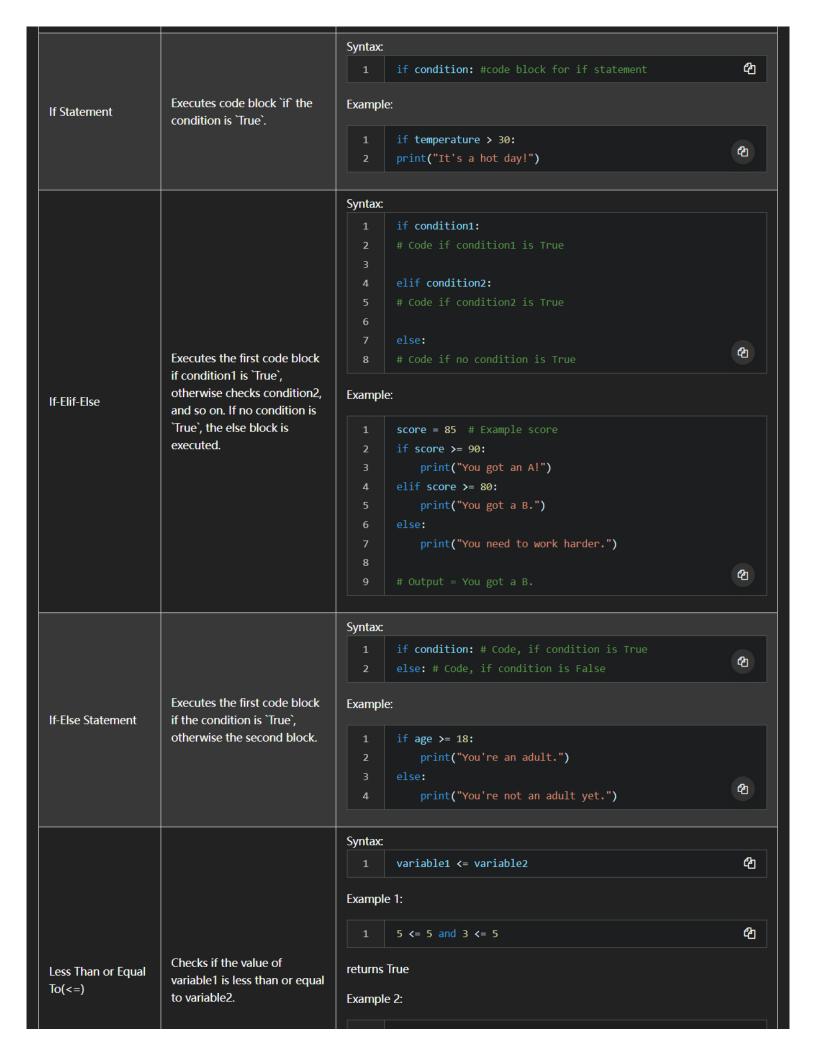
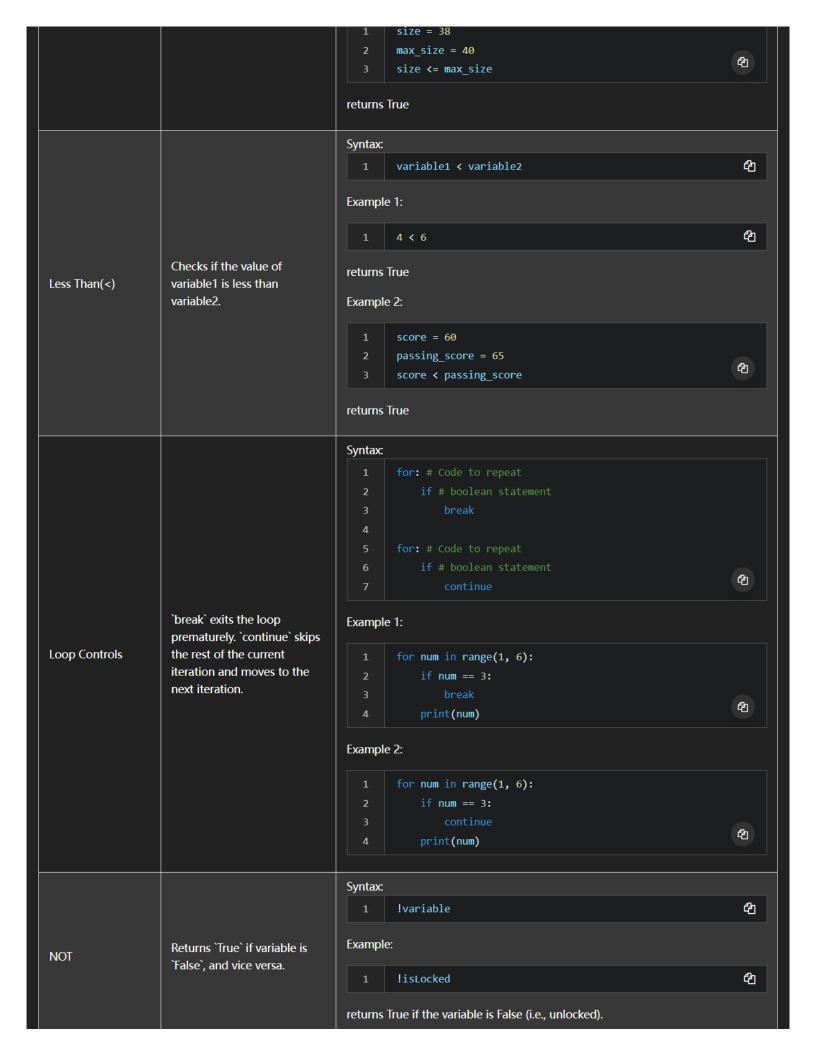
Python Programming Fundamentals Cheat Sheet

Package/Method	Description	Syntax and Code Example	
		Syntax:	
		1 statement1 and statement2	අු
	Returns `True` if both statement1 and statement2 are `True`. Otherwise, returns `False`.	Example:	
		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")	
		8	an a
		9 # Output = qualify for honors	4 2
		Syntax	
		Syntax: 1 class ClassName: # Class attributes and methods	අු
	Defines a blueprint for creating objects and defining their attributes and behaviors.	T Class Classiance: If Class deci 15aces and incensus	
		Example:	
Class Definition		1 class Person:	
		2 definit(self, name, age):	
		3 self.name = name	മ
		4 self.age = age	7
	A `function` is a reusable block of code that performs a specific task or set of tasks when called.	Syntax:	
		1 def function_name(parameters): # Function body	අු
Define Function		Example:	
		Example.	
		1 def greet(name): print("Hello,", name)	අු
		Syntax:	
Equal(==)	Checks if two values are equal.	Syntax: 1 variable1 == variable2	අු
		Example 1:	
		1 5 == 5	අු
		returns True	
		Example 2:	
		1 age = 25 age == 30	අු

		returns False	
		Syntax:	
For Loop	A `for` loop repeatedly executes a block of code for a specified number of iterations or over a sequence of elements (list, range, string, etc.).	1 for variable in sequence: # Code to repeat	අු
		Example 1:	
		1 for num in range(1, 10): 2 print(num)	අු
		Example 2:	
		<pre>fruits = ["apple", "banana", "orange", "grape", "kiwi"] for fruit in fruits: print(fruit)</pre>	අු
	A function call is the act of executing the code within the function using the provided arguments.	Syntax:	
		1 function_name(arguments)	එ
Function Call		Example:	
		1 greet("Alice")	අු
	Checks if the value of variable1 is greater than or equal to variable2.	Syntax:	
		1 variable1 >= variable2	එ
		Example 1:	
		1 5 >= 5 and 9 >= 5	අු
Greater Than or		returns True	
Equal To(>=)		Example 2:	
		1 quantity = 105 2 minimum = 100	
		3 quantity >= minimum	අු
		returns True	
	Checks if the value of variable1 is greater than variable2.	Syntax:	1 L
		1 variable1 > variable2	එ
		Example 1: 9 > 6	
Greater Than(>)		returns True Example 2:	
		1 age = 20	
		2 max_age = 25	4 2
		3 age > max_age	
		returns False	





		Syntax:
		returns True if the variable is False (i.e., unlocked).
		Syntax:
Not Equal(!=)	Checks if two values are not equal.	1 variable1 != variable2
		Example:
		1 a = 10
		2 b = 20
		3 a != D
		returns True
		Example 2:
		1 count=0 2 count != 0
		returns False
	Creates an instance of a class (object) using the class constructor.	Syntax:
		1 object_name = ClassName(arguments) 但
Object Creation		Example:
		1 person1 = Person("Alice", 25)
		Syntax:
	Returns `True` if either statement1 or statement2 (or both) are `True`. Otherwise, returns `False`.	1 statement1 statement2 但
		Example:
OR		
		1 "Farewell Party Invitation" 2 Grade = 12 grade == 11 or grade == 12
		returns True
	Generates a sequence of numbers within a specified range.	Syntax: 1 range(stop)
		2 range(start, stop)
		3 range(start, stop, step)
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 to 4
		Syntax:
		1 return value
Return Statement	`Return` is a keyword used to send a value back from a	Example:

