Validation OTP for CodeTantra | CodeTantra Edu

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c166d&bd=AMTc3M19jdF9jaA==

Q1

100%

L3 / Variables and Assignment / Understanding Variables

Variables in Python:

- In Python, a variable is a reserved memory location used to store values.
- Python being a dynamically typed language, there is no need to declare variables or declare their types before using them.
- Python has no command for declaring a variable. A variable is created the moment a value is assigned to it.
- The **equal-to (=)** operator is used to assign value to a variable.

> **Note**: **Operators** are special **symbols** used in programming languages that represent particular actions. = is called the **assignment operator**.

For example :

```
marks = 100 # Here marks is the variable and 100 is the value assigned to it.
```

> **Note**: As a good programming practice, we should use meaningful names for variables.

- Python interpreter automatically determines the type of the data, based on the data assigned to it.
- In Python, we can even change the type of the variable after it has been set once (or initialized with some other type). This can be done by assigning a value of a different type to the variable.
- A variable in Python can hold any type of value like `12` (integer), `560.09` (float), `"CodeTantra"` (string) etc.

We will learn more about data types in the coming modules.

VariablesExample.py

Submit

```
1  # In the below line, assign value 18 to the length variable
2
3  length = 18
4
5  # In the below line, assign value "King Cobra" to the snake variable
6
7  snake = "King Cobra"
8
9
10 print(snake, "can grow up to a length of", length, "feet")
```

Close | Reset | Submit

Type here to search

39°C Sunny

ENG US 4:42 PM 07/05/2024

Validation OTP for CodeTantra | CodeTantra Edu

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c166d&bd=AMTc3M19jdF9jaA==

Q2

100%

L3 / Variables and Assignment / Assigning Different Values to Variables

Associating a **value** with a **variable** using the assignment operator (=) is called as **binding**.

In Python, when you **assign** a value to a variable, you're actually creating a reference to that value, not making a copy.

Python uses **reference semantics**, which means that variables essentially point to objects rather than containing the objects themselves.

We cannot use Python variables without assigning any value.

If we try to use a variable without assigning any value then, the Python interpreter shows an error saying "name is not defined".

Fill the missing code in the given program as per the below instructions:

1. Assign the variable `value1` with value **99**
2. Assign the variable `value2` with string value **"Hello Python"**
3. Assign the variable `value3` with string value **"Hello World"**

then print the variables in the same order.

Expected Output:

```
Hello
Hello
Hello
99
Hello Python
```

VariablesExample1.py

Submit

```
1  value1 = value2 = value3 = "Hello"
2  print(value1)
3  print(value2)
4  print(value3)
5  # Assign values to variables and print again
6  value1 = 99
7  value2 = "Hello Python"
8  value3 = "Hello World"
9  print(value1)
10 print(value2)
11 print(value3)
```

Close Reset Submit

Type here to search

Hot weather

ENG US 4:42 PM 07/05/2024

Validation OTP for CodeTantra | CodeTantra Edu

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c166d&bd=AMTc3M19jdF9jaA==

Q3

100%

L3 / Variables and Assignment / Assignment of Variables

Python variables do not need explicit declaration to reserve memory space.

The declaration happens automatically when you assign a value to a variable.

The equals to sign (=) is used to assign values to variables.

The operand to the left of the = operator is the name of the variable and the operand (or expression) to the right of the = operator is the value stored in the variable.

Let us consider the below example:

```
counter = 100 # An integer assignment
print(counter) # Prints 100
miles = 1000.50 # A floating point assignment
print(miles) # Prints 1000.5
name = "John" # A string assignment
print(name) # Prints John
```

In the above example **100**, **1000.50**, and **"John"** are the values assigned to the variables **counter**, **miles**, and **name** respectively.
The above example program produces the result as:

```
100
1000.5
John
```

Complete the given code in which :

VariablesExample2.py

Submit

```
1 kilometers = int(input("Enter a value: ")) # assign the correct value
2 convertfactor = 0.621371 # assign the correct value
3 miles =  (kilometers) * (convertfactor)# multiply kilometers and convertfactor
4 print("Miles:", miles)
5
```

Close | Reset | Submit

Type here to search

Hot weather

ENG US 4:42 PM 07/05/2024

Validation OTP for CodeTantra | CodeTantra Edu

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c166d&bd=AMTc3M19jdF9jaA==

Q4

100%

L3 / Variables and Assignment / Understanding Multiple Assignment

Python allows you to assign a single value to several variables **simultaneously**.

Let us consider an example:

```
number1 = number2 = number3 = 100
```

Here an **integer object** is created with the value `100`, and all the three variables are references to the same memory location. This is called `chained assignment`.

We can also assign multiple objects to multiple variables, this is called `multiple assignment`.

Let us consider the below example:

```
value1, value2, value3 = 1, 2.5, "Ram"
```

Here the **integer object** with value `1` is assigned to variable **value1**, the **float object** with value `2.5` is assigned to variable **value2** and the **string object** with the value `"Ram"` is assigned to the variable **value3**.

Fill in the missing code in the given program to assign the integer value `25` to the variable named **age**, float value `24.789` to the variable named **length** and the string value "`Python`" to the variable named **language** using multiple assignment syntax so that their individual values are printed by the existing code.

Expected Output:

```
ggg
```

VariablesExample3.py

Submit

```
1  age,length,language =25,24.789,"Python"
2  # Assign the values to the variables
3  print(age)
4  print(length)
5  print(language)
```

Close | Reset | Submit

Type here to search

Hot weather

ENG US 4:42 PM 07/05/2024

Validation OTP for CodeTantra | CodeTantra Edu

csmssengg.codetantra.com/secure/topic-details1.jsp?cid=65969c70422c5e79de0c160e&tid=65969c71422c5e79de0c166d&bd=AMTc3M19jdF9jaA==

Q5

100%

L3 / Variables and Assignment / Chained Assignment

In **Python**, assignment statements do not return a value. Chained assignment is recognised and supported as a special case of the assignment statement.

`a = b = c = x` is called a **chained assignment** in which the value of **x** is assigned to multiple variables **a**, **b**, and **c**.

Let us consider a simple example:

```
a = b = c = d = 10
print(a) # will print result as 10
print(b) # will print result as 10
print(c) # will print result as 10
print(d) # will print result as 10
```

Here, we are initializing the variables **a**, **b**, **c**, **d** with value `10`.

Write a program to assign a user given value to `a`, `b`, `c` variables.

Note: Let us assume that `input()` is used to read values given by the user. We will learn about `input()` in later sections.

Here,

```
a = b = c = str1
```

Sample Input and Output:

```
Enter a value: Hello Python
```

ChainedAssignment.py

Submit

```
1  str1 = input("Enter a value: ")
2  #str2=input("Enter a value: ")
3  #str3=float(input("Enter a value: "))
4  # Assign str1 to three objects a, b and c
5
6  print("Value of a:",  str1  ) # Print a
7  print("Value of b:",  str1 ) # Print b
8  print("Value of c:",  str1 ) # Print c
9
10
11 #print("Value of a: ", str2)
12 #print("Value of b: ", str2)
13 #print("Value of c: ", str2)
14
15 #print("Value of a: ", str3)
16 #print("Value of b: ", str3)
17 #print("Value of c: ", str3)
```

Close | Reset | Submit

Type here to search

Hot weather

ENG US 4:42 PM 07/05/2024