Variables, Types, Expressions, and Statemetns

Contents

Primary Uses of the Console

- Can be used from a programming point of view or a ‘user’ point of view.
- Users use the console to interact with programs.
- Often useful to have the console print values of computations or objects in your programs.
    - Helps determine intermediate values.
        - ⋆ Aids the debugging process.

Variables and Expressions: Giving Names and Values to Things

Giving Names to Things

- We reference things by using variables.
- Not exactly the same thing as a ‘variable’ in Math.
    - In programming, = is an ‘assignment’ as opposed to an equivalence.
    - We call the thing to the right of an equal sign an *expression* with a *value*.
    - The only thing that you are allowed to have to the left of an equal sign is the name of a variable.

An expression is a line of code that can be reduced to a value.

Introducing variables

Objects Are Things That Can Be Manipulated

- In python, everything is an object, and h as the following:
    - A type
        - ⋆ Tells you the data/values/attributes/etc.
    - A set of operations
        - ⋆ Commands that you can tell the object to do.

Objects Have Names

- Names are *variables*.
- Used to refer to objects

A variable is used to bind a name to an object. A variable name refers to a particular object.

Allowed Object Names

- Must begin with a letter or an underscore _.
- Other characters can be letters, numbers, or an underscore.
- Case sensitive.
- Of any length.

It’s good practice to limit lines of code to 80 characters for readability purposes. Try to make names as concise as possible.

Note: Python has a vew reserved words that can't be used as variable names. Many editors will highlight these special keywords.

Variable Good Practices

- Names should be meaningful and descriptive.
- Underscores should be used to add space between variable words.
- Variable names shouldn't be too long.
- Be consistent.

Creating Variables

- Must initialize the variableby assigning it ot an object using the equal sign. This binds the object to the variable name.

Updating a Variable

- Simply type out the variable, an equal sign, and a new value to the right of the equal sign.

Object Types and Statements of Code

- Objects have:
  - A type - Dictates the values they can have.
  - Operations you can do with them.

An object type tells you what kinds of values the object can have.

Python has certain types of objects that are called 'primitives' or 'scalars'. These primitives are the basic building blocks for the language. These are built into the language. All other objects are made up of combinations of the primitives.

Python's Five Primitives:

- Integers
- Floating Point
- Booleans
- Strings
- Special (absence of a value)

Integers as Whole Numbers

Python integers are *real* whole numbers. E.g., 0,1,2,5,1234,-4,-1000.

Valid operations:

- Addition
- Subtraction
- Multiplication
- Division

You might have to surround negative numbers with parentheses in order to distinguish negative numbers from a subtraction operation depending on what you're doing.

Incrementing

```
x = x + 1
```

Shorthand for `x=x+1`:

```
x += 1
```

Also valid:

```
*=
-=
/=
```

Note: '1' can be replaced with any other value

Floating Point as Decimal Numbers

Python floating points are objects whose values are *decimal* numbers (e.g., 0.0,1.2,3.1415927, -23.3).

The division of two integers produces a floating point object.

Booleans as True/False Data

Booleans = `bool` type.

Two possible values:

- True
- False

Strings as Sequences of Characters

A string is a sequence of characters surrounded by quotation marks. (e.g., `'hello'`,`'we're #1'`,`'m.ss.ng??'`)

The Absence of a Value

Referred to as 'None', of type `NoneType`