

# Angular

# Flex

# Layout

Sophisticated  
component layout  
engine for  
Angular



**YOU DONT JUST MAKE  
A RESPONSIVE WEBSITE**

**WITHOUT FLEXBOX**

# Flexbox CSS

+

# mediaQuery

# Browser support

Flexible Box Layout Module  - CR

Method of positioning elements in horizontal or vertical stacks. Support includes the support for the all properties prefixed with `flex` as well as `display: flex`, `display: inline-flex`, `align-content`, `align-items`, `align-self`, `justify-content` and `order`.

Global      82.98% + 14.34% = 97.32%  
unprefixed: 82.11% + 3.77% = 85.88%

| IE | Edge * | Firefox | Chrome | Safari | Opera | iOS Safari * | Opera Mini * | Android Browser | Chrome for Android |
|----|--------|---------|--------|--------|-------|--------------|--------------|-----------------|--------------------|
|    |        |         | 49     |        |       |              |              |                 |                    |
|    |        |         | 51     |        |       |              |              |                 |                    |
|    |        |         | 53     |        |       |              |              |                 |                    |
|    |        | 49      | 54     | 9.1    |       | 9.3          |              | 4.4             |                    |
| 11 | 14     | 50      | 55     | 10     | 41    | 10.1         | all          | 53              | 54                 |
|    | 15     | 51      | 56     | TP     | 42    |              |              |                 |                    |
|    |        | 52      | 57     |        | 43    |              |              |                 |                    |
|    |        | 53      | 58     |        |       |              |              |                 |                    |

4 Partial support is due to large amount of bugs present (see known issues)

Notes Known issues (8) Resources (12) Feedback

Most partial support refers to supporting an [older version](#) of the specification or an [older syntax](#).

Pure TypeScript  
layout engine

Independent of  
Angular Material

# Angular CLI Integration

# Install Angular Flex-Layout components

```
npm install --save @angular/flex-layout@latest
```

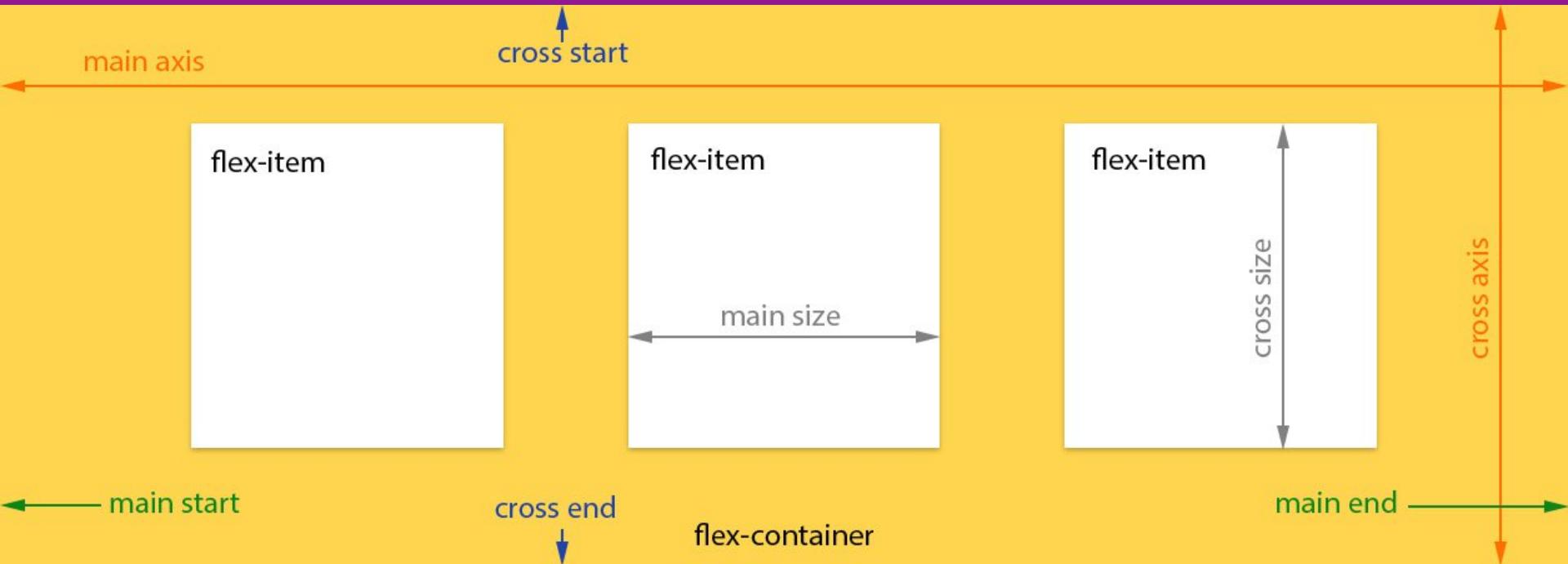
v2.0.0-beta.8

## Import Angular Flex-Layout NgModule

```
import { FlexLayoutModule }  
    from '@angular/flex-layout';  
...  
@NgModule({  
    imports: [FlexLayoutModule],  
    ...  
})  
export class AppModule { }
```

# Static Layout API

# CSS Flexbox Model



# API for DOM containers

**fxLayout**

```
<div fxLayout="row"  
      fxLayout.xs="column"> </div>
```

**fxLayoutWrap**

```
<div fxLayoutWrap> </div>
```

**fxLayoutGap**

```
<div fxLayoutGap="10px"> </div>
```

**fxLayoutAlign**

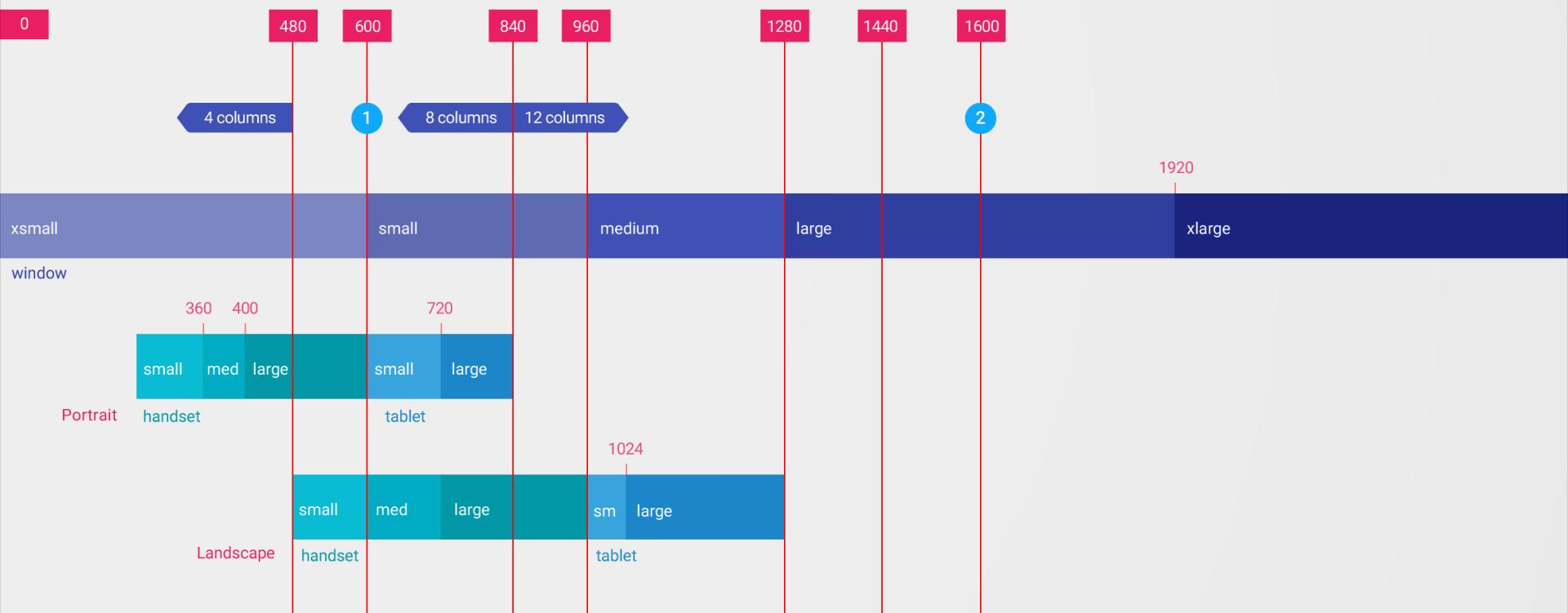
```
<div fxLayoutAlign="start stretch">  
  </div>
```

# API for flex elements

|                     |   |
|---------------------|---|
| <b>fxFlex</b>       | <pre>&lt;div fxFlex="1 2<br/>calc(15em + 20px)"&gt;&lt;/div&gt;</pre> |
| <b>fxFlexOrder</b>  | <pre>&lt;div fxFlexOrder="2"&gt;&lt;/div&gt;</pre>                    |
| <b>fxFlexOffset</b> | <pre>&lt;div fxFlexOffset="20px"&gt;&lt;/div&gt;</pre>                |
| <b>fxFlexAlign</b>  | <pre>&lt;div fxFlexAlign="center"&gt;&lt;/div&gt;</pre>               |
| <b>fxFlexFill</b>   | <pre>&lt;div fxFlexFill&gt;&lt;/div&gt;</pre>                         |

# Responsive API

# Material Design Breakpoints



# Media Queries and Aliases

| breakpoint   | mediaQuery   |
|--------------|--|
| <b>xs</b>    | 'screen and (max-width: 599px)'                          |
| <b>sm</b>    | 'screen and (min-width: 600px) and (max-width: 959px)'   |
| <b>md</b>    | 'screen and (min-width: 960px) and (max-width: 1279px)'  |
| <b>lg</b>    | 'screen and (min-width: 1280px) and (max-width: 1919px)' |
| <b>xl</b>    | 'screen and (min-width: 1920px) and (max-width: 5000px)' |
| <b>lt-sm</b> | 'screen and (max-width: 599px)'                          |
| <b>lt-md</b> | 'screen and (max-width: 959px)'                          |
| <b>lt-lg</b> | 'screen and (max-width: 1279px)'                         |
| <b>lt-xl</b> | 'screen and (max-width: 1919px)'                         |
| <b>gt-xs</b> | 'screen and (min-width: 600px)'                          |
| <b>gt-sm</b> | 'screen and (min-width: 960px)'                          |
| <b>gt-md</b> | 'screen and (min-width: 1280px)'                         |
| <b>gt-lg</b> | 'screen and (min-width: 1920px)'                         |

# Breakpoint Activation Fallback Algorithm

- Flex-Layout responsive engine uses a *fallback, descending-scan* algorithm
- For non-overlapping breakpoints: the search scans from *largest-to-small* breakpoint
- For overlapping breakpoints: the search scans from *smallest-to-largest* breakpoint range

# Visibility example

```
<div fxShow fxHide.xs="false" fxHide.lg="true"></div>
```

- **xl**, then *fallback* to the default fxShow; so the div is shown
- **lg**, then the div is hidden (since the value === 'true')
- **md**, then *fallback* to the default fxShow; so the div is shown
- **sm**, then *fallback* to the default fxShow; so the div is shown
- **xs**, then the div is shown (since the value === 'false')

# Sizing example

```
<div fxFlex="50%" fxFlex.gt-sm="100%"></div>
```

- **xl**, then *fallback* to 'gt-sm' so the div sizing is 100%
- **lg**, then *fallback* to 'gt-sm' so the div sizing is 100%
- **md**, then *fallback* to 'gt-sm' so the div sizing is 100%
- **sm**, then *fallback* to the default fxFlex="50%"; so the div is 50%
- **xs**, then *fallback* to the default fxFlex="50%"; so the div is 50%

# Special Responsive Features

## fxShow

```
<div fxShow  
      [ fxShow.xs ]="isVisibleOnMobile( )" ></div>
```

## fxHide

```
<div fxHide  
      [ fxHide.gt-sm ]="isVisibleOnDesktop( )" ></div>
```

## ngClass

```
<div  
      [ ngClass.sm ]="{'fxClass-sm': hasStyle}" ></div>
```

## ngStyle

```
<div  
      [ ngStyle.xs ]="{'color: 'blue'}" ></div>
```

# JavaScript API (Imperative)

# Programmatic features

## ObservableMedia

```
constructor(public  
    media:ObservableMedia ) {}
```

## BREAKPOINTS

```
providers: [{provide: BREAKPOINTS,  
    useValue: MY_CUSTOM_BREAKPOINTS }]
```

## BaseFxDirectiveAdapter

```
export class ClassDirective  
    extends NgClass {}}
```

# Subscribe to mediaQuery activations

```
import {Subscription} from "rxjs/Subscription";
import {MediaChange, ObservableMedia}
  from "@angular/flex-layout";

constructor(media: ObservableMedia) {
  this.watcher = media.subscribe(
    (change: MediaChange) => {

      if ( change.mqAlias == 'xs' ) {
        this.loadMobileContent();
      }

    } );
}

}
```

"Holy Grail" of  
layouts

header

nav

article

aside

footer

```
<div>
  <header>header</header>
  <div>

    <nav>nav</nav>

    <article>article</article>

    <aside>aside</aside>

  </div>
  <footer>footer</footer>
</div>
```

```
<div fxLayout="column">
  <header>header</header>
  <div fxLayout="row" fxFlex>

    <nav fxFlex="1 6 20%">nav</nav>

    <article fxFlex="3 1 60%">article</article>

    <aside fxFlex="1 6 20%">aside</aside>

  </div>
  <footer>footer</footer>
</div>
```

header

article

nav

aside

footer

```
<div fxLayout="column">
  <header>header</header>
  <div fxLayout="row" fxLayout.xs="column" fxFlex>
    <nav fxFlex="1 6 20%" fxFlexOrder
      fxFlexOrder.xs="2">nav</nav>
    <article fxFlex="3 1 60%" fxFlexOrder
      fxFlexOrder.xs="1">article</article>
    <aside fxFlex="1 6 20%" fxFlexOrder
      fxFlexOrder.xs="3">aside</aside>
  </div>
  <footer>footer</footer>
</div>
```

# Resources

# Angular Flex-Layout

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

Visual guide to Flex

<http://cssreference.io/flexbox/>

# Thank you!

Katrine Orlova

[github.com/  
cheerypick](https://github.com/cheerypick)  
[@cheerypick](https://twitter.com/cheerypick)