Python for Beginners

JetBrains Academy

# Introduction Python for Beginners

Topics completed: **0 / 50 (0%)**

You'll complete the track in **15 hours**.

Activities left: **63.**

## What is an Activity?

Activity is a single element of study plan, for example topic, stage implementation, or survey. Note: this total number is updated every 5 minutes.

## Problem of the day

No problems to solve. Complete more topics to unlock this feature and repeat what you've learned.

## Study plan

Goal to become a Python for Beginners

**Your current project:** [Zookeeper](https://hyperskill.org/curriculum#about)

Stages completed: **0 / 4**.

You'll complete the current stage in 25 minutes, the project in 5 hours.

## Zookeeper

### About

This short code challenge can help the local zoo look after its denizens. You will create a tool for monitoring animals and their status.

### Learning outcomes

This project is aimed at our beginners. It helps you understand some syntax basics and learn how to work with variables, data storage types such as lists, and while loops.

### What you’ll do and what you’ll learn

#### Stage 1/4: Rush into print.

First, let's try to get some simple output from your code.

* Introduction to Python
* [Overview of the basic program](https://hyperskill.org/learn/step/5208" \t "_self)
* [Multi-line programs](https://hyperskill.org/learn/step/5233" \t "_self)
* [Stage implementation](https://hyperskill.org/projects/98/stages/539/implement)

#### Stage 2/4: Show me an animal!

Show the zookeeper an image of her ward.

#### Stage 3/4: What's inside?

The zookeeper wants to know what is inside each habitat, by its number.

#### Stage 4/4: Sustainable care <3

Finally, your program can work for as long as needed!

### Result will look like

(see [demonstration.mp4](file:///C:\Users\maham\AppData\Roaming\Microsoft\Word\JetBrains%20Academy%20-%20Python%20for%20Beginners\demonstration.mp4))

### Stage 1/4: Rush into Print

First, let's try to get some simple output from your code.

#### What you will do in this stage 1/4 : Rush into print

Project: [Zookeeper](https://hyperskill.org/projects/98)

 Medium  2 minutes

##### Description

There are many animals in the zoo, and all of them need care. The animals must be fed, cleaned, surrounded by their kin, and kept happy. That is a difficult task for our large zoo, so one of your employers has suggested a more convenient way to keep track of everything. She wants to be able to pull up a video feed of any animal in the zoo with the help of a program. Being able to check on each habitat would help the zookeepers take care of our furry friends more efficiently!

In this project, you will create a program that helps the zookeepers check on the animals and make sure that they're doing well. Your product will be able to process commands from the zookeepers and display the animals on a monitor.

##### Objectives

To begin with, you will develop a simple printer. Your program should display the text from the output example.

##### Example

###### Output:

1. I love animals!
2. Let's check on the animals...
3. The deer looks fine.
4. The bat looks happy.
5. The lion looks healthy.

You are starting a new topic

Study the theory, answer questions, solve coding problems. When you complete all the topics, you can continue with the stage implementation.

#### Theory: Introduction to Python

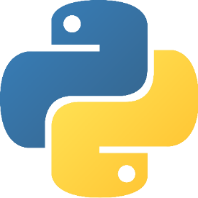
 3 minutes0 / 3 problems solved

##### What is Python?

**Python** is a modern general-purpose programming language initially developed by a Dutch programmer named Guido van Rossum in the late 1980s. The name comes from the popular Monty Python show, not the snake as you might think. This language has a clean, uniform and well-readable syntax and is **designed to be easy to learn and use in practice**.

Nowadays, Python is one of the most popular programming languages worldwide according to the [TIOBE](https://www.tiobe.com/tiobe-index/) index and the number of programmers who use it is growing every day. The language has a huge community of developers around the world. If you have a problem, you can always ask other programmers for help or find a suitable answer on a site like [Stack Overflow](https://stackoverflow.com/questions/tagged/python).

Developing software with Python is easy and fun :)



**The Python logo**

Python has a wide range of possible applications, especially in:

* web development
* data science (including machine learning)
* scripting (task automation, e.g. text processing or a simulation of typical user actions)

Less commonly, it is also used in **desktop development**.

**Current topic:** [Introduction to Python](https://hyperskill.org/learn/step/5204) (Stage 1)

**Topic is required for**[: Overview of the basic program](https://hyperskill.org/learn/step/5208) (Stage 1) [and Intro to Machine Learning](https://hyperskill.org/learn/step/10403" \t "_self)

**Table of contents:**

[↑ Theory: Introduction to Python](https://hyperskill.org/learn/step/5204#step-title)

##### Python in data science

Python's huge popularity in recent years is mostly due to its use in data science. What makes it better than other languages for this purpose? Well, there're several reasons:

* its simple syntax allows people from non-programming backgrounds to use it for data processing and model training without spending much time learning a new language;
* Python supports a very large number of third-party libraries for machine learning, neural networks, statistics, and numeric calculations, which makes your job much easier;
* with Python, it is possible to collect, clean, and explore data, as well as train models and visualize the results — all in one setting;
* the Python ML developer's community is very large, so you can always find support for your tasks.

As you can see, Python does have a lot to offer for data science enthusiasts.

##### Short history of Python

Like other programming languages, Python has gone through several versions. Python 1.0 was released in 1994 and laid the basic principles of the language with emphasis on simplicity.

Python 2.0 was released in 2000. This version has become very popular among programmers. Different 2.x subversions (2.6, 2.7) are still used in various projects and libraries. The symbol **x** in 2.x means any subversion of Python 2.

Python 3.0 was the next major version released in 2008. Its broke backward compatibility with its predecessors in order to rid the language of historic clutter and make Python more readable and consistent.

So, today two similar but incompatible versions of Python are commonly in use. **Throughout this course, we will learn Python 3.x**.

##### First program example

Here is a single line of Python code that prints Learn Python to be great! .

print("Learn Python to be great!")

Now, you do not need to understand how this code works, just start to appreciate the syntax looking like English :)

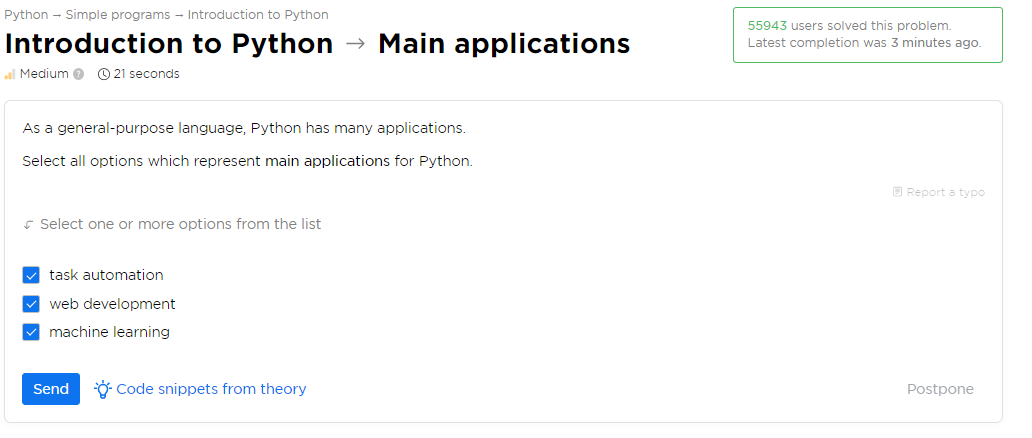
#### Simple Programs

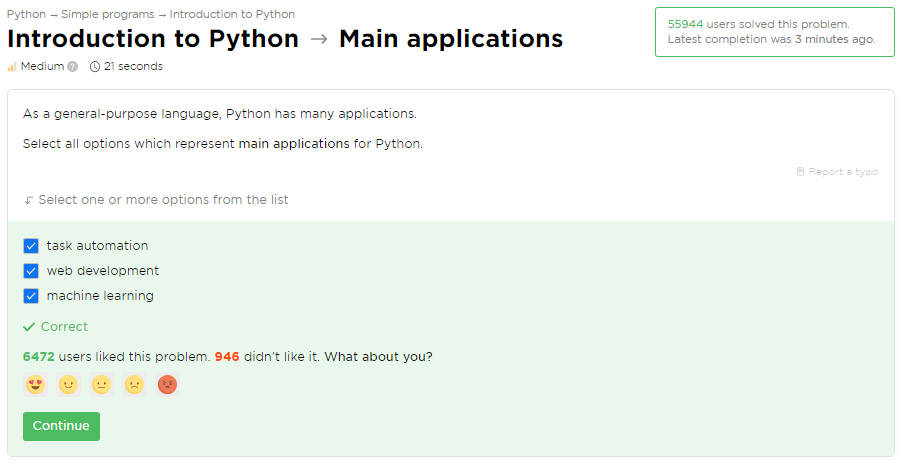
##### Introduction to Python -> Versions of Python





##### Introduction to Python -> Main applications

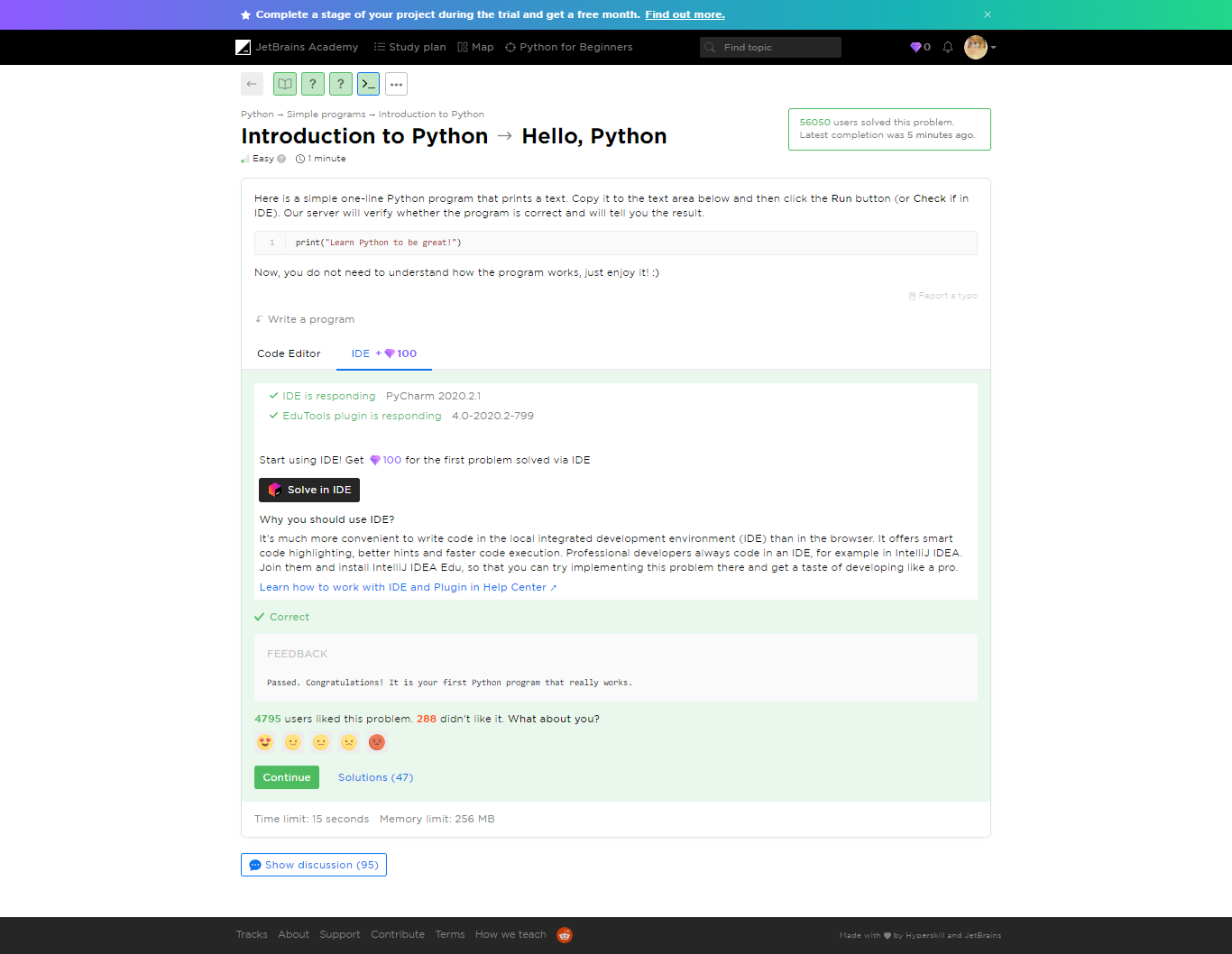




##### Introduction to Python -> Hello, Python





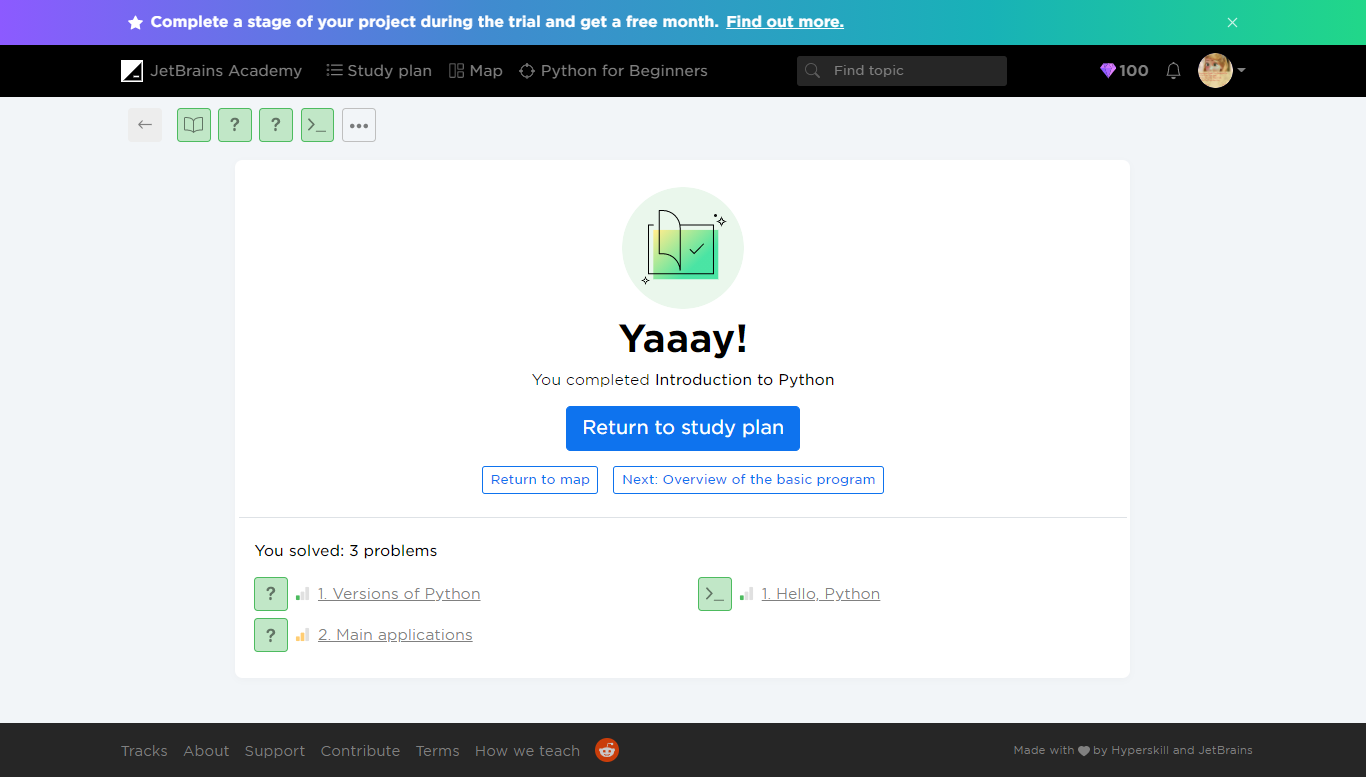


#### Yaaay!

You completed **Introduction to Python**

**You solved: 3 problems**

1. Versions of Python
2. Main applications
3. Hello, Python



#### Theory: Overview of the basic program

 6 minutes 0 / 5 problems solved

**Table of contents:**

[↑ Theory: Overview of the basic program](https://hyperskill.org/learn/step/5208#step-title)

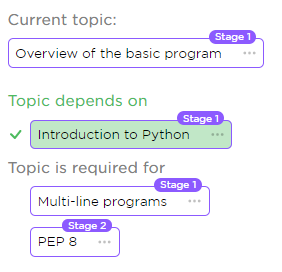
[§1. The Hello World program](https://hyperskill.org/learn/step/5208#the-hello-world-program)

[§2. Short explanation](https://hyperskill.org/learn/step/5208#short-explanation)

[§3. Printing quotes](https://hyperskill.org/learn/step/5208#printing-quotes)

[§4. Possible errors](https://hyperskill.org/learn/step/5208#possible-errors)

In this topic, you will learn how to develop your first Python programs. Despite the fact that these programs are quite simple, they are still syntactically correct and show that programming in Python is a treat.



##### The Hello World program

Our first example will be **Hello, World!** It is traditionally used to introduce beginners to a new programming language.

print("Hello, World!")

As you can see, it consists of a single line and just prints a string passed in the parentheses, but without quotes. You may run this code [online](https://repl.it/languages/python3) (just copy it and click on the triangle) or follow these [installation tips](https://www.python.org/about/gettingstarted/#installing). You should get this result:

Hello, World!

Although this code is very simple, we will go through it in some more detail.

##### Short explanation

Here,**print** is the name of a function. A **function** is a block of code that does some useful work for you, e.g. prints a text. In some sense, a function is a subprogram that can be reused within your programs. When the name of a function is followed by parentheses, it means that it was **called** to get the result.

Let's go further, **"Hello, World!"** is a Python string. All strings are surrounded by either single or double quotes, so **'Hello, World!'** is also a valid string. You may replace this string with another one, and the program will print the new string. For example:

print('Python 3.x')

As you might guess, this program will print:

Python 3.x

##### Printing quotes

If you would like to include quotes into a string, then enclose this string in quotes of another type, e.g.:

print("Yes, I'm ready to learn Python.")

It prints:

Yes, I'm ready to learn Python.

You can try to run all the examples using the [link](https://repl.it/languages/python3) provided earlier. This will help you familiarize yourself with Python.

##### Possible errors

Even this simple line of code may contain errors, most common of them are:

* **putting extra indentation**

   print("Hello, World!")

This does not work because of extra spaces before **print**.

* **calling the function by the wrong name**

pint("Hello, World!")

This line contains **pint** instead of **print.** Make sure to refer to every function by its proper name.

* **writing names in the wrong case**

PRINT("All caps")

Again, **Print**, **print** and **PRINT** are not the same. Names are case-sensitive in Python.

* **missing one or both quotes for a string**

print("Python)

This does not work because of missing closing quotes.

* **missing one or more parentheses**

print("I have no end"

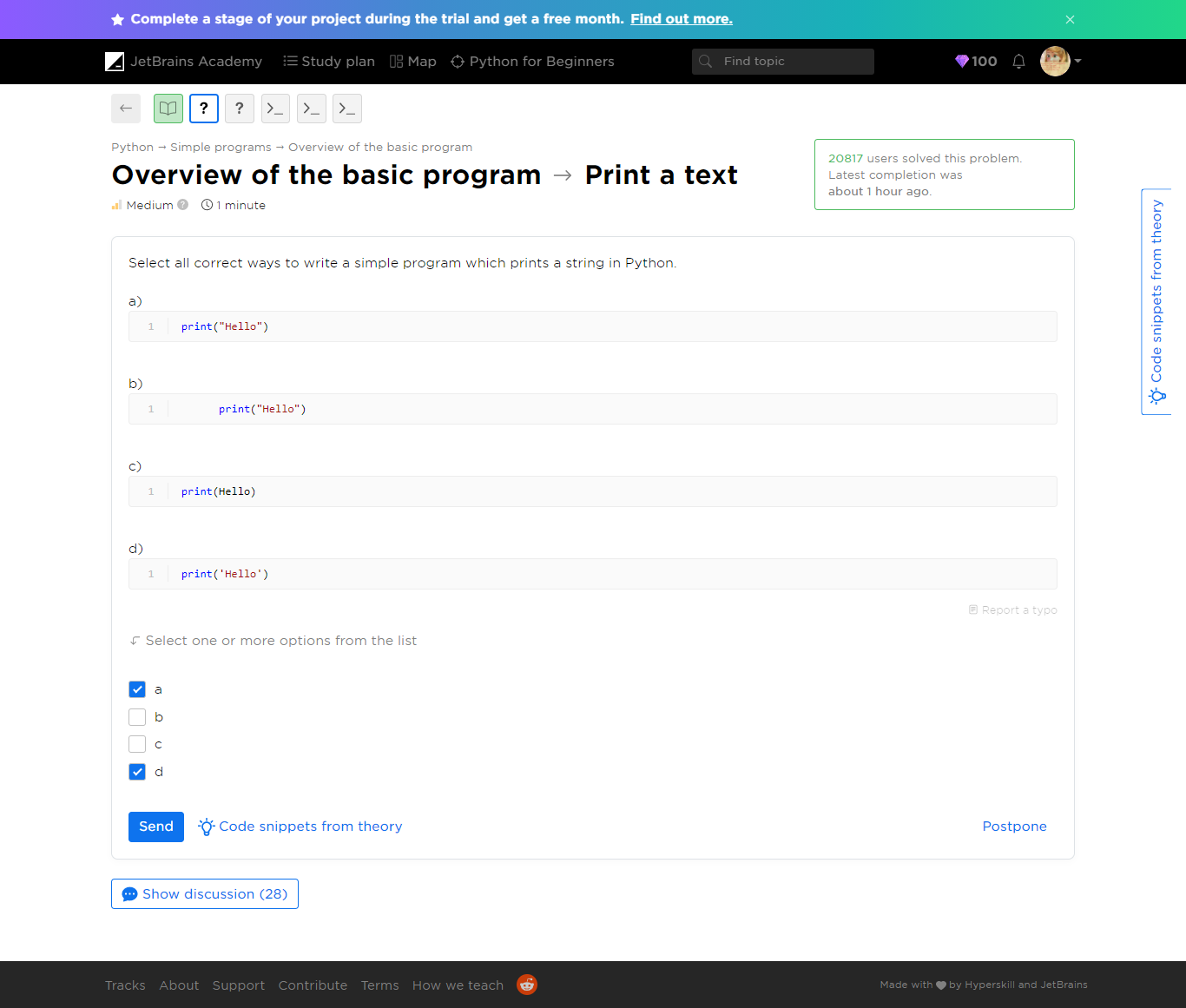
Be careful with parentheses, especially when calling a function.

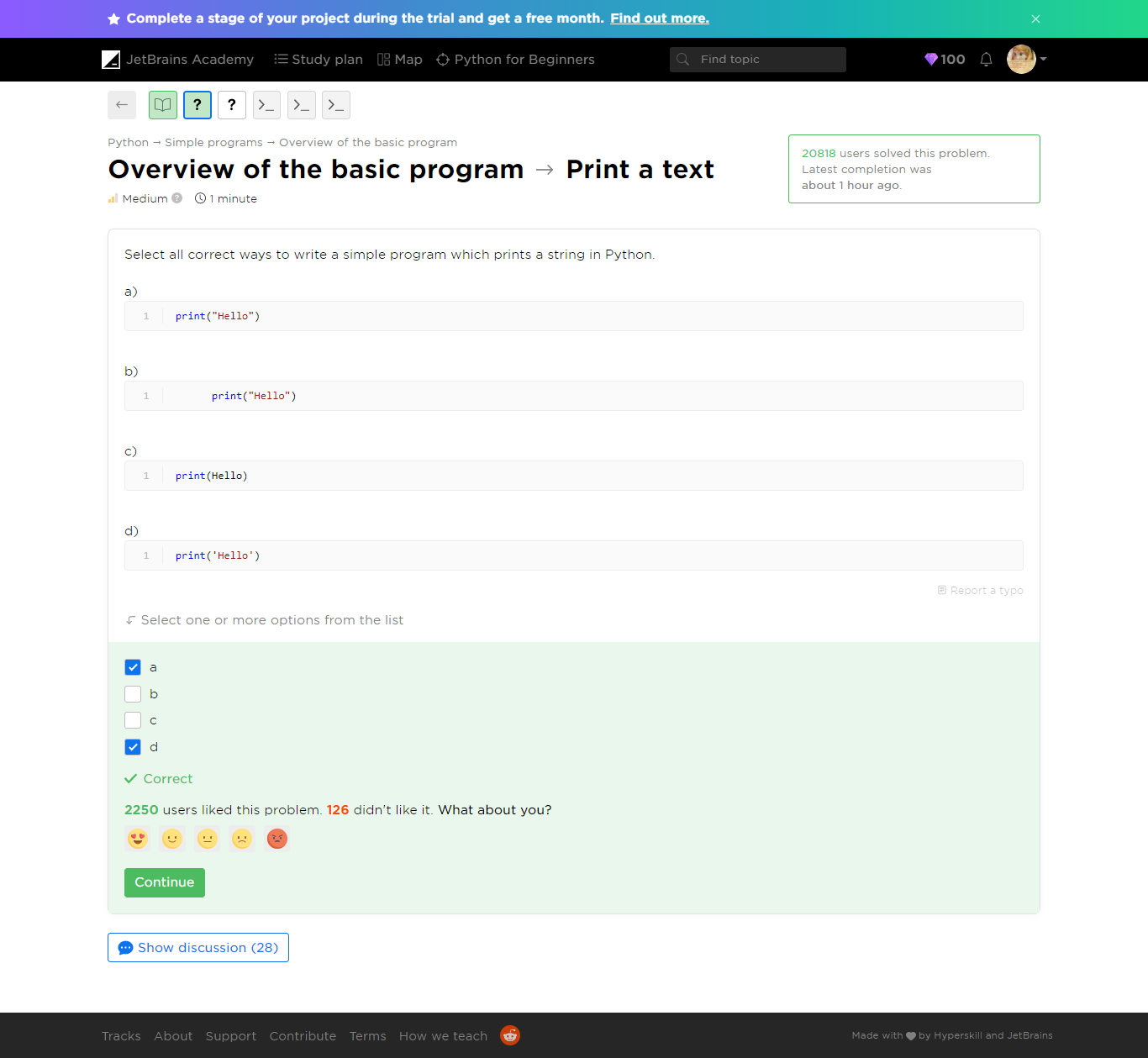
Now you shouldn't have any serious trouble with such programs.

#### Problems

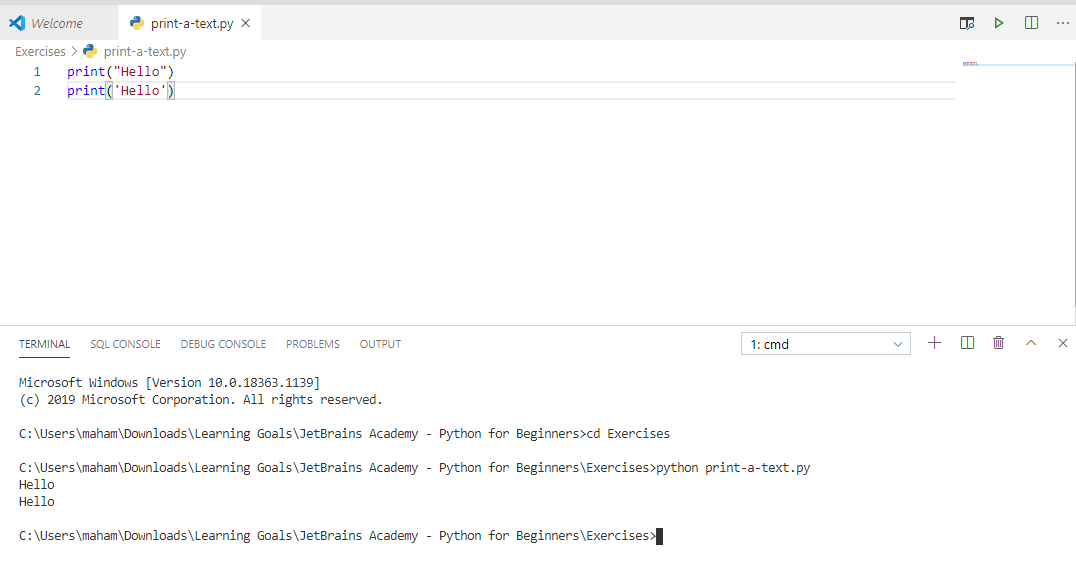
##### Overview of the basic program  -> Print a text

 Medium  1 minute

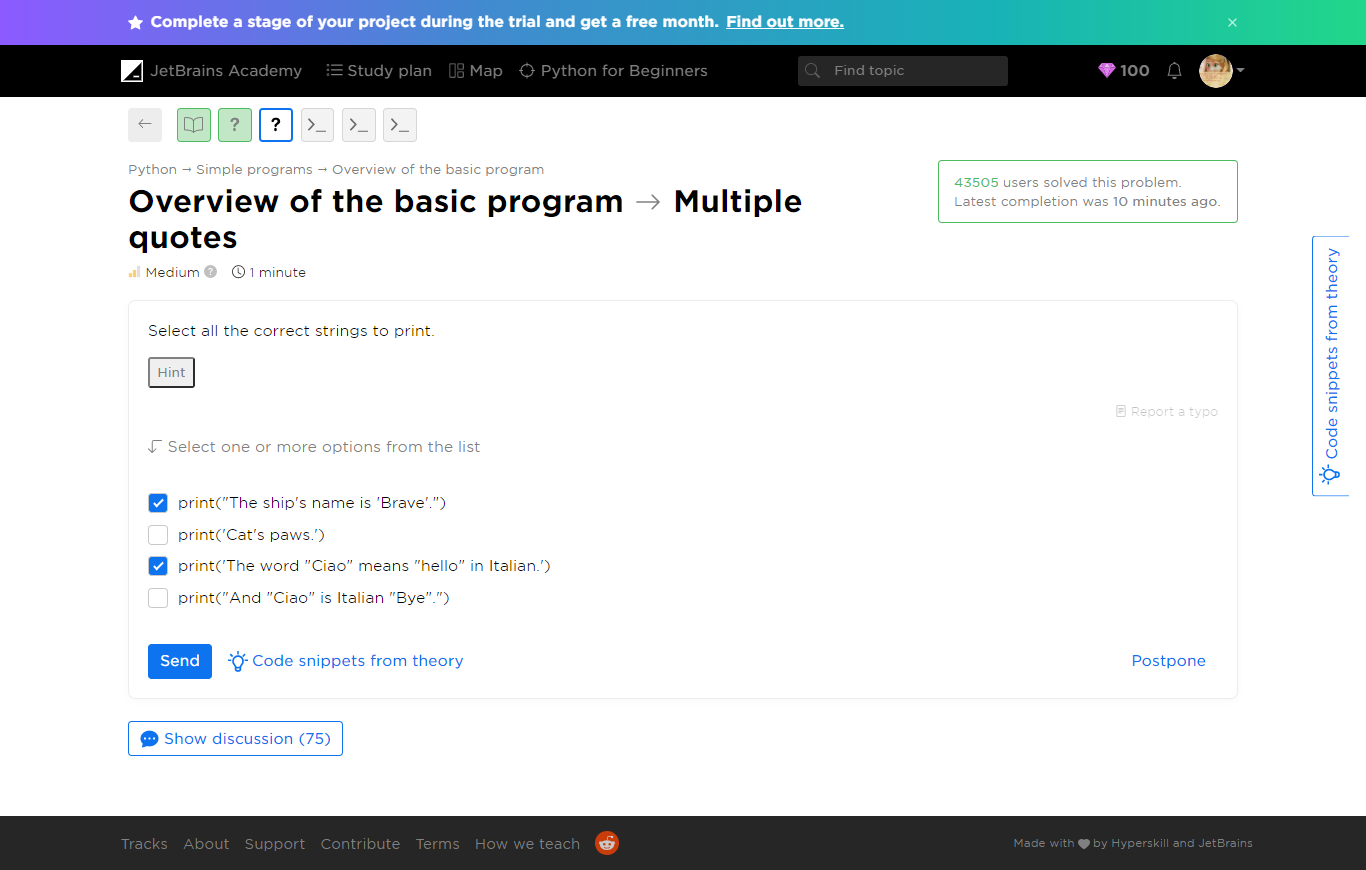


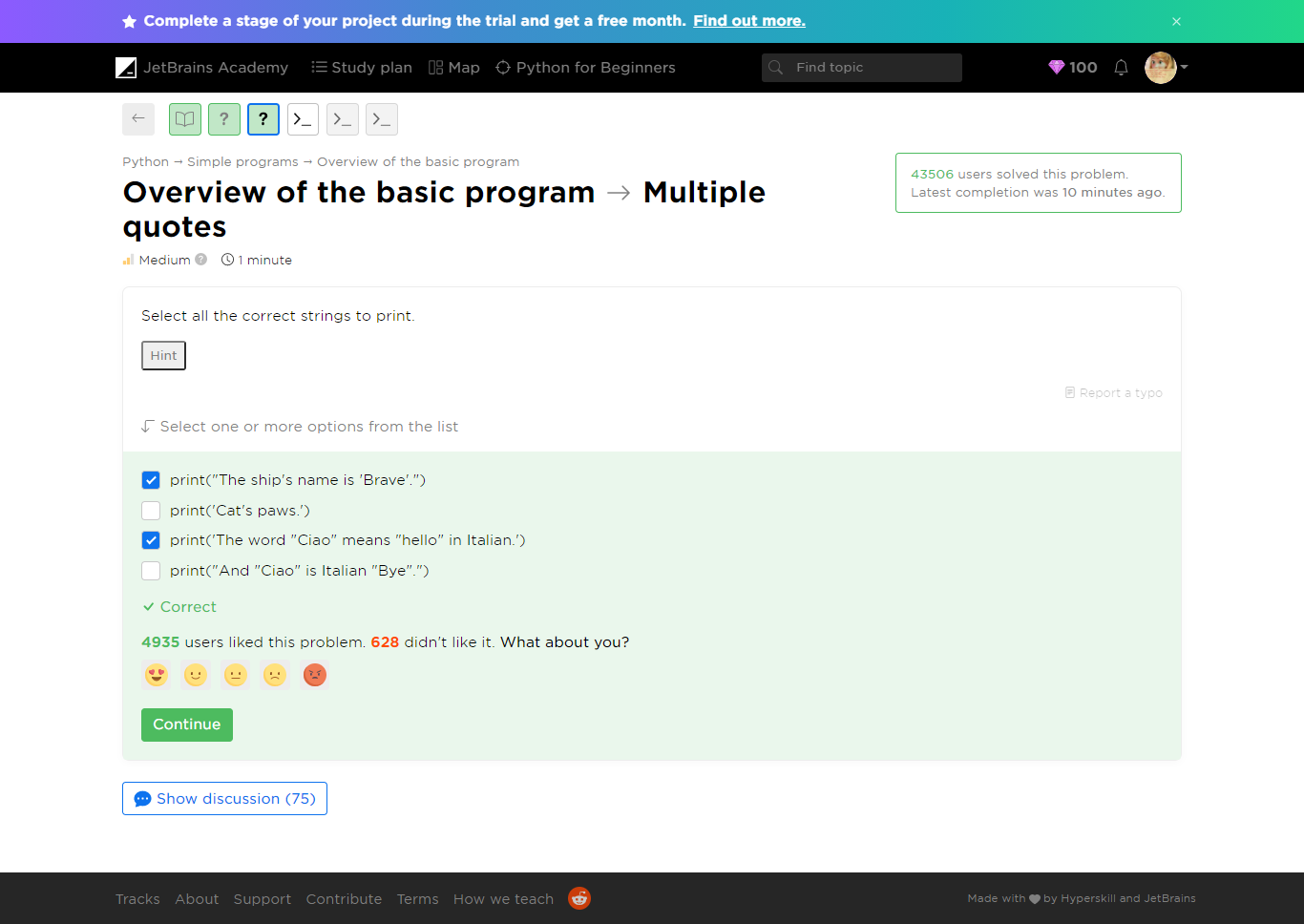


Refer to ***print-a-text.py*** file



##### Overview of the basic program  -> Multiple quotes





C:\Users\maham\Downloads\Learning Goals\JetBrains Academy - Python for Beginners\Exercises>python multiple-quotes.py

File "multiple-quotes.py", line 2

print("And "Ciao" is Italian "Bye".")

^

SyntaxError: invalid syntax

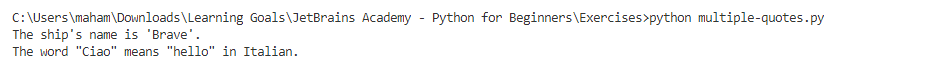
C:\Users\maham\Downloads\Learning Goals\JetBrains Academy - Python for Beginners\Exercises>python multiple-quotes.py

File "multiple-quotes.py", line 3

print('Cat's paws.')

^

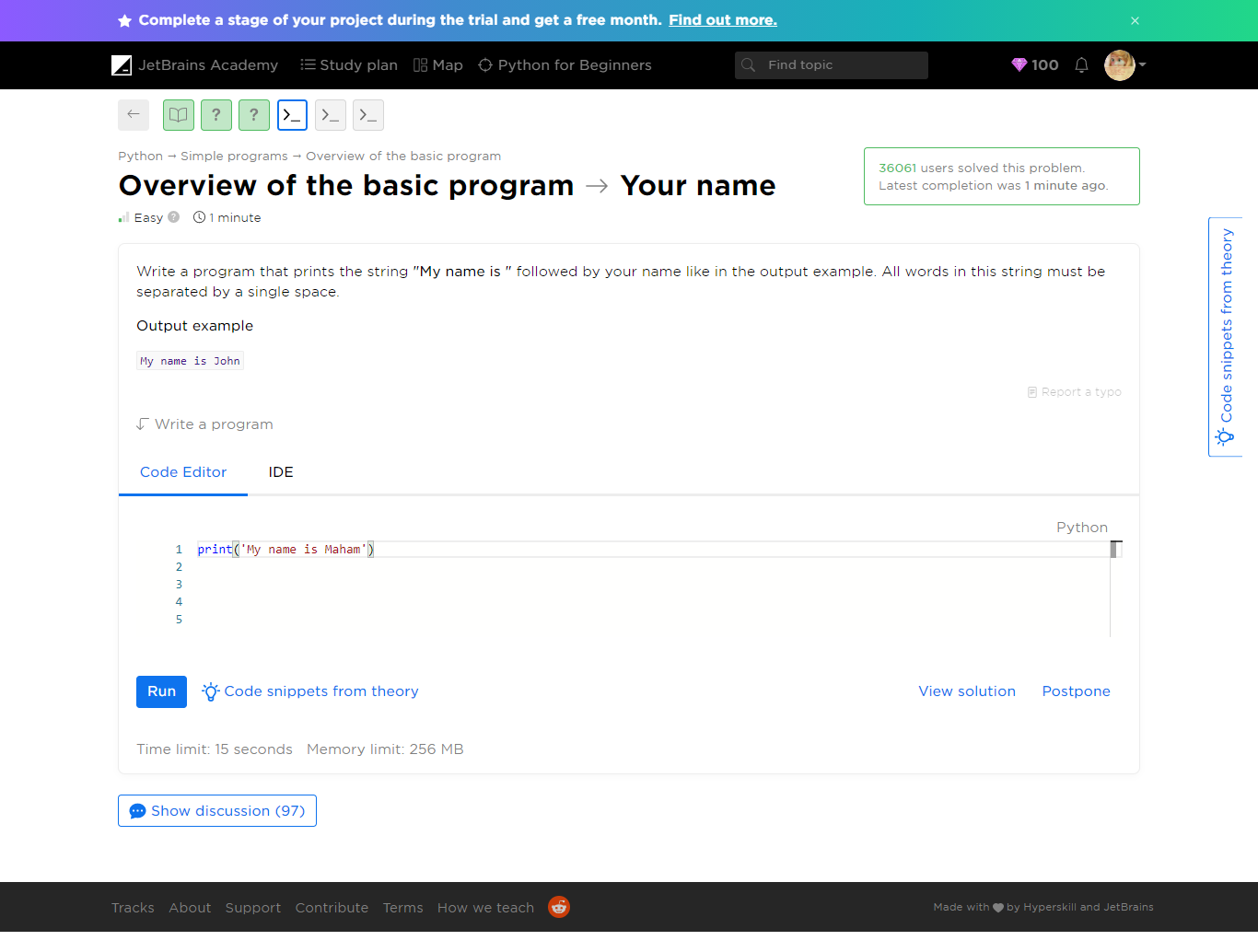
SyntaxError: invalid syntax

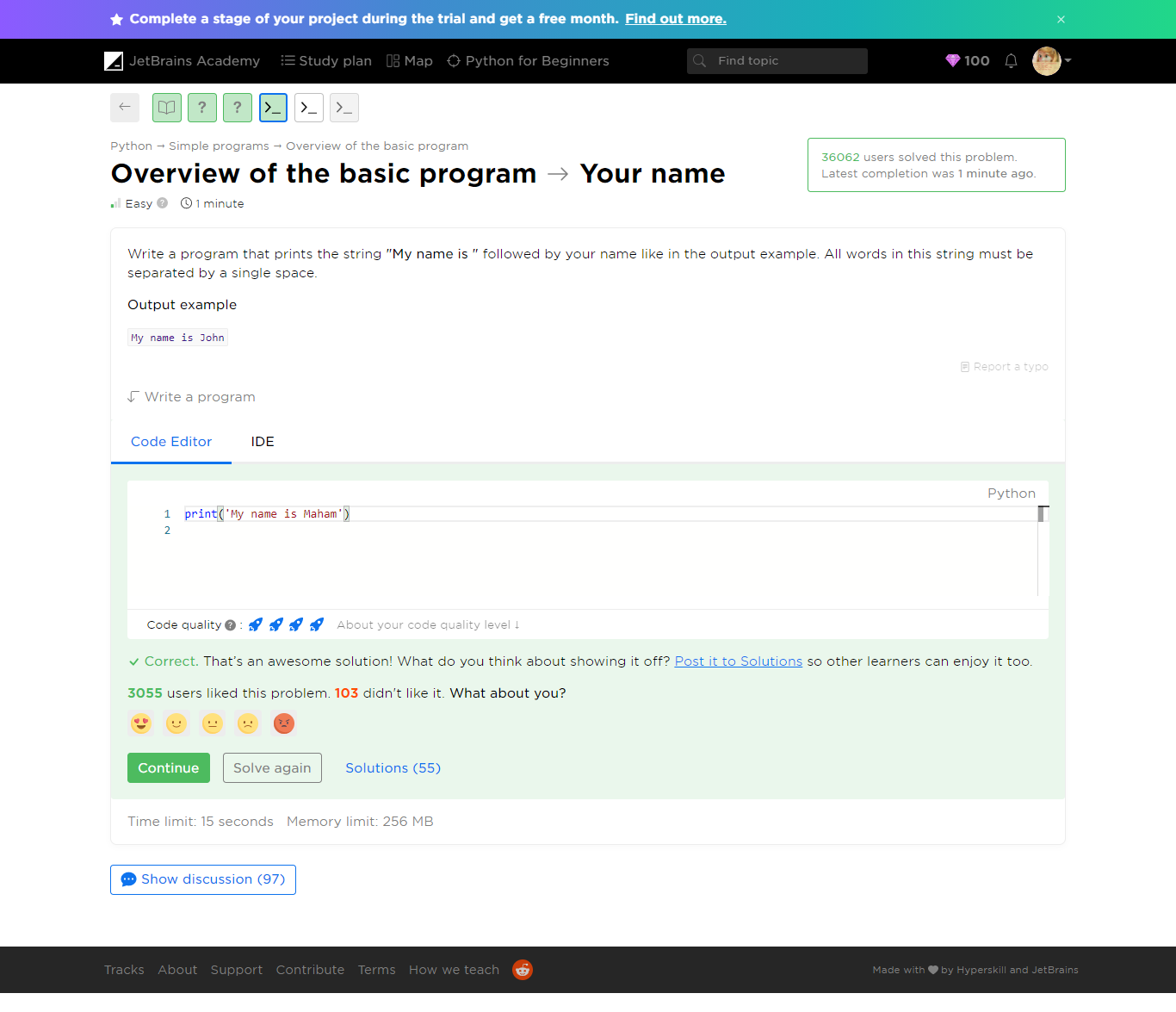


Refer to ***multiple-quotes.py***

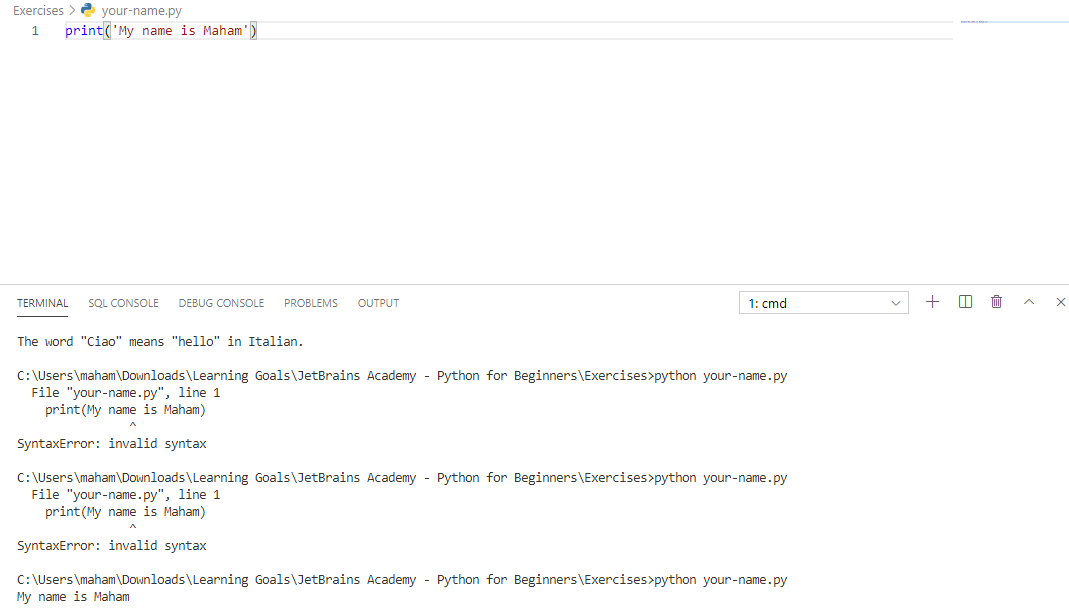


##### Overview of the basic program -> Your name

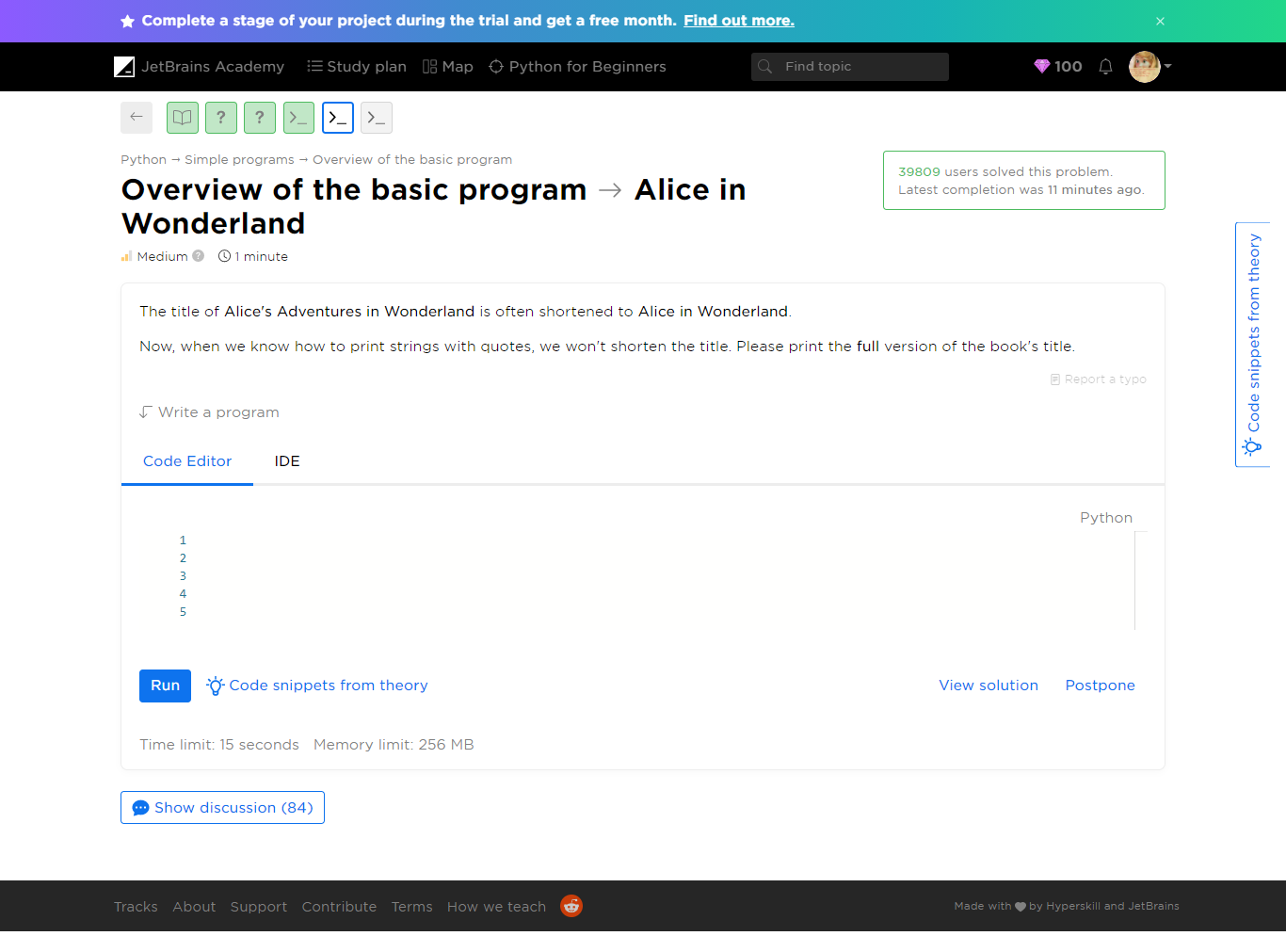


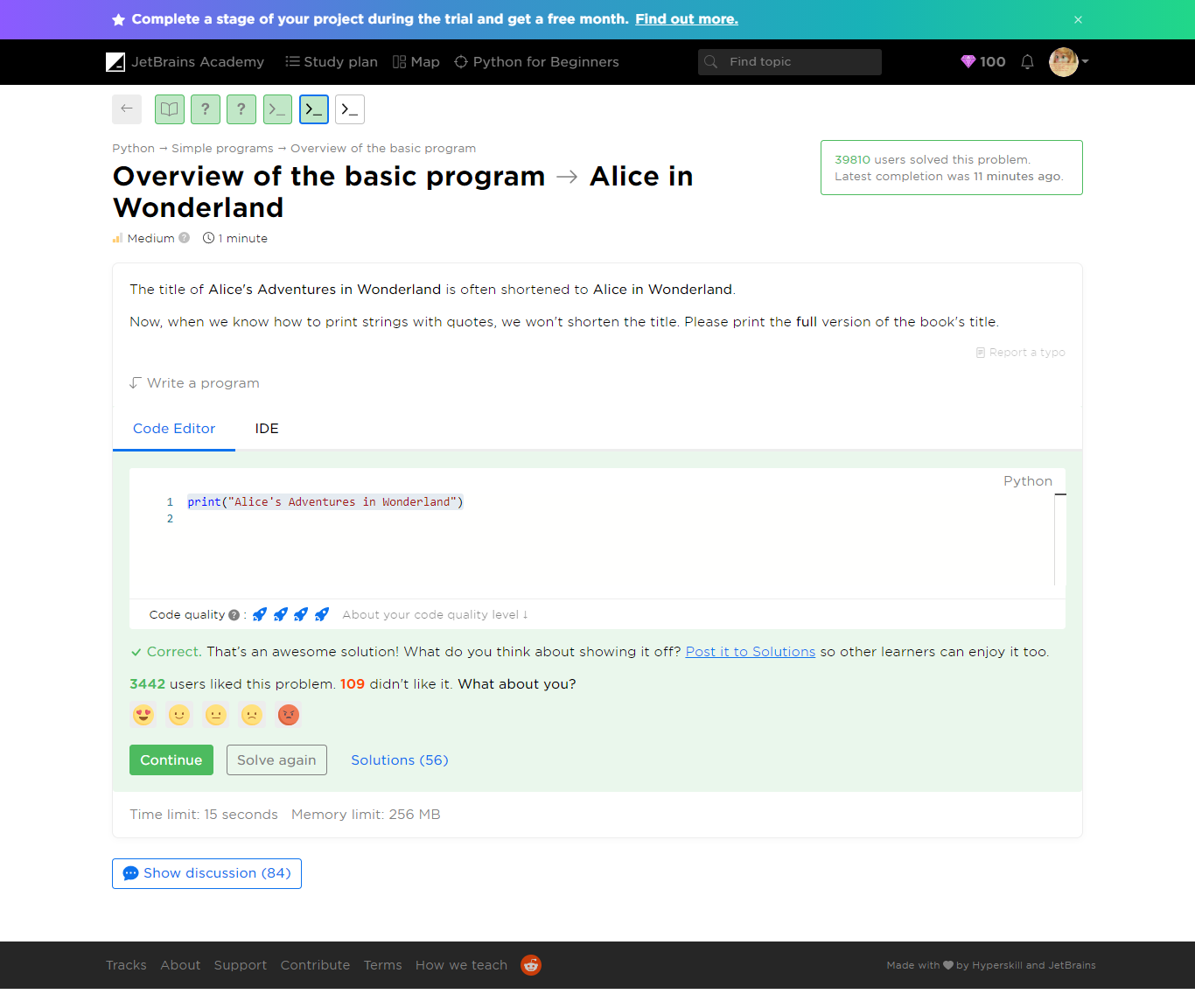


Refer to ***your-name.py***

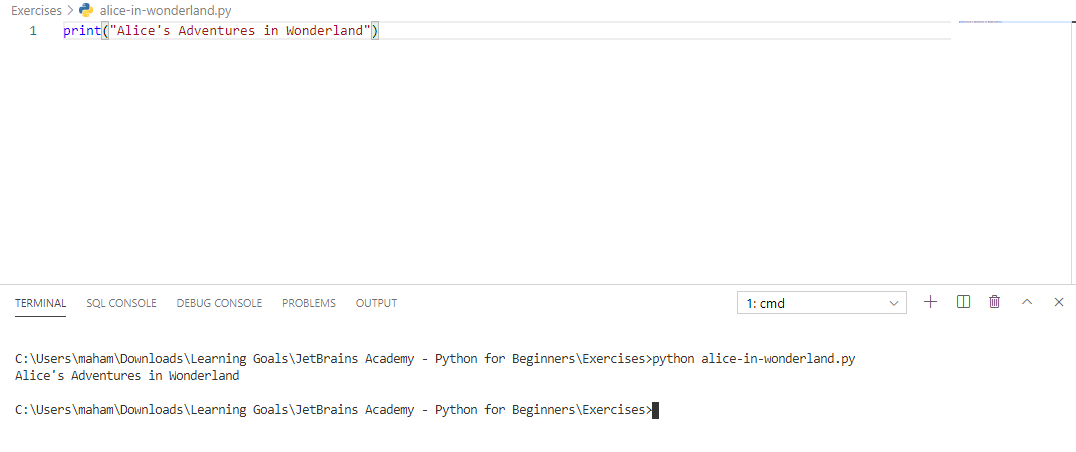


##### Overview of the basic program -> Alice in Wonderland

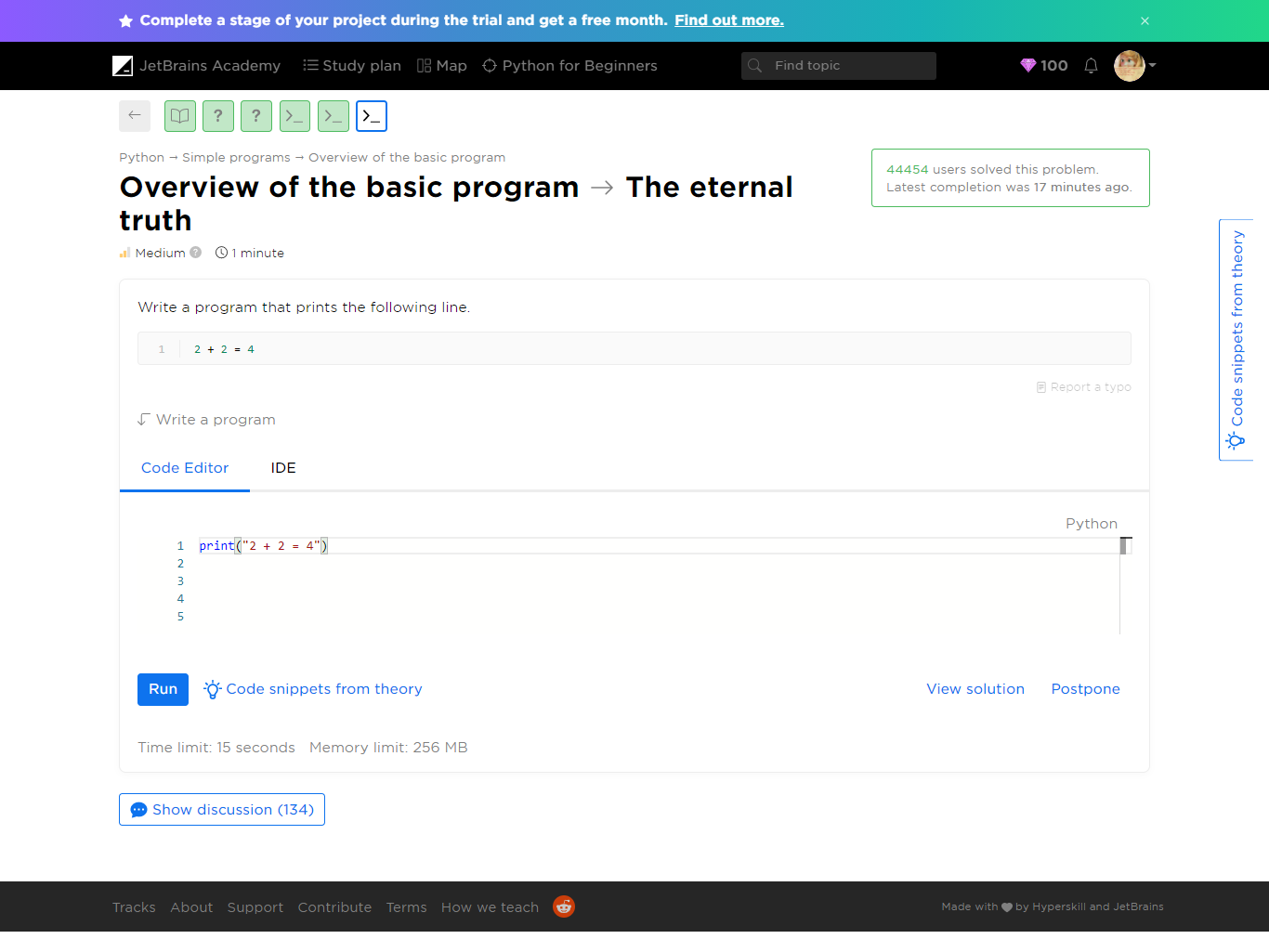


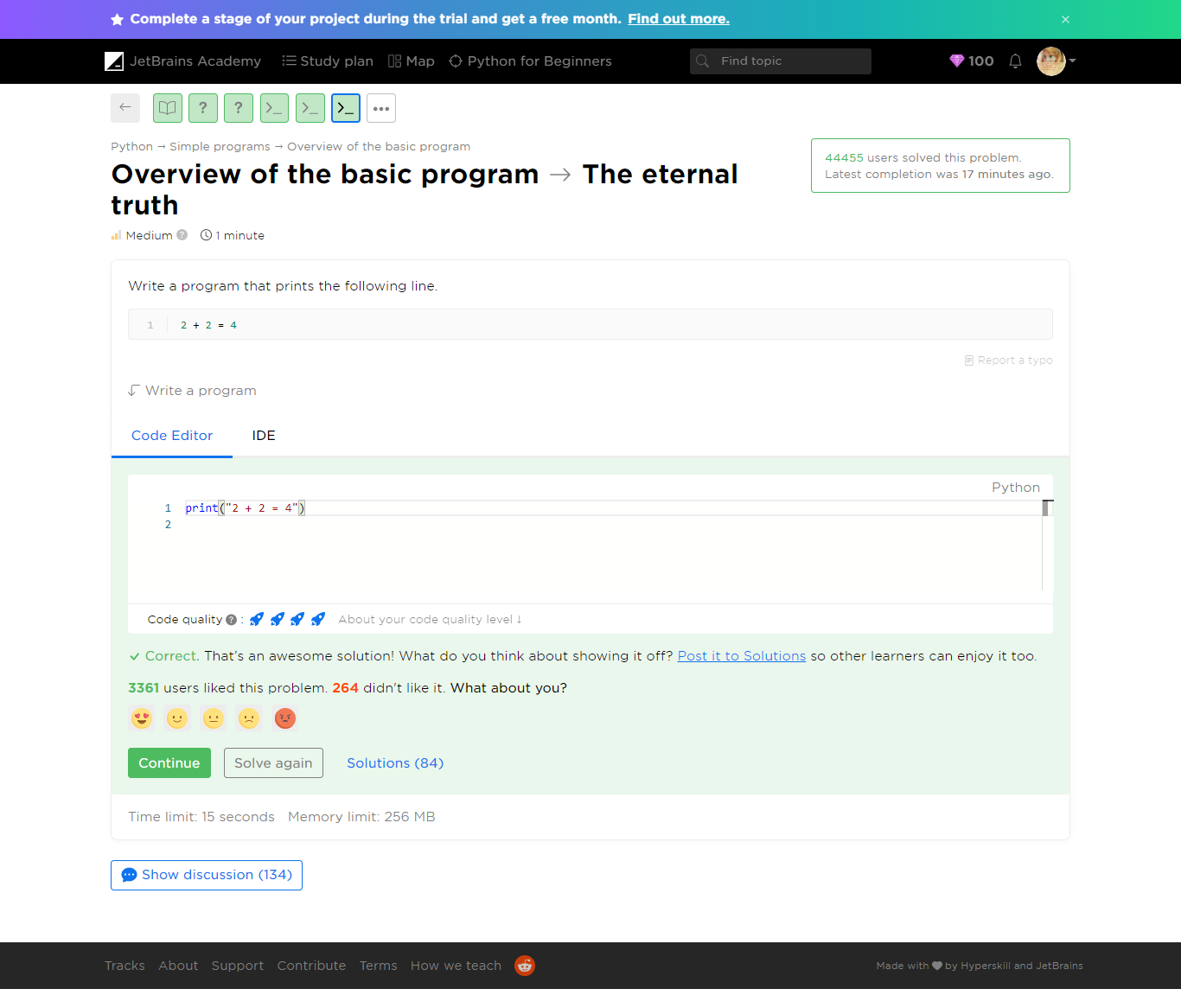


Refer to ***alice-in-wonderland.py***

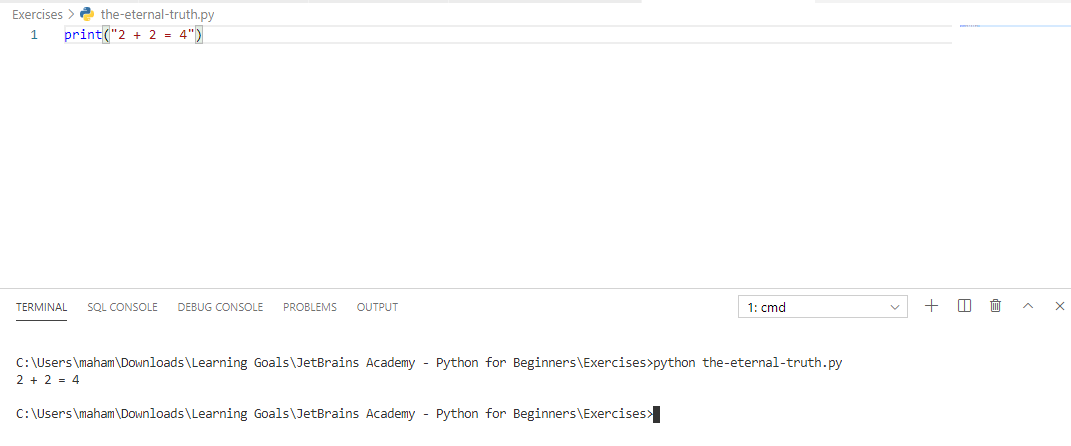


##### Overview of the basic program -> The eternal truth





Refer to ***the-eternal-truth.py***



#### Yaaay!

You completed **Overview of the basic program**

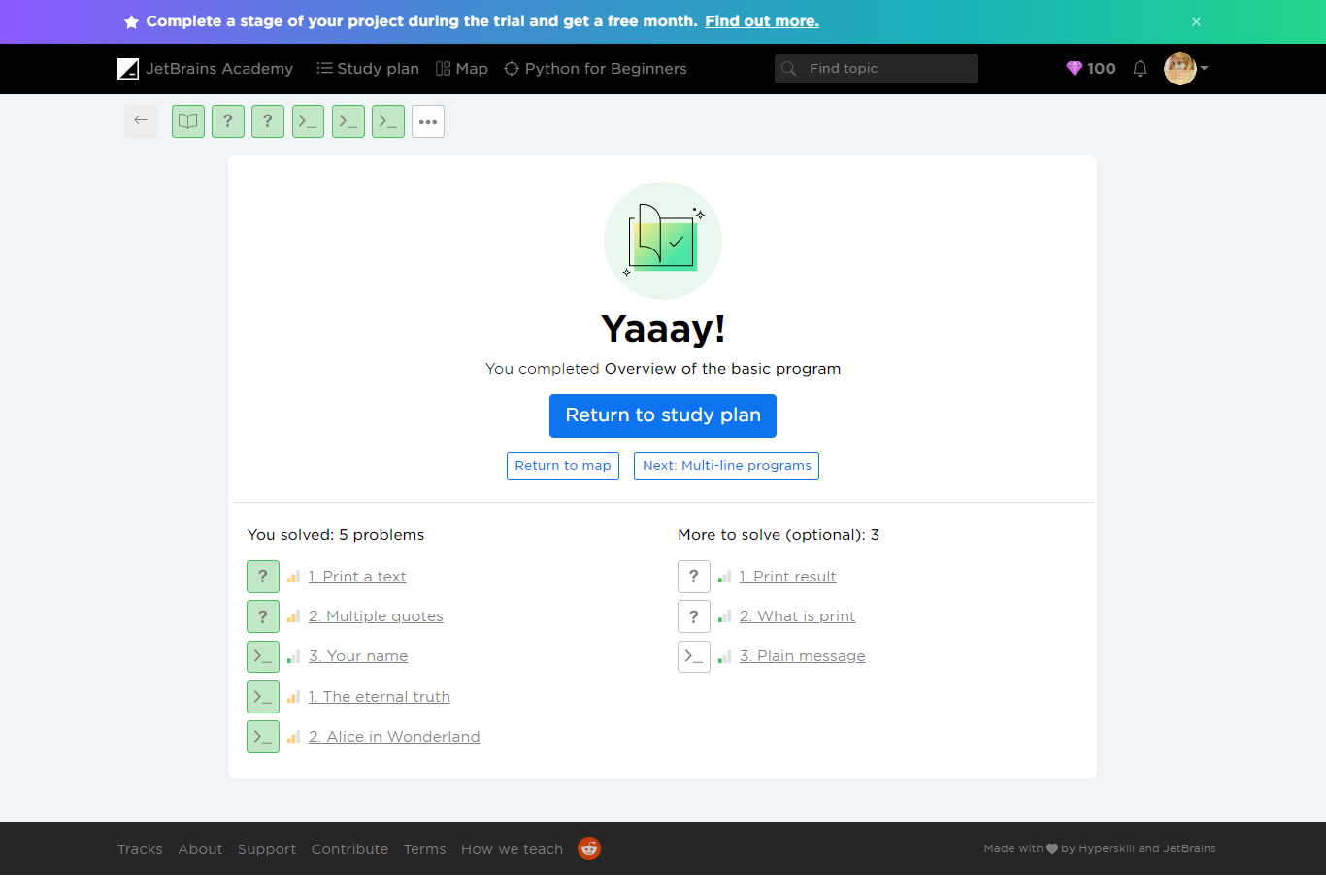
**You solved: 5 problems**

1. Print a text2. Multiple quotes3. Your name

1. The eternal truth2. Alice in Wonderland

**More to solve (optional): 3**

1. Print result2. What is print3. Plain message



#### Additional Problems

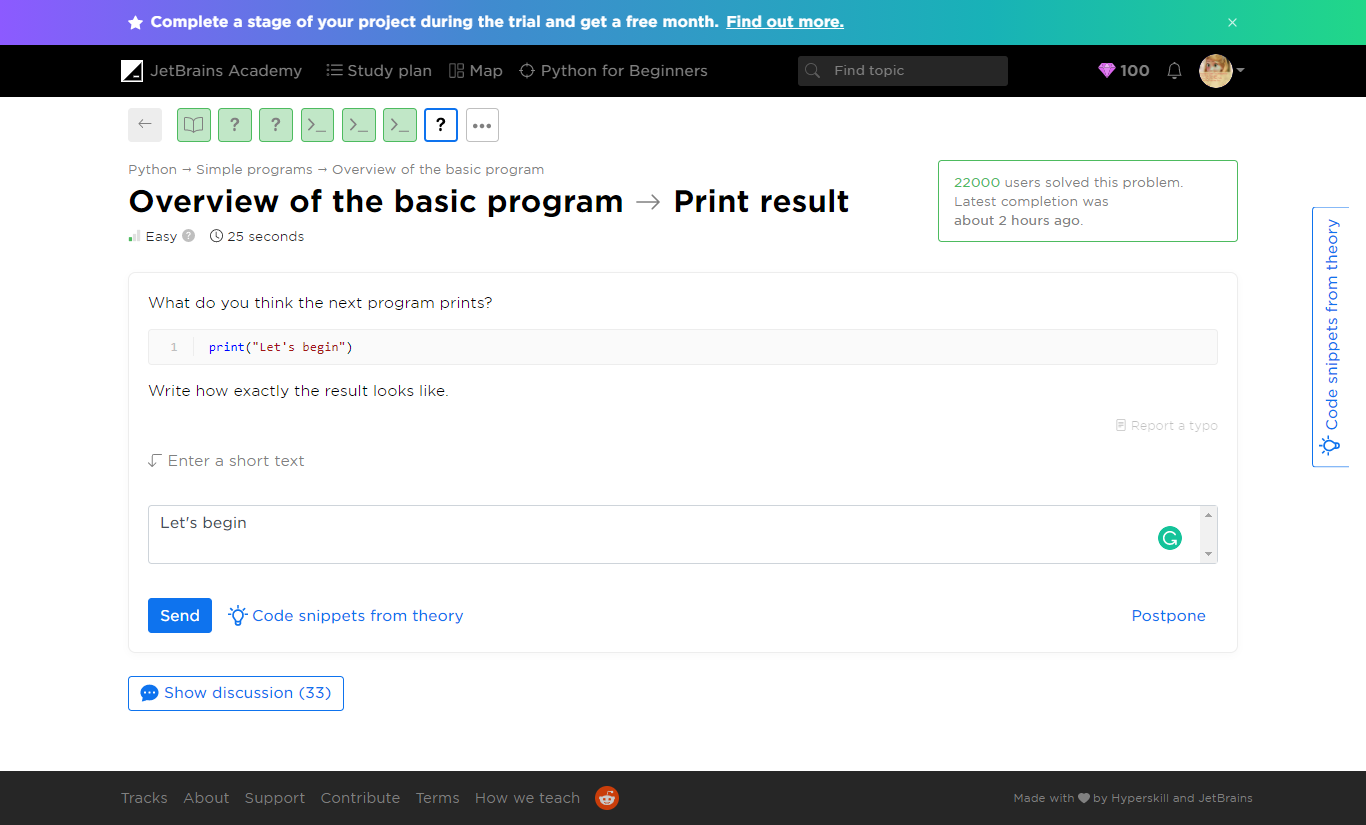
Optional Problems

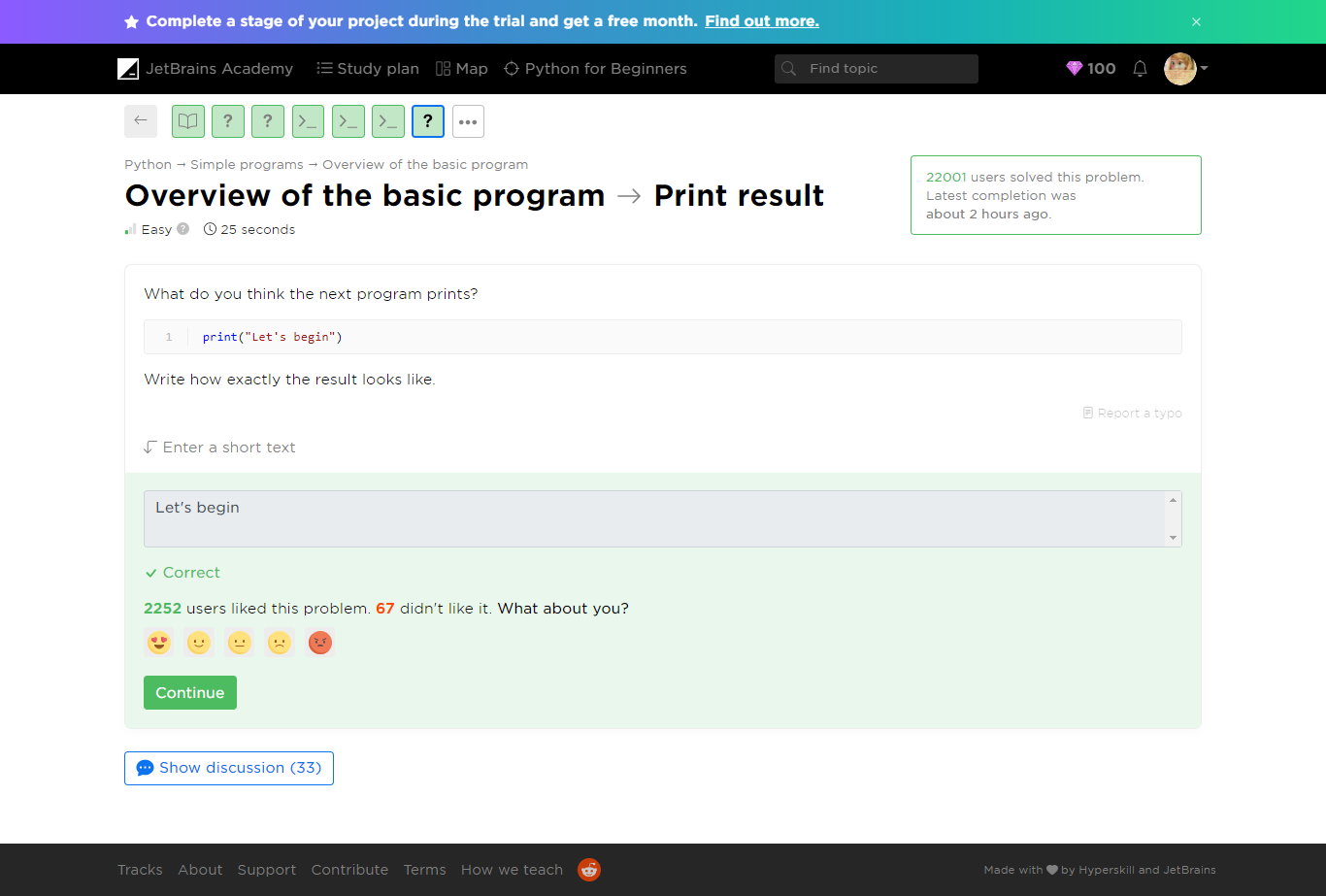
1. Print result

2. What is print

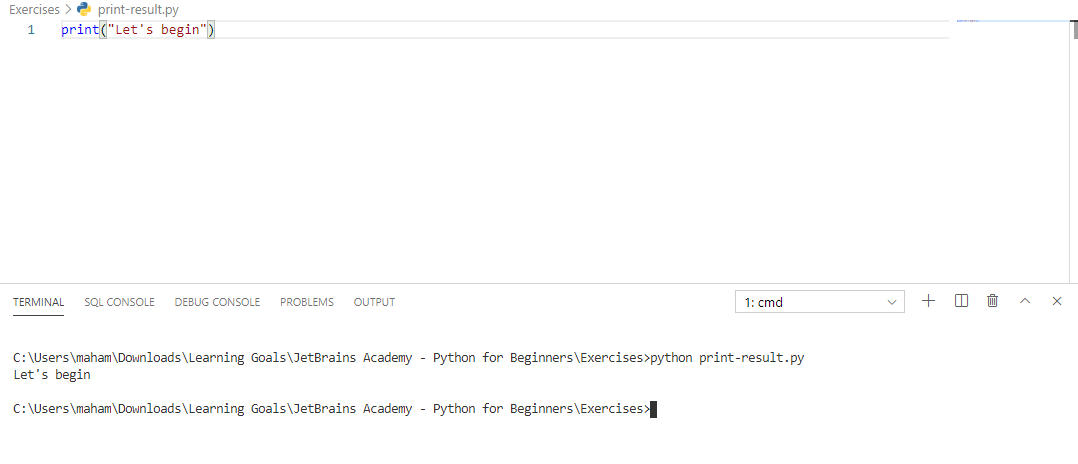
3. Plain message

##### Overview of the basic program -> Print result

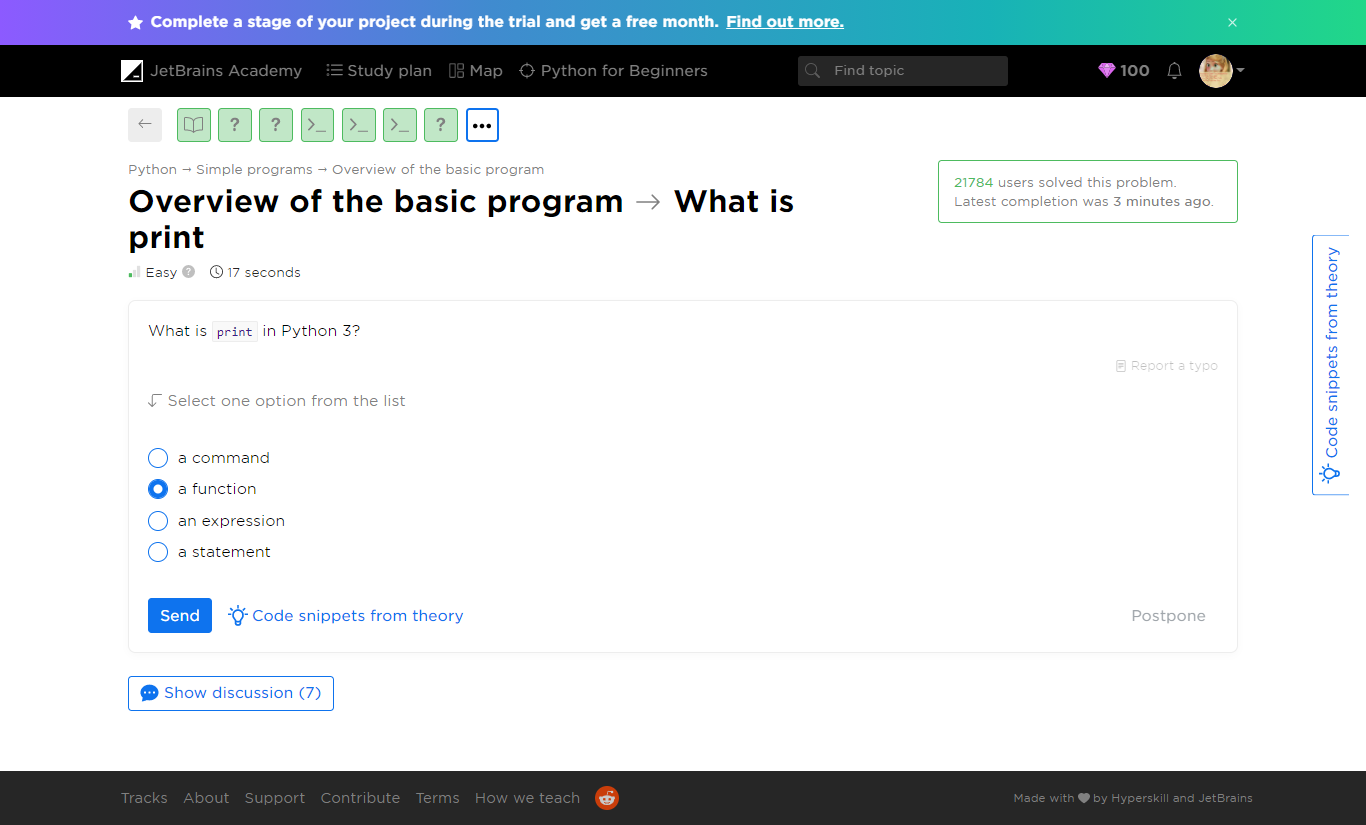


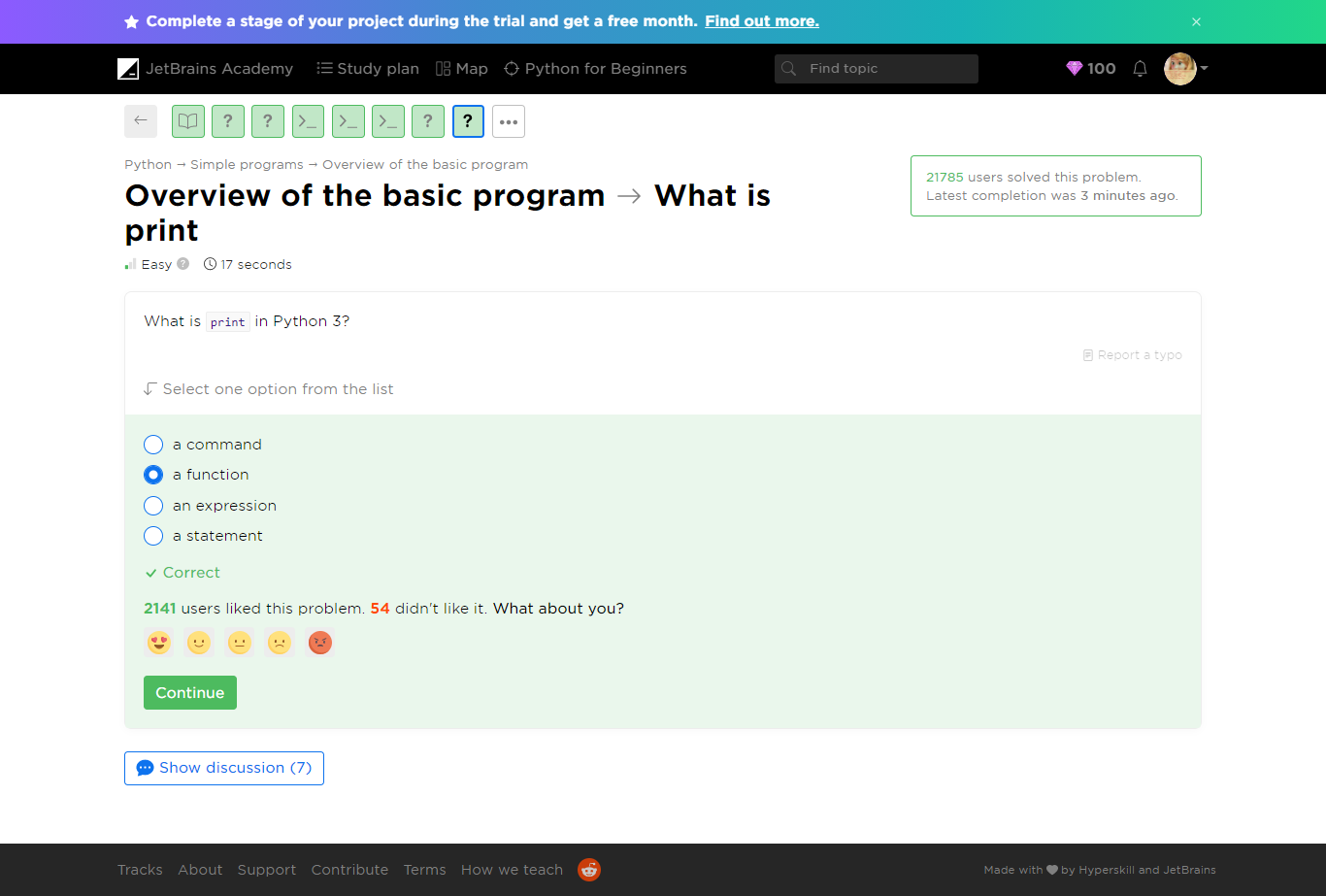


Refer to print-result.py

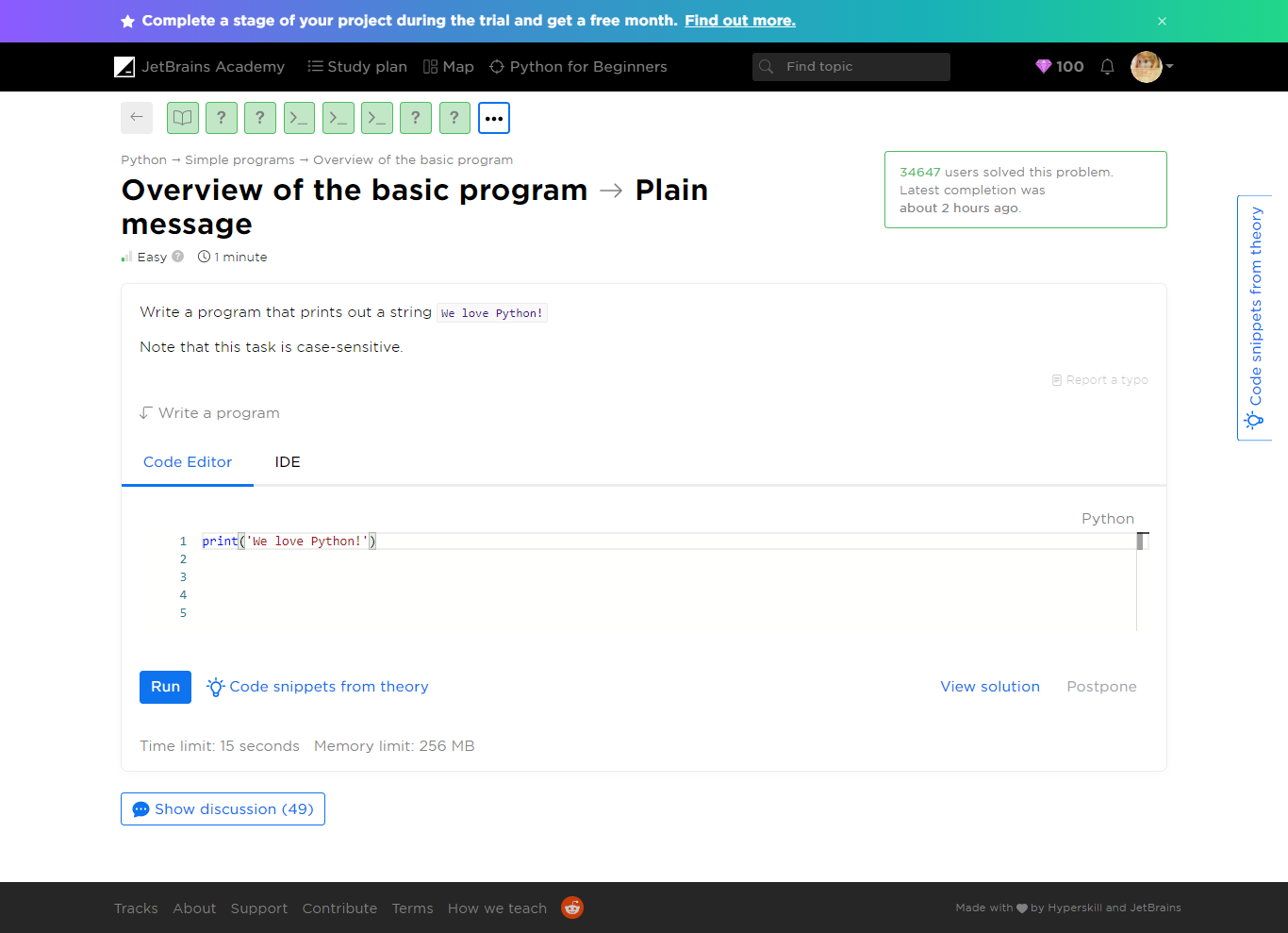


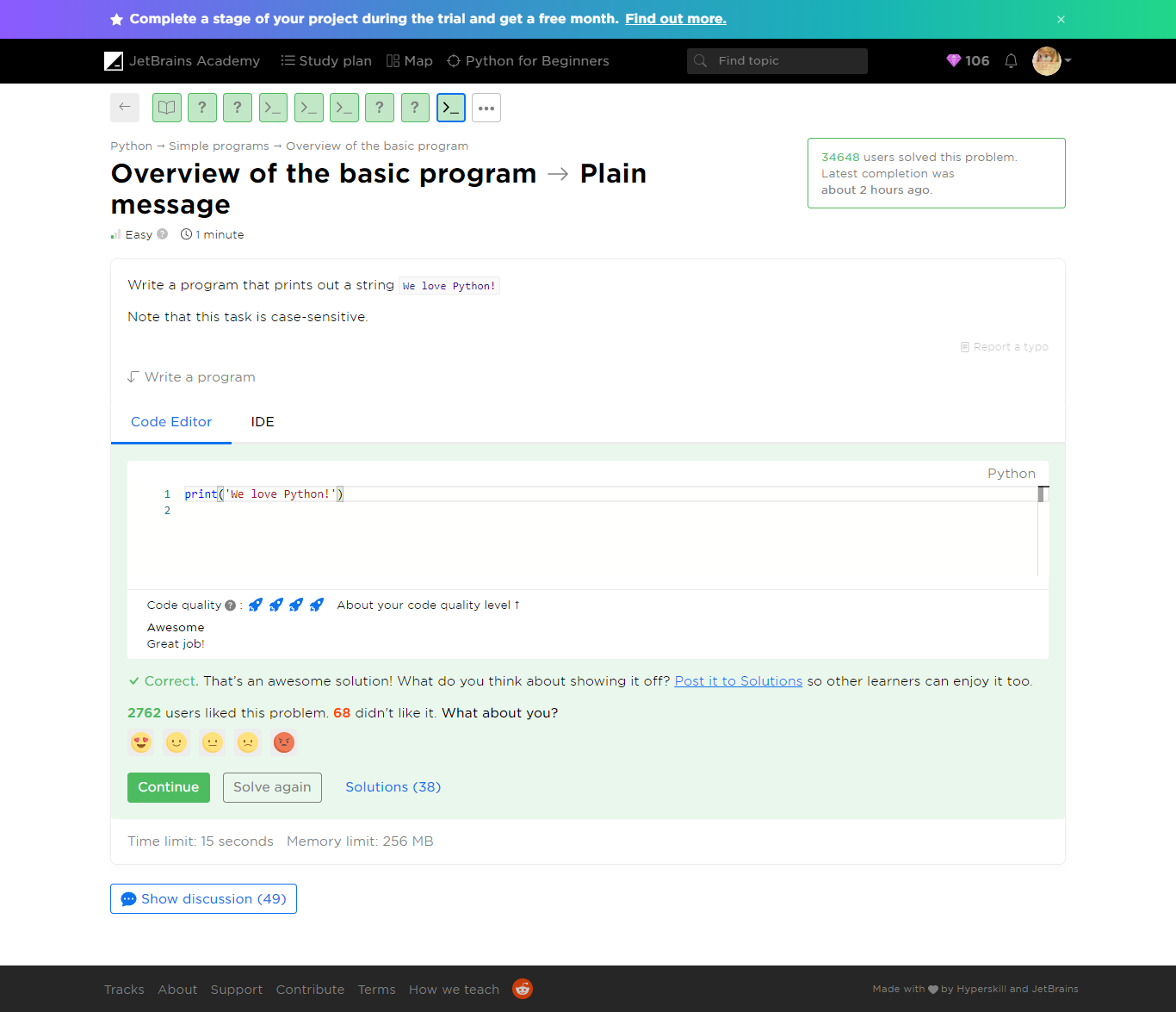
##### Overview of the basic program -> What is print



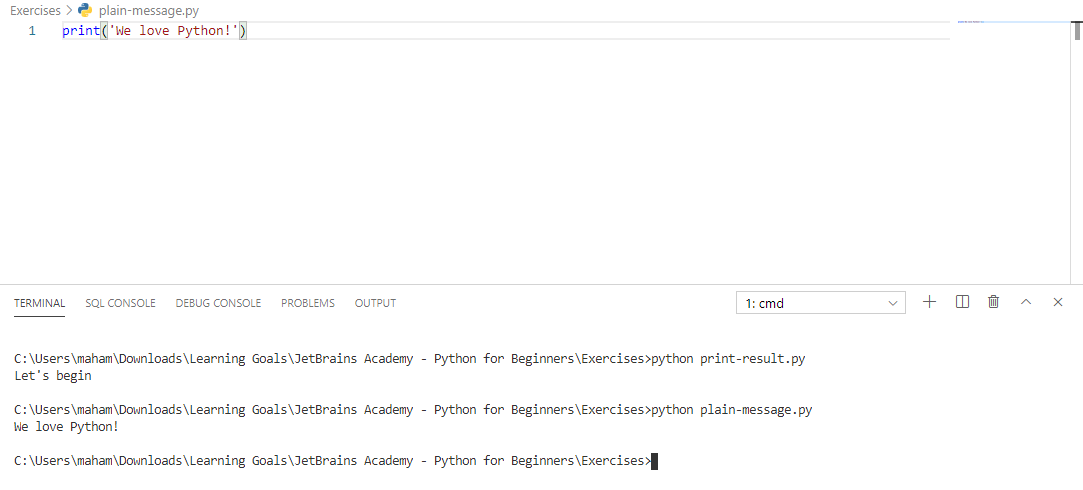


##### Overview of the basic program -> Plain message





Refer to ***plain-message.py***



#### Yaaay!

You completed **Overview of the basic program**

**You solved: 8 problems**

1. Print a text

2. Print result

3. What is print

4. Multiple quotes

5. Your name

1. The eternal truth

2. Alice in Wonderland

3. Plain message

#### Theory: Multi-line programs

 5 minutes 0 / 5 problems solved

We hope you have already learned how to write simple Python programs consisting of a single line that just prints a text. However, real programs contain a significant number of lines: from tens and hundreds for small scripts to thousands and even more for large projects. So, in this lesson, you will write programs that print multiple lines.

Let's consider an example, the following code prints exactly three strings, each on a new line:

print("I")  
print("know")  
print("Python")

The output is:

I  
know  
Python

You can run this example [here](https://repl.it/languages/python3) or locally if you have already installed Python on your computer.

There are other ways to print the same text using just one function call. We will consider them in the next topics.

The print function also allows you to print an empty line with no string specified:

print("I")  
print()  
print("know")  
print()  
print("Python")

Here's the output:

I  
  
know  
  
Python

However, skipping the line will have no effect:

print("And")  
  
print("you?")

The output is:

And  
you?

Now it's time to solve some problems.

**Current topic:**

[Multi-line programs Stage 1](https://hyperskill.org/learn/step/5233" \t "_self)

**Topic depends on**

[Overview of the basic program Stage 1](https://hyperskill.org/learn/step/5208" \t "_self)

**Topic is required for**

Program execution

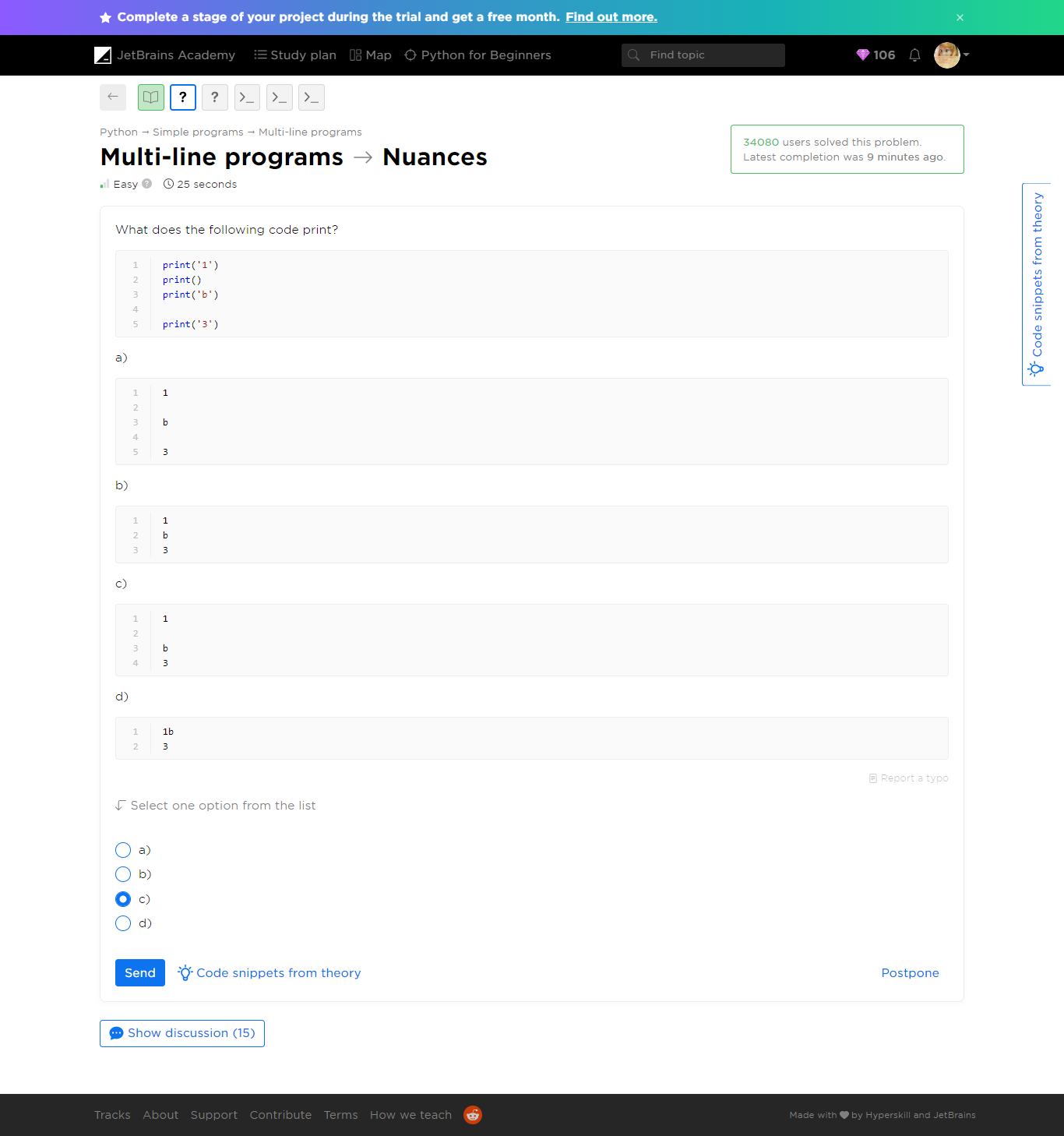
[Comments Stage 2](https://hyperskill.org/learn/step/6081" \t "_self)

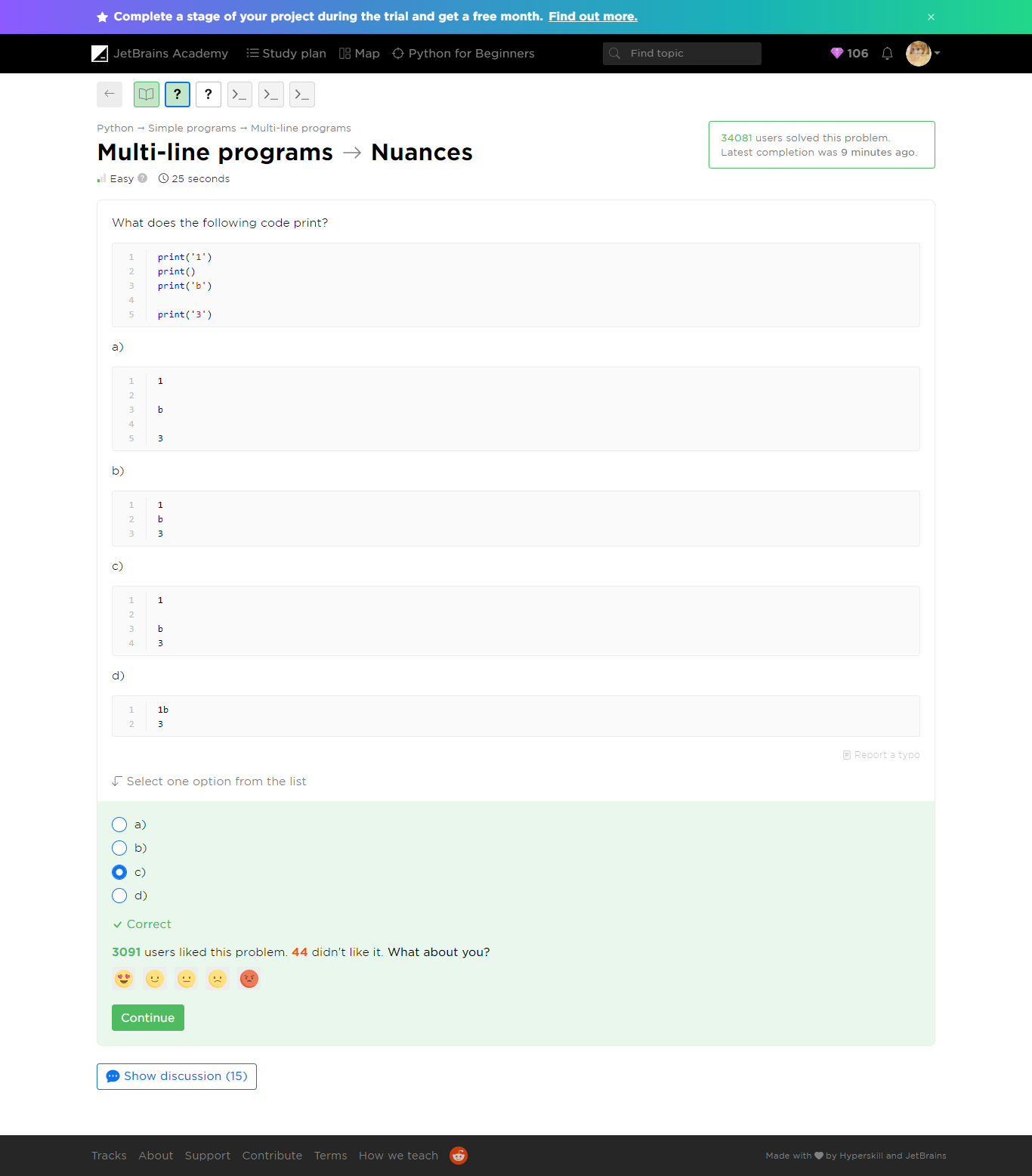
**Table of contents:**

[↑ Multi-line programs](https://hyperskill.org/learn/step/5233#step-title)

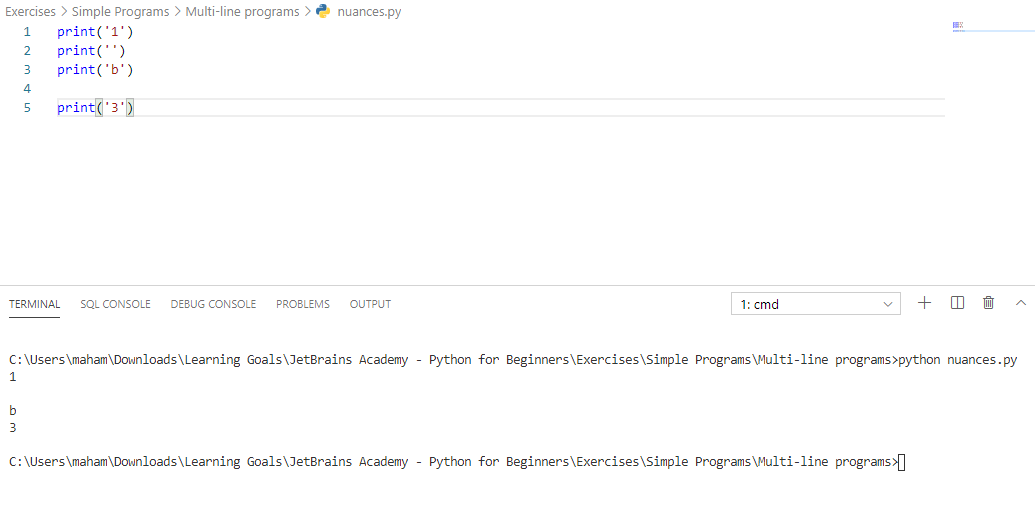
##### Multi-line programs -> Nuances

 Easy  25 seconds



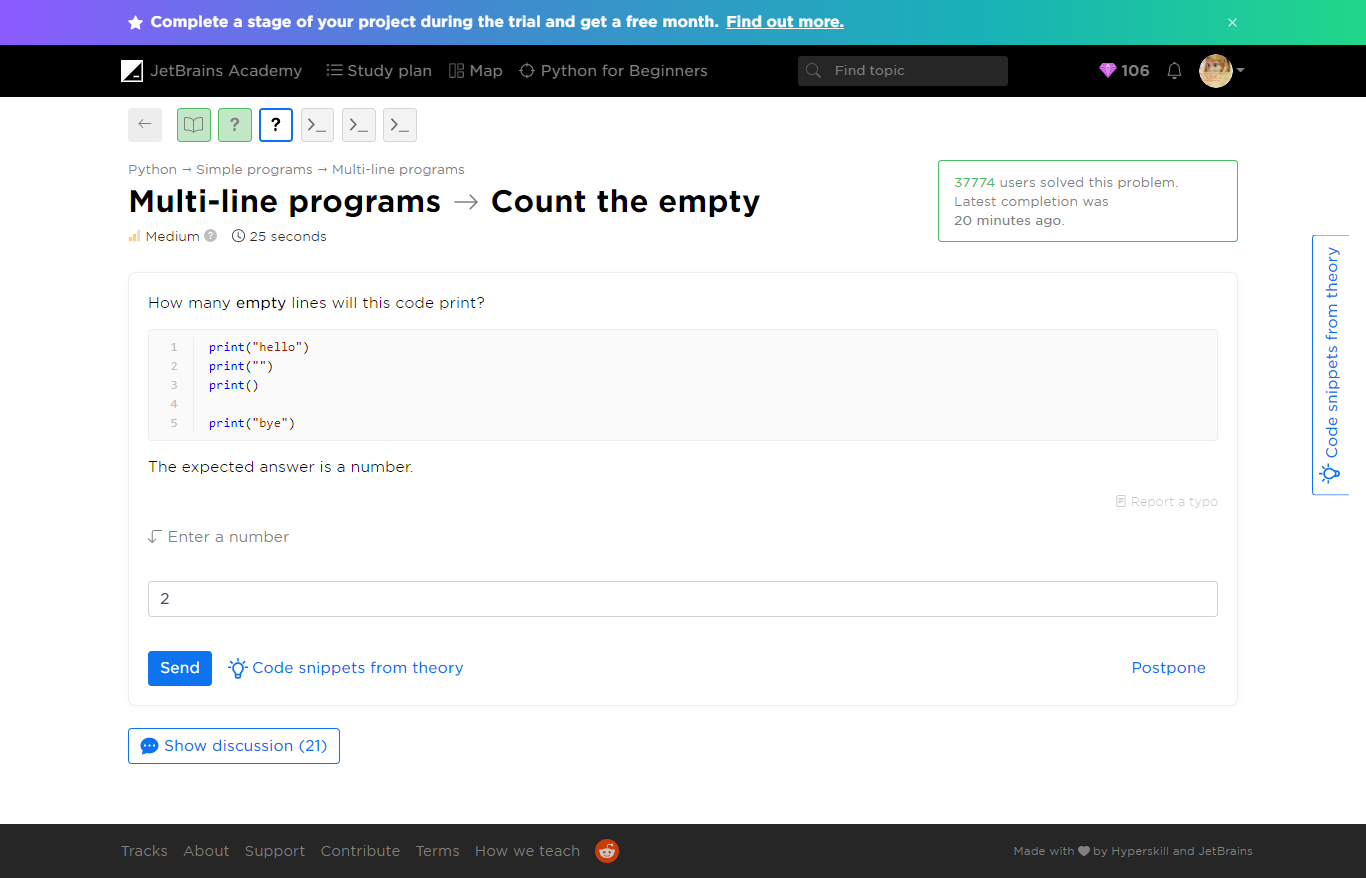


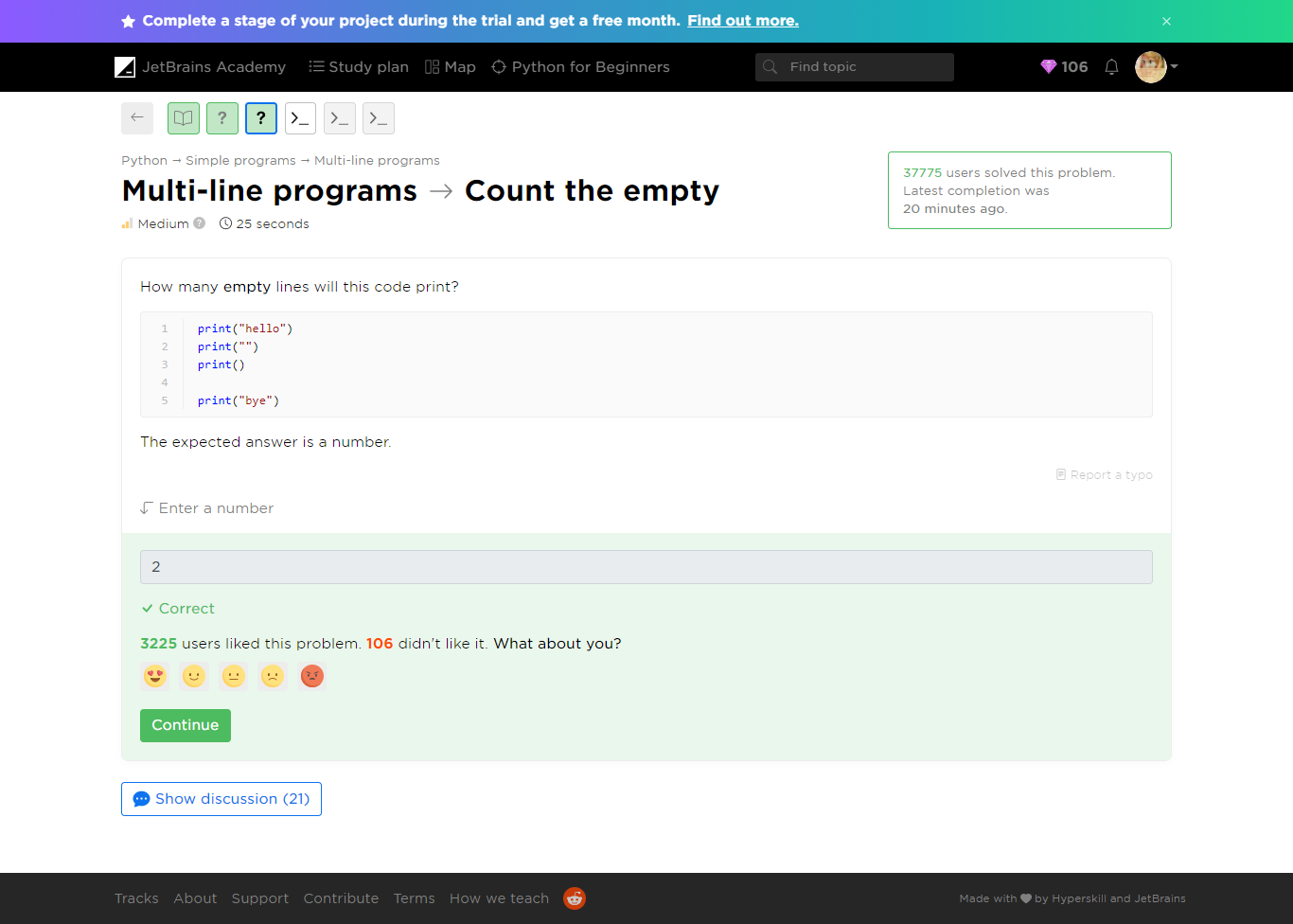
Refer to ***nuances.py***



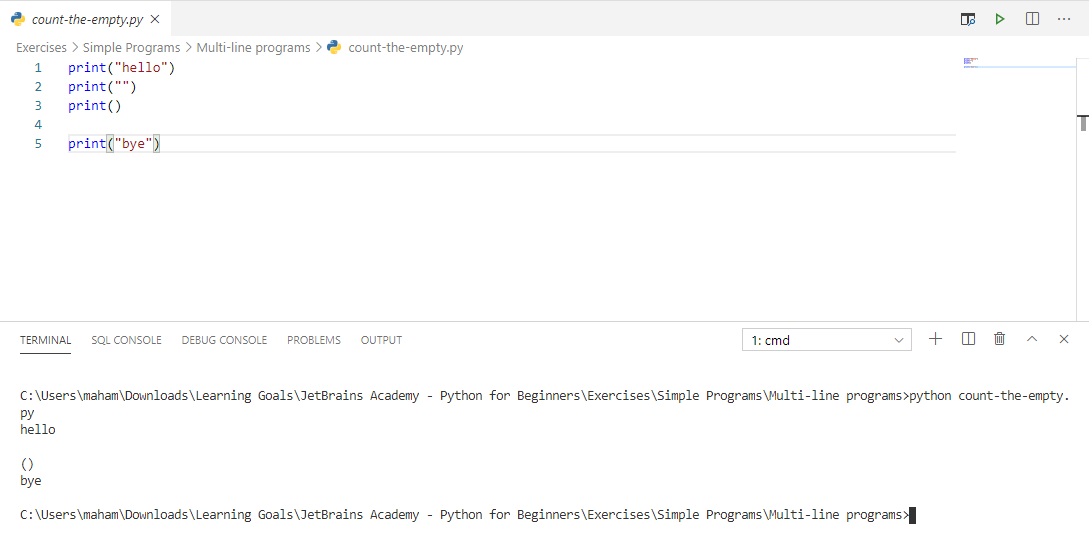
##### [Multi-line programs](https://hyperskill.org/learn/step/5233)-> Count the empty

 Medium  25 seconds



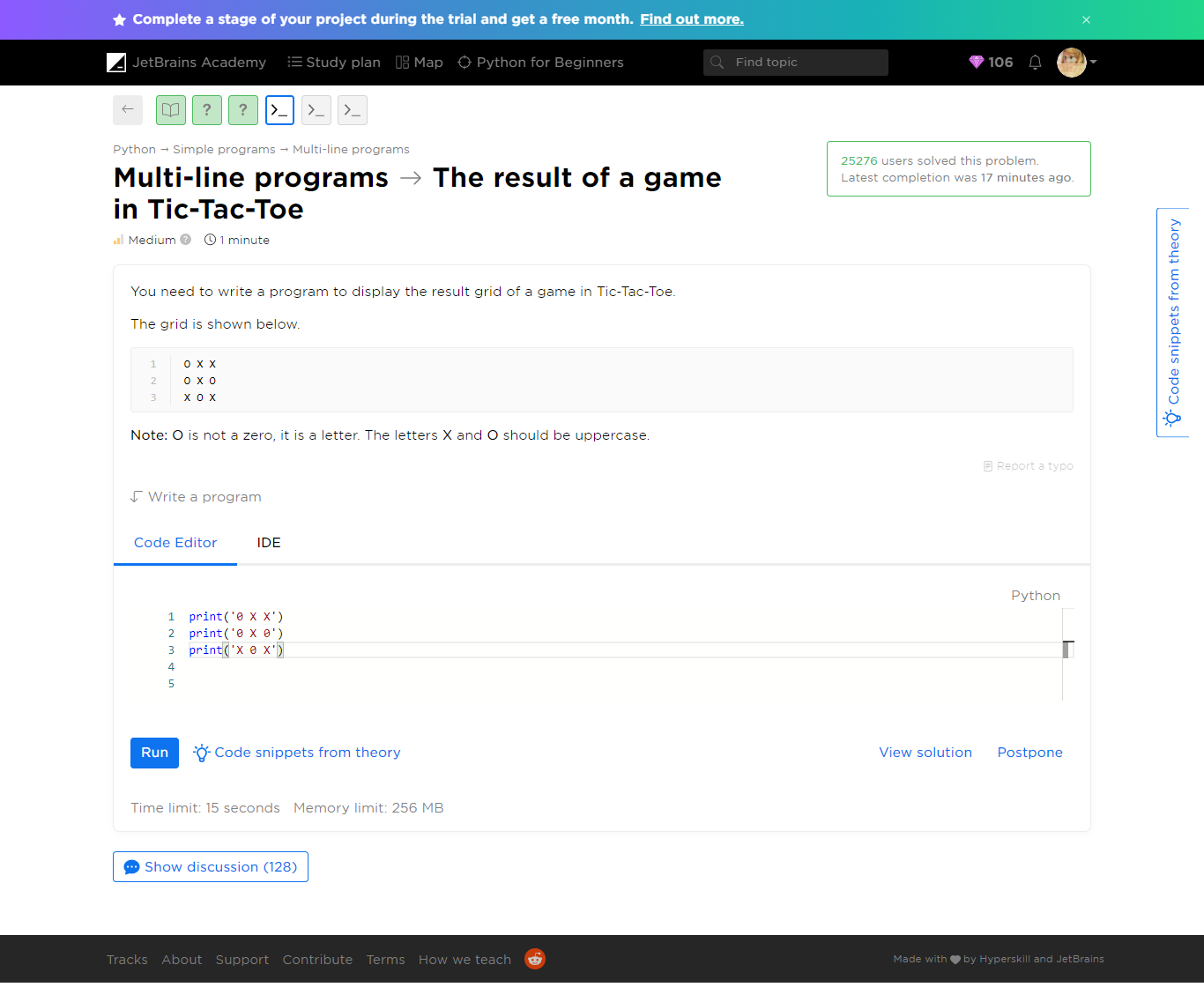


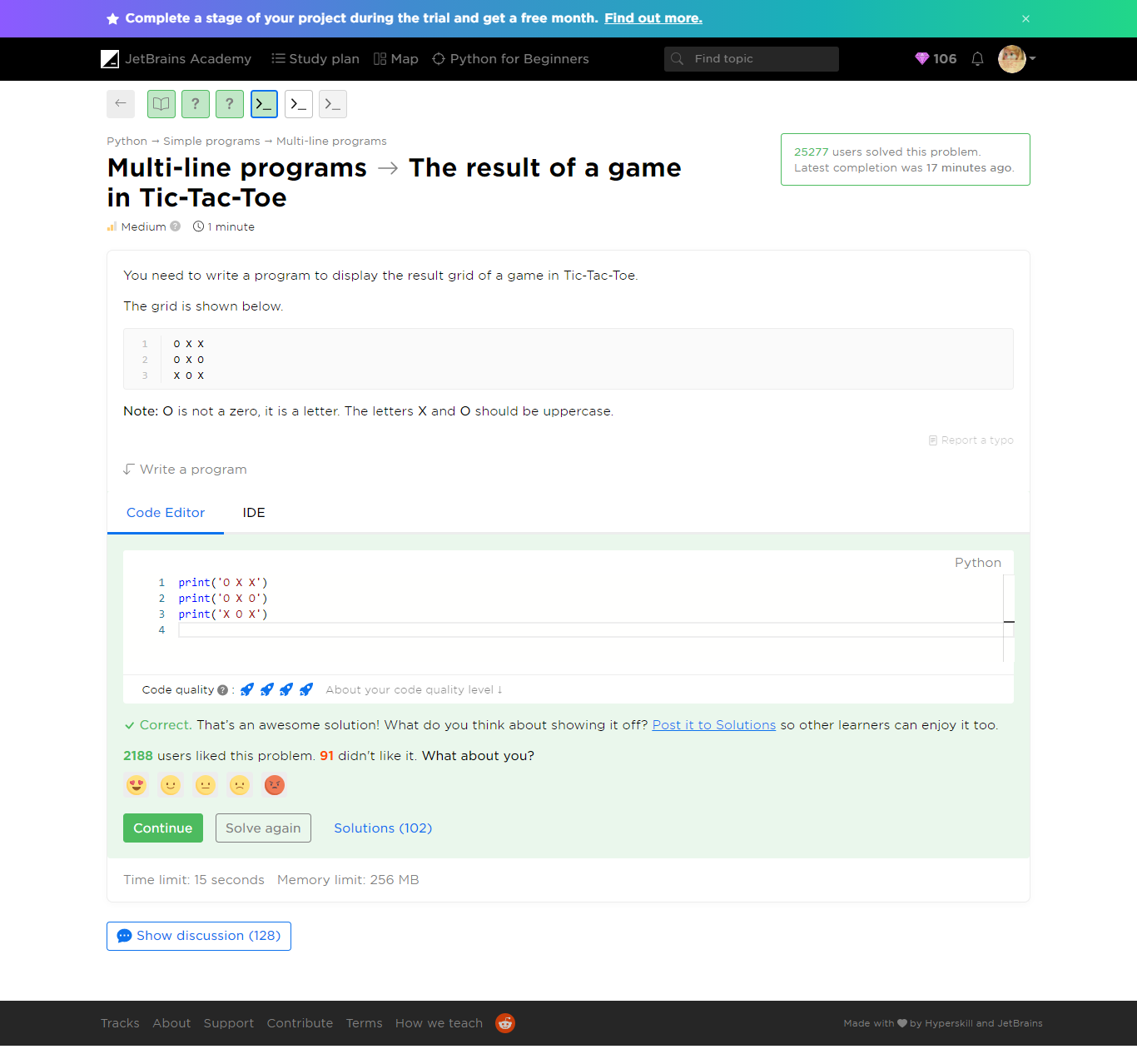
Refer to ***count-the-empty.py***



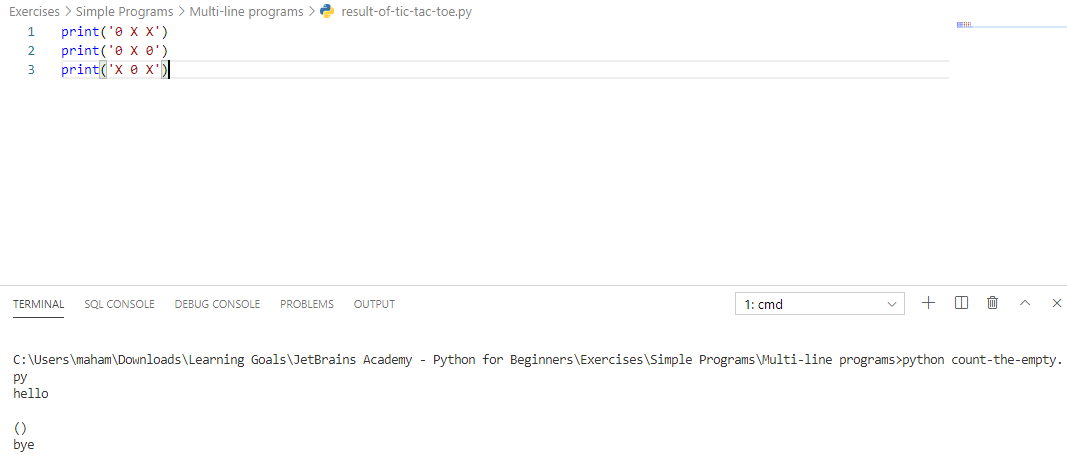
##### [Multi-line programs](https://hyperskill.org/learn/step/5233) -> The result of a game in Tic-Tac-Toe

 Medium  1 minute



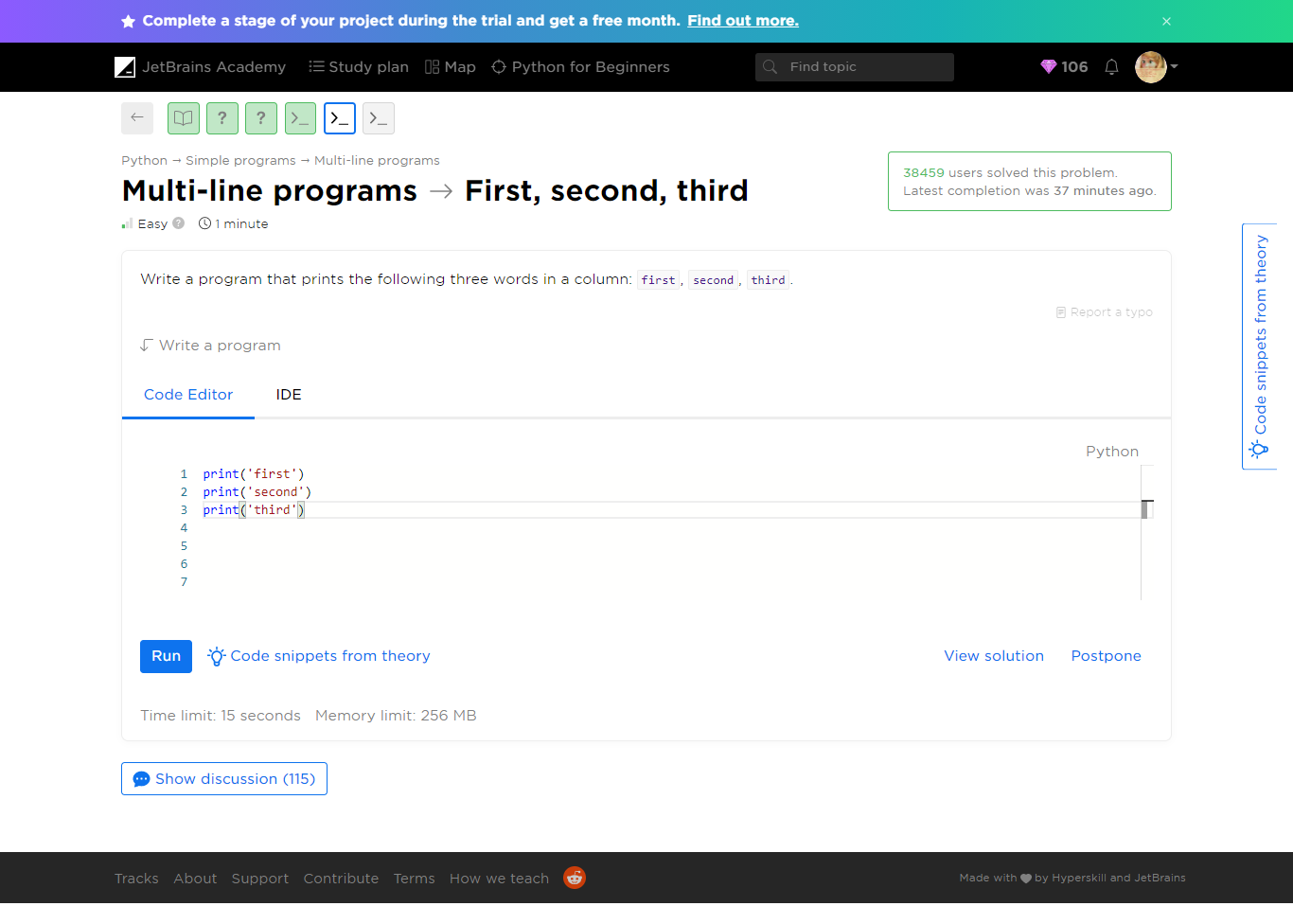


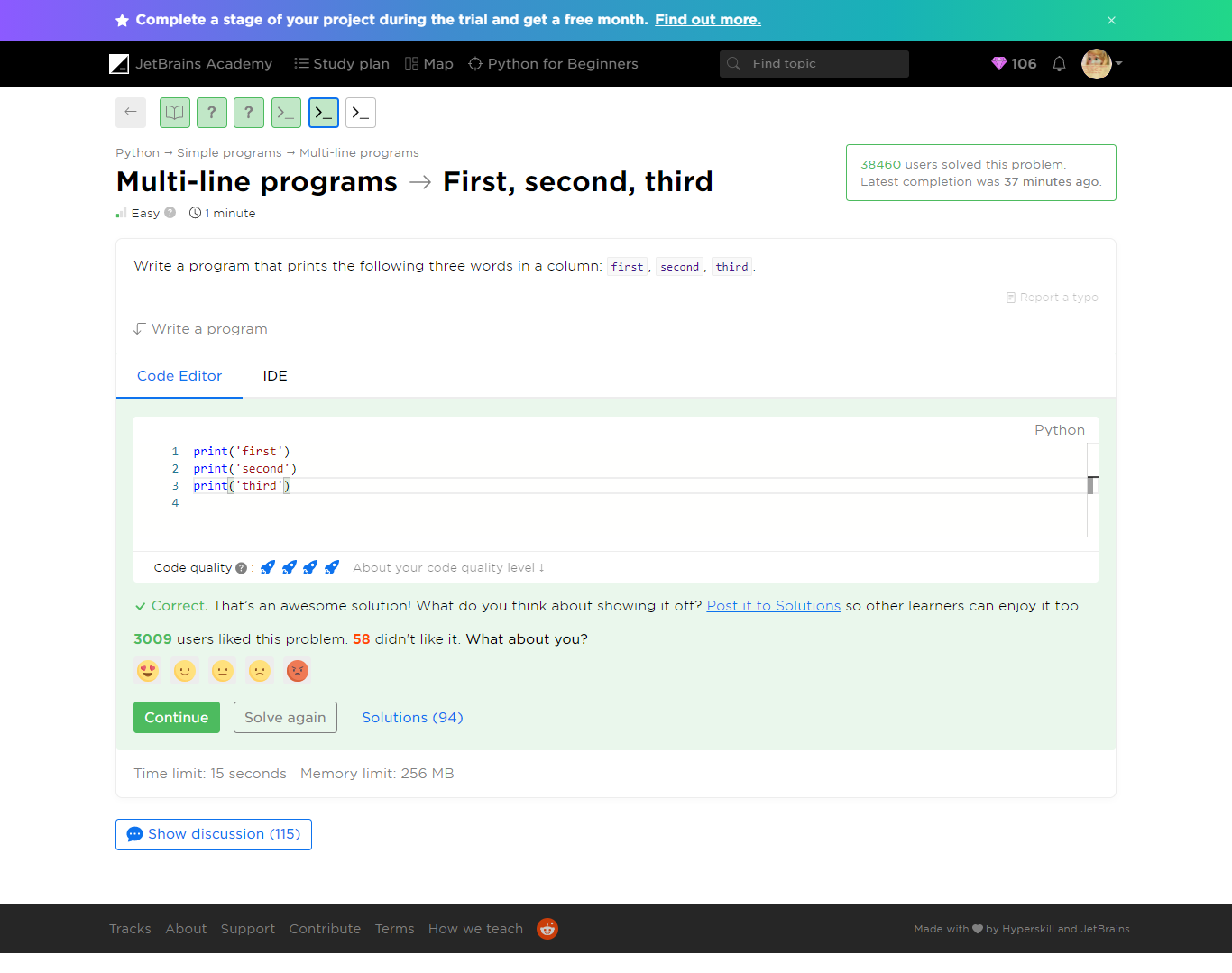
Refer to ***result-of-tic-tac-toe.py***



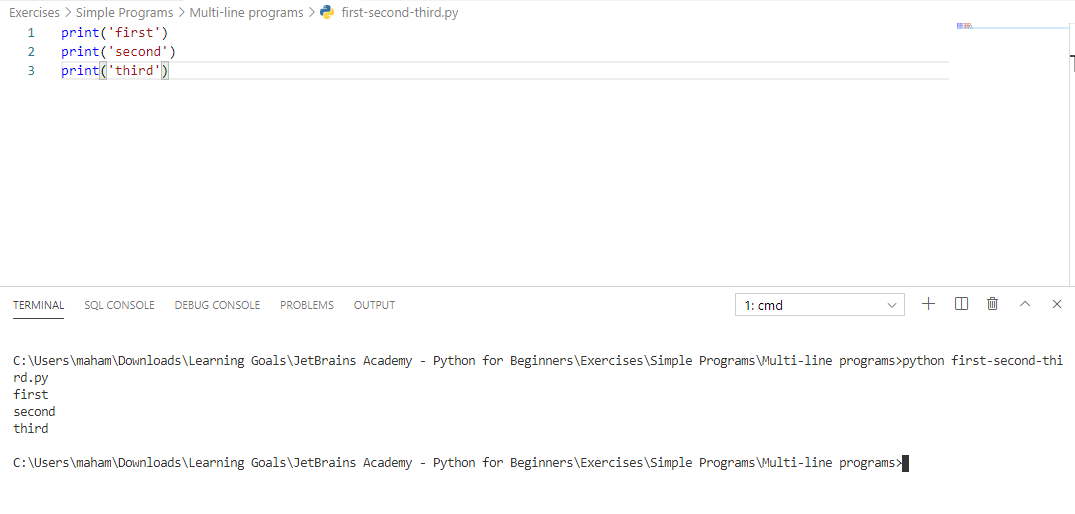
##### [Multi-line programs](https://hyperskill.org/learn/step/5233) -> First, second, third

 Easy  1 minute

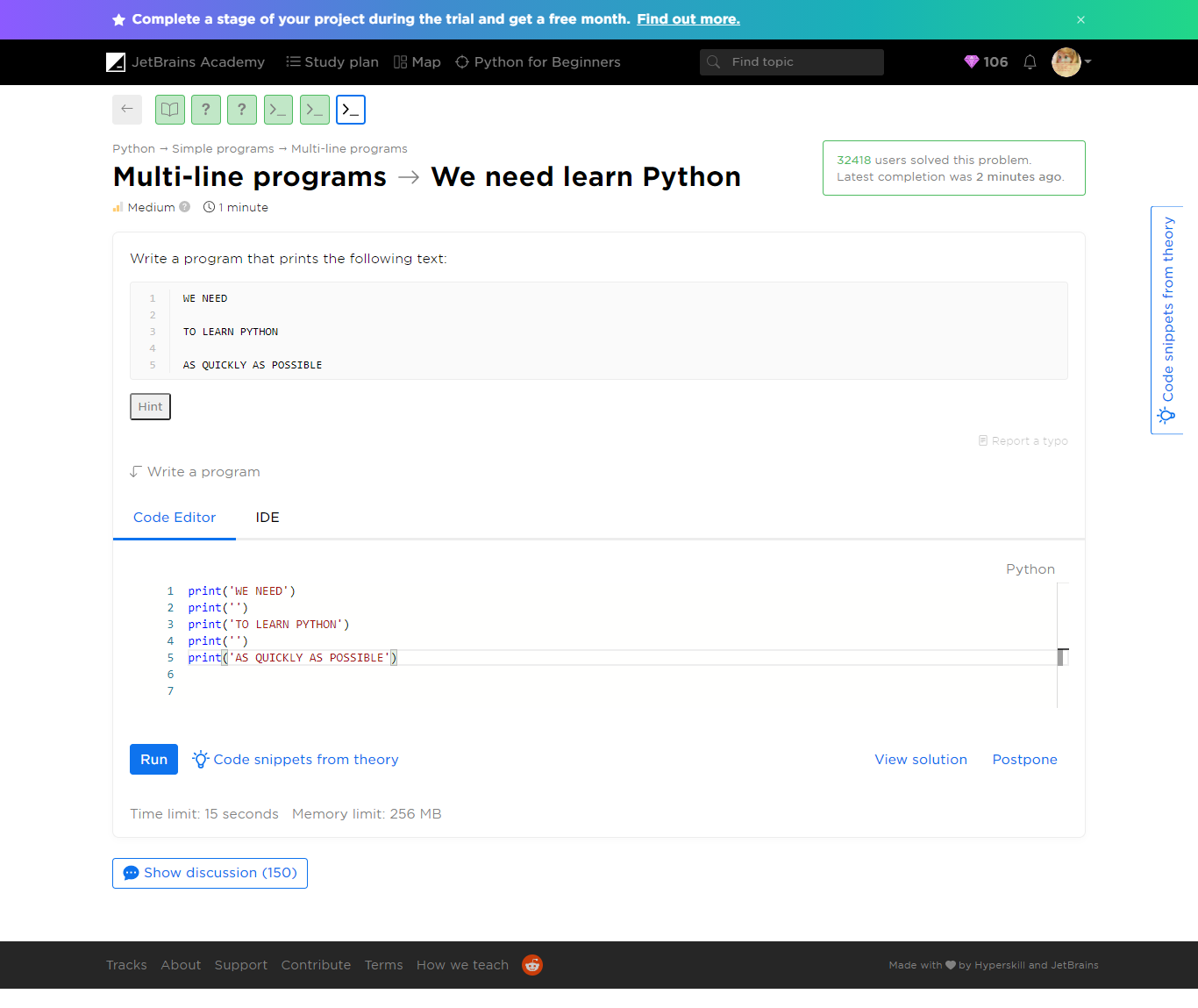


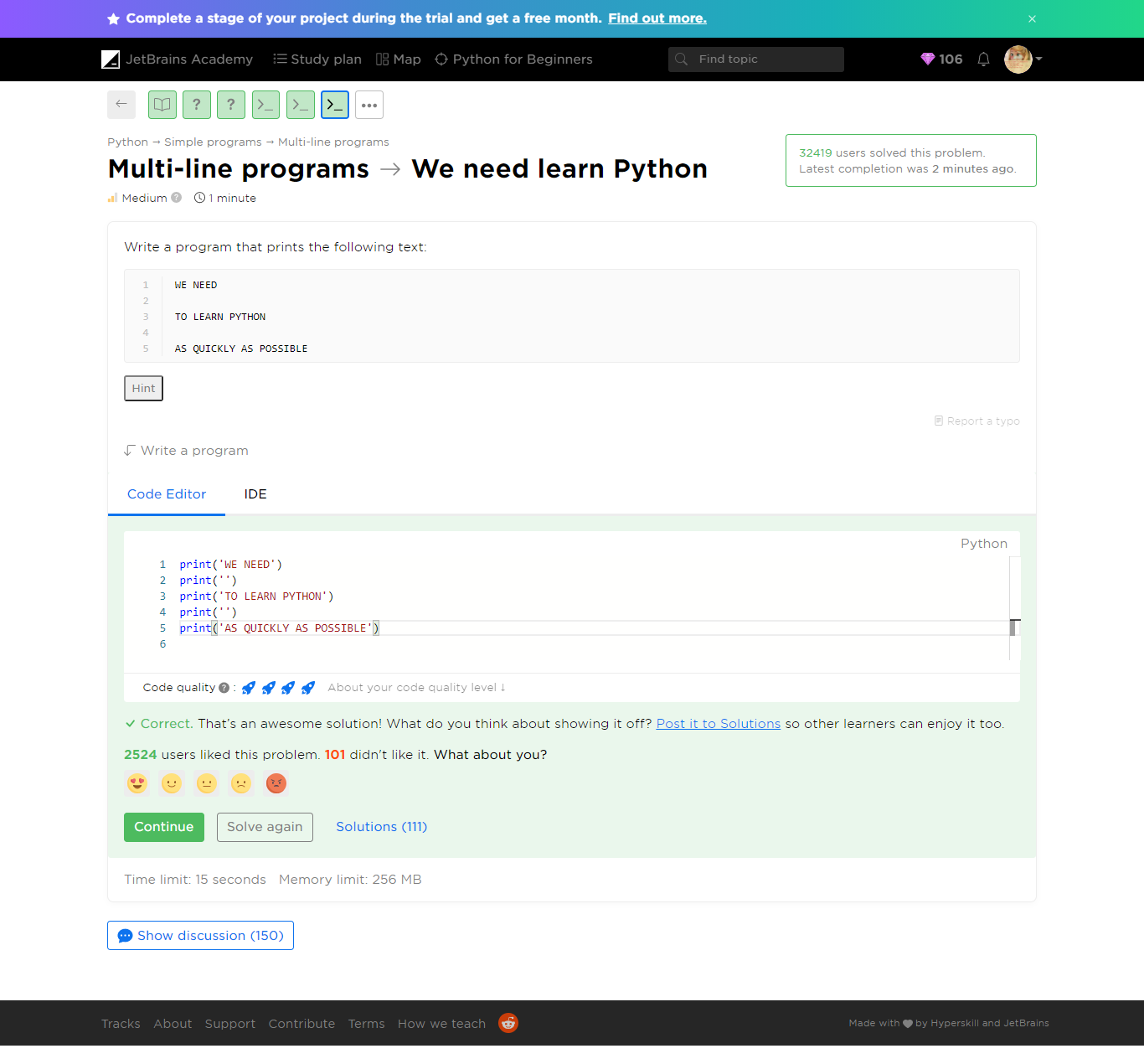


Refer to ***first-second-third.py***

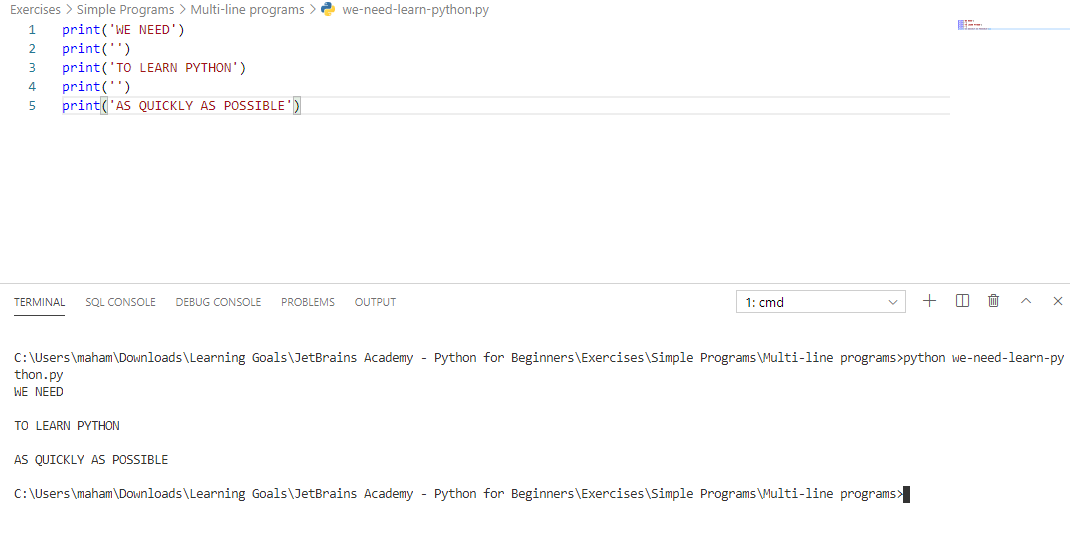


##### [Multi-line programs](https://hyperskill.org/learn/step/5233) -> We need learn Python





Refer to we-need-learn-python.py



#### Yaaay!

You completed **Multi-line programs**

**You solved: 5 problems**

1. Count the empty

2. Nuances

3. We need learn Python

1. First, second, third

2. The result of a game in Tic-Tac-Toe

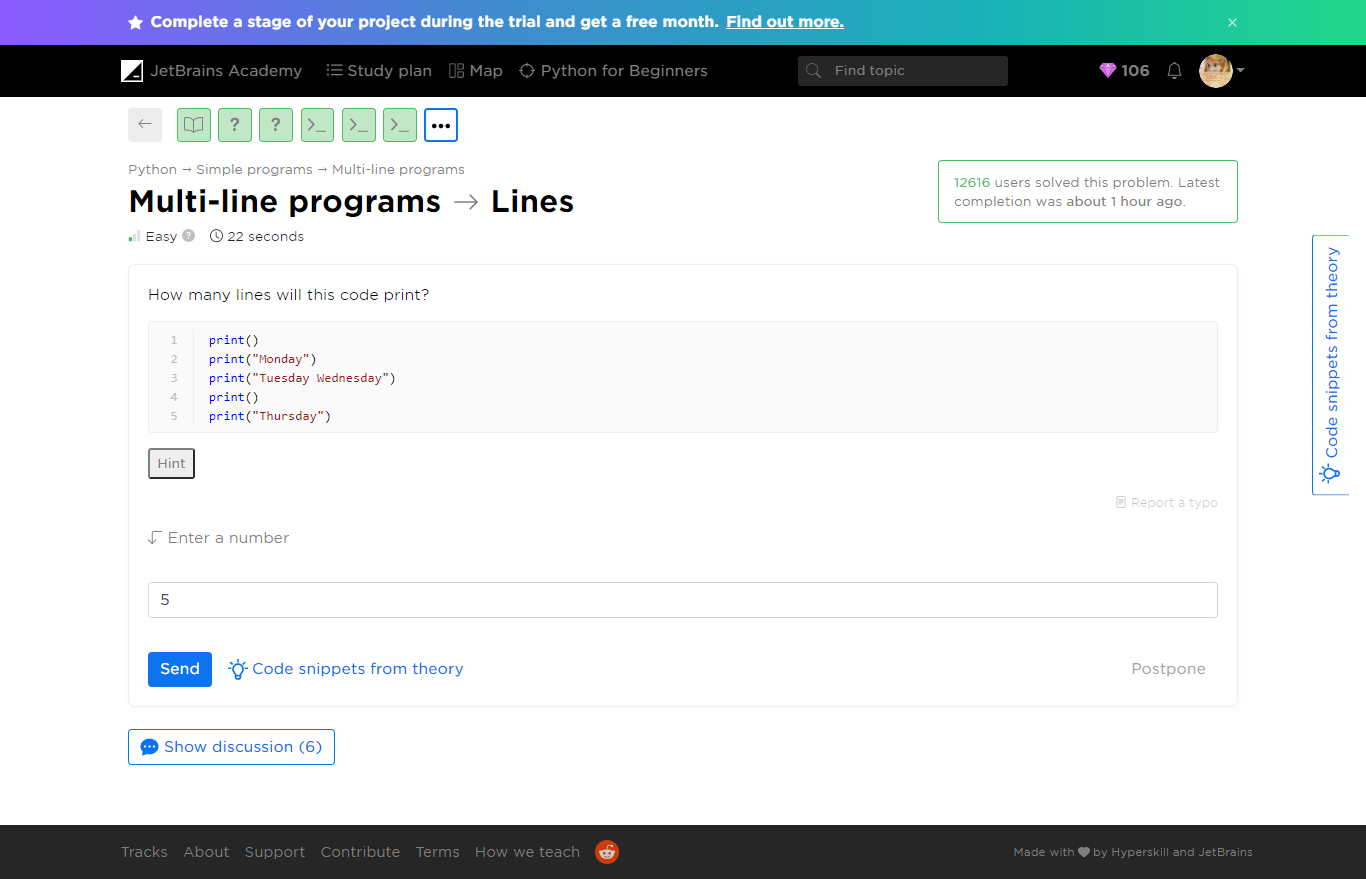
**More to solve (optional): 2**

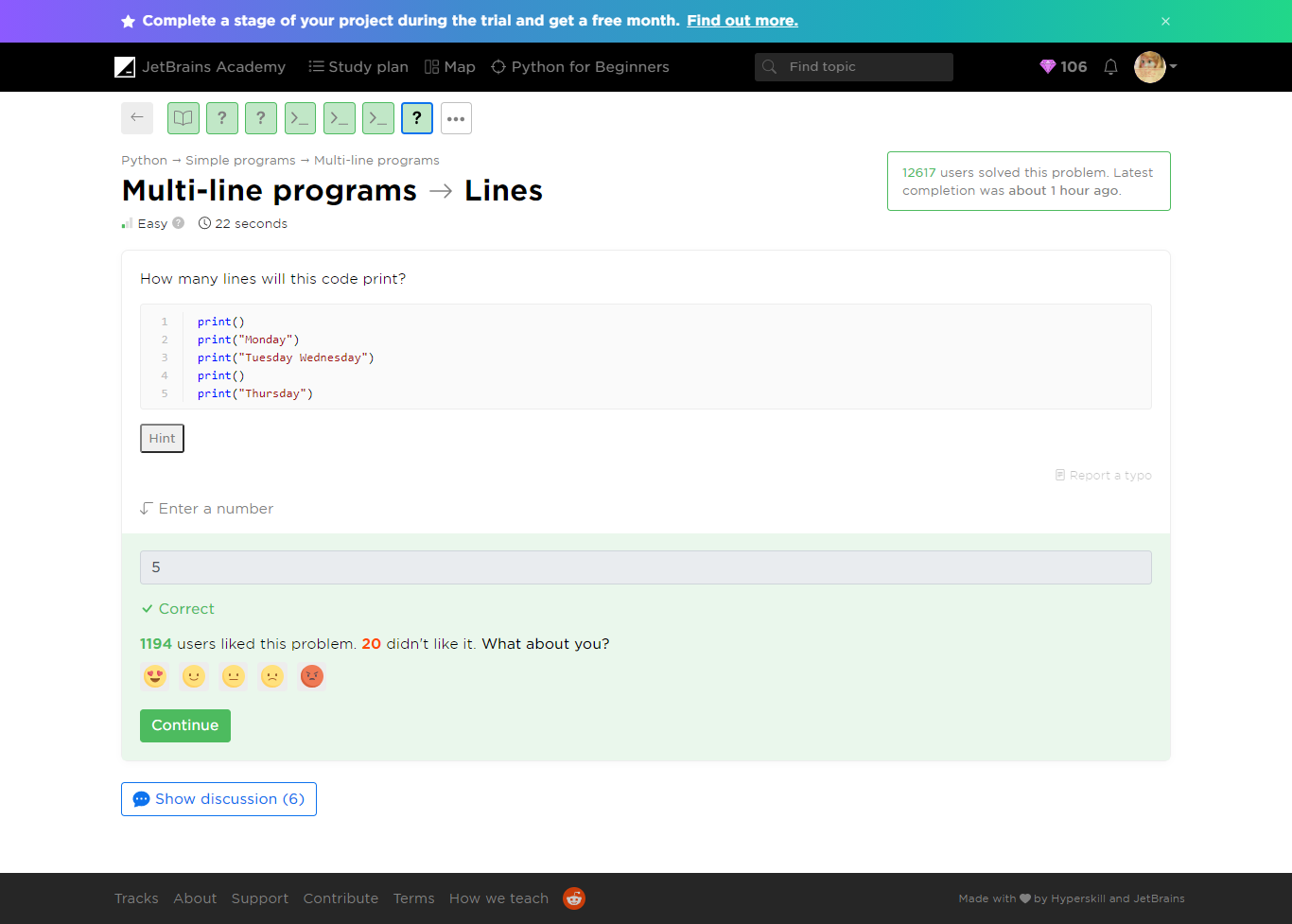
1. Lines

2. Square

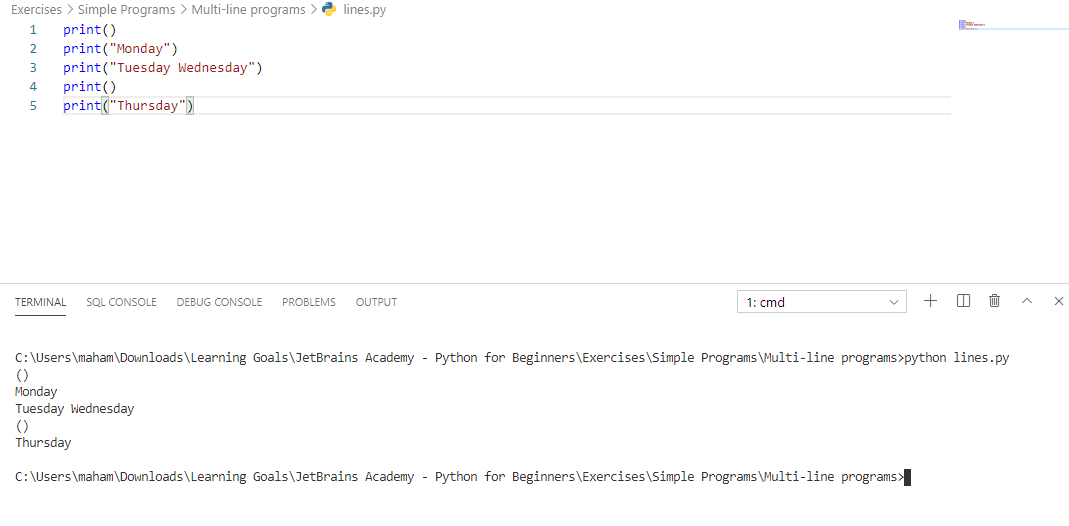
#### Additional Problems

##### Multi-line programs -> Lines





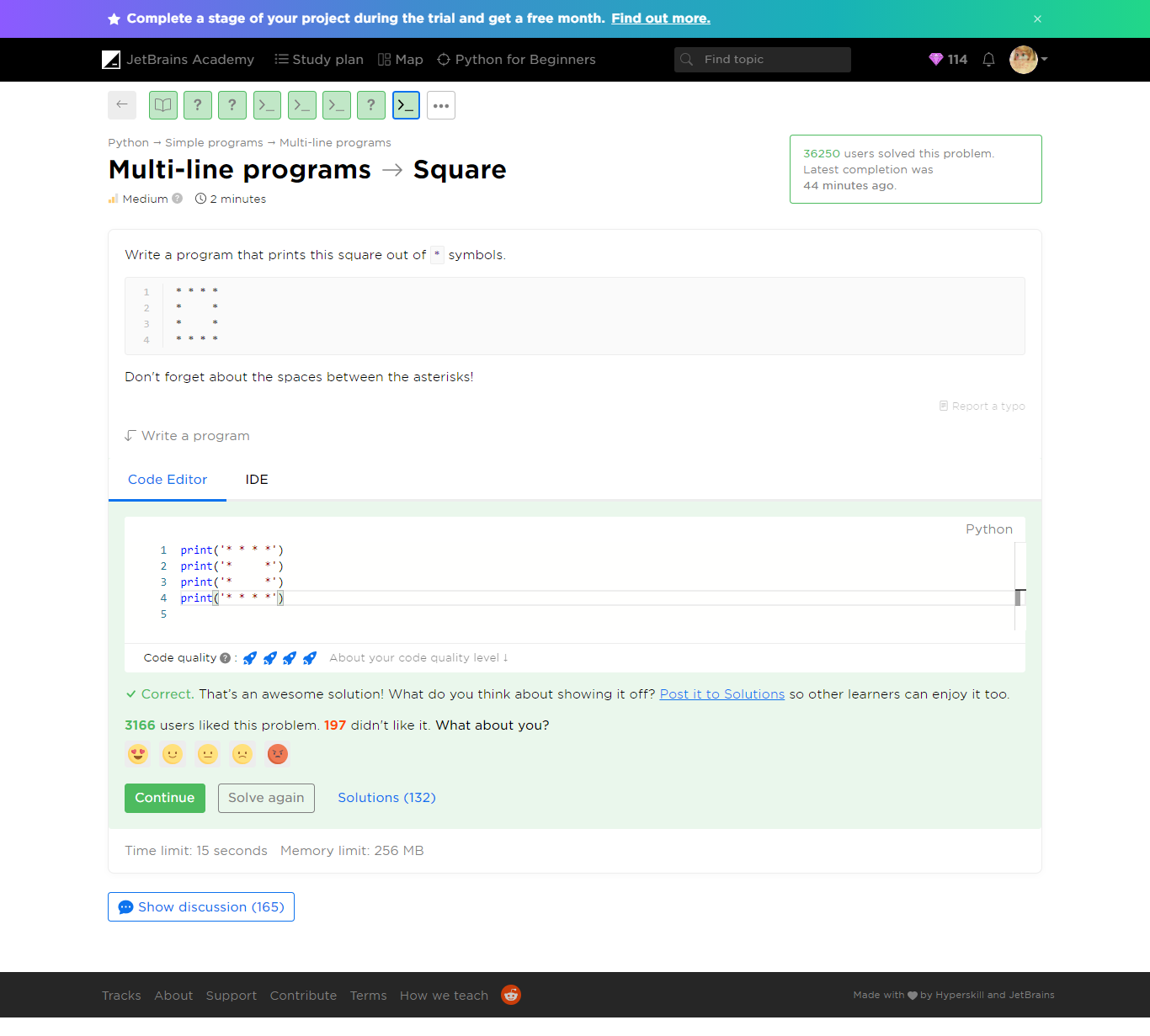
Refer to lines.py



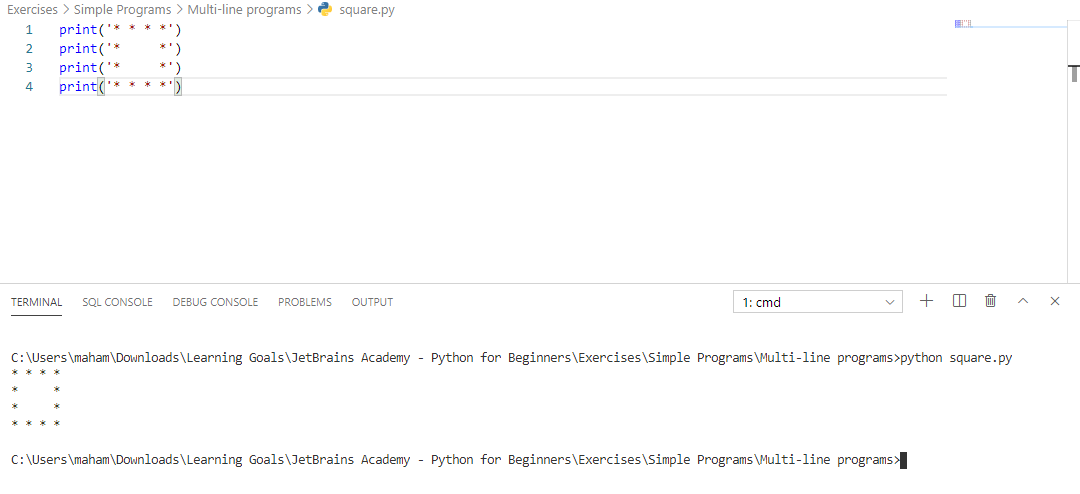
##### [Multi-line programs](https://hyperskill.org/learn/step/5233) ->Square

 Medium  2 minutes





Refer to square.py



#### Yaaay!

You completed **Multi-line programs**

**You solved: 7 problems**

1. Count the empty

2. Lines

3. Nuances

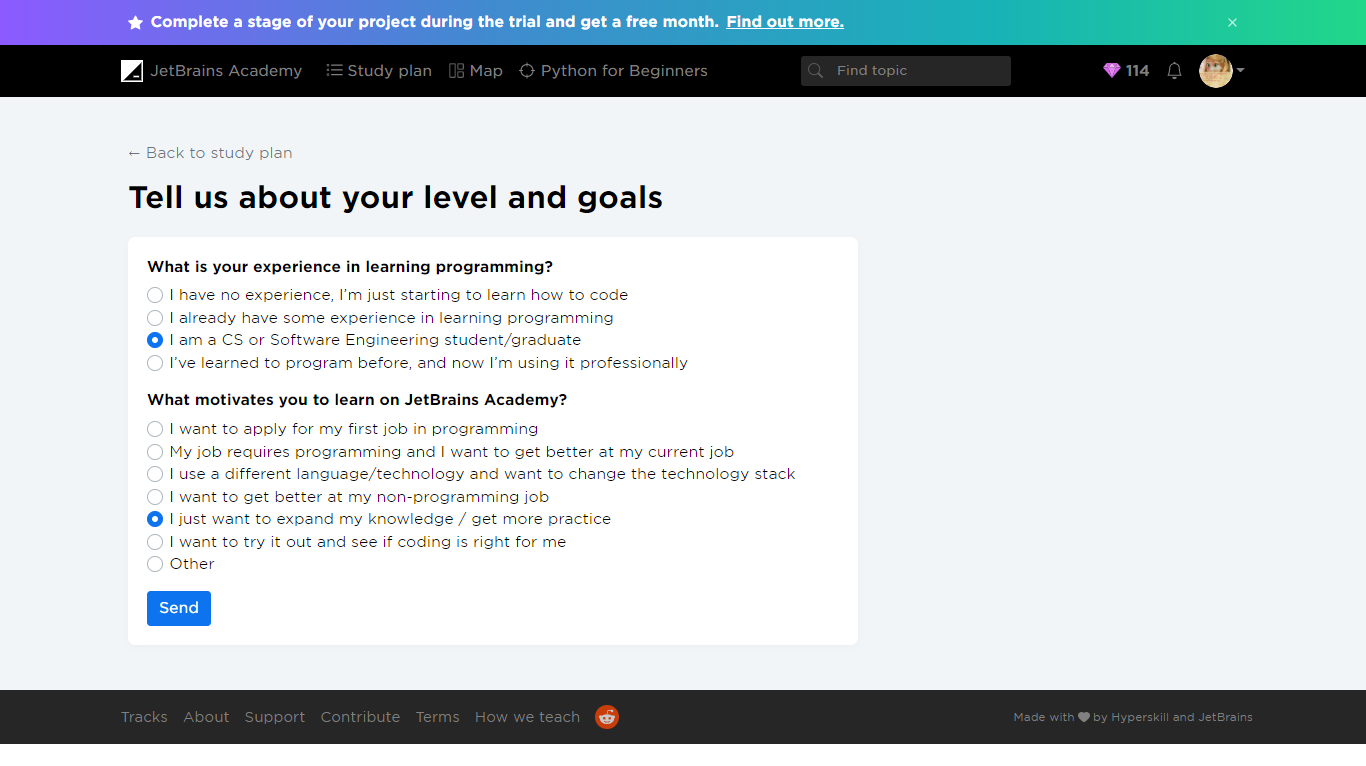
4. We need learn Python

1. First, second, third

2. The result of a game in Tic-Tac-Toe

3. Square

#### Tell us about your level and goals

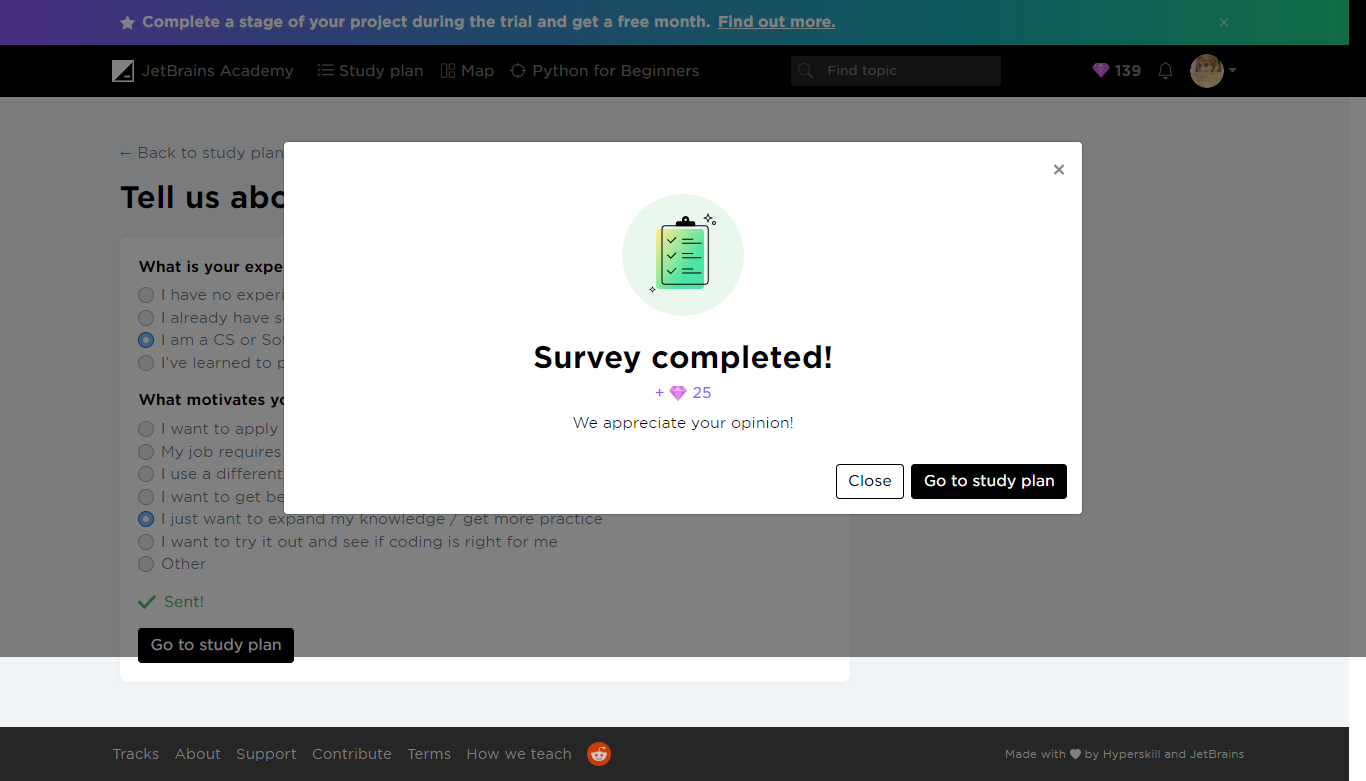




##### Survey completed!

**+  25 gems**

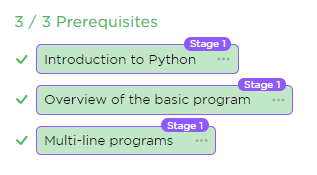
We appreciate your opinion!



#### Work on project. Stage 1/4: Rush into print.

Project: [Zookeeper](https://hyperskill.org/projects/98)

 Medium  2 minutes



#### Description

There are many animals in the zoo, and all of them need care. The animals must be fed, cleaned, surrounded by their kin, and kept happy. That is a difficult task for our large zoo, so one of your employers has suggested a more convenient way to keep track of everything. She wants to be able to pull up a video feed of any animal in the zoo with the help of a program. Being able to check on each habitat would help the zookeepers take care of our furry friends more efficiently!

In this project, you will create a program that helps the zookeepers check on the animals and make sure that they're doing well. Your product will be able to process commands from the zookeepers and display the animals on a monitor.

#### Objectives

To begin with, you will develop a simple printer. Your program should display the text from the output example.

#### Example

Output:

I love animals!  
Let's check on the animals...  
The deer looks fine.  
The bat looks happy.  
The lion looks healthy.