seek() method

In Python, seek() function is used to **change the position of the File Handle** to a given specific position. File handle is like a cursor, which defines from where the data has to be read or written in the file.

Syntax: f.seek(offset, from_what), where f is file pointer
Parameters:
Offset: Number of positions to move forward
from_what: It defines point of reference.
Returns: Return the new absolute position.

The reference point is selected by the **from_what** argument. It accepts three values:

- **0**: sets the reference point at the beginning of the file

- **1**: sets the reference point at the current file position

- **2**: sets the reference point at the end of the file

By default from_what argument is set to 0.

Example 1: Let's suppose we have to read a file named "GfG.txt" which contains the following text:

```
"Code is like humor. When you have to explain it, it's bad."
```

Python3

```python
# Python program to demonstrate
# seek() method

# Opening 'GfG.txt' text file
f = open('GfG.txt', 'r')

# Second parameter is by default 0
# sets Reference point to twentieth
# index position from the beginning
f.seek(20)

# prints current position
print(f.tell())

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

Output:

```
20
When you have to explain it, it's bad.
```

Example 2: Seek() function with negative offset only works when file is opened in binary mode. Let's suppose the binary file contains the following text

```
b'Code is like humor. When you have to explain it, its bad.'
```

Python3

```python
# Python code to demonstrate
# use of seek() function


# Opening 'GfG.txt' text file
# in binary mode
f = open('data.txt', "rb")

# sets Reference point to tenth
# position to the left from end
f.seek(-10, 2)

# prints current position
print(f.tell())

# Converting binary to string and
# printing
print(f.readline().decode('utf-8'))

f.close()
```

Output:

```
47

, its bad.
```