
Welcome to
lecture 2!

Agenda

Session Goals

- A brief history of programming languages
- Introduction to Javascript
- Learn to write pop-up dialogs (alert, prompt, confirm)
- Understand variables and data types
- Apply arithmetic and logical operators
- A mini-task for you to work on!

Learning Outcomes

- Master pop-up dialogs
- Understand variables and data types
- Use arithmetic & logical operators
- Be able to write some code!

A Brief History of Programming Languages

- What's a programming Language?
  - A structured set of instructions and syntax used to communicate with and control a computer to perform specific tasks
- Early Beginnings
  - Machine Code & Assembly – the first "languages" used to directly control hardware
- 1950s - 1960s
  - High-Level Languages emerge (e.g., Fortran, COBOL) to simplify coding
- 1970s - 1980s
  - Structured & Procedural Languages (C, Pascal) improve clarity and maintenance
  - Introduction of Object-Oriented Concepts (C++, Smalltalk)
- 1990s Onwards
  - Languages for the Web (JavaScript, PHP) and modern languages (Python, Ruby, Rust)

Programming Paradigms: Building Blocks

- Procedural Programming
  - Focuses on a sequence of instructions
  - Example: Using functions to structure code
- Object-Oriented Programming (OOP)
  - Organizes code into objects and classes
  - Promotes reuse and modularity
- Functional Programming
  - Emphasizes immutability and pure functions
  - Allows for concise, expressive code (increasingly relevant in modern JavaScript)
- JavaScript’s Flexibility
  - Supports procedural, object-oriented, and functional styles

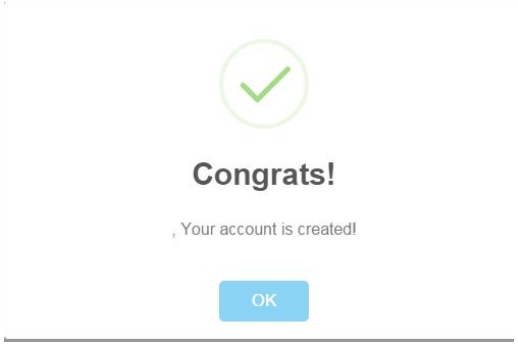
Language of the web: Javascript

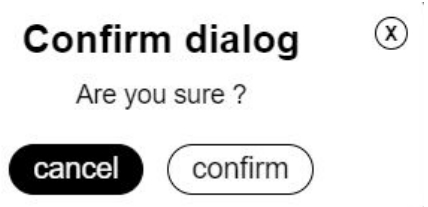
- What is JavaScript?
  - A high-level, dynamic scripting language developed in 1995 by Brendan Eich
- Core Characteristics
  - Runs in the browser to add interactivity
  - Lightweight, interpreted, and versatile
  - Supports multiple paradigms (procedural, OOP, functional)
- Role in Web Development
  - Brings static web pages to life through interactive elements
  - Works seamlessly with HTML & CSS
- Relevance to backend development
  - Introduction of Node.js means Javascript could be used on the backend

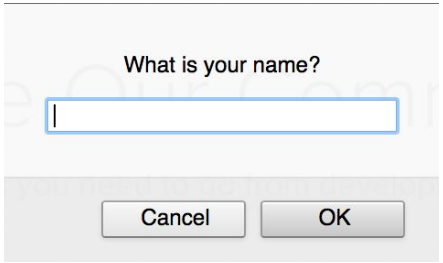
Javascript: Basics

Alerts, Prompts & Confirm Dialogs

- What's the purpose?
  - To improving the interaction with the user
- What is an alert?
  - Displays a simple message
- What is a prompt?
  - Asks for user input
  - Returns a string
- What's a confirm?
  - Asks user for confirmation
  - Returns true/false





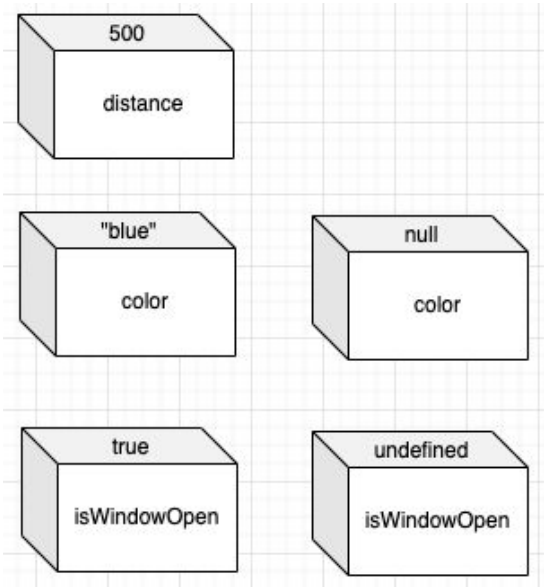


Let’s practice pop ups!

- Alert
  - Write a simple alert with a message of your choice. Eg: “My name is __”
- Prompt
  - Use a prompt to ask for the user’s favorite color
- Confirm
  - Write a confirm to ask the user whether they like programming

Variables & Data Types

- What are variables?
  - Containers for storing data
  - Two kinds of variables: primitive and object types
- Data Types
  - String: e.g. “hello”
    - Concatenate strings using “+”
  - Number: e.g. 42, 3.14
  - Boolean: true/false
  - Undefined: Declared but not assigned
  - Null: Explicitly empty
- Variables can be declared using let, const or var
  - var was the original way to declare a variable, but had limitations
  - let and const is the recommended way in new versions of Javascript
- Converting from one data type to another is called Type Casting
  - Number(“12”) -> converts string to number



Let’s practice variables!

- Favorite Food
  - Declare a variable, favoriteFood
  - Assign the variable a value representing your favorite food (eg: dosa)
  - Then use an alert to display “My favorite food is __”

Understanding Operators

- Arithmetic Operators
  - Addition (+) & Subtraction (-)
  - Multiplication (*) & Division (/)
  - Modulo (%)
- Logical Operators
  - AND (&&)
  - OR (||)

    NOT (!)

Let’s practice operators!

- Simple math
  - Declare two number variables, a and b
  - Assign values to a and b
  - Create a third variable, c, to hold the sum of a & b
  - Use an alert to show the sum

Understanding Operators

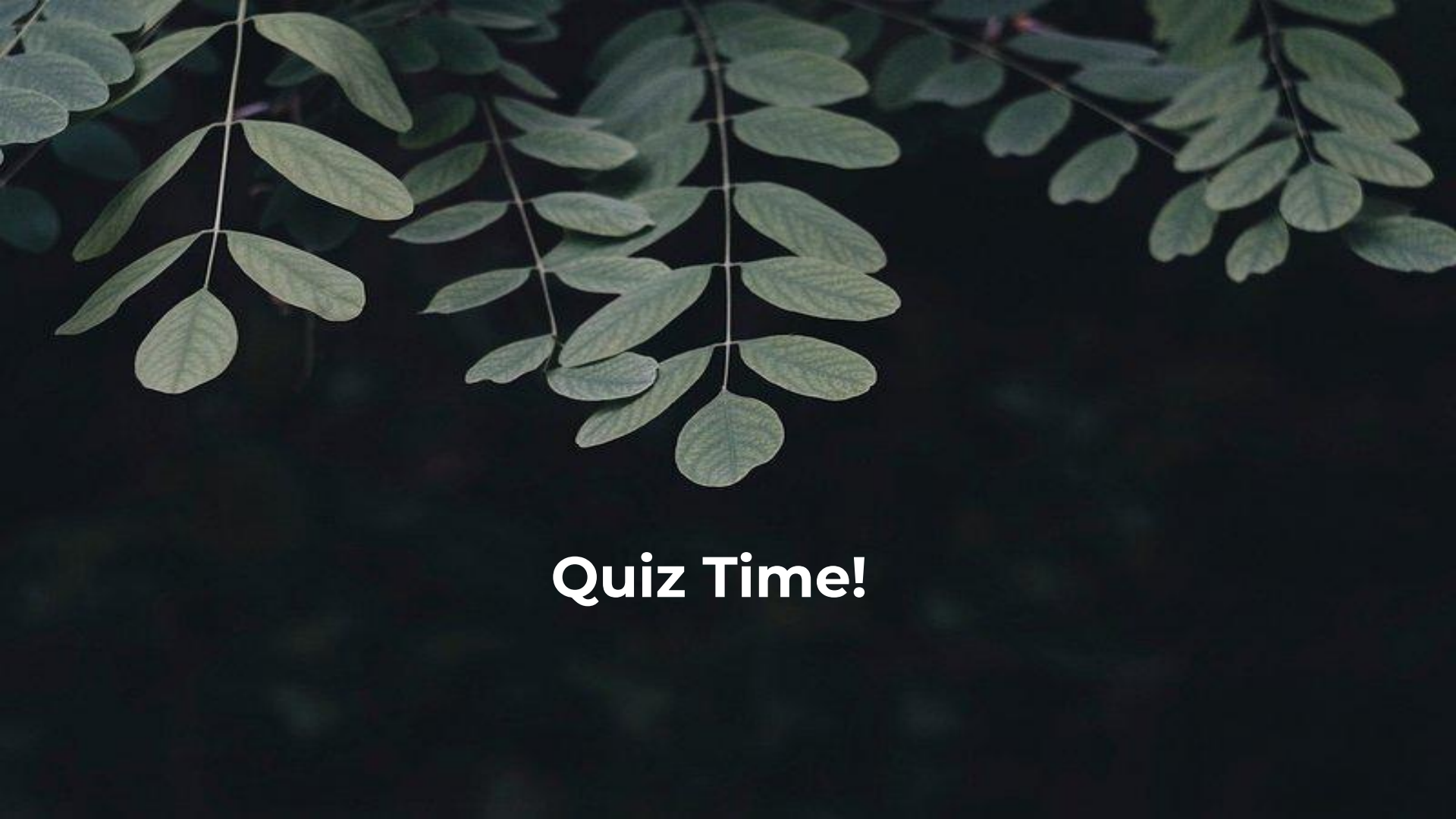
- Comparison Operators
  - **==**: Checks if two values are equal. Performs type conversion. Eg: 5 == “5” is true
  - **===**: Strict equality. Checks if two values are equal and of the same type. 5 === “5” is false
  - **!=**: Checks if two values are not equal, with type conversion. 5 != “5” is false
  - **!==**: Strict inequality. Checks if two values are not equal or not of the same type. 5 !== "5" is true
  - **>**: Greater than
  - **<**: Less than
  - **>=**: Greater than or equal to
  - **<=**: Less than or equal to

Exercise: Operator Challenges

- Task 1
  - Prompt for two numbers and display their sum, difference, product, quotient, and remainder.
- Task 2
  - Use comparison operators to check if the first number is greater than the second.
- Task 3
  - Combine logical expressions to output a true/false result.

Mini-task: Age Calculator

- Prompts the user for their birth year.
- Calculates the current age using the current year.
- Calculates the year the user will turn 100 (by adding 100 to the birth year).
- Displays both messages using alerts.


Quiz Time!

What does prompt() return?

- A) Number
- B) Boolean
- C) String
- D) Undefined

Correct Answer: C

Which operator checks for strict equality?

- A) ==
- B) =
- C) ===
- D) !==

Correct Answer: C

What is the output of this code snippet?

```
let result = "5" + 3;

alert(result);
```

- A) 8
- B) “8”
- C) “53”
- D) NaN

Correct Answer: 53
Why?
When a string is added to a number, JavaScript converts the number to a string and concatenates them, so the correct answer is “53”

Recap

- Introduction to programing languages & paradigms
- Pop-Up Dialogs: alert(), prompt(), confirm()
- Variables & Data Types: Strings, Numbers, Booleans, Undefined, Null
- Operators: Arithmetic, Comparison, & Logical
- Mini Project: Integrated example (Age Calculator)

References

- Mozilla Developer Network (documentation)
- W3Schools JavaScript Tutorial (beginner friendly)
- Eloquent Javascript (oook)
- Javascript Info (beginner friendly)