TUGAS MACHINE LEARNING

PEMBELAJARAN PHYTON

DISUSUN OLEH :

FATHUR RAMADHAN FAHMI

2311533012

DOSEN PENGAMPU :

Afdhal Dinilhak, M.Kom.

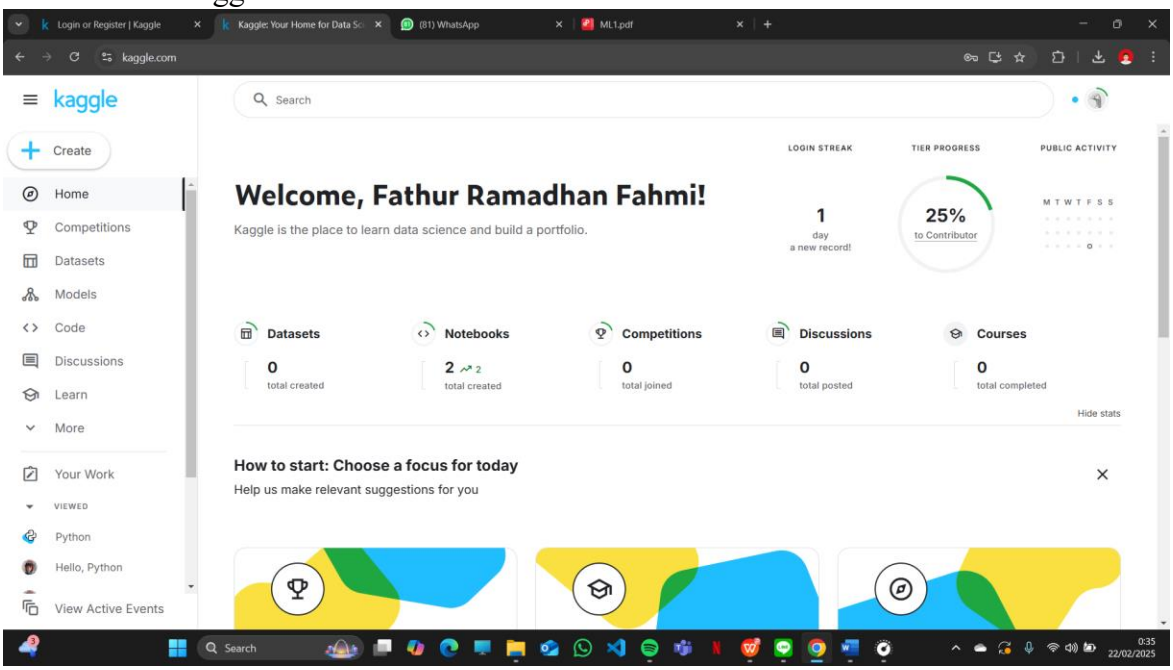
PROGRAM STUDI INFORMATIKA

FAKULTAS TEKNOLOGI INFORMASI

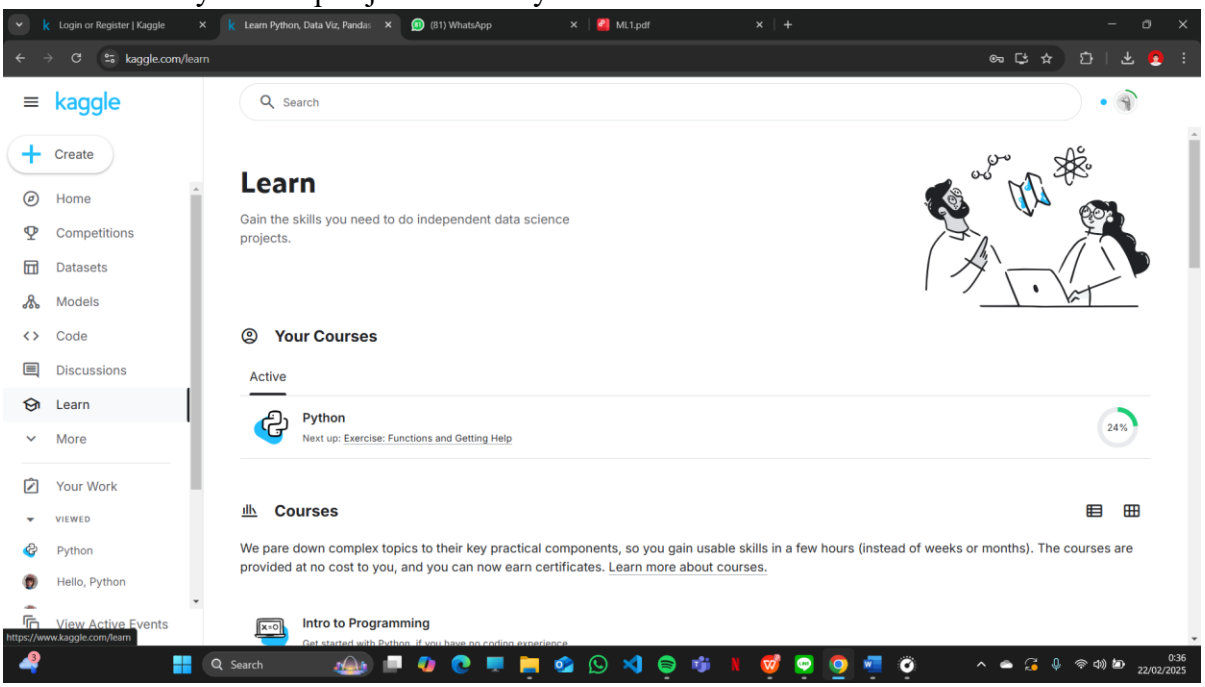
UNIVERSITAS ANDALAS

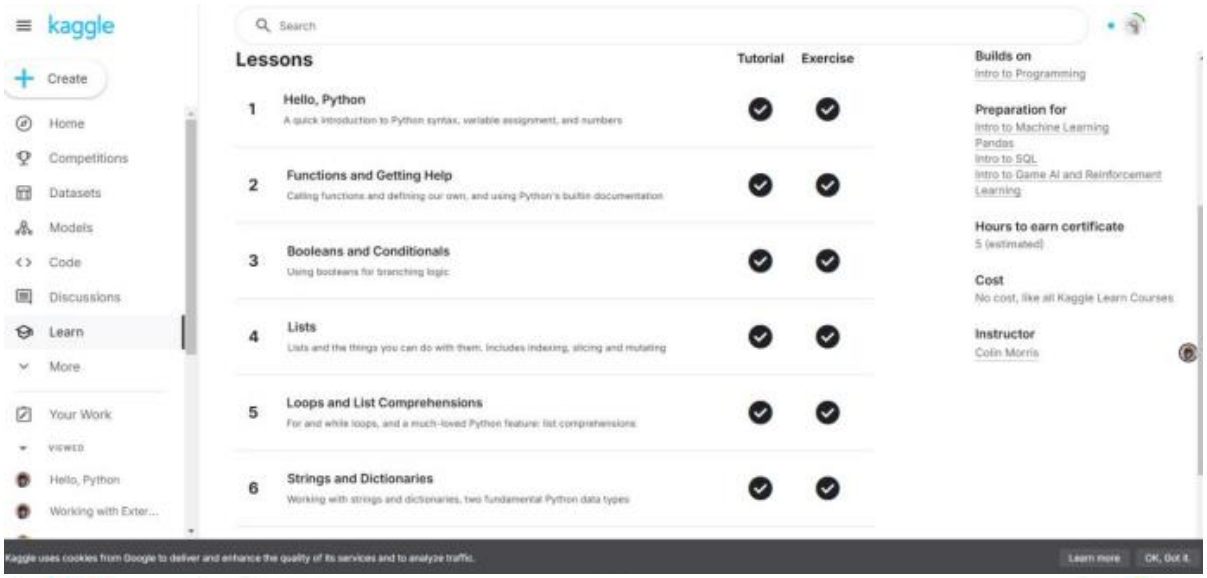
2025

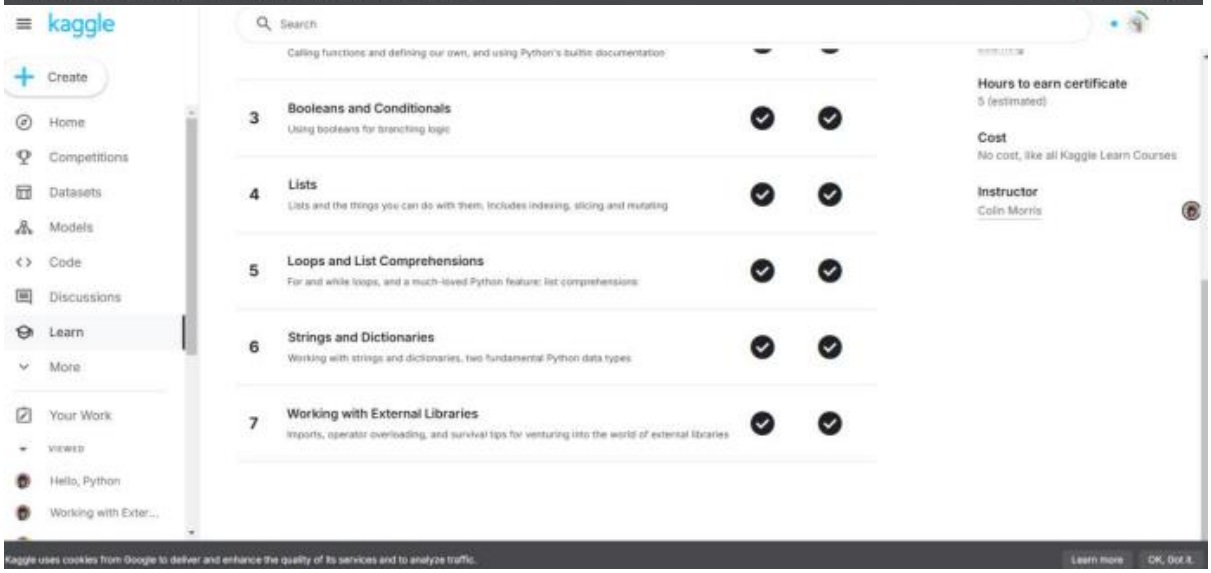
1. Buat akun di Kaggle



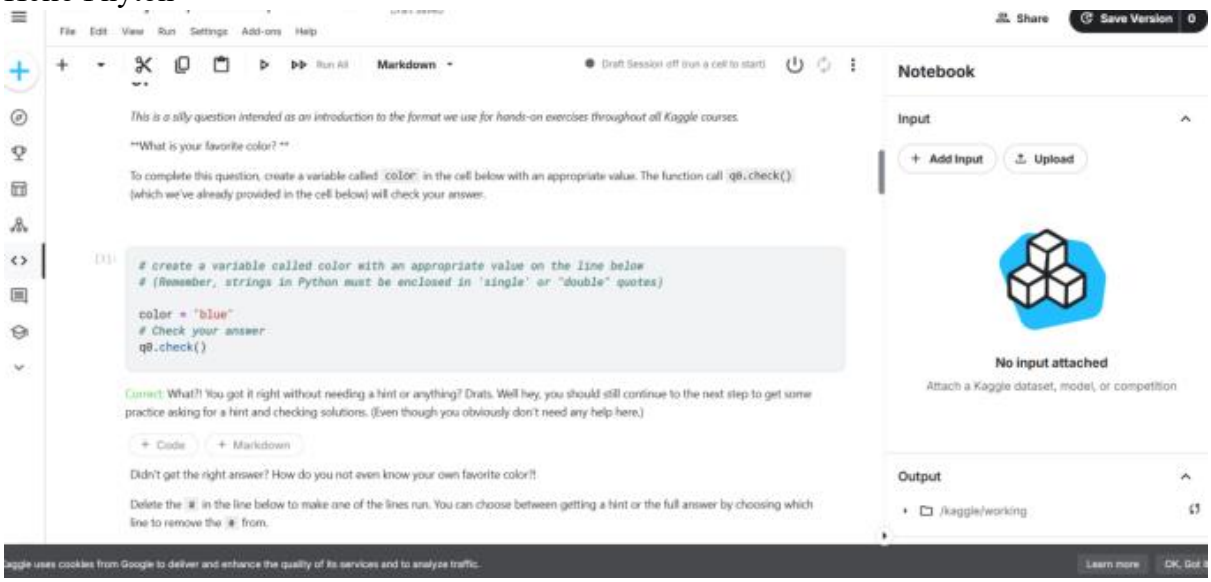
2. Buka Learn Phyton dan pelajari Course nya



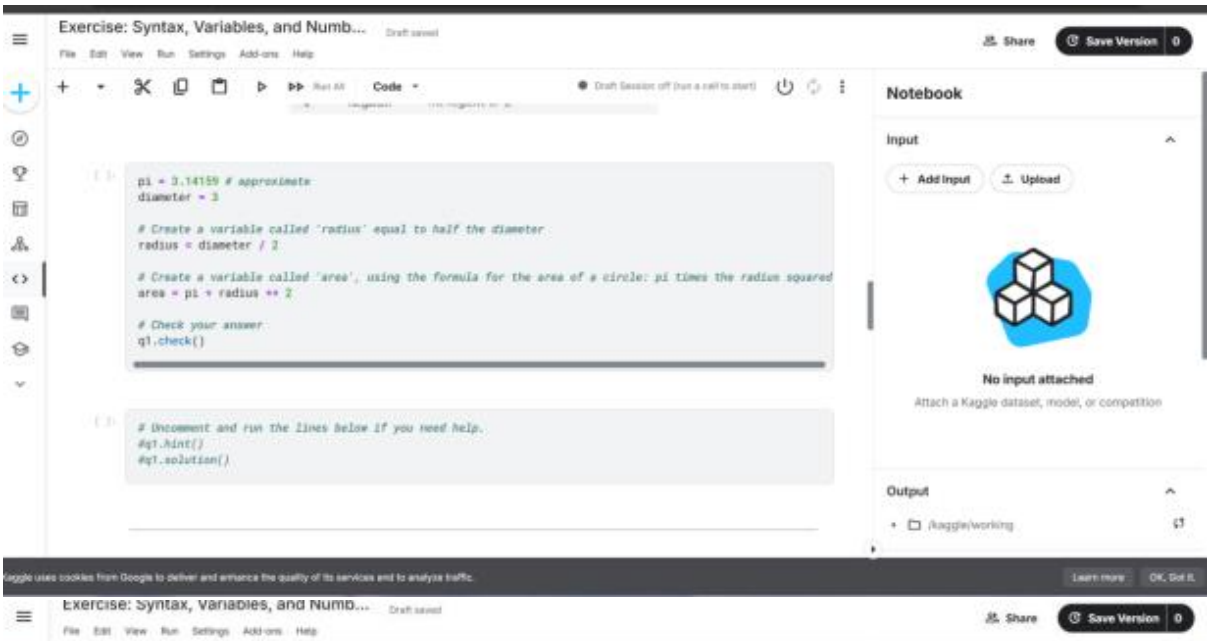
3. 

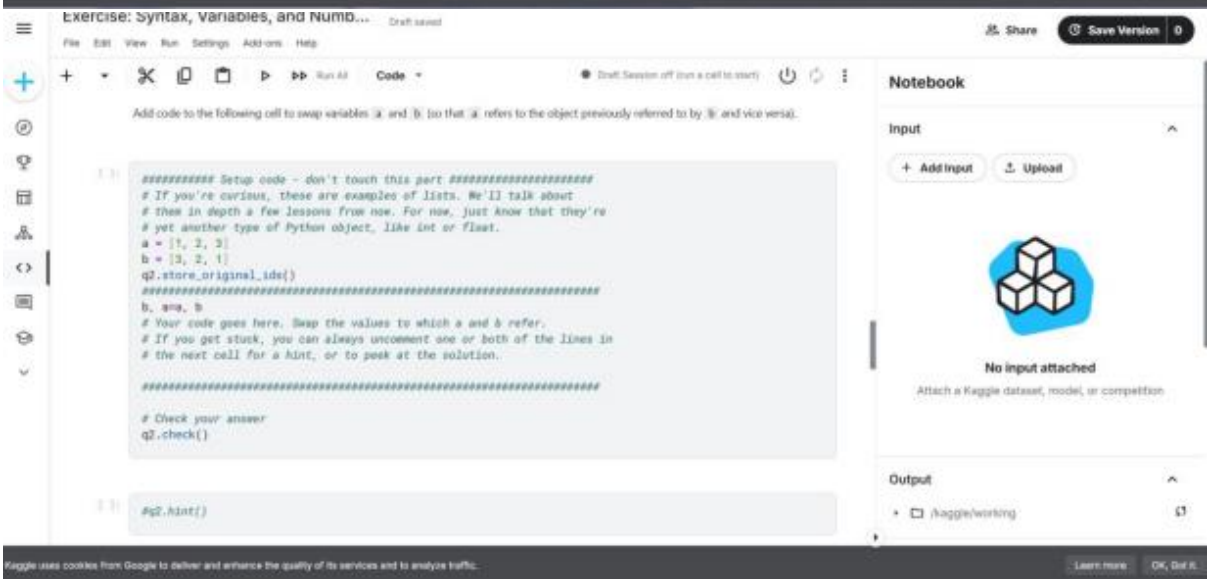
4. 

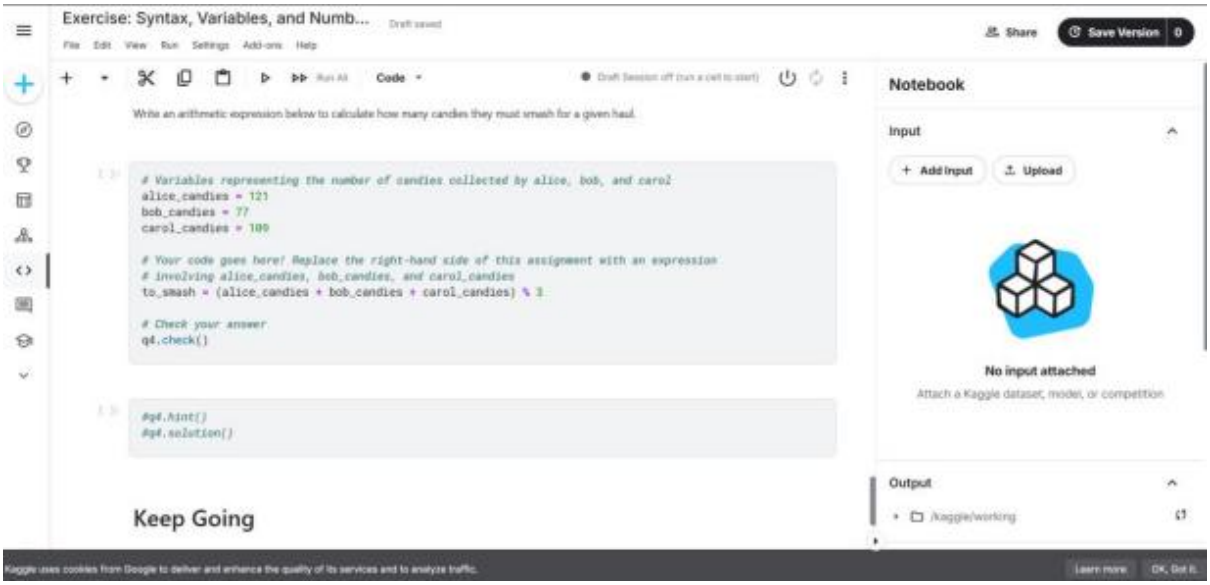
5. Hello Phyton



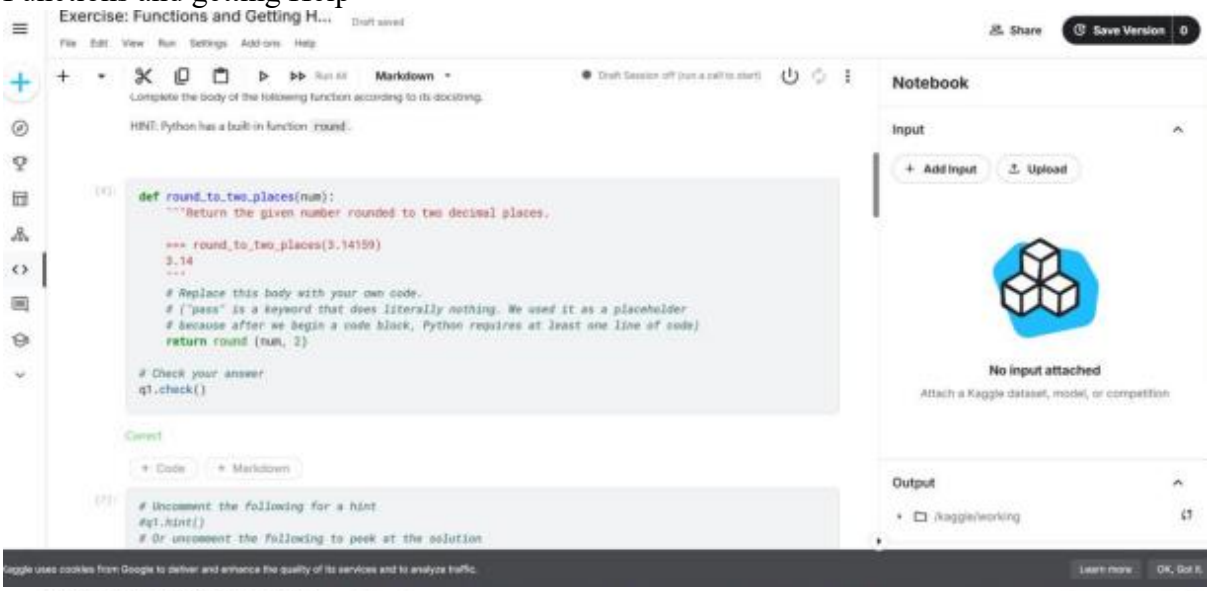

Exercise: Syntax, Variables, and Numb...
Share
Save Version 0
Code
(which we've already provided in the cell below) will check your answer.
# create a variable called color with an appropriate value on the line below
# (Remember, strings in Python must be enclosed in 'single' or "double" quotes)
color = "blue"
# Check your answer
q0.check()
Didn't get the right answer? How do you not even know your own favorite color?!
Delete the # in the line below to make one of the lines run. You can choose between getting a hint or the full answer by choosing which line to remove the # from.
Removing the # is called uncommenting, because it changes that line from a "comment" which Python doesn't run to code, which Python does run.
#q0.hint()
#q0.solution()
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
Learn more
OK, Got it.

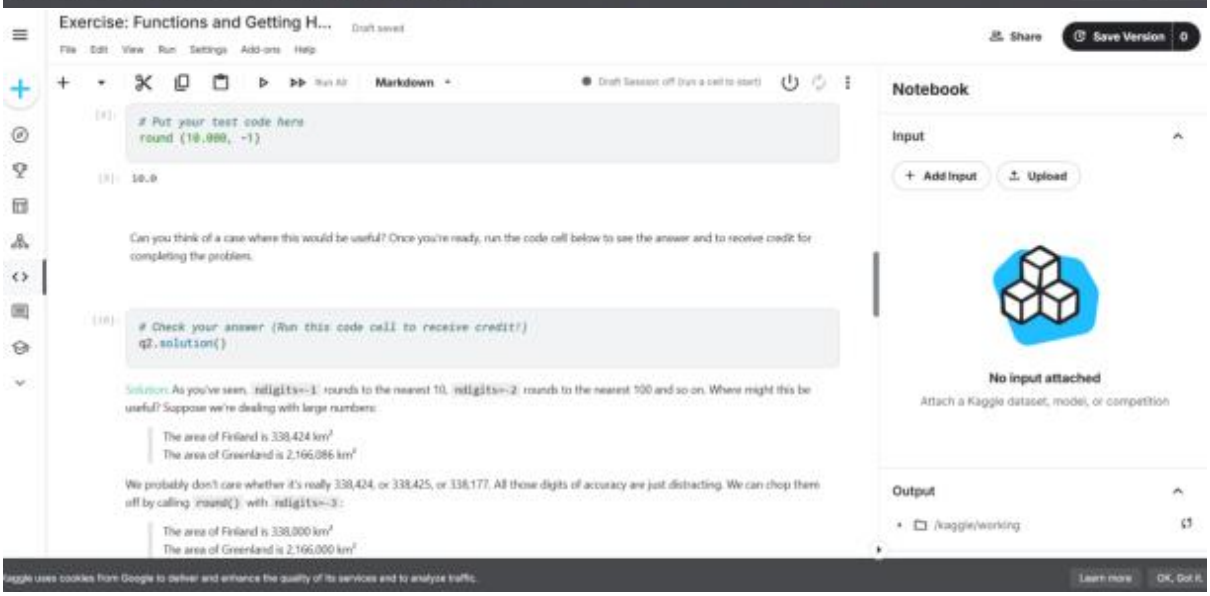

Exercise: Syntax, Variables, and Numb...
Share
Save Version 0
Code
pi = 3.14159 # approximate
diameter = 3
# Create a variable called 'radius' equal to half the diameter
radius = diameter / 2
# Create a variable called 'area', using the formula for the area of a circle: pi times the radius squared
area = pi * radius ** 2
# Check your answer
q1.check()
# Uncomment and run the lines below if you need help.
#q1.hint()
#q1.solution()
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
Learn more
OK, Got it.

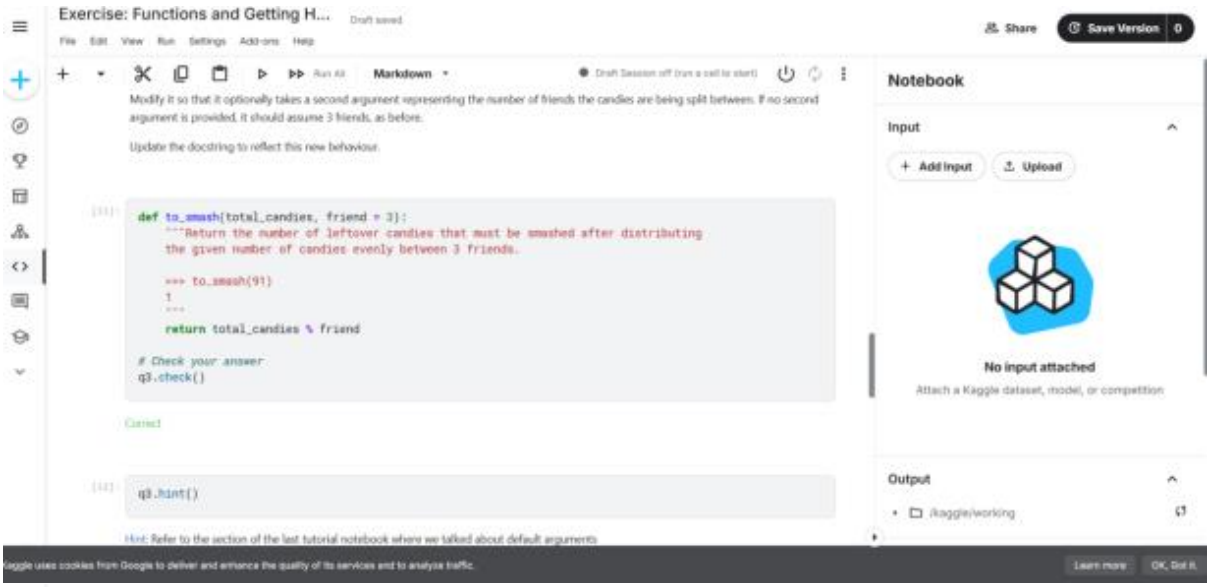

Exercise: Syntax, Variables, and Numb...
Share
Save Version 0
Code
Add code to the following cell to swap variables a and b (so that a refers to the object previously referred to by b and vice versa).
########## Setup code - don't touch this part ####################
# If you're curious, these are examples of lists. We'll talk about
# them in depth a few lessons from now. For now, just know that they're
# yet another type of Python object, like int or float.
a = [1, 2, 3]
b = [3, 2, 1]
q2.store_original_ids()
##
Your code goes here. Swap the values to which a and b refer.
If you get stuck, you can always uncomment one or both of the lines in
the next cell for a hint, or to peek at the solution.
##
# Check your answer
q2.check()
#q2.hint()
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
Learn more
OK, Got it.

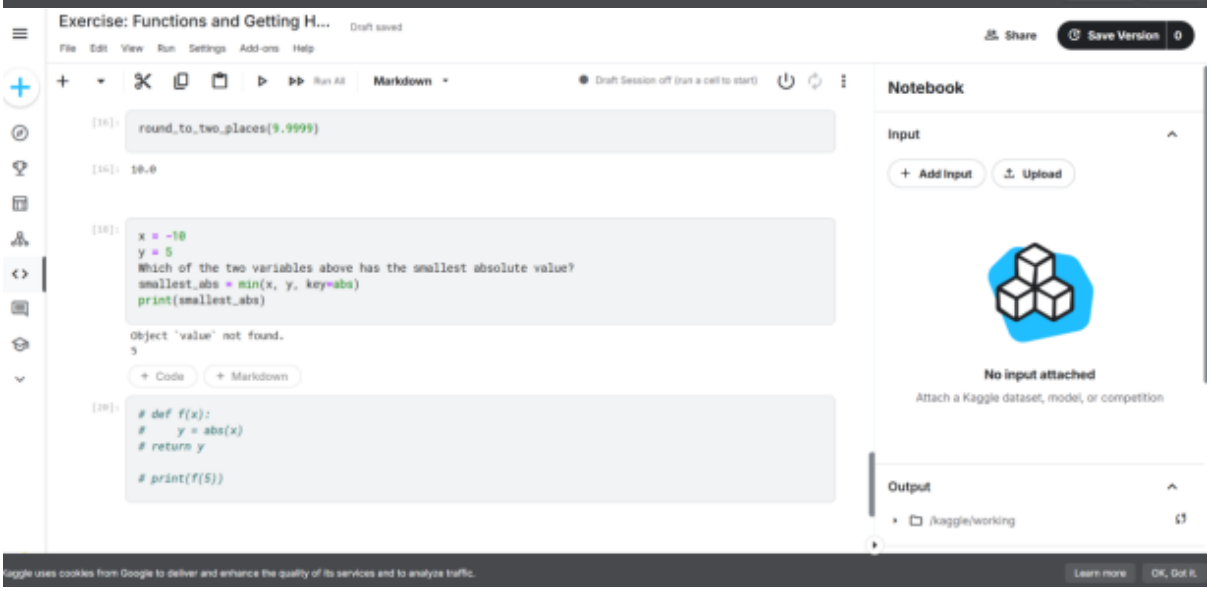


6. Functions and getting Help

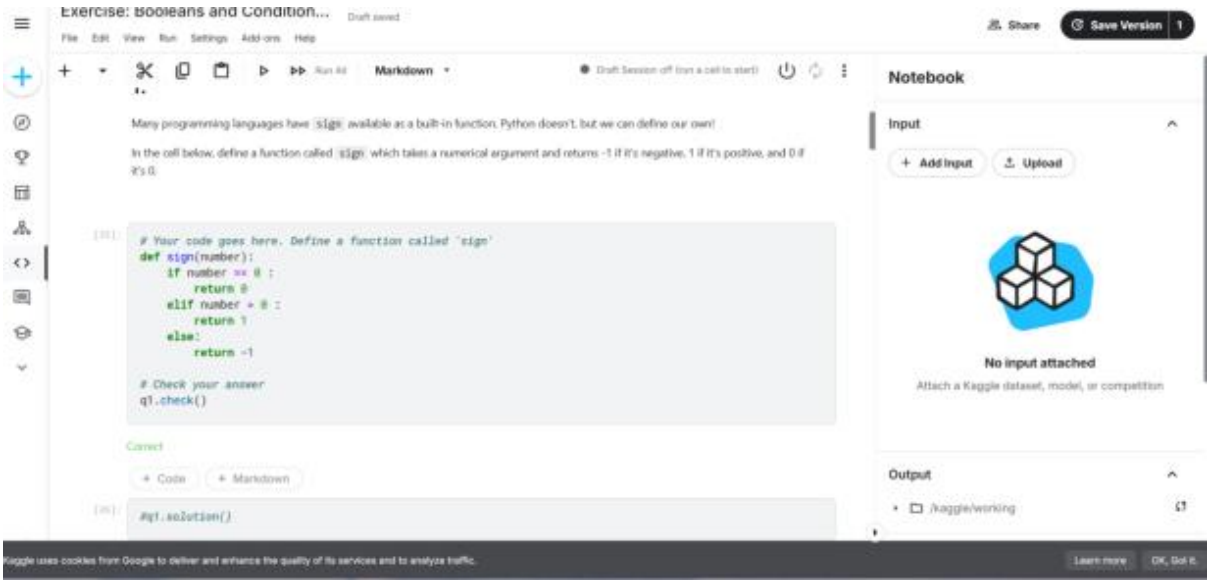


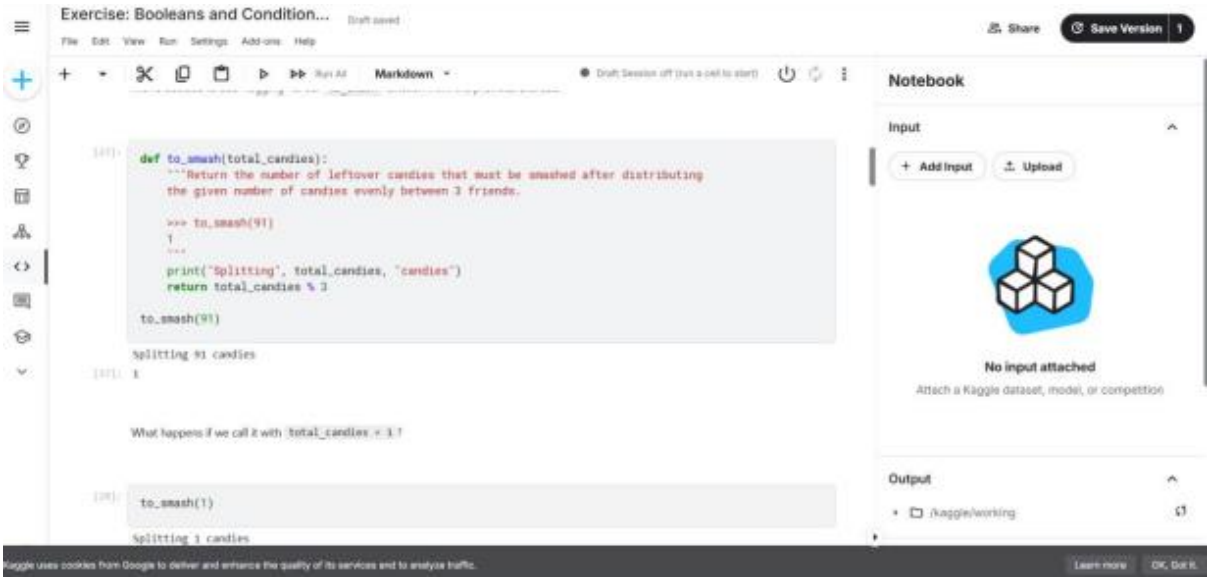


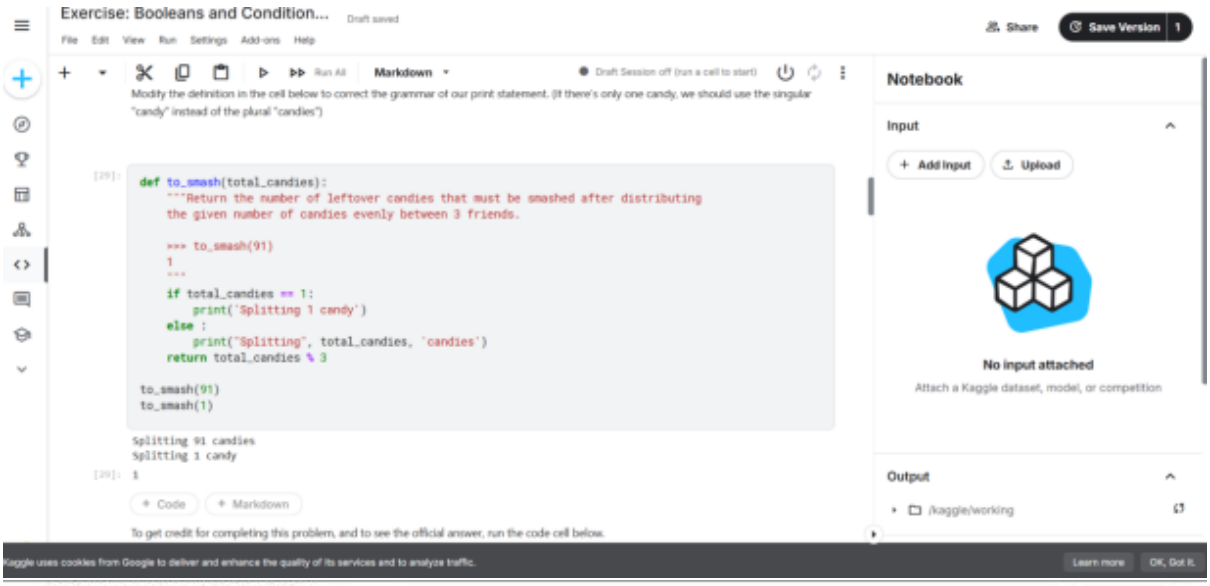


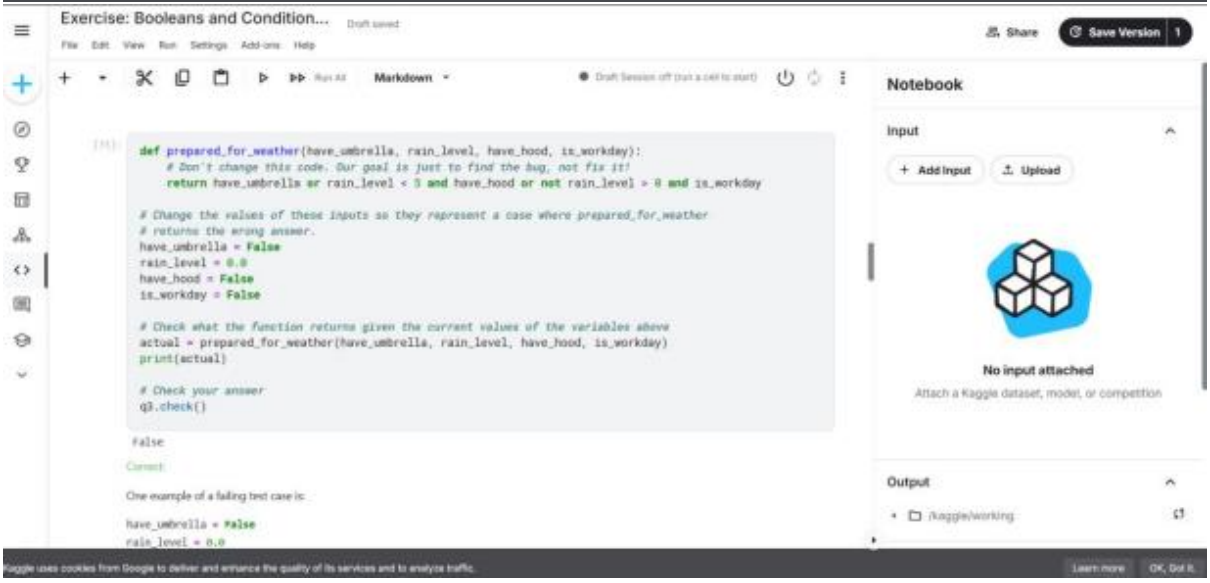


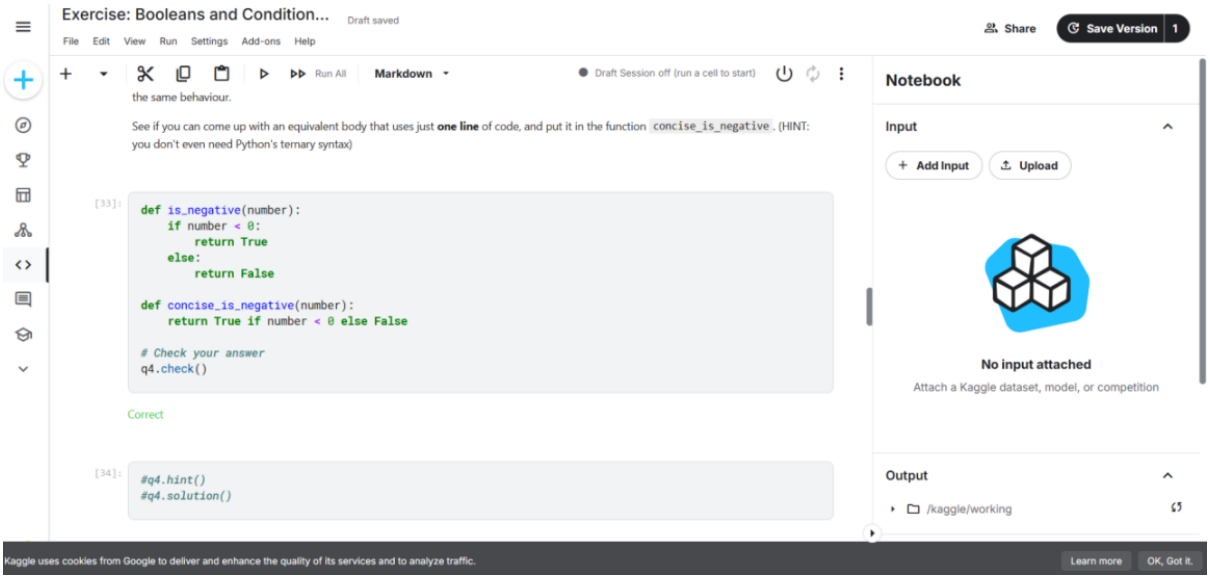
7. Booleans and Conditionals

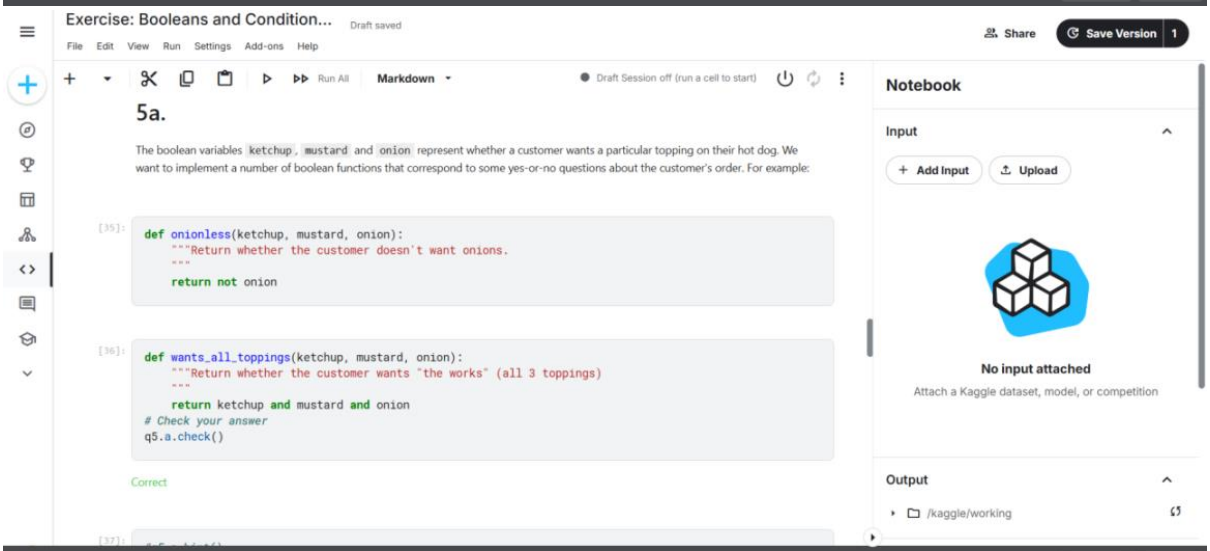


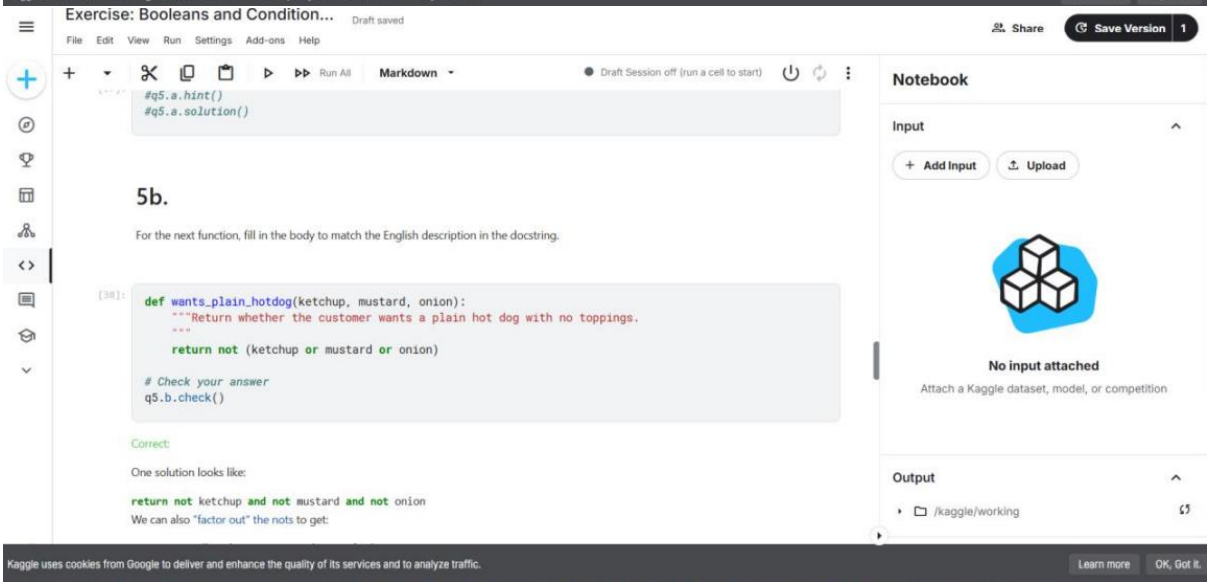

Exercise: Booleans and Condition...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 1
Markdown
Draft Session off (run a cell to start)
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
def to_smash(total_candies):
"""Return the number of leftover candies that must be smashed after distributing
the given number of candies evenly between 3 friends.
>>> to_smash(91)
1
"""
print("Splitting", total_candies, "candies")
return total_candies % 3
to_smash(91)
Splitting 91 candies
1
What happens if we call it with total_candies = 1 ?
to_smash(1)
Splitting 1 candies
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.

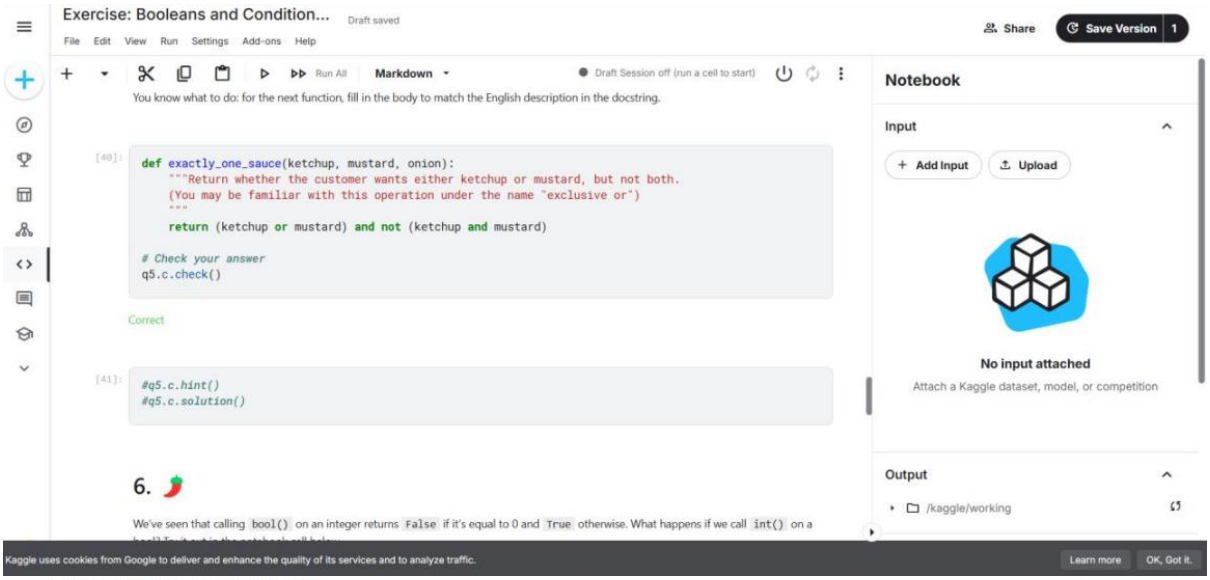

Exercise: Booleans and Condition...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 1
Markdown
Draft Session off (run a cell to start)
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
Modify the definition in the cell below to correct the grammar of our print statement. (If there's only one candy, we should use the singular "candy" instead of the plural "candies")
def to_smash(total_candies):
"""Return the number of leftover candies that must be smashed after distributing
the given number of candies evenly between 3 friends.
>>> to_smash(91)
1
"""
if total_candies == 1:
print('Splitting 1 candy')
else :
print("Splitting", total_candies, 'candies')
return total_candies % 3
to_smash(91)
to_smash(1)
Splitting 91 candies
Splitting 1 candy
1
+ Code
+ Markdown
To get credit for completing this problem, and to see the official answer, run the code cell below.
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.

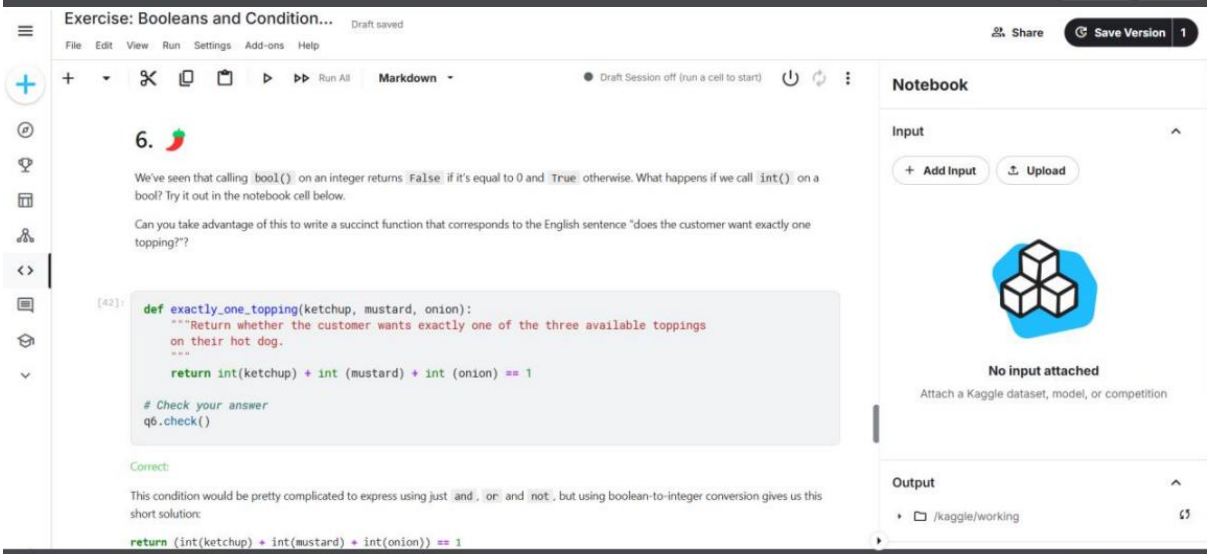

Exercise: Booleans and Condition...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 1
Markdown
Draft Session off (run a cell to start)
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
def prepared_for_weather(have_umbrella, rain_level, have_hood, is_workday):
# Don't change this code. Our goal is just to find the bug, not fix it!
return have_umbrella or rain_level < 3 and have_hood or not rain_level > 0 and is_workday
# Change the values of these inputs so they represent a case where prepared_for_weather
# returns the wrong answer.
have_umbrella = False
rain_level = 0.0
have_hood = False
is_workday = False
# Check what the function returns given the current values of the variables above
actual = prepared_for_weather(have_umbrella, rain_level, have_hood, is_workday)
print(actual)
# Check your answer
q3.check()
False
Correct
One example of a failing test case is:
have_umbrella = False
rain_level = 0.0
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.

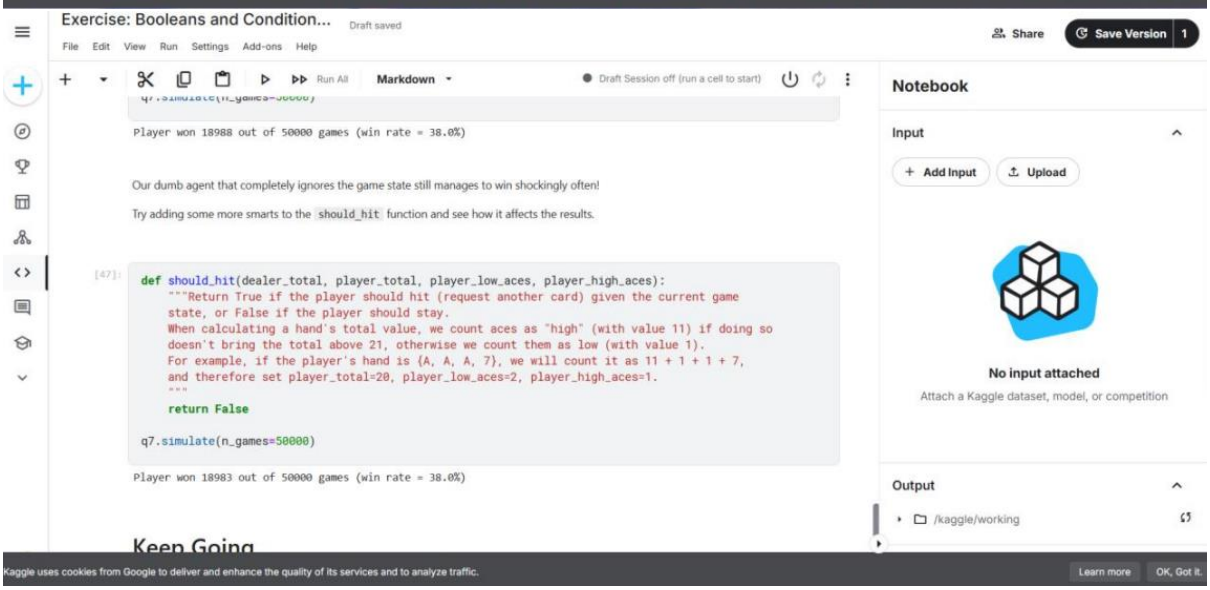

Exercise: Booleans and Condition...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 1
Run All
Markdown
Draft Session off (run a cell to start)
the same behaviour.
See if you can come up with an equivalent body that uses just one line of code, and put it in the function concise_is_negative. (HINT: you don't even need Python's ternary syntax)
[33]:
def is_negative(number):
if number < 0:
return True
else:
return False
def concise_is_negative(number):
return True if number < 0 else False
# Check your answer
q4.check()
Correct
[34]:
#q4.hint()
#q4.solution()
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.


Exercise: Booleans and Condition...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 1
Run All
Markdown
Draft Session off (run a cell to start)
5a.
The boolean variables ketchup, mustard and onion represent whether a customer wants a particular topping on their hot dog. We want to implement a number of boolean functions that correspond to some yes-or-no questions about the customer's order. For example:
[35]:
def onionless(ketchup, mustard, onion):
"""Return whether the customer doesn't want onions.
"""
return not onion
[36]:
def wants_all_toppings(ketchup, mustard, onion):
"""Return whether the customer wants "the works" (all 3 toppings)
"""
return ketchup and mustard and onion
# Check your answer
q5.a.check()
Correct
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.

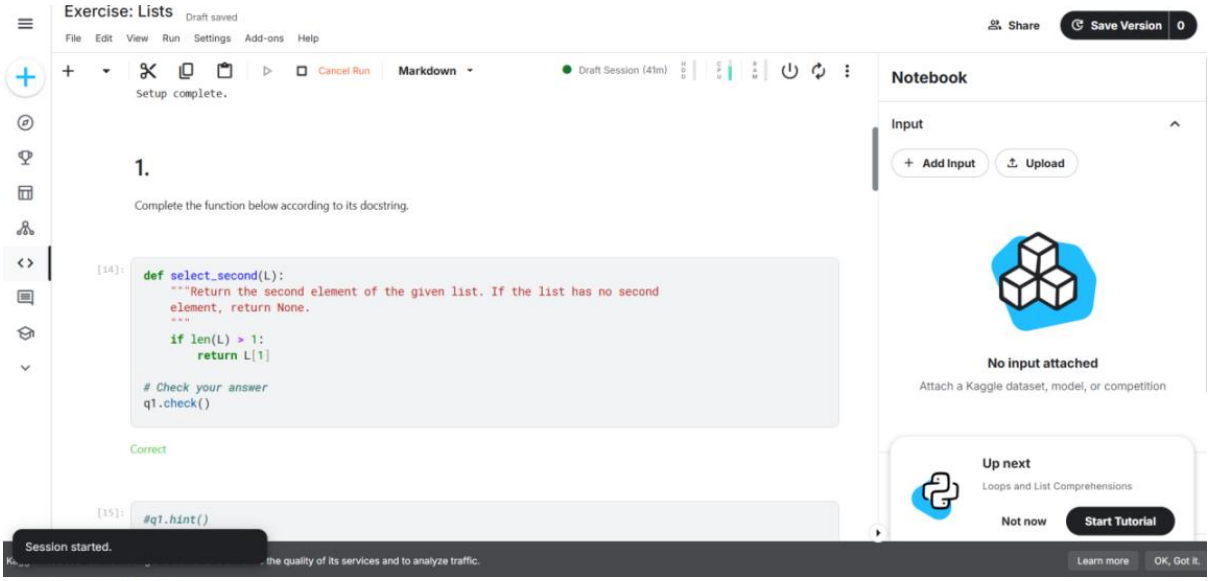

Exercise: Booleans and Condition...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 1
Run All
Markdown
Draft Session off (run a cell to start)
#q5.a.hint()
#q5.a.solution()
5b.
For the next function, fill in the body to match the English description in the docstring.
[38]:
def wants_plain_hotdog(ketchup, mustard, onion):
"""Return whether the customer wants a plain hot dog with no toppings.
"""
return not (ketchup or mustard or onion)
# Check your answer
q5.b.check()
Correct:
One solution looks like:
return not ketchup and not mustard and not onion
We can also "factor out" the nots to get:
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.

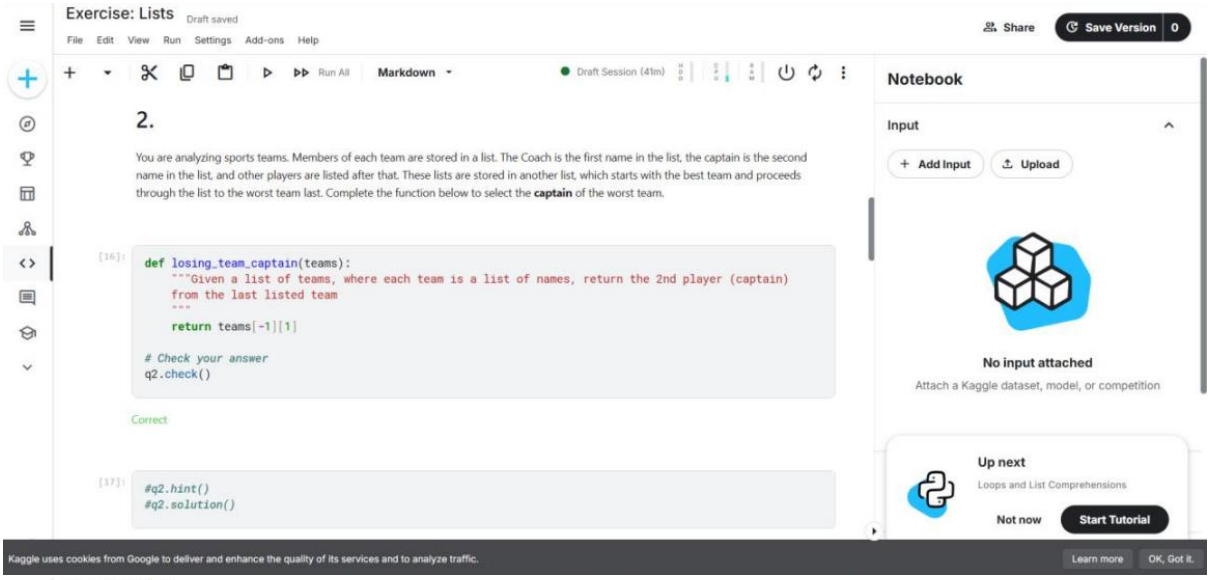

Exercise: Booleans and Condition...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 1
Run All
Markdown
Draft Session off (run a cell to start)
You know what to do: for the next function, fill in the body to match the English description in the docstring.
[40]:
def exactly_one_sauce(ketchup, mustard, onion):
"""Return whether the customer wants either ketchup or mustard, but not both.
(You may be familiar with this operation under the name "exclusive or")
"""
return (ketchup or mustard) and not (ketchup and mustard)
# Check your answer
q5.c.check()
Correct
[41]:
#q5.c.hint()
#q5.c.solution()
6.
We've seen that calling bool() on an integer returns False if it's equal to 0 and True otherwise. What happens if we call int() on a
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.

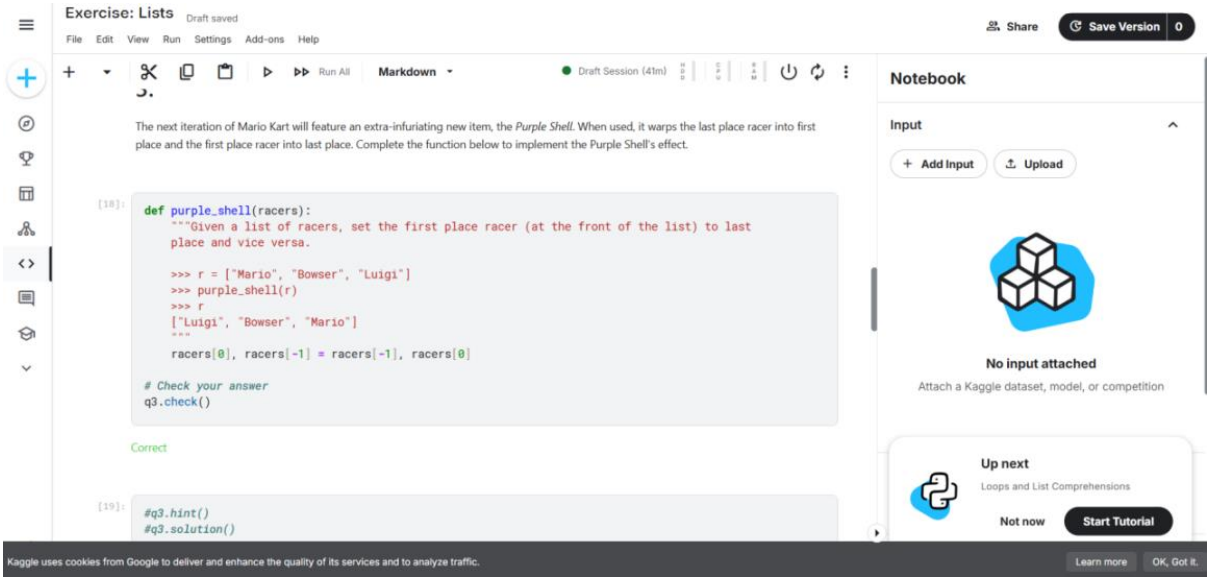

Exercise: Booleans and Condition...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 1
Run All
Markdown
Draft Session off (run a cell to start)
6.
We've seen that calling bool() on an integer returns False if it's equal to 0 and True otherwise. What happens if we call int() on a bool? Try it out in the notebook cell below.
Can you take advantage of this to write a succinct function that corresponds to the English sentence "does the customer want exactly one topping?"?
[42]:
def exactly_one_topping(ketchup, mustard, onion):
"""Return whether the customer wants exactly one of the three available toppings
on their hot dog.
"""
return int(ketchup) + int (mustard) + int (onion) == 1
# Check your answer
q6.check()
Correct:
This condition would be pretty complicated to express using just and , or and not , but using boolean-to-integer conversion gives us this short solution:
return (int(ketchup) + int(mustard) + int(onion)) == 1
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.

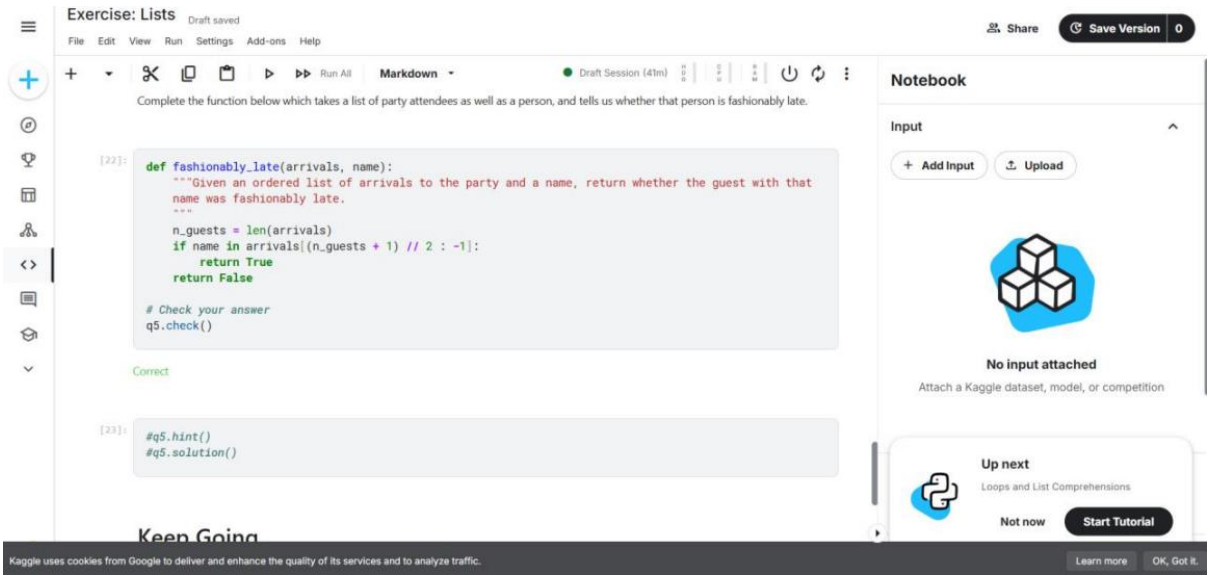

Exercise: Booleans and Condition...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 1
Run All
Markdown
Draft Session off (run a cell to start)
Player won 18988 out of 50000 games (win rate = 38.0%)
Our dumb agent that completely ignores the game state still manages to win shockingly often!
Try adding some more smarts to the should_hit function and see how it affects the results.
[47]:
def should_hit(dealer_total, player_total, player_low_aces, player_high_aces):
"""Return True if the player should hit (request another card) given the current game
state, or False if the player should stay.
When calculating a hand's total value, we count aces as "high" (with value 11) if doing so
doesn't bring the total above 21, otherwise we count them as low (with value 1).
For example, if the player's hand is {A, A, A, 7}, we will count it as 11 + 1 + 1 + 7,
and therefore set player_total=20, player_low_aces=2, player_high_aces=1.
"""
return False
q7.simulate(n_games=50000)
Player won 18983 out of 50000 games (win rate = 38.0%)
Keep Going
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output
/kaggle/working
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.

8. Lists

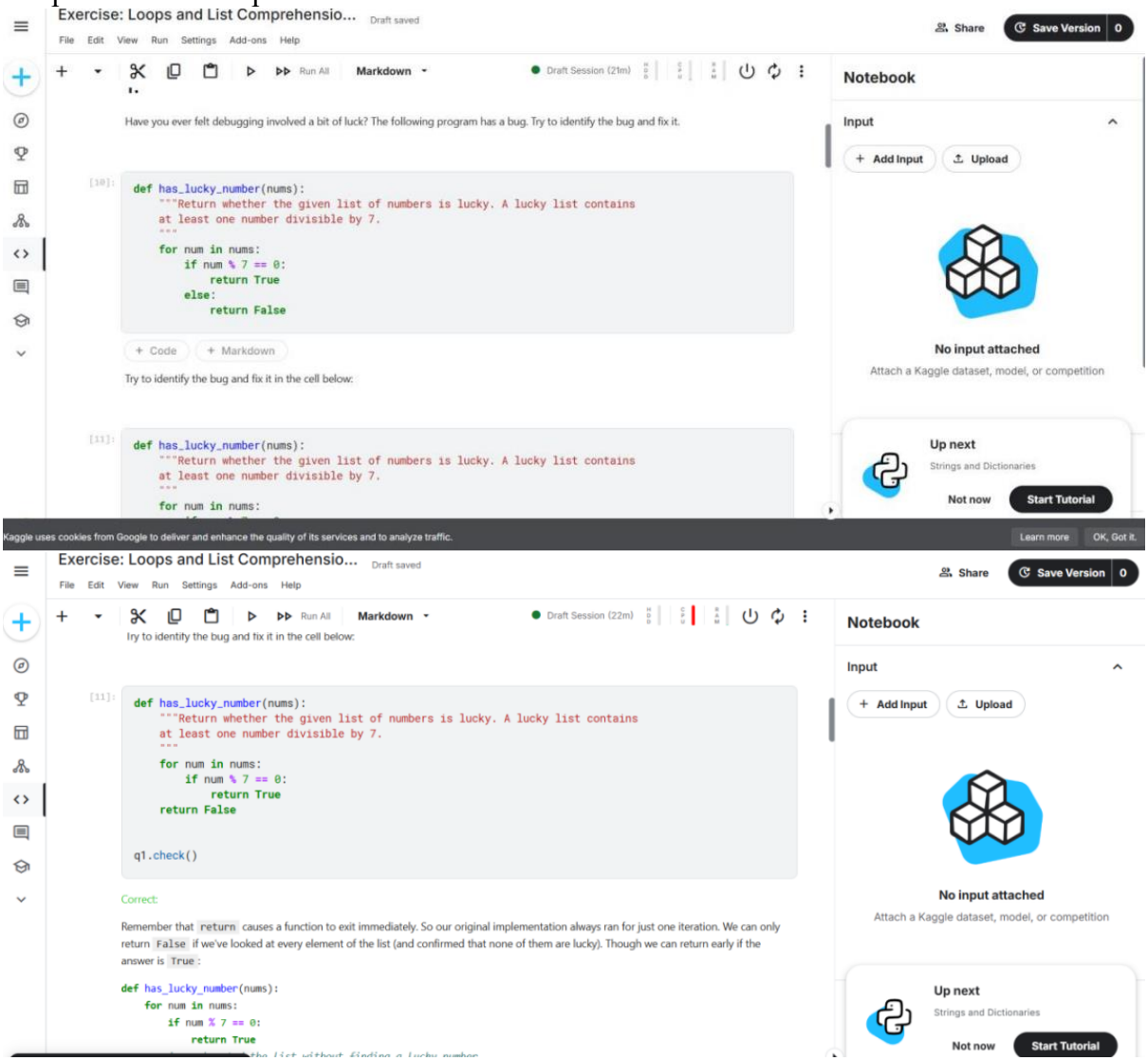

Exercise: Lists
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 0
Cancel Run
Markdown
Draft Session (41m)
Setup complete.
1.
Complete the function below according to its docstring.
[14]:
def select_second(L):
"""Return the second element of the given list. If the list has no second element, return None.
"""
if len(L) > 1:
return L[1]
# Check your answer
q1.check()
Correct
[15]:
#q1.hint()
Session started.
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Up next
Loops and List Comprehensions
Not now
Start Tutorial
Learn more
OK, Got it.

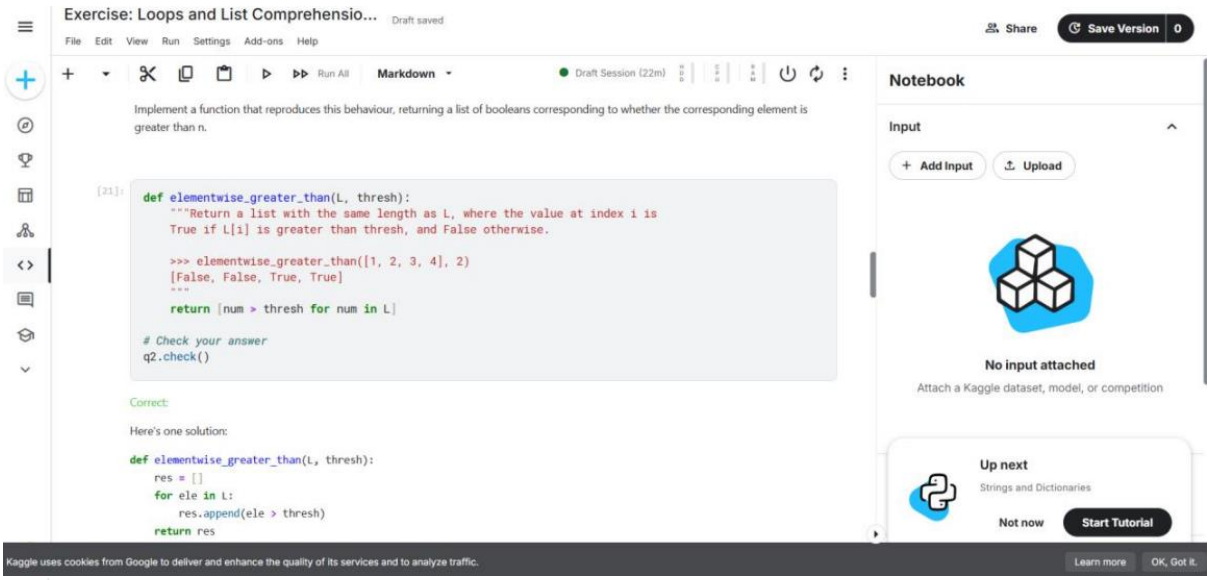

Exercise: Lists
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 0
Run All
Markdown
Draft Session (41m)
2.
You are analyzing sports teams. Members of each team are stored in a list. The Coach is the first name in the list, the captain is the second name in the list, and other players are listed after that. These lists are stored in another list, which starts with the best team and proceeds through the list to the worst team last. Complete the function below to select the captain of the worst team.
[16]:
def losing_team_captain(teams):
"""Given a list of teams, where each team is a list of names, return the 2nd player (captain) from the last listed team
"""
return teams[-1][1]
# Check your answer
q2.check()
Correct
[17]:
#q2.hint()
#q2.solution()
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Up next
Loops and List Comprehensions
Not now
Start Tutorial
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.

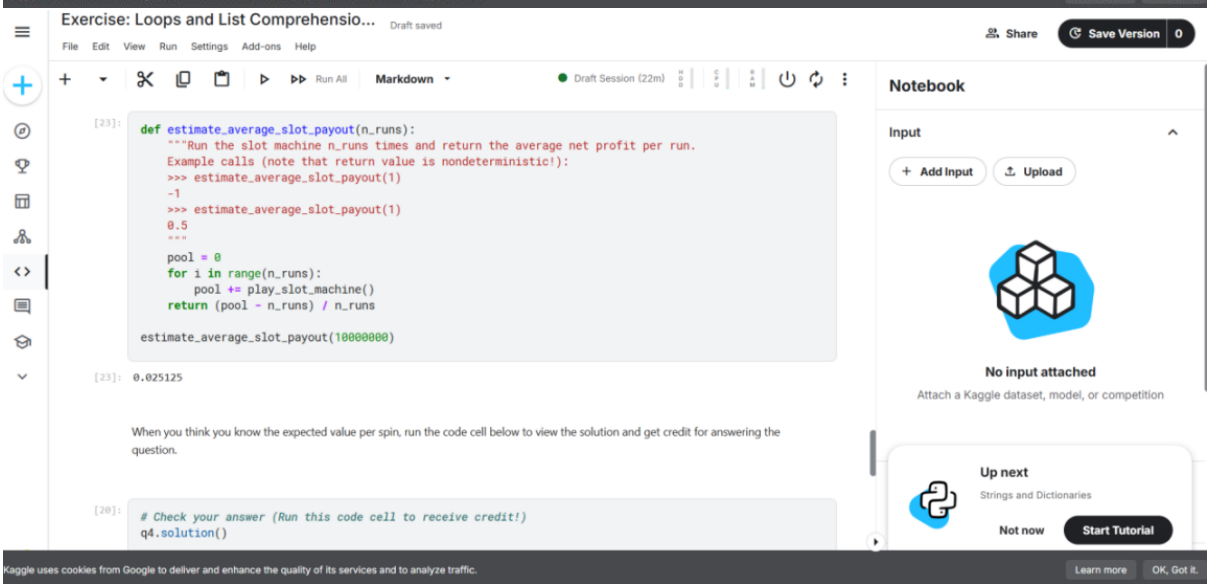

Exercise: Lists
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 0
Run All
Markdown
Draft Session (41m)
3.
The next iteration of Mario Kart will feature an extra-infuriating new item, the Purple Shell. When used, it warps the last place racer into first place and the first place racer into last place. Complete the function below to implement the Purple Shell's effect.
[18]:
def purple_shell(racers):
"""Given a list of racers, set the first place racer (at the front of the list) to last place and vice versa.
>>> r = ["Mario", "Bowser", "Luigi"]
>>> purple_shell(r)
>>> r
["Luigi", "Bowser", "Mario"]
"""
racers[0], racers[-1] = racers[-1], racers[0]
# Check your answer
q3.check()
Correct
[19]:
#q3.hint()
#q3.solution()
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Up next
Loops and List Comprehensions
Not now
Start Tutorial
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.



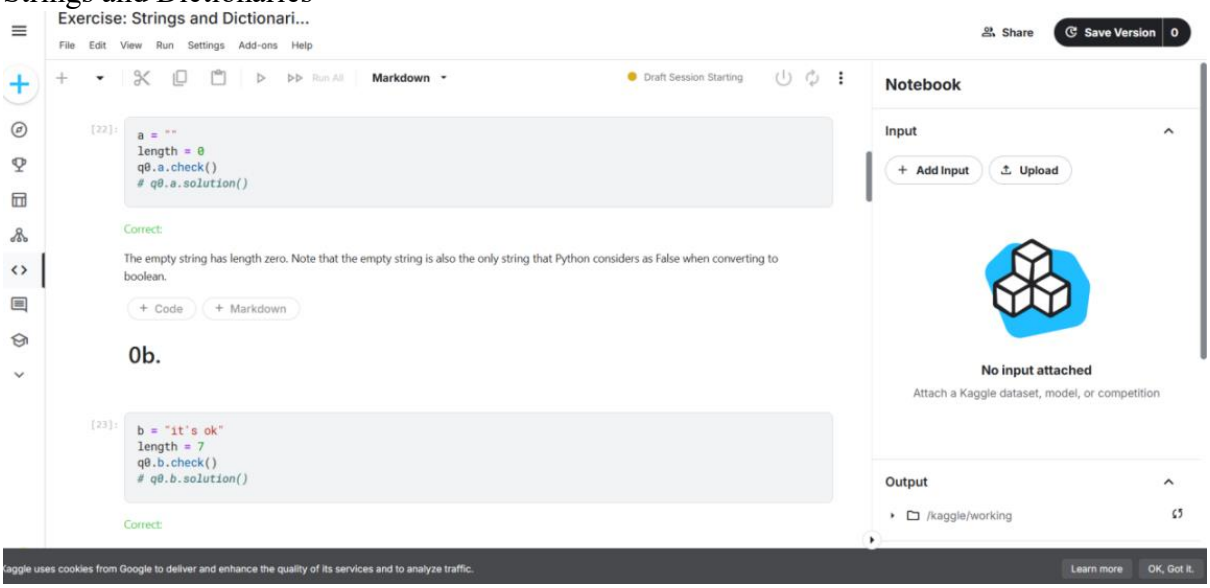
9. Loops and List Comprehensions

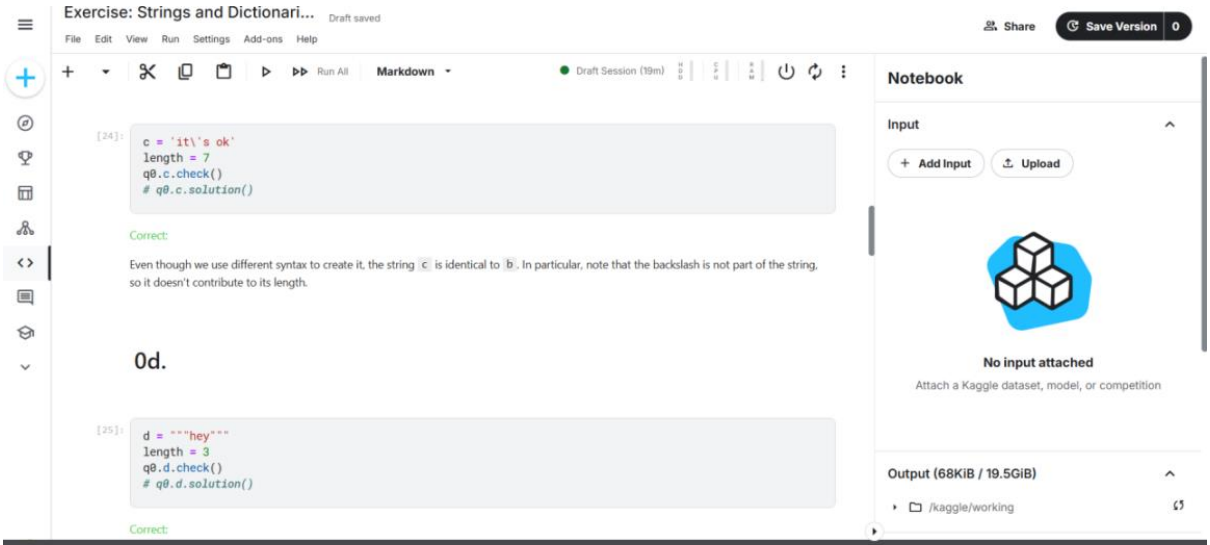


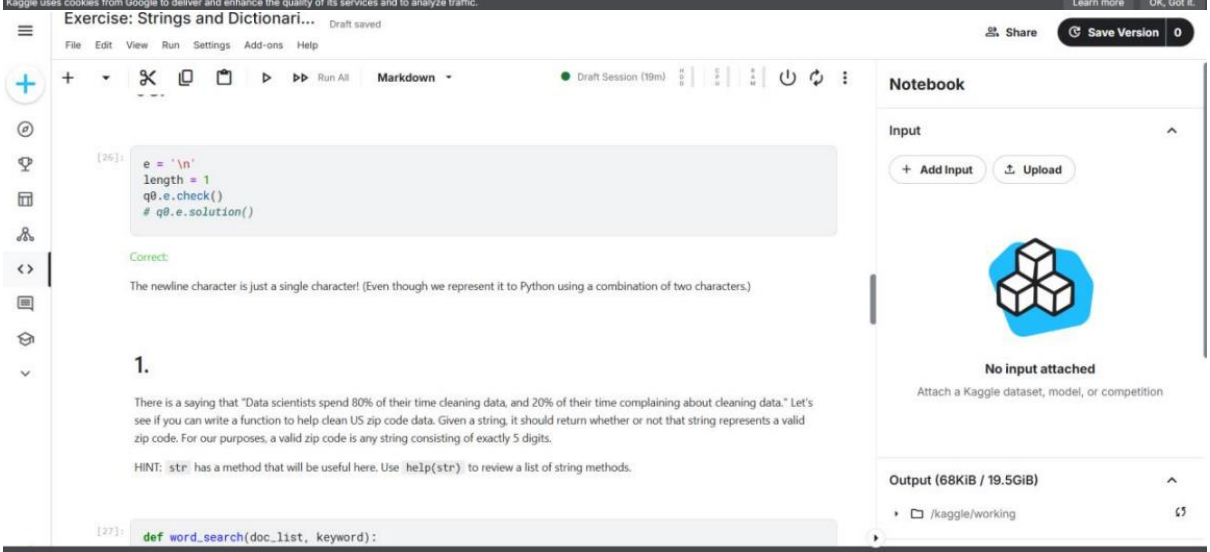


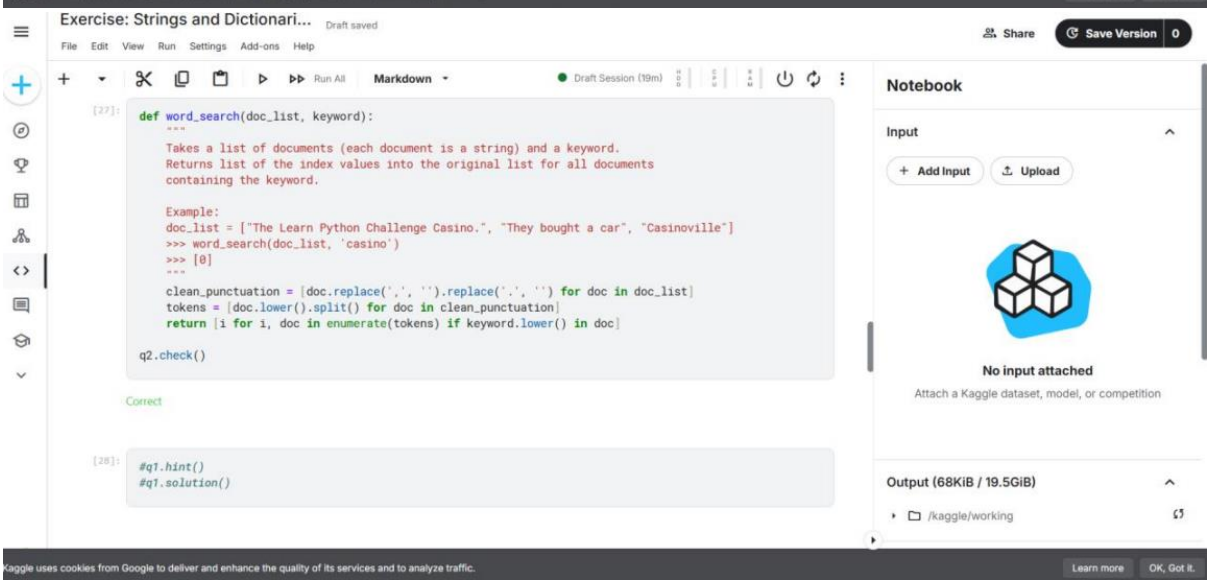


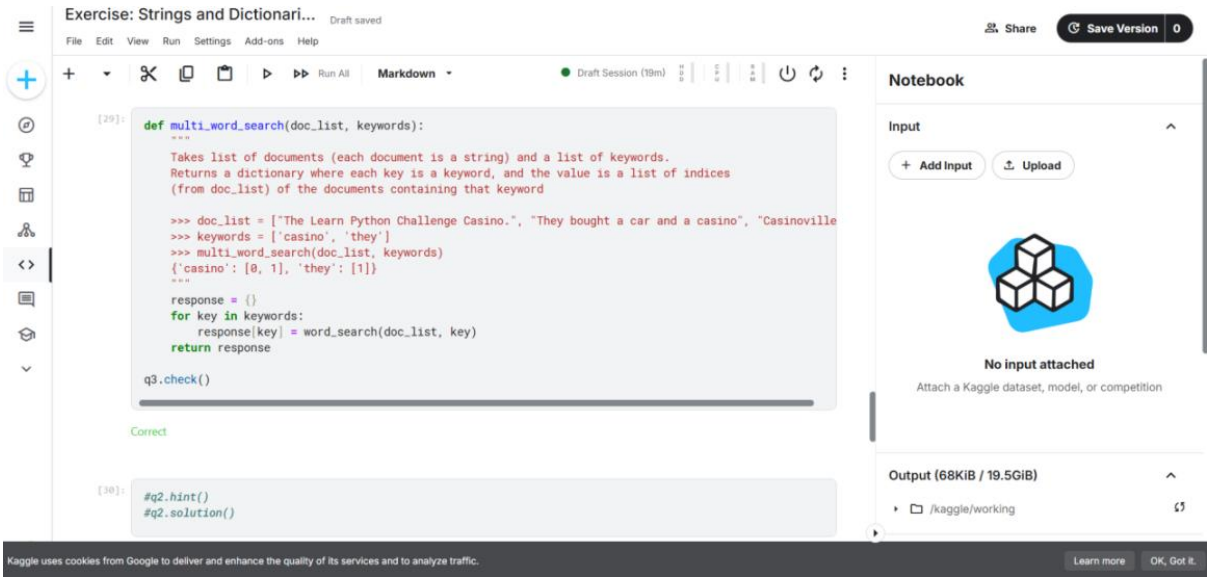
10. Strings and Dictionaries

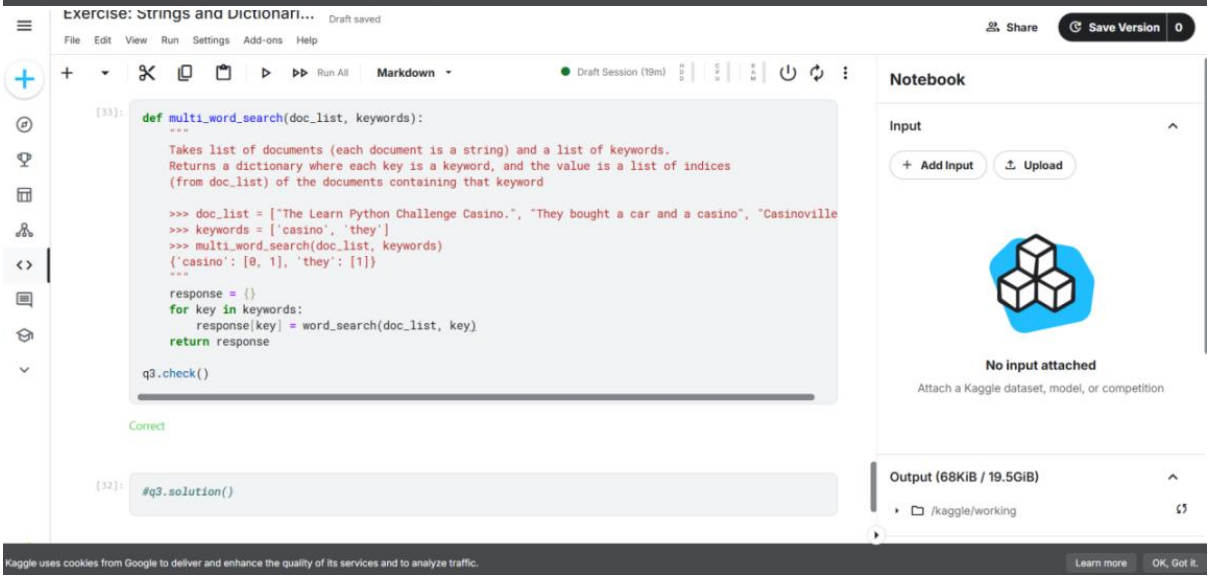



Exercise: Strings and Dictionari...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 0
Run All
Markdown
Draft Session (19m)
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output (68KiB / 19.5GiB)
/kaggle/working
[24]:
c = 'it\'s ok'
length = 7
q0.c.check()
# q0.c.solution()
Correct:
Even though we use different syntax to create it, the string c is identical to b. In particular, note that the backslash is not part of the string, so it doesn't contribute to its length.
0d.
[25]:
d = """hey"""
length = 3
q0.d.check()
# q0.d.solution()
Correct:
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.

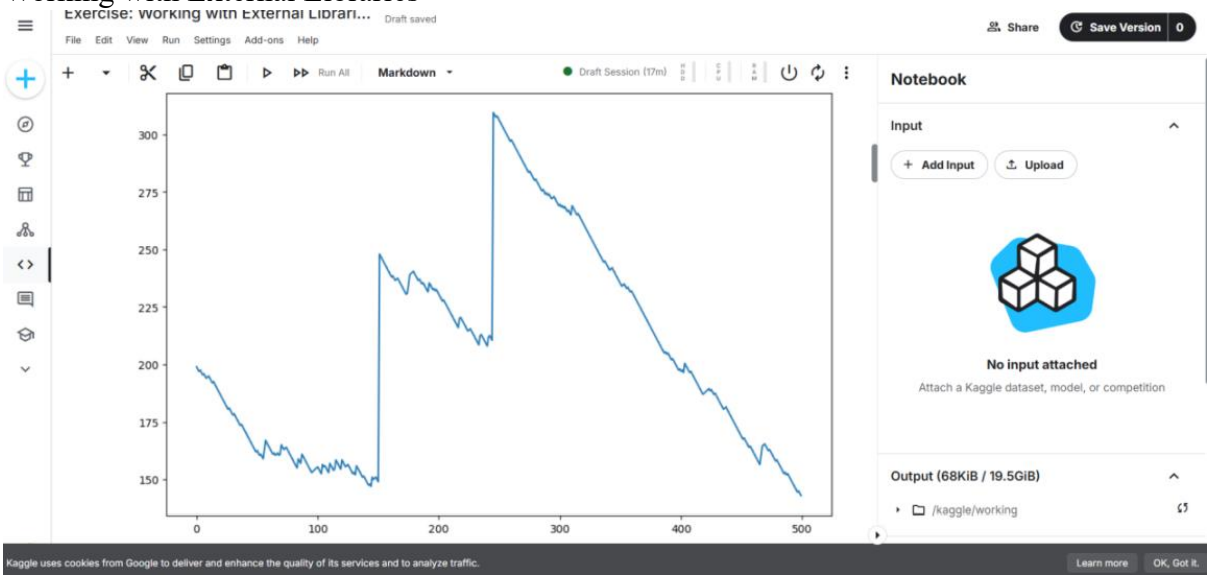

Exercise: Strings and Dictionari...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 0
Run All
Markdown
Draft Session (19m)
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output (68KiB / 19.5GiB)
/kaggle/working
[26]:
e = '\n'
length = 1
q0.e.check()
# q0.e.solution()
Correct:
The newline character is just a single character! (Even though we represent it to Python using a combination of two characters.)
1.
There is a saying that "Data scientists spend 80% of their time cleaning data, and 20% of their time complaining about cleaning data." Let's see if you can write a function to help clean US zip code data. Given a string, it should return whether or not that string represents a valid zip code. For our purposes, a valid zip code is any string consisting of exactly 5 digits.
HINT: str has a method that will be useful here. Use help(str) to review a list of string methods.
[27]:
def word_search(doc_list, keyword):
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.


Exercise: Strings and Dictionari...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 0
Run All
Markdown
Draft Session (19m)
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output (68KiB / 19.5GiB)
/kaggle/working
[27]:
def word_search(doc_list, keyword):
"""
Takes a list of documents (each document is a string) and a keyword.
Returns list of the index values into the original list for all documents
containing the keyword.
Example:
doc_list = ["The Learn Python Challenge Casino.", "They bought a car", "Casinoville"]
>>> word_search(doc_list, 'casino')
>>> [0]
"""
clean_punctuation = [doc.replace(',', '').replace('.', '') for doc in doc_list]
tokens = [doc.lower().split() for doc in clean_punctuation]
return [i for i, doc in enumerate(tokens) if keyword.lower() in doc]
q2.check()
Correct
[28]:
#q1.hint()
#q1.solution()
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.





11. Working with External Libraries

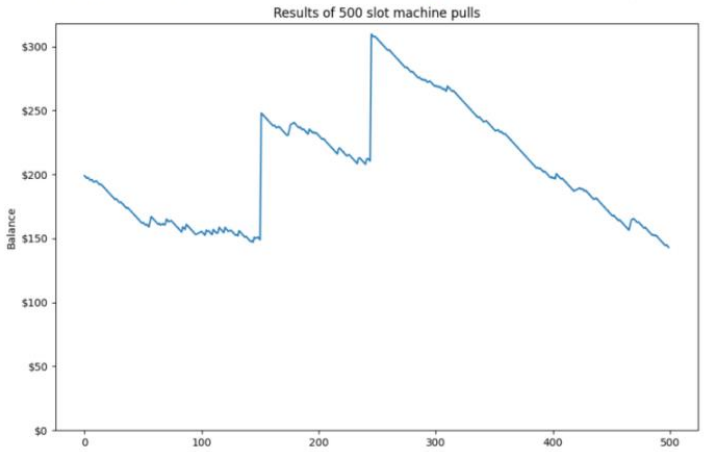


Exercise: Working with External Librari... Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Markdown Draft Session (17m)



Notebook

Input

+ Add Input Upload

No input attached

Attach a Kaggle dataset, model, or competition

Output (68KiB / 19.5GiB)

/kaggle/working

Exercise: Working with External Librari... Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Markdown Draft Session (17m)

Notebook

Input

+ Add Input Upload

No input attached

Attach a Kaggle dataset, model, or competition

Output (68KiB / 19.5GiB)

/kaggle/working

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic. Learn more OK, Got it.

[76]:

```
def best_items(racers):
    """Given a list of racer dictionaries, return a dictionary mapping items to the number
    of times those items were picked up by racers who finished in first place.
    """
    winner_item_counts = {}
    for i in range(len(racers)):
        # The i'th racer dictionary
        racer = racers[i]
        # We're only interested in racers who finished in first
        if racer['finish'] == 1:
            for i in racer['items']:
                # Add one to the count for this item (adding it to the dict if necessary)
                if i not in winner_item_counts:
                    winner_item_counts[i] = 0
                winner_item_counts[i] += 1

        # Data quality issues :/ Print a warning about racers with no name set. We'll take care of it late
        if racer['name'] is None:
            print("WARNING: Encountered racer with unknown name on iteration {}/{} (racer = {})".format(
                i+1, len(racers), racer['name'])
                 )
    return winner_item_counts
```

Exercise: Working with External Librari... Draft saved

File Edit View Run Settings Add-ons Help

Share Save Version 0

Run All Markdown Draft Session (17m)

Notebook

Input

+ Add Input Upload

No input attached

Attach a Kaggle dataset, model, or competition

Output (68KiB / 19.5GiB)

/kaggle/working

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic. Learn more OK, Got it.

He tried it on a small example list above and it seemed to work correctly:

[77]:

```
sample = [
    {'name': 'Peach', 'items': ['green shell', 'banana', 'green shell',], 'finish': 3},
    {'name': 'Bowser', 'items': ['green shell',], 'finish': 1},
    {'name': None, 'items': ['mushroom',], 'finish': 2},
    {'name': 'Toad', 'items': ['green shell', 'mushroom'], 'finish': 1},
]
best_items(sample)
```

```
WARNING: Encountered racer with unknown name on iteration 3/4 (racer = None)
```

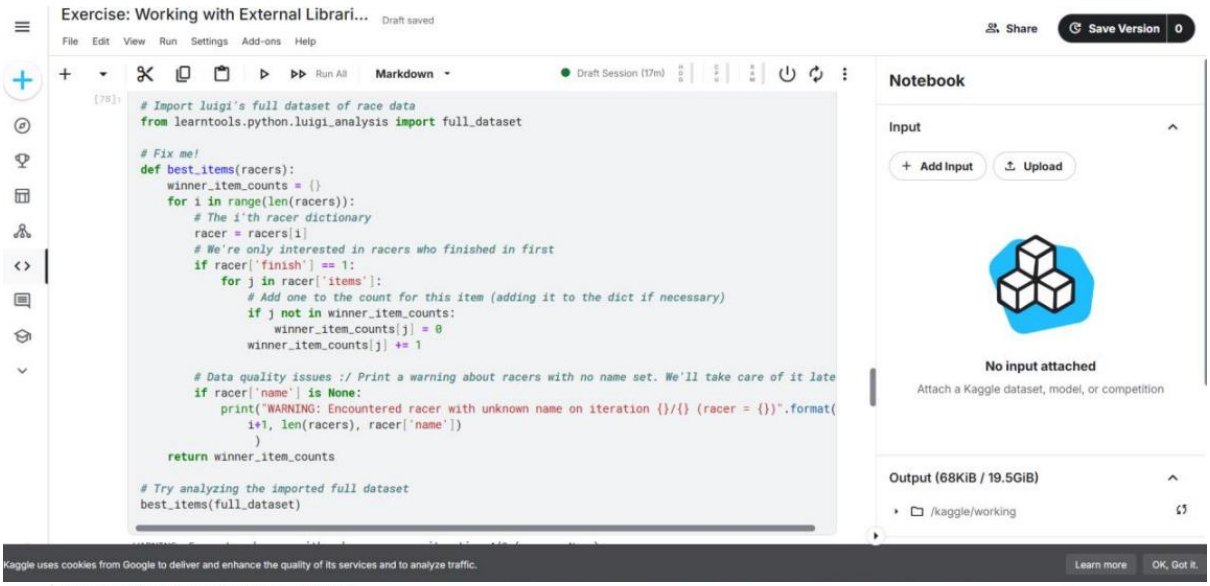
[77]:

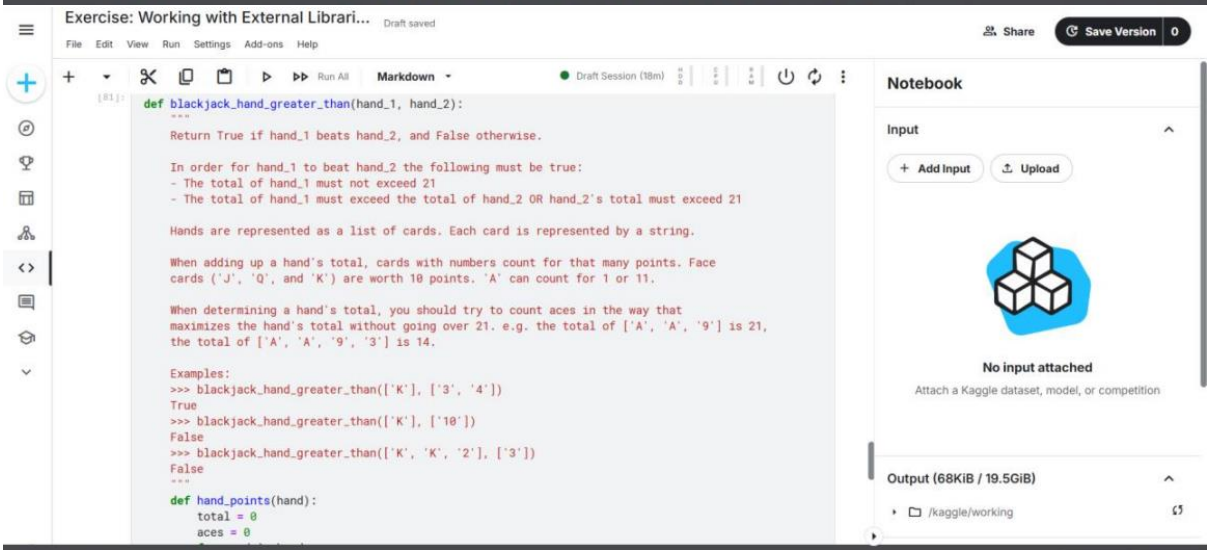
```
{'green shell': 2, 'mushroom': 1}
```

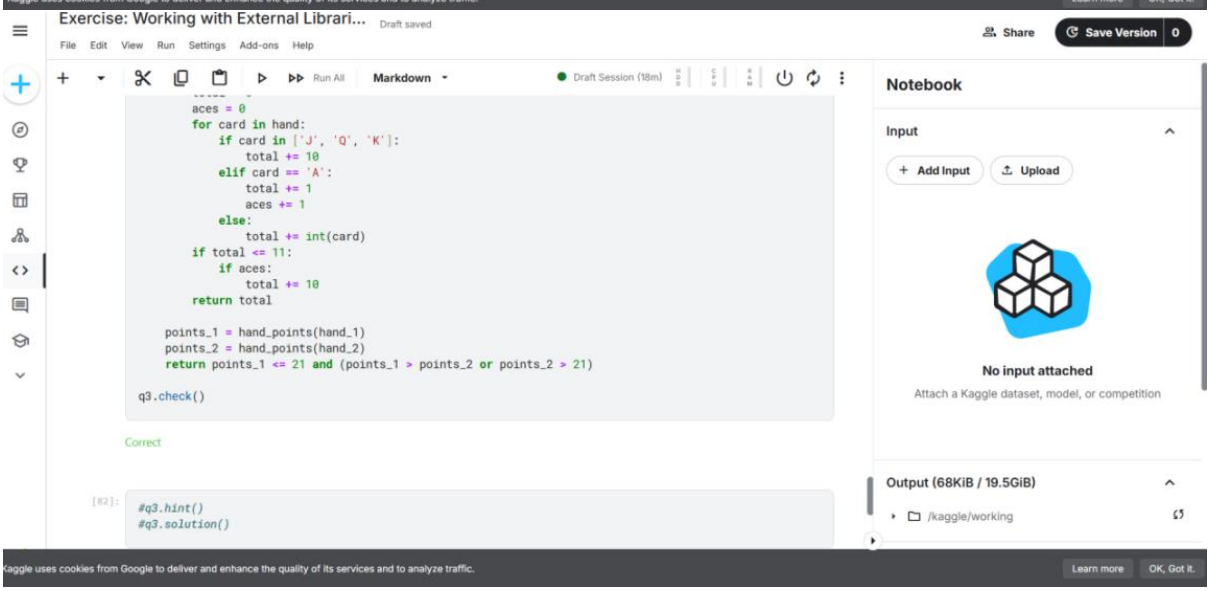
However, when he tried running it on his full dataset, the program crashed with a `TypeError`.

Can you guess why? Try running the code cell below to see the error message Luigi is getting. Once you've identified the bug, fix it in the cell below (so that it runs without any errors).

Hint: Luigi's bug is similar to one we encountered in the tutorial when we talked about star imports.


Exercise: Working with External Librari...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 0
Run All
Markdown
Draft Session (17m)
[78]:
# Import luigi's full dataset of race data
from learntools.python.luigi_analysis import full_dataset
# Fix me!
def best_items(racers):
winner_item_counts = {}
for i in range(len(racers)):
# The i'th racer dictionary
racer = racers[i]
# We're only interested in racers who finished in first
if racer['finish'] == 1:
for j in racer['items']:
# Add one to the count for this item (adding it to the dict if necessary)
if j not in winner_item_counts:
winner_item_counts[j] = 0
winner_item_counts[j] += 1
# Data quality issues :/ Print a warning about racers with no name set. We'll take care of it late
if racer['name'] is None:
print("WARNING: Encountered racer with unknown name on iteration {}/{} (racer = {})".format(
i+1, len(racers), racer['name'])
)
return winner_item_counts
# Try analyzing the imported full dataset
best_items(full_dataset)
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output (68KiB / 19.5GiB)
/kaggle/working
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.


Exercise: Working with External Librari...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 0
Run All
Markdown
Draft Session (18m)
[81]:
def blackjack_hand_greater_than(hand_1, hand_2):
"""
Return True if hand_1 beats hand_2, and False otherwise.
In order for hand_1 to beat hand_2 the following must be true:
- The total of hand_1 must not exceed 21
- The total of hand_1 must exceed the total of hand_2 OR hand_2's total must exceed 21
Hands are represented as a list of cards. Each card is represented by a string.
When adding up a hand's total, cards with numbers count for that many points. Face
cards ('J', 'Q', and 'K') are worth 10 points. 'A' can count for 1 or 11.
When determining a hand's total, you should try to count aces in the way that
maximizes the hand's total without going over 21. e.g. the total of ['A', 'A', '9'] is 21,
the total of ['A', 'A', '9', '3'] is 14.
Examples:
>>> blackjack_hand_greater_than(['K'], ['3', '4'])
True
>>> blackjack_hand_greater_than(['K'], ['10'])
False
>>> blackjack_hand_greater_than(['K', 'K', '2'], ['3'])
False
"""
def hand_points(hand):
total = 0
aces = 0
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output (68KiB / 19.5GiB)
/kaggle/working
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.


Exercise: Working with External Librari...
Draft saved
File Edit View Run Settings Add-ons Help
Share
Save Version 0
Run All
Markdown
Draft Session (18m)
aces = 0
for card in hand:
if card in ['J', 'Q', 'K']:
total += 10
elif card == 'A':
total += 1
aces += 1
else:
total += int(card)
if total <= 11:
if aces:
total += 10
return total
points_1 = hand_points(hand_1)
points_2 = hand_points(hand_2)
return points_1 <= 21 and (points_1 > points_2 or points_2 > 21)
q3.check()
Correct
[82]:
#q3.hint()
#q3.solution()
Notebook
Input
Add Input
Upload
No input attached
Attach a Kaggle dataset, model, or competition
Output (68KiB / 19.5GiB)
/kaggle/working
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.
Learn more
OK, Got it.