Setup & JavaScript Refresher

Week 1

EDU-JS-12W

Agenda

- Environment setup (Node + npm + REPL)
- Quick intro to Node.js & npm
- Core JavaScript essentials
- Functions & scope
- ES6+ syntax highlights
- Modules in Node
- Live demo: “Hello World” script
- Best practices (formatting, debugging, commits)
- Q&A & preview of Week 2

Environment Setup

- **Install Node.js LTS**
  - Download from nodejs.org → verify with bash
    - node -v
    - npm -v
- **Choosing a code editor**
  - ***VSCode recommended***
  - Extensions: ESLint, Prettier, GitLens
  - Integrated terminal & debugger

- **Using the Node REPL**

Try expressions:

```
> 2 + 3
5
> const msg = ‘Hello’
> msg.toUpperCase()
'HELLO'
```

Node.JS Introduction

- **What is Node.js?**
  - JavaScript runtime built on Chrome’s V8 engine
  - Enables JS on the server for I/O-bound apps.

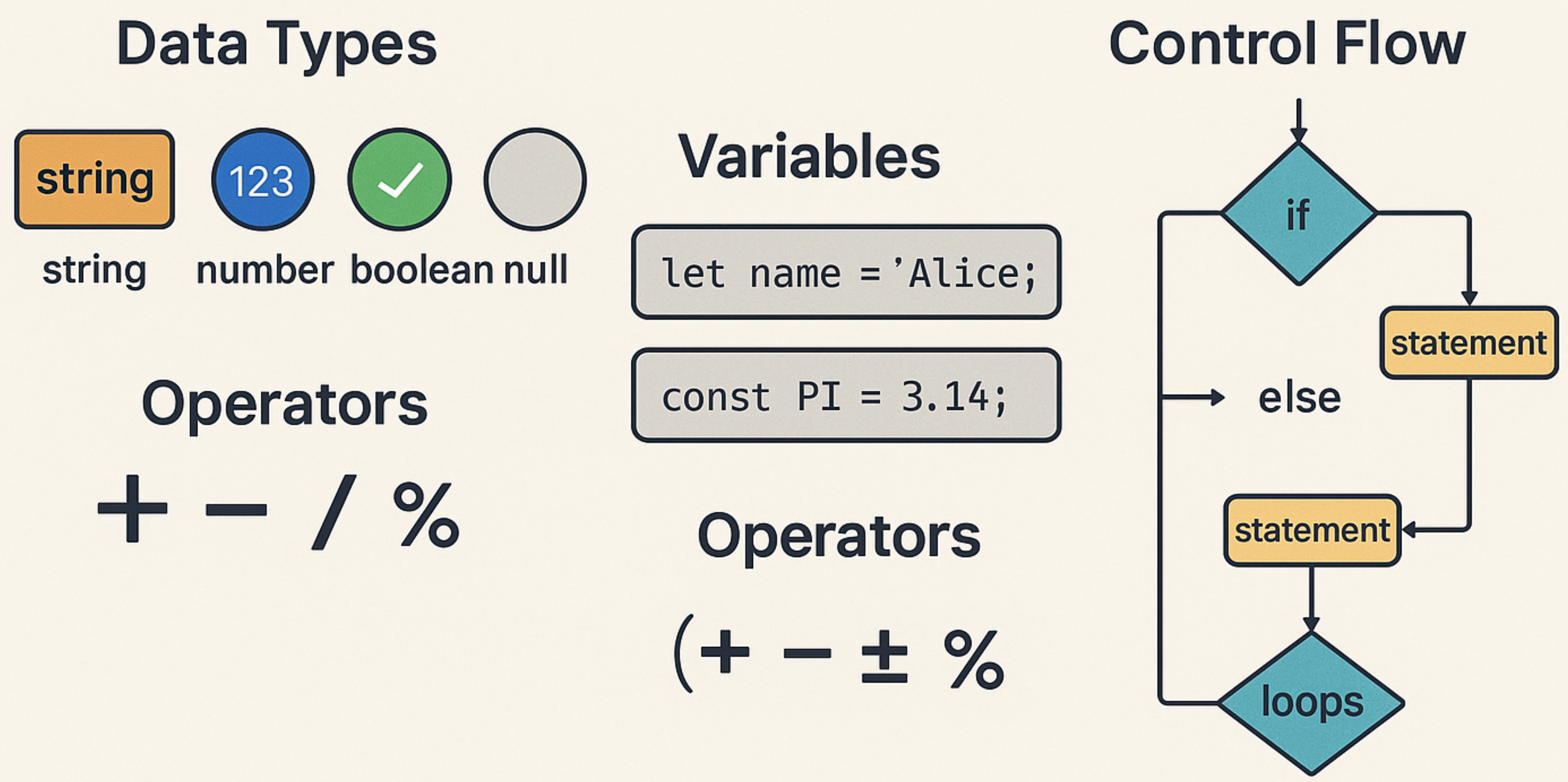
- **Node Package Manager(NPM)**
  - Ships with Node.js
  - Registry of >2 million packages
  - Examples:

```
> npm init   # create package.json
> npm install express --save   # add a dependency
> npm install eslint --save-dev # add a dev-tool
```

- **Why use npm?**
  - Manage dependencies & versioning
  - Share and reuse community code

Javascript Essentials

- **Data Types**
  - Primitives: string, number, boolean, null, undefined, symbol
  - Objects & Arrays
- **Operators & Control Flow**
  - Arithmetic: +, -, *, /, %
  - Comparison: ===, !==, <, >
  - Logical: &&, ||, !
  - if/else, switch, loops (for, while)
- **Variables**
  - let & const vs var (block scope vs function scope)



Functions/Methods

- **Traditional Function**

```
function add(a, b) {
  return a + b;
}
```

- **Arrow Function**

```
const add = (a, b) => a + b;
```

- **Scope & Hoisting**
  - Function vs block scope
  - var is hoisted, let/const are not

- ***this* Keyword**
  - In regular functions, context depends on call-site
  - In arrow functions, this is lexically bound

ES6+ Syntax

- **Template Literals**

```
const name = 'Alice';

console.log(`Hello, ${name}!`);
```

- **Destructuring**

```
const [first, second] = [10, 20];

const { title, author } = bookObj;
```

- **Spread Operator**

```
const arr2 = [...arr1, 4, 5];

const obj2 = { ...obj1, extra: true };
```

- **Classes**

```
class Person {

  constructor(name) { this.name = name; }

  greet() { console.log(`Hi, I’m ${this.name}`); }

}
```

Modules

- **CommonJS (Node default)**

```
// add.js
module.exports = (a, b) => a + b;
// index.js
const add = require('./add');
```

- **ES6 Modules**

```
// add.mjs
export const add = (a, b) => a + b;
// index.mjs
import { add } from './add.mjs';
```

- **Demo:**
- Split a small utility (e.g. math functions) into two files
- require vs import usage

Quick Demo

- **“Hello World” Script**

```
// hello.js

console.log('👋 Hello, Node!');

> node hello.js
```

- **Add-Two-Numbers Example**

1. *add.js: export an add(a,b) function*
2. *index.js: import and log add(2,3)*
3. *Run and confirm output*

Questions and Summary

- **Recap**
- Installed Node.js & npm, ran scripts & REPL
- Reviewed JS fundamentals & ES6+ features
- Explored modules and did a live Node demo

- **Next Week(Week 2)**
  - Deep dive into Node core modules (fs, http, events)
  - npm workflows, package.json, and scripting

Questions? …

REPO: https://github.com/TeoSlayer/edu-js-12w