Directives

- Directives and components hand-in-hand are the fundamental building blocks of any Angular app
- Directives are components without templates. Conversely, components are directives without templates
- Directives allow you to attach behavior to elements in the DOM
- A directive is defined using the `@directive` decorator
- There are two types of directives in Angular:
  - Structural
  - Attribute
- The `selector` attribute uses a css selector to find the element. However, parent-child relationship selectors are not supported
- You can use the following possible selectors:
  - `element`
  - `[attribute]`
  - `.classname`
  - `:not()`
  - `.some-class:not(div)`

Shadow DOM Basics

TODO (shadow dom, light dom, etc.)

Simple Directive

TODO (writing a custom directive)

Accessing Directives from Parents

TODO (access directives on parent elements)

Accessing Directives from Children

TODO (access directives on children and descendants)

Built-in Directives

Angular has a couple of useful built-in directives.

TODO(Note on directive names, docs and template usage)

`NgClass`

- `import {NgClass} from 'angular2/common';` , `directives: [NgClass]`
- Template Usage:

`<div class="button" [ngClass]="{active: isActive, disabled: !isActive}"` **Note** that we are using `ngClass` in the template, but not `NgClass`

`NgIf`

TODO

`NgSwitch`

TODO

`NgFor`

- Usage: `<ul> <li *ngFor="#item of items"> ... </li> </ul>`

TODO(Details)