# Introduction to Angular & Setup

## 1.1 What is Angular & Why It's Used?

🔷 **Angular** is a **TypeScript-based front-end framework** maintained by Google for building **single-page applications (SPAs)**.

✅ **Key Features:**

- **Component-based**: UI is split into reusable components.
- **Two-way Data Binding**: Syncs data between model (TypeScript) and view (HTML).
- **Dependency Injection**: Manages services and shared logic efficiently.
- **Directives**: Add dynamic behavior to HTML (`ngIf`, `ngFor`).
- **Pipes**: Transform data in templates (`date`, `currency`).
- **RxJS Support**: Reactive programming with Observables.
- **Routing**: Build multi-page navigation inside a single-page app.
- **Cross-Platform**: Build Web, Mobile (with Ionic), and Desktop apps.

✅ **Why it's used?**

- Enterprise-grade applications (Google, Microsoft, Upwork use it).
- Strong **community & Google support**.
- **Scalable** for large projects.
- **Efficient change detection** for faster apps.

📌 Example real-world use:

- **Gmail** is an Angular SPA → dynamic navigation, no full page reloads.

# 1.2 Install Node.js + Angular CLI

Angular requires **Node.js** (JavaScript runtime) and **npm** (Node package manager).

## Step 1: Install Node.js

1. Download from: https://nodejs.org
    - **LTS version (recommended)** (Long-Term Support, stable).
2. Install → it also installs **npm**.

🔍 Verify installation:

```
node -v
npm -v
```

---

## Step 2: Install Angular CLI (Command Line Interface)

Angular CLI is the official tool to create, build, test, and deploy Angular apps.

Run:

```
npm install -g @angular/cli
```

Verify:

```
ng version
```

You'll see Angular CLI + Node.js + npm versions.

---

# 1.3 Create First Angular Project

1. Navigate to the folder where you want the project:

```
cd Desktop
```

1. Create new project:

```
ng new my-app
```

1. CLI asks:
    - **Add Angular routing?** → Yes (for navigation support).
    - **Which stylesheet format?** → CSS (or SCSS/SASS/LESS if preferred).
2. Go inside project:

```
cd my-app
```

---

# 1.4 Angular Project Structure Explained

Inside `my-app/` you'll see:

```
my-app/
│
├── node_modules/      → All installed dependencies (don't touch directly)
├── src/               → Main application source code
│   ├── app/           → App logic (components, services, modules)
│   │   ├── app.component.ts   → Root component (TypeScript)
│   │   ├── app.component.html → Root template (HTML)
│   │   ├── app.component.css  → Styles
│   │   └── app.module.ts      → Root Angular module
│   │
│   ├── assets/        → Images, static files
│   ├── environments/  → Environment configs (dev/prod)
│   ├── index.html     → Main HTML file (single page)
```

```
│   ├── main.ts        → Application entry point
│   └── styles.css     → Global CSS styles
│
├── angular.json       → Angular project configuration
├── package.json       → Project dependencies & scripts
├── tsconfig.json      → TypeScript config
└── README.md          → Project info
```

📌 **How Angular loads?**

- `index.html` → loads `<app-root>`
- `<app-root>` comes from `app.component.html`
- Bootstrapped via `main.ts` → `AppModule` → `AppComponent`

---

# 1.5 Running & Building Angular Applications

## Run Development Server

```
ng serve
```

or

```
ng serve -o
```

(`-o` opens in browser automatically at `http://localhost:4200`)

Now, any code changes **auto-refresh** in browser (Hot Reload).

---

## Build for Production

```
ng build
```

- Creates `dist/` folder with optimized files (minified, tree-shaken).
- These files can be deployed to a server (Apache, Nginx, Firebase, etc.).

```
ng build --prod
```

(Optimized build for production, smaller and faster).