

Screenshot of a web browser showing the Kaggle Learn Python course page.

The URL in the address bar is kaggle.com/learn/python.

The page title is "Learn Python | Kaggle".

The main content area displays the "Python" course. A banner at the top says "Learn the most important language for data science." and shows a progress bar indicating "100% complete! Congrats!" with a "View Certificate" button.

The sidebar on the left includes a "Create" button and links to Home, Competitions, Datasets, Models, Code, Discussions, Learn (selected), More, Your Work, VIEWED, Working with External Data, and Strings and Dictionaries.

The main content area has tabs for Courses and Discussions, with "Courses" selected.

The "Lessons" section lists three lessons:

- 1 Hello, Python: A quick introduction to Python syntax, variable assignment, and numbers. Status: Both Tutorial and Exercise are checked.
- 2 Functions and Getting Help: Calling functions and defining our own, and using Python's builtin documentation. Status: Both Tutorial and Exercise are checked.
- 3 Booleans and Conditionals: Using booleans for conditionals. Status: Both Tutorial and Exercise are checked.

On the right side, there are sections for "Builds on" (Intro to Programming), "Preparation for" (Intro to Machine Learning, Pandas, Intro to SQL, Intro to Game AI and Reinforcement Learning), and "Hours to earn certificate" (5 (estimated)).

A cookie consent message at the bottom states: "Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic." with buttons for "Learn more", "OK, Got it.", and "X".

The system tray at the bottom right shows the date as 2/21/2025 and the time as 10:18 AM.

The image displays two nearly identical screenshots of the Kaggle Learn Python course interface, stacked vertically. Both screenshots show a sidebar on the left with navigation links like Home, Competitions, Datasets, Models, Code, Discussions, Learn (which is selected), and More. The main content area is titled 'Lessons' and lists six lessons numbered 1 through 6. Each lesson card includes a title, a brief description, and two circular icons representing 'Tutorial' and 'Exercise'. To the right of the lessons, there's a sidebar with information such as 'Builds on', 'Preparation for', 'Hours to earn certificate', 'Cost', and 'Instructor'. A status bar at the bottom of each screenshot shows 'CLOVE' and a system tray with various icons.

Lessons

Lesson	Title	Tutorial	Exercise
1	Hello, Python	✓	✓
2	Functions and Getting Help	✓	✓
3	Booleans and Conditionals	✓	✓
4	Lists	✓	✓
5	Loops and List Comprehensions	✓	✓
6	Strings and Dictionaries	✓	✓

Builds on
Intro to Programming

Preparation for
Intro to Machine Learning
Pandas
Intro to SQL
Intro to Game AI and Reinforcement Learning

Hours to earn certificate
5 (estimated)

Cost
No cost, like all Kaggle Learn Courses

Instructor
Colin Morris

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

10:20 2/21/2023

1. Hello Python

Screenshot of a Kaggle Notebook interface titled "Exercise: Syntax, Variables, and Numbers".

The notebook contains the following code:

```
[3]: # 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()
```

Feedback message: **Correct!** What?! You got it right without needing a hint or anything? Drats. Well hey, you should still continue to the next step to get some practice asking for a hint and checking solutions. (Even though you obviously don't need any help here.)

Input section: No input attached. Attach a Kaggle dataset, model, or competition.

Output section: /kaggle/working

Bottom status bar: Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic. Learn more OK, Got It. 10:20 2/21/2025

Functions and Getting Help | Learn Python, Data Viz, Pandas | Exercise: Syntax, Variables, and Numb... | +

kaggle.com/code/ridhoradyapratama/exercise-syntax-variables-and-numbers/edit

Exercise: Syntax, Variables, and Numb... Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Code

Draft Session off (run a cell to start) ⚡

(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.

+ Code + Markdown

[]:

```
#q0.hint()
#q0.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. 10:23 AM 2/21/2023

Functions and Getting Help | Learn Python, Data Viz, Pandas | Exercise: Syntax, Variables, and Numb... | +

kaggle.com/code/ridhoradyapratama/exercise-syntax-variables-and-numbers/edit

Exercise: Syntax, Variables, and Numb... Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Code

Draft Session off (run a cell to start) ⚡

[]:

```
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

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Exercise: Syntax, Variables, and Numb... Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Code

Draft Session off (run a cell to start) ⚡ ⚡ :

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).

[1]:

```
##### 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()
```

[1]:

```
#q2.hint()
```

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.

Exercise: Syntax, Variables, and Numb... Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Code

Draft Session off (run a cell to start) ⚡ ⚡ :

Write an arithmetic expression below to calculate how many candies they must smash for a given haul.

[1]:

```
# Variables representing the number of candies collected by alice, bob, and carol
alice_candies = 121
bob_candies = 77
carol_candies = 109

# Your code goes here! Replace the right-hand side of this assignment with an expression
# involving alice_candies, bob_candies, and carol_candies
to_smash = (alice_candies + bob_candies + carol_candies) % 3

# Check your answer
q4.check()
```

[1]:

```
#q4.hint()
#q4.solution()
```

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.

CLOVE

10:23 2/21/2023

2. Functions and Getting Help

Exercise: Functions and Getting Help Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Markdown

Draft Session off (run a cell to start) ⚡ ⚡ :

Complete the body of the following function according to its docstring.

HINT: Python has a built-in function `round`.

```
[4]: def round_to_two_places(num):
    """Return the given number rounded to two decimal places.

    >>> round_to_two_places(3.14159)
    3.14
    """
    # Replace this body with your own code.
    # ("pass" is a keyword that does literally nothing. We used it as a placeholder
    # because after we begin a code block, Python requires at least one line of code)
    return round(num, 2)

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

Correct

+ Code + Markdown

```
[7]: # Uncomment the following for a hint
#q1.hint()
# Or uncomment the following to peek at the solution
```

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Learn more OK, Got it.

Notebook

Input

+ Add Input Upload



No input attached

Attach a Kaggle dataset, model, or competition

Output

/kaggle/working

Exercise: Functions and Getting Help Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Markdown

Draft Session off (run a cell to start) ⚡ ⚡ :

```
[8]: # Put your test code here
round (10.000, -1)
```

```
[8]: 10.0
```

Can you think of a case where this would be useful? Once you're ready, run the code cell below to see the answer and to receive credit for completing the problem.

```
[10]: # Check your answer (Run this code cell to receive credit!)
q2.solution()
```

Solution: As you've seen, `ndigits=-1` rounds to the nearest 10, `ndigits=-2` rounds to the nearest 100 and so on. Where might this be useful? Suppose we're dealing with large numbers:

```
The area of Finland is 338,424 km2
The area of Greenland is 2,166,086 km2
```

We probably don't care whether it's really 338,424, or 338,425, or 338,177. All those digits of accuracy are just distracting. We can chop them off by calling `round()` with `ndigits=-3`:

```
The area of Finland is 338,000 km2
The area of Greenland is 2,166,000 km2
```

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Learn more OK, Got it.

Exercise: Functions and Getting Help Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Markdown

Draft Session off (run a cell to start) ⚡

Modify it so that it optionally takes a second argument representing the number of friends the candies are being split between. If no second argument is provided, it should assume 3 friends, as before.

Update the docstring to reflect this new behaviour.

```
[11]: def to_smash(total_candies, friend = 3):
    """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
    ==
    return total_candies % friend

# Check your answer
q3.check()
```

Correct

```
[12]: q3.hint()
```

Hint: Refer to the section of the last tutorial notebook where we talked about default arguments

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Learn more OK, Got it.

Notebook

Input

+ Add Input Upload



No input attached
Attach a Kaggle dataset, model, or competition

Output

/kaggle/working

Exercise: Functions and Getting Help Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Markdown

Draft Session off (run a cell to start) ⚡

```
[16]: round_to_two_places(9.9999)
```

10.0

```
[18]: x = -10
y = 5
Which of the two variables above has the smallest absolute value?
smallest_abs = min(x, y, key=abs)
print(smallest_abs)
```

Object 'value' not found.
5

+ Code + Markdown

```
[20]: # def f(x):
#     y = abs(x)
#     return y

# print(f(5))
```

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Learn more OK, Got it.

Notebook

Input

+ Add Input Upload



No input attached
Attach a Kaggle dataset, model, or competition

Output

/kaggle/working

3. Booleans and Conditionals

Exercise: Booleans and Condition... Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Markdown

Draft Session off (run a cell to start) ⚡

Many programming languages have `sign` available as a built-in function. Python doesn't, but we can define our own!

In the cell below, define a function called `sign` which takes a numerical argument and returns -1 if it's negative, 1 if it's positive, and 0 if it's 0.

```
[25]: # Your code goes here. Define a function called 'sign'
def sign(number):
    if number == 0:
        return 0
    elif number > 0:
        return 1
    else:
        return -1

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

Correct

+ Code + Markdown

```
[26]: #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.

10:25 AM 2/21/2023

Exercise: Booleans and Condition... Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Markdown

Draft Session off (run a cell to start) ⚡

```
[27]: 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
```

[27]: 1

What happens if we call it with `total_candies = 1`?

```
[28]: 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.

10:25 AM 2/21/2023

Exercise: Booleans and Condition... Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Markdown

Draft Session off (run a cell to start) ⚡

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")

```
[29]: 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
```

[29]: 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.

10:26 AM 2/21/2023

Exercise: Booleans and Condition... Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Markdown

Draft Session off (run a cell to start) ⚡

```
[31]: 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 < 5 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.

10:26 AM 2/21/2023

Exercise: Booleans and Condition... Draft saved

File Edit View Run Settings Add-ons Help

+ X D 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

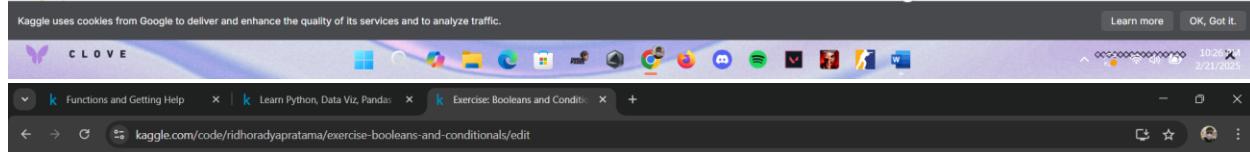
```
[34]: #q4.hint()
#q4.solution()
```

No input attached

Attach a Kaggle dataset, model, or competition

Output

/kaggle/working



Exercise: Booleans and Condition... Draft saved

File Edit View Run Settings Add-ons Help

+ X D 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

```
[37]: #q5.a.hint()
#q5.a.solution()
```

No input attached

Attach a Kaggle dataset, model, or competition

Output

/kaggle/working



Exercise: Booleans and Condition... Draft saved

File Edit View Run Settings Add-ons Help

+ X Run All Markdown

#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:

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Learn more OK, Got it.

10:26 2/21/2023

Exercise: Booleans and Condition... Draft saved

File Edit View Run Settings Add-ons Help

+ X Run All 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

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Learn more OK, Got it.

10:26 2/21/2023

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 float? Try it out with the code below.

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Learn more OK, Got it.

10:26 2/21/2023

Exercise: Booleans and Condition... Draft saved

Notebook

Input

+ Add Input Upload



No input attached
Attach a Kaggle dataset, model, or competition

Output

/kaggle/working

Exercise: Booleans and Condition... Draft saved

Notebook

Input

+ Add Input Upload



No input attached
Attach a Kaggle dataset, model, or competition

Output

/kaggle/working

Keen Going

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.

```
[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
```

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Learn more OK, Got it.

10:26 2/21/2023

Exercise: Booleans and Condition... Draft saved

Notebook

Input

+ Add Input Upload



No input attached
Attach a Kaggle dataset, model, or competition

Output

/kaggle/working

Exercise: Booleans and Condition... Draft saved

Notebook

Input

+ Add Input Upload



No input attached
Attach a Kaggle dataset, model, or competition

Output

/kaggle/working

Keen Going

Player won 18983 out of 50000 games (win rate = 38.0%)

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Learn more OK, Got it.

10:27 2/21/2023

4. Lists

Functions and Getting Help | Learn Python, Data Viz, Pandas | Kaggle Notebook Editor

kaggle.com/code/ridhoradyaprata/exercise-lists/edit

Exercise: Lists Draft saved

File Edit View Run Settings Add-ons Help

+ X ☐ 📁 ↻ ⏪ ⏴ 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.

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

CLOVE

Up next

Loops and List Comprehensions

Not now Start Tutorial

Learn more OK, Got it.

No input attached

Attach a Kaggle dataset, model, or competition

Functions and Getting Help | Learn Python, Data Viz, Pandas | Kaggle Notebook Editor

kaggle.com/code/ridhoradyaprata/exercise-lists/edit

Exercise: Lists Draft saved

File Edit View Run Settings Add-ons Help

+ X ☐ 📁 ↻ ⏪ 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()
```

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

CLOVE

Up next

Loops and List Comprehensions

Not now Start Tutorial

Learn more OK, Got it.

No input attached

Attach a Kaggle dataset, model, or competition

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()

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Complete the function below which takes a list of party attendees as well as a person, and tells us whether that person is fashionably late.

```
[22]: def fashionably_late(arrivals, name):
    """Given an ordered list of arrivals to the party and a name, return whether the guest with that
    name was fashionably late.

    n_guests = len(arrivals)
    if name in arrivals[(n_guests + 1) // 2 : -1]:
        return True
    return False

    # Check your answer
    q5.check()

Correct
```

[23]: #q5.hint()
#q5.solution()

Keep Going

Up next
Loops and List Comprehensions
Not now Start Tutorial

Learn more OK, Got it. 10:28 AM 2/21/2023

5. Loops and List Comprehensions

Screenshot of a Kaggle Notebook Editor session titled "Exercise: Loops and List Comprehension..." showing a Python code challenge.

The notebook interface includes:

- Header: Functions and Getting Help, Learn Python, Data Viz, Pandas, Kaggle Notebook Editor
- Toolbar: File, Edit, View, Run, Settings, Add-ons, Help
- Cell 10: [10]: def has_lucky_number(nums):
 """Return whether the given list of numbers is lucky. A lucky list contains
 at least one number divisible by 7.
 """

 for num in nums:
 if num % 7 == 0:
 return True
 else:
 return False
- Cell 11: [11]: def has_lucky_number(nums):
 """Return whether the given list of numbers is lucky. A lucky list contains
 at least one number divisible by 7.
 """

 for num in nums:
 if num % 7 == 0:
 return True
 else:
 return False
- Right sidebar: Notebook, Input (+ Add Input, Upload), No input attached (Attach a Kaggle dataset, model, or competition), Up next (Strings and Dictionaries, Start Tutorial), and a cookie consent banner from Clove.

Exercise: Loops and List Comprehension... Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Markdown

Draft Session (22m)

[11]:

```
def has_lucky_number(nums):
    """Return whether the given list of numbers is lucky. A lucky list contains
    at least one number divisible by 7.

    for num in nums:
        if num % 7 == 0:
            return True
    return False

q1.check()
```

Correct:

Remember that `return` causes a function to exit immediately. So our original implementation always ran for just one iteration. We can only return `False` if we've looked at every element of the list (and confirmed that none of them are lucky). Though we can return early if the answer is `True`:

```
def has_lucky_number(nums):
    for num in nums:
        if num % 7 == 0:
            return True
    # Implement the list without finding a lucky number
```

Session started.

No input attached

Attach a Kaggle dataset, model, or competition

Up next

Strings and Dictionaries

Not now Start Tutorial

Learn more OK, Got it.

10:29 AM 2/21/2023

CLOVE

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Exercise: Loops and List Comprehension... Draft saved

File Edit View Run Settings Add-ons Help

+ X D Run All Markdown

Draft Session (22m)

[21]:

```
def elementwise_greater_than(L, thresh):
    """Return a list with the same length as L, where the value at index i is
    True if L[i] is greater than thresh, and False otherwise.

    >>> elementwise_greater_than([1, 2, 3, 4], 2)
    [False, False, True, True]
    """
    return [num > thresh for num in L]

# Check your answer
q2.check()
```

Correct:

Here's one solution:

```
def elementwise_greater_than(L, thresh):
    res = []
    for ele in L:
        res.append(ele > thresh)
    return res
```

No input attached

Attach a Kaggle dataset, model, or competition

Up next

Strings and Dictionaries

Not now Start Tutorial

Learn more OK, Got it.

10:29 AM 2/21/2023

CLOVE

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

The screenshot shows a Kaggle Notebook Editor window. The title bar says "Exercise: Loops and List Comprehension..." and "Draft saved". The menu bar includes File, Edit, View, Run, Settings, Add-ons, and Help. The toolbar has icons for file operations like New, Open, Save, Print, Run All, and a dropdown for Markdown. A status bar at the top right shows "Draft Session (22m)" with a progress bar and some icons. The main area contains a code cell [23]:

```
[23]: def estimate_average_slot_payout(n_runs):
    """Run the slot machine n_runs times and return the average net profit per run.
    Example calls (note that return value is nondeterministic!):
    >>> estimate_average_slot_payout(1)
    -1
    >>> estimate_average_slot_payout(1)
    0.5
    ...
    pool = 0
    for i in range(n_runs):
        pool += play_slot_machine()
    return (pool - n_runs) / n_runs

estimate_average_slot_payout(1000000)
```

The output of the cell is [23]: 0.025125.

Below the code cell, there is a note: "When you think you know the expected value per spin, run the code cell below to view the solution and get credit for answering the question."

Another code cell [28] is shown:

```
[28]: # Check your answer (Run this code cell to receive credit!)
q4.solution()
```

A message at the bottom of the screen says "Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic." with "OK, Got it." and "Not now" buttons. On the right side, there's a sidebar titled "Notebook" with "Input" section, "No input attached" message, and a "Up next" section for "Strings and Dictionaries" with a "Start Tutorial" button.

6. Strings and Dictionaries

Functions and Getting Help | Learn Python, Data Viz, Pandas | Exercise: Strings and Dictionaries | kaggle.com/code/ridhoradapratama/exercise-strings-and-dictionaries/edit

Exercise: Strings and Dictionaries...

File Edit View Run Settings Add-ons Help

[22]:

```
a = ""
length = 0
q0.a.check()
# q0.a.solution()
```

Correct:

The empty string has length zero. Note that the empty string is also the only string that Python considers as False when converting to boolean.

+ Code + Markdown

0b.

[23]:

```
b = "it's ok"
length = 7
q0.b.check()
# q0.b.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.](#)

10:30 AM 2/21/2023

Draft Session Starting

Notebook

Input

+ Add Input [Upload](#)

No input attached

Attach a Kaggle dataset, model, or competition

Output

/kaggle/working

Functions and Getting Help | Learn Python, Data Viz, Pandas | Kaggle Notebook Editor | kaggle.com/code/ridhoradapratama/exercise-strings-and-dictionaries/edit

Exercise: Strings and Dictionaries...

Draft saved

File Edit View Run Settings Add-ons Help

[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.](#)

10:31 AM 2/21/2023

Draft Session (19m)

Notebook

Input

+ Add Input [Upload](#)

No input attached

Attach a Kaggle dataset, model, or competition

Output (68KiB / 19.5GiB)

/kaggle/working

Exercise: Strings and Dictionaries... Draft saved

File Edit View Run Settings Add-ons Help

[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.

File Edit View Run Settings Add-ons Help

[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.

Output (68KiB / 19.5GiB)

/kaggle/working

Learn more OK, Got it.

10:31 2/21/2023

No input attached

Attach a Kaggle dataset, model, or competition

Notebook

Input

+ Add Input Upload



No input attached

Output (68KiB / 19.5GiB)

/kaggle/working

Learn more OK, Got it.

10:31 2/21/2023

No input attached

Attach a Kaggle dataset, model, or competition

Notebook

Input

+ Add Input Upload



No input attached

Output (68KiB / 19.5GiB)

/kaggle/working

Learn more OK, Got it.

10:31 2/21/2023

Kaggle Notebook Editor

Exercise: Strings and Dictionaries...

```
[29]: def multi_word_search(doc_list, keywords):
    """
    Takes list of documents (each document is a string) and a list of keywords.
    Returns a dictionary where each key is a keyword, and the value is a list of indices
    (from doc_list) of the documents containing that keyword

    >>> doc_list = ["The Learn Python Challenge Casino.", "They bought a car and a casino", "Casinoville"]
    >>> keywords = ['casino', 'they']
    >>> multi_word_search(doc_list, keywords)
    {'casino': [0, 1], 'they': [1]}
    """
    response = {}
    for key in keywords:
        response[key] = word_search(doc_list, key)
    return response

q3.check()
```

Correct

```
[30]: #q2.hint()
#q2.solution()
```

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Output (68KiB / 19.5GiB)

Learn more OK, Got it.

10:31 2/21/2025

Kaggle Notebook Editor

Exercise: Strings and Dictionaries...

```
[33]: def multi_word_search(doc_list, keywords):
    """
    Takes list of documents (each document is a string) and a list of keywords.
    Returns a dictionary where each key is a keyword, and the value is a list of indices
    (from doc_list) of the documents containing that keyword

    >>> doc_list = ["The Learn Python Challenge Casino.", "They bought a car and a casino", "Casinoville"]
    >>> keywords = ['casino', 'they']
    >>> multi_word_search(doc_list, keywords)
    {'casino': [0, 1], 'they': [1]}
    """
    response = {}
    for key in keywords:
        response[key] = word_search(doc_list, key)
    return response

q3.check()
```

Correct

```
[34]: #q3.solution()
```

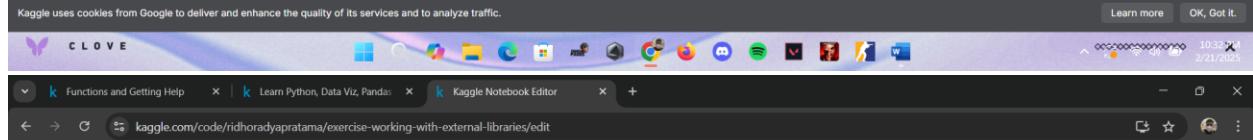
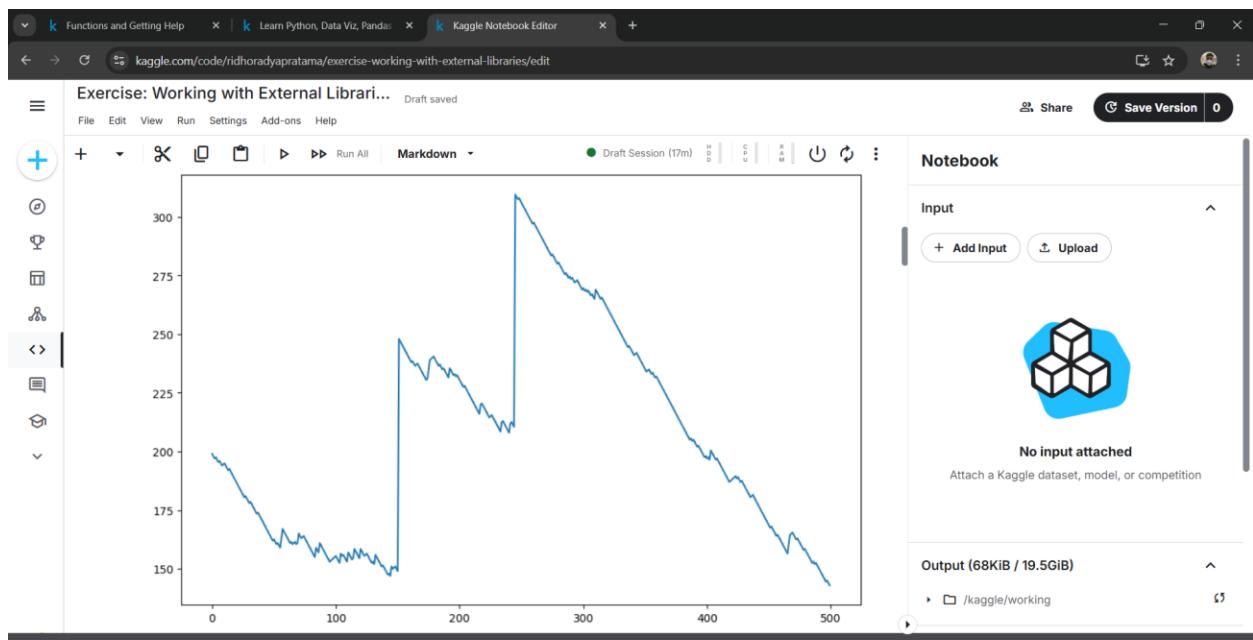
Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Output (68KiB / 19.5GiB)

Learn more OK, Got it.

10:31 2/21/2025

7. Working with External Libraries



Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Exercise: Working with External Librari... Draft saved

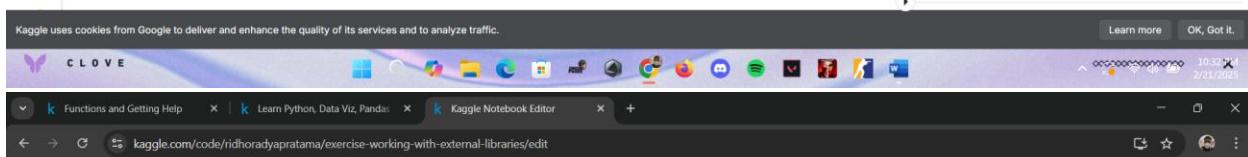
File Edit View Run Settings Add-ons Help

[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 item in racer['items']:
                # Add one to the count for this item (adding it to the dict if necessary)
                if item not in winner_item_counts:
                    winner_item_counts[item] = 0
                winner_item_counts[item] += 1

    # Data quality issues :/ Print a warning about racers with no name set. We'll take care of it later
    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
```



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.

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Exercise: Working with External Librari... Draft saved

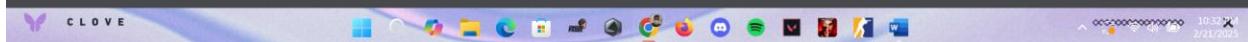
File Edit View Run Settings Add-ons Help

[77]:

```
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 item in racer['items']:
                # Add one to the count for this item (adding it to the dict if necessary)
                if item not in winner_item_counts:
                    winner_item_counts[item] = 0
                winner_item_counts[item] += 1

    # Data quality issues :/ Print a warning about racers with no name set. We'll take care of it later
    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 Libraries... Draft saved

File Edit View Run Settings Add-ons Help

[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 later.
    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)
```

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Notebook

Input

No input attached

Output (68KiB / 19.5GiB)

Learn more OK, Got it.

Exercise: Working with External Libraries... Draft saved

File Edit View Run Settings Add-ons Help

[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
```

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

Notebook

Input

No input attached

Output (68KiB / 19.5GiB)

Learn more OK, Got it.

Functions and Getting Help | Learn Python, Data Viz, Pandas | Kaggle Notebook Editor

kaggle.com/code/ridhoradypatama/exercise-working-with-external-libraries/edit

Exercise: Working with External Librari...

Draft saved

File Edit View Run Settings Add-ons Help

Markdown Draft Session (18m)

```
acces = 0
for card in hand:
    if card in ['J', 'Q', 'K']:
        total += 10
    elif card == 'A':
        total += 1
        acces += 1
    else:
        total += int(card)
if total <= 11:
    if acces:
        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 Learn more OK, Got it.

Kaggle uses cookies from Google to deliver and enhance the quality of its services and to analyze traffic.

CLOVE

10:33 2/21/2025