Intro to Programming

2-19-2019

Sign In!

https://tinyurl.com/CADIntroPySp19-1

Python as a Starting Language

https://www.python.org/downloads/

- One of the most versatile and popular languages
  - Research - bioinformatics, NLP, scientific computing (NumPy)
  - Webapps - frameworks such as django, flask; Instagram, YouTube, Reddit
  - Machine learning, data science
- Easy-to-understand syntax and has great readability

The Scope of This Course

- Meant for people with limited or no prior programming experience
- We will cover primitive data types, lists, dictionaries, control structures, user input, and functions

Text Editors

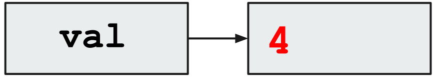
- While Python comes bundled with a text editor, we recommend using one of these two:
  - Visual Studio Code - https://code.visualstudio.com/
  - Notepad++ - https://notepad-plus-plus.org/

Running Python

- There are two ways to run your Python code:
    - Through an interpreter
        - On windows, search for “IDLE”
        - On OSX, open up terminal and run “Python”
        - Try writing **`print(“Hello World”)`** and hitting enter
    - Saving .py files and running them
        - Open up VSCode, Notepad++, or your preferred text editor
        - Write **`print(“Hello World”)`** and save the file
        - Run it from the terminal using **`python <FileName>.py`**

Variables

- Think of these as pointers to a value in memory, not as setting them equal to a value.
- EX: `val = 4`



- The variable that is named `val` points to the value `4` store in the computer's memory.

Naming Conventions

- Uses letters, numbers, and underscores
- Cannot start with a number
- No spaces

1. **BAD:** `averylongvariablename`
2. **Camel Case:** `aVeryLongVariableName`
3. **Snake Case:** `a_very_long_variable_name`

Variables, cont'd

You can also set variables to store other values. For instance, you can do:

```
x = 4
y = 3
y = x
```

The variable **`y`** now stores what the variable **`x`** stores, which is **`4`**.

Primitive Data Types

Data Types

- Integers
- Floats
- Booleans
- Strings*

Examples

- `x = 4`
- `y = 4.2`
- `is_raining = False`
- `first_name = "Calvin"`
- `last_name= 'Lin'`

You can check the type of a variable using `type(variable_name).`

* more on this next week

Primitive Data Types, cont’d

- Note that **`‘4’`** is not the same as **`4`**
- **`True`** is not the same as **`‘True’`**

- **`int_1 = 4`**
- **`string_1 = ‘4’`**
- **`bool_1 = True`**
- **`string_2 = ‘True’`**

Try printing the types of all of these variables.

You can check the type of a variable using `type(variable_name)`.

Exercises with Variables and Data Types

1. Store the value of 17 in a variable.
2. `is_raining = True`
    a. What data type is `is_raining`?
    b. How can I change this to make `is_raining` to store false?
3. `x = 101`
    a. Suppose I have a variable `y = False`. Can I set `x = y`? What is the new data type?
    b. Now suppose I have a variable `var = 17`. Can I make `x` store what `var` stores? How?

Print Statements

- In order to display something on your screen, you have to `print` that something

Syntax: `print(something)`

- `something` can be either a variable or a value

Examples

```
Input: print(4)
Output: 4

Input: is_raining = False
Input: print(is_raining)
Output: False

Input: print(“Calvin”)
Output: Calvin
```

Exercises with Print Statements

1. Print `2019` on your screen in two ways. (HINT: Use a variable!)
2. `first_name = “John”`
   a. What is the output for `print(first_name)`?
   b. Suppose I then change `first_name = “Bob”`. What is the output now?
3. `calvin = “Mike”`
   a. What is the output for `print(calvin)`?
   b. What about `print(“calvin”)`?

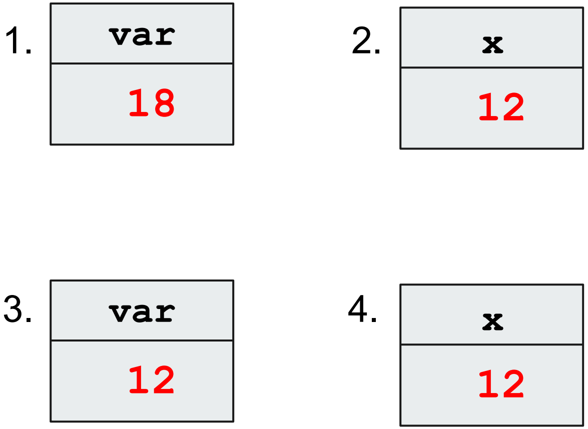
Exercises with Print Statements, cont’d

```
var = 18
x = 12
var = x
x = var
print(x)
print(var)
```

What is the output?

Explanation

```
1.  var = 18
2.  x = 12
3.  var = x
4.  x = var
print(x)
print(var)
```



Output:

```
12
12
```