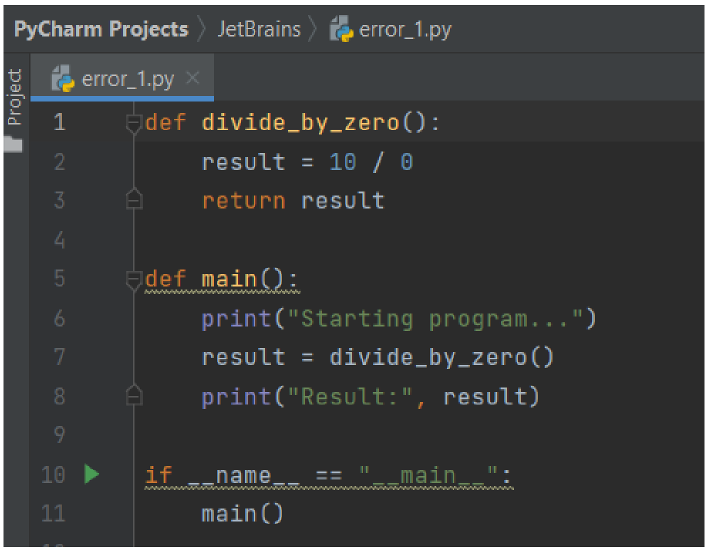
Debugging a Python Script in PyCharm

Introduction:

This guide will walk you through debugging a Python script in PyCharm which is a powerful IDE for Python development. Debugging is an essential skill which helps you identify and fix errors in your code, allowing you to write more robust and efficient programs.

Prerequisites:

- PyCharm installed on your computer.
- A Python script with an error. The script in Figure 1 will use be used as sample script for this documentation.

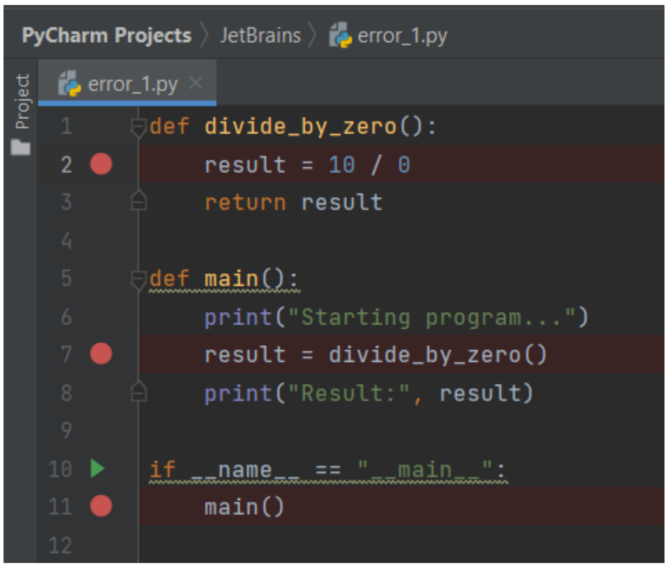


```python
def divide_by_zero():
    result = 10 / 0
    return result

def main():
    print("Starting program...")
    result = divide_by_zero()
    print("Result:", result)

if __name__ == "__main__":
    main()
```

Figure 1 - Python Script with Error

Steps:

1. **Create or Open your Python Script:**
   - Open your Python script in PyCharm by either creating a new project or opening an existing one.
2. **Set Breakpoints:**
   - Breakpoints are lines of code where PyCharm will pause execution, allowing you to examine variables and the program state.
   - Click on the line number where you suspect the error might occur. A red circle will appear, indicating a breakpoint is set as done in Figure 2.

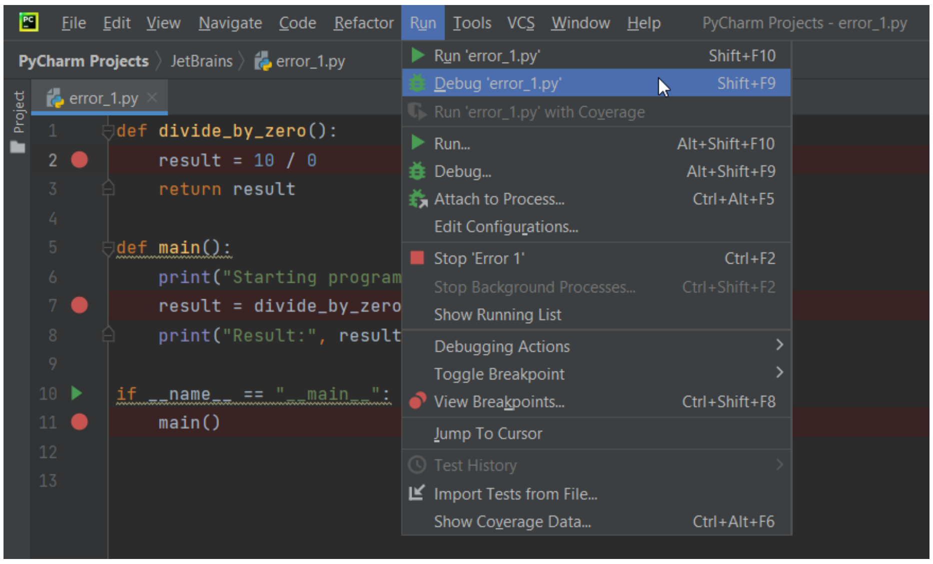


```
def divide_by_zero():
    result = 10 / 0
    return result

def main():
    print("Starting program...")
    result = divide_by_zero()
    print("Result:", result)

if __name__ == "__main__":
    main()
```

Figure 2 - Red Dots

3. **Run the Script in Debug Mode:**

   - Navigate to the **Run** menu and select **Debug error_1.py'** as seen in Figure 3. Alternatively, use the keyboard shortcut (**Shift+F9** by default).


File Edit View Navigate Code Refactor Run Tools VCS Window Help PyCharm Projects - error_1.py
PyCharm Projects 〉 JetBrains 〉 error_1.py
Project
error_1.py
Run 'error_1.py' Shift+F10
Debug 'error_1.py' Shift+F9
Run 'error_1.py' with Coverage
Run... Alt+Shift+F10
Debug... Alt+Shift+F9
Attach to Process... Ctrl+Alt+F5
Edit Configurations...
Stop 'Error 1' Ctrl+F2
Stop Background Processes... Ctrl+Shift+F2
Show Running List
Debugging Actions
Toggle Breakpoint
View Breakpoints... Ctrl+Shift+F8
Jump To Cursor
Test History
Import Tests from File...
Show Coverage Data... Ctrl+Alt+F6

```
def divide_by_zero():
    result = 10 / 0
    return result

def main():
    print("Starting program
    result = divide_by_zero
    print("Result:", result

if __name__ == "__main__":
    main()
```

Figure 3 - Running Script in Debug Mode

4. **The Debugger Panel:**

   - PyCharm will launch the script in debug mode and open a new panel at the bottom of the window. This panel displays various debugging tools.

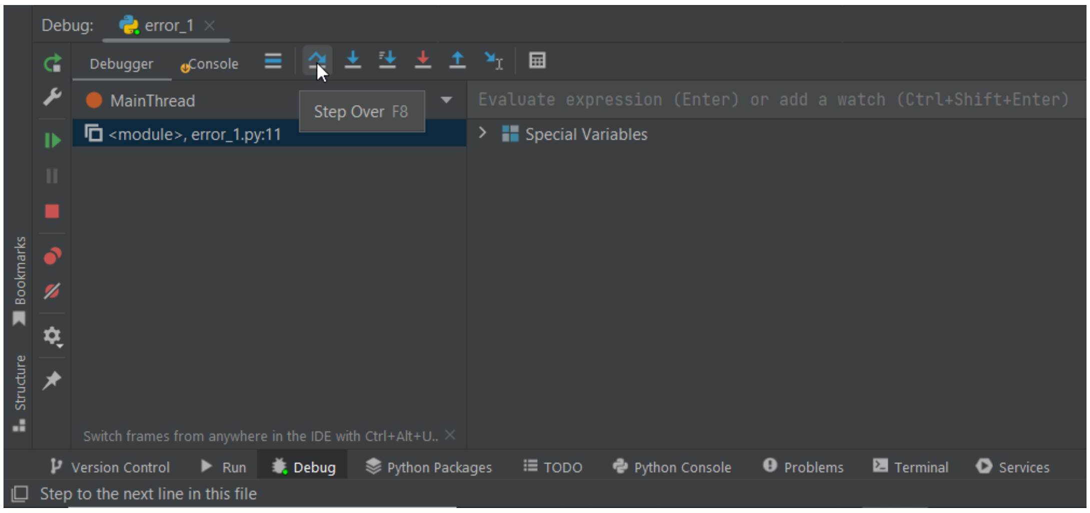


Figure 4 - Debugger Panel

5. **Step Through the Code:**

- Use the control buttons in the debugger panel to navigate your code line by line.
    - The green play button (**F9**) resumes execution until the next breakpoint. In Figure 5, all breakpoints can be seen as lines with (**F9**) resumes execution.
    - The pause button (**Shift+F9**) pauses execution at any time.
    - The step-over button (**F8**) executes the current line and moves to the next line. When all lines are finished, the errors can be also seen in Console as like in Figure 6.
    - The step-into button (**F7**) steps into function calls, allowing you to debug functions line by line.

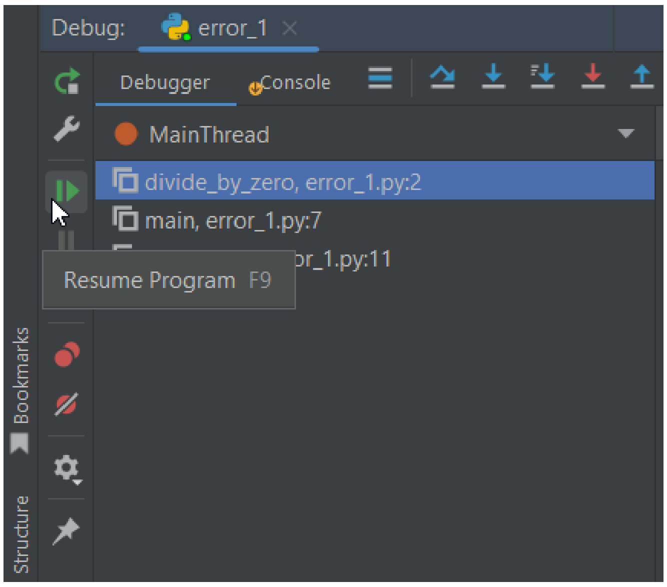
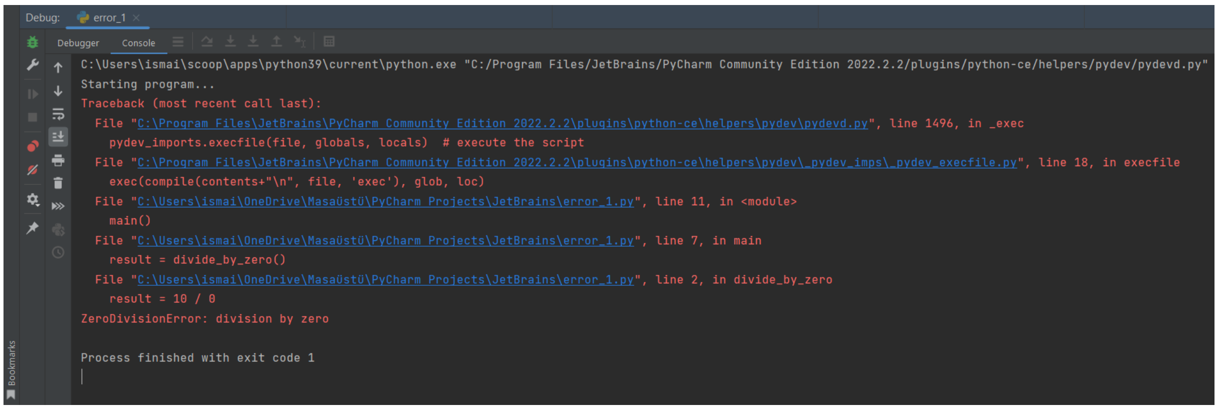


Figure 5 - All Breakpoints

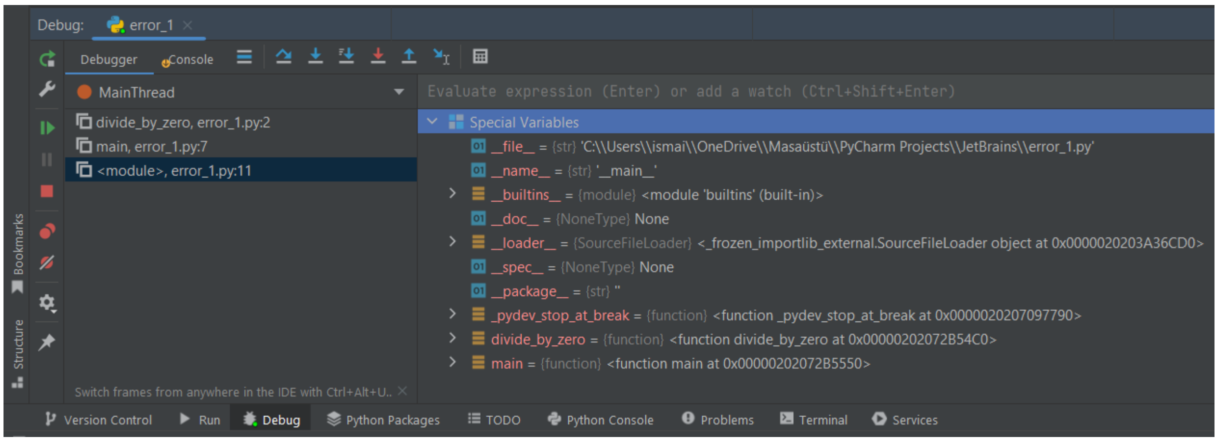


```
C:\Users\ismai\scoop\apps\python39\current\python.exe "C:/Program Files/JetBrains/PyCharm Community Edition 2022.2.2/plugins/python-ce/helpers/pydev/pydevd.py"
Starting program...
Traceback (most recent call last):
  File "C:\Program Files\JetBrains\PyCharm Community Edition 2022.2.2\plugins\python-ce\helpers\pydev\pydevd.py", line 1496, in _exec
    pydev_imports.execfile(file, globals, locals)  # execute the script
  File "C:\Program Files\JetBrains\PyCharm Community Edition 2022.2.2\plugins\python-ce\helpers\pydev\_pydev_imps\_pydev_execfile.py", line 18, in execfile
    exec(compile(contents+"\n", file, 'exec'), glob, loc)
  File "C:\Users\ismai\OneDrive\MasaÜstü\PyCharm Projects\JetBrains\error_1.py", line 11, in <module>
    main()
  File "C:\Users\ismai\OneDrive\MasaÜstü\PyCharm Projects\JetBrains\error_1.py", line 7, in main
    result = divide_by_zero()
  File "C:\Users\ismai\OneDrive\MasaÜstü\PyCharm Projects\JetBrains\error_1.py", line 2, in divide_by_zero
    result = 10 / 0
ZeroDivisionError: division by zero

Process finished with exit code 1
```

Figure 6 - Errors in Console

6. **Examine Variables:**

   - The **Variables** window on the right side of the IDE shows the values of variables at the current breakpoint. You can inspect and modify these values to understand the program's behavior.



```
MainThread
divide_by_zero, error_1.py:2
main, error_1.py:7
<module>, error_1.py:11

Switch frames from anywhere in the IDE with Ctrl+Alt+U..
```

```
Special Variables
  __file__ = {str} 'C:\\Users\\ismai\\OneDrive\\Masaüstü\\PyCharm Projects\\JetBrains\\error_1.py'
  __name__ = {str} '__main__'
> __builtins__ = {module} <module 'builtins' (built-in)>
  __doc__ = {NoneType} None
> __loader__ = {SourceFileLoader} <_frozen_importlib_external.SourceFileLoader object at 0x0000020203A36CD0>
  __spec__ = {NoneType} None
  __package__ = {str} ''
> _pydev_stop_at_break = {function} <function _pydev_stop_at_break at 0x0000020207097790>
> divide_by_zero = {function} <function divide_by_zero at 0x00000202072B54C0>
> main = {function} <function main at 0x00000202072B5550>
```

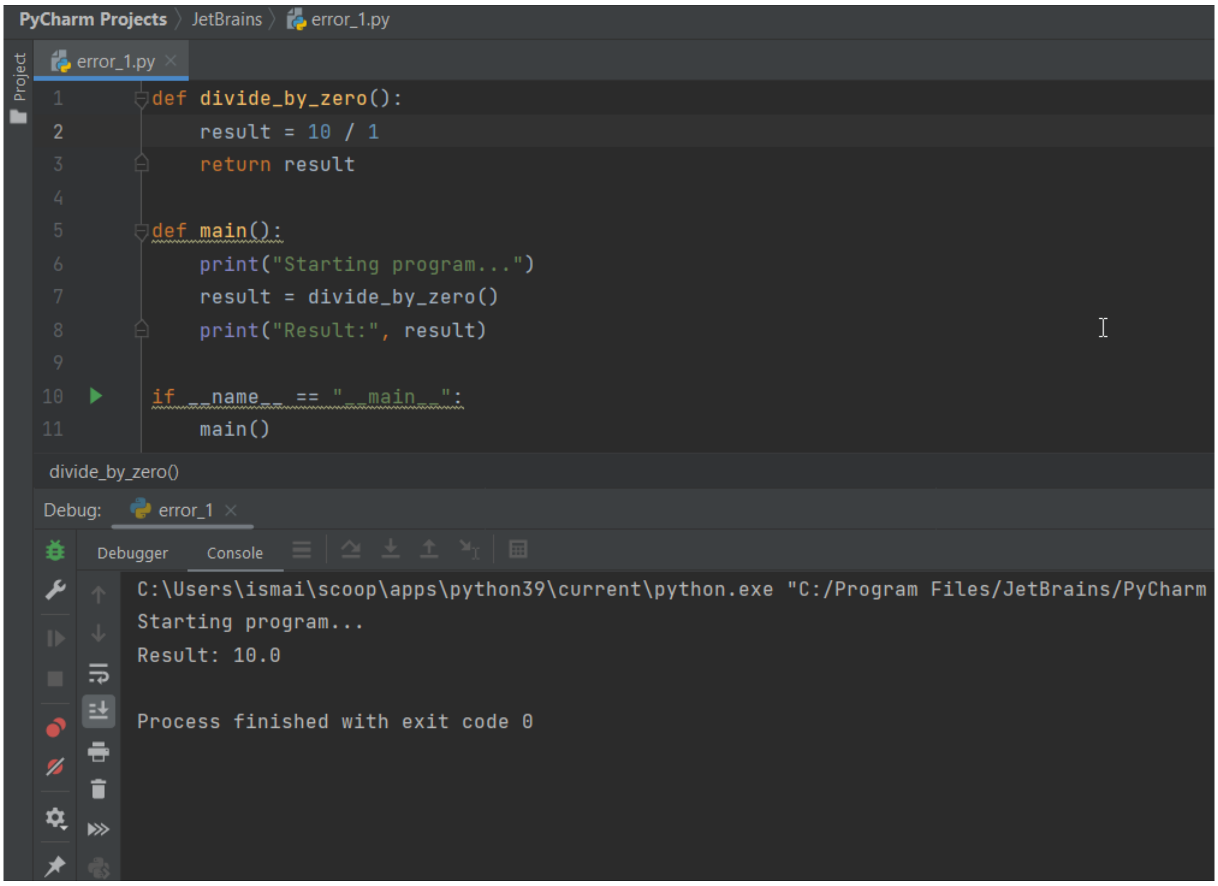
Figure 7 - The Variables Window

7. **Fix the Error and Continue Debugging:**

   - Once you've identified the source of the error, make the necessary code changes in the editor window.
   - Use the debug controls to resume execution and test your fixes.

8. **Remove Breakpoints (Optional):**

   - Right-click on the red circle breakpoint marker and select **Disable Breakpoint**. Currently, the script is working without error as resulted in Figure 8.



```
def divide_by_zero():
    result = 10 / 1
    return result

def main():
    print("Starting program...")
    result = divide_by_zero()
    print("Result:", result)

if __name__ == "__main__":
    main()
```

```
C:\Users\ismai\scoop\apps\python39\current\python.exe "C:/Program Files/JetBrains/PyCharm
Starting program...
Result: 10.0

Process finished with exit code 0
```

Figure 8 - Script without Error

Additional Tips:

- Use the **Print** statement strategically in your code to inspect variable values during execution.
- PyCharm offers a visual debugger for inspecting data structures like lists and dictionaries.
- Explore the extensive debugging features in PyCharm's documentation for more advanced scenarios.