

## Error Messages & Debugging

---

### Textbook

---

## Error Messages & Debugging



Working through bugs is just an everyday part of coding. Finding problems and looking for ways to overcome them is part of the creative problem solving process of coding. We try something to see if it works, tweak it, adjust things, fix the bugs, and move forward. This is also true for life as we run into challenges.

In programming, sometimes it can be difficult to find where the bug is. Learning more about different kinds of bugs and error messages can help us find bugs quicker and easier.

### Error Types

Keep in mind that there are several types of problems that the program could have. If a piece of code isn't doing what you want it to, it may be because you actually did type in the rules for the code correct, but the logic is off. While debugging, it's important to consider different reasons it might not be working.

There are different kinds of bugs/errors:

**Syntax error** - The characters were typed in incorrectly (for example, missing a semicolon in the right place)

**Runtime error** -- When the code is correct, but can run into issues depending on how the program is run. An example of a runtime error is when we try to divide any number by zero.

**Logic error** -- The code is technically written without mistakes, but the logic doesn't accomplish what it is supposed to. For example, the code is running just fine, but it's not passing the challenge. This is because the code is fine, but isn't accomplishing what the directions say. Logic errors are particularly common in Skill Struck.

When combing through code that's not working, consider each kind of problem to find the bug. It may not be a problem with the way it's typed.

## Computation Error

When computers perform calculations, especially with very large or very small numbers, or when converting between different number types (like whole numbers and decimals), they can sometimes introduce tiny inaccuracies. This is known as a **computation error**, or sometimes a floating-point error, and it happens because computers have limited space to store numbers.

For example, dividing 1 by 3 results in 0.3333... repeating, but a computer can only store a finite number of 3s, leading to a slightly imprecise result. These small inaccuracies can accumulate over many calculations and lead to an overall result that isn't exactly what you'd expect, even if the code's logic is perfectly sound and there are no syntax or runtime issues.

## Python and JavaScript Read the Error

Often, people get an error and immediately give up or get frustrated without ever reading what the error itself says. It often gives clues as to where the problem is.

- Check for any mention of a line in your code to check.
- If something is **undefined**, that often means there is a variable that hasn't been created yet.
- For Python, check for indentation inconsistencies.

Take some time to read through the error and find clues there. You can practice with a few common error messages here.

### Example 1

```
File "/home/main.py", line 3
```

```
= earningsgoal/12
```

```
IndentationError: unexpected indent
```

This error message shows that on line 3 there is an indentation problem. So check your indentation near line 3. Check the lines directly before and after line 3 to see if there's a problem.

### Example 2

```
File "/home/main.py", line 3, in <module>
```

```
months = earningsgoal/12
```

```
TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

This error message means that on line 3, a variable is being used that is the wrong variable type. **str** and **int** are referring to the variable types string and integer. Check to see if you are trying to use a string to do math. If you are, remember to convert it to an integer using `int()`.

### Example 3

```
File "/home/main.py", line 3, in <module>
```

```
months = earningsgoal/12
```

```
NameError: name 'earningsgoal' is not defined
```

This error message means that a variable is being used that hasn't been defined (or created) yet. In this case, the variable `earningsgoal` needs to be created before we can do something with it.

## Python and JavaScript: Where is the Error Message?

If the error message shows up in the black console on the right of your Skill Struck code page, this means there's a problem with the actual functionality of your code. This indicates that your bug is more likely a **syntax error**.

If the error message shows up in the requirements of a specific challenge or checkpoint on the left of your Skill Struck code page, this means that your code doesn't meet what the challenge or checkpoint is looking for. This indicates that your bug is more likely a **logic error**.

Sometimes identifying what kind of problem you are having with your code can help you find the bug.

## No Error Messages in HTML or CSS?

The biggest indicator of a bug in HTML is that your tag just isn't showing or manifesting on the page.

### Color Scheme

The code editor is set up in a color scheme that gives you clues as to where the bug is. You'll be able to notice an error in the code because the color pattern will be messed up. Each code editor has a slightly different color scheme, but they all follow some sort of pattern. You'll start to catch on to what the pattern is and your eye will naturally notice when something is off.

Take a look at the following example. You'll notice there's a pattern to which bits of code have a certain color. See if you can find where the pattern gets off.

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" type="text/CSS" href="cssExercises.css">
  </head>

  <body>
    <h1> Summer </h1>

    <h3 style="color:blue ; font-family:cursive"> I love summer! </h3>
    <h3 style="background-color: green; color: white">  There are so many
    <h3 style "background-color: yellow: font-family: fantasy"> I love to
  </body>
</html>
```

Do you see how the third `<h3>` tag has a different color pattern? This is where the bug is.

One of the `<h3>` tags is missing an equals sign `=`.

Taking a look at the color patterns can help you identify where the colors might be off.

Also, the color scheme helps to find **syntax errors** by highlighting code that has been typed incorrectly.

## Retype the Code

Sometimes some strange spacing causes code to not run correctly. Try retyping the code out from scratch and see if it eliminates unusual hiccups in code.

Retyping out code is a good approach because it forces you to look at each character individually. This often helps identify bugs that your eye just passes over when you're just looking for it.

## Rubber Duck Method



A great method that is common in computer science is referred to as the "rubber duck method." The idea is that you talk through the problem out loud -- some people find it easier to talk "to something" like a rubber duck on their desk. Often, through the process of vocalizing your problem you're able to think through the answer before you've even finished explaining the problem.

## Backtrack to a Working Code

When code breaks, it's helpful to instruct students to backtrack what they've done back to a code that still works. From there you can step by step retrace the code you tried until you can identify what they added that stopped it from working. Then you can identify where the bug is.

## Summary

Debugging is just a part of coding. It's all part of the puzzle solving of putting working code together. Sometimes it can be difficult to identify the problem.

In this lesson we talked about different strategies that help with debugging. We examined the different types of errors. We explored different strategies such as noticing the color scheme, retyping the code, the rubber duck method, reading the error message, and backtracking to a working code. These tips will help everyone debug quicker and easier.

## Questions (10)

**1. What are the three main types of errors? Select 3.**

SELECT MULTIPLE

Select all that apply:

- A. syntax
- B. logic
- C. runtime
- D. language

**2. What is the biggest indicator of a bug in HTML?**

MULTIPLE CHOICE

Choose the correct answer:

- A. An error message
- B. That the HTML tag isn't showing up on the webpage
- C. The program crashes
- D. Where the error message shows

**3. True or False: Working through bugs is just a part of the coding process.**

MULTIPLE CHOICE

Choose the correct answer:

- A. True
- B. False

**4. Which of the following are strategies you can use to find the bug? Select all that apply.**

SELECT MULTIPLE

Select all that apply:

- A. rubber duck method
- B. read the error message
- C. check the color of the code in the editor
- D. backtrack to a working code
- E. retype the code
- F. keep coding and often it will fix itself

**5. True or False: The rubber duck method is only used by beginners who are learning code**

MULTIPLE CHOICE

**Choose the correct answer:**

- A. True
- B. False

**6. In the following error, what line of code has a problem?**

MULTIPLE CHOICE

File `"/home/main.py"`, line 5 `days = weeks / 7` IndentationError: unexpected indent

**Choose the correct answer:**

- A. Line 5
- B. Line 7
- C. Line 1
- D. Line 10

**7. What is the problem with the code that threw this error?**

MULTIPLE CHOICE

File `"/home/main.py"`, line 4, in `weeks = months / 4` NameError: name 'months' is not defined

**Choose the correct answer:**

- A. There's an indentation problem
- B. The variable named "months" hasn't been created yet
- C. The wrong variable type is being used
- D. The tag isn't showing up

**8. Which type of error is where the characters were typed in incorrectly?**

MULTIPLE CHOICE

**Choose the correct answer:**

- A. syntax error
- B. logic error
- C. runtime error



**9. True or False: All code editors have the same color scheme.**

**Choose the correct answer:**

- A. True
- B. False

**10. What is likely the problem with a chunk of code that threw this error?**

File "/home/main.py", line 3 = earningsgoal/12 IndentationError: unexpected indent

**Choose the correct answer:**

- A. Use of a variable that hasn't been created yet.
- B. A runtime error
- C. An indentation problem
- D. A missing closing tag

## Answer Keys & Solutions

### Questions

1. What are the three main types of errors? Select 3.

SELECT MULTIPLE

Correct Answers:

- A. syntax ✓ Correct
- B. logic ✓ Correct
- C. runtime ✓ Correct
- D. language ✗ Incorrect

**Explanation:**

language is not an error type

2. What is the biggest indicator of a bug in HTML?

MULTIPLE CHOICE

Correct Answer:

- A. An error message ✗ Incorrect
- B. That the HTML tag isn't showing up on the webpage ✓ Correct
- C. The program crashes ✗ Incorrect
- D. Where the error message shows ✗ Incorrect

**Explanation:**

When a tag isn't showing on the webpage, that means there's a problem with it.

3. True or False: Working through bugs is just a part of the coding process.

MULTIPLE CHOICE

Correct Answer:

- A. True ✓ Correct



B. False

✗ Incorrect

**Explanation:**

Professional programmers work through bugs every day

**4. Which of the following are strategies you can use to find the bug? Select all that apply.**

SELECT MULTIPLE

**Correct Answers:**

A. rubber duck method

✓ Correct

B. read the error message

✓ Correct

C. check the color of the code in the editor

✓ Correct

D. backtrack to a working code

✓ Correct

E. retype the code

✓ Correct

F. keep coding and often it will fix itself

✗ Incorrect

**Explanation:**

There are 5 correct answers

**5. True or False: The rubber duck method is only used by beginners who are learning code**

MULTIPLE CHOICE

**Correct Answer:**

A. True

✗ Incorrect

B. False

✓ Correct

**Explanation:**

Professional programmers frequently use the rubber duck method

MULTIPLE CHOICE

## 6. In the following error, what line of code has a problem?

Correct Answer:

- |            |             |
|------------|-------------|
| A. Line 5  | ✓ Correct   |
| B. Line 7  | ✗ Incorrect |
| C. Line 1  | ✗ Incorrect |
| D. Line 10 | ✗ Incorrect |

### Explanation:

Check where it says "line"

## 7. What is the problem with the code that threw this error?

MULTIPLE CHOICE

Correct Answer:

- |  |             |
|--|-------------|
| A. There's an indentation problem                      | ✗ Incorrect |
| B. The variable named "months" hasn't been created yet | ✓ Correct   |
| C. The wrong variable type is being used               | ✗ Incorrect |
| D. The tag isn't showing up                            | ✗ Incorrect |

### Explanation:

"not defined" is the same as "hasn't been created yet."

## 8. Which type of error is where the characters were typed in incorrectly?

MULTIPLE CHOICE

Correct Answer:

- |                  |             |
|------------------|-------------|
| A. syntax error  | ✓ Correct   |
| B. logic error   | ✗ Incorrect |
| C. runtime error | ✗ Incorrect |

### Explanation:

Syntax refers to the characters used in the code.

## 9. True or False: All code editors have the same color scheme.

MULTIPLE CHOICE

Correct Answer:

- A. True ✗ Incorrect
- B. False ✓ Correct

### Explanation:

Code editors have their own color scheme that you will need to get used to.

## 10. What is likely the problem with a chunk of code that threw this error?

MULTIPLE CHOICE

Correct Answer:

- A. Use of a variable that hasn't been created yet. ✗ Incorrect
- B. A runtime error ✗ Incorrect
- C. An indentation problem ✓ Correct
- D. A missing closing tag ✗ Incorrect

### Explanation:

Read the error carefully