

# Chapter 7. Grasping Angular's philosophy

To write an Angular application, you have to grasp a few things on the framework's philosophy.

First and foremost, Angular is component-oriented. You will write tiny components and, together, they will constitute a whole application. A component is a group of HTML elements in a template, dedicated to a particular task. For this, you will usually also need to have some logic linked to that template, to populate data, and react to events for example. For the veterans of AngularJS 1.x, it's a bit like a 'template/controller' duo, or a directive.

It has to be said that a standard has been established around this component thing: the Web Component standard. Even if it's not completely supported by browsers yet, you can build small and isolated components, reusable in different applications - an old dream of computer programming. This component orientation is something that is becoming widely shared across front-end frameworks: [ReactJS](#), the latest cool kid from Facebook, has been doing it that way from the beginning; [EmberJS](#) and [AngularJS](#) have their way of doing something similar; and newcomers like [Aurelia](#) or [Vue.js](#) are betting on building small components too.

