

# CodeRunner question type

**Question 1**  
Correct  
Marked out of 1.00

Write a function `double(n)` that returns twice its parameter `n`.

**For example:**

Test	Result
<code>print(double(3))</code>	6

**Answer:** (penalty regime: 10, 20, ... %)

```
1 def double(n):  
2     return n * 2
```

Check

	Test	Expected	Got	
✓	<code>print(double(3))</code>	6	6	✓
✓	<code>print(double(-4))</code>	-8	-8	✓
✓	<code>print(double(0))</code>	0	0	✓
✓	<code>print(double(10.5))</code>	21.0	21.0	✓

Passed all tests! ✓

Start again Save Fill in correct responses Submit and finish Close preview

An example of CodeRunner question type

## Overview:

This guide will show you how to create a CodeRunner question type.

## What to know:

CodeRunner allows students to submit code as answers to various programming questions in different languages. Typically, it works in Core's adaptive quiz mode, where students enter their code for each question and receive immediate feedback on their test results. They can then make corrections and resubmit their code, usually with a minor penalty.

## Table of Contents

[Step 1: Access the Quiz activity where you want to create the question](#)

[Step 2: Open the 'Questions' tab.](#)

[Step 3: Add a new question](#)

[Step 4: Choose a question type](#)

[Step 5: Choose a 'Question type'](#)

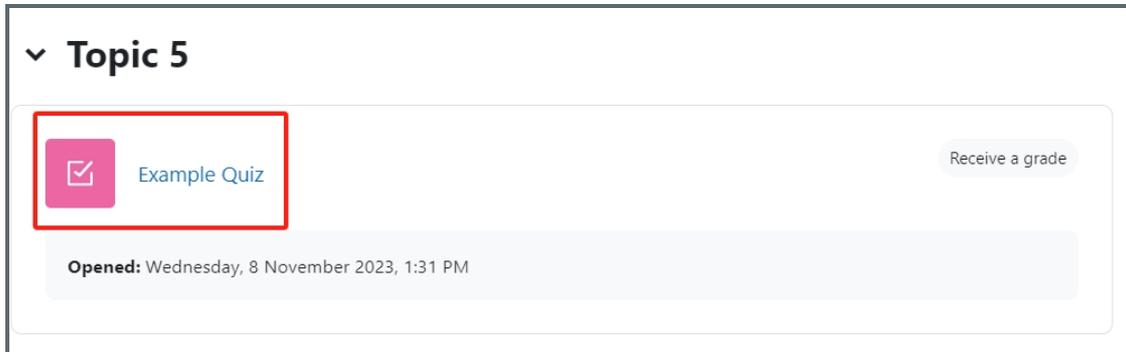
[Step 6: Input 'Question name' and 'Question text'](#)

[Step 7: Input the 'Test case'](#)

[Step 8: Save changes](#)

## Step 1: Access the Quiz activity where you want to create the question

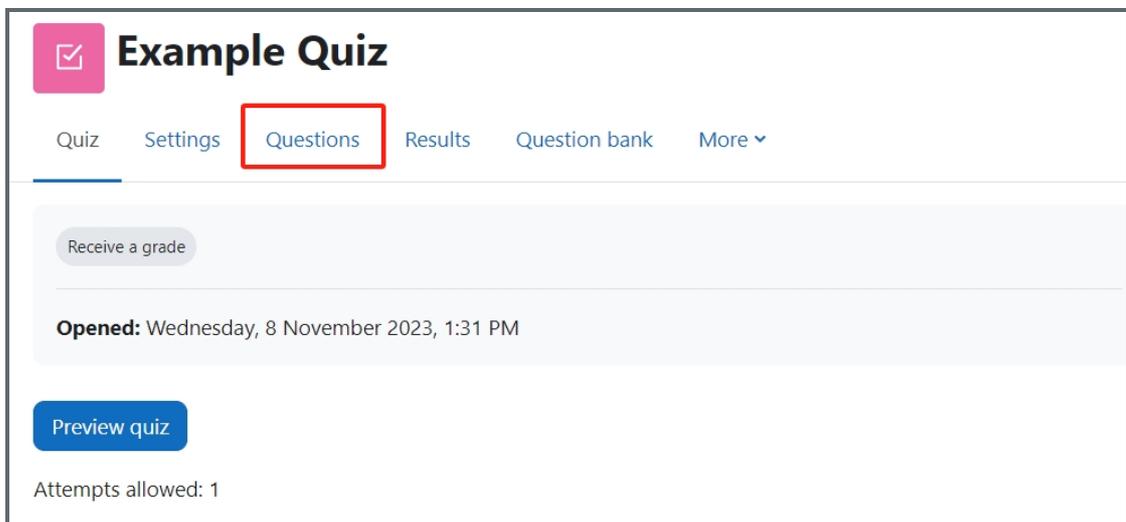
In your module area/course, click the link of the Quiz activity where you want to create a new CodeRunner question.



Note: If an appropriate Quiz activity doesn't exist in your module area/course, please see [Add a quiz](#) for details of how to create one.

## Step 2: Open the 'Questions' tab.

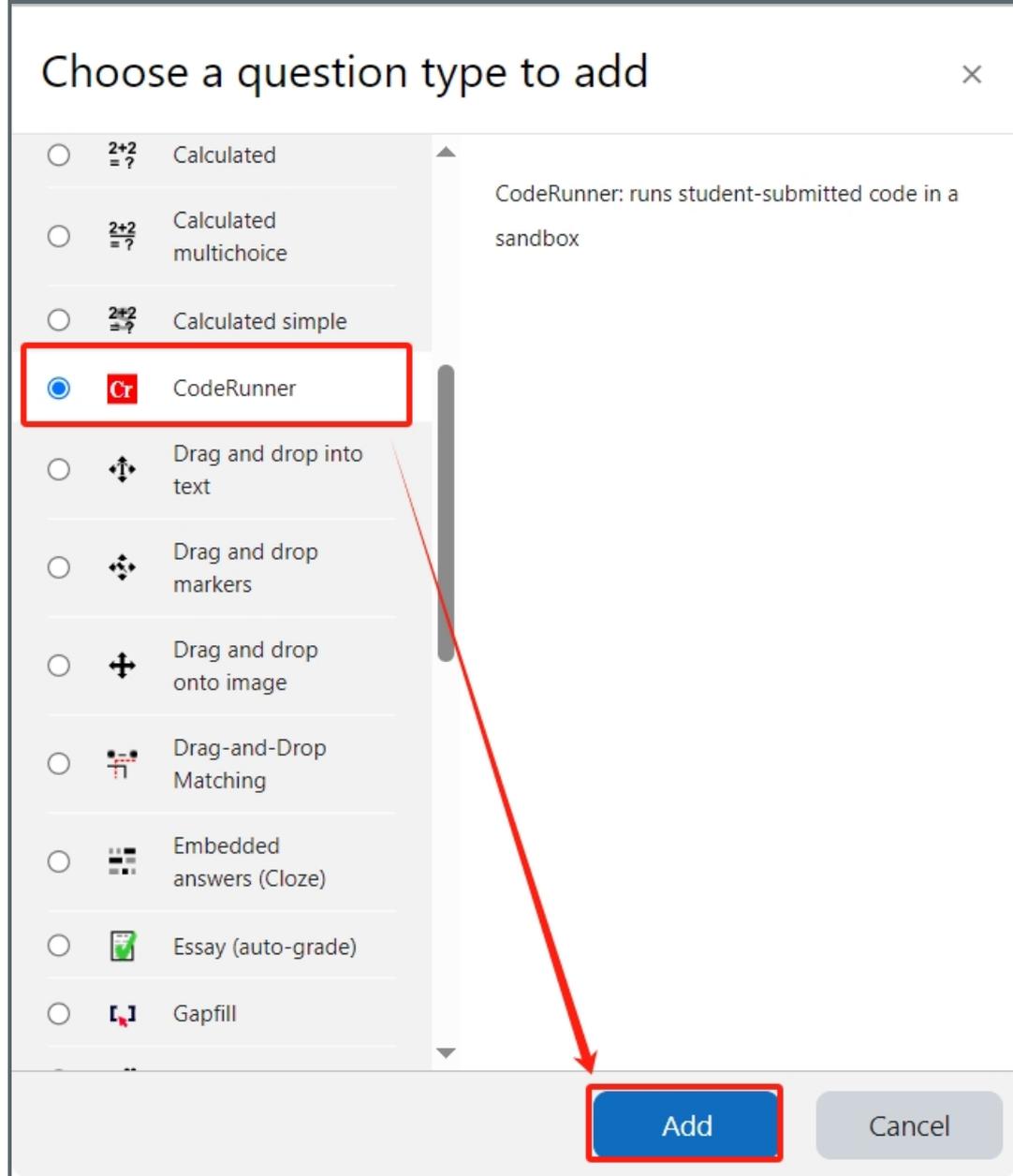
1. Open the 'Questions' tab.



## Step 3: Add a new question

1. On the following page, click the 'Add' link.
2. Click the 'a new question' link in the dropdown menu.





### Step 5: Choose a 'Question type'

Choose a 'Question type' according to your requirement.

## CodeRunner question type

Question type	?	python3
Customisation	?	Undefined
Answer box	?	c_function
Submit buttons	?	c_program
Stop button	?	cpp_function
Feedback	?	cpp_program
Marking	?	directed_graph
Template params	?	java_class
Template param controls	?	java_method
UI parameters	?	java_program
		multilanguage
		nodejs
		octave_function
		pascal_function
		pascal_program
		php
		python2
		<b>python3</b>
		python3_w_input

Then, you can find more details about the question type you choose in the 'Question type details' area.

**Question type details**

CodeRunner question type: python3

A Python3 question type, which can handle write-a-function, write-a-class or write-a-program question types. For each test case, the student-answer code is executed followed by the test code. Thus, for example, if the student is asked to write a function definition, their definition will be executed first, followed by the author-supplied test code, which will typically call the function and print the result or some value derived from it.

If there are no standard inputs defined for all test cases, the question actually wraps all the tests into a single run, printing a separator string between each test case output. Please be aware that this isn't necessarily the same as running each test case separately. For example, if there are any global variables defined by the student code, these will hold their values across the multiple runs. If this is likely to prove a problem, the easiest work-around is to define one of the test case standard input fields to be a non-empty value - this forces CodeRunner into a fallback mode of running each test case separately.

Step 6: Input 'Question name' and 'Question text'

**General**

Category: Default for Chenhui's Test Module (13)

Question name: Double function in python

Question text: Write a function double(n) that returns twice its parameter n.

## Step 7: Input the 'Test case'

Input Test cases.

You can set the Test case as an example to provide students with a reference.

▼ **Test cases**

Test case 1	<input type="text" value="print(double(3))"/>
Standard Input	<input type="text"/>
Expected output	<input type="text" value="6"/>
Extra template data	<input type="text"/>
Test properties:	<input checked="" type="checkbox"/> Use as example   Display: Show   <input type="checkbox"/> Hide rest if fail   Mark: 1.000   Ordering: 10

Test case 2	<input type="text" value="print(double(-4))"/>
Standard Input	<input type="text"/>
Expected output	<input type="text" value="-8"/>
Extra template data	<input type="text"/>
Test properties:	<input type="checkbox"/> Use as example   Display: Show   <input type="checkbox"/> Hide rest if fail   Mark: 1.000   Ordering: 20

## Step 8: Save changes

## Step 9: Preview and test

Page 1

1 **Double function in python** Write a function double(n) that returns twice ...

Always latest   1.00

Shuffle

**Question 1**

Correct

Marked out of 1.00

Write a function double(n) that returns twice its parameter n.

**For example:**

Test	Result
print(double(3))	6

**Answer:** (penalty regime: 10, 20, ... %)

```
1 def double(n):  
2     return n * 2
```

**Check**

	Test	Expected	Got	
✓	print(double(3))	6	6	✓
✓	print(double(-4))	-8	-8	✓
✓	print(double(0))	0	0	✓
✓	print(double(10.5))	21.0	21.0	✓

Passed all tests! ✓

Start again

Save

Fill in correct responses

Submit and finish

Close preview

**Optional settings:**

Answer box: set the number of rows to allocate for the answer box.

Display &gt; Hide: allows for additional validation of student answers, preventing them from using alternative methods to arrive at the correct output and ensuring they provide the intended solution.

Online URL: <https://knowledgebase.xjtlu.edu.cn/article/coderunner-question-type-416.html>