

Making an App-friendly Stylesheet



Maximiliano Firtman

MOBILE+WEB DEVELOPER

@firt firt.mobi

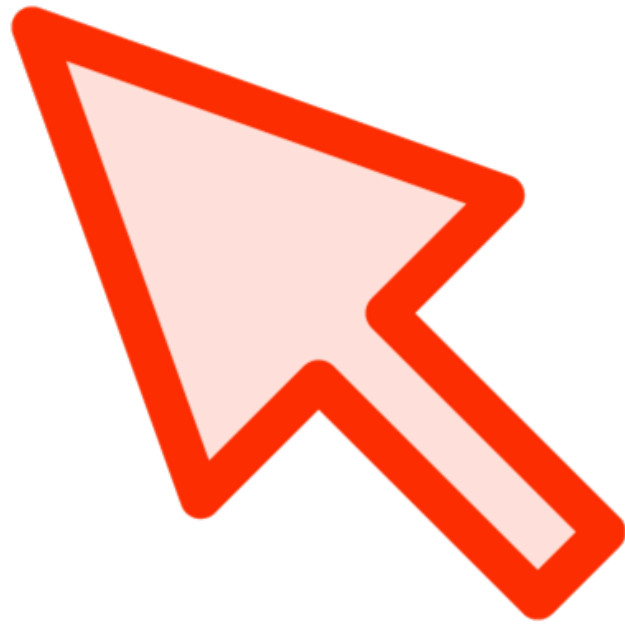
Overview

App-friendly Stylesheets

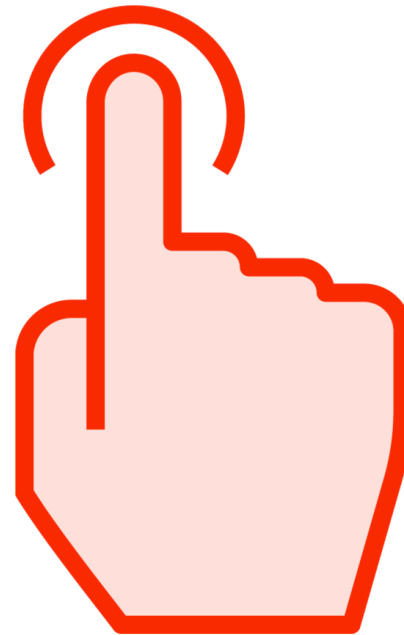
- Improve User Experience
- Media Queries for PWAs
- Notch-based and Foldable Devices
- Add Accessibility and Dark Mode Support

Improving User Experience

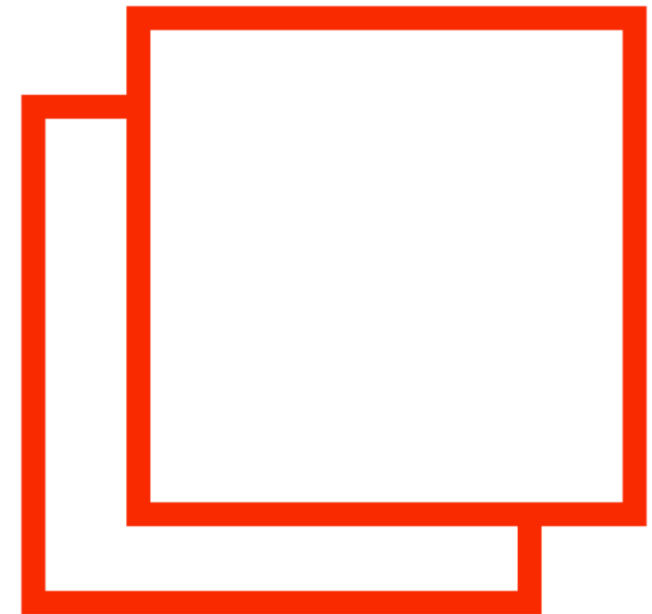
How to improve App's User Experience?



Content Selection

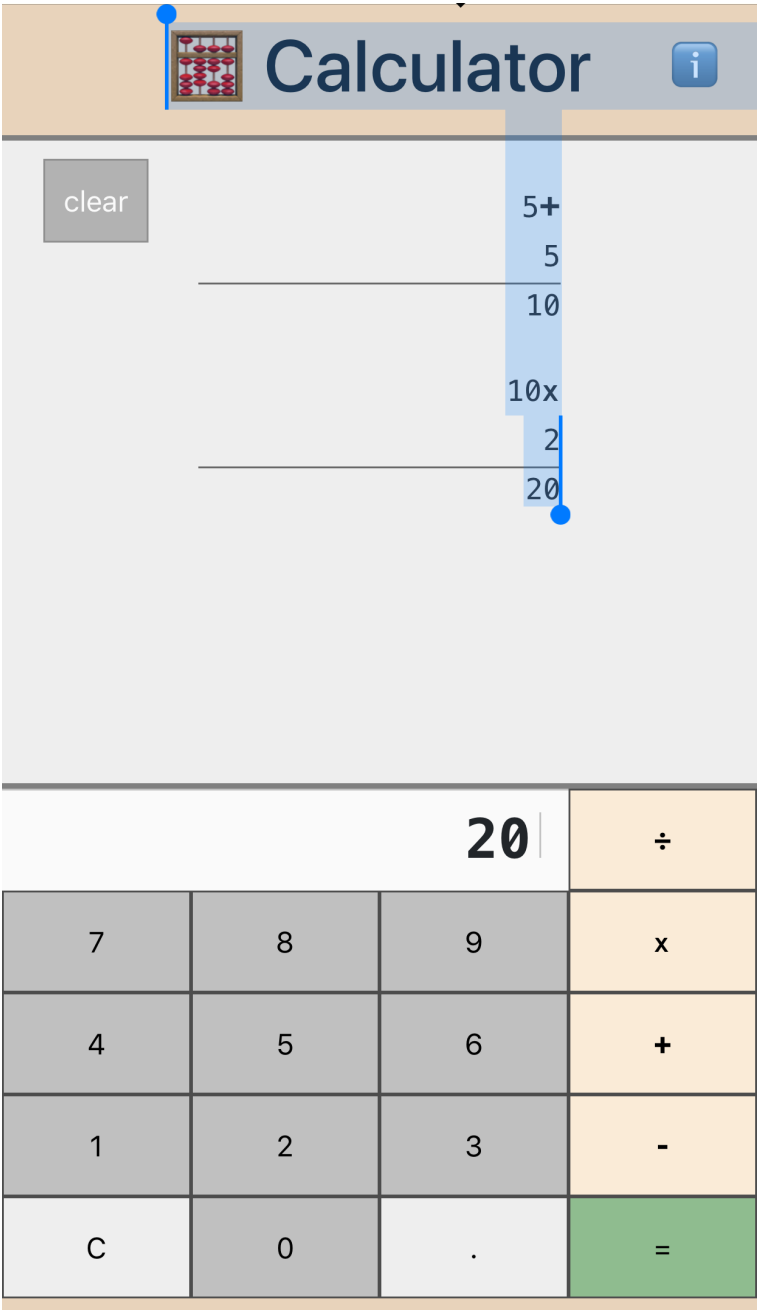
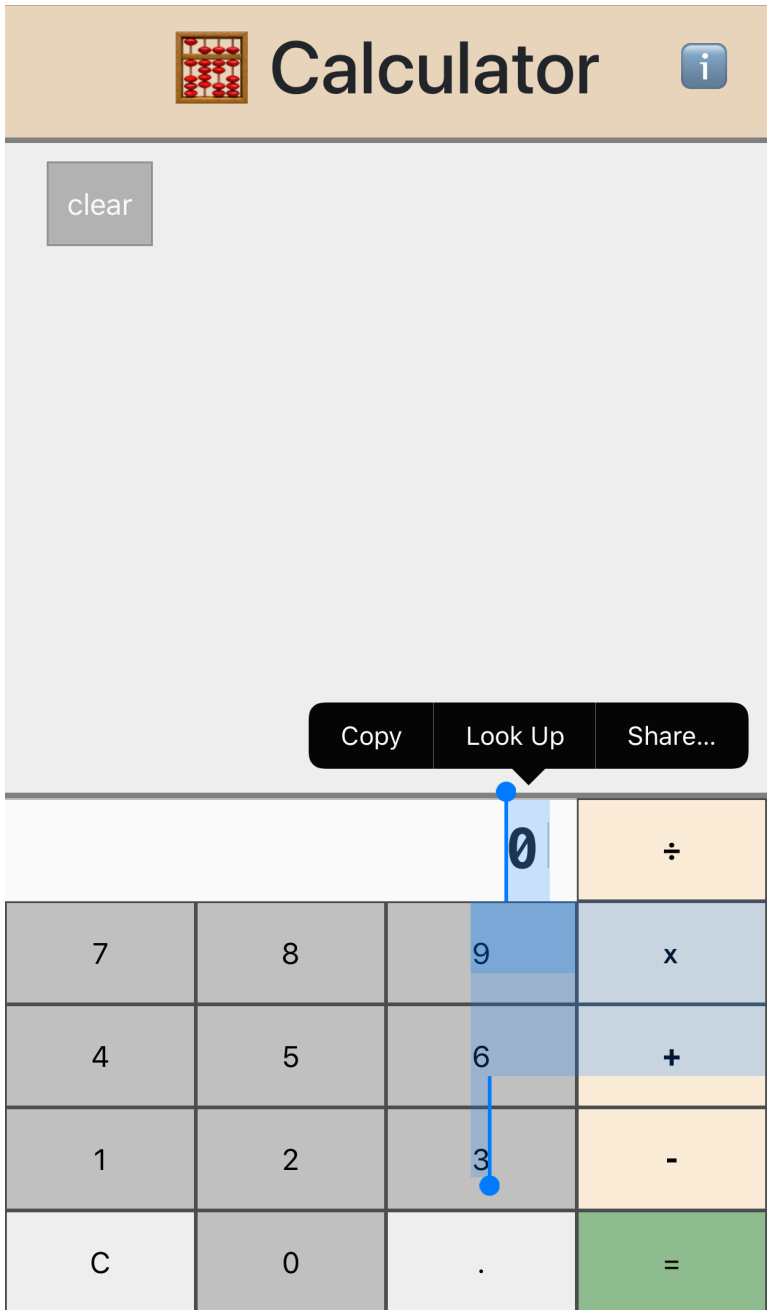
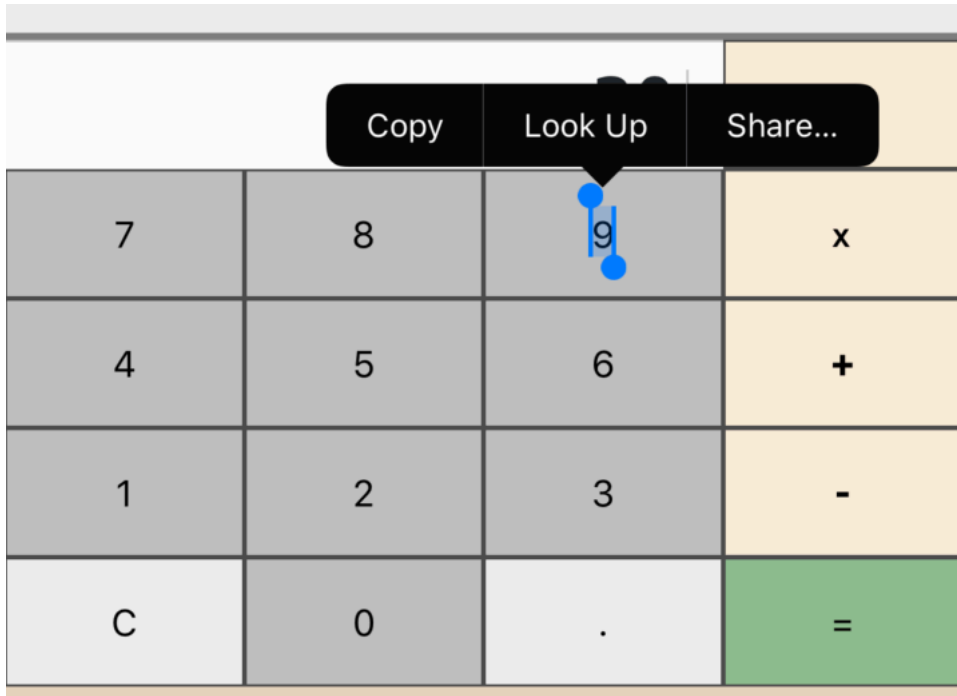


Tap Effects



Safe areas

Content Selection



```
.elements {  
  user-select: none;  
  -webkit-user-select: none;  
}
```

Avoid Content Selection

We use user-select to disable selection on elements that are not data

Use this with responsibility!

Tap Highlight Color

| | | | |
|----|---|---|---|
| 20 | | | ÷ |
| 7 | 8 | 9 | x |
| 4 | 5 | 6 | + |
| 1 | 2 | 3 | - |
| C | 0 | . | = |

Tap Highlight Color

| | | | |
|----|---|---|---|
| 20 | | | ÷ |
| 7 | 8 | 9 | x |
| 4 | 5 | 6 | + |
| 1 | 2 | 3 | - |
| C | 0 | . | = |

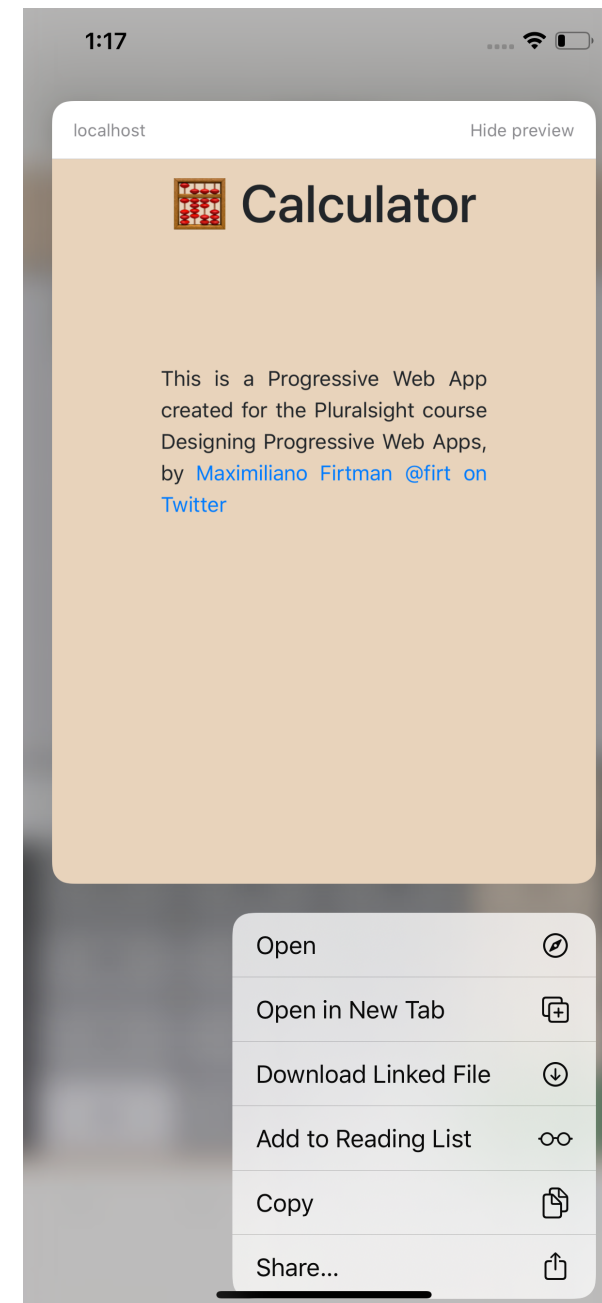
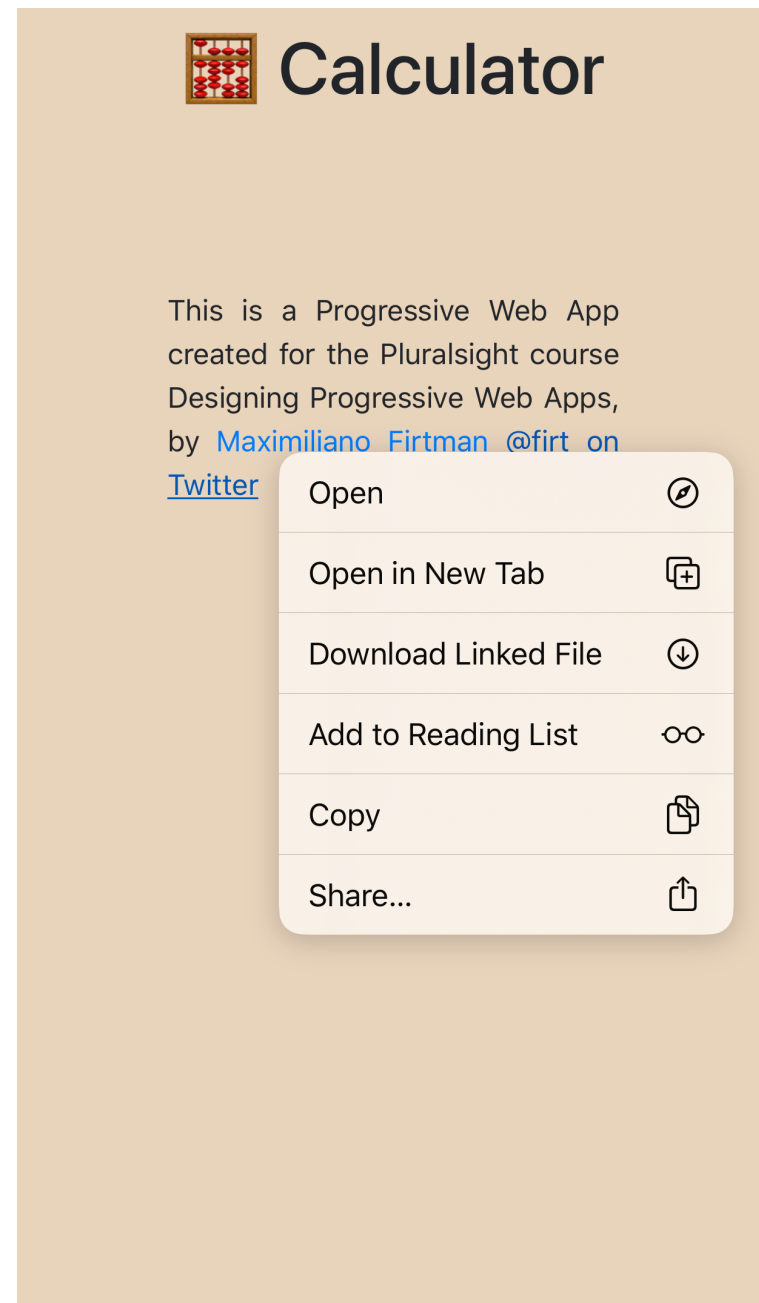
```
button {  
    -webkit-tap-highlight-color: rgba(255, 120, 120, 0.5)  
}
```

Define Tap Highlight Color

It's a color to be used as an over box when user touches a clickable area

Careful: always use RGBA colors with transparency!

iOS Touch Callout

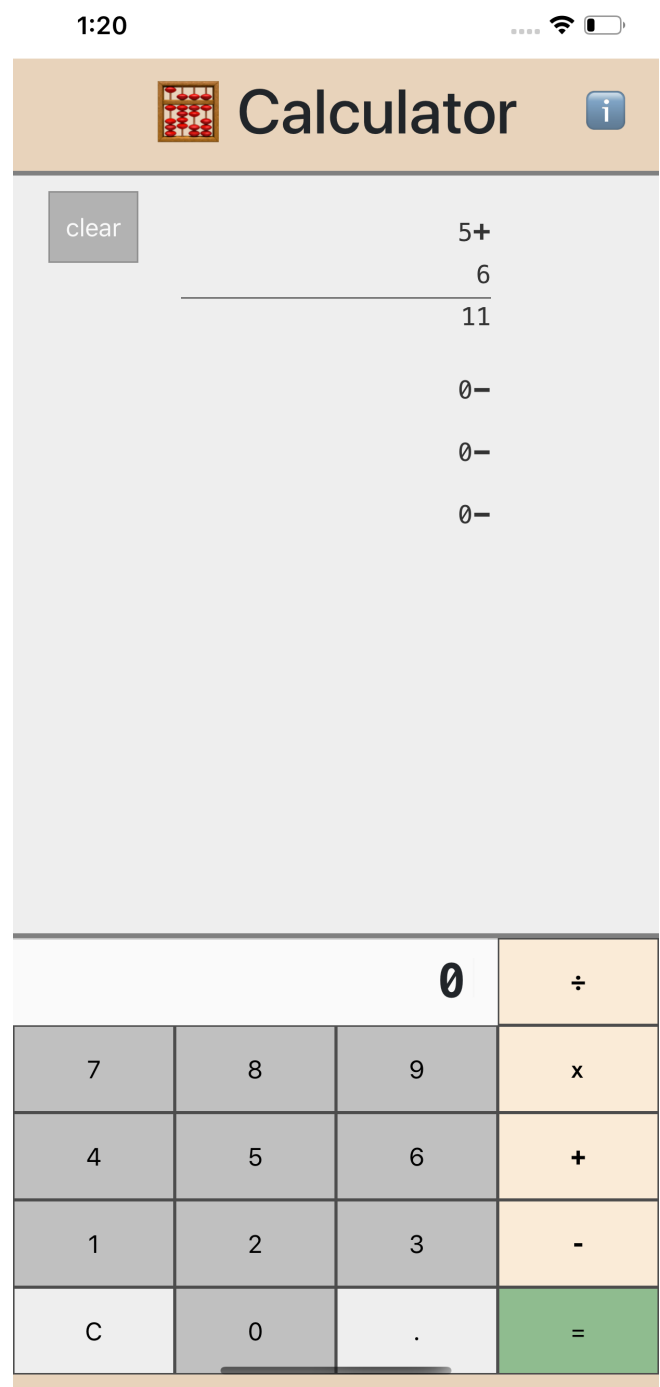


```
a {  
    -webkit-touch-callout: none;  
}
```

Disable Touch Callout

We should apply this to every link including javascript: protocol

Safe Areas



Safe Areas

| | | | |
|---|---|---|---|
| | | | |
| 0 | | | ÷ |
| 7 | 8 | 9 | x |
| 4 | 5 | 6 | + |
| 1 | 2 | 3 | - |
| C | 0 | . | = |

Safe Areas

| | | | |
|---|---|---|---|
| | | | |
| 0 | | | ÷ |
| 7 | 8 | 9 | x |
| 4 | 5 | 6 | + |
| 1 | 2 | 3 | - |
| | | | |
| C | 0 | . | = |

Non-safe area

```
.container {  
  margin: env(safe-area-inset-top)  
         env(safe-area-inset-right)  
         env(safe-area-inset-bottom)  
         env(safe-area-inset-left) !important;  
}
```

Use Safe Areas Paddings or Margins

With the new CSS Environmental Variables support through the `env()` function and four variables defined today for safe areas


```
.container {  
  margin: env(safe-area-inset-top, 5px)  
         env(safe-area-inset-right, 5px)  
         env(safe-area-inset-bottom)  
         env(safe-area-inset-left) !important;  
}
```

Use Safe Areas Paddings or Margins

We can also add default values if that variable is not available, but only when env() function is supported

```
.container {  
  margin: 5px 5px 0 0;  
  margin: env(safe-area-inset-top, 5px)  
          env(safe-area-inset-right, 5px)  
          env(safe-area-inset-bottom)  
          env(safe-area-inset-left) !important;  
}
```

Use Safe Areas Paddings or Margins

Adding two declaration in this orden will make these default values available for every device and browser

Demo

Improving User Experience

- Disable Content Selection on some elements
- Define a Tap Highlight Color
- Remove Touch Callout Effect
- Add Paddings for Safe Areas

PWA Media Queries

Display Options

Browser

Standalone

Fullscreen

Minimal-UI

```
@media (display-mode: standalone) {  
  
}
```

Display Mode Media Query

We can query for browser, standalone, minimal-ui or fullscreen. We can use a multiple declaration for detecting many

Sometimes we want to do
something only for iOS and
iPadOS


```
@supports (-webkit-touch-callout: none) {  
  
}
```

Detecting iOS and iPadOS

This is a hack and it should be used with care

```
@supports (-webkit-touch-callout: none) {  
  @media (display-mode: standalone) {  
    // this will apply on iOS/iPadOS on installed PWA  
  }  
}
```

Detecting iOS and iPadOS

We can mix it with the media query

Supporting Notch-based and Foldable Devices

Devices with Notch



Web Apps with a Notch-based Device



Browser

Installed PWA

Viewport Area

0

÷

7

8

9

x

4

5

6

+

1

2

3

-

C

0

.

=

clear

5+
6
11
0-
0-
0-

PWAs in FullScreen or Standalone mode

Ask for Cover Viewport

Use Safe Areas


```
<meta name="viewport" content="width=device-width,  
initial-scale=1.0, viewport-fit=cover">
```

Viewport Fit Cover

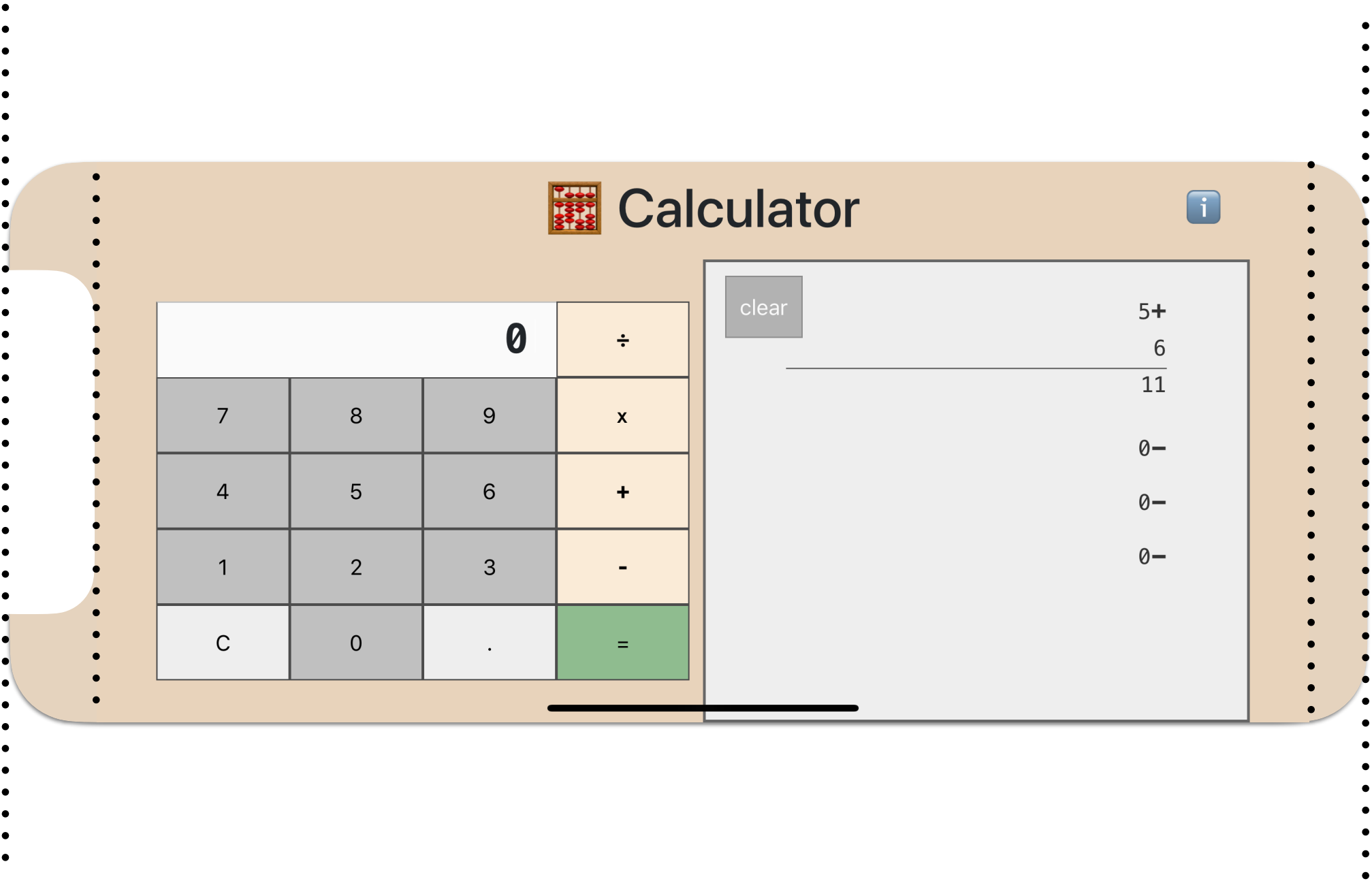
We can ask the availability for all the viewport for us with the viewport meta tag


```
.container {  
  margin: env(safe-area-inset-top)  
         env(safe-area-inset-right)  
         env(safe-area-inset-bottom)  
         env(safe-area-inset-left) !important;  
}
```

Use Safe Areas Paddings or Margins

With the new CSS Environmental Variables support through the `env()` function and four variables defined today for safe areas

Viewport Area



Foldable Devices



Today, only with Responsive
Web Design we can support
foldable devices properly

Some proposals are in progress to let us define what to render on each screen or to query about the current screen

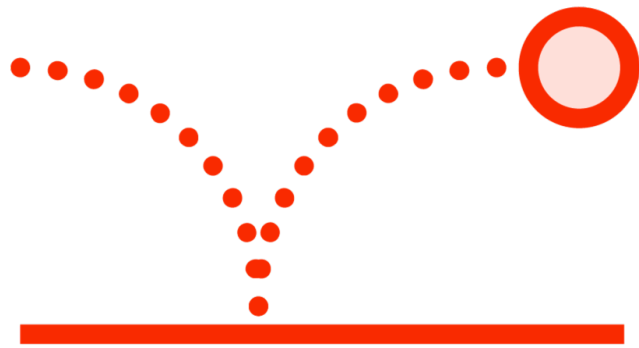
Demo

Notch-devices

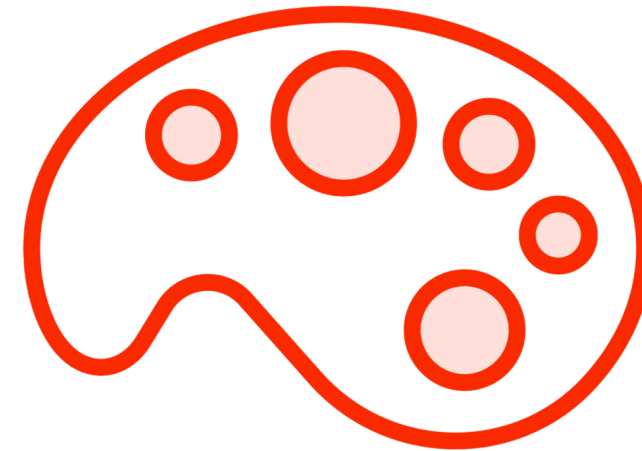
- Change our viewport meta tag

Adding Accessibility and Dark Mode Support

Media Queries for Accessibility

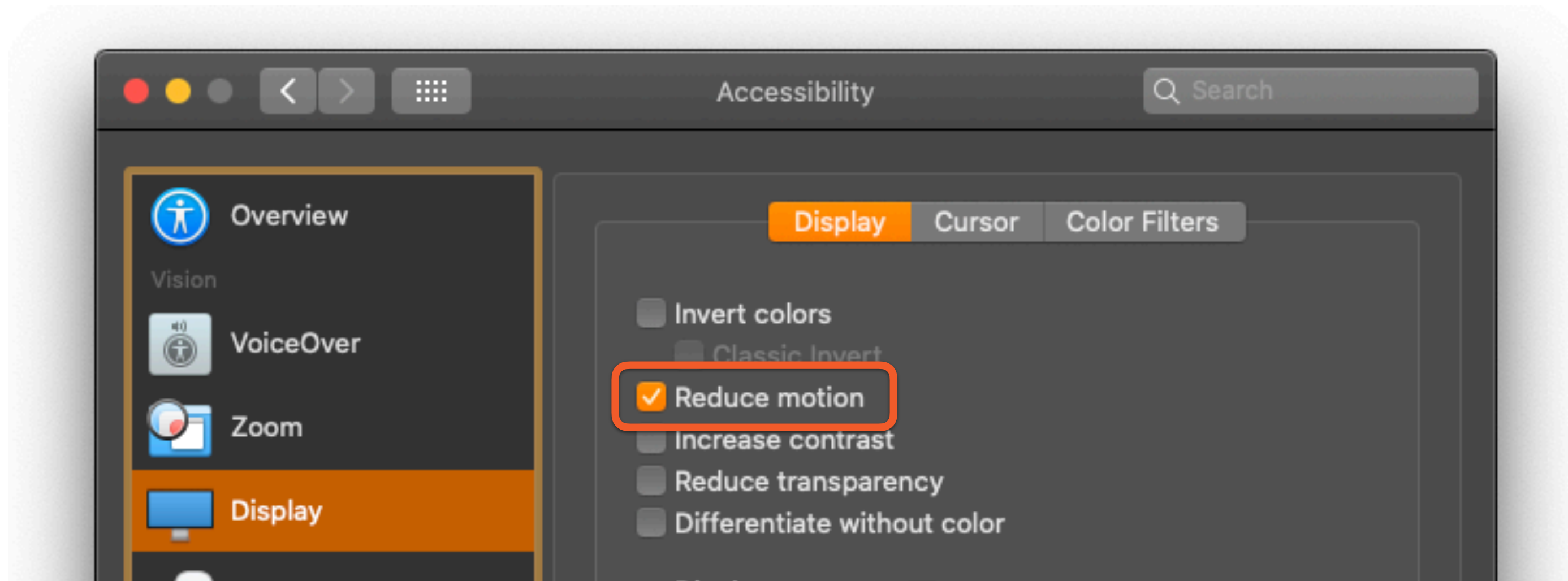


Reduced Motion

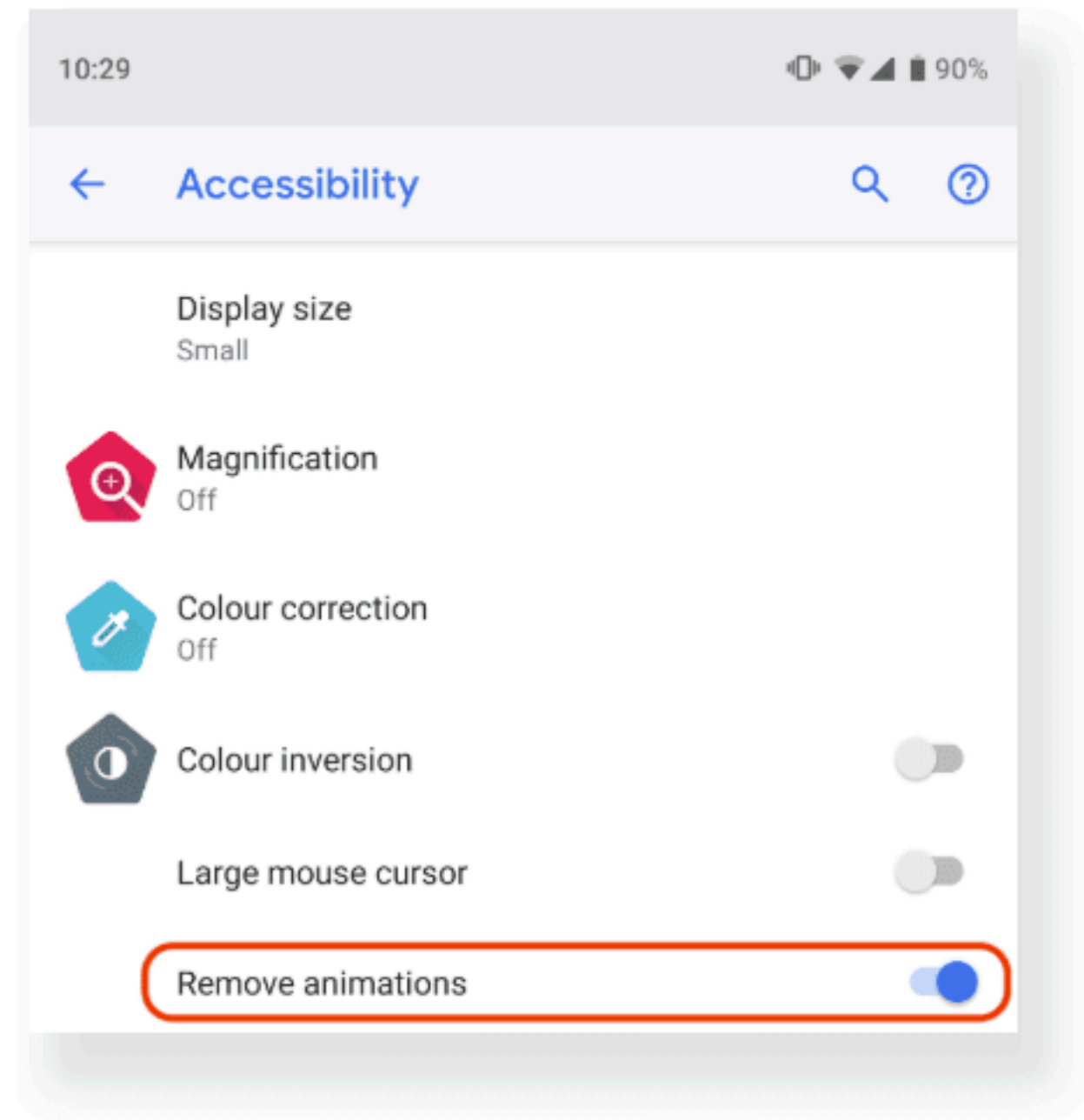
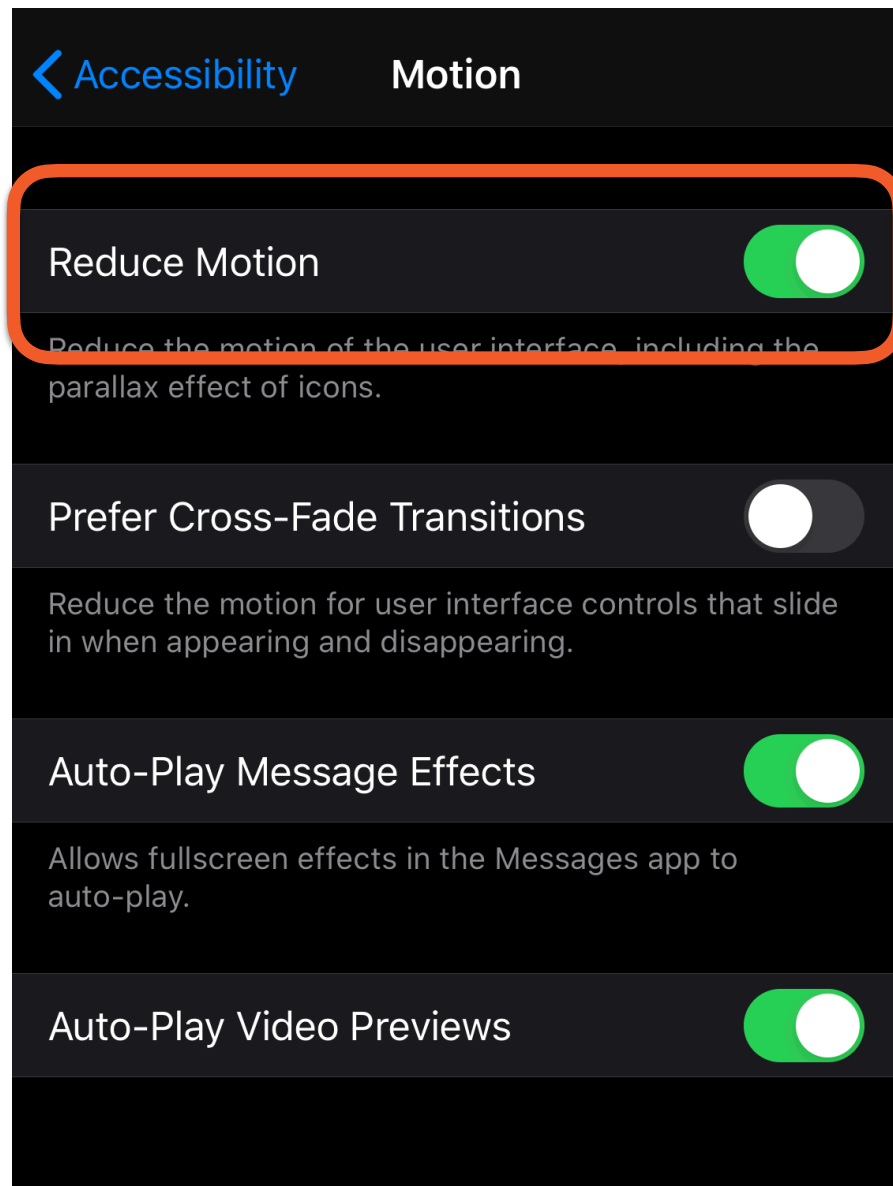


High Contrast Mode

Reduced Motion



Reduced Motion

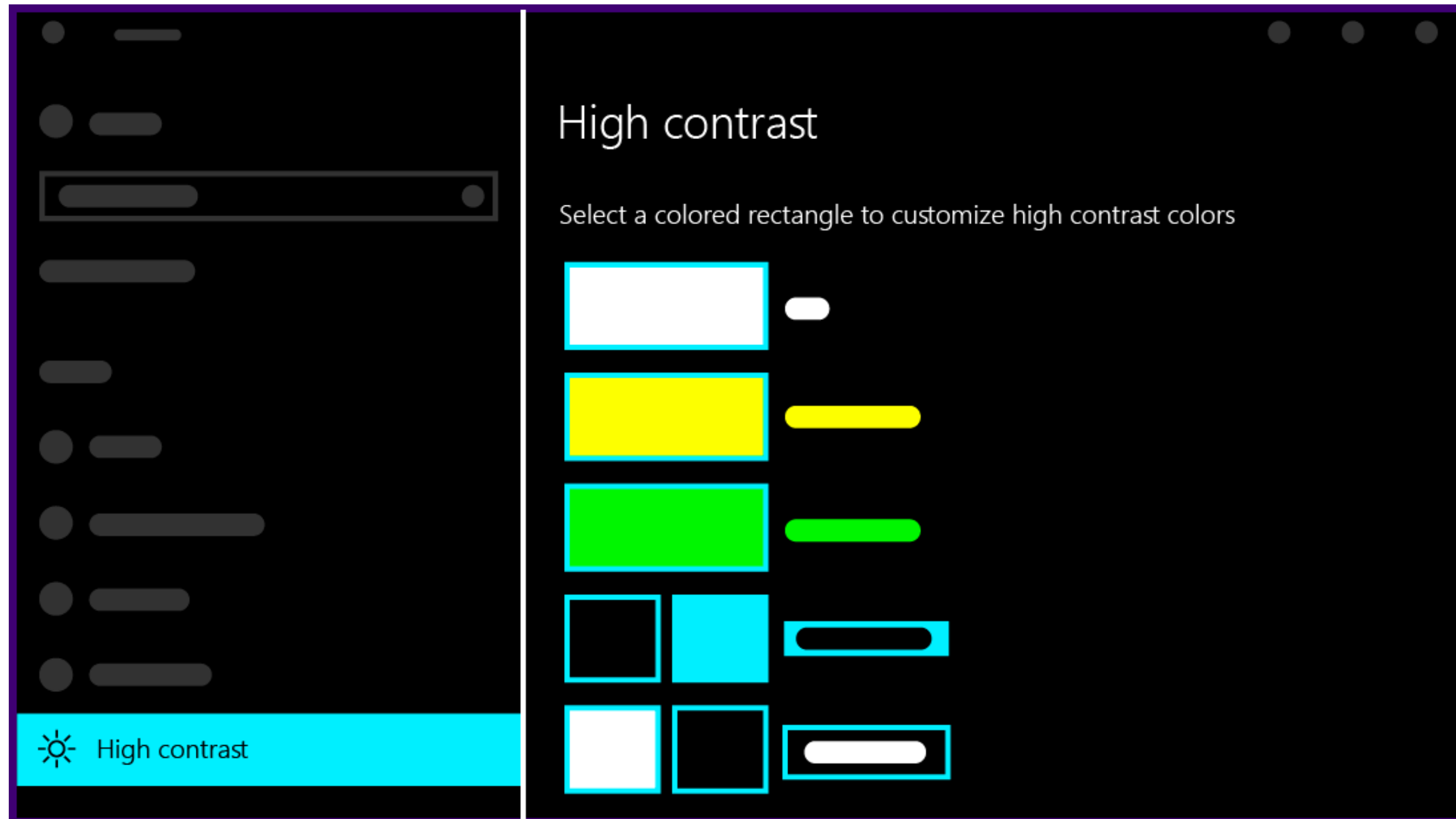


```
@media (prefers-reduced-motion: reduce) {  
  
}
```

Reduced Motion Media Query

We should stop animations and transitions for users with this setting enabled

High Contrast Mode



```
@media (-ms-high-contrast: active) {  
  
}
```

Microsoft High Contrast Media Query

This is an extension for Windows and it means we should use high-contrast colors


```
@media (-ms-high-contrast: black-on-white) {  
  
}  
@media (-ms-high-contrast: white-on-black) {  
  
}
```

Microsoft High Contrast Media Query

There are values to know the preferred user scheme

PWA Dark Mode Support

Windows

iOS and iPadOS

macOS

Android 10+

```
@media (prefers-color-scheme: dark) {  
  
}
```

Dark Mode Media Query

We can define alternative colors for dark mode. If you use CSS variables, you can declare the new values here.

Demo

Accessibility and Dark Mode

- Remove Cursor Animation for Accessibility
- Add a Dark Mode version
- Check PWA Status with Lighthouse

Summary

App-friendly Stylesheets

- Improved User Experience
- Media Queries for PWAs
- Notch-based and Foldable Devices
- Added Accessibility and Dark Mode Support

Up Next:

Preparing PWA for Distribution
