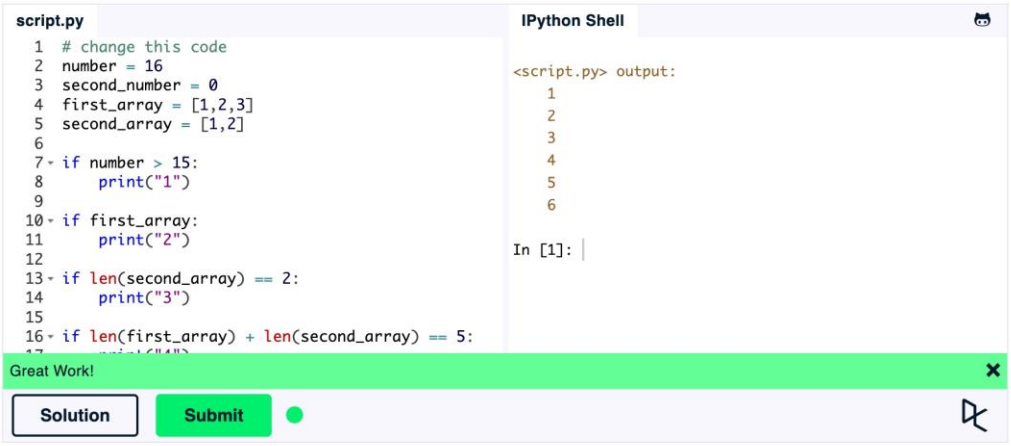
Conditions

Exercise

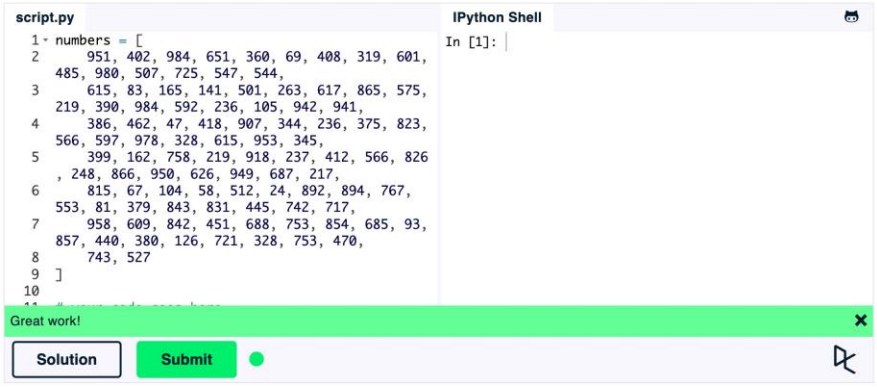
Change the variables in the first section, so that each if statement resolves as True.



Loops

Exercise

Loop through and print out all even numbers from the numbers list in the same order they are received. Don't print any numbers that come after 237 in the sequence.

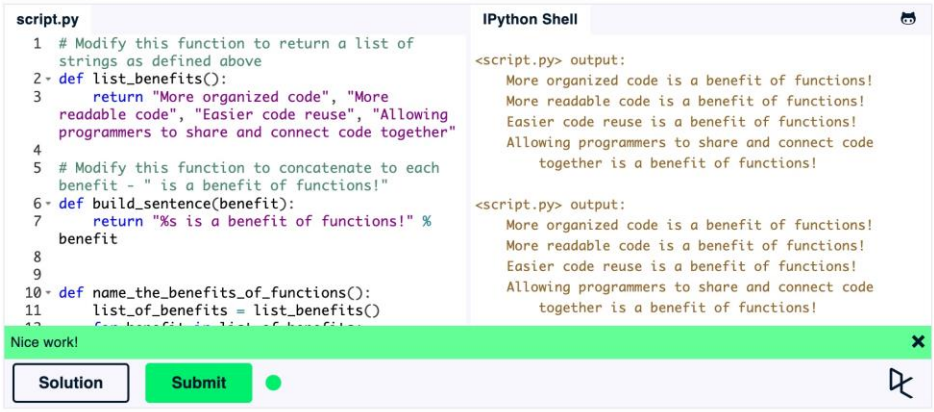


Functions

Exercise

In this exercise you'll use an existing function, and while adding your own to create a fully functional program.

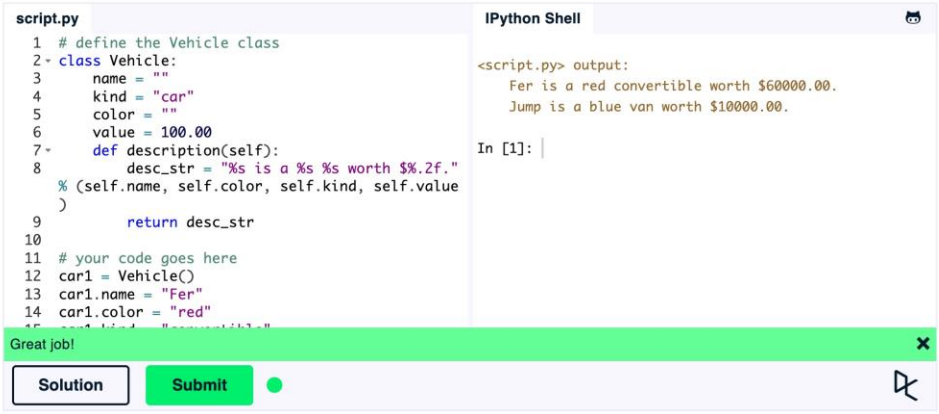
1. Add a function named `list_benefits()` that returns the following list of strings: "More organized code", "More readable code", "Easier code reuse", "Allowing programmers to share and connect code together"
2. Add a function named `build_sentence(info)` which receives a single argument containing a string and returns a sentence starting with the given string and ending with the string " is a benefit of functions!"
3. Run and see all the functions work together!



Classes and Objects

Exercise

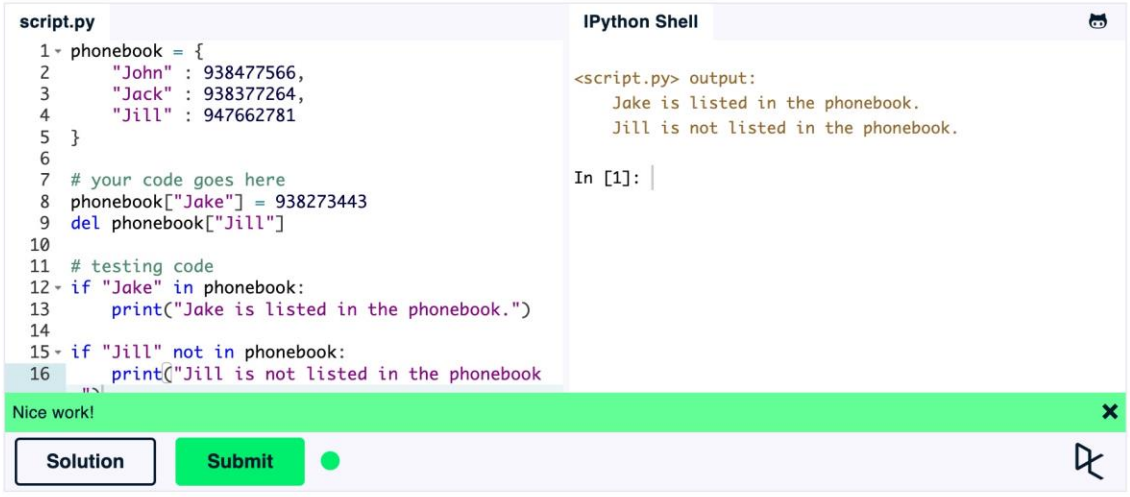
We have a class defined for vehicles. Create two new vehicles called car1 and car2. Set car1 to be a red convertible worth $60,000.00 with a name of Fer, and car2 to be a blue van named Jump worth $10,000.00.



Dictionaries

Exercise

Add "Jake" to the phonebook with the phone number 938273443, and remove Jill from the phonebook.



Modules and Packages

Exercise

In this exercise, print an alphabetically sorted list of all the functions in the `re` module containing the word `find`.