

QHack

Quantum Coding Challenges

RANK

TEAM

CHALLENGES

SUBMISSIONS

SUPPORT

CHALLENGE COMPLETED

[View successful submissions](#)

▼ Jump to code

— Collapse text

1. It Adds Up

0 points

Welcome to the QHack 2023 daily challenges! Every day for the next four days, you will receive two new challenges to complete. These challenges are worth no points — they are specifically designed to get your brain active and into the right mindset for the competition. You will also learn about various aspects of PennyLane that are essential to quantum computing, quantum machine learning, and quantum chemistry. Have fun!

Tutorial # 1 — Adding two numbers

By completing this tutorial problem, you will gain confidence with interacting with this interface for submitting solutions to the challenges.

The code block below intentionally contains editable sections as well as non-editable sections to ensure that vital pieces of the entire code are not modified — they are essential to evaluating your solution. Within the editable sections, you must write your solution to the given problem. When you're ready to submit your solution, you can do so by clicking the "Submit" button at the bottom-right of this page. Your solution will run against the public inputs and outputs given by `test_cases` as well as test cases that are private (not visible to you).

When you submit your solution, its status will automatically pop-up on the screen. To see your submission history, you can click the [Submissions](#) tab near the top of this page. There, all of your submissions to our grader service will be displayed.

What should you do if your submission isn't passing? It would be nice to be able to edit the *entire* code block, not just what's editable here. To do this, you have a couple of options:

- At the bottom of the page, if you click the button that says "Open Notebook", this will open a notebook interface on Xanadu Cloud that functions exactly the same as a standard Jupyter Notebook.
- You can also use the "Copy all" button to copy the challenge code to your clipboard, paste it into your favourite IDE — e.g., VSCode, Sublime, VIM, etc. — and run the challenge on your own device.

In either of these cases, modify the code to your heart's content, noting that you can only paste code back into the editor here in the designated editable area.

To get you familiar with how this process works, try completing the function below and submitting your answer.

Challenge code

In the code below, you are given a function called `add_numbers` . **You must complete this function.**

Input

As input to this problem, you are given a `list(float)` containing two numbers to add together: `x` and `y` .

Output

This code must output `x + y` (`float`).

If your solution matches the correct one within the given tolerance specified in `check` (in this case it's a `1e-4` relative error tolerance), the output will be `"Correct!"` Otherwise, you will receive a `"Wrong answer"` prompt.

Good luck and have fun with all of the coding challenges!

Code

?

Help

▼

1

import json

2

import pennylane as qml

3

import pennylane.numpy as np

▼

4

Welcome to the QHack 2023 coding challenges!

5

▼

6

def add_numbers(x, y):

7

"""This function adds two numbers together.

8

9

Args:

10

11

x (float): A number.

12

y (float): A number.

13

14

Returns:

15

(float): The result of adding x and y.

16

"""

▼

16

Put your code here

17

return x+y

18

▼

19

These functions are responsible for testing the solution.

20

▼

21

def run(test_case_input: str) -> str:

22

x, y = json.loads(test_case_input)

23

result = add_numbers(x, y)

24

return str(result)

25

▼

26

def check(solution_output: str, expected_output: str) -> None:

27

solution_output = json.loads(solution_output)

28

expected_output = json.loads(expected_output)

29

assert np.allclose(

30

solution_output, expected_output, rtol=1e-4

31

), "Your addition function isn't quite right!"

32

▼

33

test_cases = [['[0.2, -0.2]', '0.0']]

▼

34

▼

35

for i, (input_, expected_output) in enumerate(test_cases):

36

print(f"Running test case {i} with input '{input_}'...")

37

▼

38

try:

39

output = run(input_)

40

▼

41

42

except Exception as exc:

43

▼

44

print(f"Runtime Error. {exc}")

45

▼

46

else:

47

▼

48

if message := check(output, expected_output):

49

print(f"Wrong Answer. Have: '{output}'. Want: '{expected_output}'.")

50

51

else:

52

print("Correct!")

▼

Copy all

Reset

Open Notebook

Submit

▼