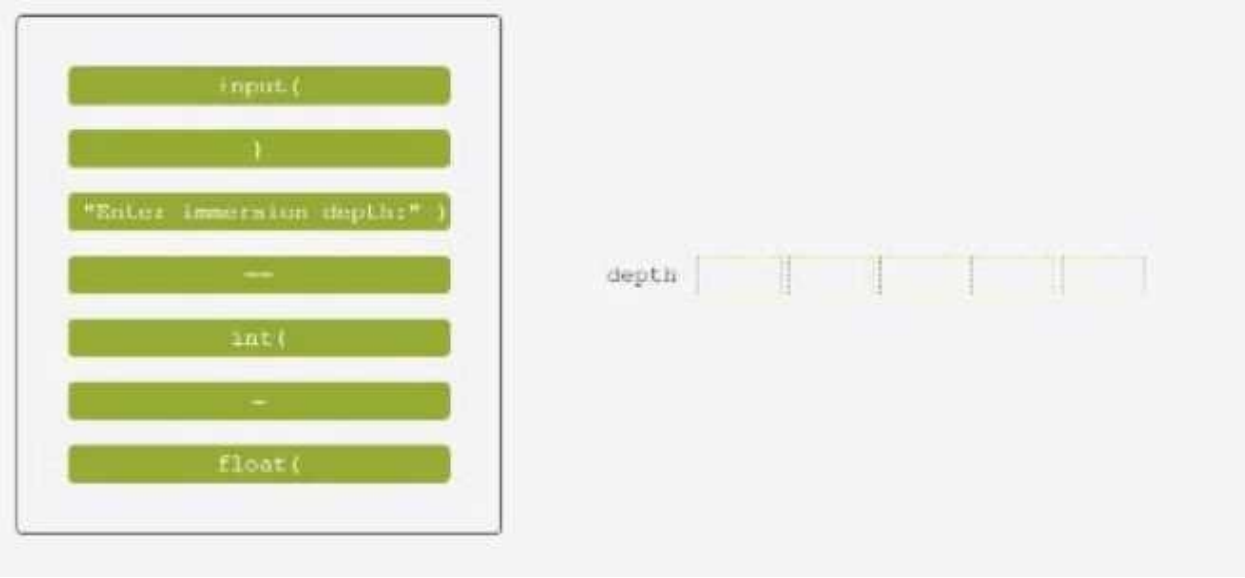
PCEP-30-02 Demo

Question: 1

DRAG DROP

Insert the code boxes in the correct positions in order to build a line of code which asks the user for an integer value and assigns it to the depth variable.

(Note: some code boxes will not be used.)



Answer:

Explanation:

One possible way to insert the code boxes in the correct positions in order to build a line of code which asks the user for an integer value and assigns it to the depth variable is:

depth = int(input("Enter the immersion depth: "))

This line of code uses the input function to prompt the user for a string value, and then uses the int function to convert that string value into an integer number. The result is then assigned to the variable depth.

You can find more information about the input and int functions in Python in the following references:

[Python input() Function]

[Python int() Function]

Question: 2

A set of rules which defines the ways in which words can be coupled in sentences is called:

A. lexis

B. syntax

C. semantics

D. dictionary

Answer: B

Explanation:

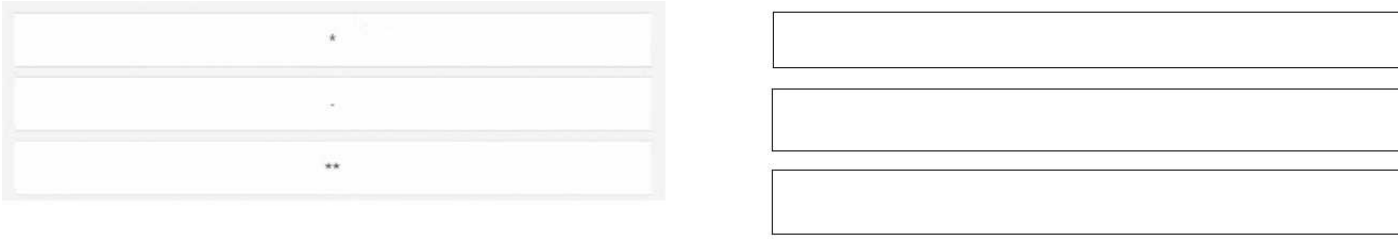
Syntax is the branch of linguistics that studies the structure and rules of sentences in natural languages. Lexis is the vocabulary of a language. Semantics is the study of meaning in language. A dictionary is a collection of words and their definitions, synonyms, pronunciations, etc.

Reference: [Python Institute - Entry-Level Python Programmer Certification]

Question: 3

DRAG DROP

Arrange the binary numeric operators in the order which reflects their priorities, where the top-most position has the highest priority and the bottom-most position has the lowest priority.



Answer:

Explanation:

The correct order of the binary numeric operators in Python according to their priorities is:

Exponentiation (**)

Multiplication (*) and Division (/, //, %)

Addition (+) and Subtraction (-)

This order follows the standard mathematical convention of operator precedence, which can be remembered by the acronym PEMDAS (Parentheses, Exponents, Multiplication/Division, Addition/Subtraction). Operators with higher precedence are evaluated before those with lower precedence, but operators with the same precedence are evaluated from left to right. Parentheses can be used to change the order of evaluation by grouping expressions.

For example, in the expression 2 + 3 * 4 ** 2, the exponentiation operator (**) has the highest priority, so it is evaluated first, resulting in 2 + 3 * 16. Then, the multiplication operator (*) has the next highest priority, so it is evaluated next, resulting in 2 + 48. Finally, the addition operator (+) has the lowest priority, so it is evaluated last, resulting in 50.

You can find more information about the operator precedence in Python in the following references:

6. Expressions — Python 3.11.5 documentation

Precedence and Associativity of Operators in Python - Programiz

Python Operator Priority or Precedence Examples Tutorial

Question: 4

Which of the following expressions evaluate to a non-zero result? (Select two answers.)

A. 2 ** 3 / A - 2

B. 4 / 2 * * 3 - 2

C. 1 * * 3 / 4 - 1

D. 1 * 4 // 2 ** 3

Answer: AB

Explanation:

In Python, the ** operator is used for exponentiation, the / operator is used for floating-point division, and the // operator is used for integer division. The order of operations is parentheses, exponentiation, multiplication/division, and addition/subtraction. Therefore, the expressions can be

evaluated as follows:

A) 2 ** 3 / A - 2 = 8 / A - 2 (assuming A is a variable that is not zero or undefined) B. 4 / 2 * * 3 - 2 = 4 / 8 - 2 = 0.5 - 2 = -1.5 C. 1 * * 3 / 4 - 1 = 1 / 4 - 1 = 0.25 - 1 = -0.75 D. 1 * 4 // 2 ** 3 = 4 // 8 = 0

Only expressions A and B evaluate to non-zero results.

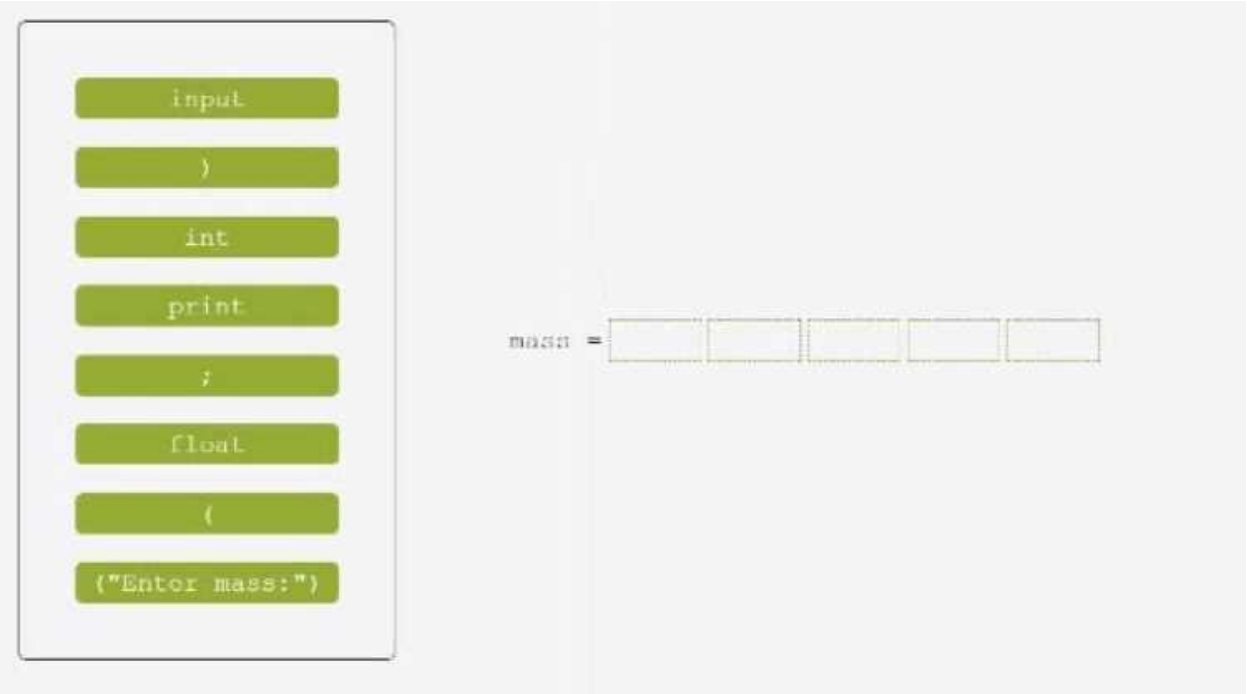
Reference: [Python Institute - Entry-Level Python Programmer Certification]

Question: 5

DRAG DROP

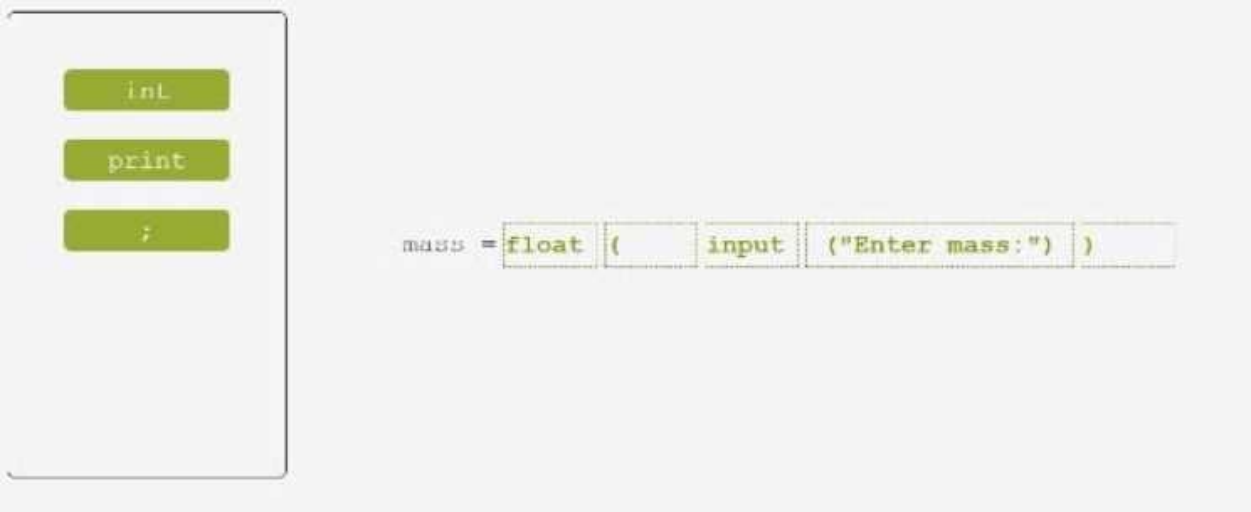
Insert the code boxes in the correct positions in order to build a line of code which asks the user for a float value and assigns it to the mass variable.

(Note: some code boxes will not be used.)



Answer:

Explanation:



One possible way to insert the code boxes in the correct positions in order to build a line of code that asks the user for a float value and assigns it to the mass variable is:

mass = float(input("Enter the mass: "))

This line of code uses the input function to prompt the user for a string value, and then uses the float function to convert that string value into a floating-point number. The result is then assigned to the variable mass.

You can find more information about the input and float functions in Python in the following references:

[Python input() Function]

[Python float() Function]