



Why?



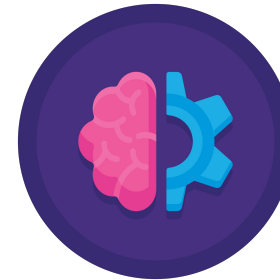
Cross platform



Speed and performance



Productivity

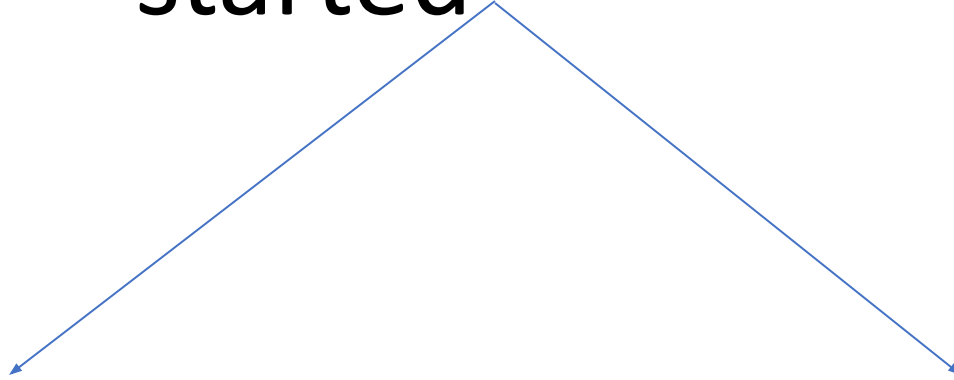


Full development process

Why?

- Modules.
- Components.
- One-way and two-way data bindings.
- Dependency injection.
- Routing.
- Pipes.
- etc.

Getting started



Webpack

Angular CLI

Getting started

- Install Node.js
- Install Angular-CLI globally
- `ng new ${project-name}`
- `ng serve`

App loading

- index.ts -> platformBrowserDynamic().bootstrapModule(AppModule)
- bootstrap: [AppComponent]

NgModule

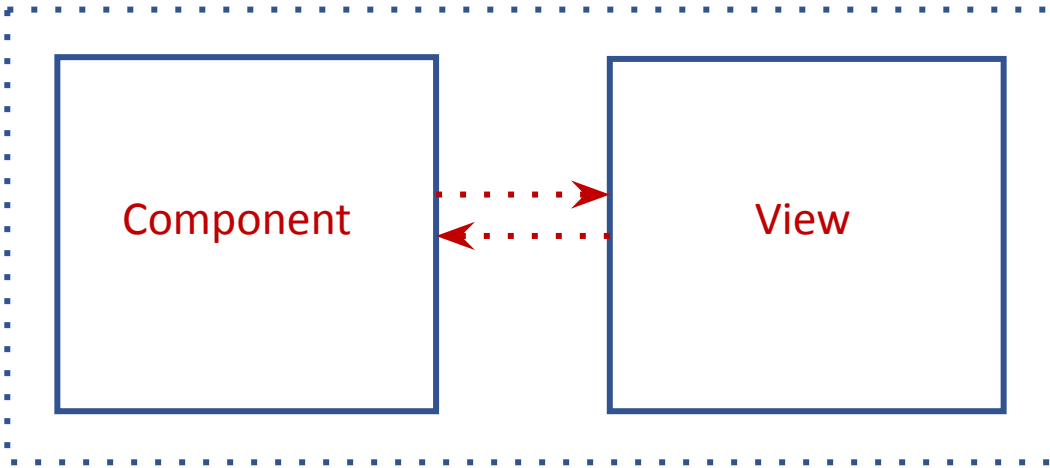
- Angular applications are modular.
- Angular has own modular system called ngModules.
- **@NgModule({})**

NgModule properties

- Declarations.
- Exports.
- Imports.
- Providers.
- Bootstrap.

What is component?

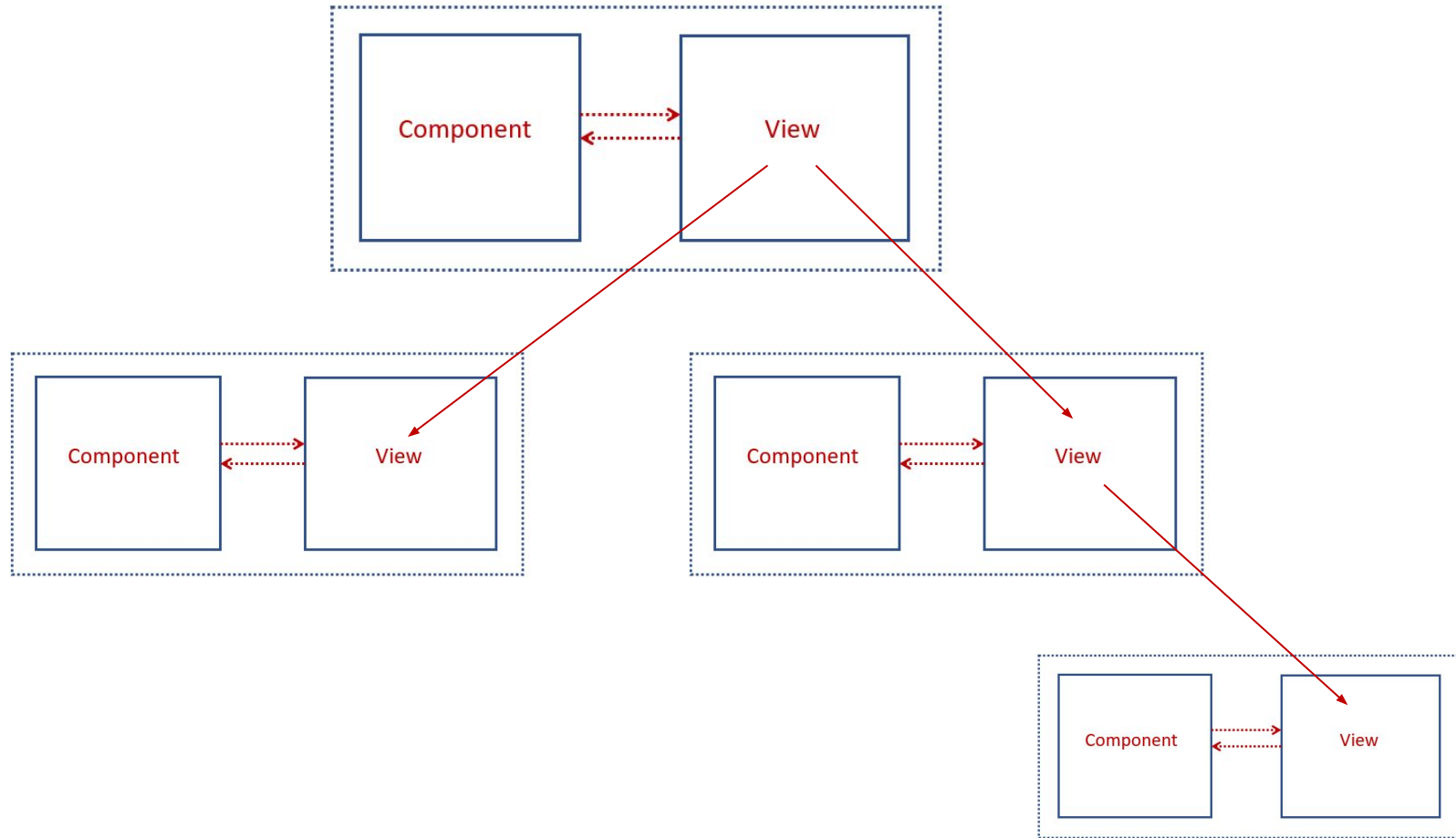
- A *component* controls a part of screen called a *view*.



Why?

- Reusable.
- Customizable.
- Independent from another components.
- Easy to understand.
- Develop faster.
- Etc.

Component structure in application



Getting started

- `ng generate component component-name`
- `ng g component directory/component-name`
- `<component-name></component-name>`

Component options

- selector
- template / templateUrl
- styles / styleUrls
- encapsulation
- **changeDetection**
- Etc.

Styles and template

- **styles**
- **template**
- **:host** selector
- **:host-context** selector

Data binding (One way / Two way)

- `<h1>{{title}}</h1>`
- `<p [title]="title">Lorem</p>`
- `<button (click)="save($event)">save</button>`
- `<input [(ngModel)]="title">`

Resources

- <https://nodejs.org/en/download/>
- <https://angular.io/cli>
- <https://github.com/angular/angular>
- <https://metanit.com/web/angular2/>
- <https://angular.io/tutorial>
- <https://webformyself.com/stili-v-angular/>
- <https://angular.io/guide/file-structure>
- <https://www.madewithangular.com/>