# **Learn AngularJS With These 5 Practical Examples**

**https://tutorialzine.com/2013/08/learn-angularjs-5-examples**

By now you've probably heard of [AngularJS](http://angularjs.org/) - the exciting open source framework, developed by Google, that changes the way you think about web apps. There has been much written about it, but I have yet to find something that is written for developers who prefer quick and practical examples. This changes today. Below you will find the basic building blocks of Angular apps - Models, Views, Controllers, Services and Filters - explained in 5 practical examples that you can edit directly in your browser. If you prefer to open them up in your favorite code editor, grab the zip above.

## **What is AngularJS?**

On a high level, [AngularJS](http://angularjs.org/) is a framework that binds your HTML (views) to JavaScript objects (models). When your models change, the page updates automatically. The opposite is also true - a model, associated with a text field, is updated when the content of the field is changed. Angular handles all the glue code, so you don't have to update HTML manually or listen for events, like you do with jQuery. As a matter of fact, none of the examples here even include jQuery!

## **1. Navigation Menu**

As a first example, we will build a navigation menu that highlights the selected entry. The example uses only Angular's [directives](http://docs.angularjs.org/guide/directive), and is the simplest app possible using the framework. Click the "**Edit**" button to see the source code. It is ready for experimentation!

## **2. Inline Editor**

For the second example, we will create a simple inline editor - clicking a paragraph will show a tooltip with a text field. We will use a controller that will initialize the models and declare two methods for toggling the visibility of the tooltip.

Adding properties or functions to it makes them available to the view. Using the ng-model binding on the text field tells Angular to update that variable when the value of the field changes (this in turn re-renders the paragraph with the value).

## 3. Order Form

In this example, we will code an order form with a total price updated in real time, using another one of Angular's useful features - [filters](http://docs.angularjs.org/guide/dev_guide.templates.filters). Filters let you modify models and can be chained together using the pipe character |. In the example below, I am using the [currency filter](http://docs.angularjs.org/api/ng.filter:currency), to turn a number into a properly formatted price, complete with a dollar sign and cents. You can easily make your own filters, as you will see in example #4.

## 4. Instant Search

This example will allow users to filter a list of items by typing into a text field. This is another place where Angular shines, and is the perfect use case for writing a custom filter. To do this though, we first have to turn our application into a module.

Modules are a way of organizing JavaScript applications into self-contained components that can be combined in new and interesting ways. Angular relies on this technique for code isolation and requires that your application follows it before you can create a filter. There are only two things that you need to do to turn your app into a module:

1. Use the angular.module("name",[]) function call in your JS. This will instantiate and return a new module;
2. Pass the name of the module as the value of the ng-app directive.

Creating a filter then is as simple as calling the filter() method on the module object returned by angular.module("name", []).

## 5. Switchable Grid

Another popular UI interaction is switching between different layout modes (grid or list) with a click of a button. This is very easy to do in Angular. In addition, I will introduce another important concept - **Services**. They are objects that can be used by your application to communicate with a server, an API, or another data source. In our case, we will write a service that communicates with [Instagram's API](http://instagram.com/developer/) and returns an array with the most popular photos at the moment.

# 5 Practical Examples For Learning The React Framework

You've probably heard about the popular JavaScript framework by Facebook - [React](http://facebook.github.io/react/). It is used by many popular websites, including Facebook itself and Instagram. In this article you will find 5 practical examples that have been built with React, and which will give you a head start with the framework.

## What is special about React?

React is built around the concept of components. This is in contrast to frameworks like Angular and Ember, which use two-way data bindings to update the HTML of the page. In my opinion React is easier to learn than Angular and Ember - it is much smaller and plays nicely with jQuery and other frameworks. It is also extremely fast, because it uses a virtual DOM, and syncs only the changed parts with the underlying page (accessing the DOM is still the slowest part of a modern web application, which is why the framework gets a performance boost by optimizing it).

However, the flip side is that it takes a bit more code in React to achieve the same things that can be easily done with a data binding, as you can see from the examples below. For a comparison, see [our article about Angular](https://tutorialzine.com/2013/08/learn-angularjs-5-examples/).

## How to use it?

To use React, you need to include a single JavaScript file in your page. Facebook kindly provides a CDN that you can directly link to:

This gives you access to the global **React** object, which holds a number of useful methods, some of which you can see in these examples. The recommended way to write React web apps is by using a language called **JSX**. This is a slightly enhanced version of JavaScript that allows you to initialize React components by using an HTML-like syntax directly in your code. It is compiled down to JavaScript before being interpreted by the browser. JSX is completely optional - you can use regular JavaScript if you prefer.

But enough talk, let's see some code!

## **1. Timer**

As I mentioned, the building blocks of react apps are components. They are created by calling React.createClass() with an object of options and methods. Each component has state (an object with data) and each is in charge of their own rendering - the render() method is called whenever the state changes. Here is an example for building a simple timer:

(Click the Result link to see the example it in action)

You can embed any JavaScript expression within braces {} when creating a component. In this example, we pass a start date, that will be used in every tick() function call to calculate the elapsed seconds.

## **2. Navigation menu**

Let's see how we can handle click events in React by building a navigation menu:

You've probably noticed from the code of these examples that we use attributes like className that don't actually exist in HTML. This is because when you return a new <p> element you don't return HTML, but an instance of the component React.DOM.p. You can read more about this [here](http://facebook.github.io/react/docs/jsx-in-depth.html).

## **3. Real-time search**

We've learned time and time again that people hate to wait. Here is how you can use React to build a real-time search experience. (For a comparison, see our version [with AngularJS](https://tutorialzine.com/2013/08/learn-angularjs-5-examples/" \l "4-instant-search)).

## 4. Order form

The real power of React comes when you combine multiple components. This allows you to structure your code better and to introduce a separation of concerns. It will also make your code reusable across different parts of your web application.

## 5. Image app with AJAX

This example will demonstrate how you can combine react with jQuery, and how to load results via AJAX. Whereas frameworks like Angular have their custom approaches when doing AJAX, React lets you use whatever other libraries you are comfortable with (see the [Angular version here](https://tutorialzine.com/2013/08/learn-angularjs-5-examples/" \l "5-switchable-grid)).