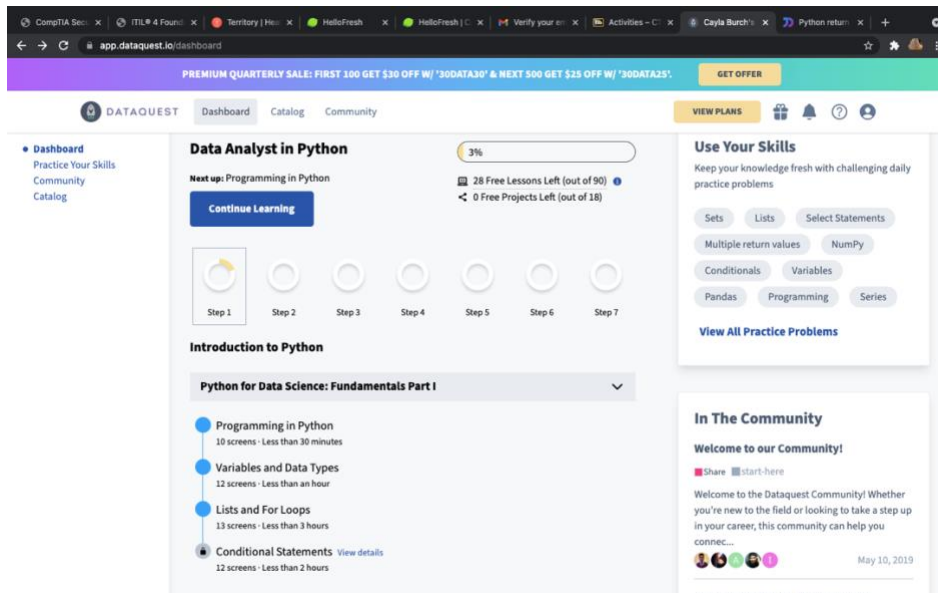


Cayla Burch
12 September 2021
CTEC 298
Dr. Bemley

DataQuest Tutorials

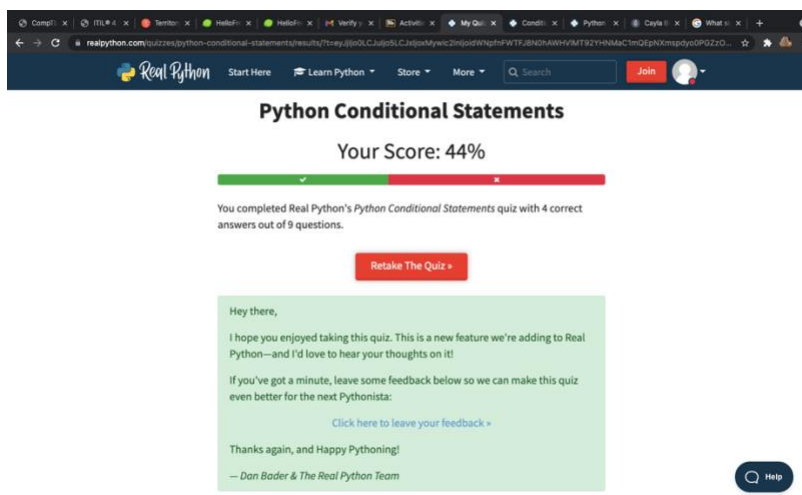
Python Missions:

1. Programming in Python, Variables & Data Types, and Lists & For Loops



Additional Tutorials:

2. Conditional Statements



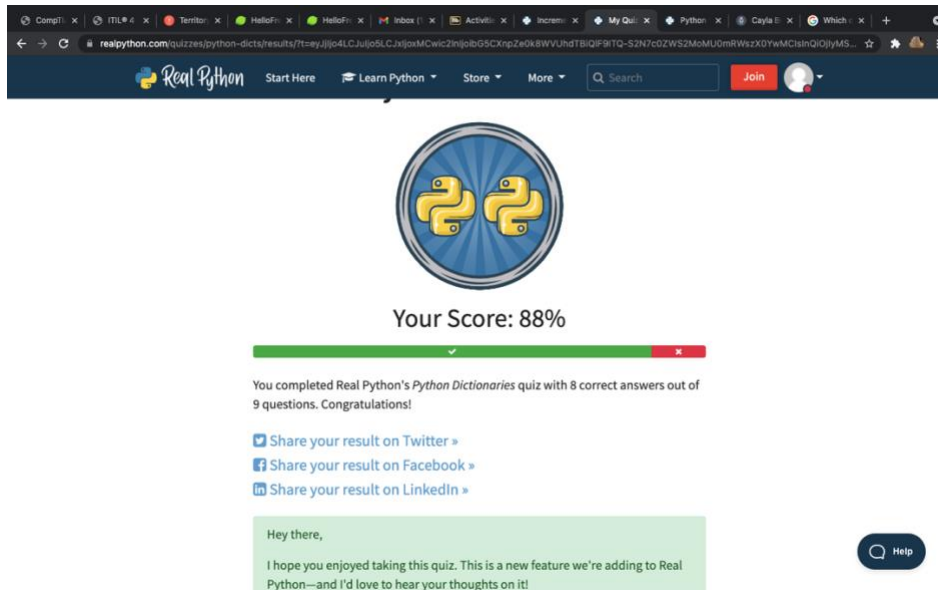
Cayla Burch
12 September 2021
CTEC 298
Dr. Bemley

3. While Loops



The screenshot shows the Real Python website with the title "Python 'while' Loops". Below the title is a large Python logo. The score is displayed as "Your Score: 55%". A progress bar shows 5 correct answers (green) and 4 incorrect answers (red). The text states: "You completed Real Python's Python 'while' Loops quiz with 5 correct answers out of 9 questions." Below this are three social sharing links: "Share your result on Twitter", "Share your result on Facebook", and "Share your result on LinkedIn". At the bottom, there is a green box with the text "Hey there," and a "Help" button.

4. Dictionaries and Frequency Tables



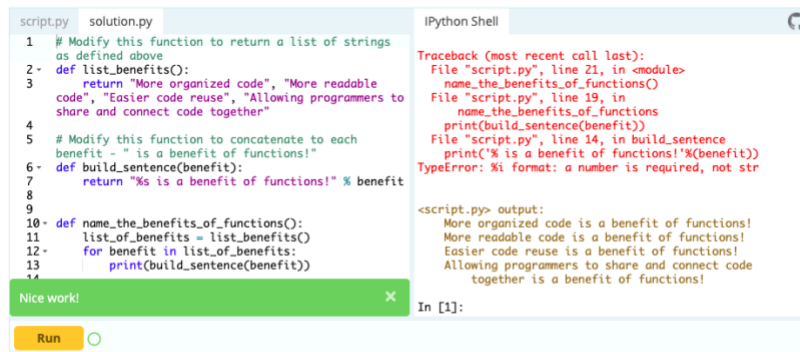
The screenshot shows the Real Python website with the title "Python Dictionaries". Below the title is a large Python logo. The score is displayed as "Your Score: 88%". A progress bar shows 8 correct answers (green) and 1 incorrect answer (red). The text states: "You completed Real Python's Python Dictionaries quiz with 8 correct answers out of 9 questions. Congratulations!" Below this are three social sharing links: "Share your result on Twitter", "Share your result on Facebook", and "Share your result on LinkedIn". At the bottom, there is a green box with the text "Hey there, I hope you enjoyed taking this quiz. This is a new feature we're adding to Real Python—and I'd love to hear your thoughts on it!" and a "Help" button.

5. Functions and Fundamentals

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!



The screenshot shows a Jupyter Notebook with two main panels. The left panel, titled 'script.py', contains the following Python code:

```
1 # Modify this function to return a list of strings
  as defined above
2- def list_benefits():
3     return "More organized code", "More readable
  code", "Easier code reuse", "Allowing programmers to
  share and connect code together"
4
5 # Modify this function to concatenate to each
  benefit - " is a benefit of functions!"
6- def build_sentence(benefit):
7     return "%s is a benefit of functions!" % benefit
8
9
10- def name_the_benefits_of_functions():
11     list_of_benefits = list_benefits()
12-     for benefit in list_of_benefits:
13         print(build_sentence(benefit))
14
```

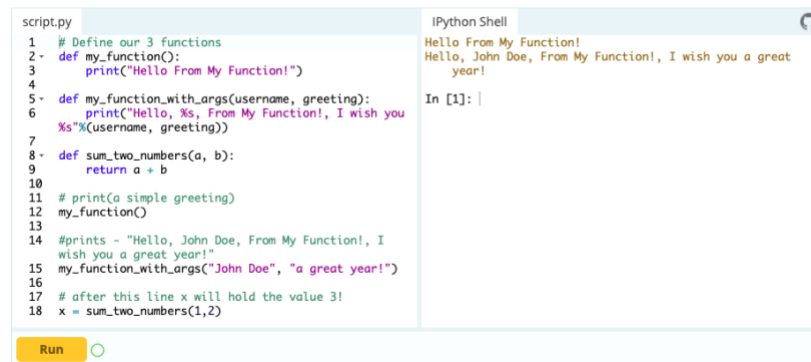
The right panel, titled 'IPython Shell', shows the output of the code execution. It displays a traceback for a `TypeError` that occurred on line 21, indicating that the `%i` format specifier requires a number, but a string was provided. Below the traceback, the output of the `name_the_benefits_of_functions()` function is shown, which prints four lines of text, each preceded by a space and followed by " is a benefit of functions!".

At the bottom of the notebook, there is a green status bar that says "Nice work!" and a yellow "Run" button.

Powered by DataCamp

How do you call functions in Python?

Simply write the function's name followed by `()`, placing any required arguments within the brackets. For example, let's call the functions written above (in the previous example):



The screenshot shows a Jupyter Notebook with two main panels. The left panel, titled 'script.py', contains the following Python code:

```
1 # Define our 3 functions
2- def my_function():
3     print("Hello From My Function!")
4
5- def my_function_with_args(username, greeting):
6     print("Hello, %s, From My Function!, I wish you
  %s"%(username, greeting))
7
8- def sum_two_numbers(a, b):
9     return a + b
10
11 # print(a simple greeting)
12 my_function()
13
14 #prints - "Hello, John Doe, From My Function!, I
  wish you a great year!"
15 my_function_with_args("John Doe", "a great year!")
16
17 # after this line x will hold the value 3!
18 x = sum_two_numbers(1,2)
```

The right panel, titled 'IPython Shell', shows the output of the code execution. It displays three lines of text: "Hello From My Function!", "Hello, John Doe, From My Function!, I wish you a great year!", and "In [1]: |".

At the bottom of the notebook, there is a yellow status bar that says "Run" and a yellow "Run" button.

Powered by DataCamp