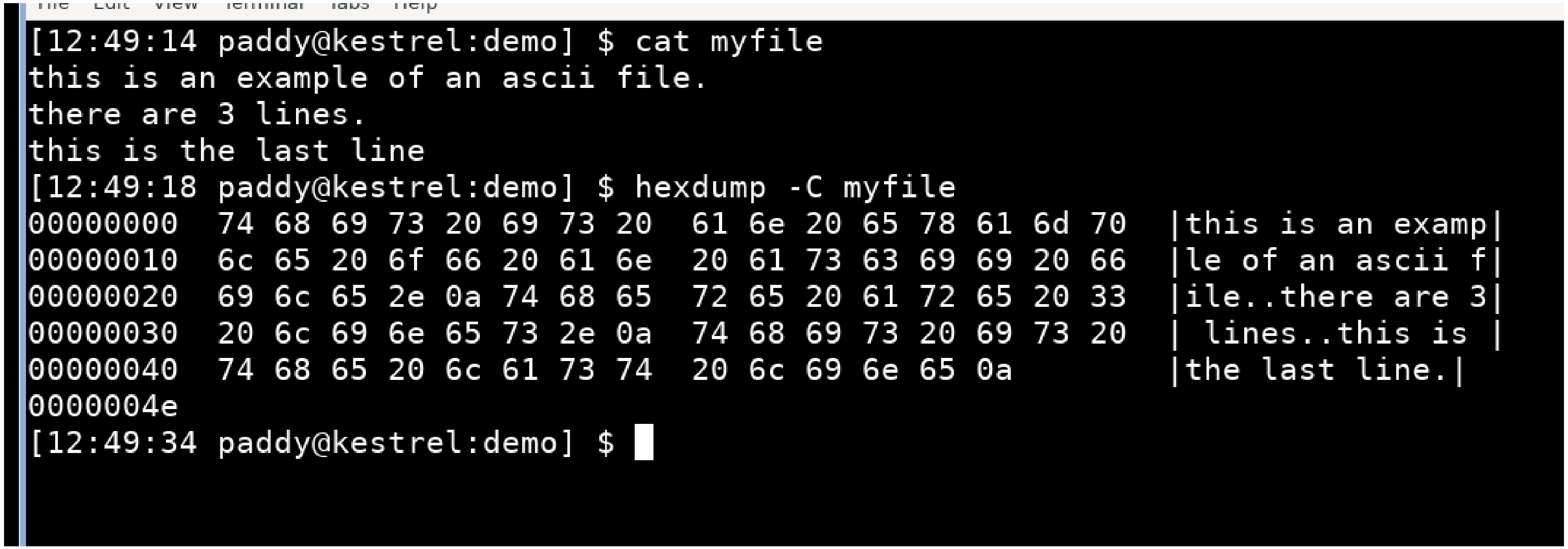
ASCII (1977/1986)

	_0	_1	_2	_3	_4	_5	_6	_7	_8	_9	_A	_B	_C	_D	_E	_F
0_ 0	NUL 0000	SOH 0001	STX 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
1_ 16	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	SUB 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
2_ 32	SP 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
3_ 48	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
4_ 64	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
5_ 80	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
6_ 96	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
7_ 112	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	DEL 007F

⬜ Letter 🟥 Number 🟦 Punctuation 🟨 Symbol 🟩 Other ⬛ Undefined ▢ Character changed from 1963 version and/or 1965 draft

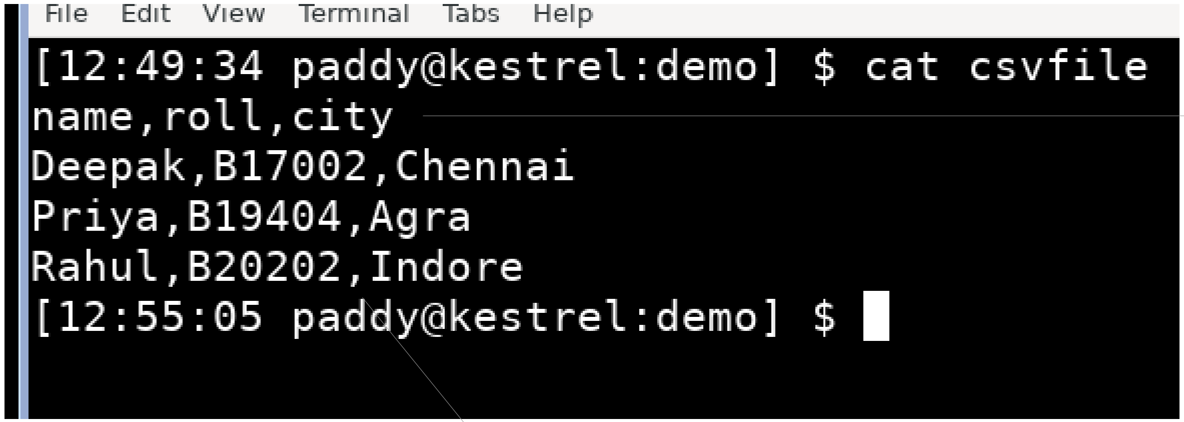
Hex code

https://en.wikipedia.org/wiki/ASCII



```
[12:49:14 paddy@kestrel:demo] $ cat myfile
this is an example of an ascii file.
there are 3 lines.
this is the last line
[12:49:18 paddy@kestrel:demo] $ hexdump -C myfile
00000000  74 68 69 73 20 69 73 20  61 6e 20 65 78 61 6d 70  |this is an examp|
00000010  6c 65 20 6f 66 20 61 6e  20 61 73 63 69 69 20 66  |le of an ascii f|
00000020  69 6c 65 2e 0a 74 68 65  72 65 20 61 72 65 20 33  |ile..there are 3|
00000030  20 6c 69 6e 65 73 2e 0a  74 68 69 73 20 69 73 20  | lines..this is |
00000040  74 68 65 20 6c 61 73 74  20 6c 69 6e 65 0a        |the last line.|
0000004e
[12:49:34 paddy@kestrel:demo] $
```

CSV files are text (ASCII) files

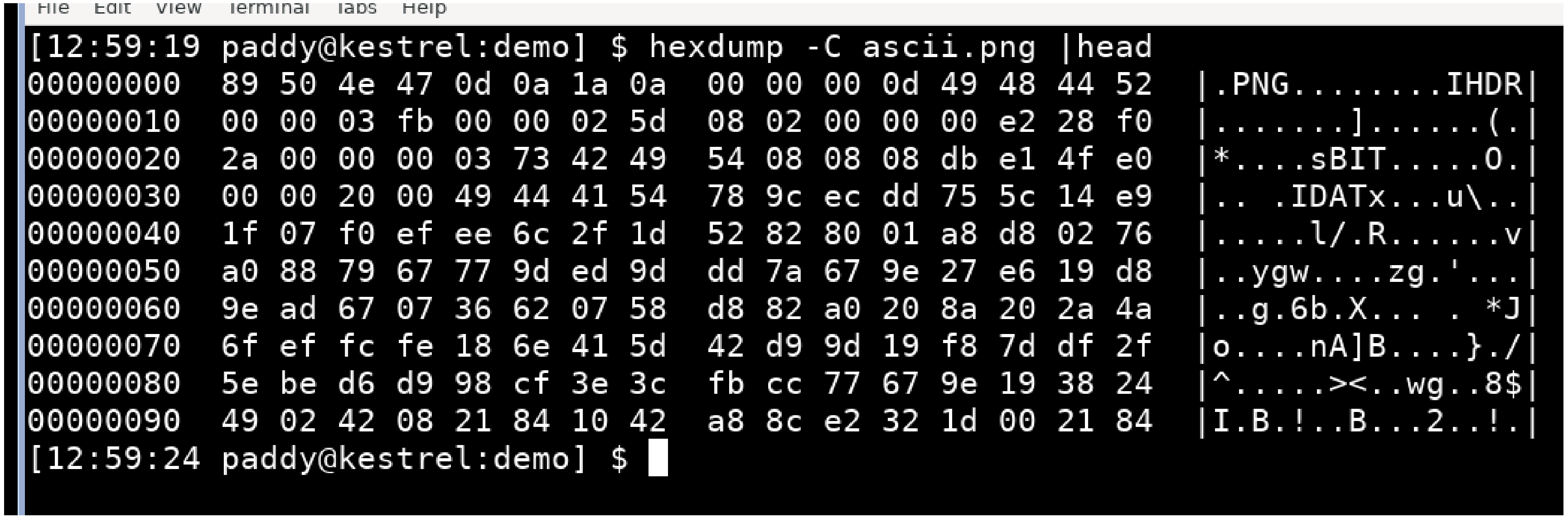


```
[12:49:34 paddy@kestrel:demo] $ cat csvfile
name,roll,city
Deepak,B17002,Chennai
Priya,B19404,Agra
Rahul,B20202,Indore
[12:55:05 paddy@kestrel:demo] $
```

First line (usually) column headings

Delimiter could be other characters

Binary files cannot be interpreted with ASCII codes



```
[12:59:19 paddy@kestrel:demo] $ hexdump -C ascii.png |head
00000000  89 50 4e 47 0d 0a 1a 0a  00 00 00 0d 49 48 44 52  |.PNG........IHDR|
00000010  00 00 03 fb 00 00 02 5d  08 02 00 00 00 e2 28 f0  |.......]......(.|
00000020  2a 00 00 00 03 73 42 49  54 08 08 08 db e1 4f e0  |*....sBIT.....O.|
00000030  00 00 20 00 49 44 41 54  78 9c ec dd 75 5c 14 e9  |.. .IDATx...u\..|
00000040  1f 07 f0 ef ee 6c 2f 1d  52 82 80 01 a8 d8 02 76  |.....l/.R......v|
00000050  a0 88 79 67 77 9d ed 9d  dd 7a 67 9e 27 e6 19 d8  |..ygw....zg.'...|
00000060  9e ad 67 07 36 62 07 58  d8 82 a0 20 8a 20 2a 4a  |..g.6b.X... . *J|
00000070  6f ef fc fe 18 6e 41 5d  42 d9 9d 19 f8 7d df 2f  |o....nA]B....}./|
00000080  5e be d6 d9 98 cf 3e 3c  fb cc 77 67 9e 19 38 24  |^.....><..wg..8$|
00000090  49 02 42 08 21 84 10 42  a8 8c e2 32 1d 00 21 84  |I.B.!..B...2..!.|
[12:59:24 paddy@kestrel:demo] $
```

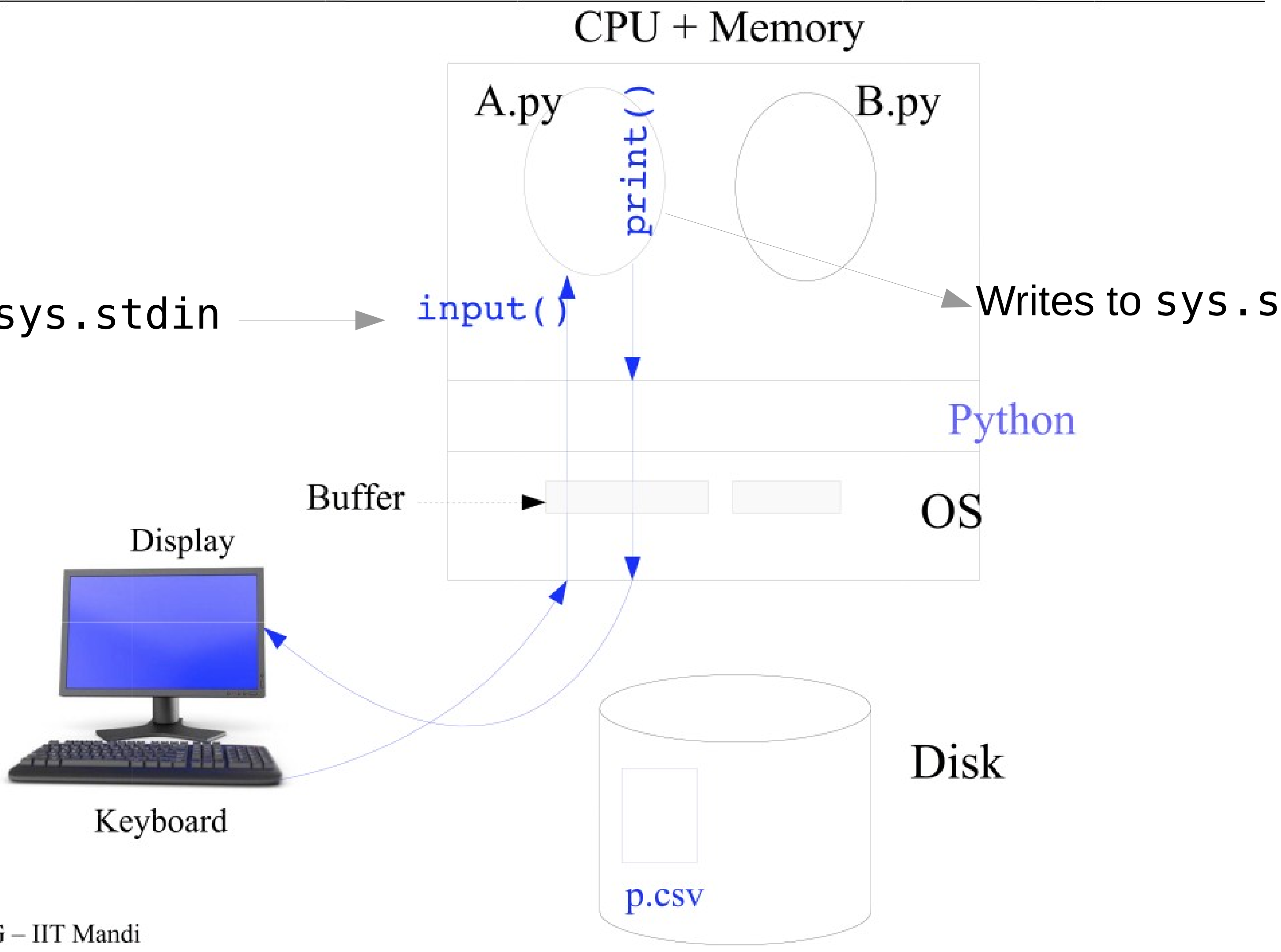
An application (eg. an image viewer) is required to make sense of the contents

Standard streams in Python

- Standard input stream, `sys.stdin`
  - Normally connected to the keyboard
- Standard output stream `sys.stdout`
  - Normally connected to the display
- Standard error stream, `sys.stderr`
  - Normally connected to the display

Reads from `sys.stdin`

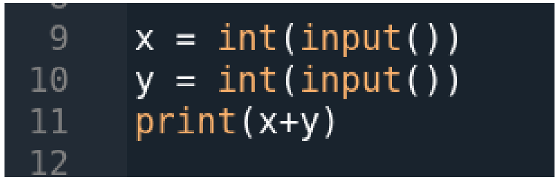
Writes to `sys.stdout`



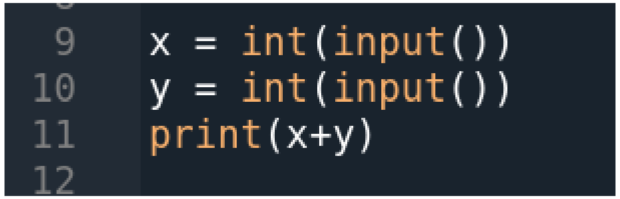
Slide from IC152 2018

I/O redirection can be used to connect `stdin` or `stdout` to a file, instead of keyboard or display

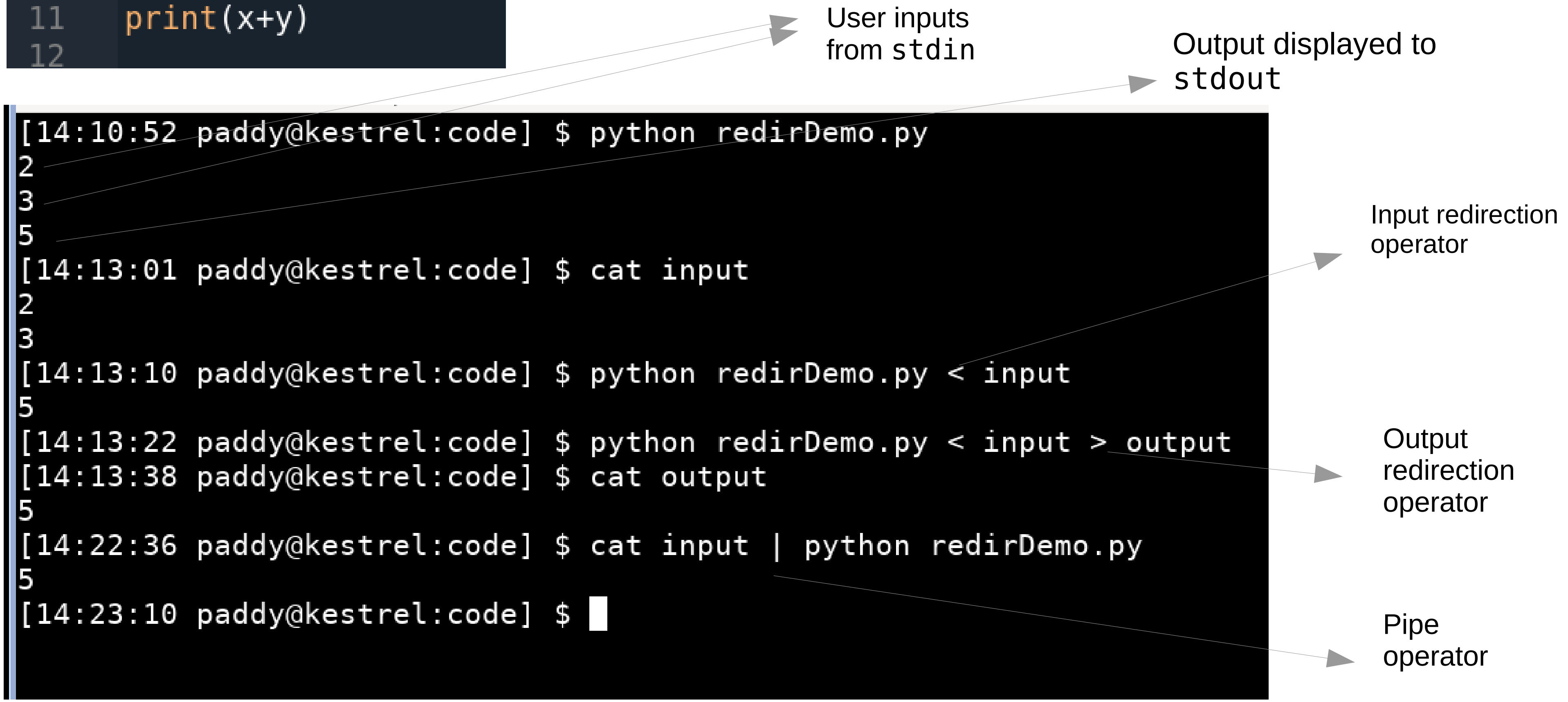
This makes an interactive program into a non-interactive program



```python
x = int(input())
y = int(input())
print(x+y)
```



```python
x = int(input())
y = int(input())
print(x+y)
```

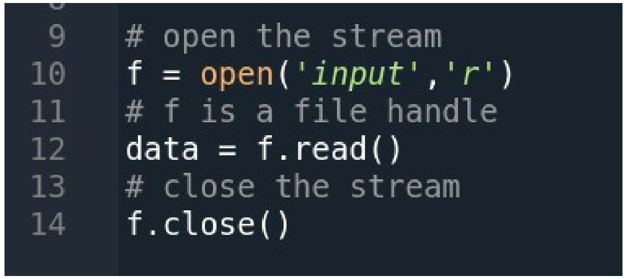


```
[14:10:52 paddy@kestrel:code] $ python redirDemo.py
2
3
5
[14:13:01 paddy@kestrel:code] $ cat input
2
3
[14:13:10 paddy@kestrel:code] $ python redirDemo.py < input
5
[14:13:22 paddy@kestrel:code] $ python redirDemo.py < input > output
[14:13:38 paddy@kestrel:code] $ cat output
5
[14:22:36 paddy@kestrel:code] $ cat input | python redirDemo.py
5
[14:23:10 paddy@kestrel:code] $ 
```

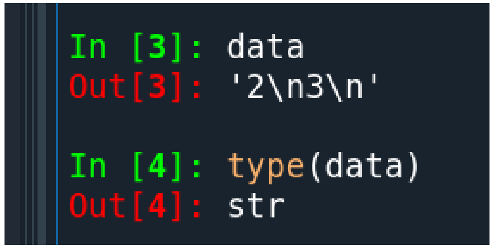
User inputs from `stdin`
Output displayed to `stdout`
Input redirection operator
Output redirection operator
Pipe operator

File access

- Basic steps:
  - Open the file
  - Read/write the file
  - Close the file

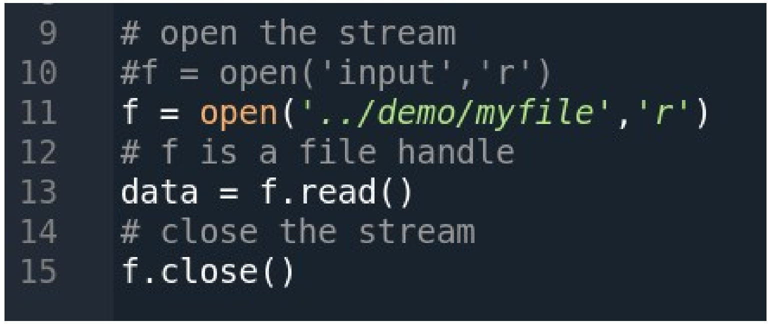


```python
# open the stream
f = open('input','r')
# f is a file handle
data = f.read()
# close the stream
f.close()
```

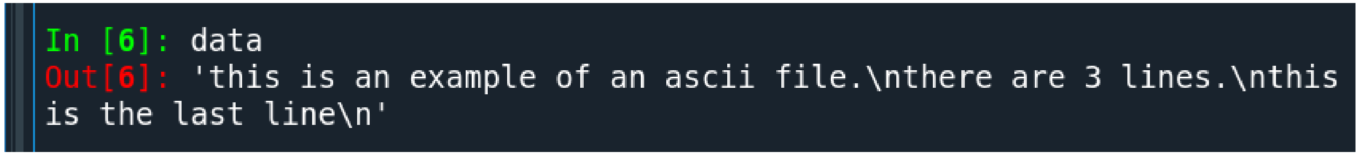


```
In [3]: data
Out[3]: '2\n3\n'

In [4]: type(data)
Out[4]: str
```

After running
the code



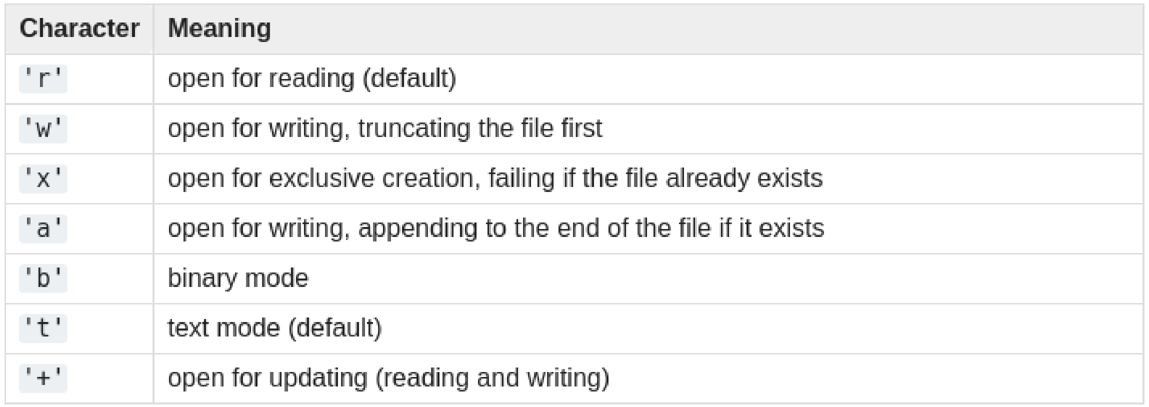
```python
# open the stream
#f = open('input','r')
f = open('../demo/myfile','r')
# f is a file handle
data = f.read()
# close the stream
f.close()
```



```
In [6]: data
Out[6]: 'this is an example of an ascii file.\nthere are 3 lines.\nthis
is the last line\n'
```

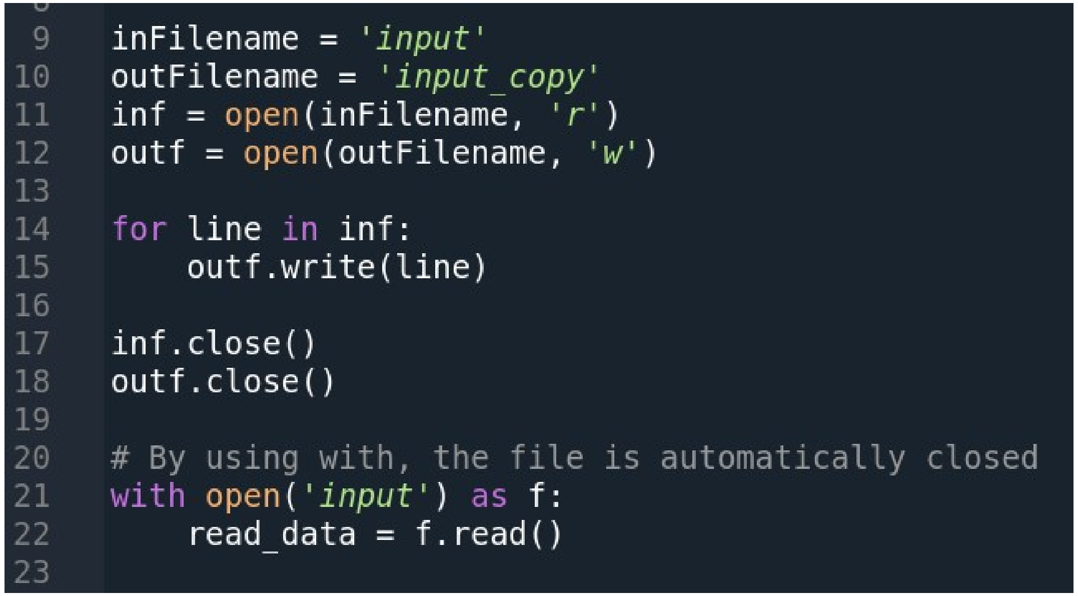
```
fp = open(filename,mode)
```

What will be the file used for?

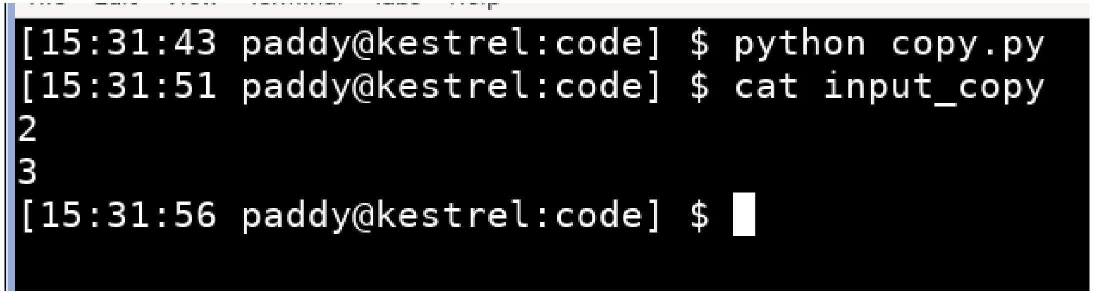


Character	Meaning
`'r'`	open for reading (default)
`'w'`	open for writing, truncating the file first
`'x'`	open for exclusive creation, failing if the file already exists
`'a'`	open for writing, appending to the end of the file if it exists
`'b'`	binary mode
`'t'`	text mode (default)
`'+'`	open for updating (reading and writing)

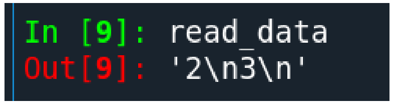
https://docs.python.org/3/library/functions.html#open



```
inFilename = 'input'
outFilename = 'input_copy'
inf = open(inFilename, 'r')
outf = open(outFilename, 'w')

for line in inf:
    outf.write(line)

inf.close()
outf.close()

# By using with, the file is automatically closed
with open('input') as f:
    read_data = f.read()

```



```
[15:31:43 paddy@kestrel:code] $ python copy.py
[15:31:51 paddy@kestrel:code] $ cat input_copy
2
3
[15:31:56 paddy@kestrel:code] $ 
```



```
In [9]: read_data
Out[9]: '2\n3\n'
```

When done with a file, it must be closed using f.close()

Else you may lose data!

Buffering of output may happen without data actually being written to disk

An exception or other issue (eg power failure) can happen

Other methods in File object

- `f.read()`: reads the entire file
- `f.readline()`: reads one line
- `f.readlines()`: reads all lines
- `f.write(str)`: writes string str to file