

pseudocode to python

For OCR (9-11) Computer Science



OCR Pseudocode to Python

Syntax Topic	OCR Pseudocode	Result	Python
Local variables	<code>x = 10</code> <code>playerName = "Sam"</code>		<code>x = 10</code> <code>playerName = "Sam"</code>
Global variables	<code>global currentUserID = 223</code>	In Python, variables are global if they are declared at the start of the program. If you need to write to the variable in a function/procedure you must use the 'global' keyword before you can use it.	<code>currentUserID = 0</code> <code>def myProcedure():</code> <code>global currentUserID</code> <code>currentUserID = 223</code>
Casting	<code>str(3)</code> <code>int("3")</code> <code>float("3.14")</code>	returns the string "3" returns the integer 3 returns the floating point number 3.14	<code>str(3)</code> <code>int(3)</code> <code>float("3.14")</code>
Output to screen	<code>print("hello everyone")</code>	Outputs "hello everyone"	<code>print("hello everyone")</code>
Input from keyboard	<code>name = input("Enter name")</code>	Outputs "Enter name" on screen. User types text. The text is stored as a string inside the variable <i>name</i>	<code>name = input("Enter name")</code>
Iteration – Count Controlled (For loop)	<code>for i = 0 to 9</code> <code>print("Hi")</code>	Outputs "Hi" on the screen 10 times (from 0 to 9) – pseudocode includes the last number, Python doesn't	<code>for i in range(0,10):</code> <code>print("Hi")</code>

Syntax Topic	OCR Pseudocode	Result	Python
Iteration – Condition Controlled (While loop and Do-While loop)	<pre>password = "" while password != "CdRtp45@" password = input("Password?") endwhile</pre>	<p>Asks the user for their <i>Password?</i> Checks if the password is equal to <i>CdRtp45@</i> Repeats if <i>Password</i> isn't equal</p>	<pre>password = "" while password != "CdRtp45@": password = input("Password?")</pre>
	<pre>do password = input("Password?") until password == "CdRtp45@"</pre>	<p>This gives the same result as the previous code. Python doesn't have a do-until structure, but this shows how the question can be asked first regardless of the condition.</p> <p><i>[Note: You could implement a do-until loop in Python by using a "while True" loop and break. This is bad programming practice. The sample code also uses two password = input("Password?") statements – this is bad practice and should instead make use of a procedure]</i></p>	<pre>password = input("Password?") while not (password == "CdRtp45@"): password = input("Password?")</pre>

Syntax Topic	OCR Pseudocode	Result	Python
Logical and Comparison Operators	<p>Example: while x <= 10 AND x >= 5</p> <p>Logical Operators: AND OR NOT</p> <p>Comparison Operators: == != < <= > >=</p>	<p>Loops if x is between 5 and 10 inclusive AND is the logical operator in this example</p> <p>Operator meaning: And Or Not</p> <p>Equal to Not equal to Less than Less than or equal to Greater than Greater than or equal to</p>	<p>Example: while x <= 10 and x >= 5</p> <p>Logical Operators: and or not</p> <p>Comparison Operators: == != < <= > >=</p>
Arithmetic Operators	<p>+</p> <p>-</p> <p>*</p> <p>/</p> <p>MOD</p> <p>DIV</p> <p>^</p>	<p>Addition</p> <p>Subtraction</p> <p>Multiplication</p> <p>Division</p> <p>Modulus (remainder)</p> <p>Quotient / Integer division / Floor division</p> <p>Exponentiation / Power of</p>	<p>+</p> <p>-</p> <p>*</p> <p>/</p> <p>%</p> <p>//</p> <p>**</p>
String Operator	+	Concatenation (combine two strings together)	+

Syntax Topic	OCR Pseudocode	Result	Python
Selection	<pre> if choice == "heads" then print("You chose heads") elseif choice == "tails" then print("You chose tails") else print("Invalid choice") endif </pre>		<pre> if choice == "heads": print("You chose heads") elif choice == "tails": print("You chose tails") else: print("Invalid choice") </pre>
Switch/Case	<pre> switch coinResult: case "heads": print("You got heads") case "tails": print("You got tails") default: print("Not valid coin") endswitch </pre>	<p>The switch statement will select one of the many different cases based on the value stored in coinResult. If none of the cases match then the default case will be executed</p> <p>Python doesn't have a switch/case structure. A dictionary structure in Python may be useful in providing similar functionality</p>	
String length	<pre>stringname.length</pre> <p>Example:</p> <pre>myName = "Sophie" print(myName.length)</pre>	<p>Outputs 6 on the screen</p>	<pre>len(stringName)</pre> <pre>myName = "Sophie" print(len(myName))</pre>
Substring	<pre>stringname.subString(startingPosition, numberOfCharacters)</pre> <p>Example:</p> <pre>myName = "Sophie" print(myName.substring(2,2))</pre>	<p>Outputs <i>ph</i> on the screen</p> <p>Note that the character position starts at the 0th character which is S in this example</p>	<pre>stringname[start:end]</pre> <pre>myName = "Sophie" print(myName[2:4])</pre>

Syntax Topic	OCR Pseudocode	Result	Python
Subroutines - Functions	<pre>function add(firstNum, secondNum) total = firstNum + secondNum return total endfunction totalScore = add(5,3)</pre>	<p>This function takes two arguments (inputs) which are firstNum and secondNum and returns the total of the two added together</p> <p>This calls the function and assigns the returned value to the variable <i>totalScore</i></p>	<pre>def add(firstNum, secondNum): total = firstNum + secondNum return total totalScore = add(5,3)</pre>
Subroutines - Procedures	<pre>procedure showNames(firstName, lastName) print("Your name is: ") print(firstName + lastName) endprocedure showNames("Sam", "Green")</pre>	<p>Your name is: Sam Green</p> <p>This procedure has two arguments (inputs) which are firstName and lastName. When the procedure is called the names are concatenated together and output to the screen</p>	<pre>def showName(firstName, lastName): print("Your name is: ") print(firstName + lastName) showNames("Sam", "Green")</pre>
Pass by Reference Pass by Value	<pre>a = 0 b = 0 procedure move(x: byVal, y: byRef) x = x + 1 y = y + 1 endprocedure move(a, b) print(a) print(b)</pre>	<p>Output: 0 1</p> <p>In the pseudocode, the variable x will be passed to the procedure by copying the value. Changing x in the procedure won't change the original value.</p> <p>The variable y will be passed to the procedure by pointing to the original value. Changing y in the procedure will change the original value.</p>	<p>Python doesn't work in the same way as the pseudocode and there is no option to pass by reference. For most cases you can think of Python as passing by value.</p>

Syntax Topic	OCR Pseudocode	Result	Python
Arrays	<pre>array testScores[4] testScores[0] = 90 testScores[1] = 100 testScores[2] = 50 testScores[3] = 75 print(testScores[2])</pre>	<p>Outputs 50</p> <p>Array referencing starts at 0, so to access the first element in the array, use: testScores[0]</p> <p>Python uses lists not arrays. These are different to array as mixed data types can be stored inside them – for example, you can store integers and strings inside the same list – in an array you would need to use two lists, one for each data type</p> <p>Python lists can also grow or shrink. Arrays have a fixed length. For this reason, you do not need to initialise a list with a length before assigning, but you can access/assign individual elements in the same way as the pseudocode.</p>	<pre>testScores = [90, 100, 50, 75] print(testScores[2])</pre> <p>Alternatively:</p> <pre>testScores = [None]*4 testScores[0] = 90 testScores[1] = 100 testScores[2] = 50 testScores[3] = 75 print(testScores[2])</pre>

Syntax Topic	OCR Pseudocode	Result	Python
2D Arrays	<pre>array board[8, 8] board[0,0] = "castle"</pre>	<p>The first line of code creates a 2D array – this is an array that contains other arrays; think of it as a grid or table.</p> <p>You would expect that the following Python code would initialise the array: <code>board = [[None]*8]*8</code> This will not work as each row in the array will be a reference to the first one. So when you change <code>board[0][0]</code> you would change the entire first column.</p> <p>The second example shows a less complicated way in which an array can be built in Python (but obviously this requires more lines of code).</p>	<pre>board = [[None]*8 for _ in range(8)] board[0][0] = "castle" #alternative way to build the array board = [] for i in range(8): row = [] for j in range(8): row.append(None) board.append(row) board[0][0] = "castle"</pre>
Opening and Reading from Files	<pre>myFile = openRead("example.txt") line = myFile.readLine() print(line) myFile.close()</pre>	<p><code>myFile</code> is a file handler. It stores the "link" to the file that has been opened. The 'r' character shows that the file is open in read mode in Python.</p> <p>Readline is a method which will read the next line in the file; this is then assigned to the line variable</p> <p>It is important to close the file once it has been used to that other applications can use the file</p>	<pre>myFile = open("example.txt", 'r') line = myFile.readline() print(line) myFile.close()</pre>

Syntax Topic	OCR Pseudocode	Result	Python
Reading to the End of File	<pre>myFile = openRead("example.txt") while NOT myFile.endOfFile() print(myFile.readLine()) endwhile myFile.close()</pre>	<p>To read all the lines in a file, first open the file, then use a loop to read each of the lines, then close the file.</p> <p>In pseudocode (and many other languages) we read the line until we reach end of file (EOF). In Python we use a for loop to do this.</p> <p>\n is a newline character. We use this in a string to mean a new line (return). When you read a file in Python, the new line character is also read. When printed this gives an extra line. The rstrip() method removes the new line character.</p>	<pre>myFile = open("example.txt", 'r') for line in myFile: print(line) myFile.close() #remove the new line character when #reading from a file myFile = open("example.txt") for line in myFile: print(line.rstrip()) myFile.close()</pre>
Writing to a File	<pre>myFile = openWrite("example.txt") myFile.writeLine("Some text") myFile.close()</pre>	<p>The 'w' character shows that the file is open in write mode in Python.</p> <p>This code will replace the contents of a file with whatever write() or writeLine() statements are written; append mode will "add" to the contents of a file</p>	<pre>myFile = open("example.txt", 'w') myFile.write("Some text") myFile.close()</pre>
Comments	<pre>print("hello") //Comments go here</pre>	<p>The // symbols in pseudocode and the # symbol in Python are used for comments. Any text (on the same line) after the comment symbol will be ignored by the program.</p>	<pre>print("hello") #Comments go here</pre>