

Code Styling Conventions: MTEL Digital Literacy and Computer Science Test

The following conventions are followed in the code samples found on this test. Alternative styles are acceptable in candidate-created pseudocode in the open-response assignments. Candidate-created pseudocode will not be evaluated for syntax, but instead for how well it demonstrates programming concepts and solves the presented problem.

Category	Notation	Example
Assignment	=	<pre>x = 5 y = TRUE my_string = "Some text." my_list = [2, 3.14, "apple"]</pre>
Spacing	indents indicate control flow	<pre>while (x != 0) x = x - 1 print(x)</pre>
Mathematics	+ addition (or string concatenation) - subtraction * multiplication / division % modulo ^ exponentiation	<pre>x + b "Some " + "text." x - b x * b x / b x % b x ^ b</pre>
Relationships	< less than > greater than <= less than or equal to >= greater than or equal to == equal to != not equal to is string equality	<pre>x < b x > b x <= b x >= b x == b x != b word = "bird" word is "bird" == TRUE</pre>
Logic	TRUE condition true FALSE condition false AND both conditions true OR at least one condition true NOT condition not true	<pre>(5 < 7) == TRUE (5 > 7) == FALSE (5 < 7) AND (5 == 5) == TRUE (5 > 7) OR (5 == 5) == TRUE NOT (5 > 7) == TRUE</pre>
Indexes	[0] the first item in a list or string [n] the (n - 1)th item in a list or string [x,y] the (x - 1)th row and (y - 1)th column in a 2d array	<pre>my_sentence[0] my_list[1] #accesses 2nd element in my_list my_array[2,5]</pre>

Category	Notation	Example
Conditionals	<pre>if (condition) block</pre> <pre>if (condition) block else block</pre>	<pre>if (x >= 100) print("A large number.")</pre> <pre>if (x <= 100) print("A small number.") else print("A large number.")</pre>
Iterables	<pre>while (condition) block</pre> <pre>for (i = a; i < b; i = i + c) block</pre>	<pre>while (x < 10) print(x) x = x + 1</pre> <pre>for (i = 1; i < 10; i = i + 1) print(i ^ 2)</pre>
Functions	<pre>function function_name(arg_1,...arg_n) block</pre> <pre>function_name(val_1,...val_n)</pre>	<pre>function adder(num_1, num_2) return num_1 + num_2</pre> <pre>adder(3, 4)</pre>
Comments	<pre># comment line</pre>	<pre># some information</pre>
Object-Oriented Notation	<pre>object.class_method()</pre> <pre>object.attribute</pre>	<pre>my_sentence.length()</pre> <pre>my_tree.grow(12)</pre> <pre>my_house.color</pre>
Classes	<pre>class Class_Name(parent Class_Name, if any) public/private attribute_name = default_value public/private function class_function_name(arg_1,...arg_n) block class_instance = Class_Name(attr_1 = value,...attr_n = value)</pre>	<pre>class Tree(Plant) public height = 10 public type = "oak" private kingdom = "Plantae" public function grow(height) height = height + 2 my_tree = Tree(height = 9, type = "elm")</pre>