

# How to Learn Angular



Today's  
Lesson  
Plan

# Class Helpers



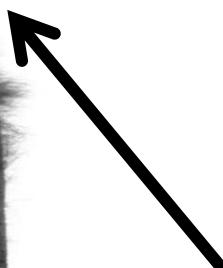
A black and white photograph of Tara Z. Manicsic. She is a woman with dark hair pulled back, wearing a dark blazer over a light-colored top. She is smiling broadly and holding a small, dark, fluffy dog, possibly a Pomeranian or similar breed, against her chest. The dog is looking slightly upwards and to the right.

# **TARA Z. MANICSIC**

**Developer Advocate for Progress  
Google Developer Expert / Web Technologies**

**Founder Cincy Women Who Code  
Founder & Director Cincy NodeSchool**

**<https://twitter.com/tzmanics>**



**#toshmagosh**

Learn  
Javascript

# Nodeschool

<https://github.com/workshopper/javascripting>





javascripting — bash — 80x30

pizza:javascripting sethvincent\$



# PLURALSIGHT

<https://www.pluralsight.com/courses/code-school-javascript-road-trip-part-1>



# Code School: JavaScript Road Trip Part 1

★★★★★ By Jason Millhouse

An introduction to the very basics of the JavaScript language.

Start a FREE 10-day trial

Course info

Description

We use cookies to make interactions with our websites and services easy and meaningful. For more information about the cookies we use or to find out how you can disable cookies, click here.

Disable cookies

JavaScript syntax and learn how to use values, variables, and files.

## The Cliffs of Value ^

🔒 Booleans, the True Path 6m

🔒 Strings on Signs 5m

## Variable Valley ^

🔒 Variable Discovery 6m

🔒 Variable Exploration 4m

## Files Falls ^

Accept cookies and close this message

# ECMAScript

<http://2ality.com/2018/02/ecmascript-2019.html>



# 2ality – JavaScript and more

About | Donate | Subscribe | Archive | Search | ReasonML | ES2018

## Feature watch: ECMAScript 2019

[2018-02-22] [dev](#), [javascript](#), [esnext](#), [es2019](#), [es proposal](#)

Check out my book (free online): "[Exploring ES2018 and ES2019](#)". Buy it now and get all future updates.

---

This blog post tracks actual and potential features of ES2019. I will continually update it.

### 1. A word on ECMAScript versions

Note that since [the TC39 process](#) was instituted, the importance of ECMAScript versions has much decreased. What really matters now is what stage a proposed feature is in: Once it has reached stage 4, it can be used safely. But even then, you still have to check if your engines of choice support it.

### 2. Proposed ECMAScript 2019 features

Recall that every ECMAScript feature proposal goes through stages:



Dr. Axel Rauschmayer  
Twitter



Free ES.next newsletter

Learn  
TypeScript

# In 5 Minutes

<https://www.typescriptlang.org/docs/handbook/typescript-in-5-minutes.html>



TypeScript 3.1 is now available. [Download](#) our latest version today!

Fork me on GitHub

 This site uses cookies for analytics, personalized content and ads. By continuing to browse this site, you agree to this use.

[Learn more](#)

Documentation

Tutorials



What's New



Handbook



Declaration Files



Project Configuration



# TypeScript in 5 minutes

Let's get started by building a simple web application with TypeScript.

## Installing TypeScript

There are two main ways to get the TypeScript tools:

- Via npm (the Node.js package manager)
- By installing TypeScript's Visual Studio plugins

Visual Studio 2017 and Visual Studio 2015 Update 3 include TypeScript by default. If you didn't install TypeScript with Visual Studio, you can still [download](#) it.

For NPM users:

```
> npm install -g typescript
```

## Building your first TypeScript file

In your editor, type the following JavaScript code in `greeter.ts`:

```
function greeter(person) {
```

# TS Blog Posts

<https://jameshenry.blog/>





## JAMES HENRY

*Microsoft Most Valuable Professional (MVP) for TypeScript*

Follow @MrJamesHenry 1,675 followers



**Don't just use TypeScript.  
Know it.**

[Learn FREE](#)

[typescriptcourses.com](http://typescriptcourses.com)

### NEVER MISS AN ARTICLE

Zero spam. Unsubscribe anytime.

jane.doe@example.com

[FREE ARTICLE UPDATES](#)

# TypeScript: Null and Undefined Types

BY JAMES HENRY ON OCTOBER 9, 2018

Trying to perform operations on `undefined` or `null` values is one of the most common causes of runtime errors in JavaScript code.

Sometimes the variables in our programs intentionally “hold” `undefined` or `null` - JavaScript is a very flexible language, and we can have legitimate reasons to want to use these values to express things in our code. Other times, we may inadvertently be interacting with the values `undefined` and `null` because we did not account for all the possible code paths, or we did not understand a third party or platform API well enough.

**JOKE**  **BREAK**

Learn about  
Components



# Angular Docs

<https://angular.io/guide/lifecycle-hooks>

## GETTING STARTED

## TUTORIAL

## FUNDAMENTALS

Architecture

## Components &amp; Templates

Displaying Data

Template Syntax

User Input

## Lifecycle Hooks

Component Interaction

Component Styles

Angular Elements

Dynamic Components

Attribute Directives

Structural Directives

Pipes

## Forms

## Observables &amp; RxJS

Bootstrapping

NgModules

Dependency Injection

# Lifecycle Hooks



A component has a lifecycle managed by Angular.

Angular creates it, renders it, creates and renders its children, checks it when its data-bound properties change, and destroys it before removing it from the DOM.

Angular offers lifecycle hooks that provide visibility into these key life moments and the ability to act when they occur.

A directive has the same set of lifecycle hooks.

## Component lifecycle hooks overview

Directive and component instances have a lifecycle as Angular creates, updates, and destroys them. Developers can tap into key moments in that lifecycle by implementing one or more of the *lifecycle hook* interfaces in the Angular core library.

Each interface has a single hook method whose name is the interface name prefixed with `ng`. For example, the `OnInit` interface has a hook method named `ngOnInit()` that Angular calls shortly after creating the component:

### peek-a-boo.component.ts (excerpt)

```
export class PeekABoo implements OnInit {  
  constructor(private logger: LoggerService) {}  
  
  // implement OnInit's `ngOnInit` method  
  ngOnInit() { this.logIt(`OnInit`); }  
  
  logIt(msg: string) {
```

- Lifecycle Hooks

Component lifecycle hooks overview

Lifecycle sequence

Interfaces are optional (technically)

Other Angular lifecycle hooks

Lifecycle examples

Peek-a-boo: all hooks

Spying `OnInit` and `OnDestroy`

`OnInit()`

`OnDestroy()`

`OnChanges()`

`DoCheck()`

AfterView

Abide by the unidirectional data flow rule

AfterContent

Content projection

AfterContent hooks

No unidirectional flow worries with

`AfterContent`

# Asim's Overview

<https://codecraft.tv/courses/angular/components/overview/>



[Quickstart](#)[ES6 JavaScript & TypeScript](#)[Angular CLI](#)[Components](#)[● Overview](#)[Architecting with Components](#)[Templates, Styles & View](#)[Encapsulation](#)[Content Projection](#)[Lifecycle Hooks](#)[ViewChildren & ContentChildren](#)[Wrapping Up](#)[Flash Cards](#)[Activity](#)[Angular 5](#) / [Components](#) / Overview

# Overview

AUTHOR:



Asim



Follow @jawache

9,171 followers



&lt; ACTIVITY

ARCHITECTING WITH COMPONENTS &gt;



Try DigitalOcean  
today and get a  
free \$100 credit.

ADS VIA CARBON

# Angular Elements

<https://scotch.io/tutorials/build-a-reusable-component-with-angular-elements>





SCOTCH

Starter  
Guides

Coding  
Courses

Written  
Posts

Really  
Cool Stuff



LOGIN

#react #angular #vue #node.js #javascript #laravel #php #css #api #authentication #editors #code-challenge #html5 #testing

# Build a Reusable Component with Angular Elements



Lee Brandt (@leebrandt) August 20 1 Comment 8,229 Views



## Build a Reusable Component with Angular Elements

# authentication

# angular

# javascript

Learn  
about  
State

# From Kent C. Dodds

<https://blog.kentcdodds.com/application-state-management-66de608ccb24>





**blog.kentcdodds.com**

Follow



11



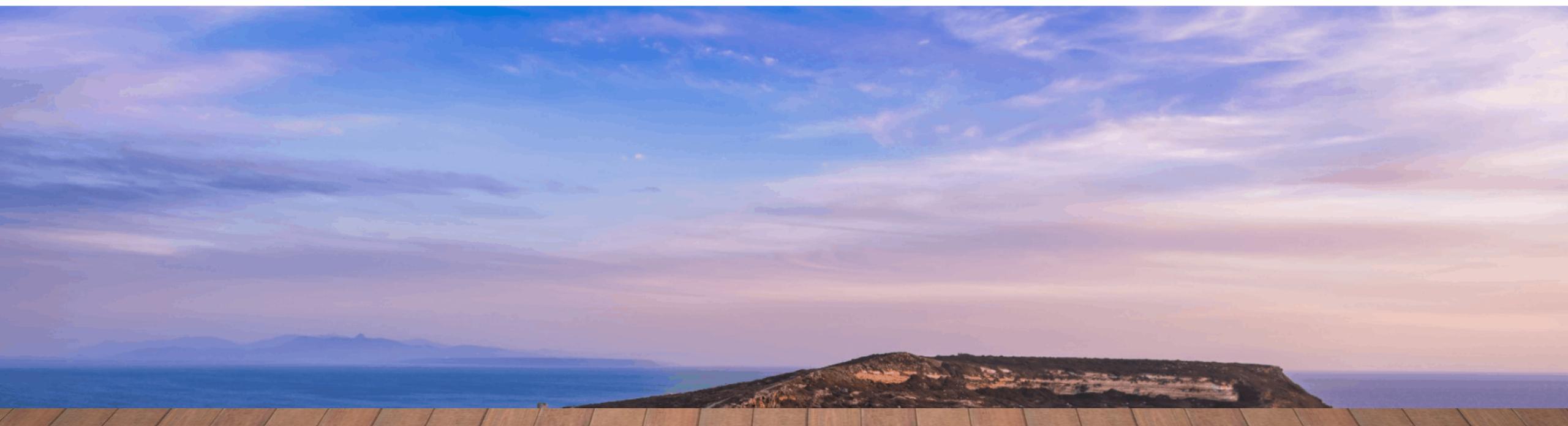
HOME SEE MORE POSTS | FIND KENT AROUND THE WEB



Kent C. Dodds [Follow](#)

Making software development more accessible · Husband, Father, Latter-day Saint, Teacher, OSS, GDE, @TC39 · @PayPalEng @eggheadio @FrontendMasters · #JS

Feb 26 · 7 min read



Learn about  
Observables  
& RxJS

# observable Posts

<https://netbasal.com/>



[Follow](#)

11

Netanel Basal [Follow](#)

I'm a Frontend Architect at Datorama, blogger, open source maintainer, creator of Akita and Spectator, Husband, Father and the Co-founder of HotJS

Jul 24 · 4 min read



# Who's Afraid of Observables?

Transform  
**FEAR**



# RxJS Docs

<http://reactivex.io/rxjs/manual/tutorial.html>

[Overview](#)[Installation](#)[Tutorial](#)[The basics](#)[Converting to observables](#)[Creating observables](#)[Controlling the flow](#)[Producing values](#)[Creating applications](#)[State stores](#)[Immutable JS](#)[React](#)[External References](#)[Tutorials](#)[Books](#)[Videos](#)[Presentations](#)[Reference](#)[Manual](#) » [Tutorial](#)

# The basics

## Converting to observables

```
// From one or multiple values
Rx.Observable.of('foo', 'bar');

// From array of values
Rx.Observable.from([1,2,3]);

// From an event
Rx.Observable.fromEvent(document.querySelector('button'), 'click');

// From a Promise
Rx.Observable.fromPromise(fetch('/users'));

// From a callback (last argument is a callback)
// fs.exists = (path, cb(exists))
var exists = Rx.Observable.bindCallback(fs.exists);
exists('file.txt').subscribe(exists => console.log('Does file exist?', exists));

// From a callback (last argument is a callback)
// fs.rename = (pathA, pathB, cb(err, result))
var rename = Rx.Observable.bindNodeCallback(fs.rename);
rename('file.txt', 'else.txt').subscribe(() => console.log('Renamed!'));
```

# Tracy Lee & Ben Lesh

<https://www.youtube.com/watch?v=lvMQybsdG98>





Effects

**JOKE**  **BREAK**

Learn  
Angular's  
Architecture

# StackBlitz

<https://stackblitz.com/>



PROJECT

INFO

Starter project for Angular apps that exports to the Angular CLI

0 0

FILES

- src
- app
  - app.component.css
  - app.component.html
  - app.component.ts**
  - app.module.ts
  - hello.component.ts
  - index.html
  - main.ts
  - polyfills.ts
  - styles.css
- angular.json
- package.json

DEPENDENCIES

## app.component.ts

```
1 import { Component } from '@angular/core';
2
3 @Component({
4   selector: 'my-app',
5   templateUrl: './app.component.html',
6   styleUrls: [ './app.component.css' ]
7 })
8 export class AppComponent {
9   name = 'Angular';
10 }
```



https://angular-agshvv.stackblitz.io



Starting dev server

Learn what  
the heck an  
NgModule is

# Many Posts

<https://medium.com/@cyrilletuzi/understanding-angular-modules-ngrouter-and-their-scopes-81e4ed6f7407>





Cyrille Tuzi

[Follow](#)

JavaScript and Angular trainer, PHP certified, @formationjs

Apr 7, 2017 · 5 min read

## Understanding Angular modules (NgModule) and their scopes



NgModule is the first basic structure you meet when coding an app with Angular, but it's also the most subtle and complex, because of **different scopes**. Angular documentation has done a whole [FAQ about NgModules](#), but it's still quite a mess to teach this during courses, students are often confused, so I decided to sum it up in this post for everyone.



# Docs, Of Course

<https://angular.io/guide/ngmodule-faq>

## GETTING STARTED

TUTORIAL



## FUNDAMENTALS



Architecture



Components &amp; Templates



Forms



Observables &amp; RxJS



Bootstrapping

## NgModules



NgModules Introduction

JS Modules vs NgModules

Frequently Used NgModules

Types of Feature Modules

Entry Components

Feature Modules

Providers

Singleton Services

Lazy Loading Feature Modules

Sharing NgModules

NgModule API

## NgModule FAQs



## NgModule FAQs

What classes should I add to the declarations array?

What is a *declarable*?

What classes should I *not* add to declarations?

Why list the same component in multiple NgModule properties?

What does "Can't bind to 'x' since it isn't a known property of 'y'" mean?

What should I import?

Should I import BrowserModule or CommonModule?

What if I import the same module twice?

What should I export?

What should I *not* export?

Can I re-export classes and modules?

What is the forRoot() method?

Why is a service provided in a feature module visible everywhere?

Why is a service provided in a lazy-loaded module visible only to that

## Prerequisites:

A basic understanding of the following concepts:

- [NgModules](#).

NgModules help organize an application into cohesive blocks of functionality.

This page answers the questions many developers ask about NgModule design and implementation.

## What classes should I add to the declarations array?

Add [declarable](#) classes—components, directives, and pipes—to a declarations list.

Declare these classes in *exactly one* module of the application. Declare them in a module if they belong to that particular module.

What is a *declarable*?

Declarables are the class types—components, directives, and pipes—that you can add to a module's `declarations` list. They're the only classes that you can add to `declarations`.

What classes should I *not* add to declarations?

Add only [declarable](#) classes to an NgModule's declarations list.

Learn the  
Template  
Syntax

# cheat Sheet!

<https://angular.io/guide/cheatsheet>



## Template syntax

```
<input [value]="firstName">
```

Binds property `value` to the result of expression `firstName`.

```
<div [attr.role]="myAriaRole">
```

Binds attribute `role` to the result of expression `myAriaRole`.

```
<div [class.extra-sparkle]="isDelightful">
```

Binds the presence of the CSS class `extra-sparkle` on the element to the truthiness of the expression `isDelightful`.

```
<div [style.width.px]="mySize">
```

Binds style property `width` to the result of expression `mySize` in pixels. Units are optional.

```
<button (click)="readRainbow($event)">
```

Calls method `readRainbow` when a click event is triggered on this button element (or its children) and passes in the event object.

```
<div title="Hello {{ponyName}}">
```

Binds a property to an interpolated string, for example, "Hello Seabiscuit". Equivalent to: `<div [title]="'Hello ' + ponyName">`

```
<p>Hello {{ponyName}}</p>
```

Binds text content to an interpolated string, for example, "Hello Seabiscuit".

# Alyssa's Helper ✩

<https://www.telerik.com/blogs/all-about-angular-bindings>



BLOGS

# All About Angular Bindings



by Alyssa Nicoll

October 17, 2018

Kendo UI, JavaScript, Developer Central

0 Comments



Angular bindings are a super-powerful mechanism built into the framework. Unfortunately, they can be difficult to

## TECHNOLOGIES

[.NET](#)[JavaScript](#)

## TOPICS

[Developer Central](#)[Web](#)

Learn about  
Dependency  
Injection

# DI Overview

<https://www.youtube.com/watch?v=IKD2-MAkXyQ>







# In-Depth

<https://auth0.com/blog/angular-2-series-part-2-domain-models-and-dependency-injection/>

## BLOG



## BLOG SERIES

# Angular 2 Series - Part 2: Domain Models and Dependency Injection

Learn about Angular 2's new Dependency Injection system and how to use models to organize your app.



Ryan Chenkie

September 17, 2015



**JOKE**  **BREAK**

Learn  
Services

# Tuts+ Tutorial

<https://code.tutsplus.com/tutorials/beginners-guide-to-angular-4-services--cms-29675>





# Beginner's Guide to Angular 4: Services

---

by Roy Agasthyan 23 Jan 2018

Difficulty: Beginner Length: Short Languages: English ▾

---

Angular JavaScript Web Development



This post is part of a series called [Beginner's Guide to Angular 4](#).

◀ [Beginner's Guide to Angular 4: Components](#)

▶ [Beginner's Guide to Angular 4: Routing](#)

In the [first part](#) of the beginner's guide to Angular tutorial series, you learnt what components are in Angular 4 and how to write Angular components.

Learn the  
Router

# Angular In-Depth

<https://blog.angularindepth.com/the-three-pillars-of-angular-routing-angular-router-series-introduction-fb34e4e8758e>



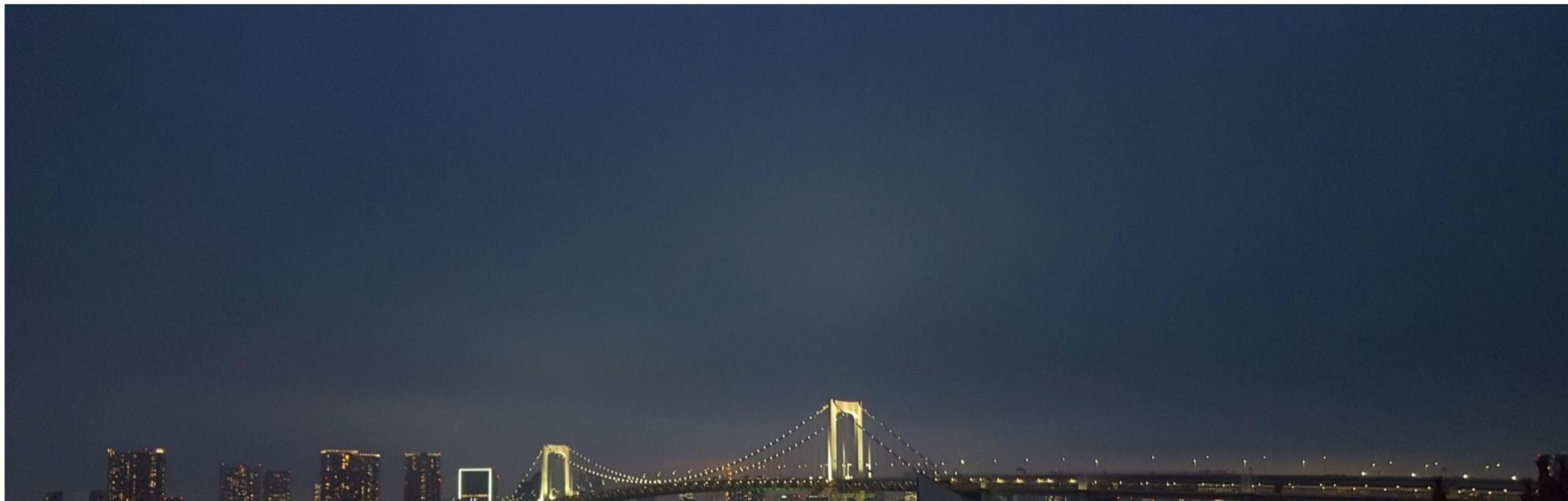


Nate Lapinski [Follow](#)

Fullstack Developer. Love digging into the internals of RxJS and Angular. Always trying to reach the next level. @nate\_lapinski on Twitter.

Sep 4 · 8 min read

# The Three Pillars of Angular Routing. Angular Router Series Introduction.



Learn How  
to Handle  
Data



# HttpClient Module

<https://blog.angular-university.io/angular-http/>



## Angular University

# Angular HTTP Client - Quickstart Guide

Last Updated: 16 JUNE 2018

Angular Core

This post will be a **quick practical guide** for the Angular HTTP Client module. We will cover how to do HTTP in Angular in general. We will be using the new `@angular/common/http` module, but a good part of this post is also applicable to the previous `@angular/http` module.



# Angular Forms

[https://www.youtube.com/watch?v=CD\\_t3m2WMM8](https://www.youtube.com/watch?v=CD_t3m2WMM8)





# Kara Erickson | **Angular**

## Angular Forms



Video sponsored by

Learn the

Angular

cli



# Just Do It!

<https://github.com/angular/angular-cli/wiki>

[Code](#)[Issues 1,241](#)[Pull requests 37](#)[Wiki](#)[Insights](#)

# Home

Hans edited this page on Aug 23 · 15 revisions

## Angular CLI

NOTE: this documentation is for Angular CLI 6. For Angular CLI 1.x go [here](#) instead.

### Overview

The Angular CLI is a tool to initialize, develop, scaffold and maintain [Angular](#) applications

### Getting Started

To install the Angular CLI:

```
npm install -g @angular/cli
```

Generating and serving an Angular project via a development server [Create](#) and [run](#) a new project:

```
ng new my-project  
cd my-project  
ng serve
```

[Pages 124](#)

- [Angular CLI](#)
- [Generate](#)
- [Stories](#)
- [Angular CLI 1.x wiki](#)

Clone this wiki locally

<https://github.com/angular/>

# Mike's Guide

<https://www.youtube.com/watch?v=nUrWugZCBNc>





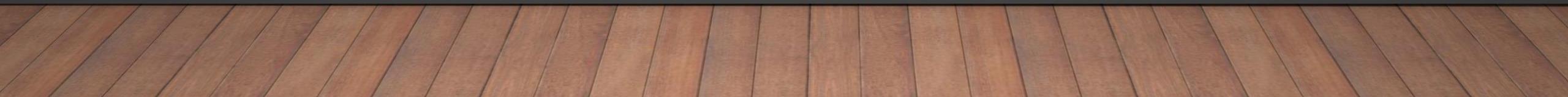
# NG-CRUISE

— PRESENTED BY NG-CONF —

**JOKE**  **BREAK**

**PHILLIPE PHILLOPE**

Learn  
Ngrx



# Posts from Nrwl

<https://blog.nrwl.io>





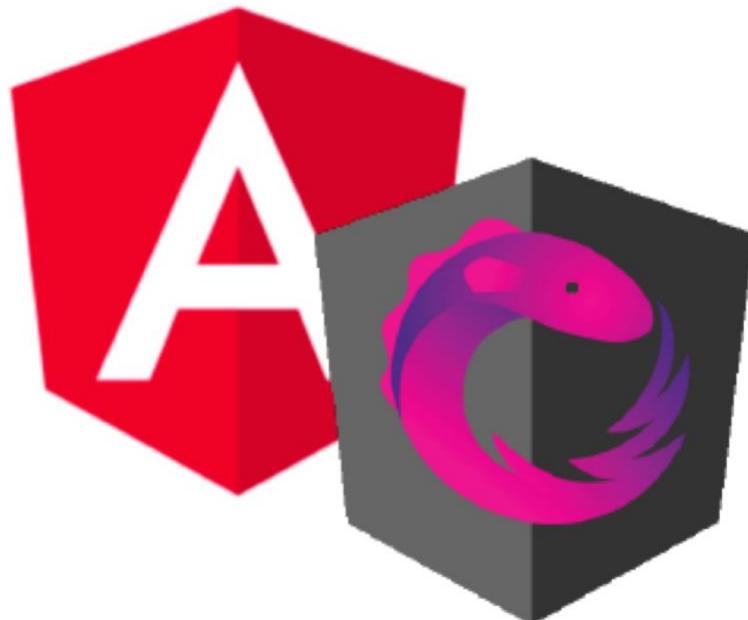
Victor Savkin [Follow](#)

Co-founder of Narwhal Technologies ([nrwl.io](#)), where we provide Angular consulting to large teams who want to get their applications to production quickly.

Jul 13, 2017 · 10 min read

# NgRx: Patterns and Techniques

Angular  
& NgRx



*Victor Savkin is a co-founder of [nrwl.io](#), providing Angular consulting to enterprise teams. He was previously on the Angular core team at Google, and*



# Ultimate Angular

<https://ultimateangular.com/ngrx-store-effects>

## The Ultimate Guide To Mastering Angular



Play featured video

### ▶ NGRX Store + Effects

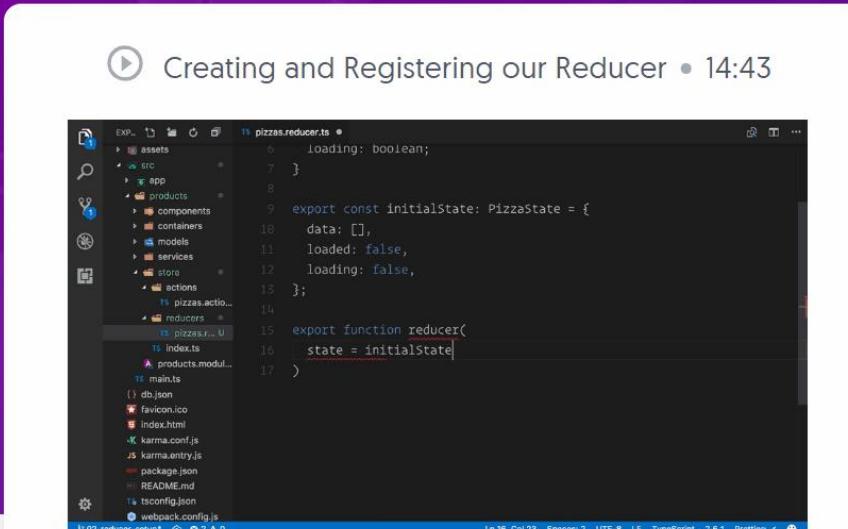
Master reactive and highly performant Redux state management for Angular apps.



ngrx  
v6.x



Duration  
41 lessons, 7 hours



A screenshot of a code editor showing a file named 'pizzas.reducer.ts'. The code defines a reducer function that takes a state object and an action. The state is initialized with a boolean 'loading' set to false, an array 'data' set to an empty array, and a boolean 'loaded' set to false. The reducer function returns the state with the 'loading' value updated based on the action type. The code editor interface shows a file tree on the left and various status bars at the bottom.

```
Creating and Registering our Reducer • 14:43
export const initialState: PizzaState = {
  loading: false,
  data: [],
  loaded: false
};

export function reducer(state = initialState, action: Action): PizzaState {
  switch (action.type) {
    case PIZZASActionTypes.LOADING:
      return { ...state, loading: true };
    case PIZZASActionTypes.LOADED:
      return { ...state, data: action.payload, loading: false, loaded: true };
    default:
      return state;
  }
}
```



Todd Motto  
Google Developer Expert

# Human Resources

# Angular Conf

<https://www.ng-conf.org/>



May 1st–3rd 2019

NG-CONF 2018 BLOG SPONSORS ORGANIZERS MORE



# WORLD'S ORIGINAL ANGULAR CONFERENCE

Save the Date! May 1st–3rd 2019, Salt Lake City, UT

STAY UP TO DATE & SIGN UP TO WIN A FREE TICKET

Enter Your Email

GET UPDATES

# Angular Podcasts

<https://angularair.com/>

<https://devchat.tv/adv-in-angular/>



# AngularAir

8,010 subscribers

SUBSCRIBE 8K

HOME

VIDEOS

PLAYLISTS

CHANNELS

DISCUSSION

ABOUT



Angular Air Episode 35: Angu...

10,151 views • 3 years ago

If you know React, you probably really love it. If you know Angular 2, you probably love it as well. We're going to talk with people who have experience with both and plan to choose Angular 2 over

[READ MORE](#)

FEATURED CHANNELS



AngularShow

[SUBSCRIBE](#)



AngularClass

[SUBSCRIBE](#)



Angular

[SUBSCRIBE](#)

[PODCASTS](#)[EVENTS & CONFERENCES](#)[ABOUT](#)[PATREON \(DONATIONS\)](#)[STORE](#)[Home](#) > 2016

## Adventures in Angular Episodes



### Adventures in Angular



AiA 211: "Azure Pipelines" with Ed Thomson LIVE at Microsoft Ignite

**Eric Begay** - October 16, 2018

0

Panel: Charles Max Wood Special Guests: Ed Thomson In this episode, the Adventures in Angular panel talks with Ed Thomson who is a Program Manager at...

AiA 210: "Zone.js" with Jia Li

**Eric Begay** - October 9, 2018

0

Panel: Joe Eames Aaron Frost John Papa Special Guests: Jia Li In this episode, the Adventures in Angular panel talks with Jia Li about Zones.js....

AiA 209: "Azure DevOps" with Donovan Brown Live at Microsoft Ignite

Suggest a Topic or Guest

**JOKE**  **BREAK**



# **You're Welcome TARA Z. MANICSIC**

**Developer Advocate for Progress  
Google Developer Expert / Web Technologies**

**Founder Cincy Women Who Code  
Founder & Director Cincy NodeSchool**

**<https://twitter.com/tzmanics>**