Study Guide: Week 1 Graded Quiz

Study Guide: Week 1 Graded Assessment

It is time to prepare for your first graded quiz! Please review the following items from this module before beginning the Week 1 Graded Quiz. If you would like to refresh your memory on these materials, please revisit the Study Guides located before each Practice Quiz in Week 1: Study Guide: Introduction to Programming, Study Guide: Introduction to Python, and Study Guide: First Programming Concepts.

Knowledge

- Benefits of the Python programming language
- How Python compares to other programming languages
- How the knowledge of one programming language affects learning and using other programming languages
- How scripting applies to automation
- Proper syntax for arithmetic operations
- Functions and keywords used to display data
- Why precision is important when programming computer instructions

Terms

- Computer programs
- Programming language
- Syntax
- Semantics
- Logic errors
- Script

- Automation
- Function

Coding skills

Skill 1

- Use the print() function to output a string

```
# Syntax for printing a string of text
print("I love Python!")

# Syntax for printing numeric values
print(360)
print(32*45)

# Syntax for printing the value of a variable
value = 8*6
print(value)
```

Skill 2

- Use arithmetic operators, with a focus on exponents

```
# Multiplication, division, addition, and subtraction
print(3*8/2+5-1)

# Exponents
print(4**6) # Syntax means 4 to the power of 6
print(4**2) # To square a number
print(4**3) # To cube a number
print(4**0.5) # To find the square root of a number

# To calculate how many different possible combinations can be
# formed using a set of "x" characters with each character in "x"
```

```
# having "y" number of possible values, you will need to use an
# exponent for the calculation:
x = 4
y = 26
print(y**x)
```

Skill 3

- Use variables with assignment and arithmetic operators

```
# Assignment of values to the variables:
years = 10
weeks_in_a_year = 52
# This variable is assigned an arithmetic calculation:
weeks_in_a_decade = years * weeks_in_a_year
# Prints the calculation stored in the "weeks_in_a_decade" variable:
print(weeks_in_a_decade)
```

Reminder: Correct syntax is critical

Using precise syntax is critical when writing code in any programming language, including Python. Even a small typo can cause a syntax error and the automated Python-coded quiz grader will mark your code as incorrect. This reflects real life coding errors in the sense that a single error in spelling, case, punctuation, etc. can cause your code to fail. Coding problems caused by imprecise syntax will always be an issue whether you are learning a programming language or you are using programming skills on the job. So, it is critical to start the habit of being precise in your code now.

No credit will be given if there are any coding errors on the automated graded quizzes - including minor errors. Fortunately, you have 3 optional retake opportunities on the graded quizzes in this course. Additionally, you have unlimited retakes on practice quizzes and can review the videos and readings as many times as you need to master the concepts in this course.

Now, before starting the graded quiz, review this list of common syntax errors coders make when writing code.

Common syntax errors:

- Misspellings
- Incorrect indentations
- Missing or incorrect key characters:
    - Parenthetical types - (curved), [square], { curly }
    - Quote types - "straight-double" or 'straight-single', “curly-double” or ‘curly-single’
    - Block introduction characters, like colons - :
- Data type mismatches
- Missing, incorrectly used, or misplaced Python reserved words
- Using the wrong case (uppercase/lowercase) - Python is a case-sensitive language

Resources

For additional Python practice, the following links will take you to several popular online interpreters and codepads:

- Welcome to Python
- Online Python Interpreter
- Create a new Repl
- Online Python-3 Compiler (Interpreter)
- Compile Python 3 Online
- Your Python Trinket