

Alpine.js Cheat Sheet

Essential Directives for Building Interactive Web Interfaces

1. x-data

Purpose: Defines reactive data for your Alpine.js component.

Example: `<div x-data="{ count: 0 }">`

Key Use: Initialize variables like counters or states.

2. x-model

Purpose: Two-way data binding for form inputs.

Example: `<input type="text" x-model="message">`

Key Use: Sync input values with your data in real-time.

3. x-for

Purpose: Loops over arrays to render lists dynamically.

Example: `<template x-for="item in items"><li x-text="item">
</template>`

Key Use: Display multiple items, like quiz questions.

4. x-on / @

Purpose: Listens for events and runs actions.

Example: `<button x-on:click="count++">Click me</button> or <button
@click="count++">Click me</button>`

Key Use: Handle clicks or other user interactions.

5. x-text

Purpose: Updates the text content of an element dynamically.

Example: `<span x-text="count"></span>`

Key Use: Show live updates, like a quiz score.

6. x-bind

Purpose: Dynamically binds attributes (e.g., class, style).

Example (Class with String): `<div x-bind:class="isActive ? 'active' : ''">
</div>`

Example (Class with Object): `<div x-bind:class="{ 'active': isActive }">
</div>`

Key Use: Styles or toggles classes based on conditions. String keeps existing classes; object overrides them.

7. x-show

Purpose: Toggles visibility based on a condition.

Example: `<div x-show="count > 0">Visible!</div>`

Key Use: Show/hide elements, like quiz feedback.

Quick Tip: Combine these directives to create lightweight, interactive components with minimal JavaScript!