

Angularで始める Web開発体験

2019/06/28 B1セミナールーム
運輸システム事業部 佐々木 朝弘

スケジュール

タイムスケジュール	内容
13:30~14:00	Angularについて説明
14:00~15:00	ハンズオン
15:00~16:00	休憩
16:00~17:00	ハンズオン
17:00~17:30	クロージング

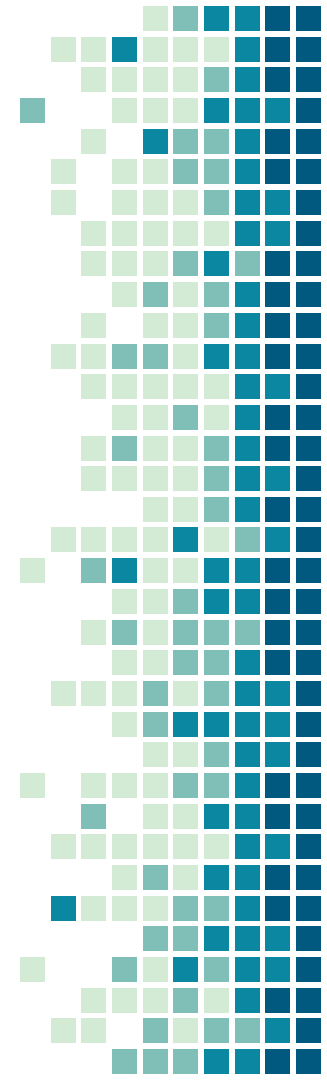


HELLO!

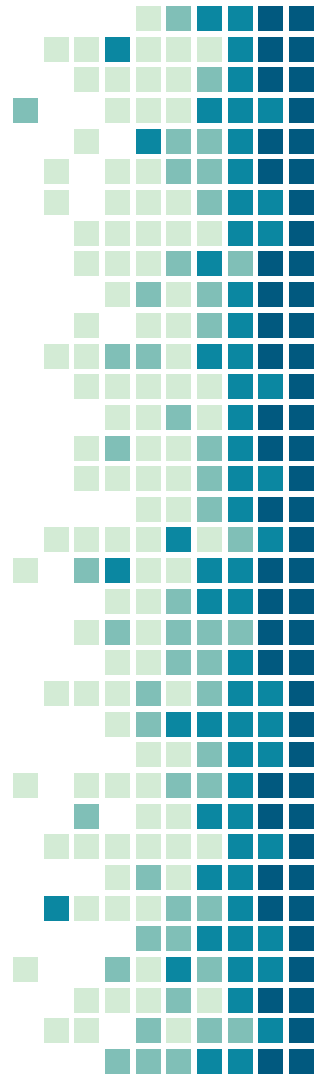
佐々木 朝弘

所属：運輸システム事業部

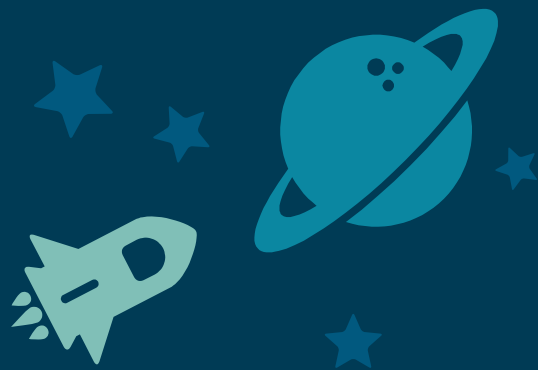
開発歴：Angular歴1年



目的：
最新Webフレームワークに触れ
てなんでもできる事を体験する



デモ：
今日作るアプリのデモをします



環境準備

まえがき

- ▶ VSCodeのターミナルを操作する場合は
- ▶ エディタでコードを修正する場合は

コマンド

エディタ

環境準備 (#1)

アプリケーションインストール ※以下、全て最新バージョンをインストールください

- ▶ Visual Studio Code

- ▶ <https://code.visualstudio.com/>

- ▶ Node.js

- ▶ <https://nodejs.org>

- ▶ Npm

- ▶ ※ Node.js (上記) をインストールすると一緒に npm もインストールされます

環境準備 (#2)

コマンド

アプリケーションインストール ※以下、全て最新バージョンをインストールください

- ▶ Angular CLI

以下のコマンドラインにてインストールします。

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

※ 詳細は以下のページを参照ください。

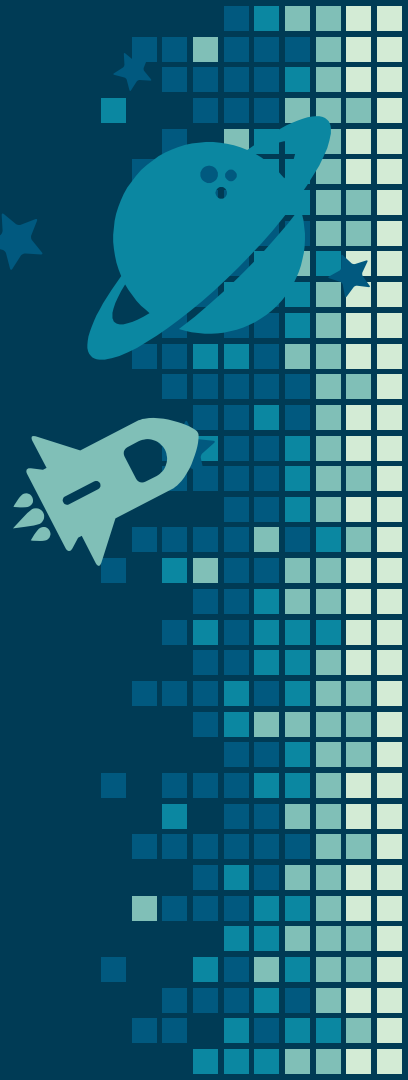
<https://www.npmjs.com/package/@angular/cli>

- ▶ Debugger for Chrome

- ▶ <https://marketplace.visualstudio.com/items?itemName=msjsdiag.debugger-for-chrome>

TypeScript

- Microsoft によって開発されたプログラミング言語でOSS。
- JavaScript のスーパーセットであり、JavaScript と互換性を持つ。
- TypeScript は、コンパイルすると JavaScript に変換される。
- オブジェクト指向、静的型付けなどの特徴がある。
- C# や Java の文法に似ている。





Angular

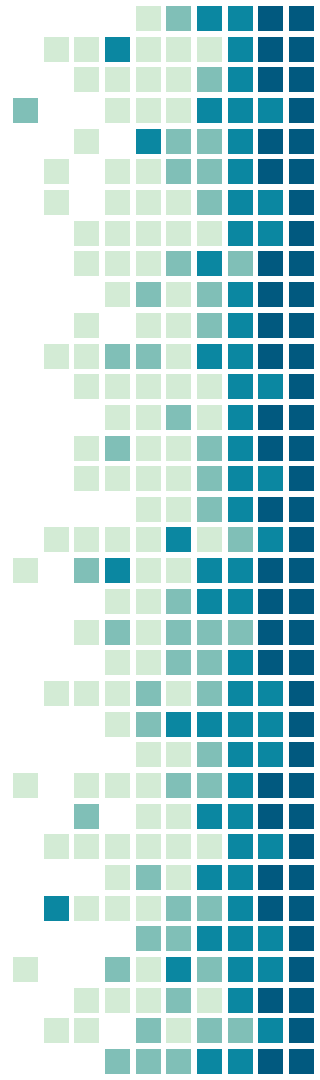
AngularはGoogle社が中心となって開発した
TypeScriptで書かれたOSSのWebアプリケーション
フレームワーク

- コンポーネント指向
- クライアントフルスタックフレームワーク



コンポーネント指向

- コンポーネントにビュー (UI 要素)、ロジック、メタ情報が含まれたまとまり。コンポーネントを組み合わせる画面を構成する



クライアント フルスタックフレームワーク

- アプリケーション開発時に必要となる機能全般が提供されている。
 - テンプレートエンジン
 - 依存性注入
 - フィルター
 - アニメーション
 - ルーティング
 - テスト

時間もないのでそれでは開始します

アプリケーションの作成

- フォルダを作成し、ファイルエクスプローラーで Visual Studio Code を開く。
- Visual Studio Codeでターミナルを起動する。
 - メニューバーにある「ターミナル」→「新しいターミナル」を選択する

アプリケーションの作成

- ターミナルから次のコマンドを実行
 1. アプリケーション作成
 - `ng new my-app1 --style=scss`
 - ? Would you like to add Angular routing?
(y/N)⇒y
 2. アプリケーションルートに移動
 - `cd my-app1`
 3. アプリケーション実行
 - `ng serve -o`

アプリケーションの実行結果



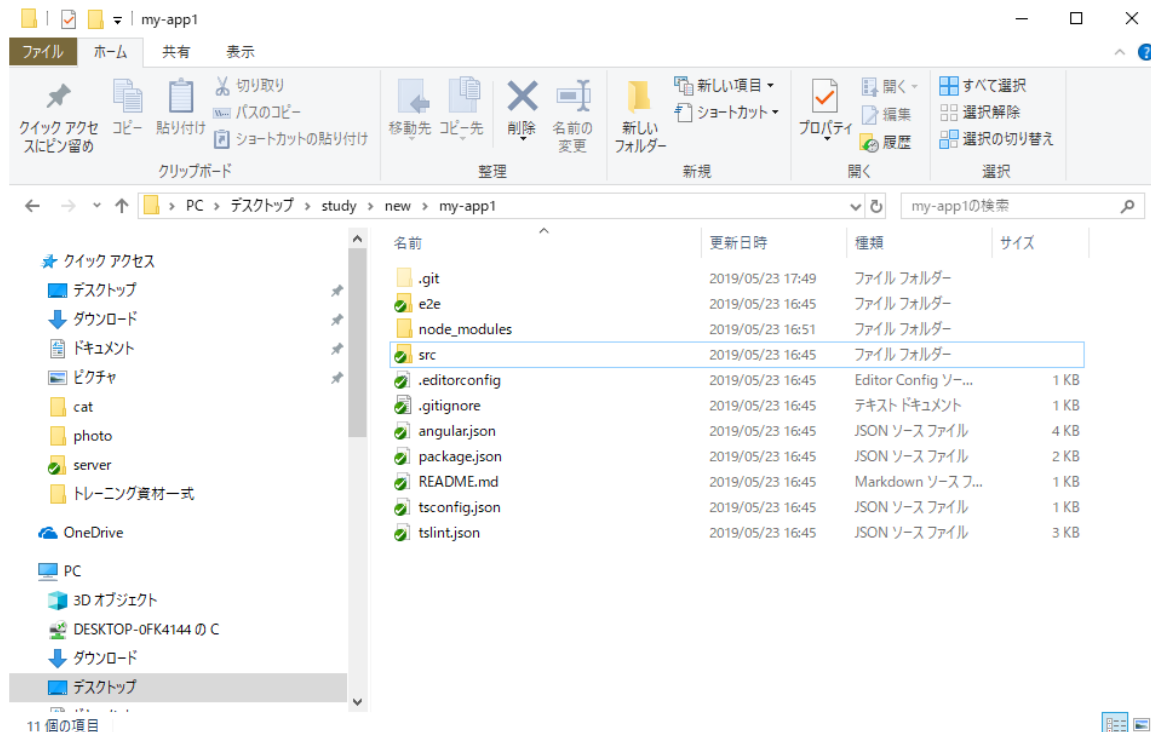
Welcome to my-app1!



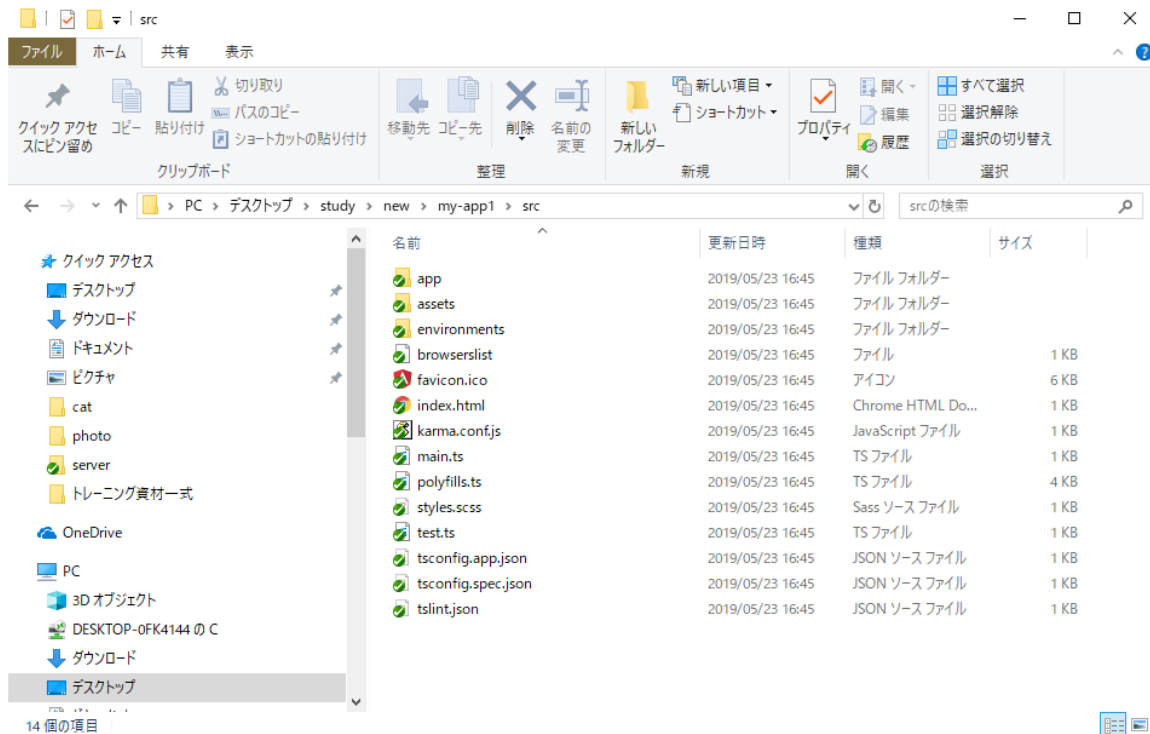
Here are some links to help you start:

- [Tour of Heroes](#)
- [CLI Documentation](#)
- [Angular blog](#)

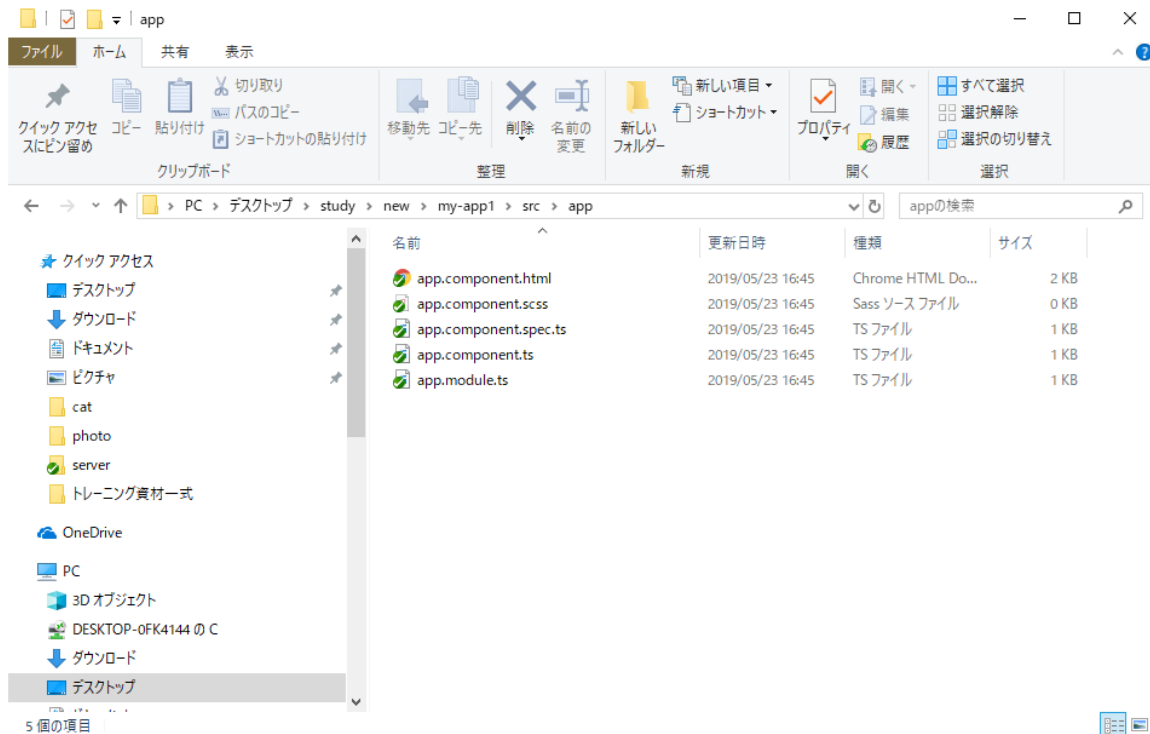
アプリケーションのフォルダ、 ファイル構成の理解



アプリケーションのフォルダ、ファイル構成の理解



アプリケーションのフォルダ、 ファイル構成の理解

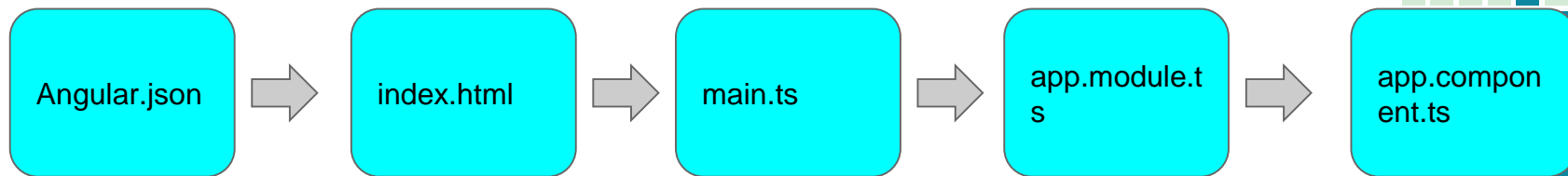


デバッグ実行

1. ターミナルでng serve
2. App.component.ts にブレイクポイントを設定
3. 「デバッグの開始」 ボタンをクリック (もしくは F5 を押下)

アプリケーションの流れ

- アプリケーション実行の流れ



参考：Angularの基本構造を理解して、アプリ開発を始めるには？
<https://www.buildinsider.net/web/angulartips/002>

ファイルの説明 1

- angular.json
 - Angular プロジェクトの各種設定が記述されているファイル。アプリケーションのエントリーポイントの指定がなされている。
- index.html
 - app.module に含まれている app.component を含んでいる。
- main.ts
 - アプリケーション起動に必要な機能のインポートを行う。また、アプリケーション起動時のモジュールを指定する。

ファイルの説明 2

- `app.module.ts` ルートモジュール。
 - `main.ts` にてアプリケーション起動時に読み込まれる。モジュール内で利用するコンポーネントを登録する。
- `app.component.ts`
 - 画面上に表示される UI (HTML) や、UI に紐づくロジックが含まれる。

枠を作る

- プロジェクト作成
 - `ng new costcalc --style=scss`
- ディレクトリ移動
 - `cd src/app`
- モジュール作成
 - `ng g m costcalc --routing`
- ディレクトリ移動
 - `cd costcalc`
- コンポーネント作成
 - `ng g c host`

ルーティングをする 1

- app-routing.module.ts
 - 以下に修正

```
const routes: Routes = [  
  { path: '', redirectTo: 'costcalc', pathMatch: 'full' },  
  { path: 'costcalc', loadChildren: './costcalc/costcalc.module#CostcalcModule' }  
];
```

ルーティングをする 2

- costcalc-routing.module.ts
 - 以下に追加及び修正

```
import { HostComponent } from '../host/host.component';  
const routes: Routes = [  
  {  
    path: '',  
    children: [  
      { path: '', redirectTo: 'host' },  
      { path: '', redirectTo: '', pathMatch: 'prefix' },  
      { path: 'host', component: HostComponent }  
    ]  
  }  
];
```

Angular Materialのインストール

- Angular Materialのサイト
 - <https://material.angular.io/>
- Get startedを参考にインストールを行う
 - `npm install --save @angular/material @angular/cdk @angular/animations`
 - `ng add @angular/material`
 - テーマとかhammer.jsのインストールを聞かれるからテーマとhammer.jsをYes

モジュールの設定

- myapp/src/app/costcalc/costcalc.module.ts
 - importとimports を書く

ここからが本格的なコーディング

- Material Designのコンポーネントサイトを見ながらデザインしていく
 - <https://material.angular.io/components/categories>

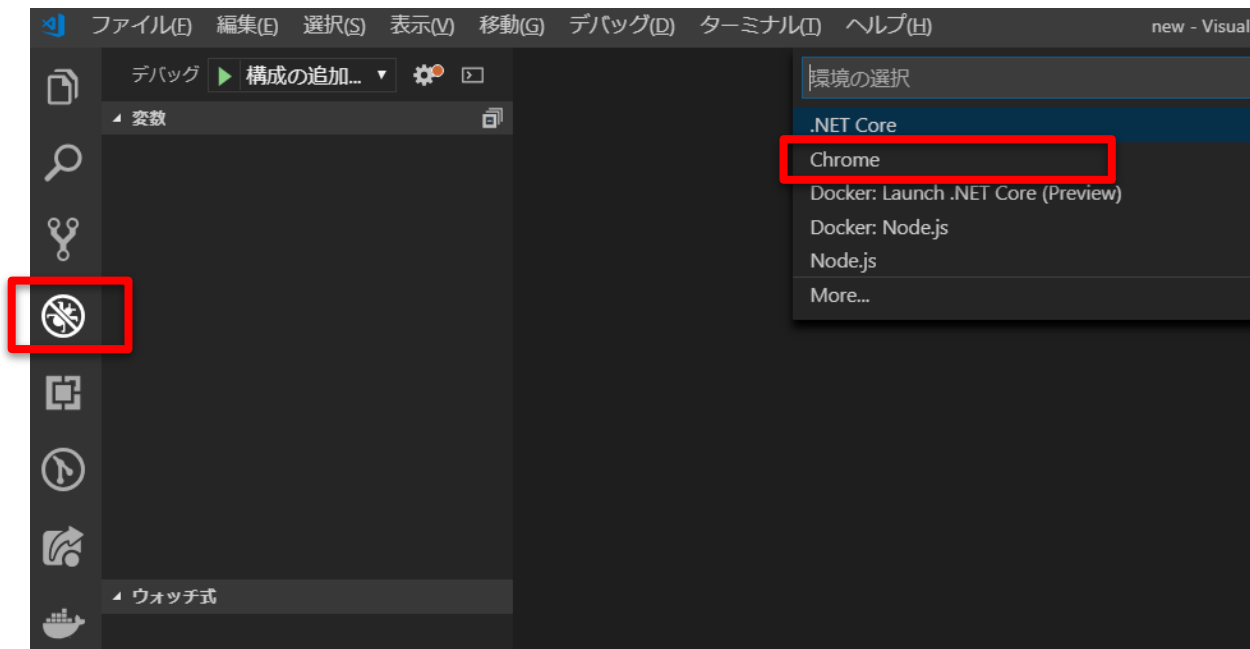
PWA

- PWA
 - <https://angular.jp/guide/service-worker-getting-started>

PWAはWebアプリケーションをネイティブアプリのように動作させる仕組み

デバック設定

- Visual Studio Code にて下記メニューを選択



デバック設定

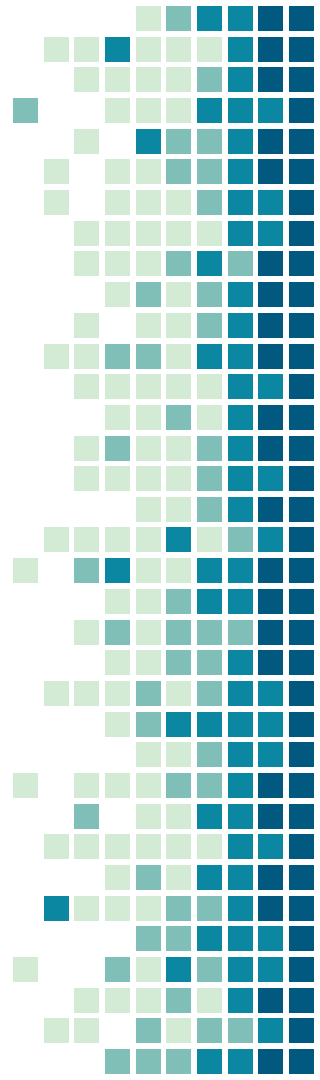
- Visual Studio Code にて下記メニューを選択

```
launch.json x
1  {
2    // IntelliSense を使用して利用可能な属性を学べます。
3    // 既存の属性の説明をホバーして表示します。
4    // 詳細情報は次を確認してください: https://go.microsoft.com/fwlink/?linkid=830387
5    "version": "0.2.0",
6    "configurations": [
7      {
8        "type": "chrome",
9        "request": "launch",
10       "name": "Launch Chrome against localhost",
11       "url": "http://localhost:4200",
12       "webRoot": "${workspaceFolder}"
13     }
14   ]
15 }
```

urlのport番号
を8080から
4200に変更し
ます

チュートリアル

- Angular公式チュートリアル
 - <https://angular.jp/tutorial>
- Angular Materialデザイン
 - <https://material.angular.io/>



THANKS!

Any questions?

質問など思い出したらメールください

次回告知

次回もやる予定です。

- PWAとFirebase（NoSQL）
7月
末予定
- Cloud画像認識
9月
末予定
- AdobeXD（プロトタイプ作成）
10月
末予定
- ハッカソンで何か作る！
11月
末予定

アンケート

アンケートに協力お願いします。

<https://forms.gle/K1A8ZK3GakTi55h67>

