Conditions

Exercise

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

```
IPython Shell
script.py
   1 # change this code
2 number = 16
                                                                           <script.py> output:
   3 second_number = 0
4 first_array = [1,2,3]
5 second_array = [1,2]
   7 if number > 15:
8     print("1")
                                                                                4
 10 - if first_array:
11    print("2")
                                                                           In [1]:
 12
  13 - if len(second_array) == 2:
 14
15
            print("3")
  16 - if len(first_array) + len(second_array) == 5:
Great Work!
                                                                                                                                              R
    Solution
                          Submit
```

Loops

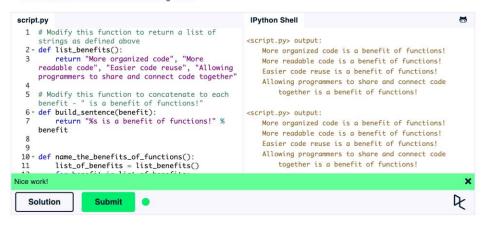
Exercise

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.

Exercise

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

- 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 <u>build_sentence(info)</u> 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.

```
script.py
                                                                              IPython Shell
       # define the Vehicle class
   2 - class Vehicle:
                                                                             <script.py> output:
            name = ""
kind = "car"
color = ""
                                                                                  Fer is a red convertible worth $60000.00.
                                                                                  Jump is a blue van worth $10000.00.
             value = 100.00
      def description(self):
    desc_str = "%s is a %s %s worth $%.2f."
% (self.name, self.color, self.kind, self.value)
                                                                             In [1]:
                  return desc_str
 10
 11
12
     # your code goes here
car1 = Vehicle()
 13
      car1.name = "Fer"
car1.color = "red"
 14
                                                                                                                                                  R
   Solution
                          Submit
```

Exercise

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

```
IPython Shell
  1 + phonebook = {
2     "John" : 938477566,
3     "Jack" : 938377264,
                                                                            <script.py> output:
                                                                                Jake is listed in the phonebook.
             "Jill" : 947662781
                                                                                Jill is not listed in the phonebook.
  5 }
  6
  # your code goes here
phonebook["Jake"] = 938273443
del phonebook["Jill"]
                                                                           In [1]:
 10
 11 # testing code

12 · if "Jake" in phonebook:

13 print("Jake is listed in the phonebook.")
 14
 15 - if "Jill" not in phonebook:
            print("Jill is not listed in the phonebook
Nice work!
    Solution
                          Submit
```

Modules and Packages

Exercise

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

