Python Programming Fundamentals Cheat Sheet

Python Programming Fundamentals Cheat Sheet					
Package/Method	Description	Syntax and Code Example			
		Syntax:			
	Returns "thue" if both statement1 and statement2 are "True". Otherwise, returns "False".	1 statement1 and statement2 Q			
		Example:			
		1 marks = 90 2 attendance_percentage = 87			
AND		3 4 if marks >= 80 and attendance_percentage >= 85:			
		5 print("qualify for honors")			
		6 else: 7 print("Not qualified for honors")			
		Syntax: 1 class ClassName: # Class attributes and methods 2			
Class Definition	Defines a blueprint for creating objects and defining their attributes and behaviors.	Example:			
		1 class Person: 2 definit(self, name, age):			
		3 self.name = name 4 self.age = age			
	A function is a revisable block of code that performs a specific task or set of tasks when called.	Syntax: 1 def function_name(parameters): # Function body &			
Define Function					
Define Function		Example:			
		1 def greet(name): print("Hello,", name)			
	Checks if two values are equal.	Syntax:			
		1 variable1 variable2			
		Example 1:			
		1 5 == 5			
Equal(==)		returns True			
		Example 2:			
		1 age = 25 age == 30 Q			
		retums False			
		Syntax: 1 for variable in sequence: # Code to repeat @			
	A 'for' loop repeatedly associates a block of code for a specified number of flerations or over a sequence of elements (list, range, stiring, etc.).	Example 1:			
For Loop		1 for num in range(1, 10): 2 print(num)			
Tor Luop		Example 2:			
		1 fruits = ["apple", "banana", "orange", "grape", "kiwi"]			
		2 for fruit in fruits:			
		3 print(fruit)			
		Syntax:			
		1 function_name(arguments)			
Function Call	A function call is the act of executing the code within the function using the provided arguments.	Example:			
		1 greet("Alice")			
		Syntax:			
		1 variable1 >= variable2 @			
		Example 1:			
		1 5 >= 5 and 9 >= 5			
	Checks If the value of variable I is greater than or equal to variable 2.	returns True			
Greater Than or Equal To(>=)		Example 2:			
		1 quantity = 105			
		2 minimum = 190 3 quantity >= minimum			
		returns True			
		Syntax: 1 variable1 > variable2			
		Example 1: 9 > 6			
	Checks If the value of variable1 is greater than variable2.	Example 1: 9 > 6 returns True			
Greater Than(>)		returns true Example 2:			
oromor many					
		1 age = 28 2 max_age = 25			
		3 age > max_age			
		retums Faise			
		Syntax:			
		1 if condition: #code block for if statement			
If Statement	Executes code block 'if' the condition is "True'.	Example:			
		1 if temperature > 30:			
		1 if temperature > 30: 2 print("It's a hot day!")			
		2 print("It's a hot day!")			
		2 print("It's a hot day!")			
		2 print('It's a hot day(') Symtax 1 if condition1: 2 # Code if condition1 is True 3			
		2 print('It's a hot day!') Symtax 1 if condition1: 2 # Code if condition1 is True 3 4 ellif condition2: 5 # Code if condition2 is True			
		2 print('It's a hot day!') Symtax 1 if condition: 2 a Code if condition1 is True 3 4 ellif condition2: 5 a Code if condition2 is True 6 7 else:			
		2 print('It's a hot day(') Symbox 1 if condition1: 2 & Code if condition1 is True 3 4 clif condition2: 5 & Code if condition2 is True 6			
In-Ein-Eine	Executes the first code block if condition I is "True", otherwise checks condition2, and so on. If no condition is "True", the else block is executed.	2 print('It's a hot day!') Syntax 1 if condition1: 2 # Code if condition1 is True 3 clif condition2: 5 # Code if condition2 is True 6 7 clse:			
II-EII-EIso	Executes the first code block if condition1 is "True", otherwise checks condition2, and so on. if no condition is "True", the site block is executed.	2 print('It's a hot day!') Syntax 1 if condition: 2 # Code if condition: True 3 4 elif condition2: 5 # Code if condition2 is True 6 7 else: 8 # Code if no condition is True Cample: Lacore # 85 # Example score			
H-EH-Else	Executes the first code block if condition1 is "True", otherwise checks condition2, and so on, if no condition is "True", the else block is executed.	2 print('It's a hot day!') Syntax 1 if condition: 2 # Code if condition is True 3 4 clif condition2 5 # Code if condition2 is True 6 7 clse: 8 # Code if no condition is True Example: 1 score * 85 # Example score 2 if score > 80;			
Il-EM-Elso	Executes the first code block if condition! is "True", otherwise checks condition2, and so on, if no condition is "True", the else block is executed.	2 print("It's a hot day!") Syntax 1 if condition: 2 # Code if condition is True 3 4 clif condition2: 5 # Code if condition2 is True 6 6 clse: 8 # Code if no condition is True Example: 1 score * 85 # Example score 2 if score > 80: 3 print("You got an Al") 4 clif score > 80: 4 clif score > 80:			
MENERO	Executes the first code block if condition is "True", otherwise checks condition2, and so on. If no condition is "True", the else block is executed.	2 print("It's a hot day!") Symtax 1 if condition1: 2 # Code if condition1 is True 3 4 elif condition2: 5 # Code if condition2 is True 6 7 else: 8 # Code if no condition is True Example: Lascore = RS # Example score 2 if score >> 80: 3 print("You get an Al") 4 elif score >> 80: 5 print("You get an Al") 5 print("You get an Al") 6			
M-EM-Eto	Executes the first code block if condition1 is 'True', otherwise checks condition2, and so on. If no condition is 'True', the else block is executed.	2 print('R's a hot day!') Syntax 1 if condition1: 2 # Code if condition1 is True 3 clif condition2: 5 # Code if condition2 is True 6 7 clse: 8 # Code if no condition is True 2 2 if core > 88 # Example score 2 if core > 89: 3 print('You get an Al') 4 clif score > 89: 5 print('You get an Al') 6 clse: 7 print('You get an Al') 6 clse: 7 print('You need to work harder.')			
M-Est-Etse	Executes the first code block if condition! is "True", otherwise checks condition2, and so on. If no condition is "True", the else block is executed.	2 print('Ti's a hot day!') Symtax 1 if condition: 2 # Code if condition is True 3 4 clif condition: 5 # Code if condition is True 6 7 clse: 8 # Code if condition is True Example: 1 score = 85 # Example score 2 if score >= 80: 3 print('You get an Al') 4 clif score >= 80: 5 print('You get a B.') 6 else:			
Il-Esr-Eise	Executes the first code block if condition! is "True", otherwise checks condition2, and so on. If no condition is "True", the else block is executed.	2 print('Pi's a hot day!') Syntax 1 if condition: 2 # Code if condition is True 3 4 clif condition: 5 # Code if condition is True 6 7 clie: 8 # Code if condition is True Example: Lample: 1 score = 88 # Example score 2 if score >= 89: 3 print('Pou get a N.') 4 clif score >= 89: 5 print('You get a N.') 6 clies: 7 print('You get a N.') 6 clies: 7 print('You need to work harder.'') 8			

		2 else: # Code, if condition is False	6
If-Else Statement	Executes the first code block if the condition is "True", otherwise the second block.	Example:	
		1 if age >= 18: 2 print("You're an adult.") 3 else:	
		4 print("You're not an adult yet.")	40
		Syntax: 1 variable1 <= variable2	අා
		Example 1: 1 5 <= 5 and 3 <= 5	අා
Less Than or Equal To(<=)	Checks if the value of variable1 is less than or equal to variable2.	returns True	
		Example 2:	
		2 max_size = 40 3 size <= max_size	
		returns True	
		Syntax: 1 variable1 < variable2	අ
		Example 1:	
Less Than(<)	Checks if the value of variables is less than variable2.	1 4 < 6 returns True	අ
Less Inan(c)	Unicode in the value or variable is less than variables.	Example 2:	
		1 score = 68 2 passing_score = 65 3 score < passing_score	
		returns True	
		Syntax: 1 for: # Code to repeat	
		7 continue Example 1:	
Loop Controls	'break' exits the loop prematurely.' continue' skips the rest of the current iteration and moves to the next iteration.	1 for num in range(1, 6): 2 if num == 3:	
		3 break 4 print(num)	
		Example 2:	
		1 for num in range(1, 6): 2 if num == 3: 3 continue	
		4 print(num)	୧ ଥ
		Syntax: 1 Ivariable	ව
NOT	Returns 'True' if variable is 'False', and vice versa.	Example:	
		!isLocked returns True if the variable is False (i.e., unlocked).	අ
		Syntax:	
		1 variable1 != variable2 Example:	
		1 a = 10 2 b = 20	
Not Equal(!=)	Checks if two values are not equal.		
		returns True Example 2:	
		1 count=0 2 count != 0	
		returns False	
		Syntax: 1 object_name = ClassName(arguments)	අ
Object Creation	Creates an instance of a class (object) using the class constructor.	Example:	
		1 person1 = Person("Alice", 25)	හ
		Syntax: 1 statement1 statement2	අ
OR	Returns 'True' if either statement1 or statement2 (or both) are "True'. Otherwise, returns 'False'.	Example:	
		1 "Farewell Party Invitation" 2 Grade = 12 grade == 11 or grade == 12	
		returns True	
		Symtax: 1 range(stop) 2 range(start, stop)	
ranga	Generates a sequence of numbers within a specified range.	3 range(start, stop, step)	
range()	Onnewser despetice or numbers limits a appealed range.	Example: 1 range(5) #generates a sequence of integers from 0 to 4	
		2 range(2, 10) #generates a sequence of integers from 2 3 range(1, 11, 2) #generates odd integers from 1 to 9.	
		Syntax: 1 return value	e ₀
Return Statement	'Return' is a keyword used to send a value back from a function to its caller.	Example:	
		1 def add(a, b): return a + b 2 result = add(3, 5)	
		Syntax:	
		1 try: # Code that might raise an exception except 2 ExceptionType: # Code to handle the exception	6
Try-Except Block	Tries to execute the code in the try block. If an exception of the specified type occurs, the code in the except block is executed.	Example:	
		1 try: 2 num = int(input("Enter a number: ")) 3 except ValueError: 4 print("Invalid input. Please enter a valid number.	

Try-Except with Else Block	Code in the "else" block is executed if no exception occurs in the try block.	Symbox 1 try: # Code that might raise an exception except 2 ExceptionType: # Code to handle the exception 3 else: # Code to execute if no exception occurs			
		Example:			
		1 try: 2 num = int(input("inter a number: ")) 3 except Valuetror: 4 print("invalid input. Please enter a valid number") 5 else: 6 print("You entered:", num)			
	Code in the "finally" block always executes, regardless of whether an exception occurred.	Syntax:			
Try-Except with Finally Block		1 try: # Code that might raise an exception except 2 ExceptionType: # Code to handle the exception 3 finally: # Code that always executes			
		Example:			
		1 try: 2 file = open("data.txt", "r") 3 data = file.read() 4 except filebutionoffror: 5 print("file not found.") 6 finally: 7 file.close()			
		Syntax: 1 while condition: # Code to repeat 4			
While Loop	A 'white' loop repeatedly executes a block of code as long as a specified condition remains "True".	Example:			
		1 count = 0 while count < 5: 2 print(count) count += 1			

Skills Networl

© IBM Corporation. All rights reserved