In []:	Len() Builtin Function Length function(len())> will give you a length of any sequence
III [].	Example of Len() Functions with Different Types of Sequence
<pre>In [4]: Out[4]:</pre>	#In case of List x=[10,20,30,40,50,60] len(x)
In [5]:	<pre>#In case of String x="Pratyush Srivastava" len(x)</pre>
Out[5]: In [6]:	#In case of Tuple x=(10,20,30,40,50) len(x)
Out[6]: In [7]:	#In case of Set x={10,20,30,40,50,10,20,30,40} len(x)
Out[7]: In [8]:	#In case of Dictionary x={"name":"Pratyush", "Class":"M.Tech", "Location":"Lucknow"} len(x)
Out[8]:	Range() Builtin Function
In []:	<pre>range()>it is used to generate a sequence of numbers. Range function is immutable Range function will always take three argument :> If you want to use range function then you need to use atleast one argument or atmost three argument.</pre>
	First Argument>Start Value> By Default take 0 Second Argument> end Value> Mandatory to give ending value if you are not giving ending value then automatic whatever value you are giving inside the range function will be trated as ending value.
	Third Argument> Gap(Step)> default> 1 Example: generate numbers from 0 to 9> 0 1 2 3 4 5 6 7 8 Syntax of Range Function: range(start value , end value , gap)
	Variations of Range Functions
In [9]: Out[9]:	#With Two Arguments x=list(range(10,20)) x [10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
<pre>In [10]: Out[10]:</pre>	#With one Argument x=list(range(10)) x [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
<pre>In [11]: Out[11]:</pre>	#With Three Arguments x=list(range(1,20,3)) x [1, 4, 7, 10, 13, 16, 19]
In [12]:	More Example of Range Function x=range(12, 36, 5) #12 17 for i in x:
	<pre>print(i) 12 17 22 27</pre>
In [13]:	<pre>start_value=int(input("Enter starting value")) ending_value=int(input("Enter ending_value")) step=int(input("Enter step value")) list = tuple(range(start_value, ending_value, step))</pre>
	<pre>print(list) Enter starting value10 Enter ending_value20 Enter step value2 (10, 12, 14, 16, 18)</pre>
In []:	Types of Statements in Python Conditional Statement> if elif and else Iterative Statement> for and while
	Iterative Statement> break, continue Transfer Statement> break, continue Iterative Statements(loops)
In []:	Iterative Statement> when we want to execute a group of statement multiple times then we will use iterative statements. In python we are having two types of iterative statement: 1.for loop 2.while loop
	Requirement of Iterative Statement: Suppose i want to write a code that will print hello world 10000 Times. For that we have two approaches: 1.We can manually write Hello World 10000 Times. 2.We can simply use iterative statement and then print hello world 10000 times.
	Note: First approach is not feasible because we need to write hello world 10000 times which decreases the code readability and also increases the code length. To fullfill such requierement in more easier manner we need to Iterative Statement(loops).
In [14]:	<pre>print("Hello World") print("Hello World") print("Hello World") print("Hello World") print("Hello World") print("Hello World") print("Hello World")</pre>
	<pre>print("Hello World") print("Hello World") print("Hello World") print("Hello World") print("Hello World") print("Hello World")</pre>
	<pre>print("Hello World") print("Hello World") Note: With the help above we need to write hello world 20 Times by writing 20 lines of statements. and It is not feasible as it decreases the readibility of the code and also increases the length of the code.</pre>
	Hello World Hello World Hello World Hello World Hello World Hello World
	Hello World
In [17]:	Hello World Hello World for i in range(10): print("Hello World")
	Note: With the help for loop we can easily print hello world 10 Times by writing 2 lines of statements. and It is more feasible as it increases the readibility of the code Hello World Hello World
	Hello World
	For Loops
In []:	> if we want to execute a number of statement in some sequence then we should always use for loop. Syntax of For loop: for i in sequence: body/Statement
	i is known as iterator Examples of For Loop
In [19]:	Python Program to print each and every Character of the Given String x="Edyoda Digital University" for i in x:
	<pre>print(i) E d y 0</pre>
	a D i g i
	t a l l l l l l l l l l l l l l l l l l
	V e r s i
In [20]:	Python Program to print each and every Element of the Given List x=[10, 20, 30, 40, 50]
111 [20].	<pre>for i in x: print(i) 10 20 30</pre>
	Python Program to print each and every Element of the Given String
In [21]:	<pre>x=(10,20,30,40,50) for i in x: print(i) 10 20 30</pre>
	Python Program to Demostrate for loop with range Function
In [22]:	<pre>for i in range(10): #0 1 2 3 4 5 6 7 8 9 print(i) 0 1 2 3</pre>
	3 4 5 6 7 8
In []:	Python Program to find the Factorial of a Number #factorial of a number. 5!>5*4*3*2*1*0 =120 61. > 6*5*4*3*2*1=720
In [23]:	<pre>6!> 6*5*4*3*2*1=720 #Solution fact=1 for i in range(1,6): fact=fact*i</pre>
	Python Program to print Even numbers within a given range
In [24]:	<pre>#Approach-1 for i in range(10,15): #10,11,12,13,14,15,16,17,18,19,20 if i%2==0: print(i)</pre>
In [25]:	12 14 #Approach -2 for i in range(10,15,2): #10,11,12,13,14,15,16,17,18,19,20 print(i)
	10 12 14 Nested Loops
In []:	Nested Loops> Nested loop means a loop statement inside another loop statement. Note> Once all the iteration of the inner loop will be completed then pointer will come to the outer loop for outer loop iteration.
	Syntax: for i in sequence: for j in sequence:
	body/sequence
In [27]:	<pre>body/sequence Example of Nested Loops for i in range(3): for j in range(3): print("I value is "+str(i)+str(" j Value is ")+str(j))</pre>
In [27]:	<pre>body/sequence Example of Nested Loops for i in range(3): for j in range(3): print("I value is "+str(i)+str(" j Value is ")+str(j)) #Nested loop are basically use in case of patterns """ i j 0 0 0 1 0 2</pre>
In [27]:	<pre>body/sequence Example of Nested Loops for i in range(3): for j in range(3): print("I value is "+str(i)+str(" j Value is ")+str(j)) #Nested loop are basically use in case of patterns """ i j 0 0 0 1</pre>
In [27]:	Example of Nested Loops for i in range(3): for j in range(3): print("I value is "+str(i)+str(" j Value is ")+str(j)) #Nested loop are basically use in case of patterns i j 0 0 0 1 0 2 1 0 1 1 2 2 2 0 2 1 2 2 """ I value is 0 j Value is 0 I value is 0 j Value is 1 I value is 0 j Value is 2 I value is 0 j Value is 2 I value is 1 j Value is 0 I value is 1 j Value is 0 I value is 1 j Value is 1 I value is 1 j Value is 1
<pre>In [27]:</pre> Out[27]:	Example of Nested Loops for i in range(3): for j in range(3): print("I value is "+str(i)+str(" j Value is ")+str(j)) #Nested loop are basically use in case of patterns """ I j 0 0 0 0 1 0 0 2 1 0 0 0 0 0 1 0 0 0 0 0
	Example of Nested Loops for i in range(3): for j in range(3): print("I value is "+str(i)+str(" j Value is ")+str(j)) ### ### ### ### ### ### ### ### ###
Out[27]:	Example of Nested Loops for i in range(3): for j in range(3): print("I value is "*str(i)*str(" J Value is ")*str(j))
Out[27]:	Example of Nested Loops for i in range(3): for j in range(3): for j in range(3): for j in range(3): for j in range(3): for j in range(3): wested loop are basically use in case of patterns i j
Out[27]: In []:	Example of Nested Loops for i in range(3): for j in range(5): for j in range(5):
Out[27]:	Example of Nested Loops for 1 in range(3): for 2 in range(3): for 3 in range(3): for 3 in range(3): for 3 in range(5): for 1 in range(5):