

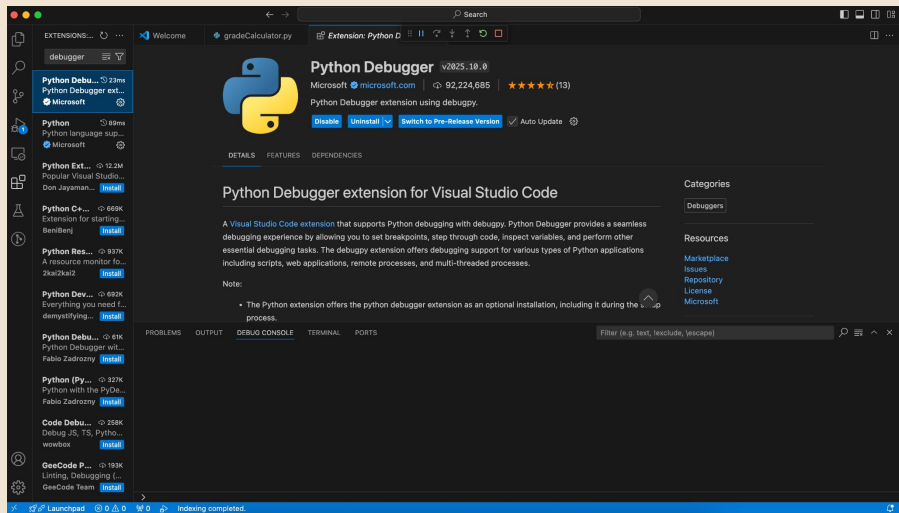


Python Debugging Cheat Sheet (VS Code Edition)



Overview: Debugging

- Computers follow strict syntax rules, so even though your code may look correct a bug can make the program fail
- Debugging is the process of identifying what type of error occurred, locating where it happened, and correcting it
- Types of errors:
 - Syntax errors - mistakes in the code's structure, code will not run
 - Runtime errors - an error that occurs while your program is running
 - Logic errors - the code runs but gives the wrong result
- Most debugging is done directly in our editor (VS Code) using features like breakpoints, variable inspection, and stepping through the code





Syntax Errors

- A syntax error happens when python cannot understand the code due to mistakes in the grammar of the language
- Common examples include:
 - Missing colons (:)
 - Unmatched parenthesis or quotes
 - Misspelled keywords
 - Missing indentation
- Ex -

```
grade = int(input("Enter your grade: "))  
if grade >= 90  
    print("A")
```





Syntax Errors



- A syntax error happens when python cannot understand the code due to mistakes in the grammar of the language
- Common examples include:
 - Missing colons (:)
 - Unmatched parenthesis or quotes
 - Misspelled keywords
 - Missing indentation
- Ex -

```
grade = int(input("Enter your grade: "))  
if grade >= 90  
    print("A")
```

SyntaxError: '(' was never closed

```
⊗ alizah@Alizahs-MacBook-Pro Debugging Example % python3 gradeCalculator.py  
File "/Users/alizah/Documents/cs1030/Debugging Example/gradeCalculator.py", line 12  
    grade = int(input("Enter your grade: "))  
                ^
```

SyntaxError: '(' was never closed

```
⊗ alizah@Alizahs-MacBook-Pro Debugging Example % python3 gradeCalculator.py  
File "/Users/alizah/Documents/cs1030/Debugging Example/gradeCalculator.py", line 13  
    if grade >= 90  
        ^
```

SyntaxError: expected ':'





Indentation Errors

- An indentation error occurs when the spacing or structure doesn't match python's structure rules
- Keeping track of indentation especially in applications such as conditional statements (if, if-else), loops (for, while), and function definitions
- Ex -

```
grade = int(input("Enter your grade: "))  
if grade >= 90:  
print("A")
```





Indentation Errors

- An indentation error occurs when the spacing or structure doesn't match python's structure rules
- Ex -

```
grade = int(input("Enter your grade: "))  
if grade >= 90:  
print("A")
```

```
⊗ alizah@Alizahs-MacBook-Pro Debugging Example % python3 gradeCalculator.py  
File "/Users/alizah/Documents/cs1030/Debugging Example/gradeCalculator.py", line 14  
print("A")  
IndentationError: expected an indented block after 'if' statement on line 13  
○ alizah@Alizahs-MacBook-Pro Debugging Example % █
```





Runtime Errors

- A runtime error is an error that occurs while the program is running
- The code syntax is fine, but something goes wrong during execution
- Common errors include:
 - array index out of bounds
 - file not found
 - type mismatch
- Ex -

```
grade = int(input("Enter your grade: "))  
if grade >= 90:  
    grade /= 0  
    print("A")
```





Runtime Errors

- A runtime error is an error that occurs while the program is running
- The code syntax is fine, but something goes wrong during execution
- Common errors include:
 - Array index out of bounds
 - File not found
 - Type mismatch
- Ex -

```
grade = int(input("Enter your grade: "))  
if grade >= 90:  
    grade /= 0  
    print("A")
```

```
alizah@Alizahs-MacBook-Pro Debugging Example % python3 gradeCalculator.py  
Enter your grade: 90  
Traceback (most recent call last):  
  File "/Users/alizah/Documents/cs1030/Debugging Example/gradeCalculator.py", line 14, in <module>  
    grade /= 0  
ZeroDivisionError: division by zero  
alizah@Alizahs-MacBook-Pro Debugging Example %
```





Logic Errors

- A logic error is when your code runs without crashing but produces the wrong result
- Python doesn't know it's wrong, only you can detect it by looking at the output
- Common examples include:
 - Incorrect formulas or calculations
 - Incorrect loop/array indices
 - Incorrect conditional statements
- Ex -

```
grade = int(input("Enter your grade: "))  
if grade >= 90:  
    print("Below A")
```





Logic Errors

- A logic error is when your code runs without crashing but produces the wrong result
- Python doesn't know it's wrong, only you can detect it by looking at the output
- Common examples include:
 - Incorrect formulas or calculations
 - Incorrect loop/array indices
 - Incorrect conditional statements
- Ex -

```
grade = int(input("Enter your grade: "))  
if grade >= 90:  
    print("Below A")
```

```
● alizah@Alizahs-MacBook-Pro Debugging Example % python3 gradeCalculator.py  
Enter your grade: 97  
Below A
```





Correct Letter Grade Program

```
grade = int(input("Enter your grade: "))  
if grade >= 90:  
    print("Below A")
```

Why this works:

- Conditions are properly ordered
- It has proper indentation and syntax

Note: although this does work, that does not mean that it is high quality code





Debugging in VS Code

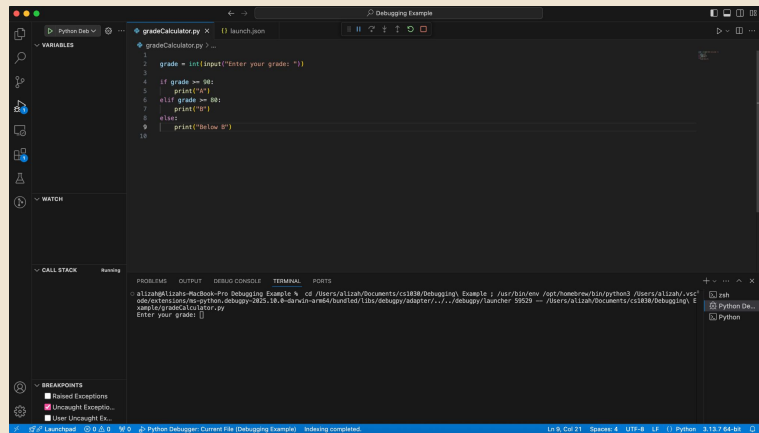
- VS Code includes built-in debugging tools to help you step through your code and inspect values
- Important debugging actions:
 - Setting a breakpoint:
 - Click to the left of a line number
 - VS Code will stop running at that line and give you control when debugging
 - Run the debugger:
 - Click on the run and debug tab then press the green start debugging button
 - The code pauses at the first breakpoint
 - Step into:
 - Goes inside the function or condition
 - Step over:
 - Executes the current line and goes to the next
 - Variables panel:
 - In the variables panel on the left, track live changes to variables
 - Observe flow control:
 - Follow which condition or branch is being executed (if, elif, else)





Debugging Tips and Tricks

- Read the error message carefully
 - It usually tells you the type of error and the line number
- Use print() to check variables
 - If you aren't debugging and want to check a variables value you can just print it out to see what value it holds (ex - print("current grade", grade))
 - This helps to identify logic issues
- Instead of guessing what the issue is, use the VS Code debugger
 - This will save you a lot of time and effort
- Save often and test small changes
 - It's easier to test your code as you go rather than writing large chunks at once
 - This helps catch small issues early and prevents bigger debugging problems later





Summary

Syntax Error

Code violates Python grammar

Ex - missing colon

Indentation Error

Incorrect spacing

Ex - misaligned print()

Runtime Error

Code runs then crashes

Ex - dividing by 0

Logic Error



Code runs but gives wrong output

Ex - conditions are in wrong order





Grade Calculator Lab

1. Download the following code file and open it up in VS Code: [lab.py](#)
 2. Look through the code and start writing comments explaining what it is doing
 3. There are errors in this code, runtime, logic, and syntax errors, your job is to find and fix them
 4. Discuss what you learned
- 
- 

Help and Resources

Official Python Documentation

- <https://docs.python.org/3/tutorial/index.html>
- Official and most accurate

W3Schools

- <https://www.w3schools.com/python/>
- Beginner-friendly explanations with live examples

GeeksForGeeks

- <https://www.geeksforgeeks.org/python/python-programming-language-tutorial/>
- Great for understanding both why and how something works

