

Programming Python Variables: Takeaways



by Dataquest Labs, Inc. - All rights reserved © 2022

Syntax

- Storing values to variables:

```
twenty = 20
result = 43 + 2**5
currency = 'USD'
```

- Updating the value stored in a variable:

```
x = 30
x += 10 # this is the same as x = x + 10
```

- Syntax shortcuts:

```
x += 2 # Addition
x -= 2 # Subtraction
x *= 2 # Multiplication
x /= 2 # Division
x **= 2 # Exponentiation
```

Concepts

- We can store values in the computer memory. We call each storage location in the computer's memory a **variable**.
- There are two syntax rules we need to follow when we're naming variables:
 - We must use only letters, numbers, or underscores (we can't use apostrophes, hyphens, spaces, etc.).
 - Variable names can't begin with a number.
- Whenever the syntax is correct, but the computer still returns an error, we call this a **runtime error**.
- In Python, the `=` operator tells us that the value on the right is **assigned** to the variable on the left. It doesn't tell us anything about equality. We call `=` an **assignment operator**, and we read code like `x = 5` as "five is assigned to x" or "x is assigned five," not "x equals five."
- In computer programming, we classify values into different **types** — or **data types**. A value's type offers the computer the information necessary to process that value. Depending on the type, the computer will know how to store a value in memory, or which operations it can perform on a value.