

Python Programming Fundamentals Cheat Sheet

Package/Method	Description	Syntax and Code Example
AND	Returns 'True' if both statement1 and statement2 are 'True'. Otherwise, returns 'False'.	<p>Syntax:</p> <pre>1. 1 2. statement1 and statement2</pre> <p>Example:<pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre><p>Syntax:</p><pre>1. 1 2. statement1 and statement2</pre><p>Example:<pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre></p></p>
Class Definition	Defines a blueprint for creating objects and defining their attributes and behaviors.	<p>Syntax:</p> <pre>1. 1 2. class ClassName: # Class attributes and methods</pre> <p>Example:<pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre></p>
Define Function	A 'Function' is a reusable block of code that performs a specific task or set of tasks when called.	<p>Syntax:</p> <pre>1. 1 2. def function_name(parameters): # Function body</pre> <p>Example:</p> <pre>1. 1 2. def greet(name): print("Hello,", name)</pre>
Equal(==)	Checks if two values are equal.	<p>Syntax:</p> <pre>1. 1 2. variable1 == variable2</pre> <p>Example 1:</p> <pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre>
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.).	<p>Syntax:</p> <pre>1. 1 2. for item in range(1, 10): 3. print(item)</pre> <p>Example 1:</p> <pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre>
Function Call	A function call is the act of executing the code within the function using the provided arguments.	<p>Syntax:</p> <pre>1. 1 2. function_name(arguments)</pre> <p>Example:</p> <pre>1. 1 2. greet("Alice")</pre>
Greater Than or Equal To(>=)	Checks if the value of variable1 is greater than or equal to variable2.	<p>Syntax:</p> <pre>1. 1 2. variable1 >= variable2</pre> <p>Example 1:</p> <pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre>
Greater Than(>)	Checks if the value of variable1 is greater than variable2.	<p>Syntax:</p> <pre>1. 1 2. variable1 > variable2</pre> <p>Example 1:</p> <pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre>
IF Statement	Executes code block 'if' the condition is 'True'.	<p>Syntax:</p> <pre>1. 1 2. if condition: 3. # Code to execute if condition is True</pre> <p>Example:</p> <pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre>
IF ELSE 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.	<p>Syntax:</p> <pre>1. 1 2. if condition1: 3. # Code to execute if condition1 is True 4. elif condition2: 5. # Code to execute if condition2 is True 6. else: 7. # Code to execute if no condition is True</pre> <p>Example:</p> <pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre>
IF Else Statement	Executes the first code block if the condition is 'True', otherwise the second block.	<p>Syntax:</p> <pre>1. 1 2. if condition: 3. # Code to execute if condition is True 4. else: 5. # Code to execute if condition is False</pre> <p>Example:</p> <pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre>
Less Than or Equal To(<=)	Checks if the value of variable1 is less than or equal to variable2.	<p>Syntax:</p> <pre>1. 1 2. variable1 <= variable2</pre> <p>Example 1:</p> <pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre>
Less Than(<)	Checks if the value of variable1 is less than variable2.	<p>Syntax:</p> <pre>1. 1 2. variable1 < variable2</pre> <p>Example 1:</p> <pre>1. 1 2. 2 3. 3 4. 4 5. 5 6. 6 7. 7 8. 8 9. 9 10. 10</pre>

 Skills Network

© IBM Corporation. All rights reserved.