

 [learn-co-curriculum](#) / [dsc-strings-floats-ints-lab](#) Public [View license](#) 0 stars  105 forks Star Watch ▾[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#) solution ▾

...

This branch is [8 commits ahead](#), [12 commits behind](#) master. Contribute ▾

hoffm386 fix objectives spacing ...

on Aug 1  13[View code](#)

Introduction to Variables: Strings and Numeric Types - Lab

Introduction

Now that we have learned about working with different types of data in Python: strings, numbers (ints and floats), and booleans, let's put that knowledge to the test. In this lab we'll imagine that we were at a nice social gathering and exchanged business cards with a few people. One of the business cards belongs to Art Vandelay, a new travel agent. We want to use our programming skills to format this information correctly.

Objectives

You will be able to:

- Apply string methods to make changes to a string

- Use concatenation to combine strings
- Perform basic mathematical operations with numeric data types

Instructions

The next morning we take out the business card, ready to format it using our programming skills, and here is what we find.



Yeah, Mr. Vandelay may not be the best at designing business cards, but like Mr. Vandelay, we know that people enter incorrect information on forms all the time.

So as Data Scientists, we often need to clean and organize data before we can make sense of it. Let's get to work.

Working on the Lab

In this lab, we'll see that there is some data already provided for us. Next to the data, we will see a comment indicating what the data should look like after we change it.

Comments are indicated in Python by a `#` followed by text. They do not get run and will not affect our code. Here we are using them to show what the intended output should be.

For example, let's say we want to capitalize all of the letters of "art vandelay". We'll see the following:

```
"art vandelay" # "ART VANDELAY"
```

Reminder: *to run the code in a jupyter notebook, press shift + enter after selecting the cell you would like to run.*

To get our output to match the comment we can use the `upper()` method as shown here:

```
"art vandelay".upper() # 'ART VANDELAY'
```

Get Going with Strings

Our first problem will be to capitalize the first letter of each word in "art vandelay". Find the correct string method to capitalize both the first and last names.

```
art_vandelay = "art vandelay".title() # 'Art Vandelay'
art_vandelay
```

Now let's turn all the letters of "Ceo" to uppercase.

```
ceo = "Ceo".upper() # 'CEO'
ceo
```

Next we will need our code to answer a question about email addresses. Every email address should end with ".com". Find the right string method to check if the email address ends with ".com" and return `True` or `False` accordingly.

```
ends_with_com = "art.vandelay@vandelay.co".endswith('.com') # False
ends_with_com
```

As you can see below, the website "vandelay.com" is not preceded by "www.". We can perform what is called string concatenation to fix this! Use the plus sign, '+', to change the website 'vandelay.com' to the string 'www.vandelay.com' by prepending 'www.'.

```
web_address = 'www.' + 'vandelay.com' # 'www.vandelay.com'
web_address
```

Working with Numbers

Finally, Mr. Vandelay gave us his phone number, but he actually has two other phone numbers that are different from the one listed on his business card. All three numbers are basically the same with the exception of the ending. Below, start by coercing the first phone number, which is currently a string, to an `int` and add one. Next do the same to the second phone number but increase it by two.

```
phone_num_one = int("7285553334") + 1 # 7285553335  
phone_num_one
```

☰ README.md

```
phone_num_two
```

Summary

In this lab, we practiced working with string methods to operate on and answer questions about strings. We wrote methods that return Booleans and changed strings to integers in order to perform addition. So much of working with data is ensuring that it is properly formatted so we can then operate on it, and in this lab, we saw how to use code to do just that.

Releases

No releases published

Packages

No packages published

Contributors 7



Languages

● Jupyter Notebook 75.0% ● Python 25.0%