FlexLayout - @angular/flex-layout

- Stand alone library
- No CSS needed – simple layout manipulation on your code
- TypeScript implementation
- Inline CSS is dynamically injected
- Static API
- Responsive API
- Indept of Angular Material
- Angular CLI integration

You can have horizontal and vertical layout –
You should try to use **horizontal layout** as much as possible
as **vertical layout** means you need to know heights.

Angular **Flex Layout** provides a stylish **layout API using Flexbox CSS + mediaQuery**.
This module provides Angular (v4.1 and higher) developers with component layout features using a
custom Layout API, mediaQuery observables + injected **DOM flexbox-2016 CSS** stylings.

The Flex Layout engine intelligently automates the process of applying appropriate Flexbox CSS to browser view hierarchies. This automation also addresses many of the complexities and workarounds encountered with the traditional, manual, CSS-only application of box CSS.

The **real** power of Flex Layout, however, is its **responsive** engine.
The **Responsive API** enables developers to easily specify
different layouts, sizing, visibilities for **different viewport sizes and display devices.**

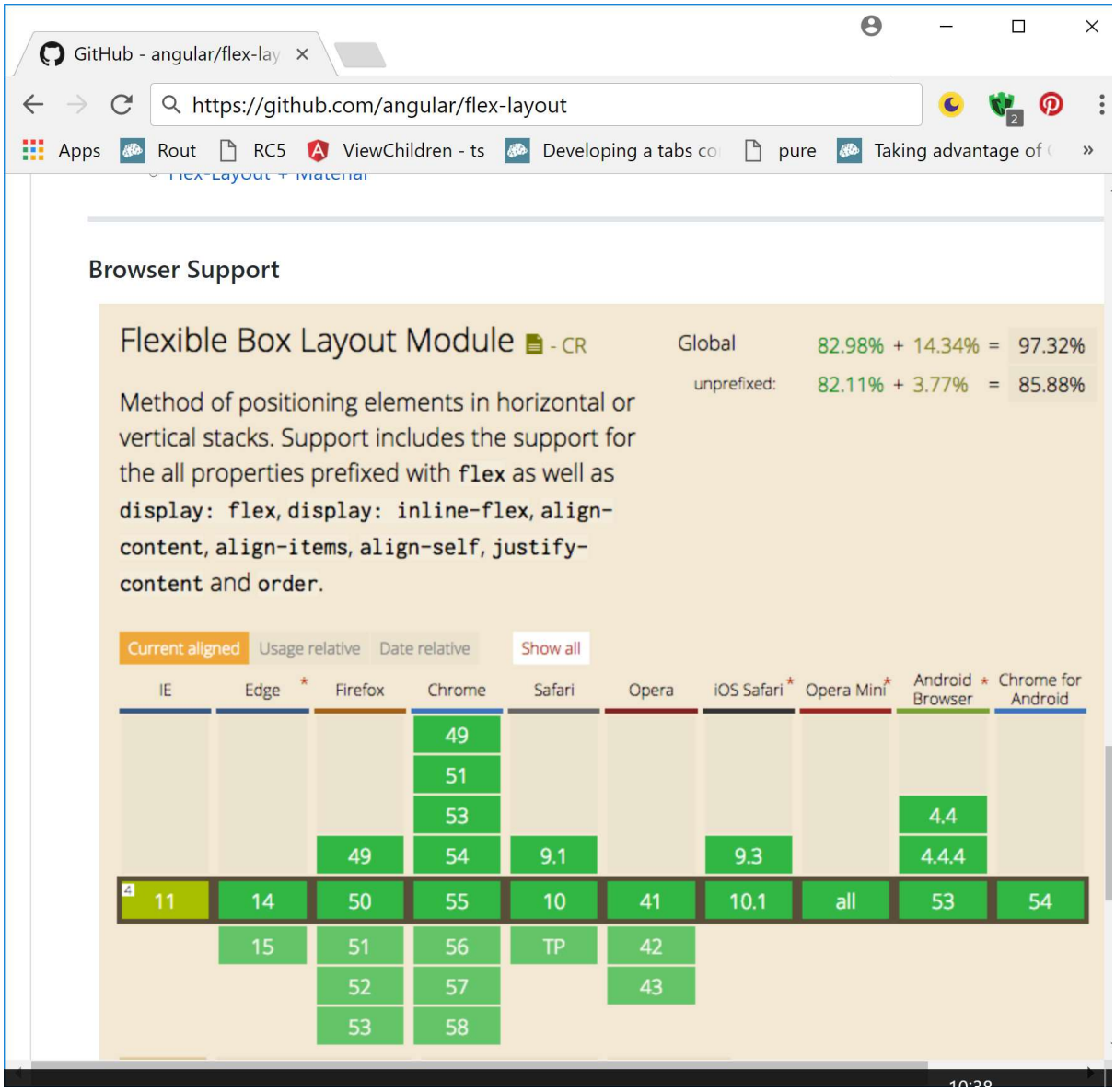
Examples of usage

Just say you have a list of 5000 objects but you want several objects per row, dependent on screen size, mobile device (iPhone 6x, LGx) its orientation (landscape or portrait), iPad etc.

With CSS this would be complex and time consuming to say the least
With Flex Layout this is relatively simple to give you a responsive nice looking layout
which will have the columns the same width for all rows in your view
and this will change dynamically as you vary the screen width for you.

https://github.com/angular/flex-layout

Browser-support is pretty much universal



fxLayout options	
Value	Equivalent CSS
Default	{flex-direction: row}
Row	{flex-direction: row}
Row-reverse	{flex-direction: row:reverse}
column	{flex-direction: column}
Column-reverse	{flex-direction: column:reverse}

Using Angular Flex-Layout

npm install –save @angular/flex-layout

```
import { FlexLayoutModule } from '@angular/flex-layout';

@NgModule({
        imports: [ FlexLayoutModule ]
})
export class AppModule{}
```

The nice thing is that after this initial setup – **everything is now in the HTML files**

- All your Flex manipulation is in the HTML files
- No CSS is required – though you can of course use it for more complex cases

fxFlex API

fxFlex="<flex-grow>" – defines how an el grows wrt other els – if there is available space
fxFlex="<flex-shrink>" – defines how an el shrinks wrt other els
fxFlex="<flex-basis>" – controls the default size of an element

fxFlex options

fxFlex-basis can be pixels| percentage| calcs | em| vw| vh| known aliases
values for fxFlex can be "2 2 calc(10em + 10px)" | "102px" | "auto"

fxFlex Alias	Equivalent CSS
Grow	{flex : 1 1 100%}
Initial	{flex : 0 1 auto}
Auto	{flex:<grow> <shrink> 100% }
None	{flex : 0 0 auto}
Nogrow	{flex : 0 1 auto}
Noshrink	{flex : 1 0 auto}

Static API

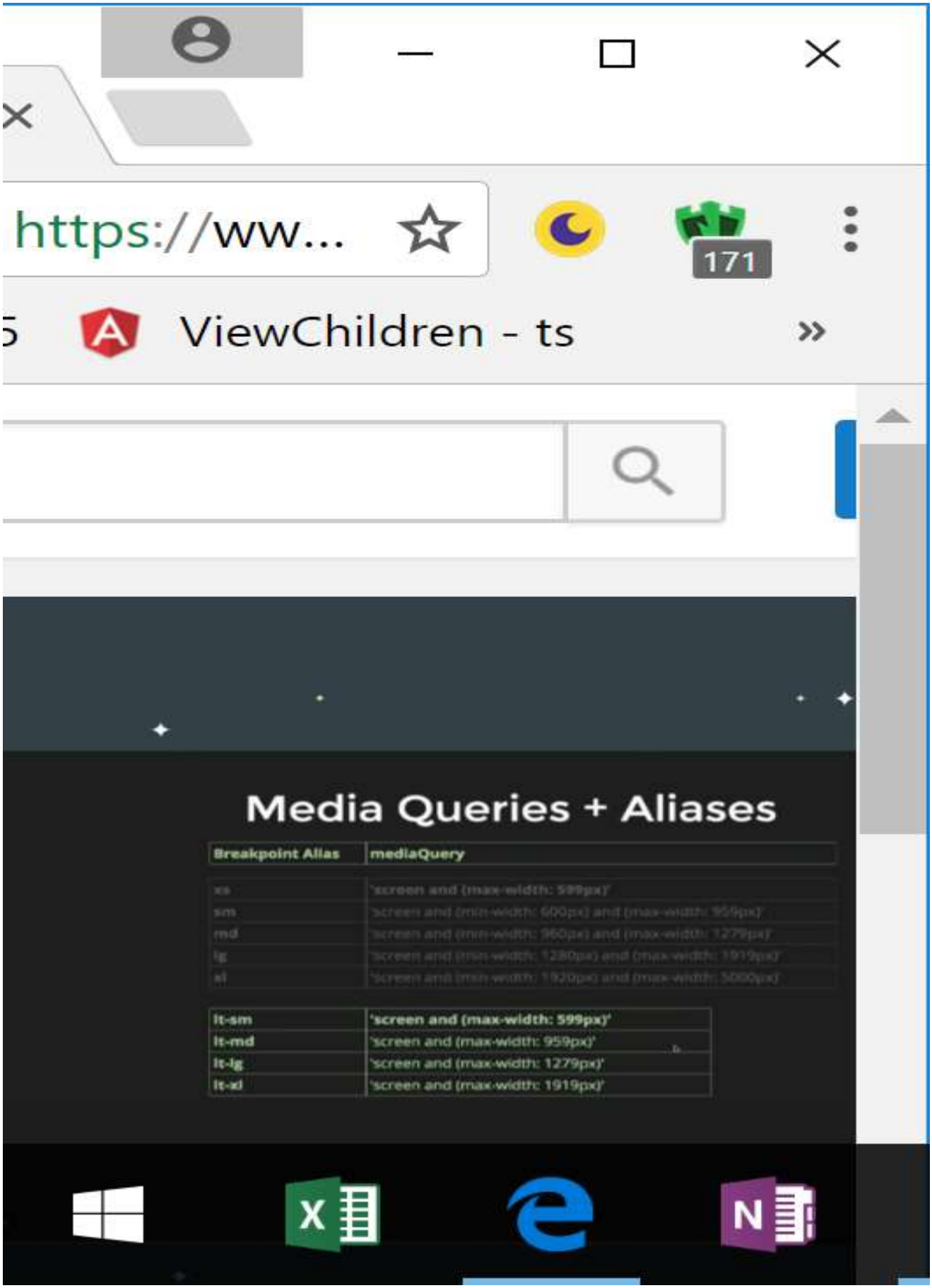
The Static API is

- **Directives in HTML**
- **Declarative**
- **Supports Data Binding**
- **Builds CSS – dynamically injected inline**
- **Supports change detection**
- **Includes fxHide and fxShow features**
- **ngClass and ngStyle features too**
- **img – (not sure terminology) – diff urls for diff size screens**
- **Performant**

Static API code options	API for DOM Containers
fxLayout	<div fxLayout=”row” fxLayout.lt-md=”column”>
fxLayoutWrap	<div fxLayoutWrap>
fxLayoutGap	
fxLayoutAlign	<div fxLayoutAlign=”start stretch”>

Note no CSS used

```
<div fxLayout=”row” fxLayoutAlign=”center”>
        <div fxFlex=”1 1 auto”>One</div>
        <div fxFlex=”5 1 auto”>Two</div>
        <div fxFlex=”1 1 auto”>Three</div>
        <div fxFlex=”1 1 auto”>Four</div>
</div>
```


https://ww...
171
ViewChildren - ts
Media Queries + Aliases
Breakpoint Alias
mediaQuery
xs
'screen and (max-width: 599px)'
sm
'screen and (min-width: 600px) and (max-width: 959px)'
md
'screen and (min-width: 960px) and (max-width: 1279px)'
lg
'screen and (min-width: 1280px) and (max-width: 1919px)'
xl
'screen and (min-width: 1920px) and (max-width: 5000px)'
lt-sm
'screen and (max-width: 599px)'
lt-md
'screen and (max-width: 959px)'
lt-lg
'screen and (max-width: 1279px)'
lt-xl
'screen and (max-width: 1919px)'

More Complex – JavaScript Imperative API

ObservableMedia	constructor(public media: ObservableMedia){}
BREAKPOINTS	provides: [{provide:BREAKPOINTS, useValue: CUSTOM_BPS}]
BaseFxDirectiveAdapter	export class ClassDirective extends NgClass

Subscribe to mediaQuery Activations