File operation			
Sr.No.	**Syntax/command**	**Output**	**Remarks**
1	l = [1,2,3,4,5] b = iter(l) next(b)	1	If you pass next() command directly on l it will give error saying list object is not a iterator. Using iter() command you can convert the list and perform next() command.
	next(b)	2	
	next(b)	3	
	next(b)	4	
	next(b)	5	
	next(b)	Error	It will give an error saying list object is not iterable.
2	def square_func(i): x = [j*j for j in range(i)] print(x)		when you enter commands within square brackets to directly convert data to a list - then it is called **"List comprehension"**
3	def square_func1(i): x = [j*j for j in range(i)] return x	[0, 1, 4, 9]	Return command will give the output of x directly.
	square_func1(4)		
4	def square_func2(i):		Whereas if you replace it with yield inplace of return it will not give the output directly.
	x = [j*j for j in range(i)]		
	square_func2(4)	Error	
5	next(square_func2(4))	[0, 1, 4, 9]	**Yield command** will make the entire data available as a iterator.

6	def fib_series1(n):
	a = 1
	b = 1
	l = []
	for i in range(n):
	l.append(a)
	a,b = b,a+b
	return l
	fib_series1(7)
		[1, 1, 2, 3, 5, 8, 13]	Function to get fibonacci numbers

7	**from functools import reduce**	36	Reduce function uses successive a,b values to calculate the sum() of the iterable. Apply a function of two arguments cumulatively to the items of a sequence, from left to right, so as to reduce the sequence to a single value. For example, reduce(lambda x, y: x+y, [1, 2, 3, 4, 5]) calculates ((((1+2)+3)+4)+5)
	reduce(lambda a , b : a+b , l)		
8	list(filter(lambda x : x%2 ==0, l))	[2, 4, 6, 8]	Filter function will extract the data from iterable w.r.t function mentioned.

9	def even_func(n):
	if n%2 == 0:
	return True
		[2, 4, 6, 8]	Alternate to above code. Filter function works on the condition of True/False

	list(filter(even_func,l))		

10	import urllib.request # url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Module%204/data/example1.txt' # filename = 'Example1.txt' # urllib.request.urlretrieve(url, filename)		Download data

	from pyodide.http import pyfetch
	import pandas as pd
	
	filename = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Module%204/data/example1.txt"
	
	async def download(url, filename):
	  response = await pyfetch(url)
	  if response.status == 200:
	  with open(filename, "wb") as f:
	  f.write(await response.bytes())
	
	await download(filename, "Example1.txt")		

11	with open(example1, "r") as file1:
	  FileContent = file1.read()
	  print(FileContent)
			A Better Way to Open a File is using the with statement. The code will run everything in the indent block then close the file object.

12	with open(example1, "r") as file1:
	  print(file1.read(4))
	  print(file1.read(4))
	  print(file1.read(7))
	  print(file1.read(15))
			Once the method .read(4) is called the first 4 characters are printed. If we call the method again, the next 4 characters are printed. And so on.

13	with open(example1, "r") as file1:
	  print(file1.readline(20)) # does not read past the end of line
	  print(file1.readline(20)) # Returns the 20 chars of next line
			We can also pass an argument to readline() to specify the number of charecters we want to read in each line

14	with open(example1,"r") as file1:
	  i = 0;
	  for line in file1:
	  print("Iteration", str(i), ": ", line)
	  i = i + 1
			Iterate through the lines

	Lines = ["This is line A\n", "This is line B\n", "This is line C\n"]
	with open('Example2.txt', 'w') as writefile:
	  for line in Lines:
	  print(line)
	  writefile.write(line)
			With \n it will write each sent on a new line

	with open('Example2.txt','r') as readfile:
	  with open('Example3.txt','w') as writefile:
	  for line in readfile:
	  writefile.write(line)
			It is used to copy data from readfile to write file