Chrome File Edit View History Bookmarks Profiles Tab Window Help 100% Mon Jun 2 1:44 PM

LIConnections - VA and DoD | Python: Introduction to Debu | Instructions - platform

about:blank

1h 51m

5 Development Workflow

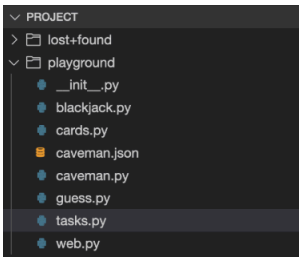
Introduction

Software development is an iterative process of writing and testing code. In certain cases the Python debugger can help to more quickly iterate on code.

The `interact` command opens up an interactive console inside the debugger. The console's global namespace includes all names in the current scope. Allowing the console to interact with a snapshot of the application at the time the interactive console was opened.

Instructions

1. Locate and open the **playground/tasks.py** file using the **IDE's Explorer/project** pane.



   Imagine inheriting incomplete code from another developer. The `tasks_by_urgency` function requires refactoring before this application will behave as expected.

2. Observe the behavior of the script.

Back

Start check

2. Observe the behavior of the script.

```
python3 playground/tasks.py
```

```
Today's TODO list:
```

When using the Python debugger as a development tool, it's common to set breakpoints in code. The built-in `breakpoint` callable opens the debugger prompt at the line after where `breakpoint` was called.

The `breakpoint` callable was introduced in Python 3.7. Prior versions imported the `pdb` module and called the `set_trace` callable. This is commonly written in a single line: `import pdb; pdb.set_trace()` .

3. Call the `breakpoint` callable on line 19.

```
def tasks_by_urgency(all_tasks):
    ''' Implement the body of this function so it returns a list of tasks sorted by urgency.
        Urgency ranges between 0 and 1 (lowest to highest).

        Args:
            _list is a list of Task objects.
    '''
    breakpoint()
    return []
```

Setting breakpoints in code doesn't require the application to be started with the `pdb` module.

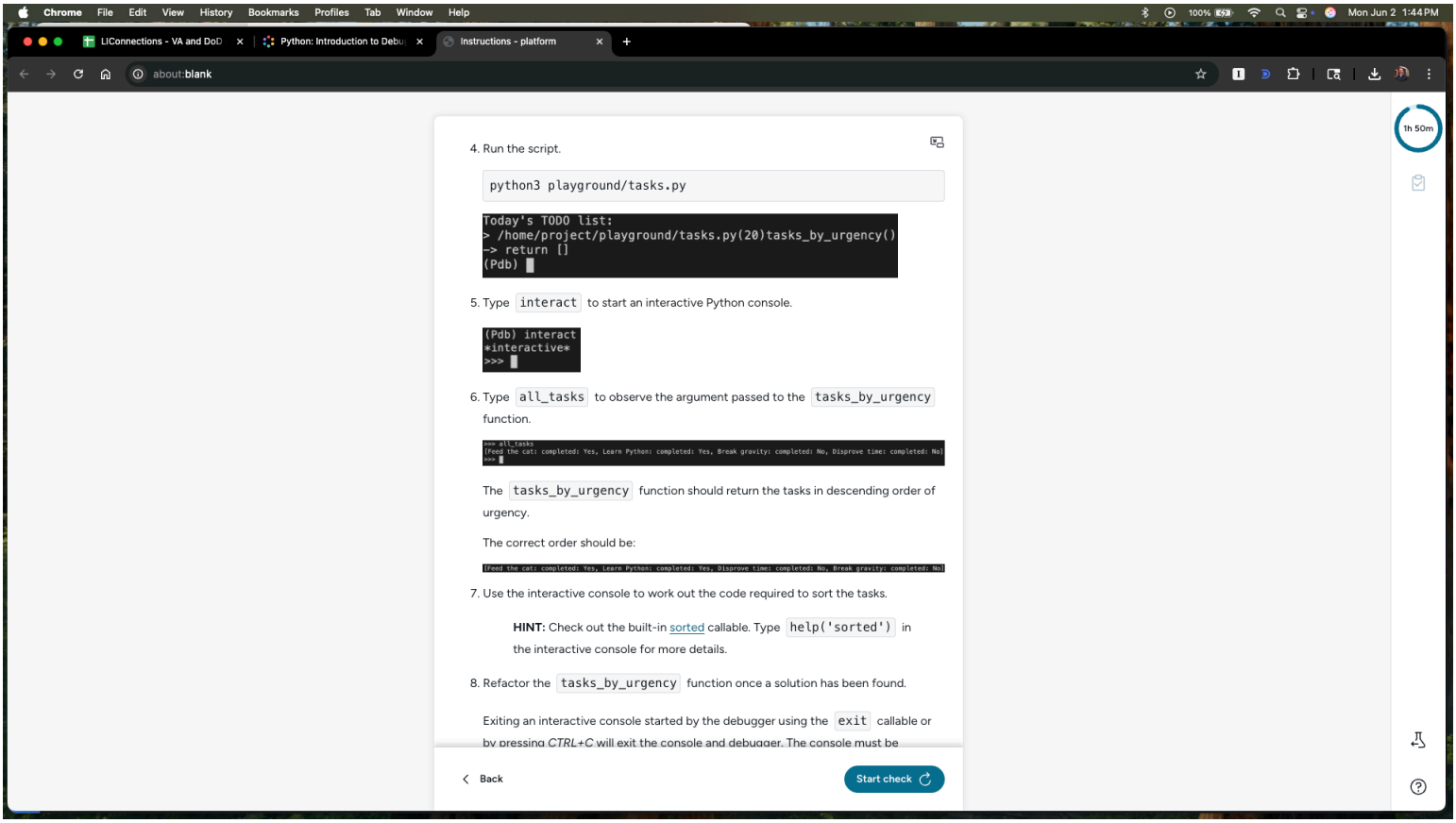
4. Run the script.

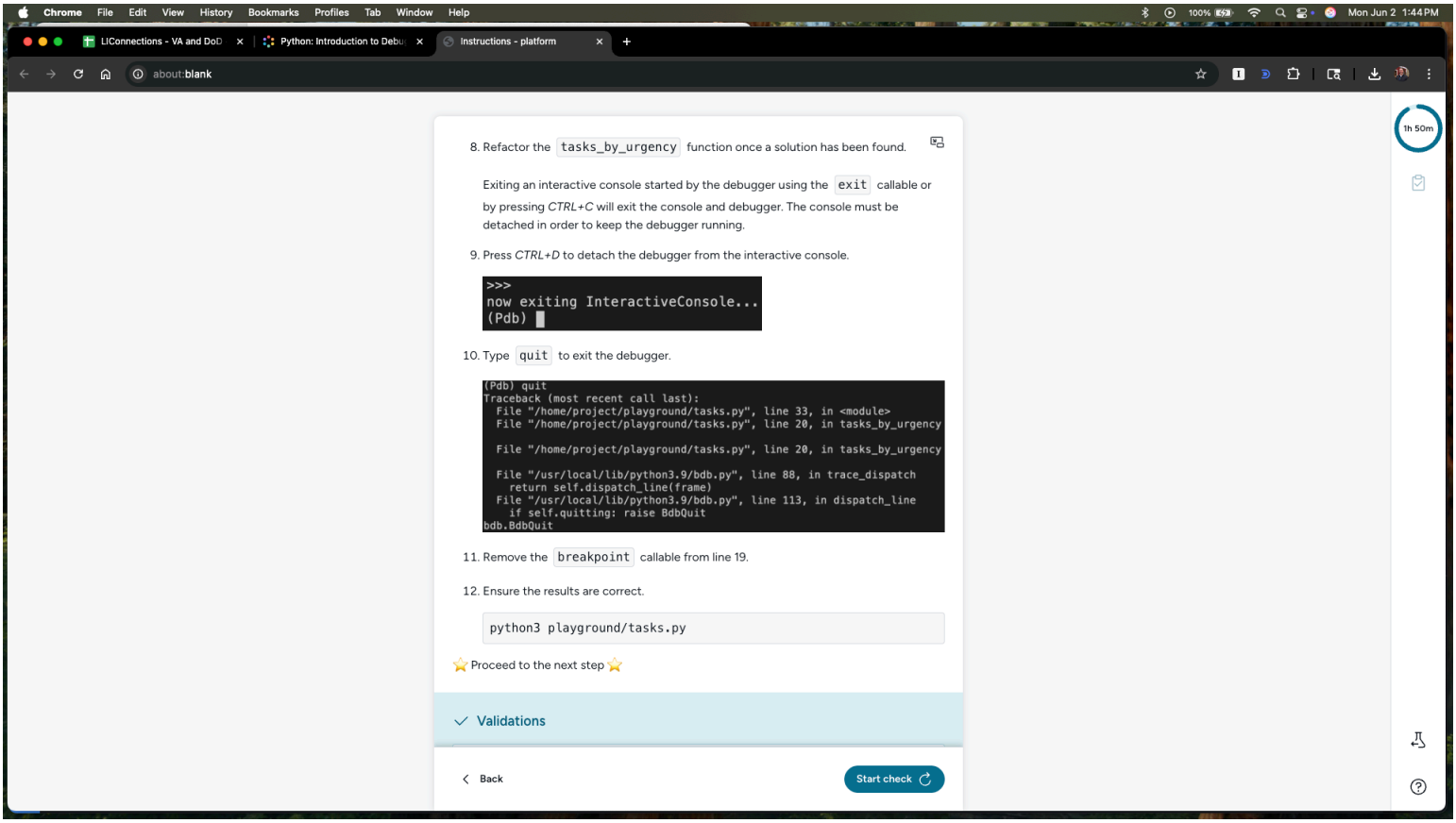
```
python3 playground/tasks.py
```

```
Today's TODO list:
> /home/project/playground/tasks.py(20)tasks_by_urgency()
```

Back

Start check


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon Jun 2 1:44 PM
LIConnections - VA and DoD
Python: Introduction to Debu
Instructions - platform
about:blank
1h 50m
4. Run the script.
python3 playground/tasks.py
Today's TODO list:
> /home/project/playground/tasks.py(20)tasks_by_urgency()
-> return []
(Pdb)
5. Type interact to start an interactive Python console.
(Pdb) interact
interactive
>>>
6. Type all_tasks to observe the argument passed to the tasks_by_urgency function.
>>> all_tasks
[Feed the cat: completed: Yes, Learn Python: completed: Yes, Break gravity: completed: No, Disprove time: completed: No]
>>>
The tasks_by_urgency function should return the tasks in descending order of urgency.
The correct order should be:
[Feed the cat: completed: Yes, Learn Python: completed: Yes, Disprove time: completed: No, Break gravity: completed: No]
7. Use the interactive console to work out the code required to sort the tasks.
HINT: Check out the built-in sorted callable. Type help('sorted') in the interactive console for more details.
8. Refactor the tasks_by_urgency function once a solution has been found.
Exiting an interactive console started by the debugger using the exit callable or by pressing CTRL+C will exit the console and debugger. The console must be
Back
Start check


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon Jun 2 1:44 PM
LIConnections - VA and DoD
Python: Introduction to Debu
Instructions - platform
about:blank
1h 50m
8. Refactor the tasks_by_urgency function once a solution has been found.
Exiting an interactive console started by the debugger using the exit callable or by pressing CTRL+C will exit the console and debugger. The console must be detached in order to keep the debugger running.
9. Press CTRL+D to detach the debugger from the interactive console.
>>>
now exiting InteractiveConsole...
(Pdb)
10. Type quit to exit the debugger.
(Pdb) quit
Traceback (most recent call last):
File "/home/project/playground/tasks.py", line 33, in <module>
File "/home/project/playground/tasks.py", line 20, in tasks_by_urgency
File "/home/project/playground/tasks.py", line 20, in tasks_by_urgency
File "/usr/local/lib/python3.9/bdb.py", line 88, in trace_dispatch
return self.dispatch_line(frame)
File "/usr/local/lib/python3.9/bdb.py", line 113, in dispatch_line
if self.quitting: raise BdbQuit
bdb.BdbQuit
11. Remove the breakpoint callable from line 19.
12. Ensure the results are correct.
python3 playground/tasks.py
⭐ Proceed to the next step ⭐
Validations
Back
Start check