

* File Random Access — Rough Work ① .

two built-in functions : seek(), tell().

- seek() — sets/changes the position of file handle/file pointer to the given specific — position in file where read/write/append-data operation will be performed.

[* file handle/pointer is like cursor which indicates the position — where data will be read/write]

* seek() sets the file pointer to the offset (argument)
~~offset is the file pointer position~~
offset is the position of file pointer — where read/write will be done

* seek() also uses another optional ~~position~~ argument which is optional position/reference position it is called from-what . argument

| <u>* from what</u> | <u>file position (in file)</u> |
|--------------------|--|
| 0 : | sets the reference point — at beginning (by default) |
| 1 : | sets the reference point — at current position |
| 2 : | set the reference point — at end of file |

* ~~one~~

* seek() function is used in two ways →

- way 1: Absolute positioning
- way 2: Relative Positioning

Rough Work

(2)

• way 1 → Absolute positioning → $f.seek(\text{absolute_location})$

(i) $f.seek(20)$

↓
file
object

↓
file
object

places file pointer at 20th byte in file (no matter where you are)

(ii) $f.seek(0)$

↳ at beginning (0th byte) in file

(iii) $f.seek(50)$

↳ at 50th byte in file

• way 2 → Relative Positioning →

⊕ $f.seek(\text{absolute_location, from_what})$

(i) $f.seek(10, 0)$ → ^{move} 10 bytes forward ~~that~~ from beginning position — in file

(ii) $f.seek(0, 0)$ → ^{move} at beginning position — in file

(iii) $f.seek(-10, 1)$ → ~~to~~ move 10 bytes back from current position

(iv) $f.seek(10, 1)$ → move 10 bytes forward ~~that~~ from current position

(v) $f.seek(0, 2)$ → move to the end of ~~ff~~ position — in file

(vi) $f.seek(-10, 2)$ → move to 10 bytes back from ~~an~~ end ~~of~~ position

(vii) $f.seek(0, 1)$ → ^{move to — in file} ~~stop~~ current position — in file
(* seek() — for read/write data)
(* seek() — no return value)

Rough Work

(3) only

• tell() — returns the current position of file object/
pointer

— it tells the ~~at~~ current position of file object/
pointer in file,

* file opened → read/write mode → file object is at
start byte / at beginning in file

* file opened → append mode → file object is at
last position in file
last byte of file

syntax → $f.tell()$
 ↓
 file object
 no argument

① $f = \text{open}(\text{"my document.txt"}, \text{"rt"})$
 $\text{print}(f.tell())$

② $f = \text{open}(\text{"my document.txt"}, \text{"a"})$
 $\text{print}(f.tell())$

③ $f = \text{open}(\text{"my document.txt"}, \text{"w"})$
 $\text{print}(f.tell())$

④ $f = \text{open}(\text{"my document.txt"}, \text{"r+"})$
 $\text{print}(f.readline())$
 $\text{print}(f.tell())$

(v) ex:- seek() + tell() →

```
f = open("mydocument.txt", "rt")
```

```
f.seek(20)
```

```
print(f.tell())
```

```
print(f.readline())
```

```
print(f.tell())
```

```
print(f.readline())
```

```
print(f.tell())
```

```
print(f.readlines())
```

```
print(f.tell())
```