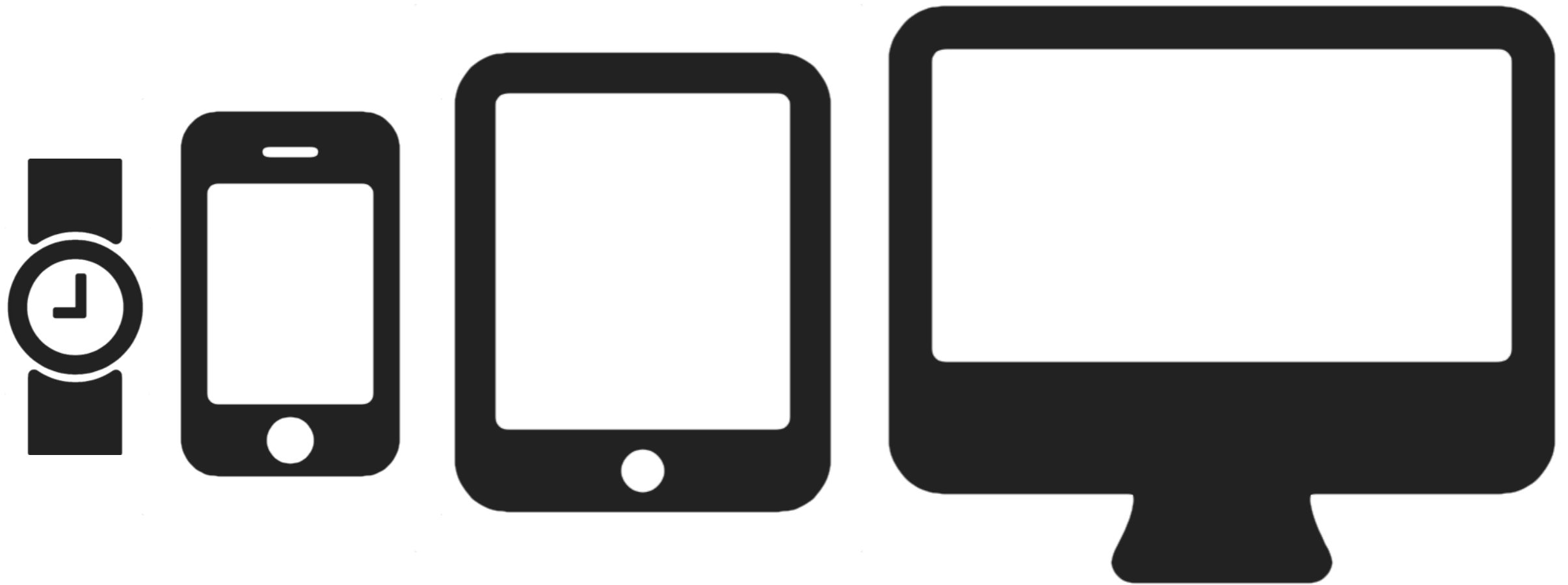


Building a desktop application is easy



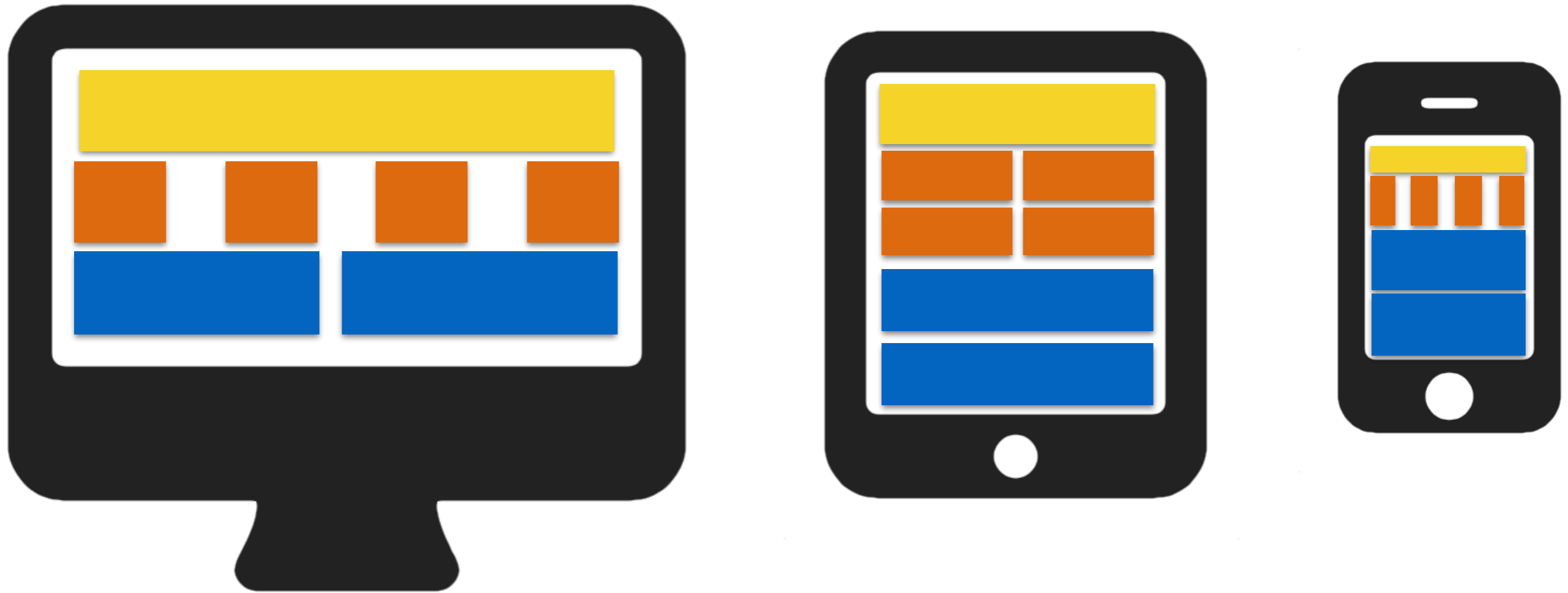
But what happens if you need to support all these devices?



Let's test this...



How other solve this problem?



Responsive Design

Twitter Bootstrap

Available classes

Use a single or combination of the available classes for toggling content across viewport breakpoints.

defined
device sizes



	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
<code>.visible-xs</code>	Visible	Hidden	Hidden	Hidden
<code>.visible-sm</code>	Hidden	Visible	Hidden	Hidden
<code>.visible-md</code>	Hidden	Hidden	Visible	Hidden
<code>.visible-lg</code>	Hidden	Hidden	Hidden	Visible
<code>.hidden-xs</code>	Hidden	Visible	Visible	Visible
<code>.hidden-sm</code>	Visible	Hidden	Visible	Visible
<code>.hidden-md</code>	Visible	Visible	Hidden	Visible
<code>.hidden-lg</code>	Visible	Visible	Visible	Hidden



CSS support

Example

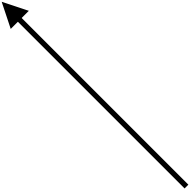
```
<div class=„hidden-xs“>icon</div>
```

```
<div class=„visible-xs“>mini toolbar</div>
```

```
<div class=„visible-sm visible-sd“>normal toolbar</div>
```

```
<div class=„visible-lg“>large toolbar</div>
```

define what you
want to see on
different devices

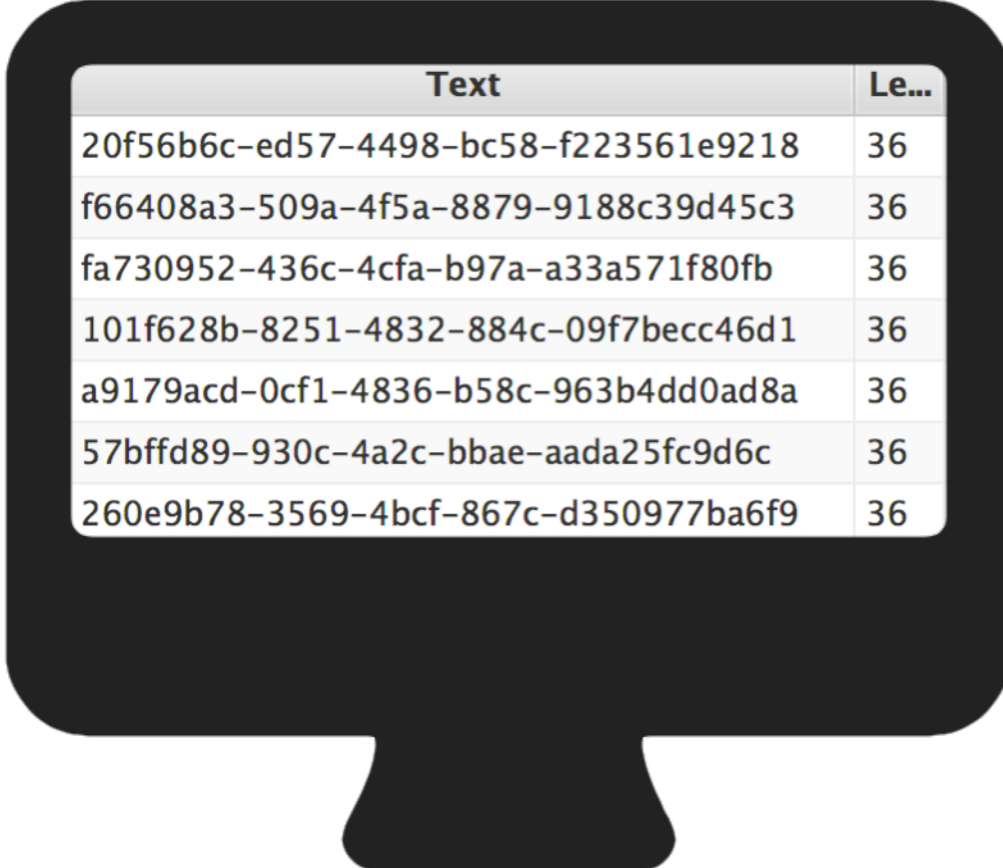


We have CSS in JavaFX, too.

Wouldn't it be cool if we can code like this:

```
TableView table = . . .  
table.getStyleClass().addAll("visible-lg", „visible-md");  
  
ListView list = . . .  
list.getStyleClass().addAll("visible-sm", „visible-xs");  
  
pane.getChildren().addAll(table, list);
```

table



Text	Le...
20f56b6c-ed57-4498-bc58-f223561e9218	36
f66408a3-509a-4f5a-8879-9188c39d45c3	36
fa730952-436c-4cfa-b97a-a33a571f80fb	36
101f628b-8251-4832-884c-09f7becc46d1	36
a9179acd-0cf1-4836-b58c-963b4dd0ad8a	36
57bffd89-930c-4a2c-bbae-aada25fc9d6c	36
260e9b78-3569-4bcf-867c-d350977ba6f9	36

list



20f56b6c-ed57-449
f66408a3-509a-4f5
fa730952-436c-4cf
101f628b-8251-483
a9179acd-0cf1-483
57bffd89-930c-4a2
260e9b78-3569-4b

You can!

```
<dependency>  
  <groupId>com.guigarage</groupId>  
  <artifactId>responsivefx</artifactId>  
  <version>X.Y</version>  
</dependency>
```


Demo

Flutter

- was introduced last year at JavaOne
- Flat UI for embedded devices
- Will be released this year after JavaOne

Demo

Ok, that's cool! But do I really need to define specific controls for all sizes?

```
Toolbar extraSmallToolbar = . . .  
extraSmallToolbar.getStyleClass().add("visible-xs");
```

```
Toolbar smallToolbar = . . .  
smallToolbar.getStyleClass().add("visible-sm");
```

```
Toolbar mediumToolbar = . . .  
mediumToolbar.getStyleClass().add("visible-md");
```

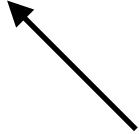
```
Toolbar largeToolbar = . . .  
largeToolbar.getStyleClass().add("visible-lg");
```

```
toolbarPane.getChildren().addAll(extraSmallToolbar,  
                                  smallToolbar,  
                                  mediumToolbar,  
                                  largeToolbar);
```

Thanks god! JavaFX supports pseudo classes!

```
Toolbar toolbar = . . .  
toolbarPane.getChildren().add(toolbar);
```

you only
need one
instance



Thanks god! JavaFX supports pseudo classes!

```
#toolbar {  
    -fx-padding: 12 12 1 12;  
    -fx-background-color: deepskyblue;  
    -fx-border-color: blue;  
    -fx-border-width: 0 0 1 0;  
}
```

Define the defaults



```
#toolbar:medium-device {  
    -fx-padding: 6 6 1 6;  
}
```

```
#toolbar:small-device {  
    -fx-padding: 2 2 1 2;  
}
```

...

responsive style



Demo

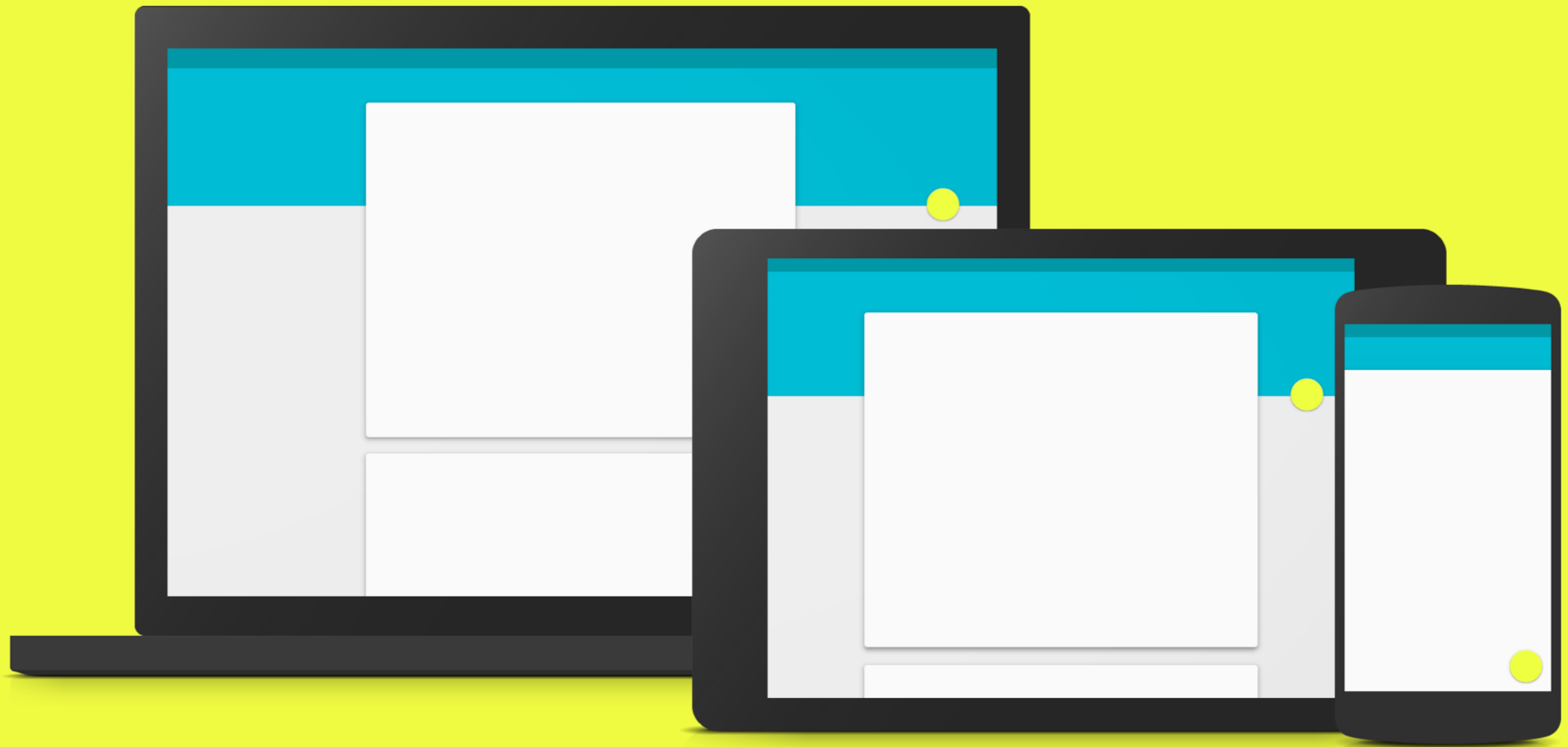
How can I get it?

Will be released as OpenSource Library by

cano

Let's go extreme

Material Design



Material Design Facts

Google show Preview on Google I/O this year

One Theme for all devices

Modern Flat UI and fresh animations

<http://www.google.com/design/spec/material-design/>

Maybe we can do this with JavaFX, too?

Demo

More extreme stuff?

Extreme GUI Makeover