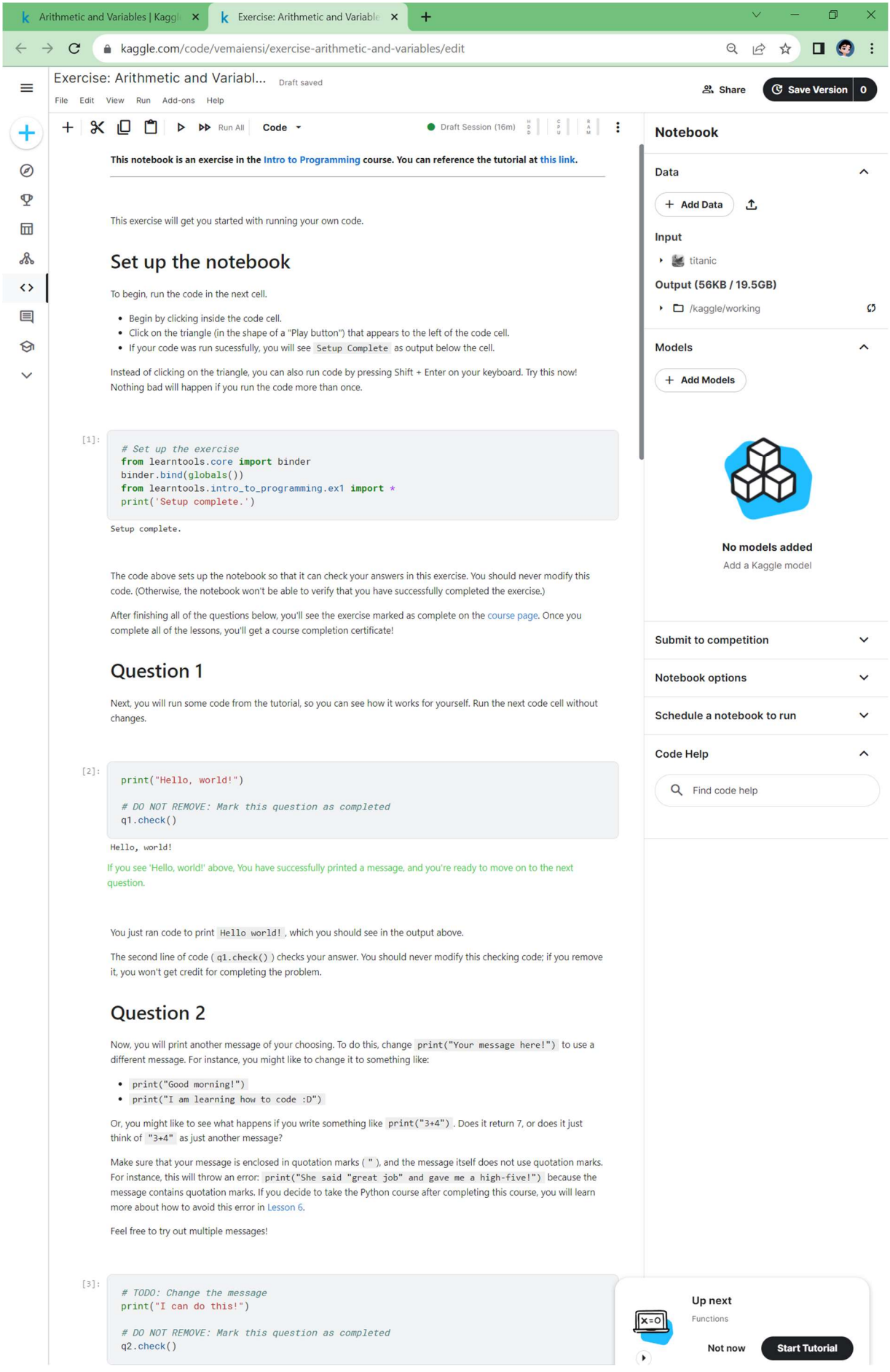
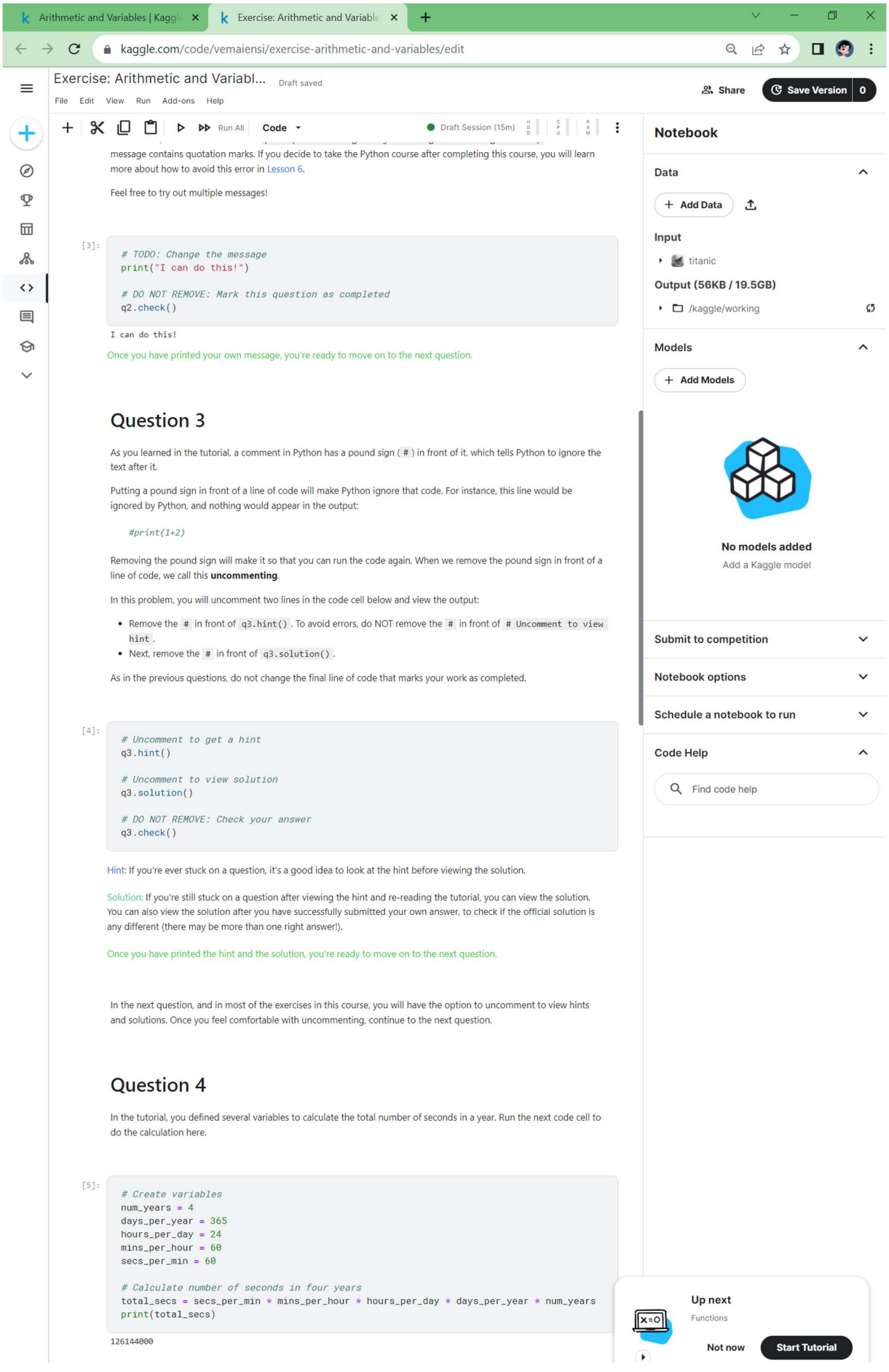
CCSEL1-18 Professional ELective

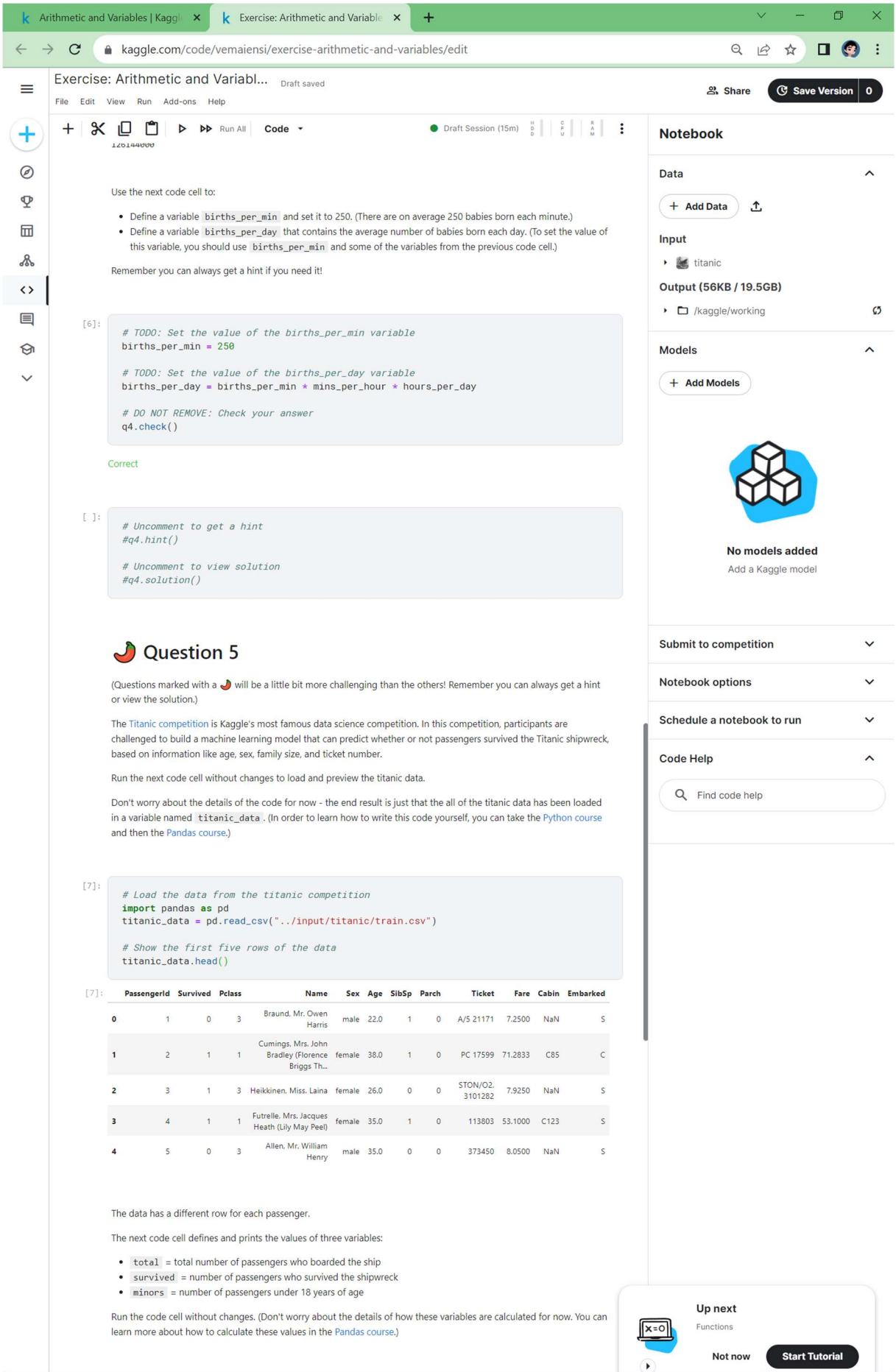
# [ Intro to Programming ]

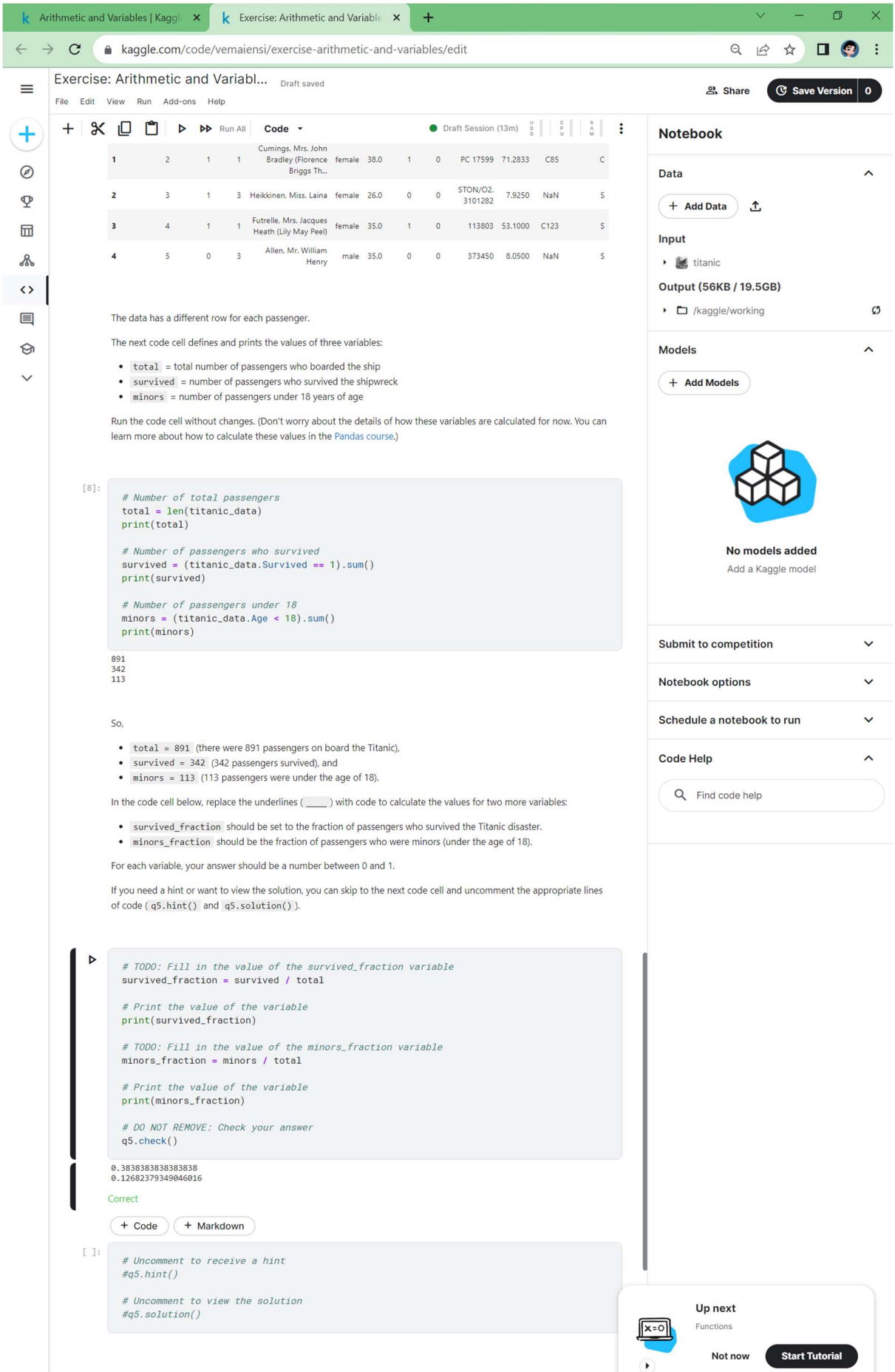
**MARASIGAN, VEM AIENSI A.**
3BSCS-1

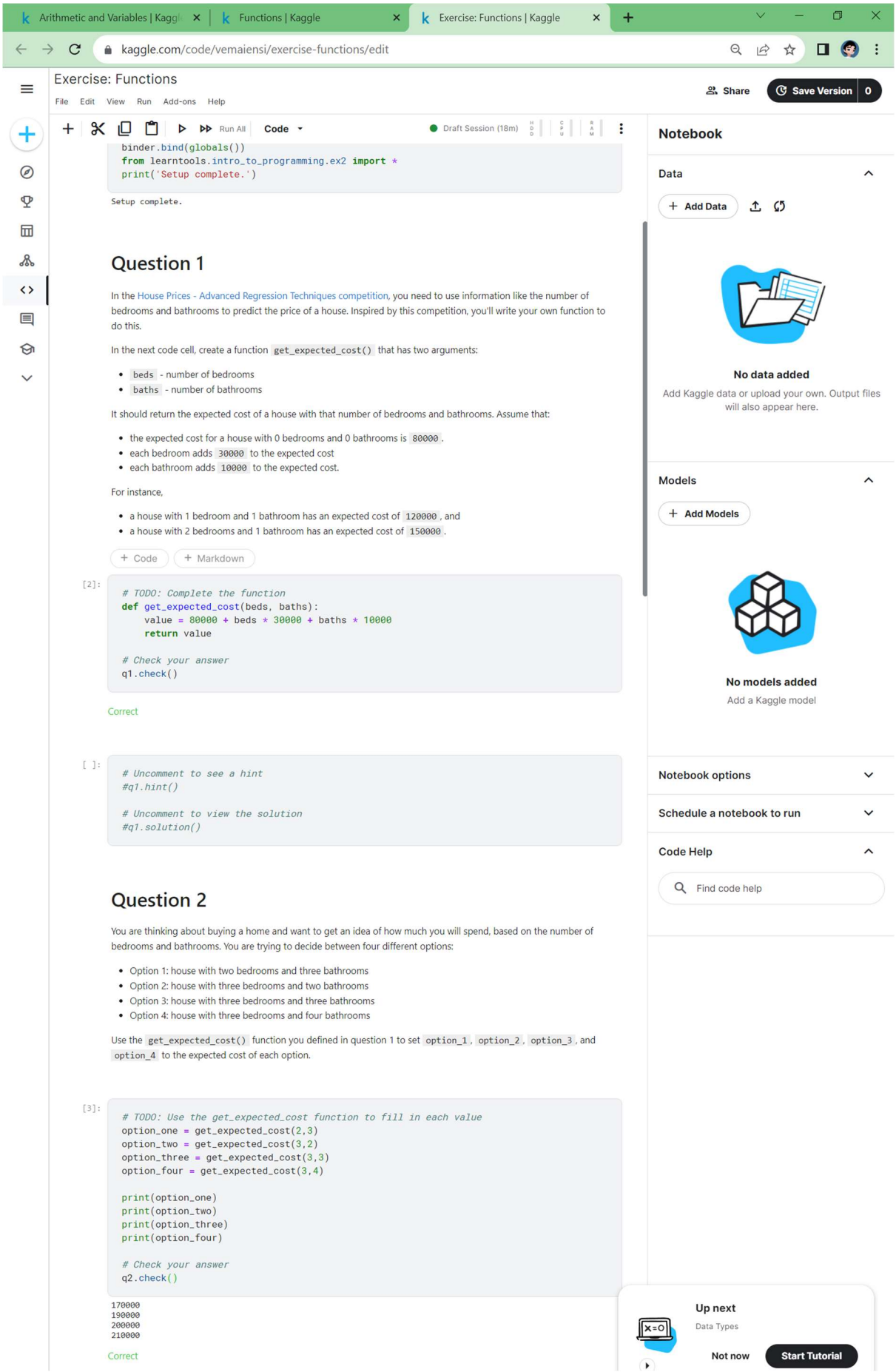
**Documentation**
November 10, 2023

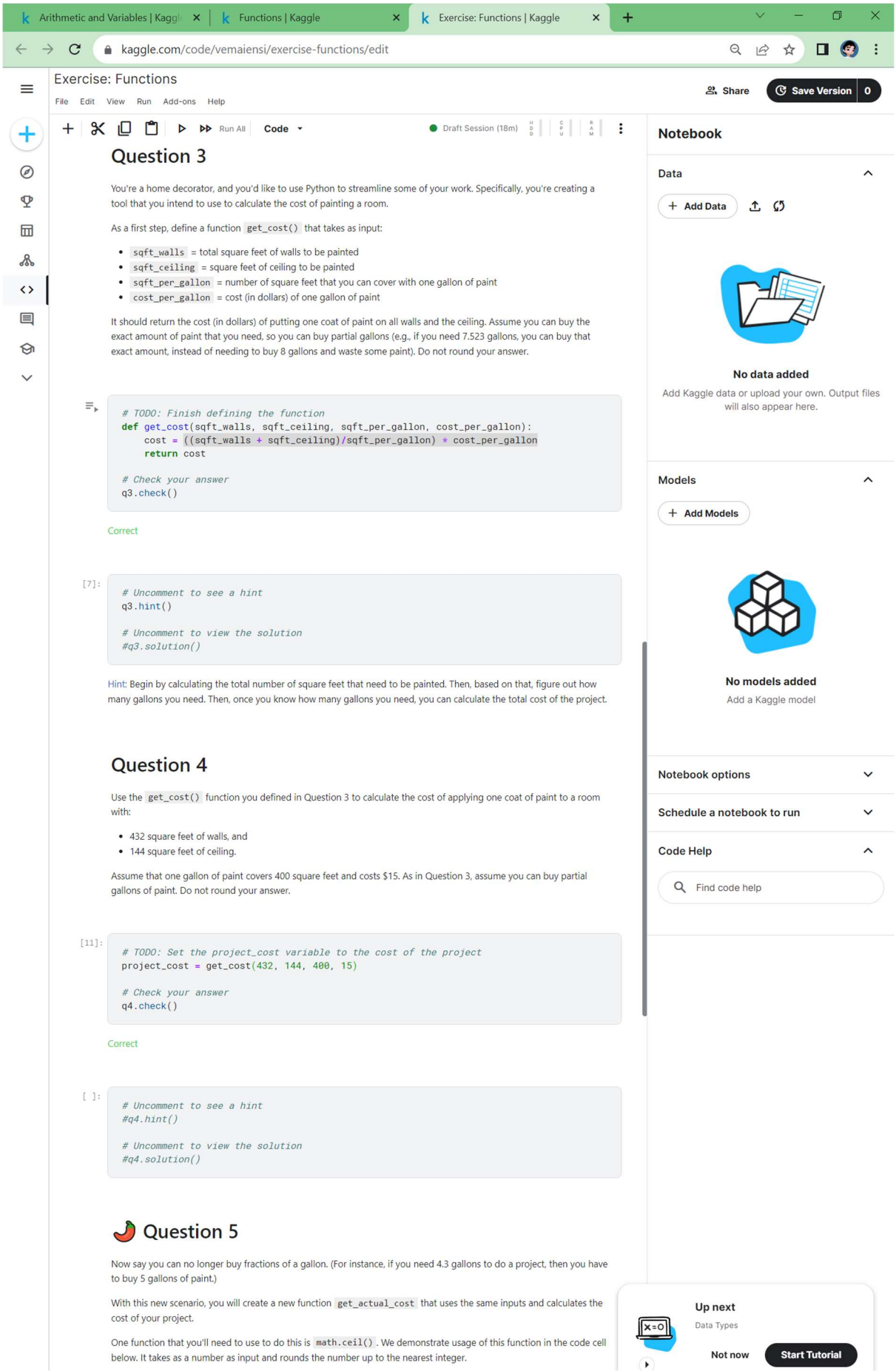


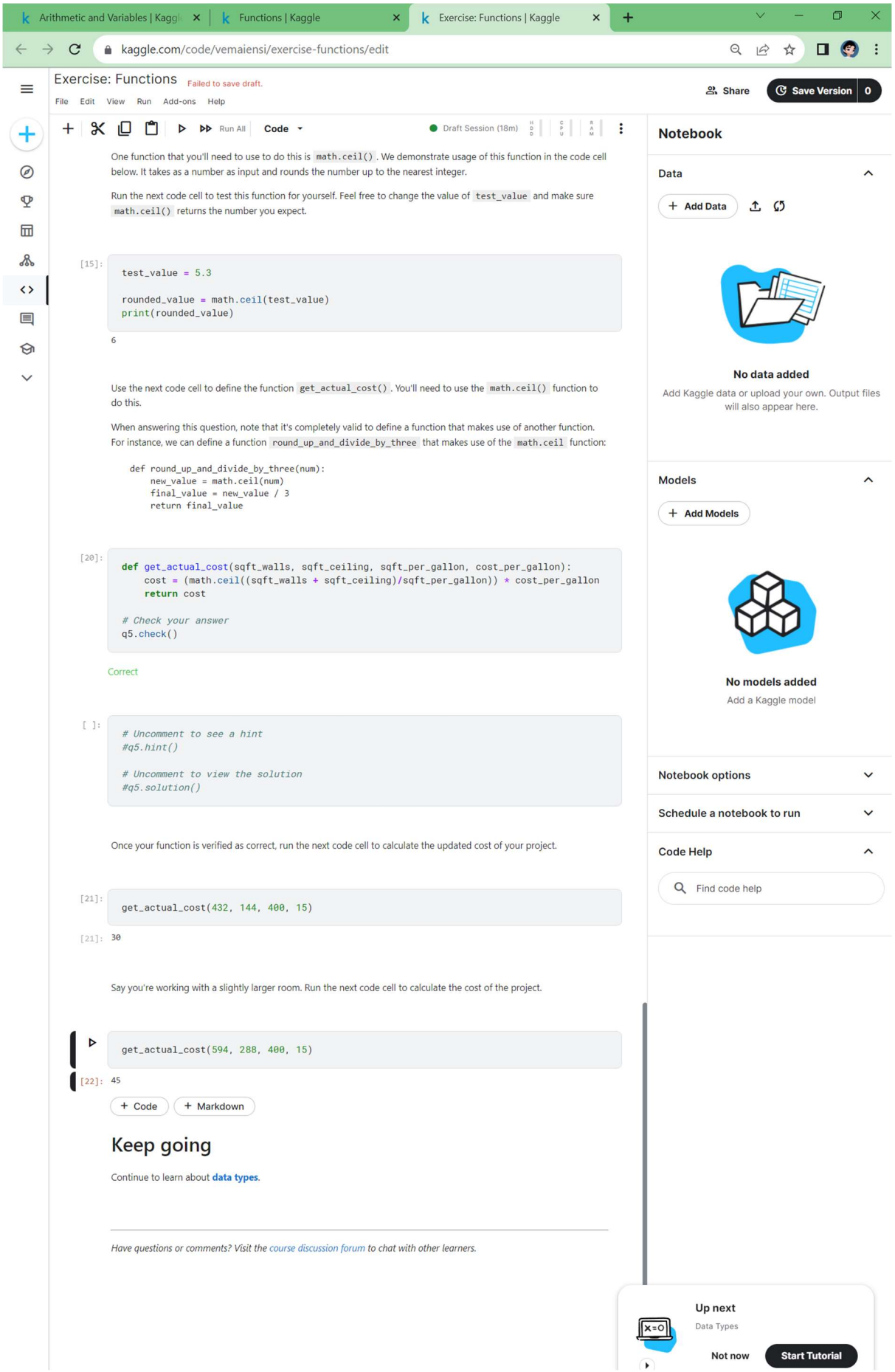

Arithmetic and Variables | Kaggle
Exercise: Arithmetic and Variable
kaggle.com/code/vemaiensi/exercise-arithmetic-and-variables/edit
Exercise: Arithmetic and Variabl...
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (15m)
message contains quotation marks. If you decide to take the Python course after completing this course, you will learn more about how to avoid this error in Lesson 6.
Feel free to try out multiple messages!
[3]:
# TODO: Change the message
print("I can do this!")
# DO NOT REMOVE: Mark this question as completed
q2.check()
I can do this!
Once you have printed your own message, you're ready to move on to the next question.
Question 3
As you learned in the tutorial, a comment in Python has a pound sign (#) in front of it, which tells Python to ignore the text after it.
Putting a pound sign in front of a line of code will make Python ignore that code. For instance, this line would be ignored by Python, and nothing would appear in the output:
#print(1+2)
Removing the pound sign will make it so that you can run the code again. When we remove the pound sign in front of a line of code, we call this uncommenting.
In this problem, you will uncomment two lines in the code cell below and view the output:
Remove the # in front of q3.hint(). To avoid errors, do NOT remove the # in front of # Uncomment to view hint.
Next, remove the # in front of q3.solution().
As in the previous questions, do not change the final line of code that marks your work as completed.
[4]:
# Uncomment to get a hint
q3.hint()
# Uncomment to view solution
q3.solution()
# DO NOT REMOVE: Check your answer
q3.check()
Hint: If you're ever stuck on a question, it's a good idea to look at the hint before viewing the solution.
Solution: If you're still stuck on a question after viewing the hint and re-reading the tutorial, you can view the solution. You can also view the solution after you have successfully submitted your own answer, to check if the official solution is any different (there may be more than one right answer!).
Once you have printed the hint and the solution, you're ready to move on to the next question.
In the next question, and in most of the exercises in this course, you will have the option to uncomment to view hints and solutions. Once you feel comfortable with uncommenting, continue to the next question.
Question 4
In the tutorial, you defined several variables to calculate the total number of seconds in a year. Run the next code cell to do the calculation here.
[5]:
# Create variables
num_years = 4
days_per_year = 365
hours_per_day = 24
mins_per_hour = 60
secs_per_min = 60
# Calculate number of seconds in four years
total_secs = secs_per_min * mins_per_hour * hours_per_day * days_per_year * num_years
print(total_secs)
126144000
Notebook
Data
Add Data
Input
titanic
Output (56KB / 19.5GB)
/kaggle/working
Models
Add Models
No models added
Add a Kaggle model
Submit to competition
Notebook options
Schedule a notebook to run
Code Help
Find code help
Up next
Functions
Not now
Start Tutorial

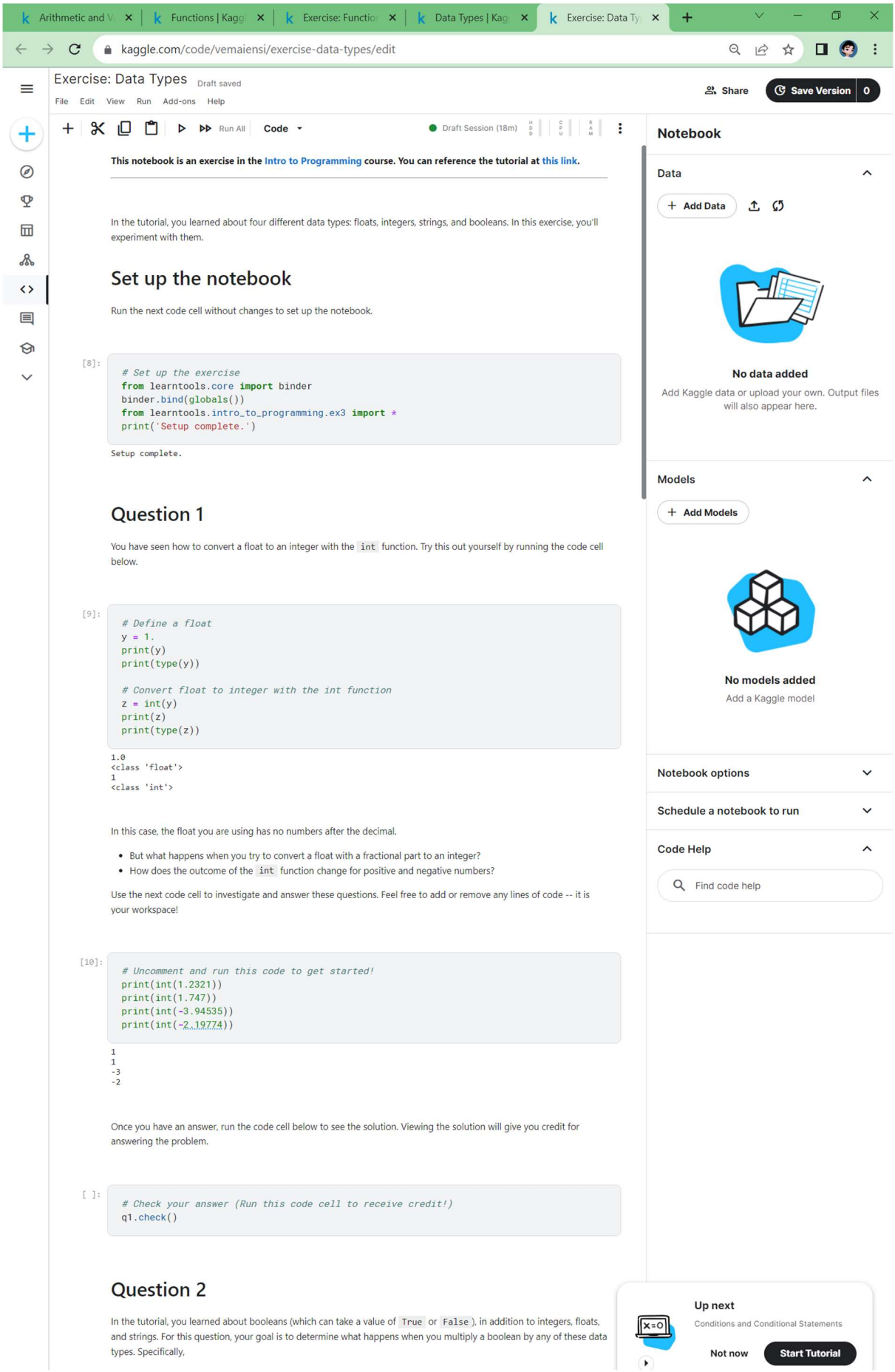

Arithmetic and Variables | Kaggle
Exercise: Arithmetic and Variable
kaggle.com/code/vemaiensi/exercise-arithmetic-and-variables/edit
Exercise: Arithmetic and Variabl...
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (15m)
Use the next code cell to:
Define a variable births_per_min and set it to 250. (There are on average 250 babies born each minute.)
Define a variable births_per_day that contains the average number of babies born each day. (To set the value of this variable, you should use births_per_min and some of the variables from the previous code cell.)
Remember you can always get a hint if you need it!
[6]:
# TODO: Set the value of the births_per_min variable
births_per_min = 250
# TODO: Set the value of the births_per_day variable
births_per_day = births_per_min * mins_per_hour * hours_per_day
# DO NOT REMOVE: Check your answer
q4.check()
Correct
[ ]:
# Uncomment to get a hint
#q4.hint()
# Uncomment to view solution
#q4.solution()
Question 5
(Questions marked with a will be a little bit more challenging than the others! Remember you can always get a hint or view the solution.)
The Titanic competition is Kaggle's most famous data science competition. In this competition, participants are challenged to build a machine learning model that can predict whether or not passengers survived the Titanic shipwreck, based on information like age, sex, family size, and ticket number.
Run the next code cell without changes to load and preview the titanic data.
Don't worry about the details of the code for now - the end result is just that the all of the titanic data has been loaded in a variable named titanic_data. (In order to learn how to write this code yourself, you can take the Python course and then the Pandas course.)
[7]:
# Load the data from the titanic competition
import pandas as pd
titanic_data = pd.read_csv("../input/titanic/train.csv")
# Show the first five rows of the data
titanic_data.head()
[7]:
PassengerId Survived Pclass Name Sex Age SibSp Parch Ticket Fare Cabin Embarked
0 1 0 3 Braund, Mr. Owen Harris male 22.0 1 0 A/5 21171 7.2500 NaN S
1 2 1 1 Cumings, Mrs. John Bradley (Florence Briggs Th... female 38.0 1 0 PC 17599 71.2833 C85 C
2 3 1 3 Heikkinen, Miss. Laina female 26.0 0 0 STON/O2. 3101282 7.9250 NaN S
3 4 1 1 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0 1 0 113803 53.1000 C123 S
4 5 0 3 Allen, Mr. William Henry male 35.0 0 0 373450 8.0500 NaN S
The data has a different row for each passenger.
The next code cell defines and prints the values of three variables:
total = total number of passengers who boarded the ship
survived = number of passengers who survived the shipwreck
minors = number of passengers under 18 years of age
Run the code cell without changes. (Don't worry about the details of how these variables are calculated for now. You can learn more about how to calculate these values in the Pandas course.)
Notebook
Data
Add Data
Input
titanic
Output (56KB / 19.5GB)
/kaggle/working
Models
Add Models
No models added
Add a Kaggle model
Submit to competition
Notebook options
Schedule a notebook to run
Code Help
Find code help
Up next
Functions
Not now
Start Tutorial


Arithmetic and Variables | Kaggle
Exercise: Arithmetic and Variable
kaggle.com/code/vemaiensi/exercise-arithmetic-and-variables/edit
Exercise: Arithmetic and Variabl...
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (13m)
1 2 1 1 Cumings, Mrs. John Bradley (Florence Briggs Th... female 38.0 1 0 PC 17599 71.2833 C85 C
2 3 1 3 Heikkinen, Miss. Laina female 26.0 0 0 STON/O2. 3101282 7.9250 NaN S
3 4 1 1 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0 1 0 113803 53.1000 C123 S
4 5 0 3 Allen, Mr. William Henry male 35.0 0 0 373450 8.0500 NaN S
The data has a different row for each passenger.
The next code cell defines and prints the values of three variables:
total = total number of passengers who boarded the ship
survived = number of passengers who survived the shipwreck
minors = number of passengers under 18 years of age
Run the code cell without changes. (Don't worry about the details of how these variables are calculated for now. You can learn more about how to calculate these values in the Pandas course.)
[8]:
# Number of total passengers
total = len(titanic_data)
print(total)
# Number of passengers who survived
survived = (titanic_data.Survived == 1).sum()
print(survived)
# Number of passengers under 18
minors = (titanic_data.Age < 18).sum()
print(minors)
891
342
113
So,
total = 891 (there were 891 passengers on board the Titanic),
survived = 342 (342 passengers survived), and
minors = 113 (113 passengers were under the age of 18).
In the code cell below, replace the underlines (____) with code to calculate the values for two more variables:
survived_fraction should be set to the fraction of passengers who survived the Titanic disaster.
minors_fraction should be the fraction of passengers who were minors (under the age of 18).
For each variable, your answer should be a number between 0 and 1.
If you need a hint or want to view the solution, you can skip to the next code cell and uncomment the appropriate lines of code (q5.hint() and q5.solution()).
# TODO: Fill in the value of the survived_fraction variable
survived_fraction = survived / total
# Print the value of the variable
print(survived_fraction)
# TODO: Fill in the value of the minors_fraction variable
minors_fraction = minors / total
# Print the value of the variable
print(minors_fraction)
# DO NOT REMOVE: Check your answer
q5.check()
0.3838383838383838
0.12682379349046016
Correct
+ Code + Markdown
[ ]:
# Uncomment to receive a hint
#q5.hint()
# Uncomment to view the solution
#q5.solution()
Notebook
Data
+ Add Data
Input
titanic
Output (56KB / 19.5GB)
/kaggle/working
Models
+ Add Models
No models added
Add a Kaggle model
Submit to competition
Notebook options
Schedule a notebook to run
Code Help
Find code help
Up next
Functions
Not now
Start Tutorial


Arithmetic and Variables | Kaggl
Functions | Kaggle
Exercise: Functions | Kaggle
kaggle.com/code/vemaiensi/exercise-functions/edit
Exercise: Functions
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (18m)
binder.bind(globals())
from learntools.intro_to_programming.ex2 import *
print('Setup complete.')
Setup complete.
Question 1
In the House Prices - Advanced Regression Techniques competition, you need to use information like the number of bedrooms and bathrooms to predict the price of a house. Inspired by this competition, you'll write your own function to do this.
In the next code cell, create a function get_expected_cost() that has two arguments:
beds - number of bedrooms
baths - number of bathrooms
It should return the expected cost of a house with that number of bedrooms and bathrooms. Assume that:
the expected cost for a house with 0 bedrooms and 0 bathrooms is 80000.
each bedroom adds 30000 to the expected cost
each bathroom adds 10000 to the expected cost.
For instance,
a house with 1 bedroom and 1 bathroom has an expected cost of 120000, and
a house with 2 bedrooms and 1 bathroom has an expected cost of 150000.
+ Code
+ Markdown
[2]:
# TODO: Complete the function
def get_expected_cost(beds, baths):
value = 80000 + beds * 30000 + baths * 10000
return value
# Check your answer
q1.check()
Correct
[ ]:
# Uncomment to see a hint
#q1.hint()
# Uncomment to view the solution
#q1.solution()
Question 2
You are thinking about buying a home and want to get an idea of how much you will spend, based on the number of bedrooms and bathrooms. You are trying to decide between four different options:
Option 1: house with two bedrooms and three bathrooms
Option 2: house with three bedrooms and two bathrooms
Option 3: house with three bedrooms and three bathrooms
Option 4: house with three bedrooms and four bathrooms
Use the get_expected_cost() function you defined in question 1 to set option_1, option_2, option_3, and option_4 to the expected cost of each option.
[3]:
# TODO: Use the get_expected_cost function to fill in each value
option_one = get_expected_cost(2,3)
option_two = get_expected_cost(3,2)
option_three = get_expected_cost(3,3)
option_four = get_expected_cost(3,4)
print(option_one)
print(option_two)
print(option_three)
print(option_four)
# Check your answer
q2.check()
170000
190000
200000
210000
Correct
Notebook
Data
+ Add Data
No data added
Add Kaggle data or upload your own. Output files will also appear here.
Models
+ Add Models
No models added
Add a Kaggle model
Notebook options
Schedule a notebook to run
Code Help
Find code help
Up next
Data Types
Not now
Start Tutorial


Arithmetic and Variables | Kaggl
Functions | Kaggle
Exercise: Functions | Kaggle
kaggle.com/code/vemaiensi/exercise-functions/edit
Exercise: Functions
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (18m)
Question 3
You're a home decorator, and you'd like to use Python to streamline some of your work. Specifically, you're creating a tool that you intend to use to calculate the cost of painting a room.
As a first step, define a function get_cost() that takes as input:
sqft_walls = total square feet of walls to be painted
sqft_ceiling = square feet of ceiling to be painted
sqft_per_gallon = number of square feet that you can cover with one gallon of paint
cost_per_gallon = cost (in dollars) of one gallon of paint
It should return the cost (in dollars) of putting one coat of paint on all walls and the ceiling. Assume you can buy the exact amount of paint that you need, so you can buy partial gallons (e.g., if you need 7.523 gallons, you can buy that exact amount, instead of needing to buy 8 gallons and waste some paint). Do not round your answer.
# TODO: Finish defining the function
def get_cost(sqft_walls, sqft_ceiling, sqft_per_gallon, cost_per_gallon):
cost = ((sqft_walls + sqft_ceiling)/sqft_per_gallon) * cost_per_gallon
return cost
# Check your answer
q3.check()
Correct
[7]:
# Uncomment to see a hint
q3.hint()
# Uncomment to view the solution
#q3.solution()
Hint: Begin by calculating the total number of square feet that need to be painted. Then, based on that, figure out how many gallons you need. Then, once you know how many gallons you need, you can calculate the total cost of the project.
Question 4
Use the get_cost() function you defined in Question 3 to calculate the cost of applying one coat of paint to a room with:
432 square feet of walls, and
144 square feet of ceiling.
Assume that one gallon of paint covers 400 square feet and costs $15. As in Question 3, assume you can buy partial gallons of paint. Do not round your answer.
[11]:
# TODO: Set the project_cost variable to the cost of the project
project_cost = get_cost(432, 144, 400, 15)
# Check your answer
q4.check()
Correct
[ ]:
# Uncomment to see a hint
#q4.hint()
# Uncomment to view the solution
#q4.solution()
Question 5
Now say you can no longer buy fractions of a gallon. (For instance, if you need 4.3 gallons to do a project, then you have to buy 5 gallons of paint.)
With this new scenario, you will create a new function get_actual_cost that uses the same inputs and calculates the cost of your project.
One function that you'll need to use to do this is math.ceil(). We demonstrate usage of this function in the code cell below. It takes as a number as input and rounds the number up to the nearest integer.
Notebook
Data
Add Data
No data added
Add Kaggle data or upload your own. Output files will also appear here.
Models
Add Models
No models added
Add a Kaggle model
Notebook options
Schedule a notebook to run
Code Help
Find code help
Up next
Data Types
Not now
Start Tutorial


Arithmetic and Variables | Kaggl
Functions | Kaggle
Exercise: Functions | Kaggle
kaggle.com/code/vemaiensi/exercise-functions/edit
Exercise: Functions
Failed to save draft.
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (18m)
One function that you'll need to use to do this is math.ceil(). We demonstrate usage of this function in the code cell below. It takes as a number as input and rounds the number up to the nearest integer.
Run the next code cell to test this function for yourself. Feel free to change the value of test_value and make sure math.ceil() returns the number you expect.
[15]:
test_value = 5.3
rounded_value = math.ceil(test_value)
print(rounded_value)
6
Use the next code cell to define the function get_actual_cost(). You'll need to use the math.ceil() function to do this.
When answering this question, note that it's completely valid to define a function that makes use of another function. For instance, we can define a function round_up_and_divide_by_three that makes use of the math.ceil function:
def round_up_and_divide_by_three(num):
new_value = math.ceil(num)
final_value = new_value / 3
return final_value
[20]:
def get_actual_cost(sqft_walls, sqft_ceiling, sqft_per_gallon, cost_per_gallon):
cost = (math.ceil((sqft_walls + sqft_ceiling)/sqft_per_gallon)) * cost_per_gallon
return cost
# Check your answer
q5.check()
Correct
[ ]:
# Uncomment to see a hint
#q5.hint()
# Uncomment to view the solution
#q5.solution()
Once your function is verified as correct, run the next code cell to calculate the updated cost of your project.
[21]:
get_actual_cost(432, 144, 400, 15)
[21]: 30
Say you're working with a slightly larger room. Run the next code cell to calculate the cost of the project.
get_actual_cost(594, 288, 400, 15)
[22]: 45
+ Code
+ Markdown
Keep going
Continue to learn about data types.
Have questions or comments? Visit the course discussion forum to chat with other learners.
Notebook
Data
+ Add Data
No data added
Add Kaggle data or upload your own. Output files will also appear here.
Models
+ Add Models
No models added
Add a Kaggle model
Notebook options
Schedule a notebook to run
Code Help
Find code help
Up next
Data Types
Not now
Start Tutorial



Arithmetic and V | Functions | Kaggl | Exercise: Functio | Data Types | Kag | Exercise: Data Ty

kaggle.com/code/vemaiensi/exercise-data-types/edit

Exercise: Data Types Draft saved

File Edit View Run Add-ons Help

Share Save Version 0

Run All Code Draft Session (18m)

**This notebook is an exercise in the Intro to Programming course. You can reference the tutorial at this link.**

In the tutorial, you learned about four different data types: floats, integers, strings, and booleans. In this exercise, you'll experiment with them.

## Set up the notebook

Run the next code cell without changes to set up the notebook.

[8]:
```
# Set up the exercise
from learntools.core import binder
binder.bind(globals())
from learntools.intro_to_programming.ex3 import *
print('Setup complete.')
```

```
Setup complete.
```

## Question 1

You have seen how to convert a float to an integer with the `int` function. Try this out yourself by running the code cell below.

[9]:
```
# Define a float
y = 1.
print(y)
print(type(y))

# Convert float to integer with the int function
z = int(y)
print(z)
print(type(z))
```

```
1.0
<class 'float'>
1
<class 'int'>
```

In this case, the float you are using has no numbers after the decimal.

- But what happens when you try to convert a float with a fractional part to an integer?
- How does the outcome of the `int` function change for positive and negative numbers?

Use the next code cell to investigate and answer these questions. Feel free to add or remove any lines of code -- it is your workspace!

[10]:
```
# Uncomment and run this code to get started!
print(int(1.2321))
print(int(1.747))
print(int(-3.94535))
print(int(-2.19774))
```

```
1
1
-3
-2
```

Once you have an answer, run the code cell below to see the solution. Viewing the solution will give you credit for answering the problem.

[ ]:
```
# Check your answer (Run this code cell to receive credit!)
q1.check()
```

## Question 2

In the tutorial, you learned about booleans (which can take a value of `True` or `False`), in addition to integers, floats, and strings. For this question, your goal is to determine what happens when you multiply a boolean by any of these data types. Specifically,

Notebook

Data

Add Data

No data added

Add Kaggle data or upload your own. Output files will also appear here.

Models

Add Models

No models added

Add a Kaggle model

Notebook options

Schedule a notebook to run

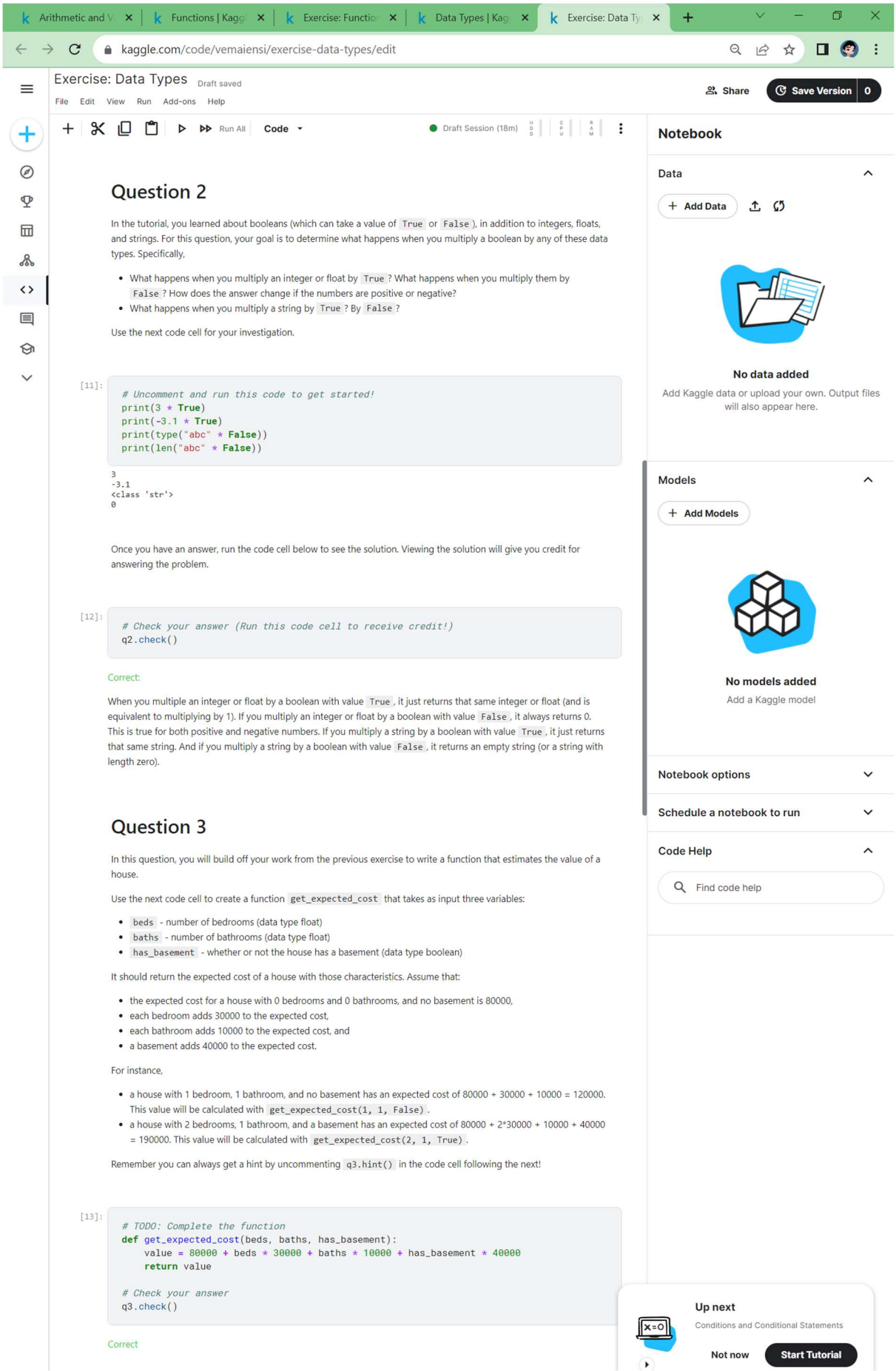
Code Help

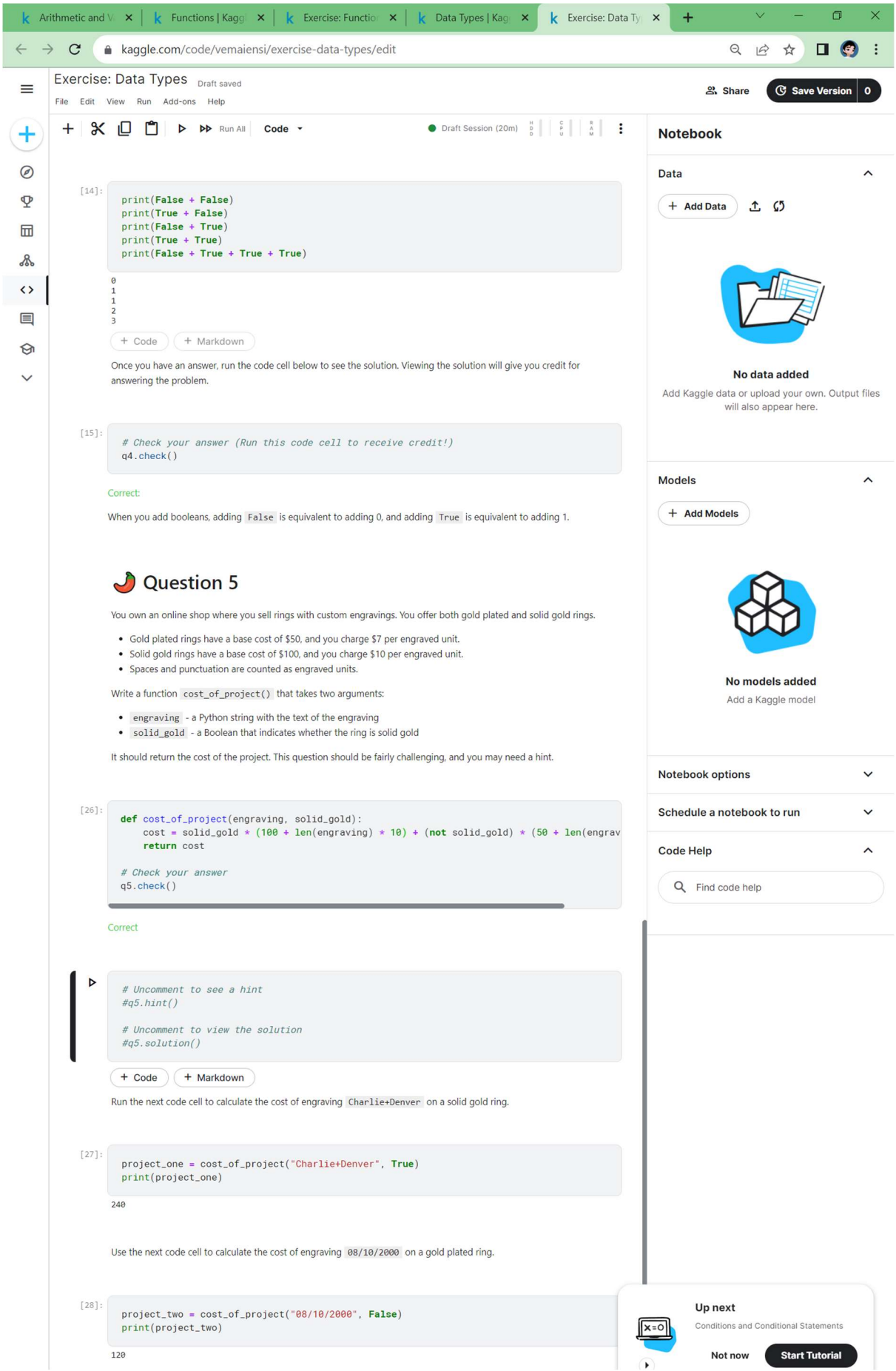
Find code help

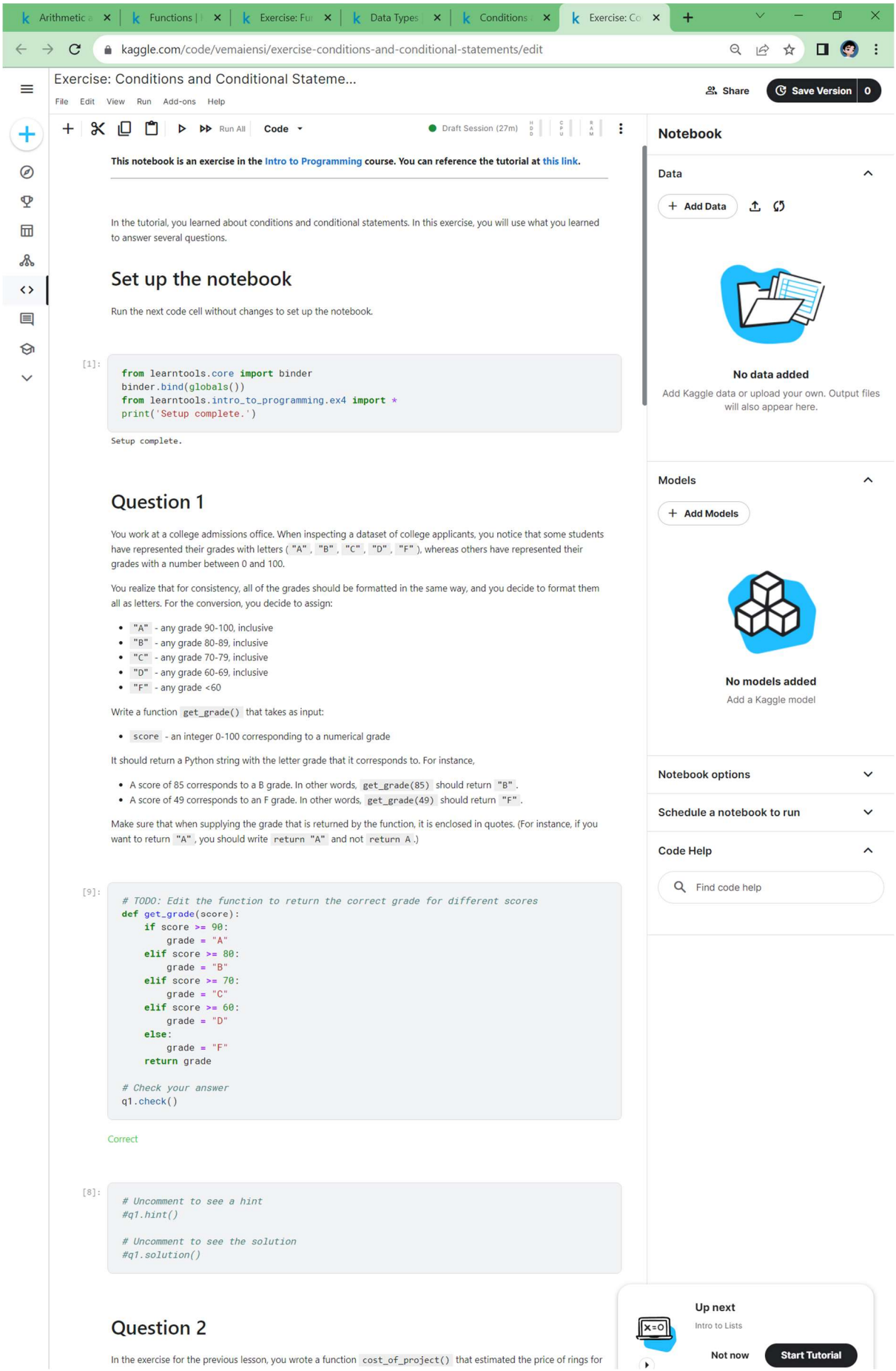
Up next

Conditions and Conditional Statements

Not now Start Tutorial


Arithmetic and V
Functions | Kaggl
Exercise: Functio
Data Types | Kag
Exercise: Data Ty
kaggle.com/code/vemaiensi/exercise-data-types/edit
Exercise: Data Types
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (18m)
Notebook
Data
Add Data
No data added
Add Kaggle data or upload your own. Output files will also appear here.
Models
Add Models
No models added
Add a Kaggle model
Notebook options
Schedule a notebook to run
Code Help
Find code help
Question 2
In the tutorial, you learned about booleans (which can take a value of True or False), in addition to integers, floats, and strings. For this question, your goal is to determine what happens when you multiply a boolean by any of these data types. Specifically,
What happens when you multiply an integer or float by True? What happens when you multiply them by False? How does the answer change if the numbers are positive or negative?
What happens when you multiply a string by True? By False?
Use the next code cell for your investigation.
[11]:
# Uncomment and run this code to get started!
print(3 * True)
print(-3.1 * True)
print(type("abc" * False))
print(len("abc" * False))
3
-3.1
<class 'str'>
0
Once you have an answer, run the code cell below to see the solution. Viewing the solution will give you credit for answering the problem.
[12]:
# Check your answer (Run this code cell to receive credit!)
q2.check()
Correct:
When you multiple an integer or float by a boolean with value True, it just returns that same integer or float (and is equivalent to multiplying by 1). If you multiply an integer or float by a boolean with value False, it always returns 0. This is true for both positive and negative numbers. If you multiply a string by a boolean with value True, it just returns that same string. And if you multiply a string by a boolean with value False, it returns an empty string (or a string with length zero).
Question 3
In this question, you will build off your work from the previous exercise to write a function that estimates the value of a house.
Use the next code cell to create a function get_expected_cost that takes as input three variables:
beds - number of bedrooms (data type float)
baths - number of bathrooms (data type float)
has_basement - whether or not the house has a basement (data type boolean)
It should return the expected cost of a house with those characteristics. Assume that:
the expected cost for a house with 0 bedrooms and 0 bathrooms, and no basement is 80000,
each bedroom adds 30000 to the expected cost,
each bathroom adds 10000 to the expected cost, and
a basement adds 40000 to the expected cost.
For instance,
a house with 1 bedroom, 1 bathroom, and no basement has an expected cost of 80000 + 30000 + 10000 = 120000. This value will be calculated with get_expected_cost(1, 1, False).
a house with 2 bedrooms, 1 bathroom, and a basement has an expected cost of 80000 + 2*30000 + 10000 + 40000 = 190000. This value will be calculated with get_expected_cost(2, 1, True).
Remember you can always get a hint by uncommenting q3.hint() in the code cell following the next!
[13]:
# TODO: Complete the function
def get_expected_cost(beds, baths, has_basement):
value = 80000 + beds * 30000 + baths * 10000 + has_basement * 40000
return value
# Check your answer
q3.check()
Correct
Up next
Conditions and Conditional Statements
Not now
Start Tutorial


Arithmetic and V
Functions | Kaggl
Exercise: Functio
Data Types | Kag
Exercise: Data Ty
kaggle.com/code/vemaiensi/exercise-data-types/edit
Exercise: Data Types
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (20m)
Notebook
Data
Add Data
No data added
Add Kaggle data or upload your own. Output files will also appear here.
Models
Add Models
No models added
Add a Kaggle model
Notebook options
Schedule a notebook to run
Code Help
Find code help
[14]:
print(False + False)
print(True + False)
print(False + True)
print(True + True)
print(False + True + True + True)
0
1
1
2
3
+ Code
+ Markdown
Once you have an answer, run the code cell below to see the solution. Viewing the solution will give you credit for answering the problem.
[15]:
# Check your answer (Run this code cell to receive credit!)
q4.check()
Correct:
When you add booleans, adding False is equivalent to adding 0, and adding True is equivalent to adding 1.
Question 5
You own an online shop where you sell rings with custom engravings. You offer both gold plated and solid gold rings.
Gold plated rings have a base cost of $50, and you charge $7 per engraved unit.
Solid gold rings have a base cost of $100, and you charge $10 per engraved unit.
Spaces and punctuation are counted as engraved units.
Write a function cost_of_project() that takes two arguments:
engraving - a Python string with the text of the engraving
solid_gold - a Boolean that indicates whether the ring is solid gold
It should return the cost of the project. This question should be fairly challenging, and you may need a hint.
[26]:
def cost_of_project(engraving, solid_gold):
cost = solid_gold * (100 + len(engraving) * 10) + (not solid_gold) * (50 + len(engrav
return cost
# Check your answer
q5.check()
Correct
# Uncomment to see a hint
#q5.hint()
# Uncomment to view the solution
#q5.solution()
+ Code
+ Markdown
Run the next code cell to calculate the cost of engraving Charlie+Denver on a solid gold ring.
[27]:
project_one = cost_of_project("Charlie+Denver", True)
print(project_one)
240
Use the next code cell to calculate the cost of engraving 08/10/2000 on a gold plated ring.
[28]:
project_two = cost_of_project("08/10/2000", False)
print(project_two)
120
Up next
Conditions and Conditional Statements
Not now
Start Tutorial



Arithmetic | Functions | Exercise: Fu | Data Types | Conditions | Exercise: Co

kaggle.com/code/vemaiensi/exercise-conditions-and-conditional-statements/edit

Exercise: Conditions and Conditional Stateme...

File Edit View Run Add-ons Help

Share | Save Version 0

Run All | Code | Draft Session (27m)

**This notebook is an exercise in the Intro to Programming course. You can reference the tutorial at this link.**

In the tutorial, you learned about conditions and conditional statements. In this exercise, you will use what you learned to answer several questions.

## Set up the notebook

Run the next code cell without changes to set up the notebook.

[1]:
```
from learntools.core import binder
binder.bind(globals())
from learntools.intro_to_programming.ex4 import *
print('Setup complete.')
```

```
Setup complete.
```

## Question 1

You work at a college admissions office. When inspecting a dataset of college applicants, you notice that some students have represented their grades with letters (`"A"`, `"B"`, `"C"`, `"D"`, `"F"`), whereas others have represented their grades with a number between 0 and 100.

You realize that for consistency, all of the grades should be formatted in the same way, and you decide to format them all as letters. For the conversion, you decide to assign:

- `"A"` - any grade 90-100, inclusive
- `"B"` - any grade 80-89, inclusive
- `"C"` - any grade 70-79, inclusive
- `"D"` - any grade 60-69, inclusive
- `"F"` - any grade <60

Write a function `get_grade()` that takes as input:

- `score` - an integer 0-100 corresponding to a numerical grade

It should return a Python string with the letter grade that it corresponds to. For instance,

- A score of 85 corresponds to a B grade. In other words, `get_grade(85)` should return `"B"`.
- A score of 49 corresponds to an F grade. In other words, `get_grade(49)` should return `"F"`.

Make sure that when supplying the grade that is returned by the function, it is enclosed in quotes. (For instance, if you want to return `"A"`, you should write `return "A"` and not `return A`.)

[9]:
```
# TODO: Edit the function to return the correct grade for different scores
def get_grade(score):
    if score >= 90:
        grade = "A"
    elif score >= 80:
        grade = "B"
    elif score >= 70:
        grade = "C"
    elif score >= 60:
        grade = "D"
    else:
        grade = "F"
    return grade

# Check your answer
q1.check()
```

Correct

[8]:
```
# Uncomment to see a hint
#q1.hint()

# Uncomment to see the solution
#q1.solution()
```

## Question 2

In the exercise for the previous lesson, you wrote a function `cost_of_project()` that estimated the price of rings for

Notebook

Data

+ Add Data

No data added

Add Kaggle data or upload your own. Output files will also appear here.

Models

+ Add Models

No models added

Add a Kaggle model

Notebook options

Schedule a notebook to run

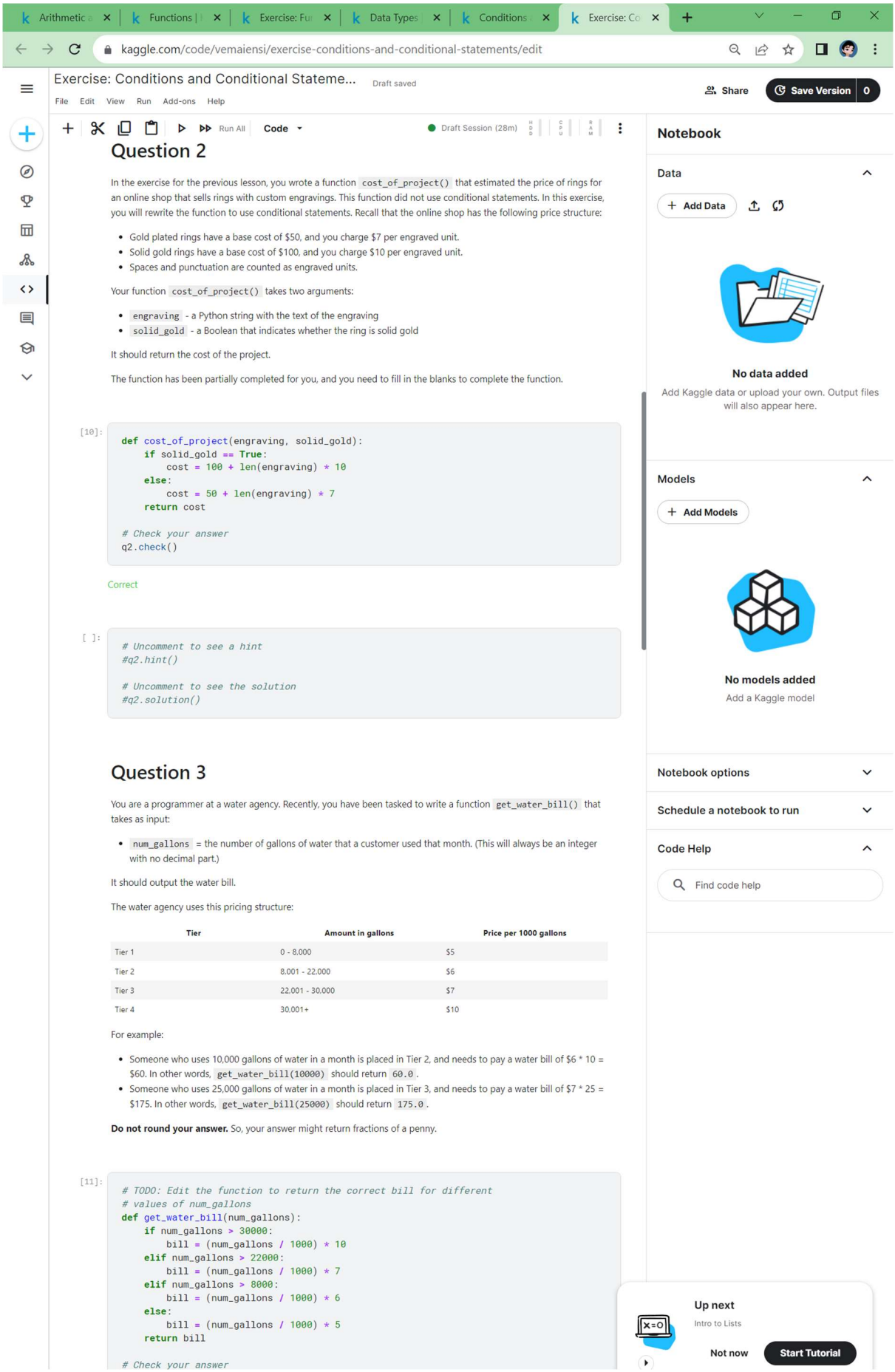
Code Help

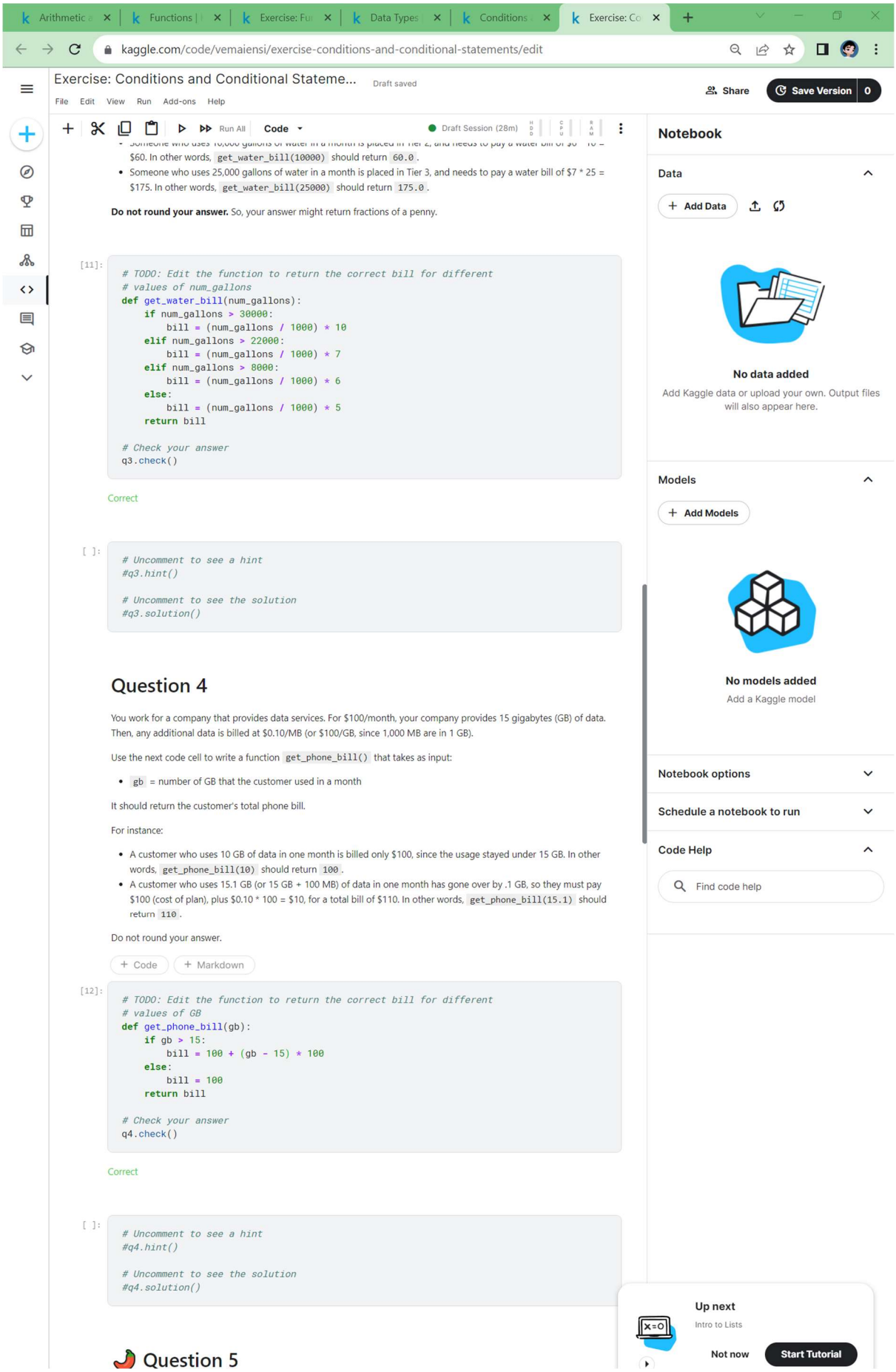
Find code help

Up next

Intro to Lists

Not now | Start Tutorial


Arithmetic
Functions
Exercise: Fu
Data Types
Conditions
Exercise: Co
kaggle.com/code/vemaiensi/exercise-conditions-and-conditional-statements/edit
Exercise: Conditions and Conditional Stateme...
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (28m)
Question 2
In the exercise for the previous lesson, you wrote a function cost_of_project() that estimated the price of rings for an online shop that sells rings with custom engravings. This function did not use conditional statements. In this exercise, you will rewrite the function to use conditional statements. Recall that the online shop has the following price structure:
Gold plated rings have a base cost of $50, and you charge $7 per engraved unit.
Solid gold rings have a base cost of $100, and you charge $10 per engraved unit.
Spaces and punctuation are counted as engraved units.
Your function cost_of_project() takes two arguments:
engraving - a Python string with the text of the engraving
solid_gold - a Boolean that indicates whether the ring is solid gold
It should return the cost of the project.
The function has been partially completed for you, and you need to fill in the blanks to complete the function.
[10]:
def cost_of_project(engraving, solid_gold):
if solid_gold == True:
cost = 100 + len(engraving) * 10
else:
cost = 50 + len(engraving) * 7
return cost
# Check your answer
q2.check()
Correct
[ ]:
# Uncomment to see a hint
#q2.hint()
# Uncomment to see the solution
#q2.solution()
Question 3
You are a programmer at a water agency. Recently, you have been tasked to write a function get_water_bill() that takes as input:
num_gallons = the number of gallons of water that a customer used that month. (This will always be an integer with no decimal part.)
It should output the water bill.
The water agency uses this pricing structure:
Tier
Amount in gallons
Price per 1000 gallons
Tier 1
0 - 8,000
$5
Tier 2
8,001 - 22,000
$6
Tier 3
22,001 - 30,000
$7
Tier 4
30,001+
$10
For example:
Someone who uses 10,000 gallons of water in a month is placed in Tier 2, and needs to pay a water bill of $6 * 10 = $60. In other words, get_water_bill(10000) should return 60.0.
Someone who uses 25,000 gallons of water in a month is placed in Tier 3, and needs to pay a water bill of $7 * 25 = $175. In other words, get_water_bill(25000) should return 175.0.
Do not round your answer. So, your answer might return fractions of a penny.
[11]:
# TODO: Edit the function to return the correct bill for different
# values of num_gallons
def get_water_bill(num_gallons):
if num_gallons > 30000:
bill = (num_gallons / 1000) * 10
elif num_gallons > 22000:
bill = (num_gallons / 1000) * 7
elif num_gallons > 8000:
bill = (num_gallons / 1000) * 6
else:
bill = (num_gallons / 1000) * 5
return bill
# Check your answer
Notebook
Data
Add Data
No data added
Add Kaggle data or upload your own. Output files will also appear here.
Models
Add Models
No models added
Add a Kaggle model
Notebook options
Schedule a notebook to run
Code Help
Find code help
Up next
Intro to Lists
Not now
Start Tutorial


Arithmetic
Functions
Exercise: Fu
Data Types
Conditions
Exercise: Co
kaggle.com/code/vemaiensi/exercise-conditions-and-conditional-statements/edit
Exercise: Conditions and Conditional Stateme...
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (28m)
$60. In other words, get_water_bill(10000) should return 60.0 .
Someone who uses 25,000 gallons of water in a month is placed in Tier 3, and needs to pay a water bill of $7 * 25 = $175. In other words, get_water_bill(25000) should return 175.0 .
Do not round your answer. So, your answer might return fractions of a penny.
[11]:
# TODO: Edit the function to return the correct bill for different
# values of num_gallons
def get_water_bill(num_gallons):
if num_gallons > 30000:
bill = (num_gallons / 1000) * 10
elif num_gallons > 22000:
bill = (num_gallons / 1000) * 7
elif num_gallons > 8000:
bill = (num_gallons / 1000) * 6
else:
bill = (num_gallons / 1000) * 5
return bill
# Check your answer
q3.check()
Correct
[ ]:
# Uncomment to see a hint
#q3.hint()
# Uncomment to see the solution
#q3.solution()
Question 4
You work for a company that provides data services. For $100/month, your company provides 15 gigabytes (GB) of data. Then, any additional data is billed at $0.10/MB (or $100/GB, since 1,000 MB are in 1 GB).
Use the next code cell to write a function get_phone_bill() that takes as input:
gb = number of GB that the customer used in a month
It should return the customer's total phone bill.
For instance:
A customer who uses 10 GB of data in one month is billed only $100, since the usage stayed under 15 GB. In other words, get_phone_bill(10) should return 100 .
A customer who uses 15.1 GB (or 15 GB + 100 MB) of data in one month has gone over by .1 GB, so they must pay $100 (cost of plan), plus $0.10 * 100 = $10, for a total bill of $110. In other words, get_phone_bill(15.1) should return 110 .
Do not round your answer.
+ Code
+ Markdown
[12]:
# TODO: Edit the function to return the correct bill for different
# values of GB
def get_phone_bill(gb):
if gb > 15:
bill = 100 + (gb - 15) * 100
else:
bill = 100
return bill
# Check your answer
q4.check()
Correct
[ ]:
# Uncomment to see a hint
#q4.hint()
# Uncomment to see the solution
#q4.solution()
Question 5
Notebook
Data
+ Add Data
No data added
Add Kaggle data or upload your own. Output files will also appear here.
Models
+ Add Models
No models added
Add a Kaggle model
Notebook options
Schedule a notebook to run
Code Help
Find code help
Up next
Intro to Lists
Not now
Start Tutorial

Arithmetic | Functions | Exercise: Fu | Data Types | Conditions | Exercise: Co

kaggle.com/code/vemaiensi/exercise-conditions-and-conditional-statements/edit

Exercise: Conditions and Conditional Stateme... Draft saved

File Edit View Run Add-ons Help

Share | Save Version 0

Run All | Code | Draft Session (28m)

## Question 5

In Mexico, foods and beverages that are high in saturated fat, trans fat, sugar, sodium, and/or calories appear with warning labels that are designed to help consumers make healthy food choices.

For instance, the box of cookies in the image below appears with two labels (in the upper right corner):

- EXCESO CALORÍAS (in English, EXCESS CALORIES)
- EXCESO AZÚCARES (in English, EXCESS SUGAR)


In this question, you'll work with a function `get_labels()` that takes the nutritional details about a food item and prints the needed warning labels. This function takes several inputs:

- `food_type` = one of `"solid"` or `"liquid"`
- `serving_size` = size of one serving (if solid, in grams; if liquid, in milliliters)
- `calories_per_serving` = calories in one serving
- `saturated_fat_g` = grams of saturated fat in one serving
- `trans_fat_g` = grams of trans fat in one serving
- `sodium_mg` = mg of sodium in one serving
- `sugars_g` = grams of sugar in one serving

Note that some of the code here should feel unfamiliar to you, since we have not shared the details of how some of the functions like `excess_sugar()` or `excess_saturated_fat()` work. But at a high level, these are functions that return a value of `True` if the food is deemed to have an excess of sugar or saturated fat, respectively. These functions are used within the `get_labels()` function, and whenever there is an excess (of sugar or saturated fat, but also of trans fat, sodium, or calories), it prints the corresponding label.

[17]:
```
# import functions needed to make get_labels work
from learntools.intro_to_programming.ex4q5 import excess_sugar, excess_saturated_fat, exc

def get_labels(food_type, serving_size, calories_per_serving, saturated_fat_g, trans_fat_
    # Print messages based on findings
    if excess_sugar(sugars_g, calories_per_serving) == True:
        print("EXCESO AZÚCARES / EXCESS SUGAR")
    if excess_saturated_fat(saturated_fat_g, calories_per_serving) == True:
        print("EXCESO GRASAS SATURADAS / EXCESS SATURATED FAT")
    if excess_trans_fat(trans_fat_g, calories_per_serving) == True:
        print("EXCESO GRASAS TRANS / EXCESS TRANS FAT")
    if excess_sodium(calories_per_serving, sodium_mg) == True:
        print("EXCESO SODIO / EXCESS SODIUM")
    if excess_calories(food_type, calories_per_serving, serving_size) == True:
        print("EXCESO CALORÍAS / EXCESS CALORIES")
```

The next code cell demonstrates how to use `get_labels()` to get the warning labels that the food item should contain. We begin with bologna. Here is an image with all of the nutritional information. Note that for this food,

- `food_type = "solid"` (because bologna is a solid and not a liquid)
- `serving_size = 32` (the serving size is 32 grams)
- `calories_per_serving = 110` (there are 110 calories per serving)
- `saturated_fat_g = 2.5` (there are 2.5 grams of saturated fat per serving)
- `trans_fat_g = 0` (there are 0 grams of trans fat per serving)
- `sodium_mg = 400` (there are 400 mg of sodium per serving)
- `sugars_g = 1` (the nutrition facts say <1g, but we will round it up to 1 gram per serving to be safe)

By supplying all of these values to the function, we can print the warning labels.

[19]:
```
# bologna https://world.openfoodfacts.org/product/4099100179378/bologna
get_labels("solid", 32, 110, 2.5, 0, 400, 1)
```

```
EXCESO GRASAS SATURADAS / EXCESS SATURATED FAT
EXCESO SODIO / EXCESS SODIUM
```

Notebook

Data

Add Data

No data added

Add Kaggle data or upload your own. Output files will also appear here.

Models

Add Models

No models added

Add a Kaggle model

Notebook options

Schedule a notebook to run

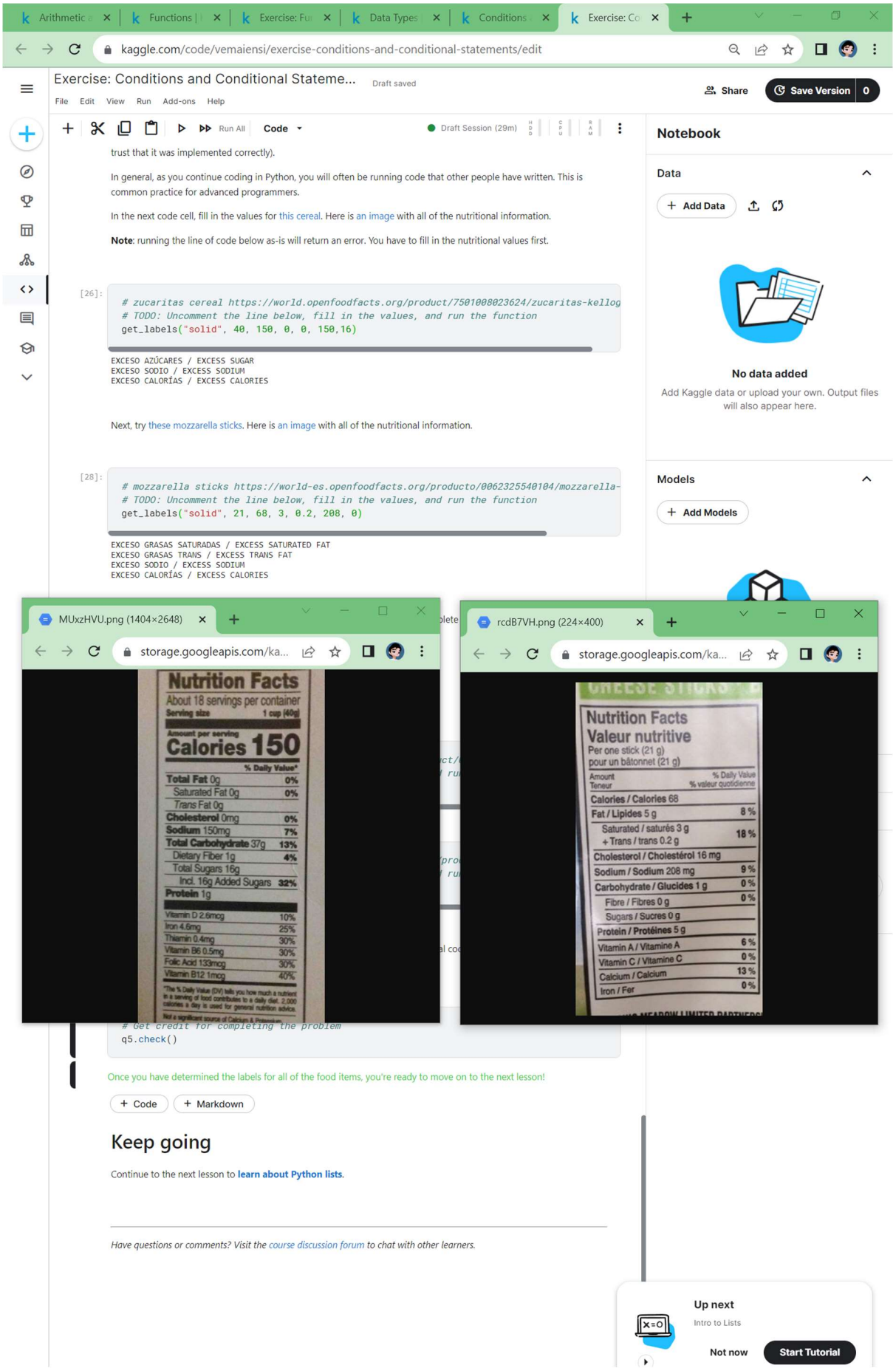
Code Help

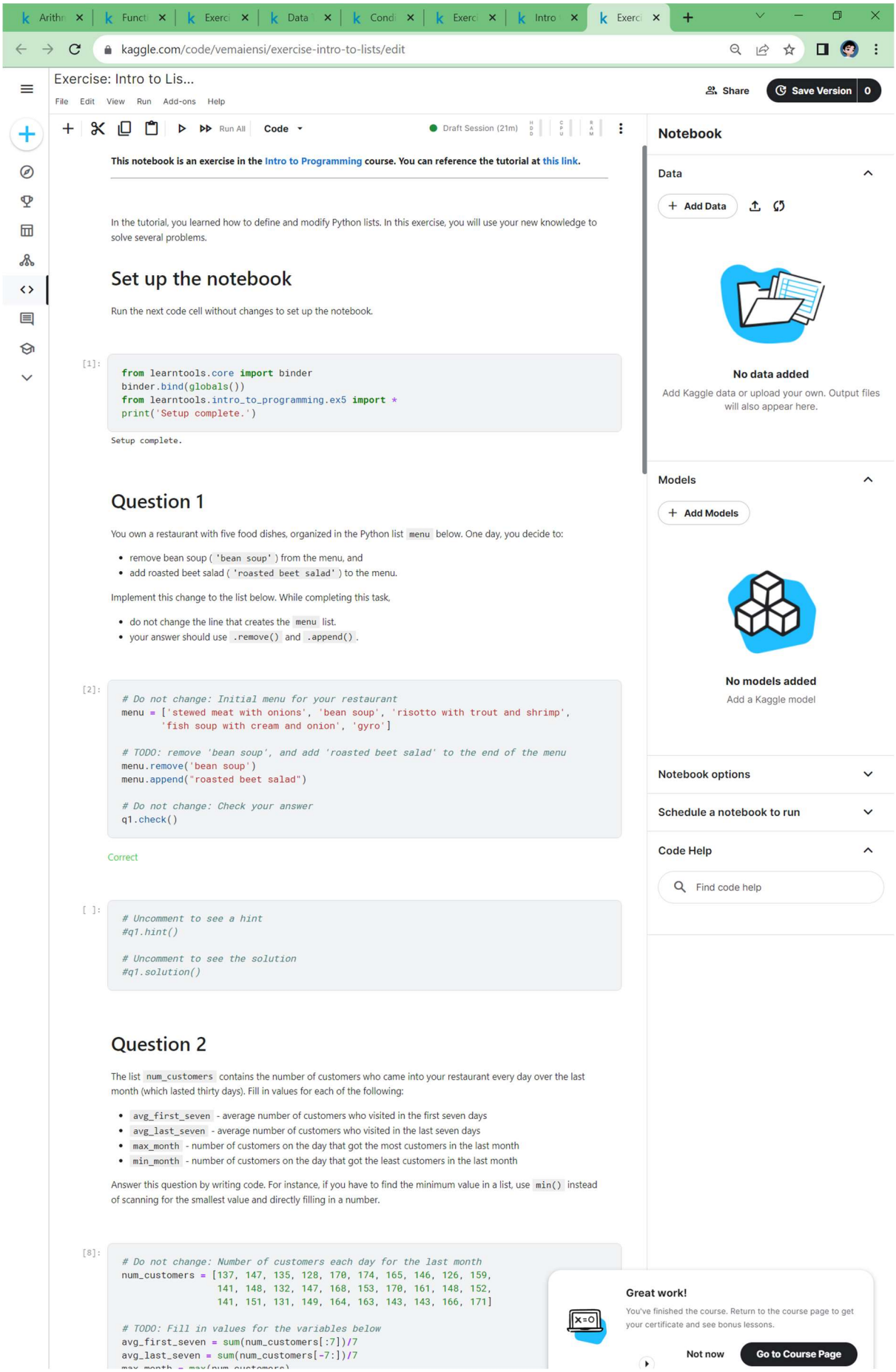
Find code help

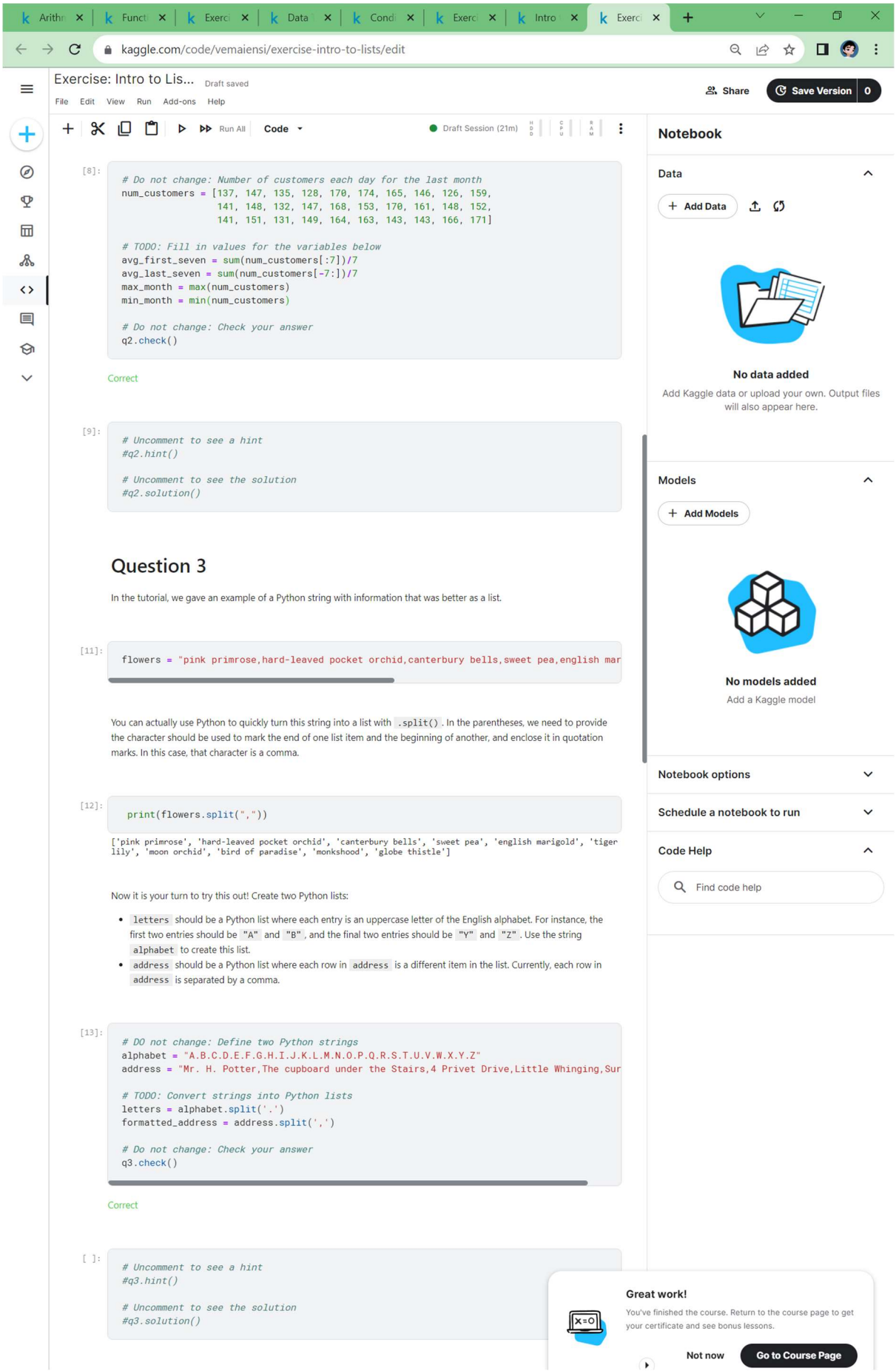
Up next

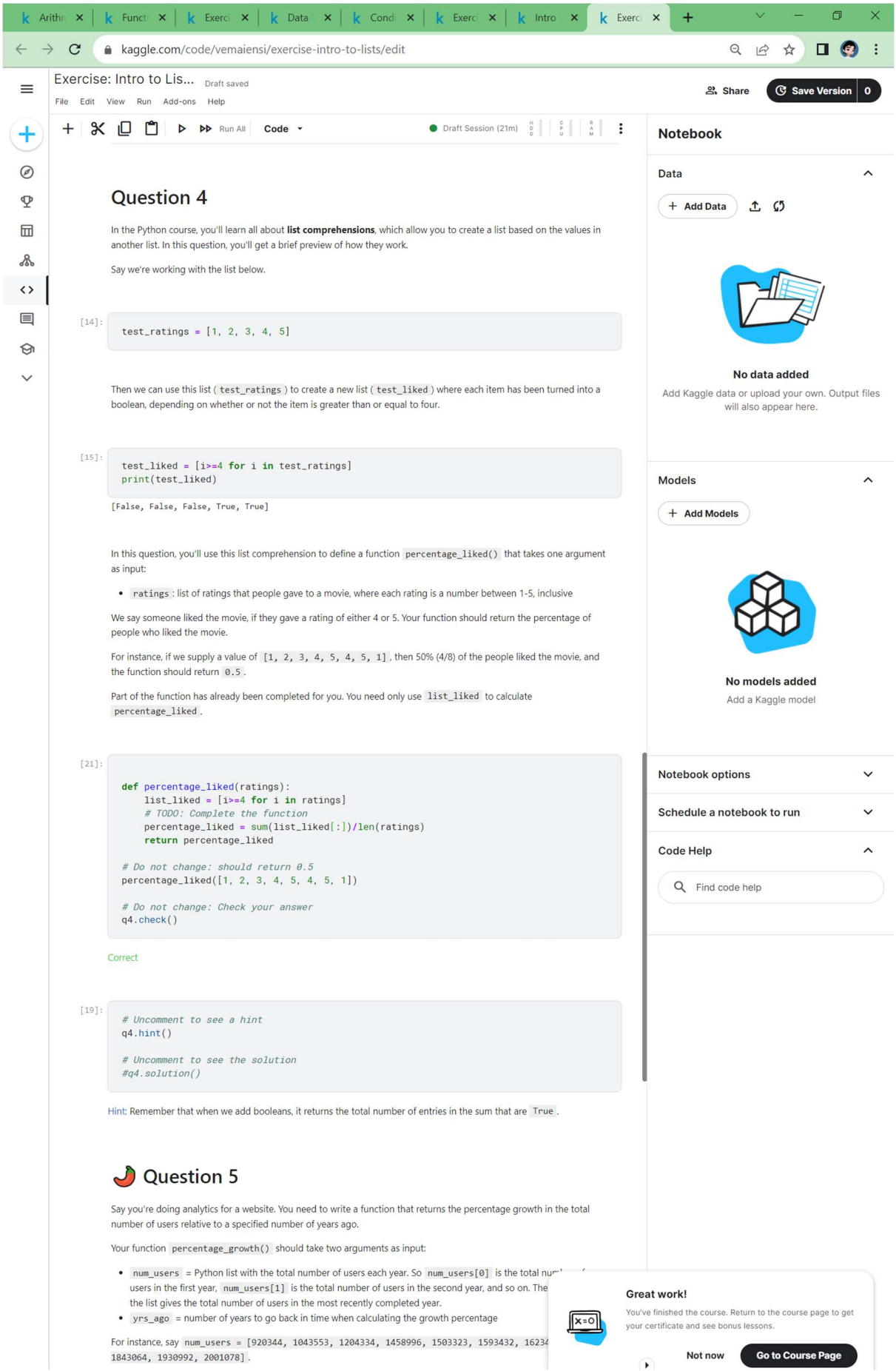
Intro to Lists

Not now | Start Tutorial


Exercise: Conditions and Conditional Stateme...
Draft saved
Share
Save Version 0
File Edit View Run Add-ons Help
Run All
Code
Draft Session (29m)
Notebook
Data
Add Data
No data added
Add Kaggle data or upload your own. Output files will also appear here.
Models
Add Models
trust that it was implemented correctly).
In general, as you continue coding in Python, you will often be running code that other people have written. This is common practice for advanced programmers.
In the next code cell, fill in the values for this cereal. Here is an image with all of the nutritional information.
Note: running the line of code below as-is will return an error. You have to fill in the nutritional values first.
[26]:
# zucaritas cereal https://world.openfoodfacts.org/product/7501008023624/zucaritas-kellog
# TODO: Uncomment the line below, fill in the values, and run the function
get_labels("solid", 40, 150, 0, 0, 150,16)
EXCESO AZÚCARES / EXCESS SUGAR
EXCESO SODIO / EXCESS SODIUM
EXCESO CALORÍAS / EXCESS CALORIES
Next, try these mozzarella sticks. Here is an image with all of the nutritional information.
[28]:
# mozzarella sticks https://world-es.openfoodfacts.org/producto/0062325540104/mozzarella-
# TODO: Uncomment the line below, fill in the values, and run the function
get_labels("solid", 21, 68, 3, 0.2, 208, 0)
EXCESO GRASAS SATURADAS / EXCESS SATURATED FAT
EXCESO GRASAS TRANS / EXCESS TRANS FAT
EXCESO SODIO / EXCESS SODIUM
EXCESO CALORÍAS / EXCESS CALORIES
MUxzHVU.png (1404×2648)
storage.googleapis.com/ka...
Nutrition Facts
About 18 servings per container
Serving size 1 cup (40g)
Amount per serving
Calories 150
% Daily Value*
Total Fat 0g 0%
Saturated Fat 0g 0%
Trans Fat 0g
Cholesterol 0mg 0%
Sodium 150mg 7%
Total Carbohydrate 37g 13%
Dietary Fiber 1g 4%
Total Sugars 16g
Incl. 16g Added Sugars 32%
Protein 1g
Vitamin D 2.6mcg 10%
Iron 4.6mg 25%
Thiamin 0.4mg 30%
Vitamin B6 0.5mg 30%
Folic Acid 133mcg 30%
Vitamin B12 1mcg 40%
rcdB7VH.png (224×400)
storage.googleapis.com/ka...
Nutrition Facts
Valeur nutritive
Per one stick (21 g)
pour un bâtonnet (21 g)
Amount / Teneur
% Daily Value / % valeur quotidienne
Calories / Calories 68
Fat / Lipides 5 g 8 %
Saturated / saturés 3 g + Trans / trans 0.2 g 18 %
Cholesterol / Cholestérol 16 mg
Sodium / Sodium 208 mg 9 %
Carbohydrate / Glucides 1 g 0 %
Fibre / Fibres 0 g 0 %
Sugars / Sucres 0 g
Protein / Protéines 5 g
Vitamin A / Vitamine A 6 %
Vitamin C / Vitamine C 0 %
Calcium / Calcium 13 %
Iron / Fer 0 %
# Get credit for completing the problem
q5.check()
Once you have determined the labels for all of the food items, you're ready to move on to the next lesson!
+ Code
+ Markdown
Keep going
Continue to the next lesson to learn about Python lists.
Have questions or comments? Visit the course discussion forum to chat with other learners.
Up next
Intro to Lists
Not now
Start Tutorial


kaggle.com/code/vemaiensi/exercise-intro-to-lists/edit
Exercise: Intro to Lis...
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (21m)
Notebook
Data
Add Data
No data added
Add Kaggle data or upload your own. Output files will also appear here.
Models
Add Models
No models added
Add a Kaggle model
Notebook options
Schedule a notebook to run
Code Help
Find code help
This notebook is an exercise in the Intro to Programming course. You can reference the tutorial at this link.
In the tutorial, you learned how to define and modify Python lists. In this exercise, you will use your new knowledge to solve several problems.
Set up the notebook
Run the next code cell without changes to set up the notebook.
[1]:
from learntools.core import binder
binder.bind(globals())
from learntools.intro_to_programming.ex5 import *
print('Setup complete.')
Setup complete.
Question 1
You own a restaurant with five food dishes, organized in the Python list menu below. One day, you decide to:
remove bean soup ( 'bean soup' ) from the menu, and
add roasted beet salad ( 'roasted beet salad' ) to the menu.
Implement this change to the list below. While completing this task,
do not change the line that creates the menu list.
your answer should use .remove() and .append() .
[2]:
# Do not change: Initial menu for your restaurant
menu = ['stewed meat with onions', 'bean soup', 'risotto with trout and shrimp',
'fish soup with cream and onion', 'gyro']
# TODO: remove 'bean soup', and add 'roasted beet salad' to the end of the menu
menu.remove('bean soup')
menu.append("roasted beet salad")
# Do not change: Check your answer
q1.check()
Correct
[ ]:
# Uncomment to see a hint
#q1.hint()
# Uncomment to see the solution
#q1.solution()
Question 2
The list num_customers contains the number of customers who came into your restaurant every day over the last month (which lasted thirty days). Fill in values for each of the following:
avg_first_seven - average number of customers who visited in the first seven days
avg_last_seven - average number of customers who visited in the last seven days
max_month - number of customers on the day that got the most customers in the last month
min_month - number of customers on the day that got the least customers in the last month
Answer this question by writing code. For instance, if you have to find the minimum value in a list, use min() instead of scanning for the smallest value and directly filling in a number.
[8]:
# Do not change: Number of customers each day for the last month
num_customers = [137, 147, 135, 128, 170, 174, 165, 146, 126, 159,
141, 148, 132, 147, 168, 153, 170, 161, 148, 152,
141, 151, 131, 149, 164, 163, 143, 143, 166, 171]
# TODO: Fill in values for the variables below
avg_first_seven = sum(num_customers[:7])/7
avg_last_seven = sum(num_customers[-7:])/7
Great work!
You've finished the course. Return to the course page to get your certificate and see bonus lessons.
Not now
Go to Course Page


kaggle.com/code/vemaiensi/exercise-intro-to-lists/edit
Exercise: Intro to Lis...
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (21m)
[8]:
# Do not change: Number of customers each day for the last month
num_customers = [137, 147, 135, 128, 170, 174, 165, 146, 126, 159,
141, 148, 132, 147, 168, 153, 170, 161, 148, 152,
141, 151, 131, 149, 164, 163, 143, 143, 166, 171]
# TODO: Fill in values for the variables below
avg_first_seven = sum(num_customers[:7])/7
avg_last_seven = sum(num_customers[-7:])/7
max_month = max(num_customers)
min_month = min(num_customers)
# Do not change: Check your answer
q2.check()
Correct
[9]:
# Uncomment to see a hint
#q2.hint()
# Uncomment to see the solution
#q2.solution()
Question 3
In the tutorial, we gave an example of a Python string with information that was better as a list.
[11]:
flowers = "pink primrose,hard-leaved pocket orchid,canterbury bells,sweet pea,english mar
You can actually use Python to quickly turn this string into a list with .split() . In the parentheses, we need to provide the character should be used to mark the end of one list item and the beginning of another, and enclose it in quotation marks. In this case, that character is a comma.
[12]:
print(flowers.split(","))
['pink primrose', 'hard-leaved pocket orchid', 'canterbury bells', 'sweet pea', 'english marigold', 'tiger lily', 'moon orchid', 'bird of paradise', 'monkshood', 'globe thistle']
Now it is your turn to try this out! Create two Python lists:
letters should be a Python list where each entry is an uppercase letter of the English alphabet. For instance, the first two entries should be "A" and "B", and the final two entries should be "Y" and "Z". Use the string alphabet to create this list.
address should be a Python list where each row in address is a different item in the list. Currently, each row in address is separated by a comma.
[13]:
# DO not change: Define two Python strings
alphabet = "A.B.C.D.E.F.G.H.I.J.K.L.M.N.O.P.Q.R.S.T.U.V.W.X.Y.Z"
address = "Mr. H. Potter,The cupboard under the Stairs,4 Privet Drive,Little Whinging,Sur
# TODO: Convert strings into Python lists
letters = alphabet.split('.')
formatted_address = address.split(',')
# Do not change: Check your answer
q3.check()
Correct
[ ]:
# Uncomment to see a hint
#q3.hint()
# Uncomment to see the solution
#q3.solution()
Notebook
Data
Add Data
No data added
Add Kaggle data or upload your own. Output files will also appear here.
Models
Add Models
No models added
Add a Kaggle model
Notebook options
Schedule a notebook to run
Code Help
Find code help
Great work!
You've finished the course. Return to the course page to get your certificate and see bonus lessons.
Not now
Go to Course Page




## Question 4

In the Python course, you'll learn all about **list comprehensions**, which allow you to create a list based on the values in another list. In this question, you'll get a brief preview of how they work.

Say we're working with the list below.

[14]:
```
test_ratings = [1, 2, 3, 4, 5]
```

Then we can use this list ( `test_ratings` ) to create a new list ( `test_liked` ) where each item has been turned into a boolean, depending on whether or not the item is greater than or equal to four.

[15]:
```
test_liked = [i>=4 for i in test_ratings]
print(test_liked)
```

```
[False, False, False, True, True]
```

In this question, you'll use this list comprehension to define a function `percentage_liked()` that takes one argument as input:

- `ratings` : list of ratings that people gave to a movie, where each rating is a number between 1-5, inclusive

We say someone liked the movie, if they gave a rating of either 4 or 5. Your function should return the percentage of people who liked the movie.

For instance, if we supply a value of `[1, 2, 3, 4, 5, 4, 5, 1]` , then 50% (4/8) of the people liked the movie, and the function should return `0.5` .

Part of the function has already been completed for you. You need only use `list_liked` to calculate `percentage_liked` .

[21]:
```
def percentage_liked(ratings):
    list_liked = [i>=4 for i in ratings]
    # TODO: Complete the function
    percentage_liked = sum(list_liked[:])/len(ratings)
    return percentage_liked

# Do not change: should return 0.5
percentage_liked([1, 2, 3, 4, 5, 4, 5, 1])

# Do not change: Check your answer
q4.check()
```

Correct

[19]:
```
# Uncomment to see a hint
q4.hint()

# Uncomment to see the solution
#q4.solution()
```

Hint: Remember that when we add booleans, it returns the total number of entries in the sum that are `True` .

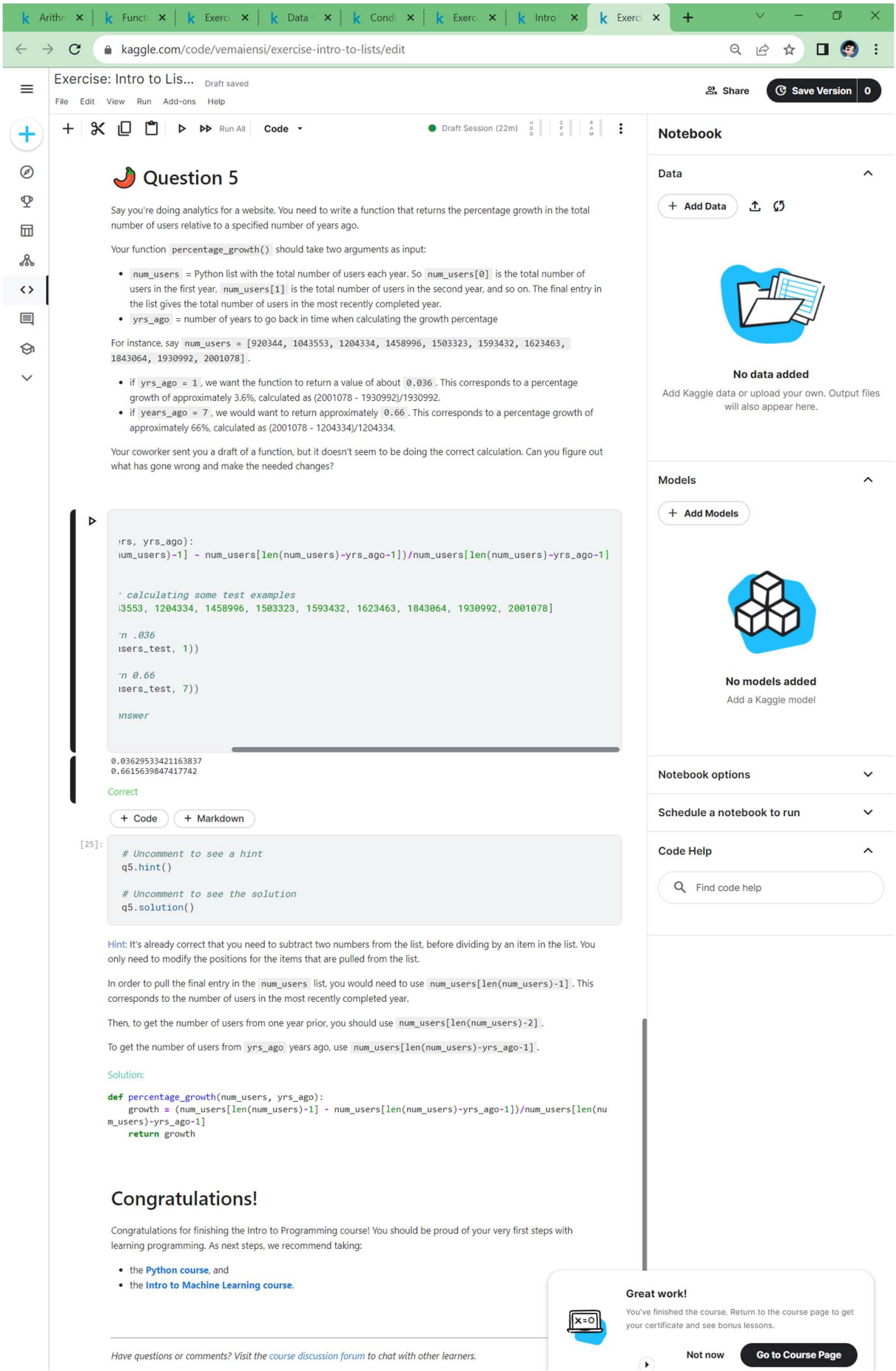
## Question 5

Say you're doing analytics for a website. You need to write a function that returns the percentage growth in the total number of users relative to a specified number of years ago.

Your function `percentage_growth()` should take two arguments as input:

- `num_users` = Python list with the total number of users each year. So `num_users[0]` is the total nu
users in the first year, `num_users[1]` is the total number of users in the second year, and so on. The
the list gives the total number of users in the most recently completed year.
- `yrs_ago` = number of years to go back in time when calculating the growth percentage

For instance, say `num_users = [920344, 1043553, 1204334, 1458996, 1503323, 1593432, 1623`
`1843064, 1930992, 2001078]` .


kaggle.com/code/vemaiensi/exercise-intro-to-lists/edit
Exercise: Intro to Lis...
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session (22m)
Notebook
Data
Add Data
No data added
Add Kaggle data or upload your own. Output files will also appear here.
Models
Add Models
No models added
Add a Kaggle model
Notebook options
Schedule a notebook to run
Code Help
Find code help
Question 5
Say you're doing analytics for a website. You need to write a function that returns the percentage growth in the total number of users relative to a specified number of years ago.
Your function percentage_growth() should take two arguments as input:
num_users = Python list with the total number of users each year. So num_users[0] is the total number of users in the first year, num_users[1] is the total number of users in the second year, and so on. The final entry in the list gives the total number of users in the most recently completed year.
yrs_ago = number of years to go back in time when calculating the growth percentage
For instance, say num_users = [920344, 1043553, 1204334, 1458996, 1503323, 1593432, 1623463, 1843064, 1930992, 2001078].
if yrs_ago = 1, we want the function to return a value of about 0.036. This corresponds to a percentage growth of approximately 3.6%, calculated as (2001078 - 1930992)/1930992.
if years_ago = 7, we would want to return approximately 0.66. This corresponds to a percentage growth of approximately 66%, calculated as (2001078 - 1204334)/1204334.
Your coworker sent you a draft of a function, but it doesn't seem to be doing the correct calculation. Can you figure out what has gone wrong and make the needed changes?
rs, yrs_ago):
um_users)-1] - num_users[len(num_users)-yrs_ago-1])/num_users[len(num_users)-yrs_ago-1]
calculating some test examples
3553, 1204334, 1458996, 1503323, 1593432, 1623463, 1843064, 1930992, 2001078]
n .036
sers_test, 1))
n 0.66
sers_test, 7))
nswer
0.03629533421163837
0.661563984741742
Correct
+ Code
+ Markdown
[25]:
# Uncomment to see a hint
q5.hint()
# Uncomment to see the solution
q5.solution()
Hint: It's already correct that you need to subtract two numbers from the list, before dividing by an item in the list. You only need to modify the positions for the items that are pulled from the list.
In order to pull the final entry in the num_users list, you would need to use num_users[len(num_users)-1]. This corresponds to the number of users in the most recently completed year.
Then, to get the number of users from one year prior, you should use num_users[len(num_users)-2].
To get the number of users from yrs_ago years ago, use num_users[len(num_users)-yrs_ago-1].
Solution:
def percentage_growth(num_users, yrs_ago):
growth = (num_users[len(num_users)-1] - num_users[len(num_users)-yrs_ago-1])/num_users[len(nu
m_users)-yrs_ago-1]
return growth
Congratulations!
Congratulations for finishing the Intro to Programming course! You should be proud of your very first steps with learning programming. As next steps, we recommend taking:
the Python course, and
the Intro to Machine Learning course.
Have questions or comments? Visit the course discussion forum to chat with other learners.
Great work!
You've finished the course. Return to the course page to get your certificate and see bonus lessons.
Not now
Go to Course Page

Search

# Intro to Programming

Get started with Python, if you have no coding experience.


View Certificate

100% complete! Congrats!

Courses Discussions

## Lessons

| | | Tutorial | Exercise |
|---|---|---|---|
| 1 | **Arithmetic and Variables**<br>Make calculations, and define and modify variables. | ✓ | ✓ |
| 2 | **Functions**<br>Organize your code and avoid redundancy. | ✓ | ✓ |
| 3 | **Data Types**<br>Explore integers, floats, booleans, and strings. | ✓ | ✓ |
| 4 | **Conditions and Conditional Statements**<br>Modify how functions run, depending on the input. | ✓ | ✓ |
| 5 | **Intro to Lists**<br>Organize your data so you can work with it efficiently. | ✓ | ✓ |

**Preparation for**
Python

**Hours to earn certificate**
5 (estimated)

**Cost**
No cost, like all Kaggle Learn Courses

**Instructor**
Alexis Cook

## Bonus Lessons

### Apply what you've learned

These lessons aren't required for your certificate, but bridge the gap between courses and applying your new skills!

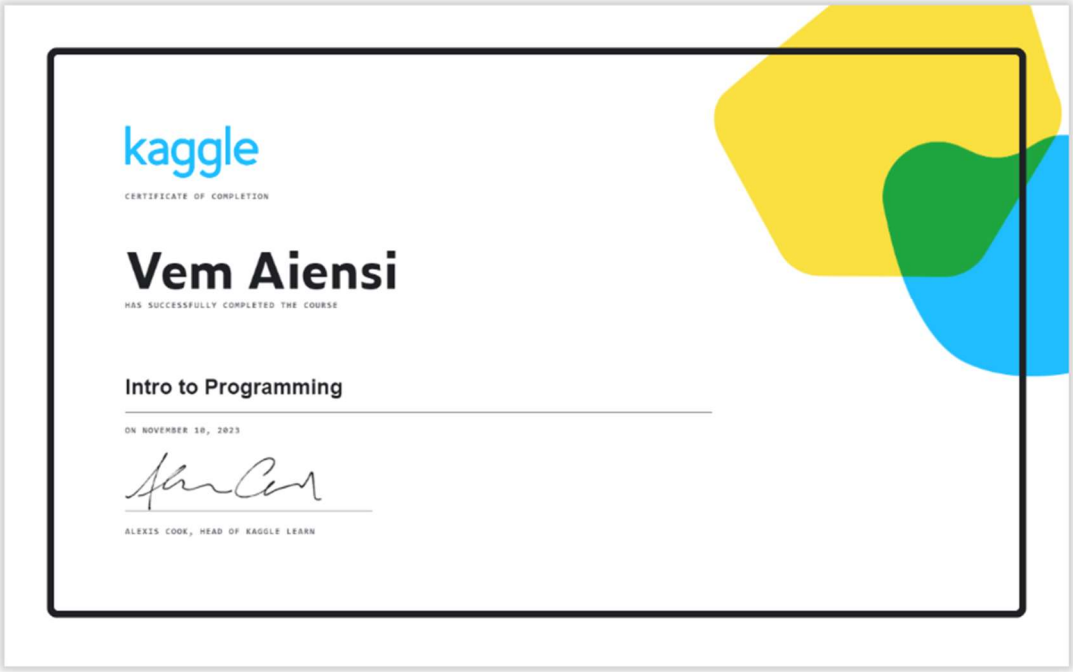
**Titanic Tutorial**

Make your first submission to a Kaggle competition!

Return to lesson →

# Congratulations!

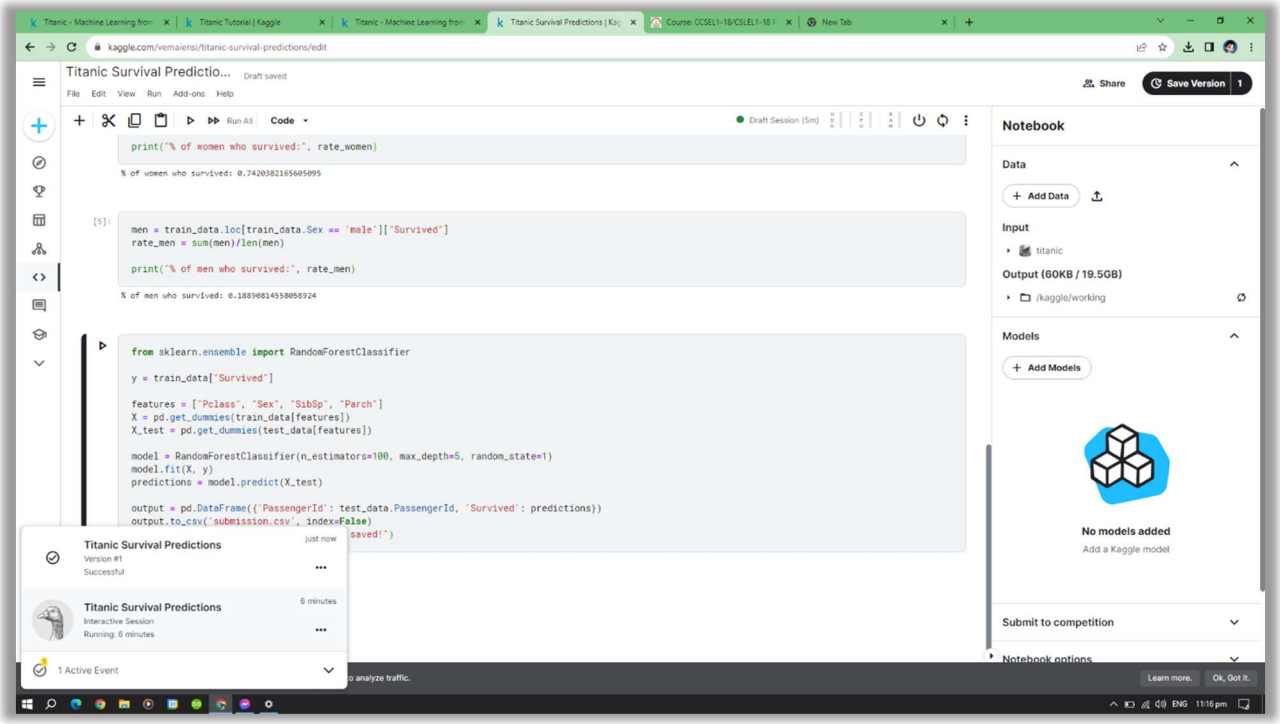
You've completed Intro to Programming.

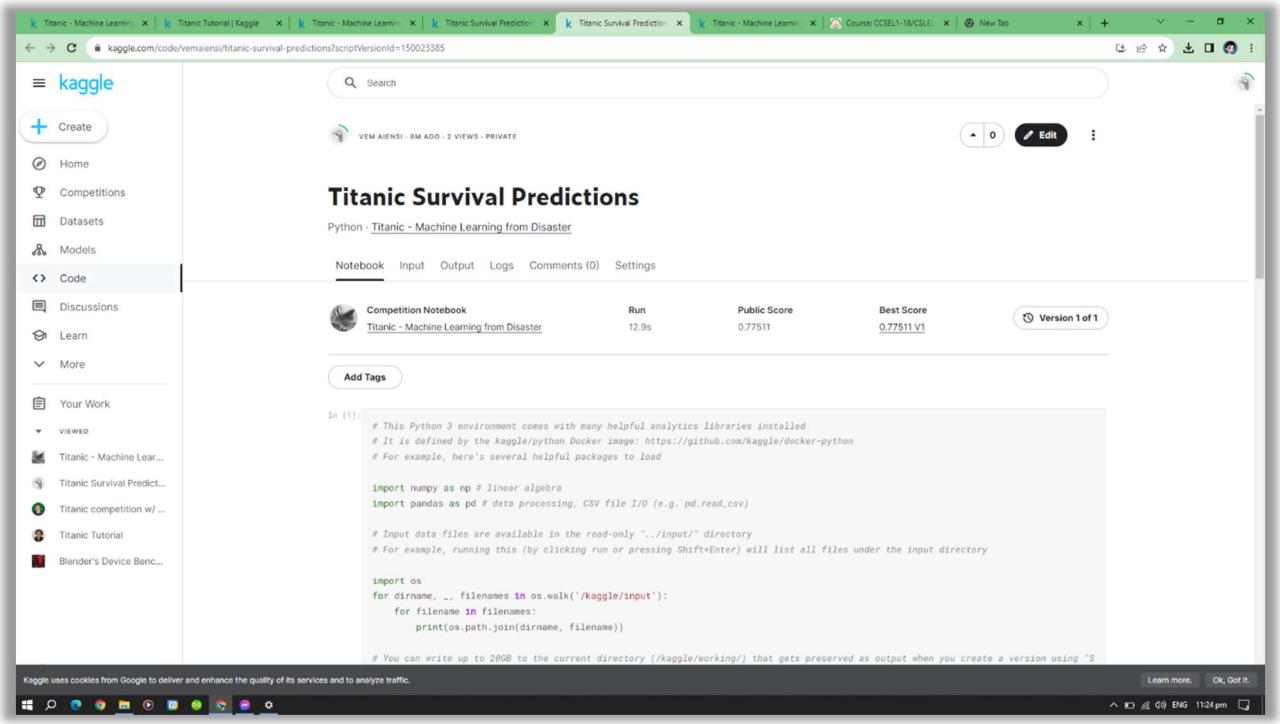


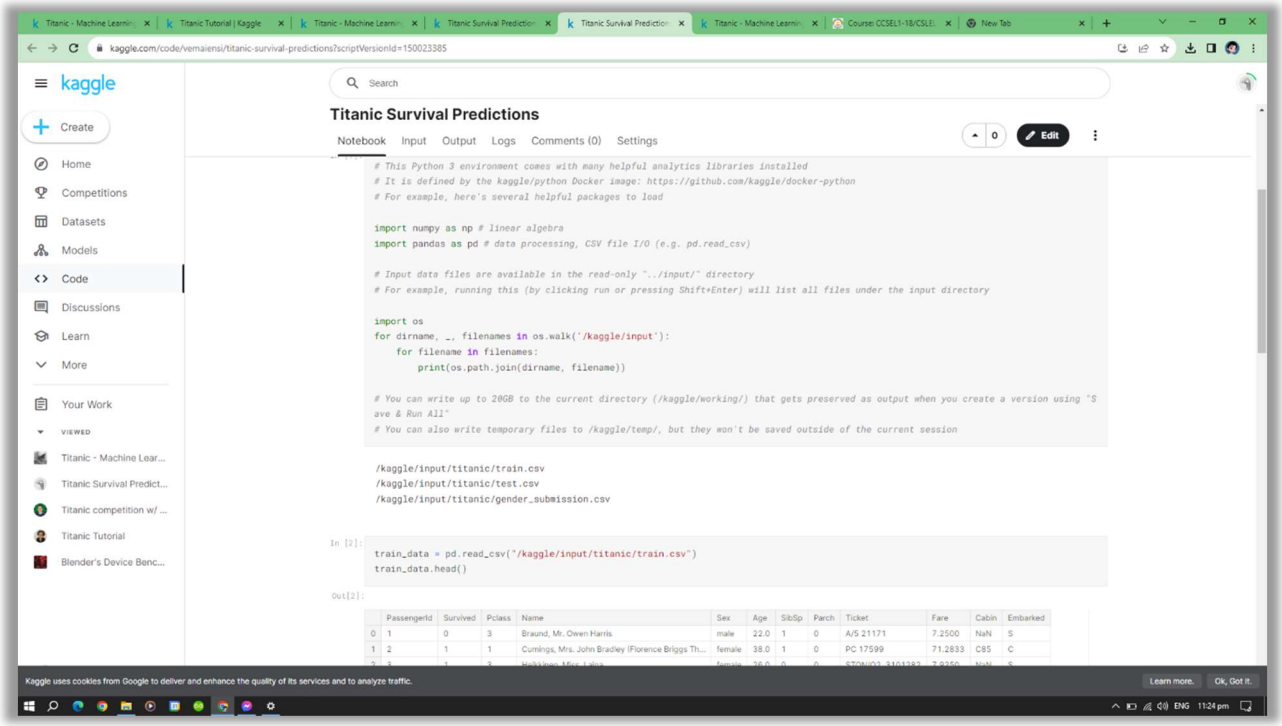

Download Certificate | Share to Twitter | Share to LinkedIn

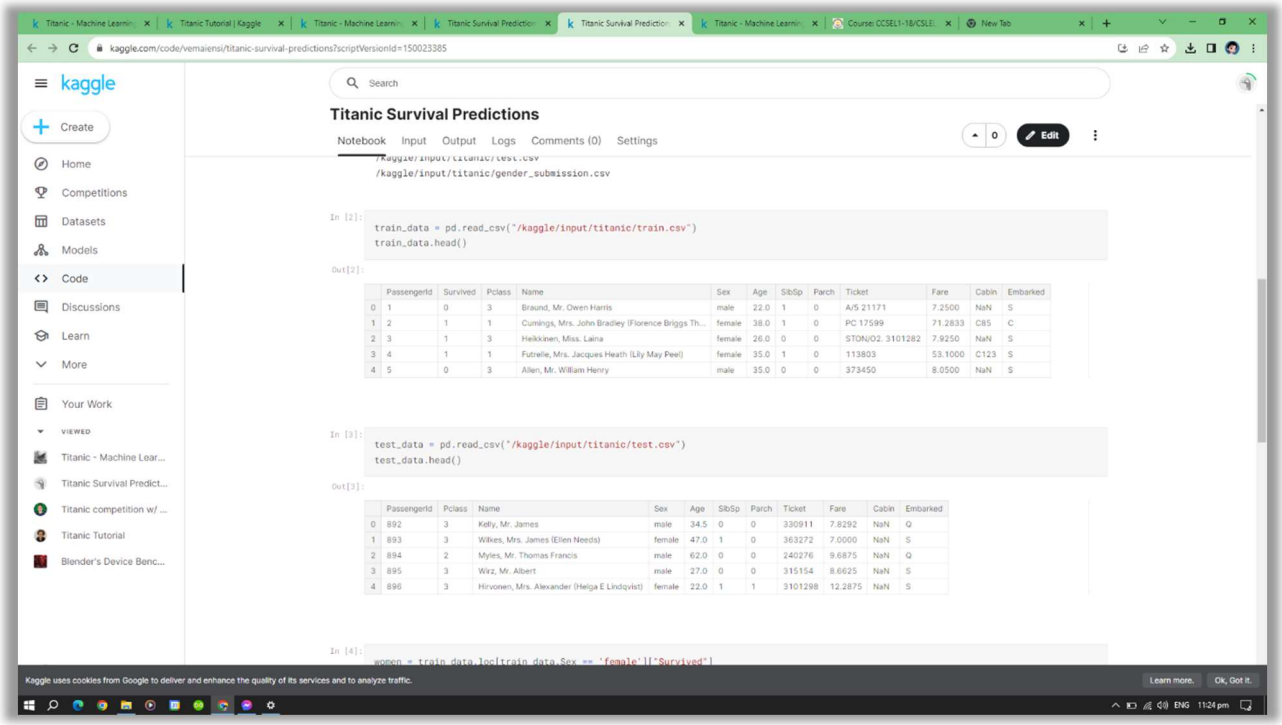
View all Kaggle Learn Courses

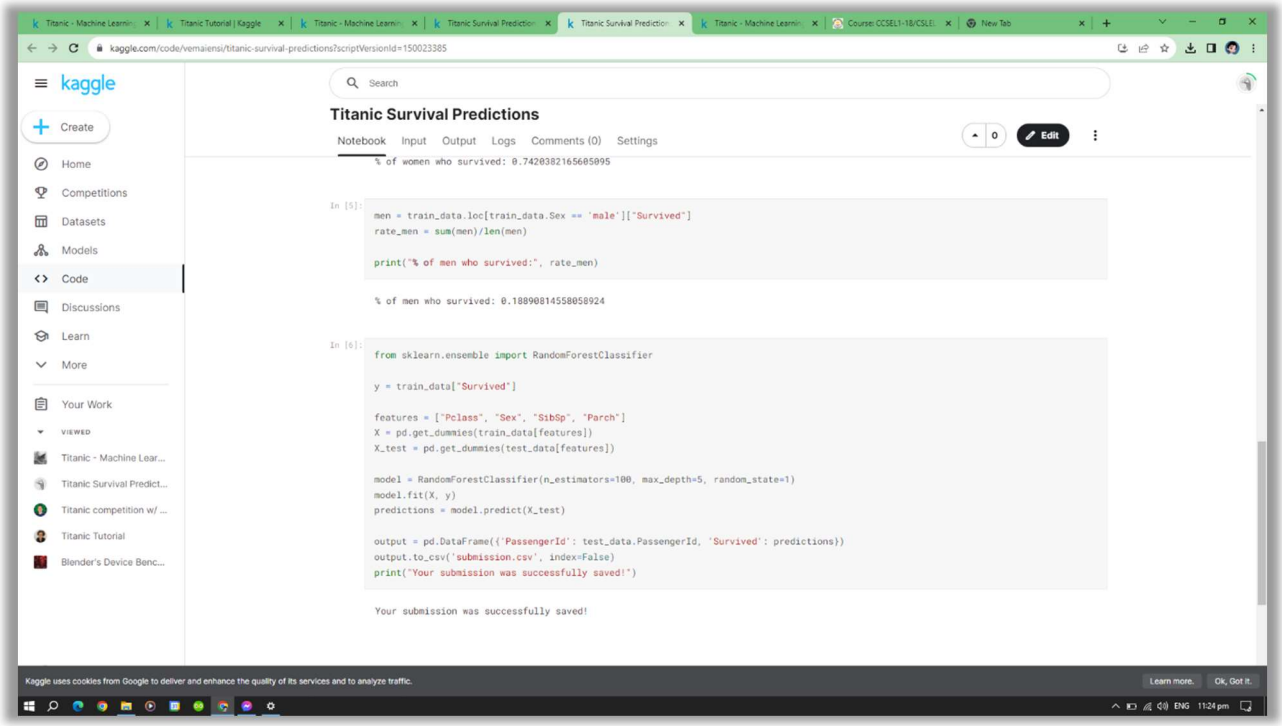
# KAGGLE COMPETITION – TITANIC SURVIVAL PREDICTION

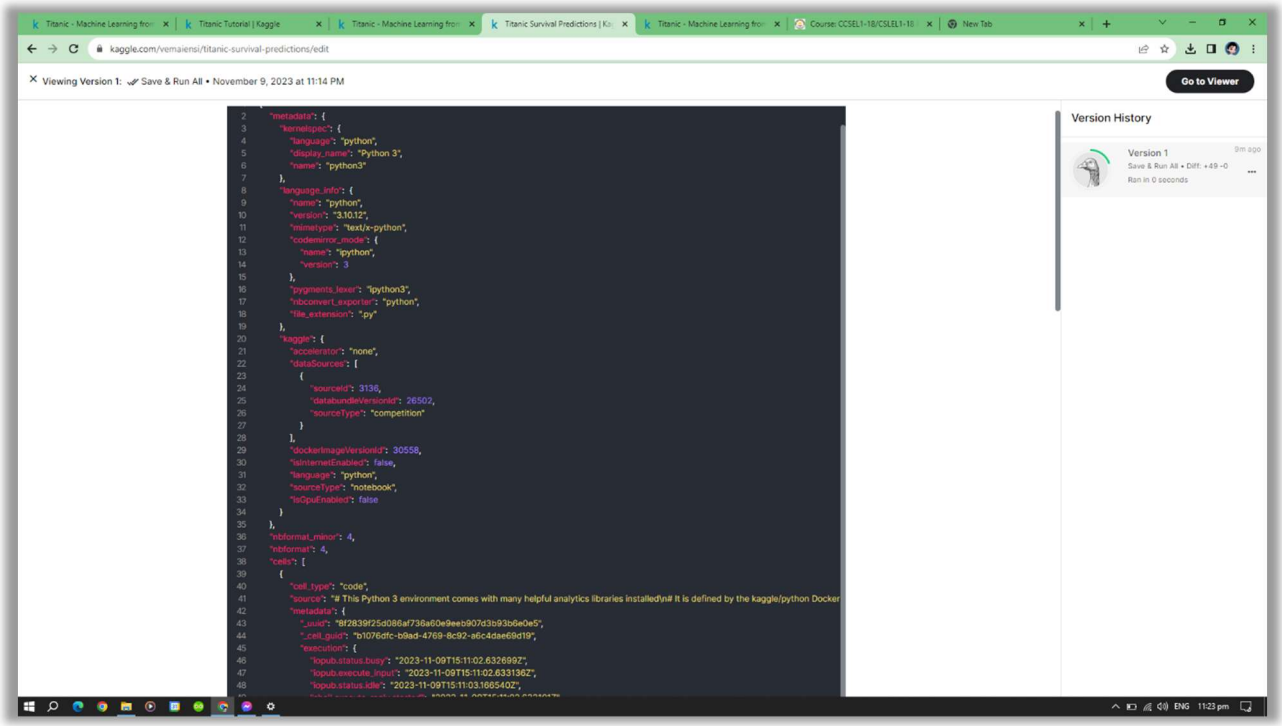


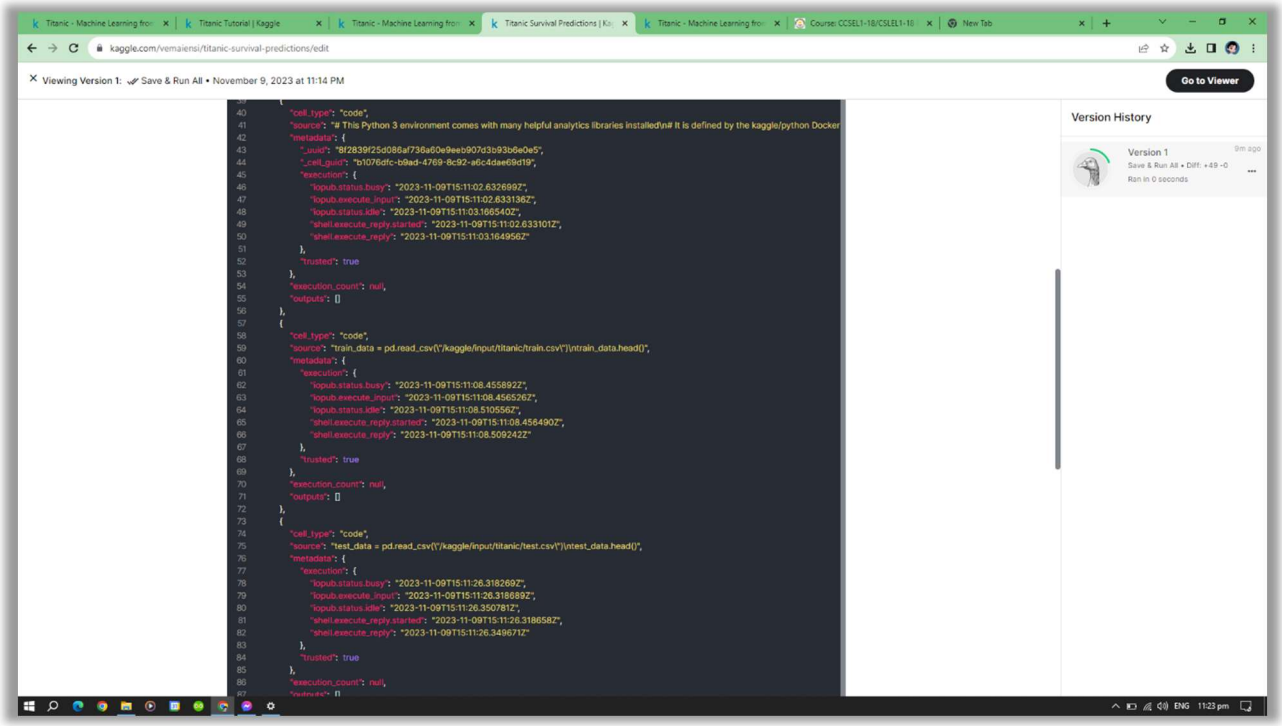



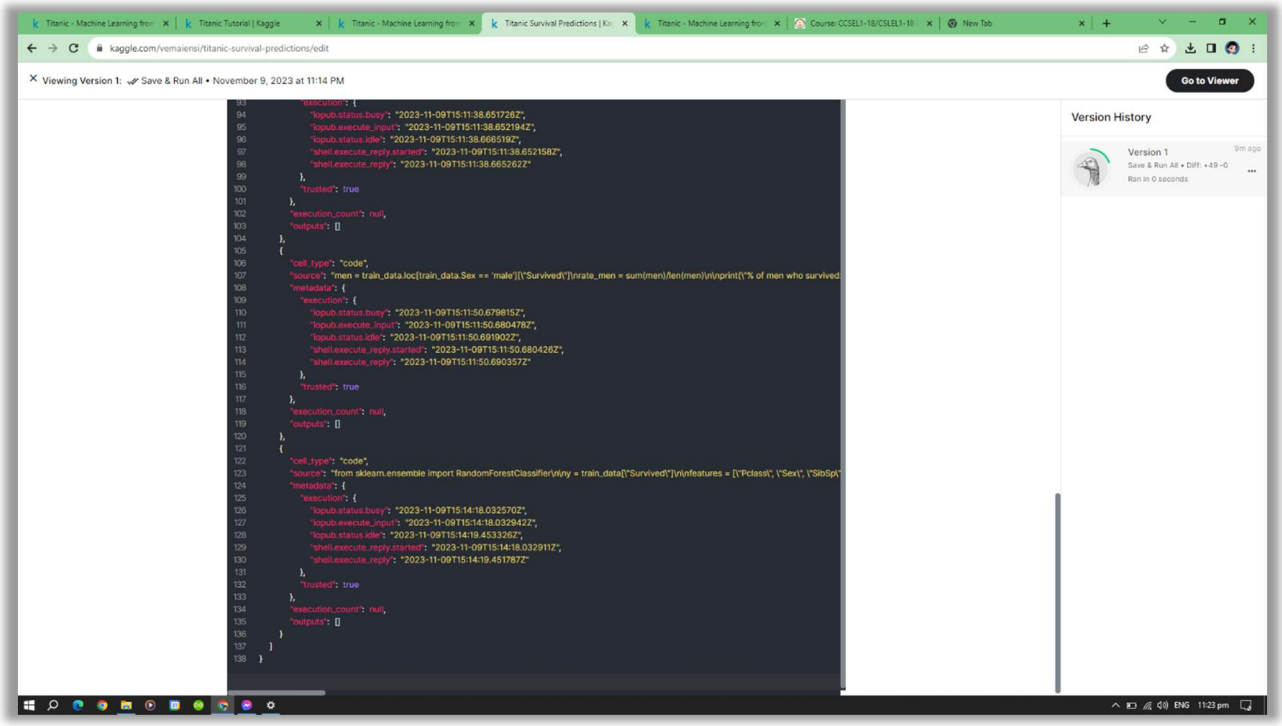

kaggle
Titanic Survival Predictions
Notebook Input Output Logs Comments (0) Settings
Edit
/kaggle/input/titanic/train.csv
/kaggle/input/titanic/test.csv
/kaggle/input/titanic/gender_submission.csv
In [2]:
train_data = pd.read_csv("/kaggle/input/titanic/train.csv")
train_data.head()
Out[2]:
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more.
Ok, Got it.

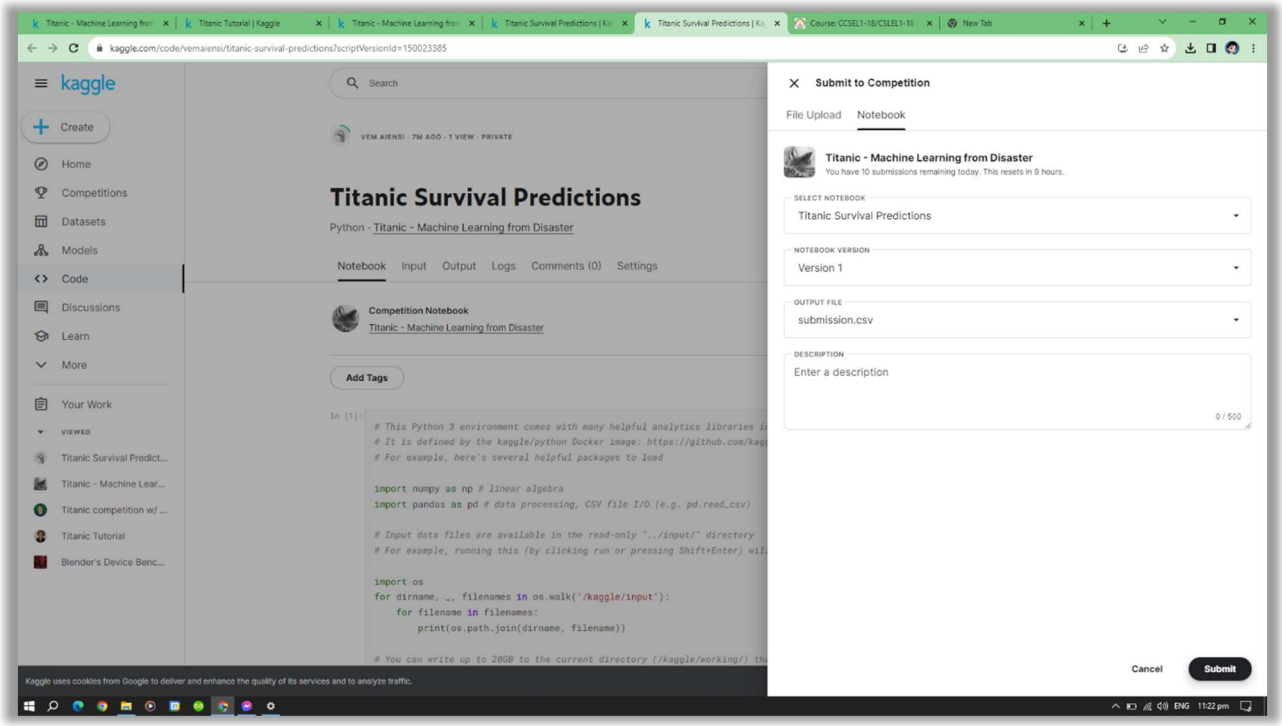

kaggle
Titanic Survival Predictions
Notebook Input Output Logs Comments (0) Settings
Edit
/kaggle/input/titanic/gender_submission.csv
In [2]:
train_data = pd.read_csv("/kaggle/input/titanic/train.csv")
train_data.head()
Out[2]:
In [3]:
test_data = pd.read_csv("/kaggle/input/titanic/test.csv")
test_data.head()
Out[3]:
In [4]:
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more.
Ok, Got it.

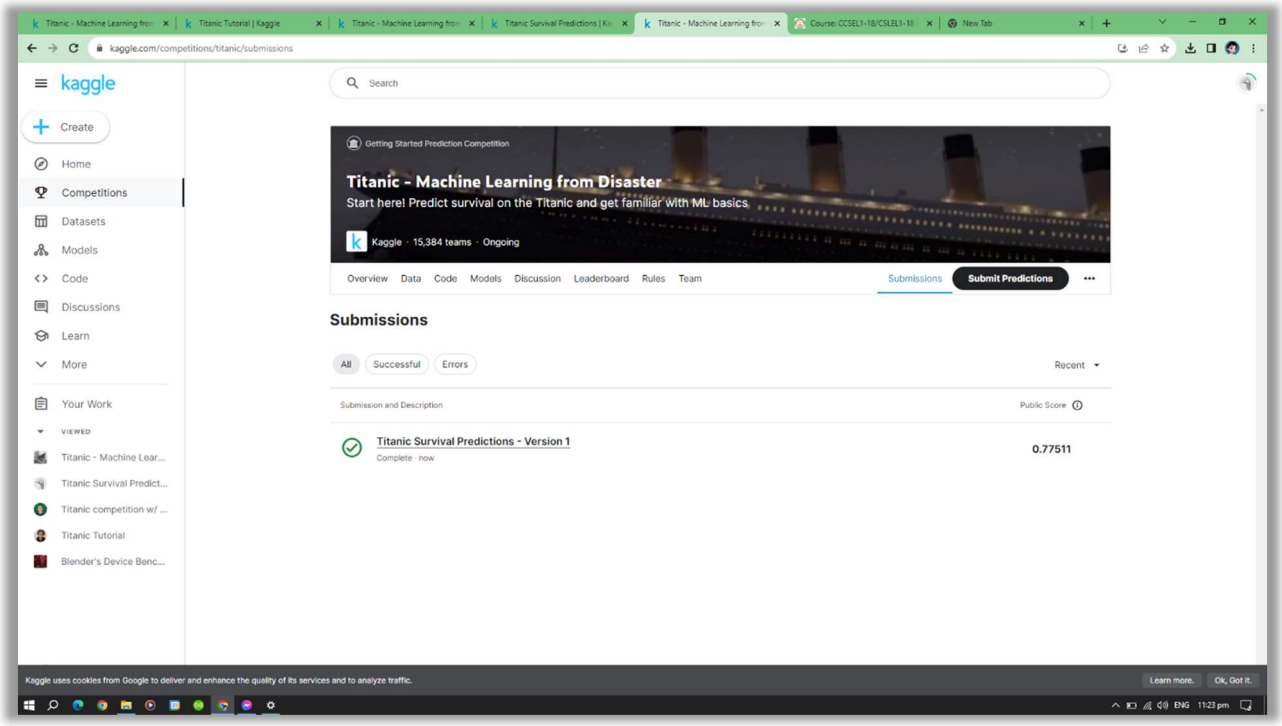

kaggle
Create
Home
Competitions
Datasets
Models
Code
Discussions
Learn
More
Your Work
Titanic Survival Predictions
Notebook
Input
Output
Logs
Comments (0)
Settings
Edit
% of women who survived: 0.7420382165605095
In [5]:
men = train_data.loc[train_data.Sex == 'male']["Survived"]
rate_men = sum(men)/len(men)
print("% of men who survived:", rate_men)
% of men who survived: 0.18890814558058924
In [6]:
from sklearn.ensemble import RandomForestClassifier
y = train_data["Survived"]
features = ["Pclass", "Sex", "SibSp", "Parch"]
X = pd.get_dummies(train_data[features])
X_test = pd.get_dummies(test_data[features])
model = RandomForestClassifier(n_estimators=100, max_depth=5, random_state=1)
model.fit(X, y)
predictions = model.predict(X_test)
output = pd.DataFrame({'PassengerId': test_data.PassengerId, 'Survived': predictions})
output.to_csv('submission.csv', index=False)
print("Your submission was successfully saved!")
Your submission was successfully saved!
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more.
Ok, Got it.


Viewing Version 1: Save & Run All • November 9, 2023 at 11:14 PM
Go to Viewer
Version History
Version 1
Save & Run All • Diff: +49 -0
Ran in 0 seconds
"metadata": {
"kernelspec": {
"language": "python",
"display_name": "Python 3",
"name": "python3"
},
"language_info": {
"name": "python",
"version": "3.10.12",
"mimetype": "text/x-python",
"codemirror_mode": {
"name": "ipython",
"version": 3
},
"pygments_lexer": "ipython3",
"nbconvert_exporter": "python",
"file_extension": ".py"
},
"kaggle": {
"accelerator": "none",
"dataSources": [
{
"sourceId": 3136,
"databundleVersionId": 26502,
"sourceType": "competition"
}
],
"dockerImageVersionId": 30558,
"isInternetEnabled": false,
"language": "python",
"sourceType": "notebook",
"isGpuEnabled": false
}
},
"nbformat_minor": 4,
"nbformat": 4,
"cells": [
{
"cell_type": "code",
"source": "# This Python 3 environment comes with many helpful analytics libraries installed\n# It is defined by the kaggle/python Docker
"metadata": {
"_uuid": "8f2839f25d086af736a60e9eeb907d3b93b6e0e5",
"_cell_guid": "b1076dfc-b9ad-4769-8c92-a6c4dae69d19",
"execution": {
"iopub.status.busy": "2023-11-09T15:11:02.632699Z",
"iopub.execute_input": "2023-11-09T15:11:02.633136Z",
"iopub.status.idle": "2023-11-09T15:11:03.166540Z",


Titanic - Machine Learning from
Titanic Tutorial | Kaggle
Titanic Survival Predictions | Ko
kaggle.com/vemaiensi/titanic-survival-predictions/edit
Viewing Version 1: Save & Run All • November 9, 2023 at 11:14 PM
Go to Viewer
Version History
Version 1
9m ago
Save & Run All • Diff: +49 -0
Ran in 0 seconds
ENG
11:23 pm


Titanic - Machine Learning from
Titanic Tutorial | Kaggle
Titanic Survival Predictions | Ko
kaggle.com/vemaiensi/titanic-survival-predictions/edit
Viewing Version 1: Save & Run All • November 9, 2023 at 11:14 PM
Go to Viewer
Version History
Version 1
9m ago
Save & Run All • Diff: +49 -0
Ran in 0 seconds
ENG
11:23 pm


kaggle
Create
Home
Competitions
Datasets
Models
Code
Discussions
Learn
More
Your Work
Titanic Survival Predictions
Python · Titanic - Machine Learning from Disaster
Notebook Input Output Logs Comments (0) Settings
Add Tags
Submit to Competition
File Upload
Notebook
Titanic - Machine Learning from Disaster
You have 10 submissions remaining today. This resets in 9 hours.
SELECT NOTEBOOK
Titanic Survival Predictions
NOTEBOOK VERSION
Version 1
OUTPUT FILE
submission.csv
DESCRIPTION
Enter a description
Cancel
Submit


kaggle
Create
Home
Competitions
Datasets
Models
Code
Discussions
Learn
More
Your Work
Getting Started Prediction Competition
Titanic - Machine Learning from Disaster
Start here! Predict survival on the Titanic and get familiar with ML basics
Kaggle · 15,384 teams · Ongoing
Overview Data Code Models Discussion Leaderboard Rules Team
Submissions
Submit Predictions
Submissions
All Successful Errors
Recent
Submission and Description
Public Score
Titanic Survival Predictions - Version 1
Complete · now
0.77511