

<

Lesson 5:
Functions

☰

Scope Recap

📖

📁

🔍

✓ 1. Intro to Functions

✓ 2. Function Example

✓ 3. Declaring Functions

✓ 4. Function Recap

✓ 5. Quiz: Laugh it Off 1 (5-1)

✓ 6. Quiz: Laugh it Off 2 (5-2)

✓ 7. Return Values

✓ 8. Using Return Values

✓ 9. Scope

✓ 10. Scope Example

✓ 11. Shadowing

✓ 12. Global Variables

✓ 13. Scope Recap

● 14. Hoisting

● 15. Hoisting Recap

● 16. Quiz: Build a Triangle (5-3)

● 17. Function Expressions

● 18. Patterns with Function Expressions

● 19. Function Expression Recap

● 20. Quiz: Laugh (5-4)

● 21. Quiz: Cry (5-5)

● 22. Quiz: Inline (5-6)

● 23. Lesson 5 Summary

What you've learned so far:

- If an identifier is declared in **global scope**, it's available *everywhere*.
- If an identifier is declared in **function scope**, it's available *in the function* it was declared in (even in functions declared inside the function).
- When trying to access an identifier, the JavaScript Engine will first look in the current function. If it doesn't find anything, it will continue to the next outer function to see if it can find the identifier there. It will keep doing this until it reaches the global scope.
- Global identifiers are a bad idea. They can lead to bad variable names, conflicting variable names, and messy code.

Mentorship

Get support and stay on track

NEXT

MENTORSHIP