



Progressive Web Apps

You Don't Need that App Store

Maximiliano Firtman

firt.dev

X x.com/firt

in linkedin.com/in/firtman

github.com/firtman

About me

Maximiliano Firtman



MOBILE+WEB DEVELOPER

HTML since 1996

JavaScript since 1998, +150 web apps

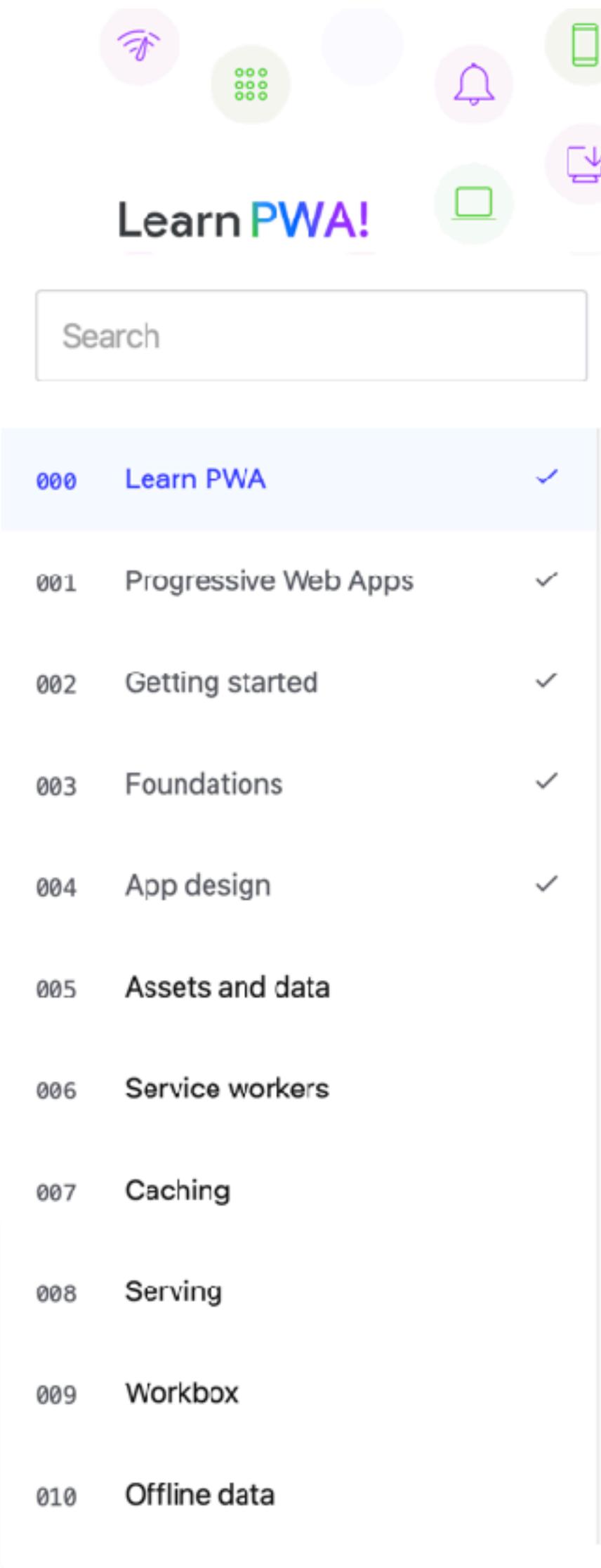
AUTHOR

Authored 13 books, +70 courses



About me

Maximiliano Firtman



The screenshot shows the 'Learn PWA!' course interface. At the top, there are several circular icons with icons inside: a purple Wi-Fi icon, a green grid icon, a white circle icon, a purple bell icon, a green smartphone icon, and a purple download/cursor icon. Below these is the title 'Learn PWA!' in a purple and white font. Underneath the title is a search bar with the placeholder 'Search'. A vertical red line on the left marks the start of the sidebar. The sidebar contains a list of 10 modules, each with a blue number and a title, followed by a checked checkbox icon:

- 000 Learn PWA
- 001 Progressive Web Apps
- 002 Getting started
- 003 Foundations
- 004 App design
- 005 Assets and data
- 006 Service workers
- 007 Caching
- 008 Serving
- 009 Workbox
- 010 Offline data

000

Learn PWA

A course that breaks down every aspect of modern progressive web app development.

Welcome to Learn Progressive Web Apps!

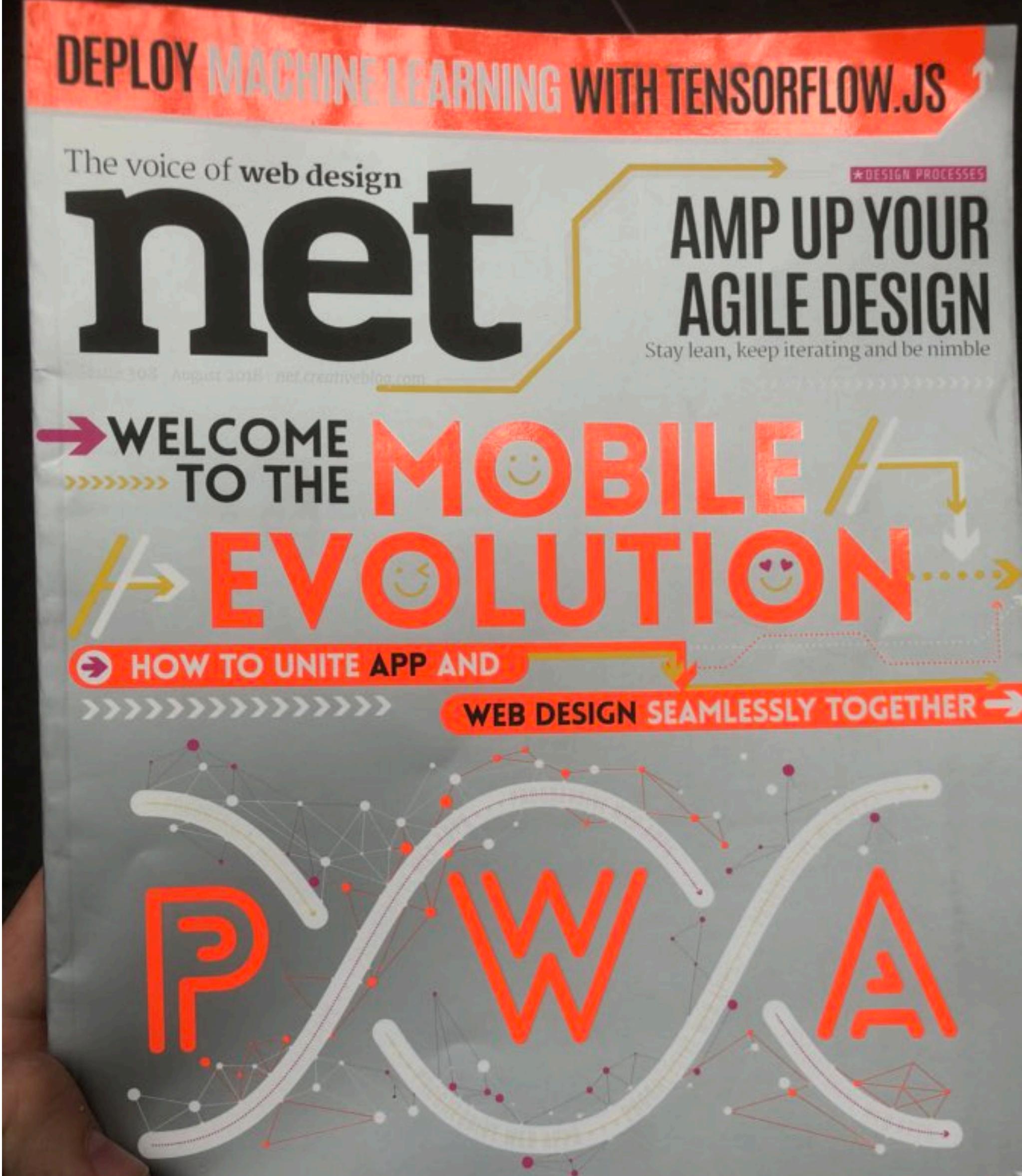
Welcome to Learn Progressive Web Apps!

This course covers the fundamentals of Progressive Web App development in easy-to-understand pieces. Over the following modules, you'll learn what a Progressive Web App is, how to create one or upgrade your existing web content, and how to add all the pieces for an offline, installable app. Use the menu pane to navigate the modules. (The menu is at left on desktop or behind the hamburger menu on mobile.)

You'll learn PWA fundamentals like the Web App Manifest, service workers, how to design with an app in mind, how to use other tools to test and debug your PWA. After these fundamentals, you'll learn about integration with the platform and operating system, how to enhance your PWA's installation and usage experience, and how to offer an offline experience.

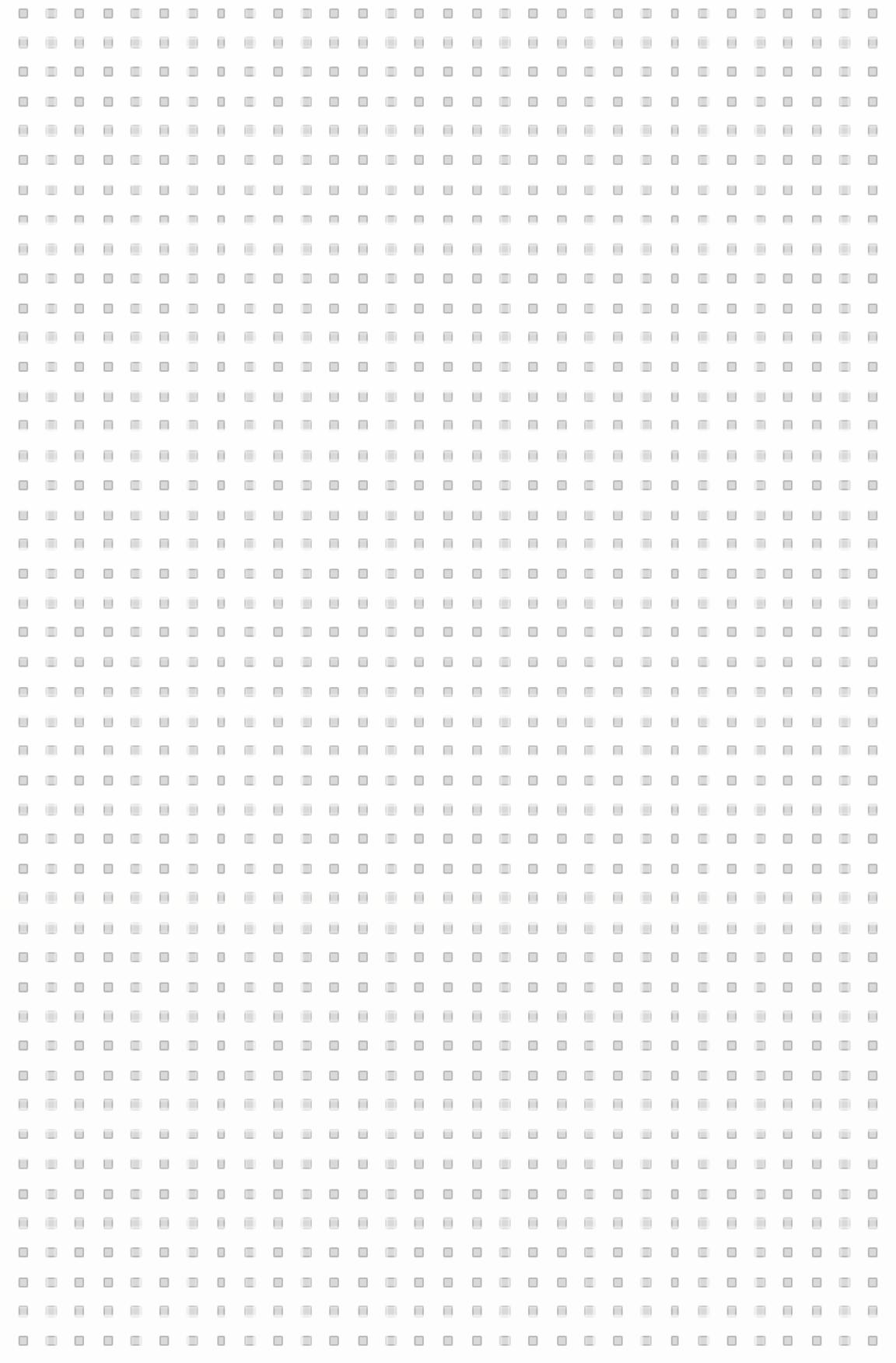
About me

Maximiliano Firtman



About me

Maximiliano Firtman



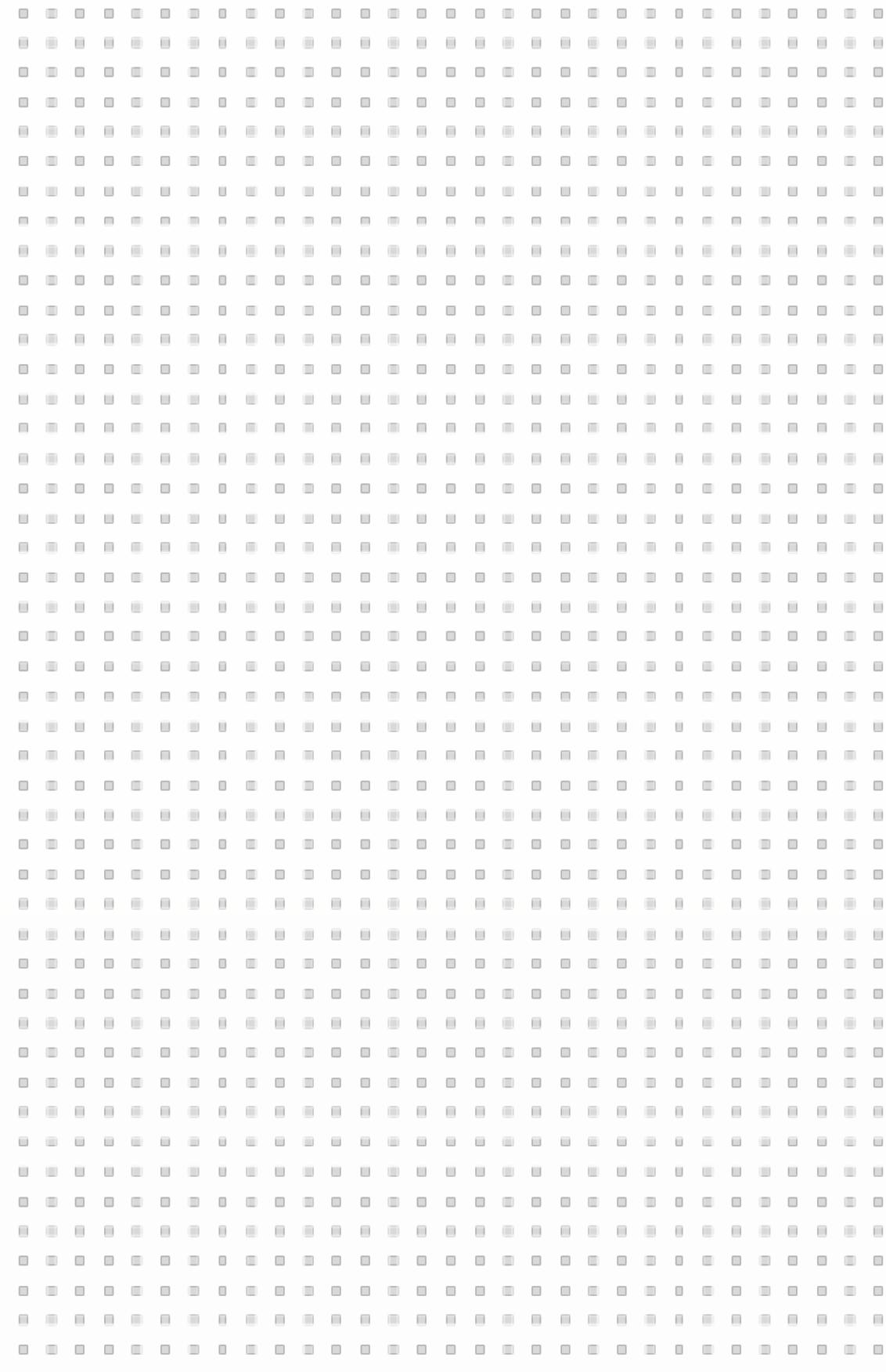
Frontend **Masters**

Frontend Courses

Mobile App Courses

Backend Courses

What we will cover today

- 
- What's a PWA or Installable Web App
 - State of the Platform
 - Mobile and Desktop Installation
 - Making the Web App Offline
 - UI and UX Techniques
 - Publishing and Distribution

Pre-requisites

Web app development

Basic JavaScript

A Chromium desktop web browser

A code editor

Optional: Xcode (on macOS)

Optional: Android Studio

Workshop Repo

github.com/firtman/frontendmasters-pwa

Questions?



Introduction

Question for You



¿What is a Progressive Web App?

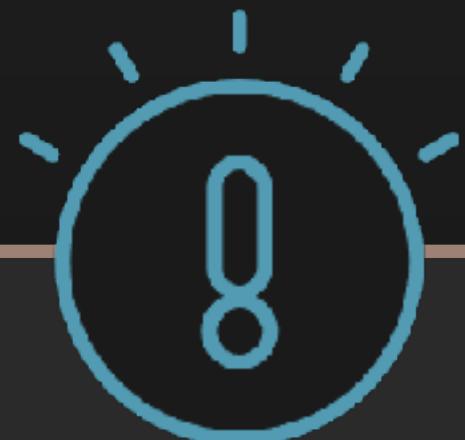
Definition

Progressive Web App (PWA)



Design Pattern to use different APIs and specs from the Web Platform to create installable, offline-capable standalone apps for mobile and desktop devices.

Important



Ideal for all those apps that consume web content or web services.

Warning

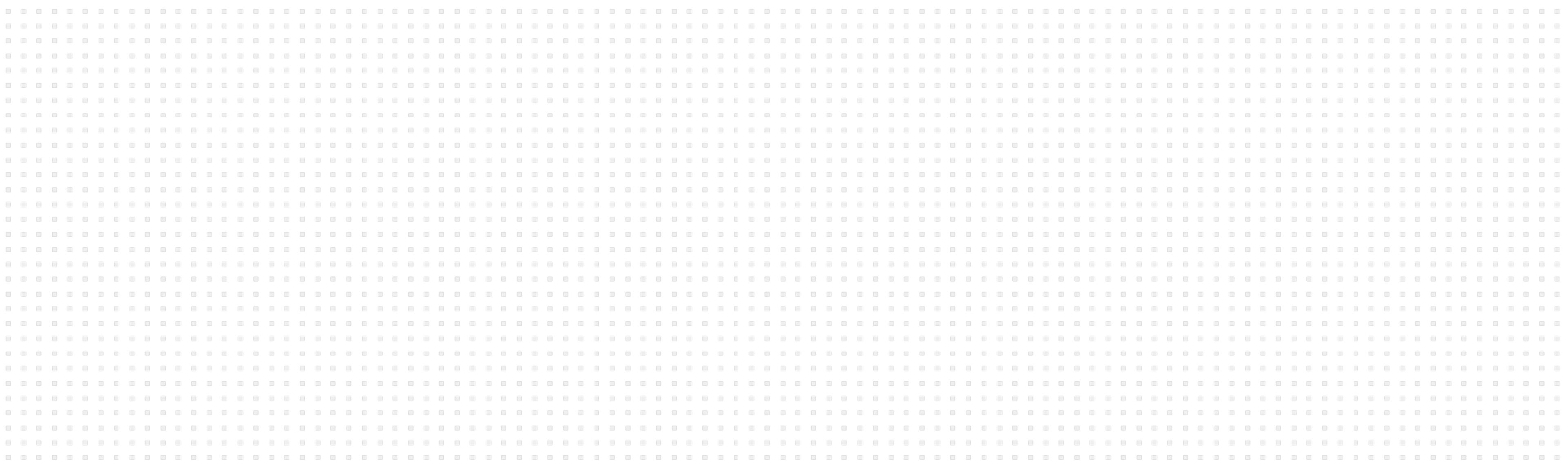


It's not an API or a framework, just a term to describe a design pattern

Web Apps, Installable Web Apps, Homescreen Web Apps



App Development Approaches



App Development Approaches

Native SDKs

App Development Approaches

Native SDKs

Web Development

App Development Approaches

Native SDKs

Device

Web Development

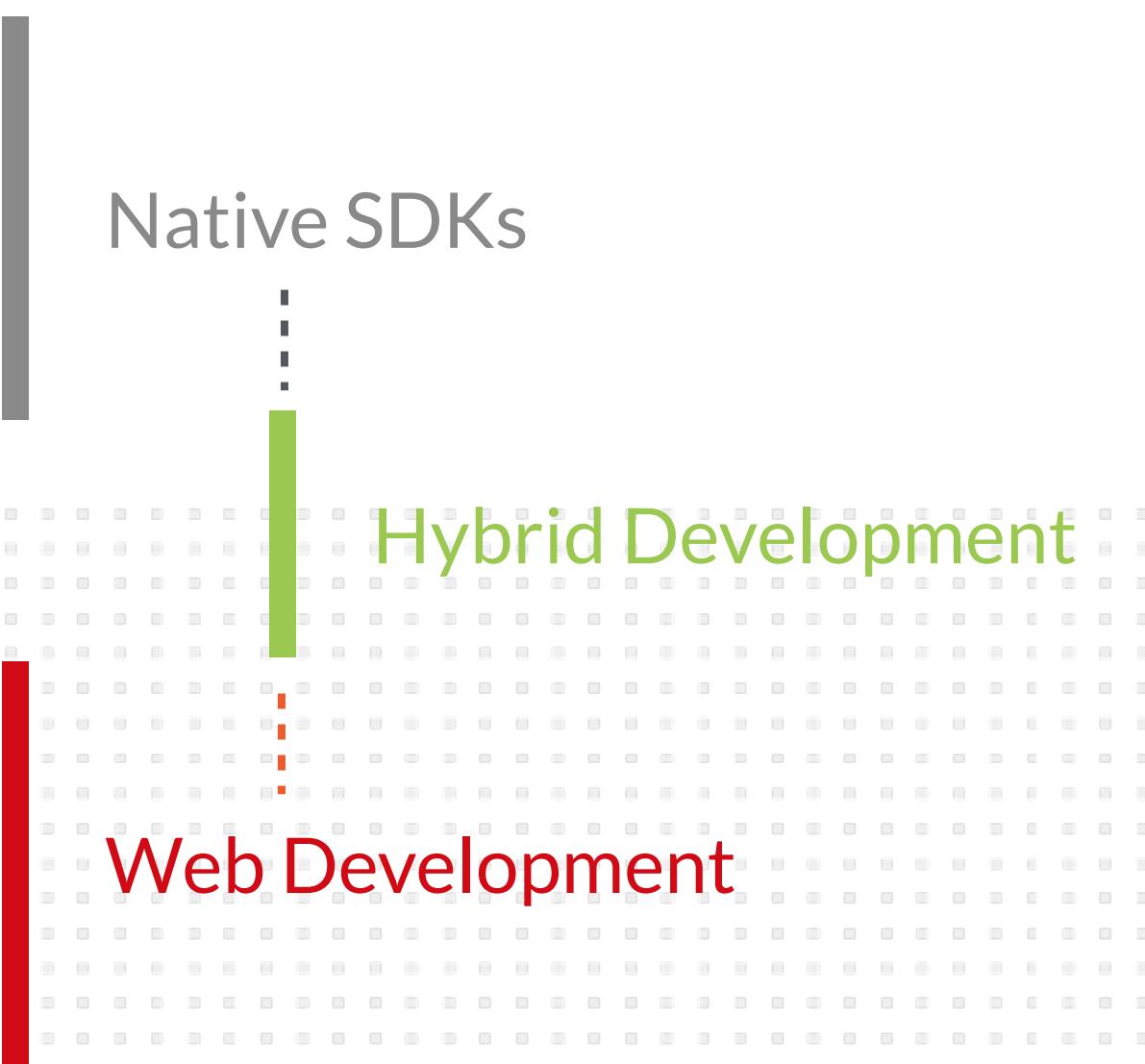
App Development Approaches

Native SDKs

Web Development

Browser

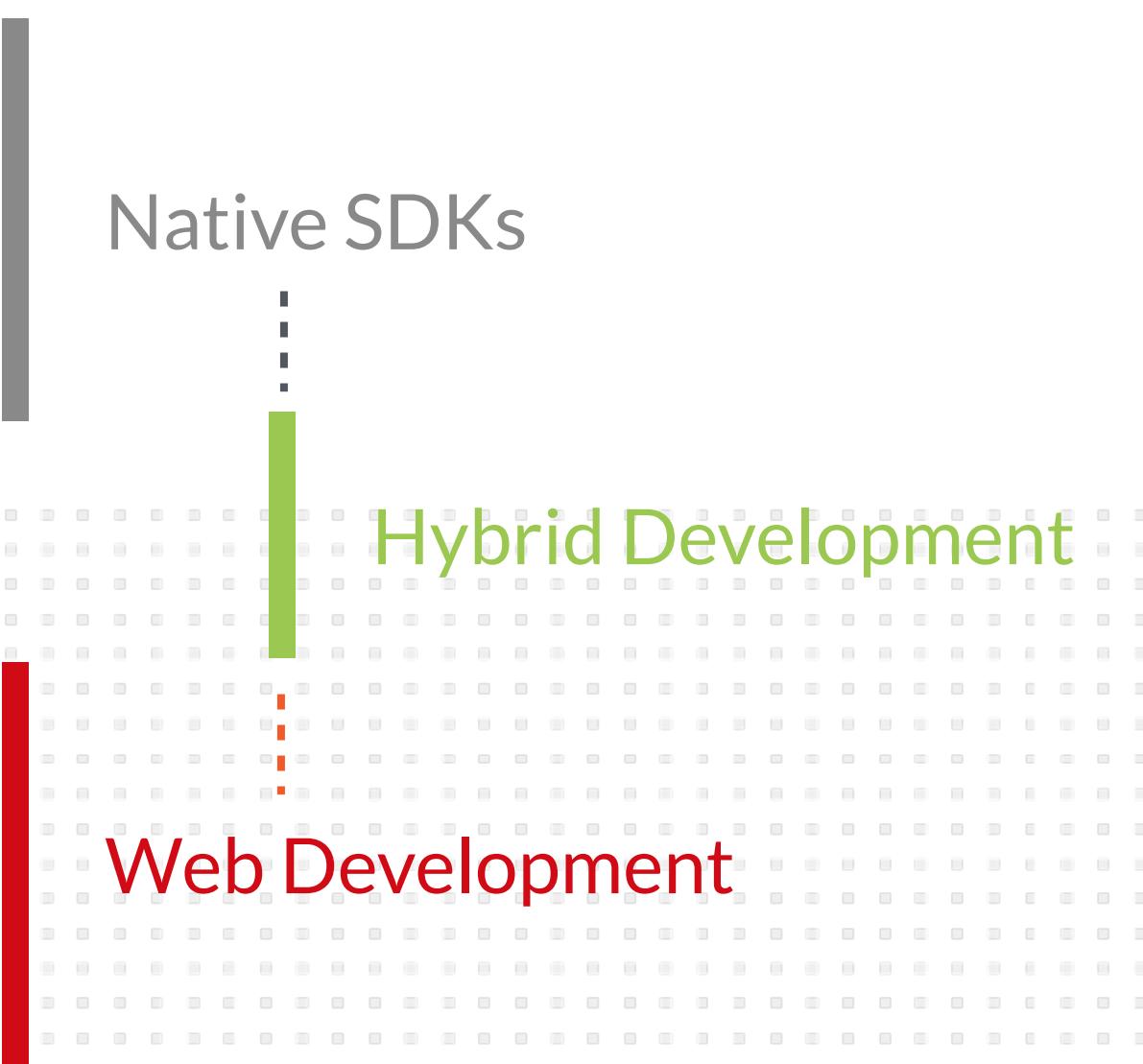
App Development Approaches



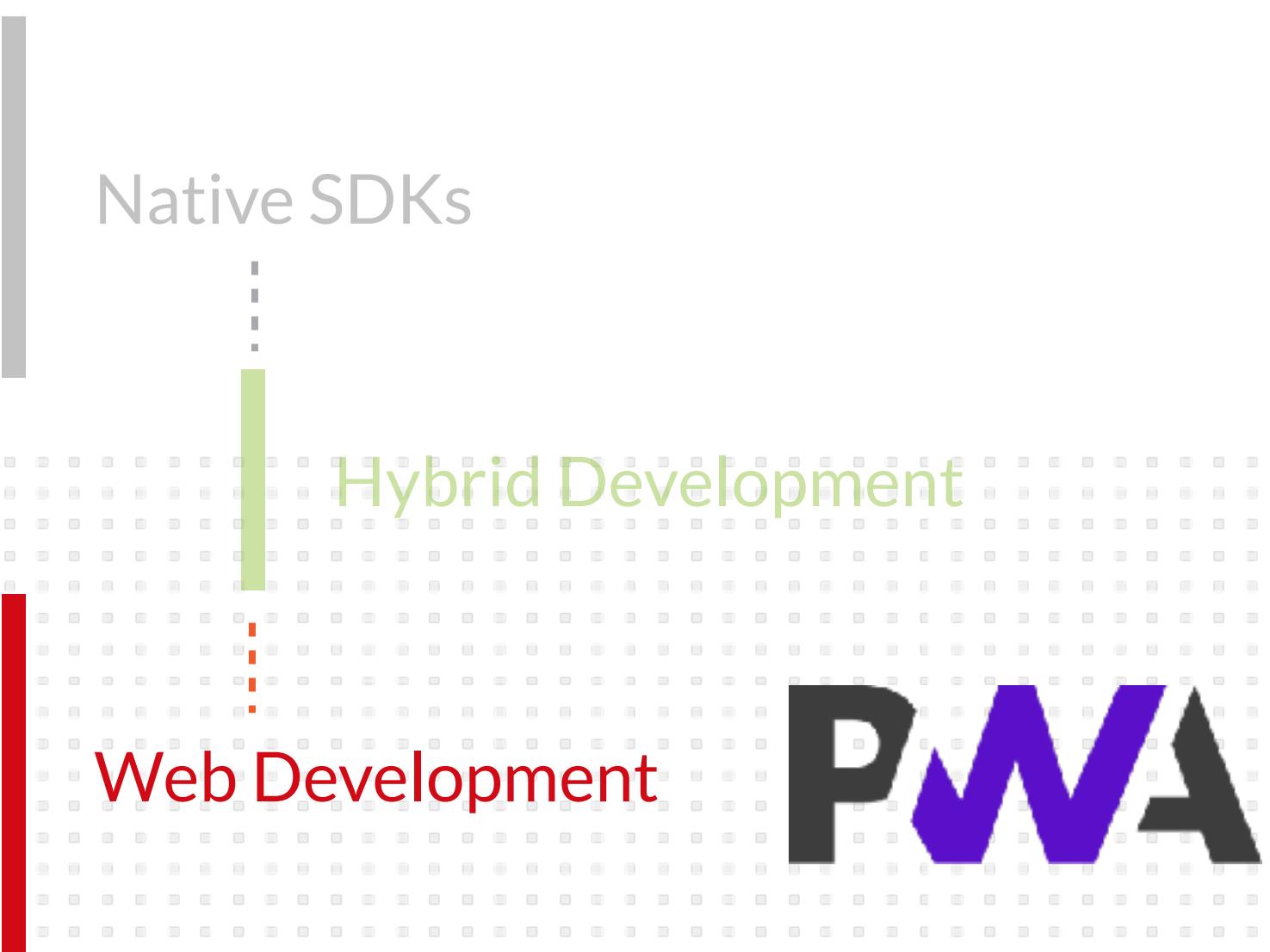
App Development Approaches



App Development Approaches



App Development Approaches



App Development Approaches



App Development Approaches



Best of both universes

Web

Native

From the Web Universe

Links and discoverability

Easy to deploy

Easy to update

Standards and tools

From the Native Universe

Offline access

Installed icon and standalone

OS Integration

Performance and UX

Best of Both Universes

Links and discoverability

Easy to deploy

Easy to update

Standards and tools

Offline access

Installed icon and standalone

OS integration

Performance and UX

State of the Platform

- Every browser 🤔

State of the Platform

firt.dev/notes/pwa

- **Android**
 - All web browsers
- **iOS and iPadOS**
 - Safari
 - Other browsers: with a twist
- **Huawei HarmonyOS**
 - Native Browser
- **Meta Horizon OS**
 - Meta Store

State of the Platform

firt.dev/notes/pwa

- **macOS**
 - Safari, Chrome, Edge
- **Linux**
 - Chrome, Edge
- **ChromeOS**
 - Chrome
- **Windows 7, 8.x, 10, 11**
 - Chrome, Edge



Find your ideal hotel and compare prices
from different websites

Q e.g. Paris

Search

Exploring destinations? I see myself...



sunning at the beach



exploring a city



by the fire, après-ski



hiking through a forest

Activate Windows
Go to Settings to activate Windows Evaluation

valid for 88 days

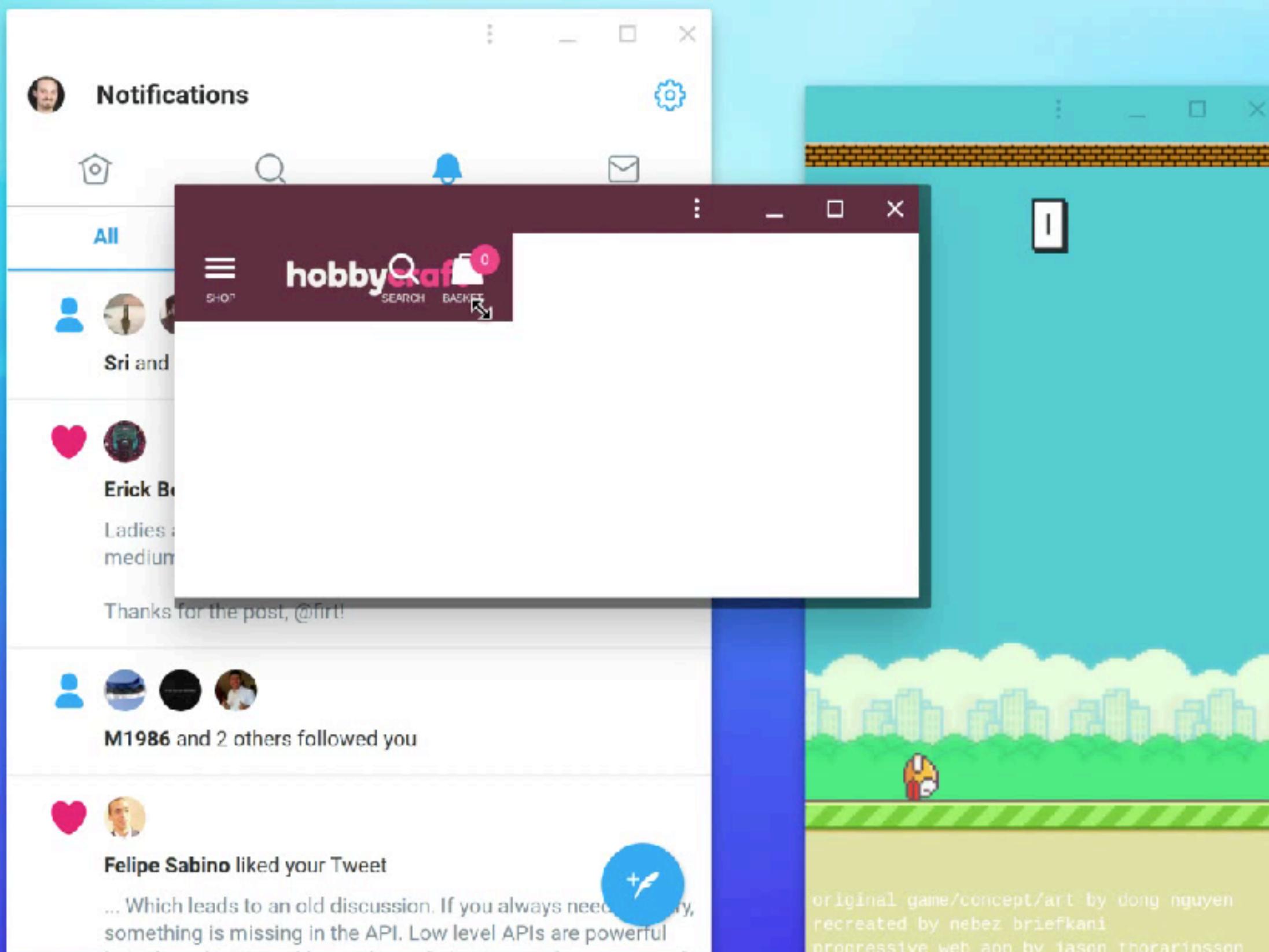
build 17134.154_release.180410-1804



Type here to search



6:56 AM
10/19/2018



Notifications

All

Sri and

Erick Bo

Ladies medium

Thanks for the post, @firt!

M1986 and 2 others followed you

Felipe Sabino liked your Tweet

... Which leads to an old discussion. If you always need something, something is missing in the API. Low level APIs are powerful

original game/concept/art by dong nguyen
recreated by nebez briefkani
progressive web app by jason thorarinsson

FTWeekend



N Korean defectors caution US against Kim's deceptive strategies

China hits back at 'trade bullying' Trump

Beijing accused Donald Trump of "trade bullying" as it imposed retaliatory tariffs against new US

Biogen success offers hope for Alzheimer's treatment

Biogen offered a rare glimmer of hope in the increasingly fruitless field of Alzheimer's treatment on

Germany's odd couple endure fiery political marriage

It was one of the most humiliating moments in Angela Merkel's 13 years as German chancellor – the

FT

Starbucks Coffee Company

ORDER CARDS GIFT STORES

Good morning. ☀

Sign in Join now

STARBUCKS REWARDS™

Recycle Bin

Google Chrome

trivago

Starbucks

App info [app.starbucks.com](#)

Copy URL

Open in Chrome

Uninstall Starbucks...

Zoom - 100% +

Print... Ctrl+P

Find... Ctrl+F

Cast...

Edit Cut Copy Paste

Join now

Learn more

Start an order

12:04 AM
10/17/2018

No support or
plans for it

- Firefox on desktop
- visionOS
- Android WebView (Instagram, TikTok, etc.)
- iOS WebView (Instagram, TikTok, etc.)
- watchOS
- tvOS
- Android TV

Components of a PWA

Web App

Web App Manifest

Service Worker

Important



The concept is changing with time and it's becoming more flexible

PWA Levels

- Default
- Basic Metadata provided
- PWA Criteria Approved
- Full offline experience

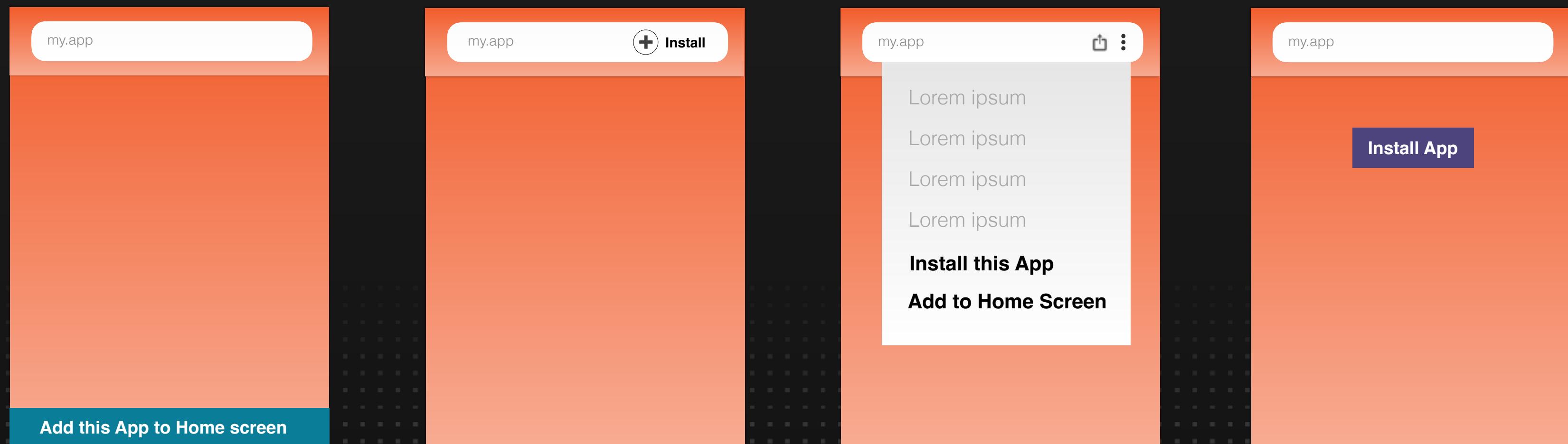
Definition

PWA Criteria



Defined by the browser. If passed, there are hints, banners, dialogs or bars explaining to the user that the current web app is installable and inviting them to install it.

Launcher Icon Installation from Browser



Custom UI

Installability Criteria for Safari on iOS and iPadOS

- Includes meta data with:
 - App's name
 - Icons
 - Start URL
 - Display mode

Installability Criteria for Safari on macOS



Installability Criteria for Safari on macOS

No requirements

Installability Criteria for Chromium

- Web app not already installed
- HTTPS
- Meet basic engagement heuristics
- Includes meta data with:
 - App's name
 - Icons
 - Start URL
 - Display mode

Installability Criteria for Firefox (Android-only)

- Web app not already installed
- HTTPS
- Includes meta data with:
 - App's name
 - Icons
 - Start URL
 - Display mode

Warning



On mobile if you don't meet the basic criteria, standalone installation won't happen. On desktop there are ways to install them as web app.

Our Lab

- Codepad Masters
- Store notes while you are coding, such as
 - - "I need to learn more about Service Workers"
- It should work offline

Our Lab

- Vanilla web app
- HTML + CSS + JavaScript
- No library, no framework
- No build process
- But it's not yet a Progressive Web App

Workshop Repo

github.com/firtman/frontendmasters-pwa

Lab time



The Web App Manifest

Definition

Web App Manifest



It's a JSON file that defines all the meta data that will help the browser and the OS to install the app and have integration into the operating system.

Web App Manifest

- W3C spec
- JSON file
- Default extension: .webmanifest
- MIME type: application/
manifest+json) or application/json
- We have to link it from the HTML
(all the documents you want)

Web App Manifest Basics

index.html

X

```
<link rel="manifest" href="web.manifest">
```

Idea



You can have more than one installable web app (PWA) in the same origin, if you point to different Web App Manifest files. It's recommended to host them in different folders.

It allows to the micro-app architectural pattern.

Web App Manifest Basics

app.webmanifest

```
{  
  "id": "/",
  "name": "My PWA",
  "short_name": "PWA",
  "display": "standalone",
  "scope": "/",
  "start_url": "/?pwa",
  "theme_color": "orange",
  "background_color": "white"
}
```

X

Idea



There is no standard way to internationalize your web app's metadata using the manifest. You can link to several manifests or create the manifest dynamically from the server.

Display modes

- browser
- standalone
- fullscreen
- minimal-ui

On a separate spec

- tabbed
- window-controls-overlay

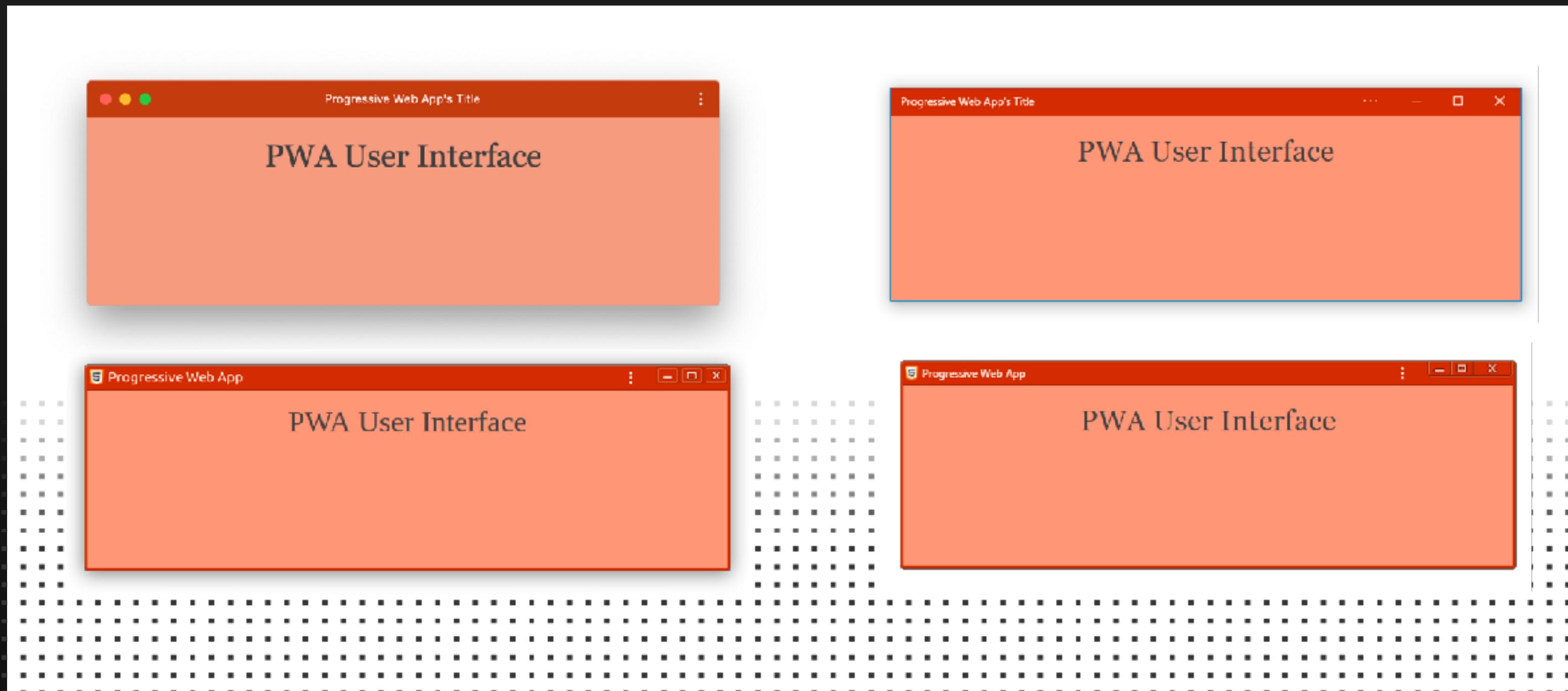
Warning



For backward compatibility you can use a standard mode in display and other modes in display_override as in:

```
"display_override": [ "tabbed" ]
```

Desktop Standalone mode



Desktop Tabbed mode

Tabbed Application Mode

Page 1 | Page 2 | +

Tabbed Application Mode

Home



Current URL: <https://tabbed-application-mode.glitch.me/>

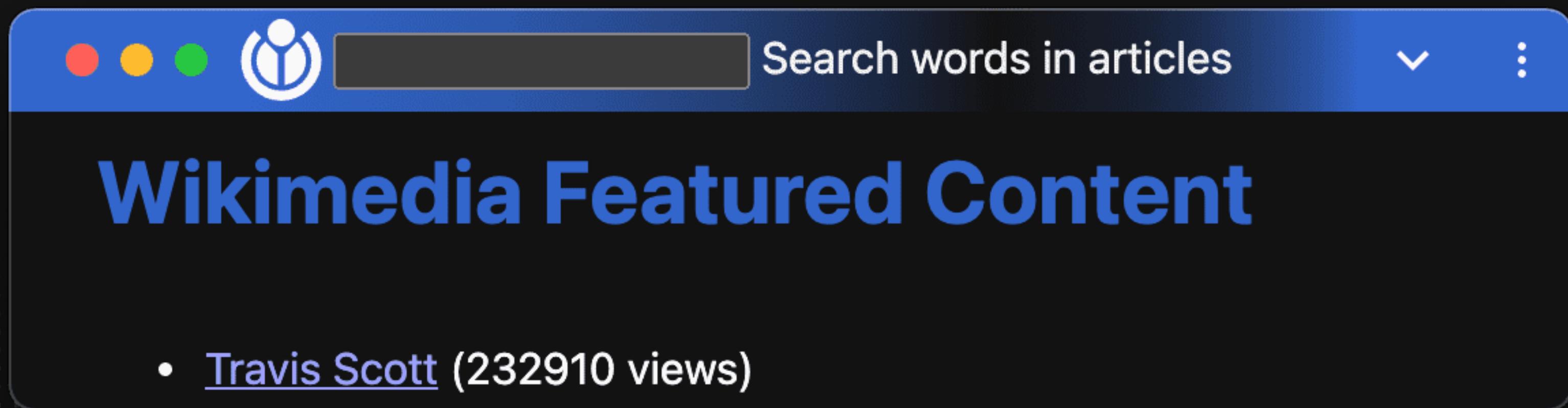
On the pinned Home tab, this link reloads the pinned Home tab:

[Home](#)

On the pinned Home tab, these links open in new tabs, and never navigate the pinned Home tab:

[Page 1](#)
[Page 2](#)

Desktop Windows Controls Overlay mode



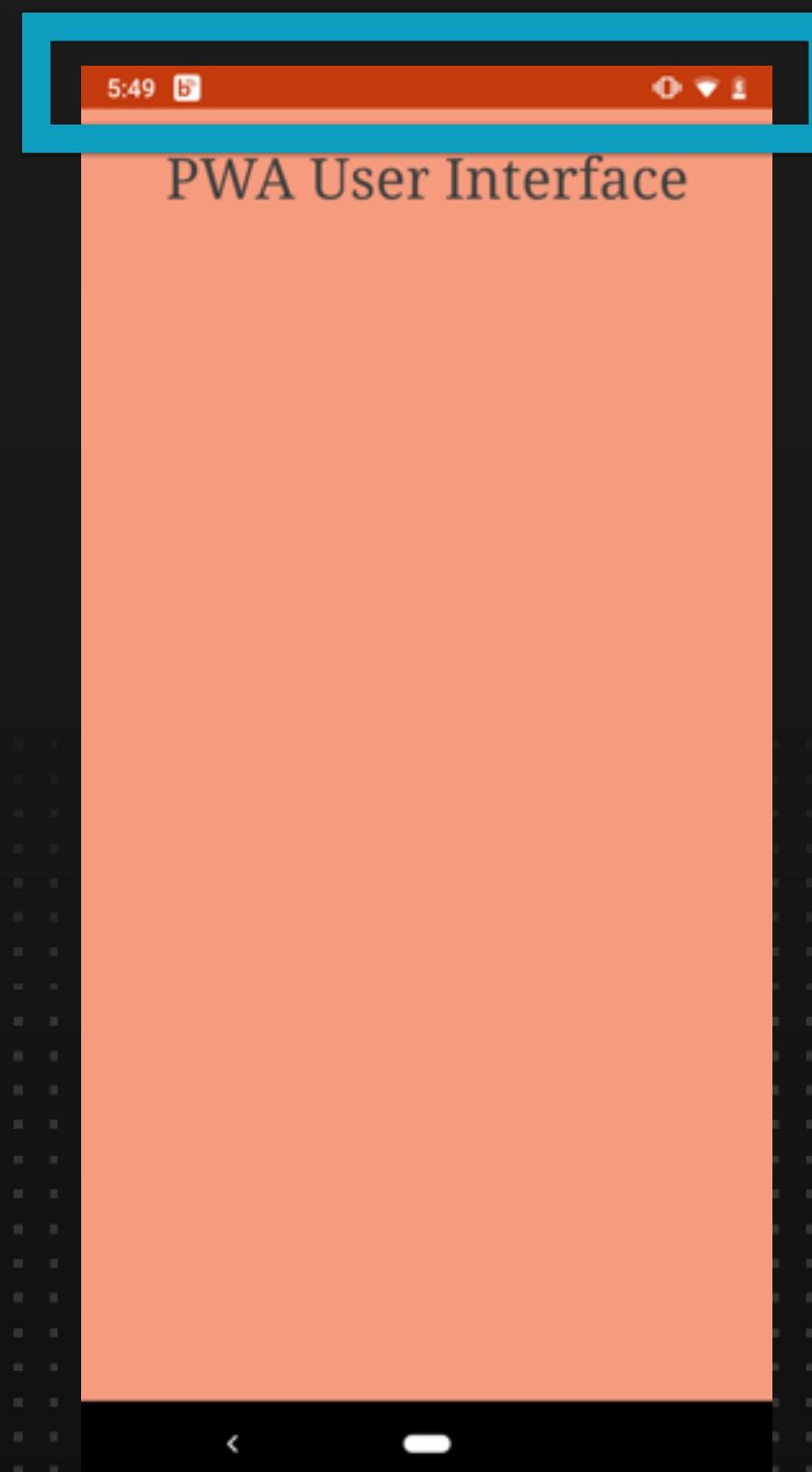
iOS and iPadOS Standalone Experience



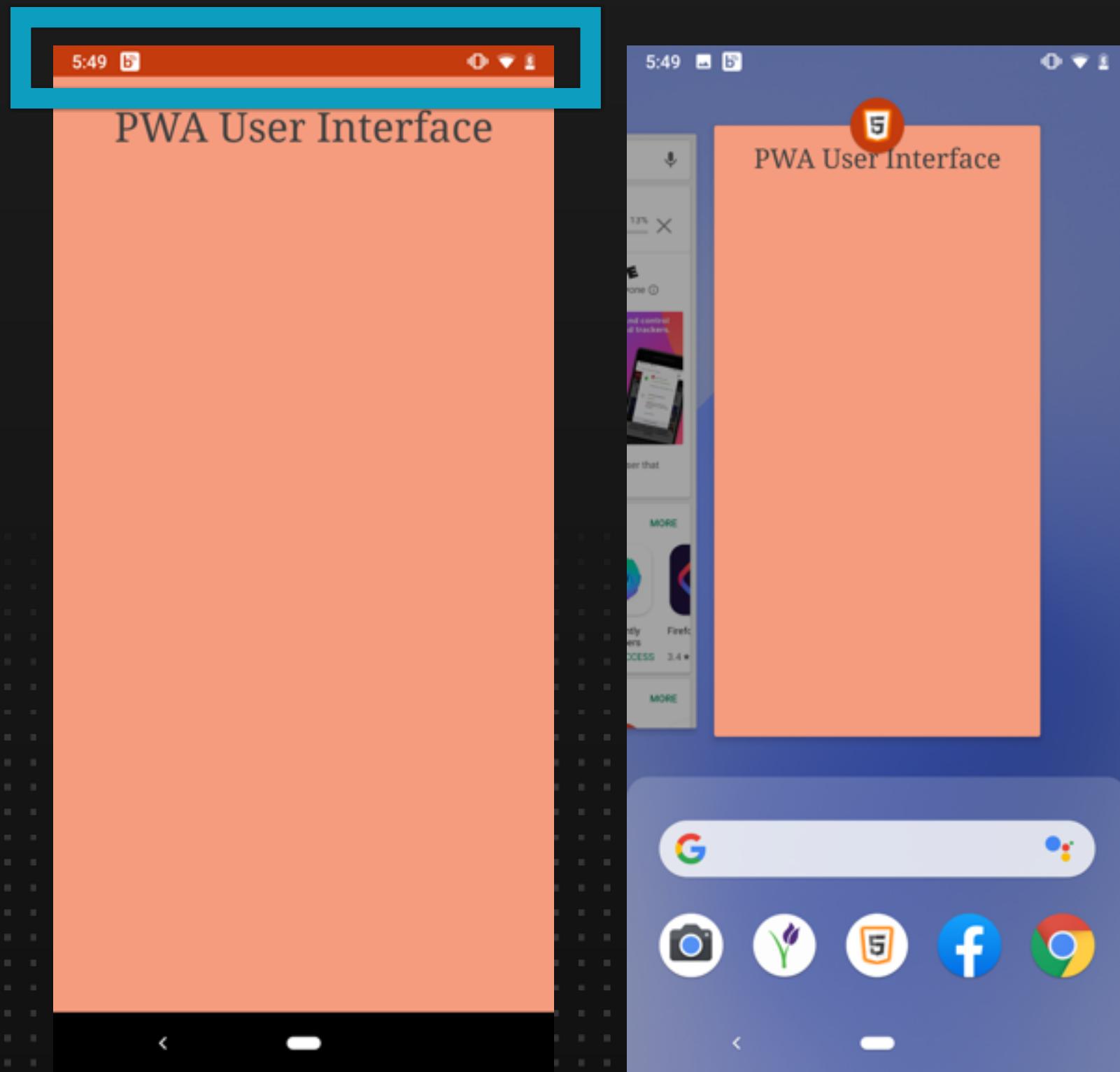
Android Standalone Experience



Android Standalone Experience



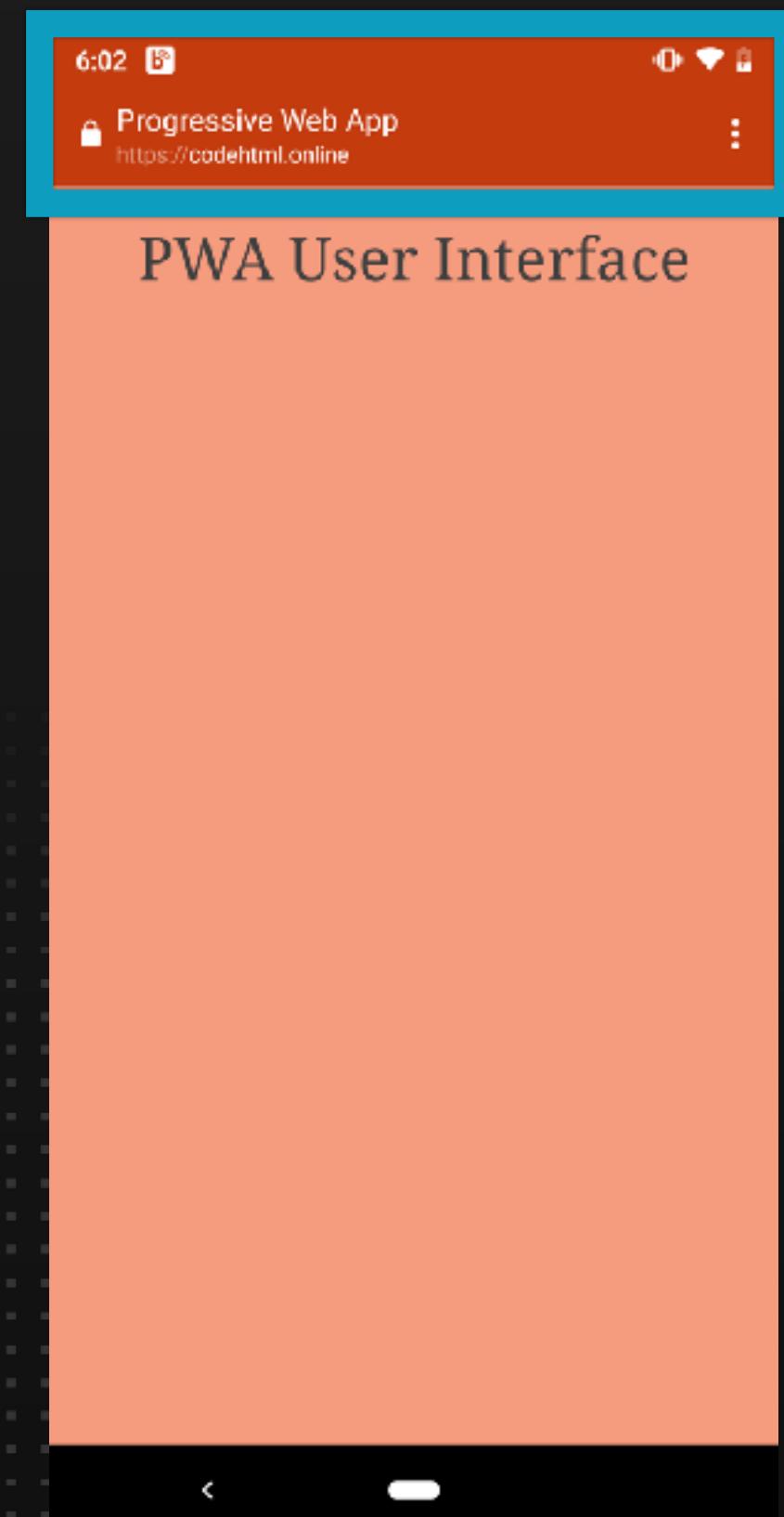
Android Standalone Experience



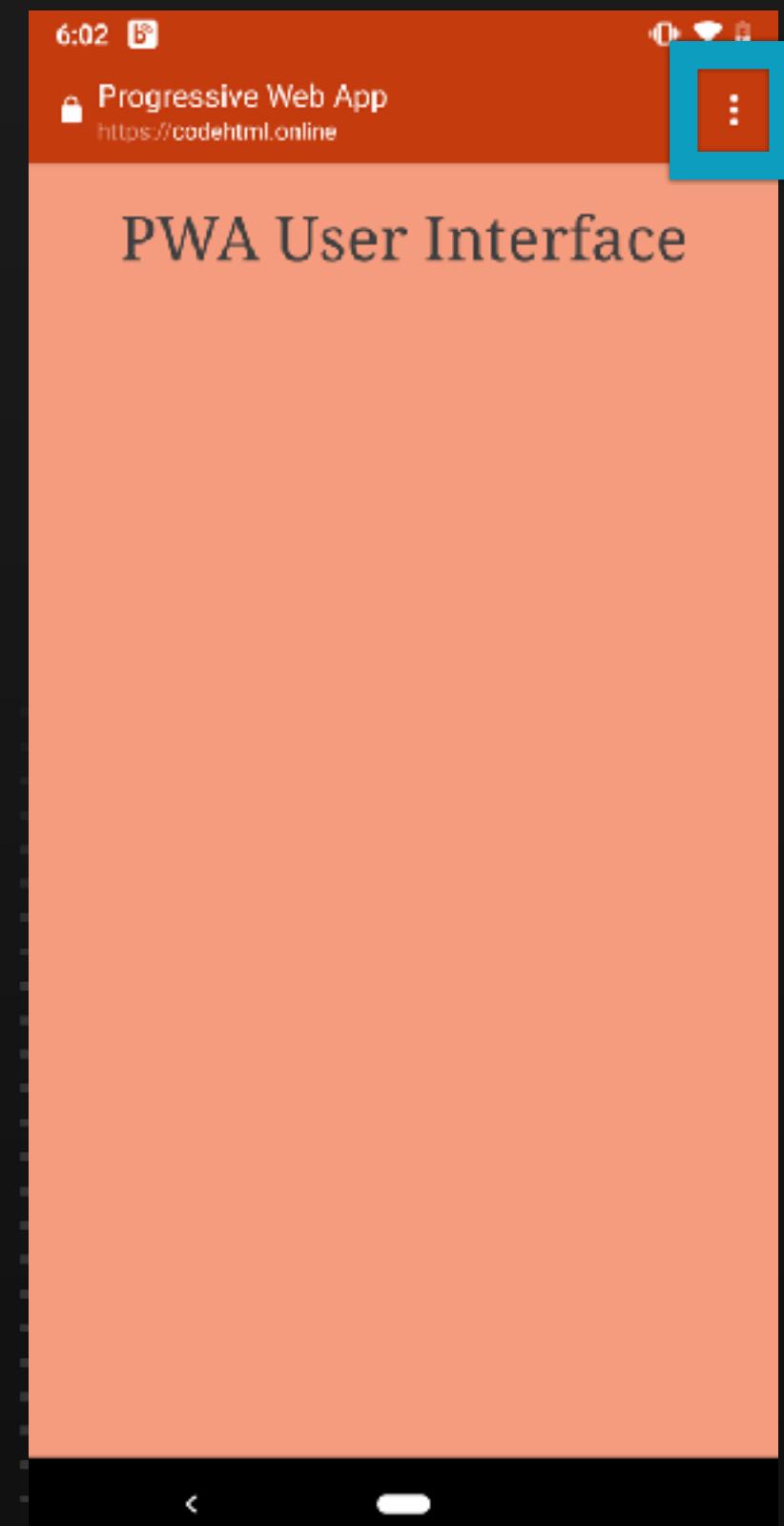
Android Minimal UI Experience



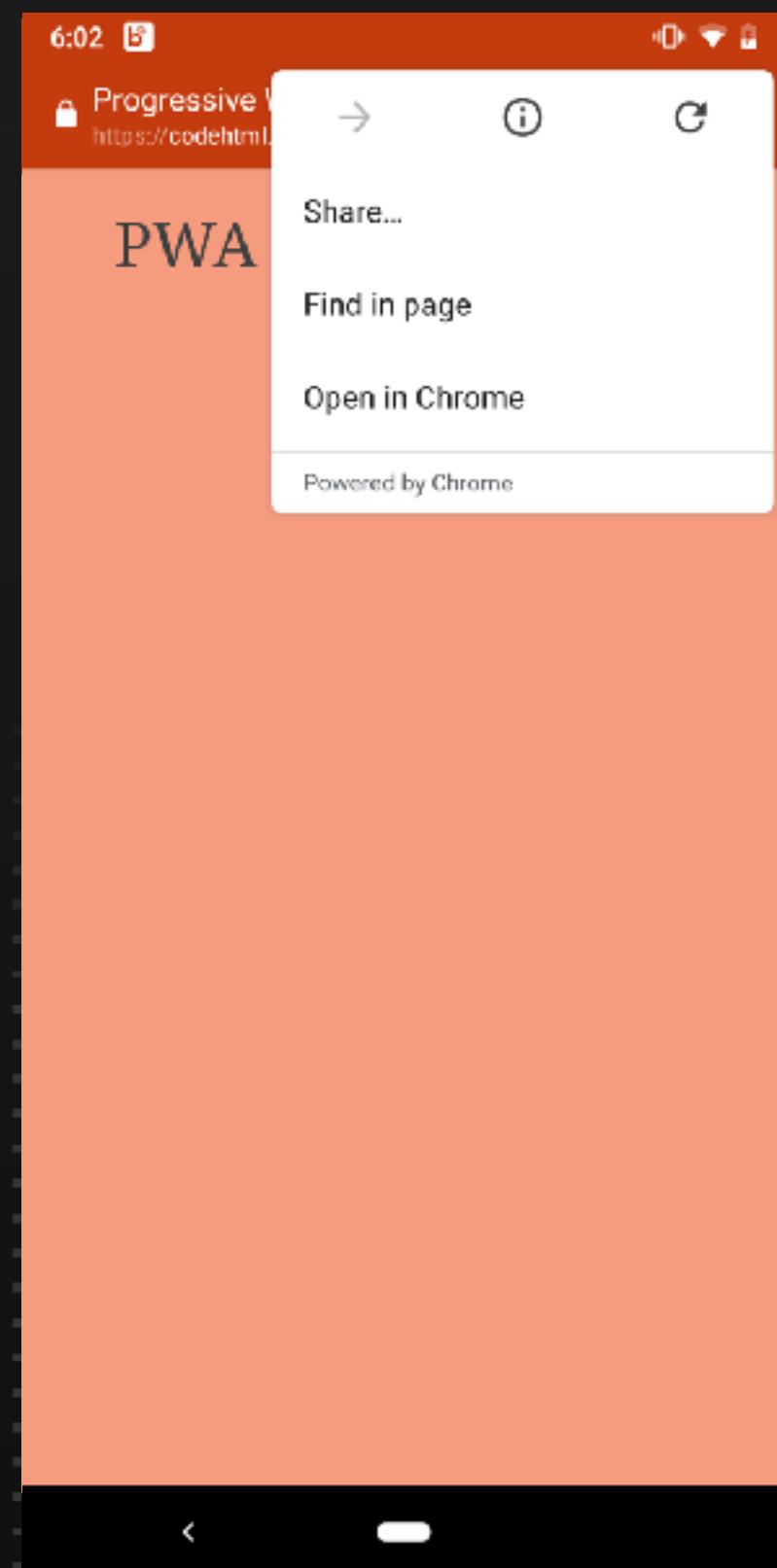
Android Minimal-UI Experience



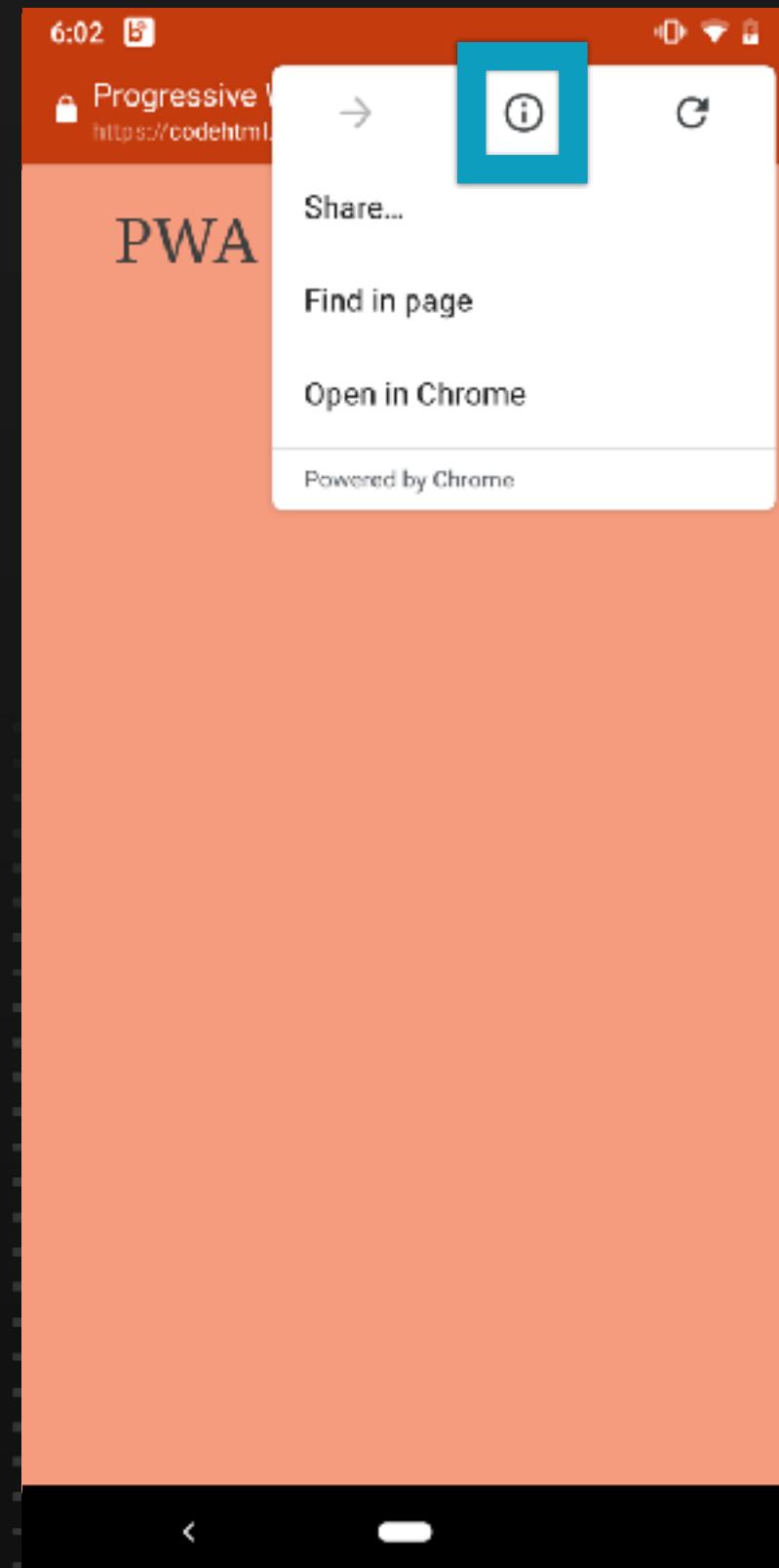
Android Minimal-UI Experience



Android Minimal-UI Experience



Android Minimal-UI Experience



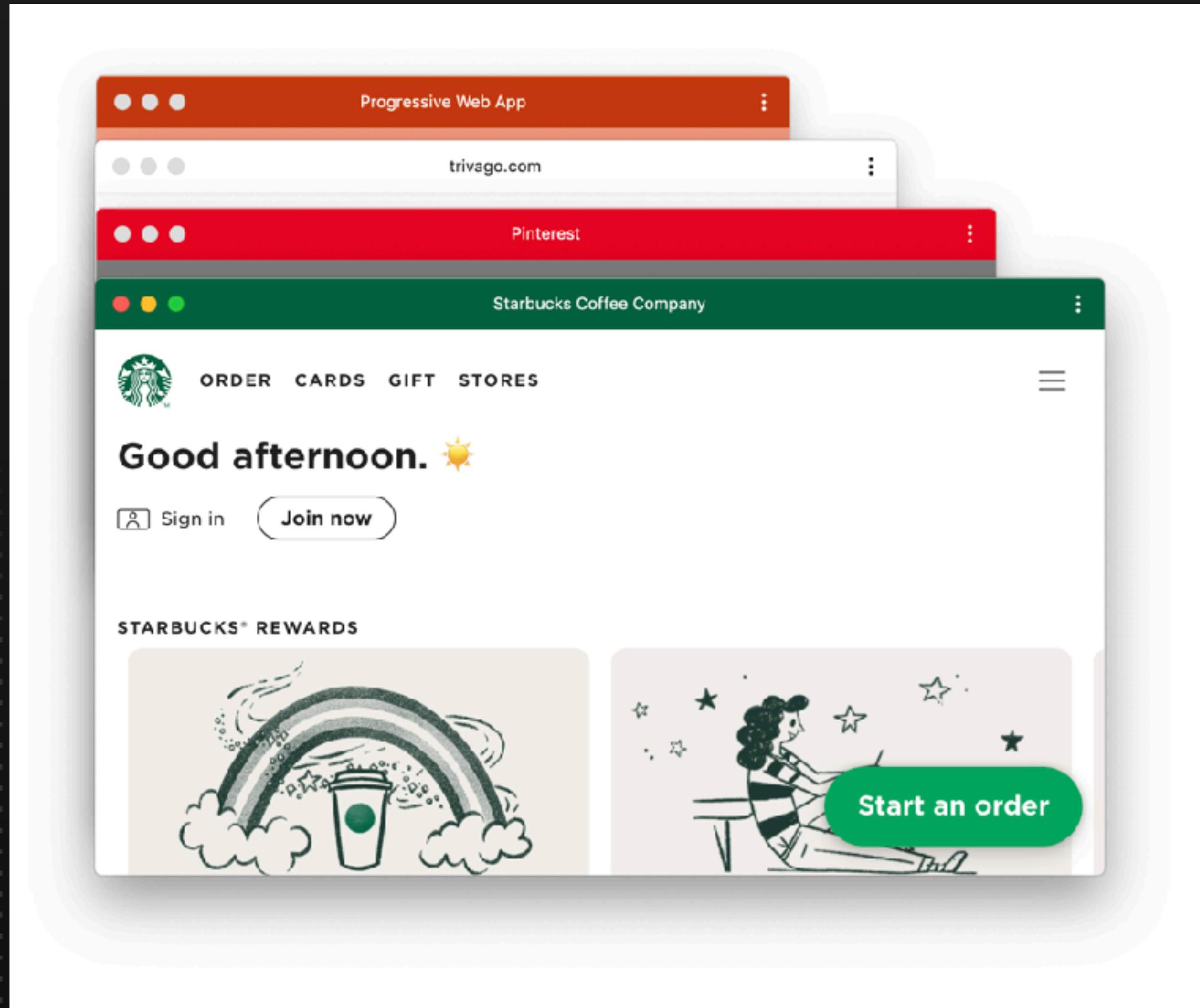
Android Full Screen Experience

PWA User Interface

Display modes on Android



The theme-color





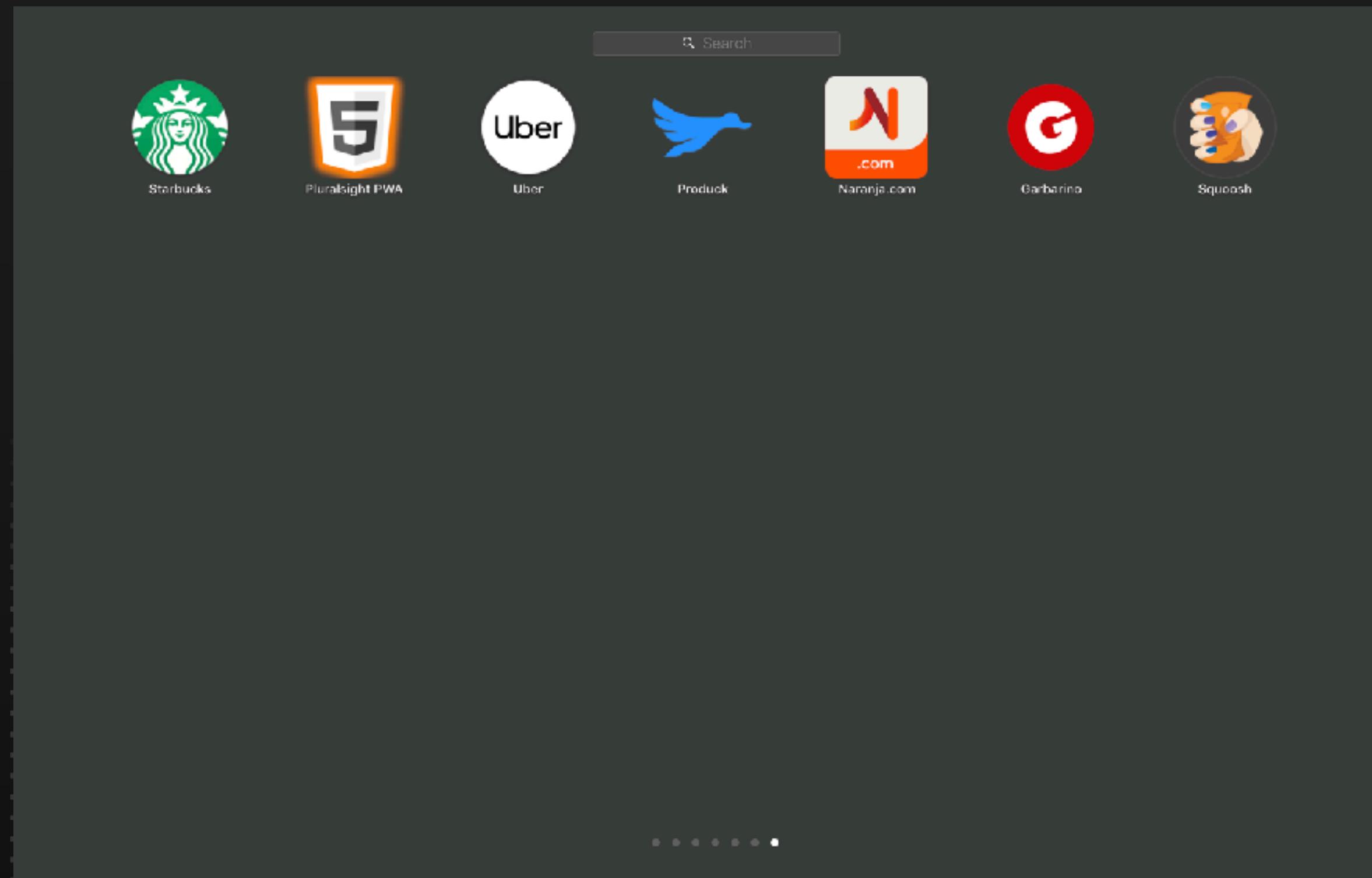
Testing and Debugging

Let's practice



Standalone User Experience

Desktop Standalone Icon



Icons on Android

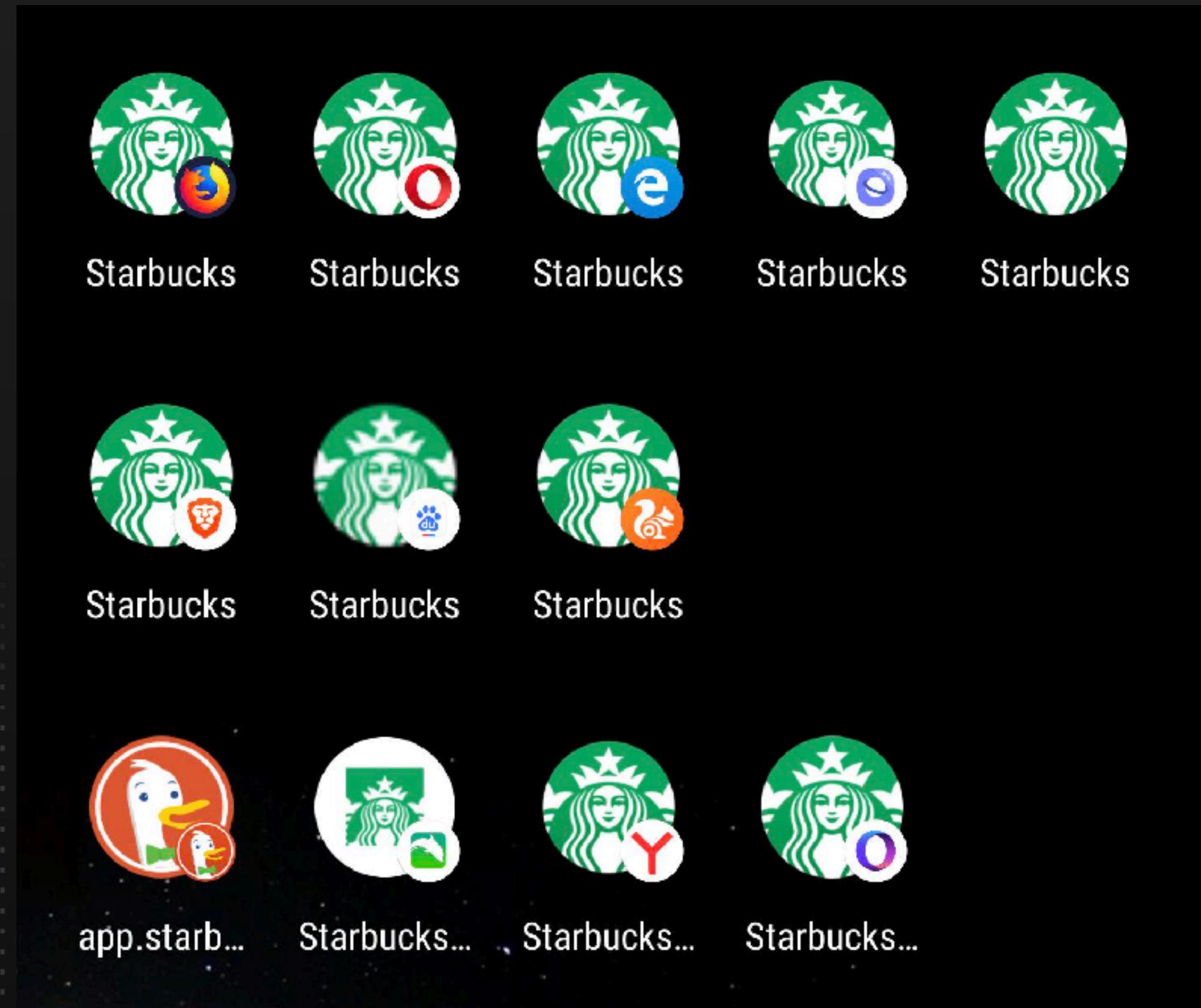
Shortcut

Web APK

Shortcut Icons on Android

- Creates a shortcut to browser's engine
- It's installed only at the home screen
- No icon in the launcher
- It doesn't appear at app's list
- All browsers use this method, by default
- Browser's badge (Android 8+)

Shortcut Icons on Android



WebAPK Icons on Android

- Only if PWA Criteria passes
- It's a full Android native package
- It contains only metadata
- APK is installed silently
- Icon to home screen and launcher
- Google Chrome with Play Services
- Internet on Samsung devices
- HarmonyOS Huawei devices
(similar idea)

Icons in the Web App Manifest

app.webmanifest

X

```
"icons": [  
  {  
    "src": "icon.png",  
    "sizes": "192x192",  
    "type": "image/png"  
  },  
  {  
    "src": "icon.png",  
    "sizes": "512x512",  
    "type": "image/png"  
  }  
]
```

Icons in the Manifest

Recommendations

- Format: PNG, Color space sRGB
- Used on Android and desktop OS
- If there is no exact icon available it will pick the closest one
- Recommended sizes:
 - At least: 192x192, 512x512, 1024x1024
 - Deprecated sizes: 72x72, 152x152, 384x384

Icons in the Manifest

Recommendations

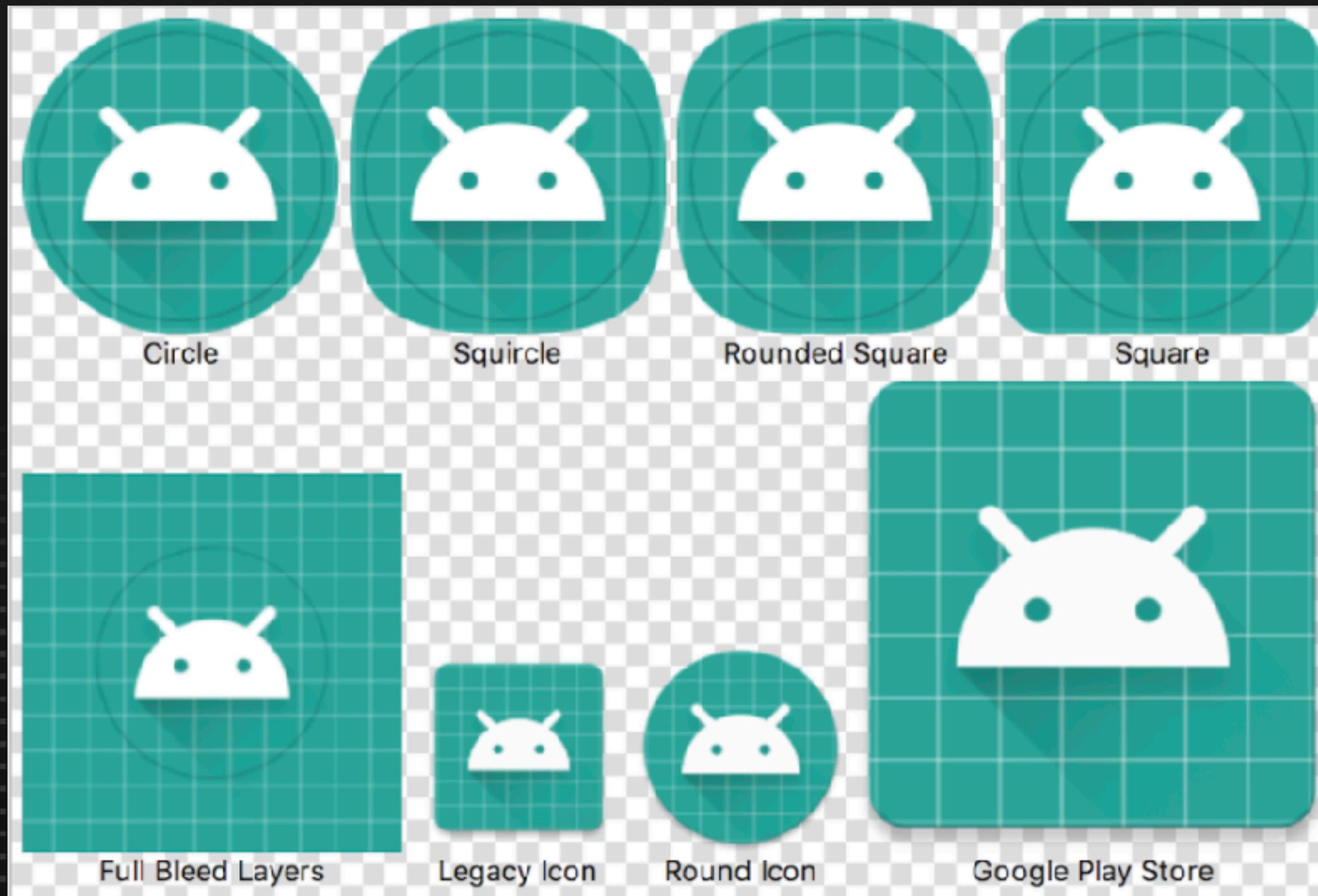
- Format: SVG
- Used on Android Web APK on some devices
- Used on desktop on some operating systems.

Icons in the Manifest

Recommendations

- Format: Maskable PNG
- Used on Android 8+ by default
- Logo is designed with icon masks and safe zone in mind, such that any part of the image that is outside the safe zone can safely be ignored and masked away

Adaptive Icons on Android



Definition

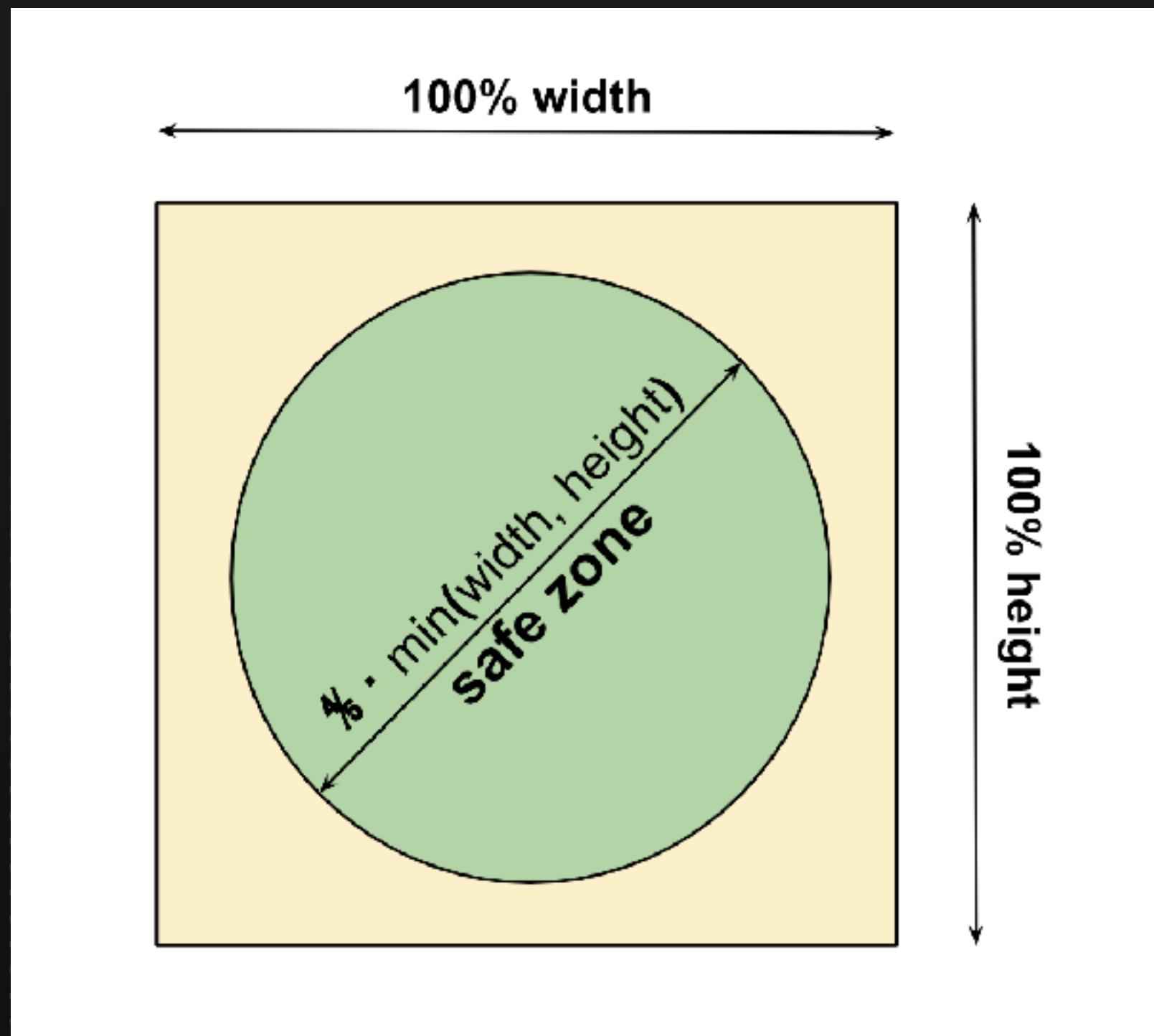
Maskable Icons



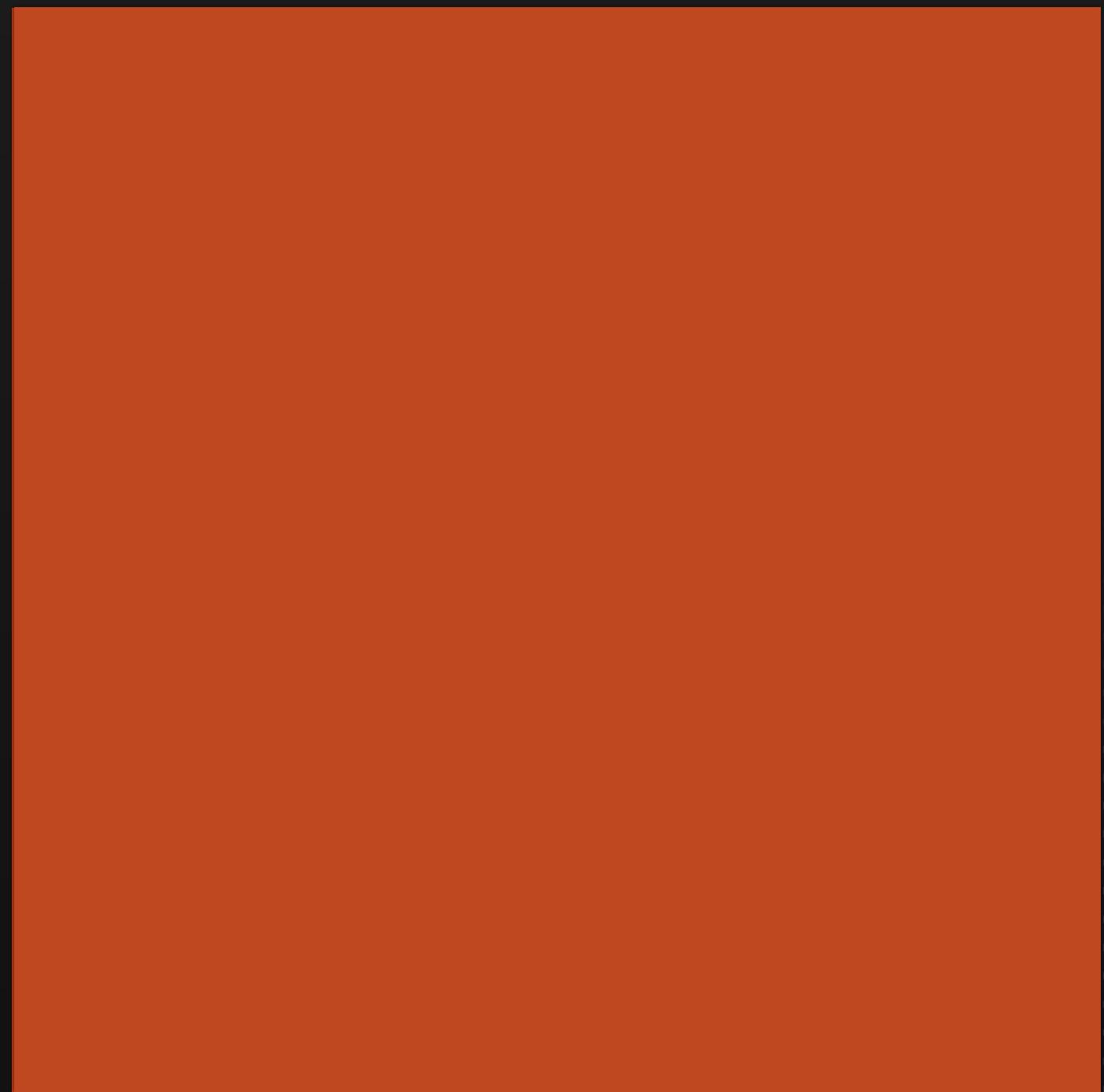
A maskable icon will solve the problem of different shapes; it's still a square icon but with enough discardable padding content out of a safe area

Safe Area on Maskable Icons

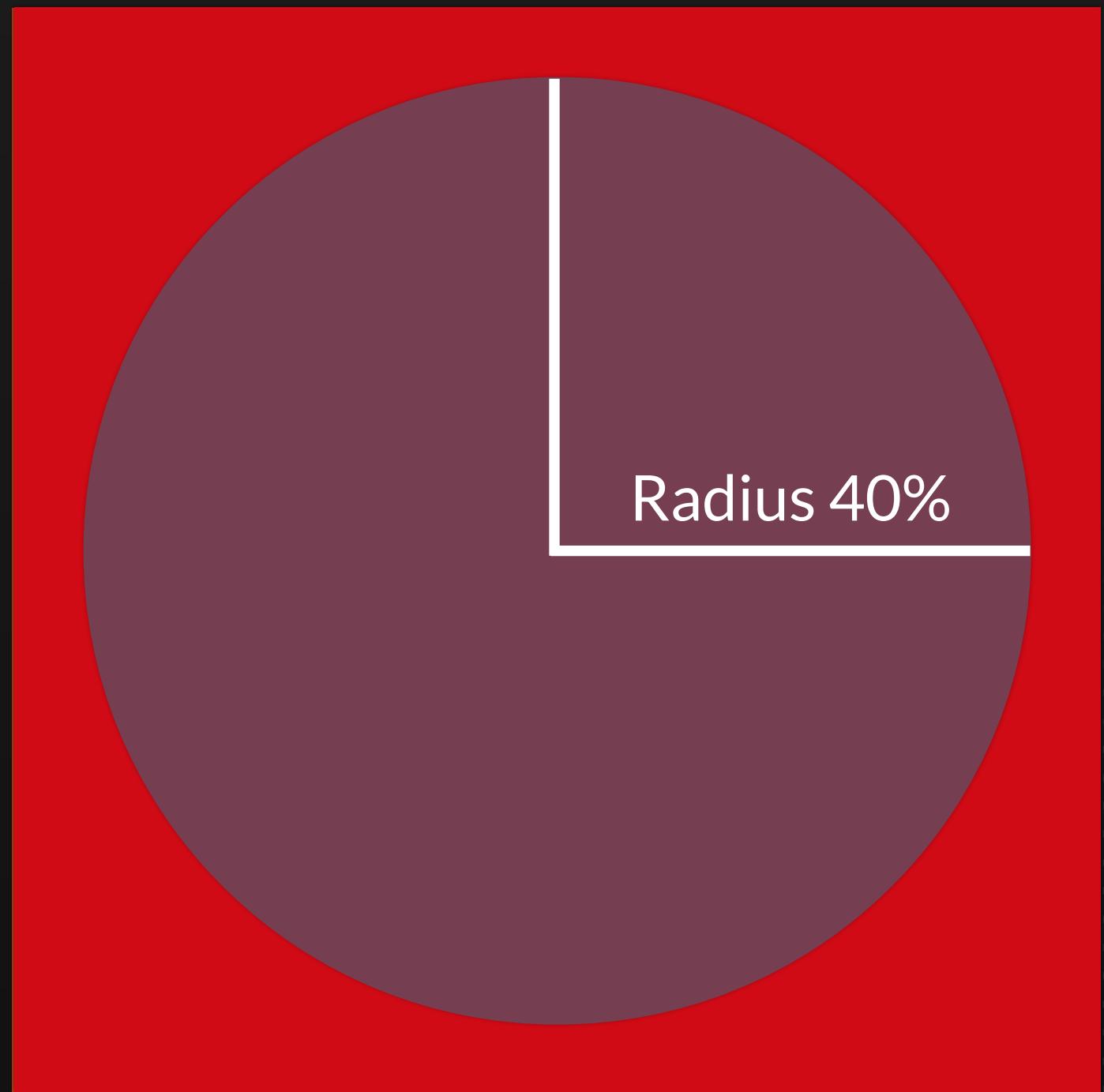
A centrally positioned circle, with radius 40% of the icon's width



Safe Area on Maskable Icons



Safe Area on Maskable Icons

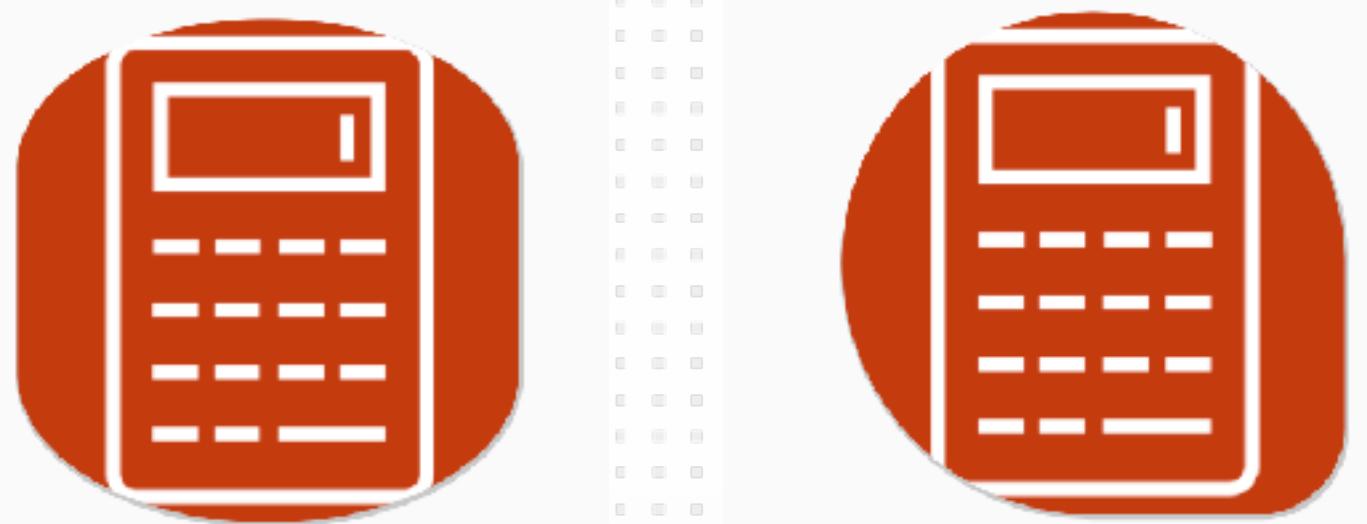


Safe Area on Maskable Icons



Safe Area on Maskable Icons





Safe Area on Maskable Icons



Safe Area on Maskable Icons



Safe Area on Maskable Icons





Maskable icons

app.webmanifest

```
"icons": [  
  {  
    "src": "icon.png",  
    "sizes": "192x192",  
    "type": "image/png"  
  },  
  {  
    "src": "maskable.png",  
    "sizes": "512x512",  
    "type": "image/png",  
    "purpose": "maskable"  
  }  
]
```

X

Maskable.app



[Open icon file](#)



Masks

- None
- Minimum safe area
- Circle
- Rounded Rectangle
- Square
- Drop
- Cylinder

Controls

- Shrink
- Show ghost image

Web App Icons for iOS and iPadOS

- They will use an icon from the manifest
- We can specify an specific icon for these platforms by using a <link> in the HTML element

Maskable icons

If one icon is provided, 180x180 is the recommended size. You can use the sizes property.

index.html

```
<link rel="apple-touch-icon" href="ios_icon.png">

<link rel="apple-touch-icon" href="ios_icon.png"
      sizes="120x120">
```

Definition

Splash Screen

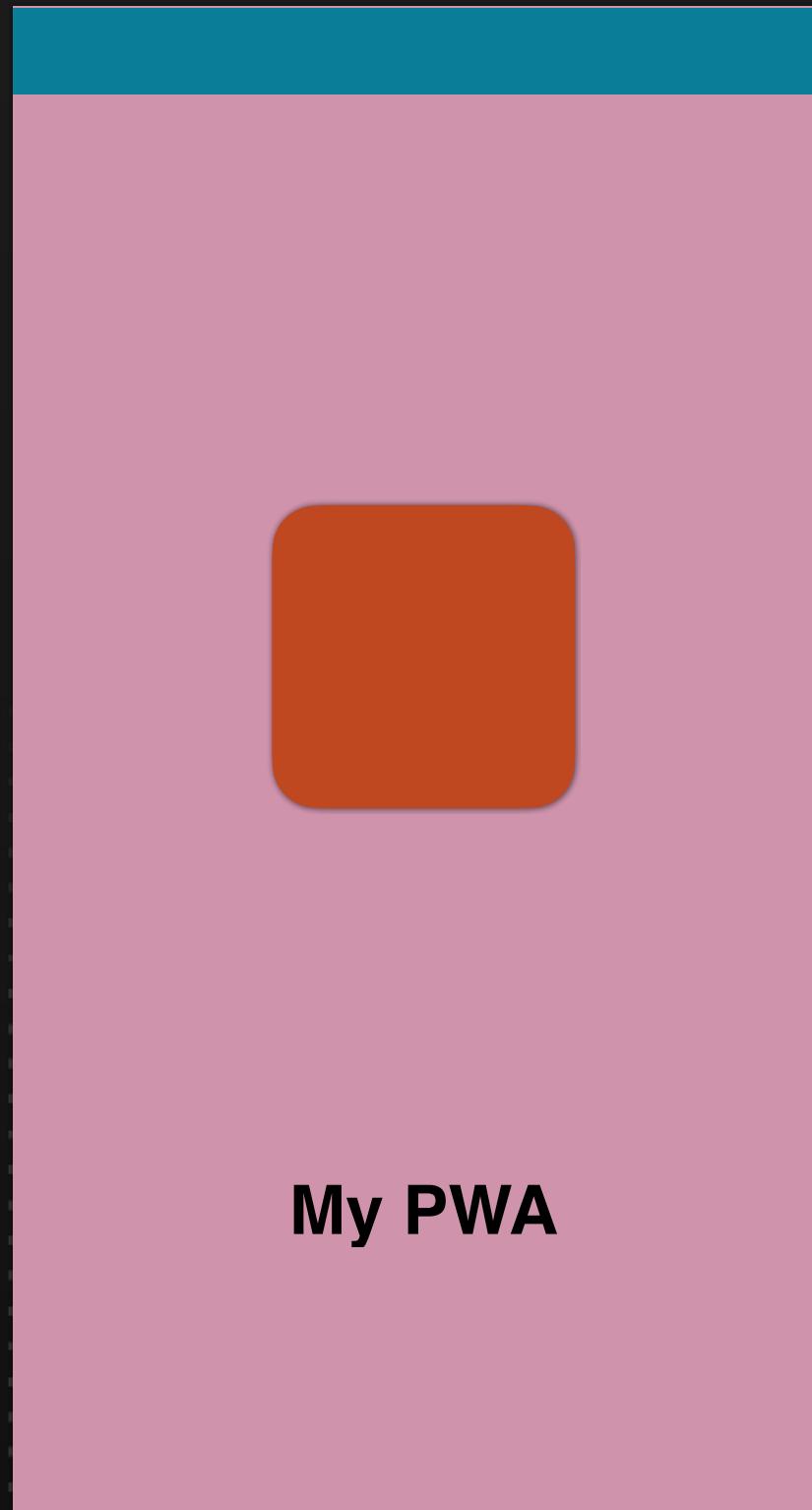


Mobile app design pattern that renders a static image while the app process is loading

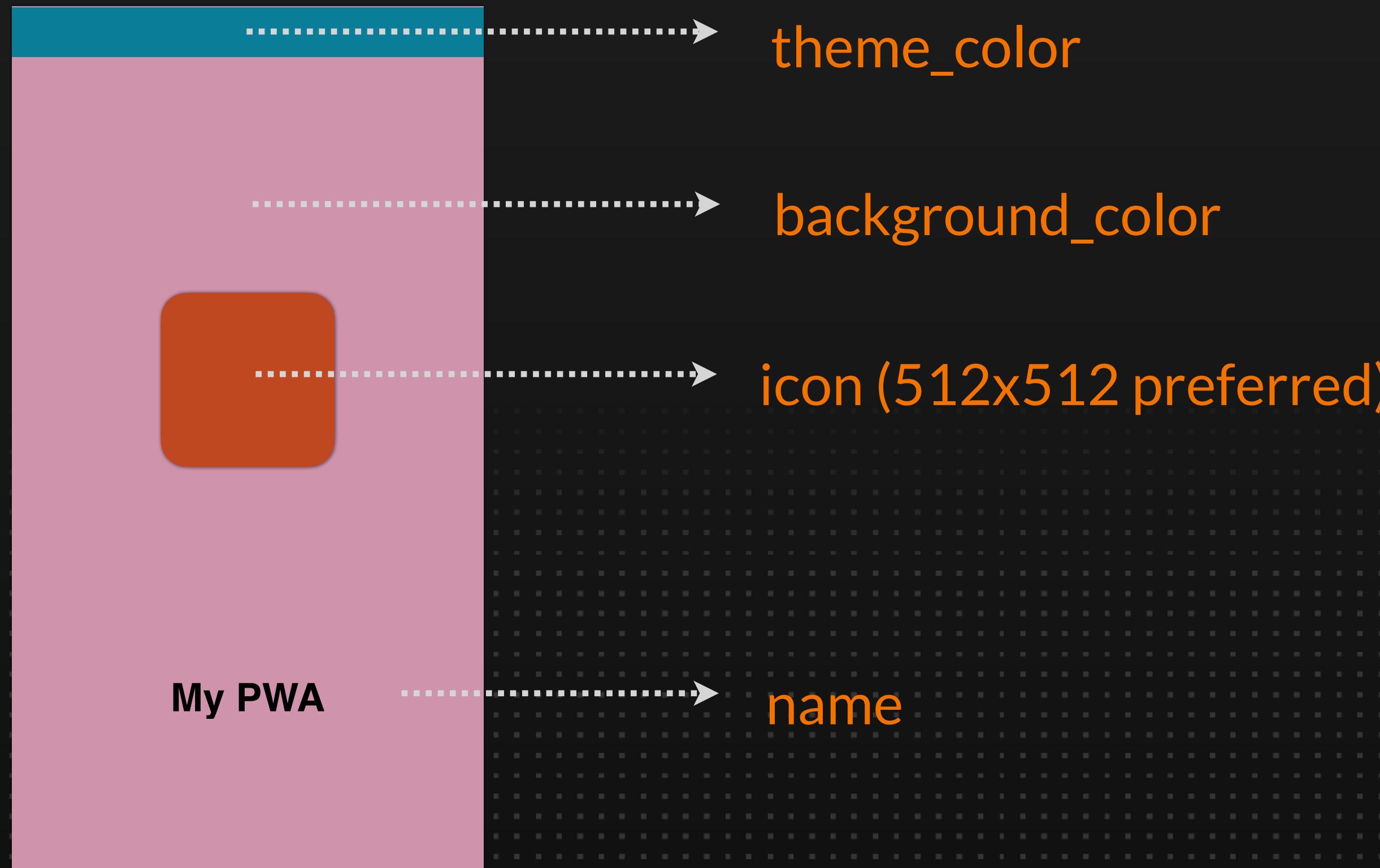
Splash Screens for Android

- They are automatically generated
- You can't provide your own image
- They use the meta data from the Web App Manifest

Splash Screens on Android



Splash Screens on Android



Splash Screens for iOS and iPadOS

- Safari won't use the App Manifest meta data
- It let us create our own static images.
- They are known as Startup Images
- We define them through a <link>
- We must use a deprecated meta tag
- It must match exactly the current device's screen size
- On iPad it gets more complicated

Splash Screens for iOS and iPadOS

index.html

X

```
<meta name="apple-mobile-web-app-capable"  
      content="yes">
```

```
<link rel="apple-touch-startup-image"  
      href="splash.png">
```

Warning



The problem on Safari is that it needs the image with the exact size of the device's screen

Startup Images for iOS

- iPhone SE
- iPhone 7, 8, SE 2nd edition
- iPhone X, XS, 11 Pro, 12 Pro, 13 Pro, 14 Pro, 15 Pro
- iPhone XR, 11, 12, 13
- iPhone 7 Plus, 8 Plus
- iPhone X Max, XS Max, 11 Pro Max, 12 Pro Max, 13 Pro Max, 14 Pro Max, 15 Pro Max
- More iPhones coming

Startup Images for iOS

Landscape

- Landscape: iPhone 7 Plus, 8 Plus
- Landscape: iPhone X Max, XS Max, 11 Pro Max, 12 Pro Max, 13 Pro Max, 14 Pro Max, 15 Pro Max

Startup Images for iPadOS

- Several iPads available
- Portrait and Landscape
- Split View
- Side View
- Any size?

Device	Portrait size	Landscape size
12.9" iPad Pro	2048px × 2732px	2732px × 2048px
11" iPad Pro	1668px × 2388px	2388px × 1668px
10.5" iPad Pro	1668px × 2224px	2224px × 1668px
9.7" iPad	1536px × 2048px	2048px × 1536px
7.9" iPad mini 4	1536px × 2048px	2048px × 1536px
iPhone Xs Max	1242px × 2688px	2688px × 1242px
iPhone Xs	1125px × 2436px	2436px × 1125px
iPhone XR	828px × 1792px	1792px × 828px
iPhone X	1125px × 2436px	2436px × 1125px
iPhone 8 Plus	1242px × 2208px	2208px × 1242px
iPhone 8	750px × 1334px	1334px × 750px
iPhone 7 Plus	1242px × 2208px	2208px × 1242px
iPhone 7	750px × 1334px	1334px × 750px
iPhone 6s Plus	1242px × 2208px	2208px × 1242px
iPhone 6s	750px × 1334px	1334px × 750px
iPhone SE	640px × 1136px	1136px × 640px

Warning



Brace yourself for more than 20
versions of the splash screen

Splash Screens for iOS and iPadOS

index.html

X

```
<!-- Sizes property is ignored -->

<link rel="apple-touch-startup-image"
      href="splash_1136.png"
      media="screen and (device-width: 320px) and
            (device-height: 568px) and (resolution: 2dppx) and
            (orientation: landscape)">
```

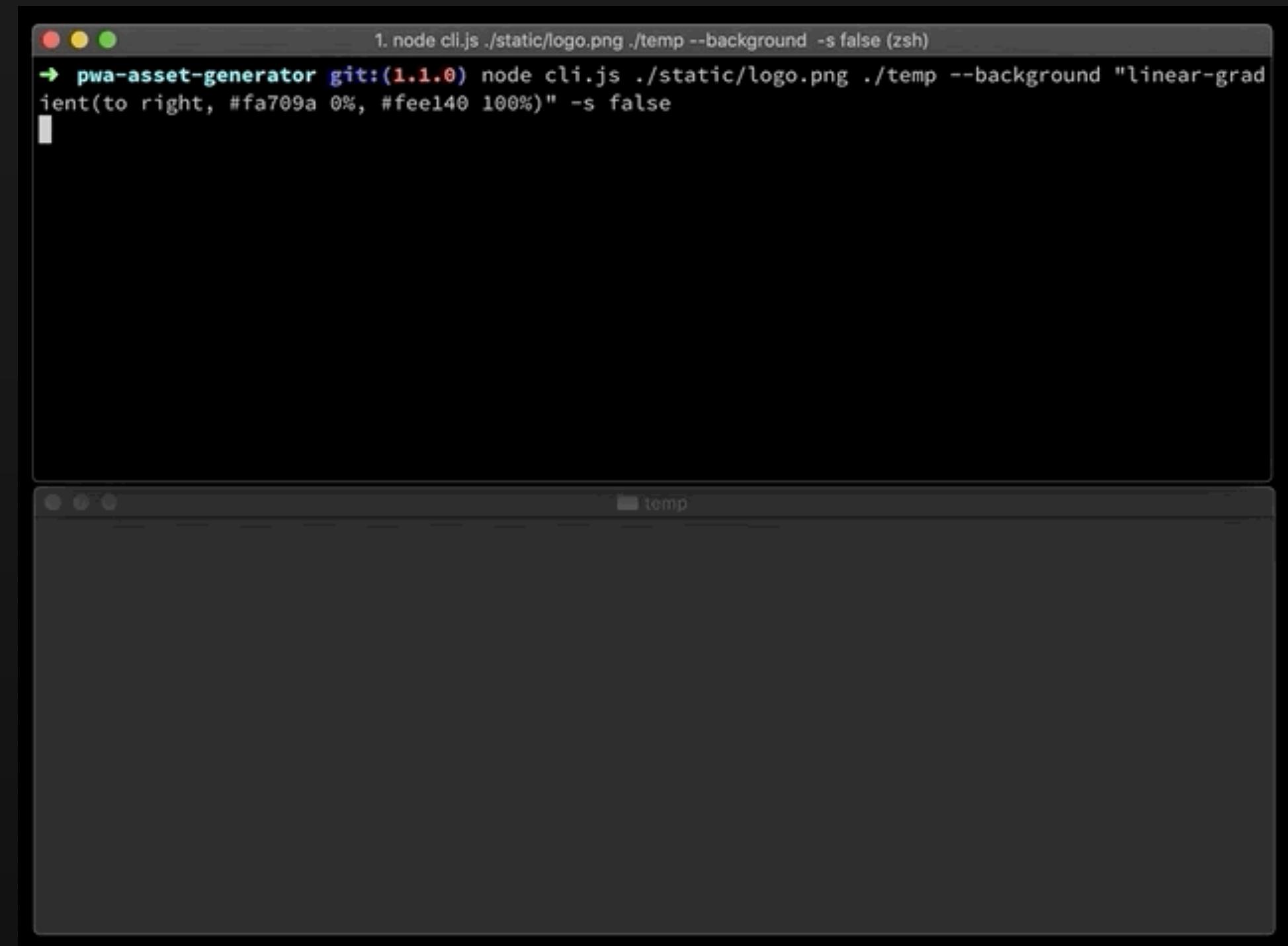
```
8   <link
9     rel="apple-touch-icon"
10    sizes="180x180"
11    href="/assets/icons/icon_180x180.png"
12  />
13  <link
14    rel="apple-touch-startup-image"
15    media="screen and (device-width: 320px) and (device-height: 568px) and (-webkit-device-pixel-ratio: 2) and (orientation: landscape)"
16    href="/assets/splash/icon_1136x640.png"
17  />
18  <link
19    rel="apple-touch-startup-image"
20    media="screen and (device-width: 375px) and (device-height: 812px) and (-webkit-device-pixel-ratio: 3) and (orientation: landscape)"
21    href="/assets/splash/icon_2436x1125.png"
22  />
23  <link
24    rel="apple-touch-startup-image"
25    media="screen and (device-width: 414px) and (device-height: 896px) and (-webkit-device-pixel-ratio: 2) and (orientation: landscape)"
26    href="/assets/splash/icon_1792x828.png"
27  />
28  <link
29    rel="apple-touch-startup-image"
30    media="screen and (device-width: 414px) and (device-height: 896px) and (-webkit-device-pixel-ratio: 2) and (orientation: portrait)"
31    href="/assets/splash/icon_828x1792.png"
32  />
33  <link
34    rel="apple-touch-startup-image"
35    media="screen and (device-width: 375px) and (device-height: 667px) and (-webkit-device-pixel-ratio: 2) and (orientation: landscape)"
36    href="/assets/splash/icon_1334x750.png"
37  />
38  <link
39    rel="apple-touch-startup-image"
40    media="screen and (device-width: 414px) and (device-height: 896px) and (-webkit-device-pixel-ratio: 3) and (orientation: portrait)"
41    href="/assets/splash/icon_1242x2688.png"
42  />
43  <link
44    rel="apple-touch-startup-image"
45    media="screen and (device-width: 414px) and (device-height: 736px) and (-webkit-device-pixel-ratio: 3) and (orientation: landscape)"
46    href="/assets/splash/icon_2208x1242.png"
47  />
48  <link
49    rel="apple-touch-startup-image"
50    media="screen and (device-width: 375px) and (device-height: 812px) and (-webkit-device-pixel-ratio: 3) and (orientation: portrait)"
51    href="/assets/splash/icon_1125x2436.png"
52  />
```

PWA Static Asset Generator

```
npx pwa-asset-generator light-logo.svg ./assets --dark-mode --  
background dimgrey --splash-only --type jpeg --quality 80 --index  
./src/app/index.html
```

```
npx pwa-asset-generator dark-logo.svg ./assets --background  
lightgray --splash-only --type jpeg --quality 80 --index  
./src/app/index.html
```

PWA Static Asset Generator



Client Side Generator

- Use <canvas> and Data URI to create the images on the fly
- Use PWACompat:
 - Open source library by Chrome
 - It takes the App Manifest and generates assets for other browsers
 - It generates Apple's custom assets, including startup images on the fly with JavaScript
 - It emulates Android splash screens for iOS

PWA Compat

index.html

X

```
<link rel="manifest" href="manifest.webmanifest" />

<script async src="https://unpkg.com/pwacompat"
crossorigin="anonymous"></script>

<style>
link[rel="manifest"] {
  --pwacompat-splash-font: 24px
  Verdana;
}</style>
```

Warning

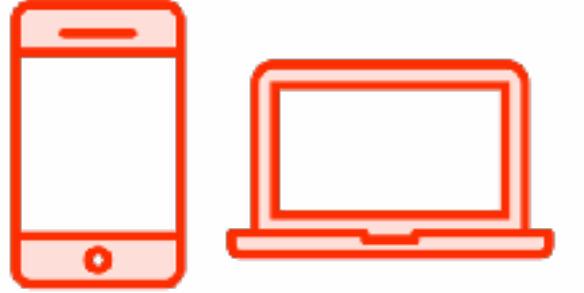


The theme color can be overridden from a <meta> tag; in the manifest we use **theme_color**; in the HTML meta element is **theme-color**



Making the App Offline-capable

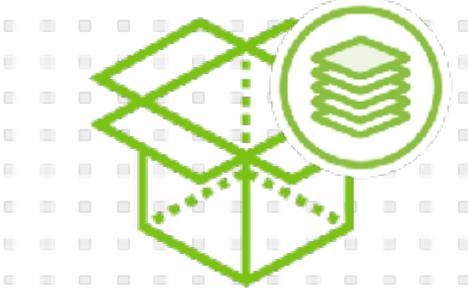
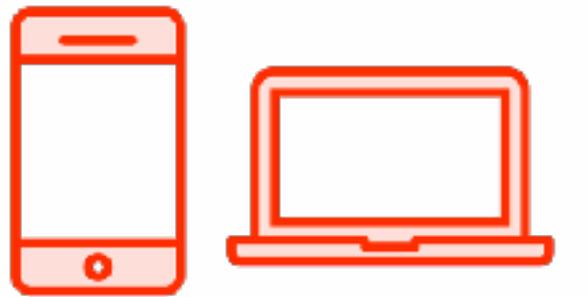
Native Apps



Native Apps



Native Apps



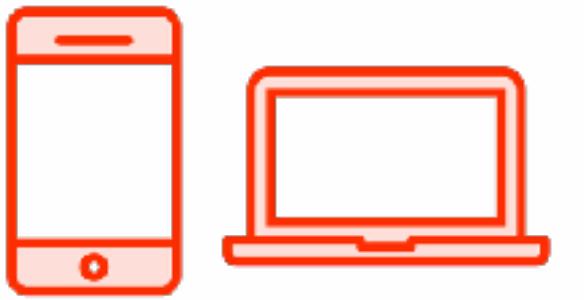
Bundle

API calls

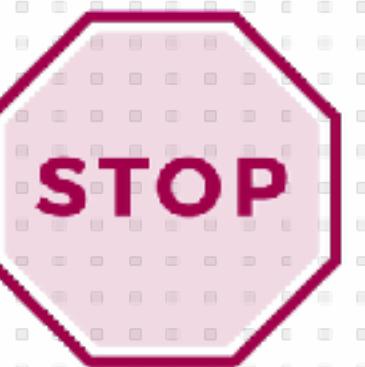


Web server

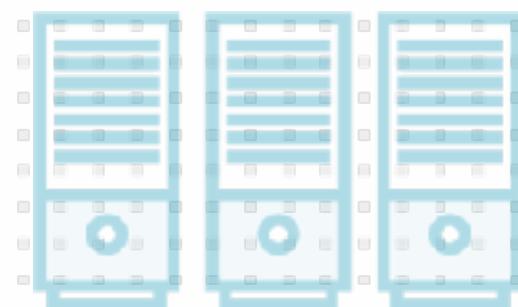
Native Apps



API calls

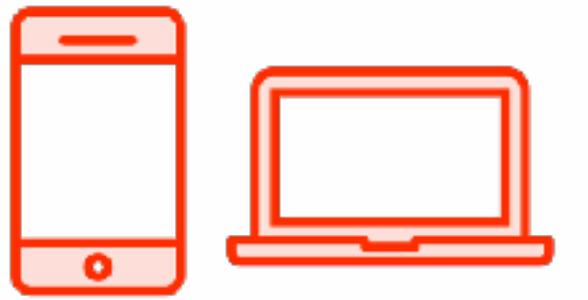


Bundle



Web server

Classic Web Apps



Web
runtime



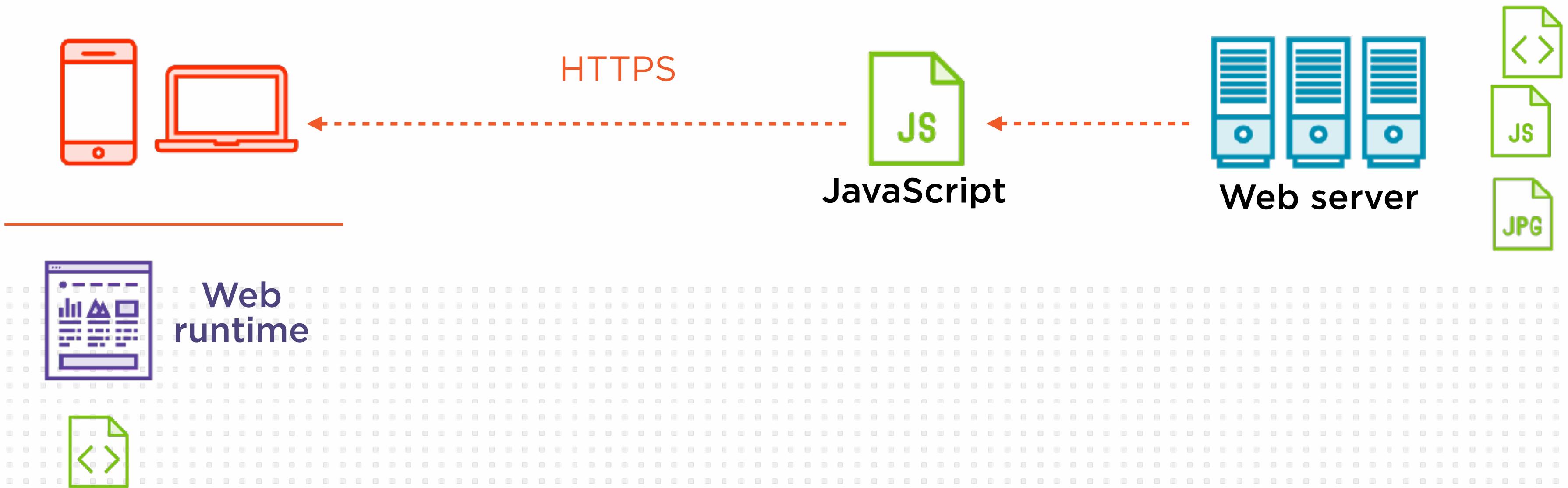
Web server



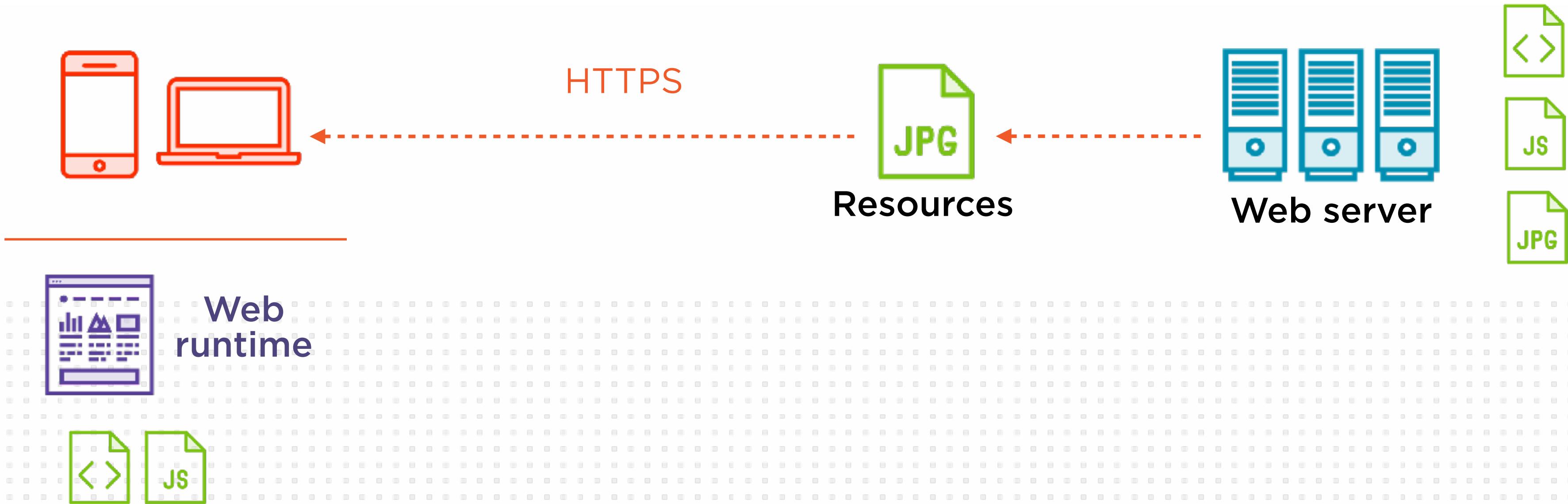
Classic Web Apps



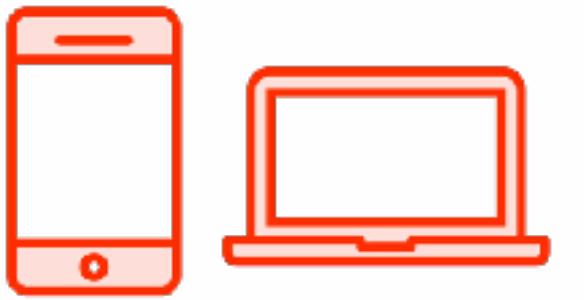
Classic Web Apps



Classic Web Apps



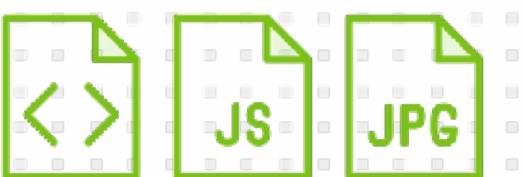
Classic Web Apps



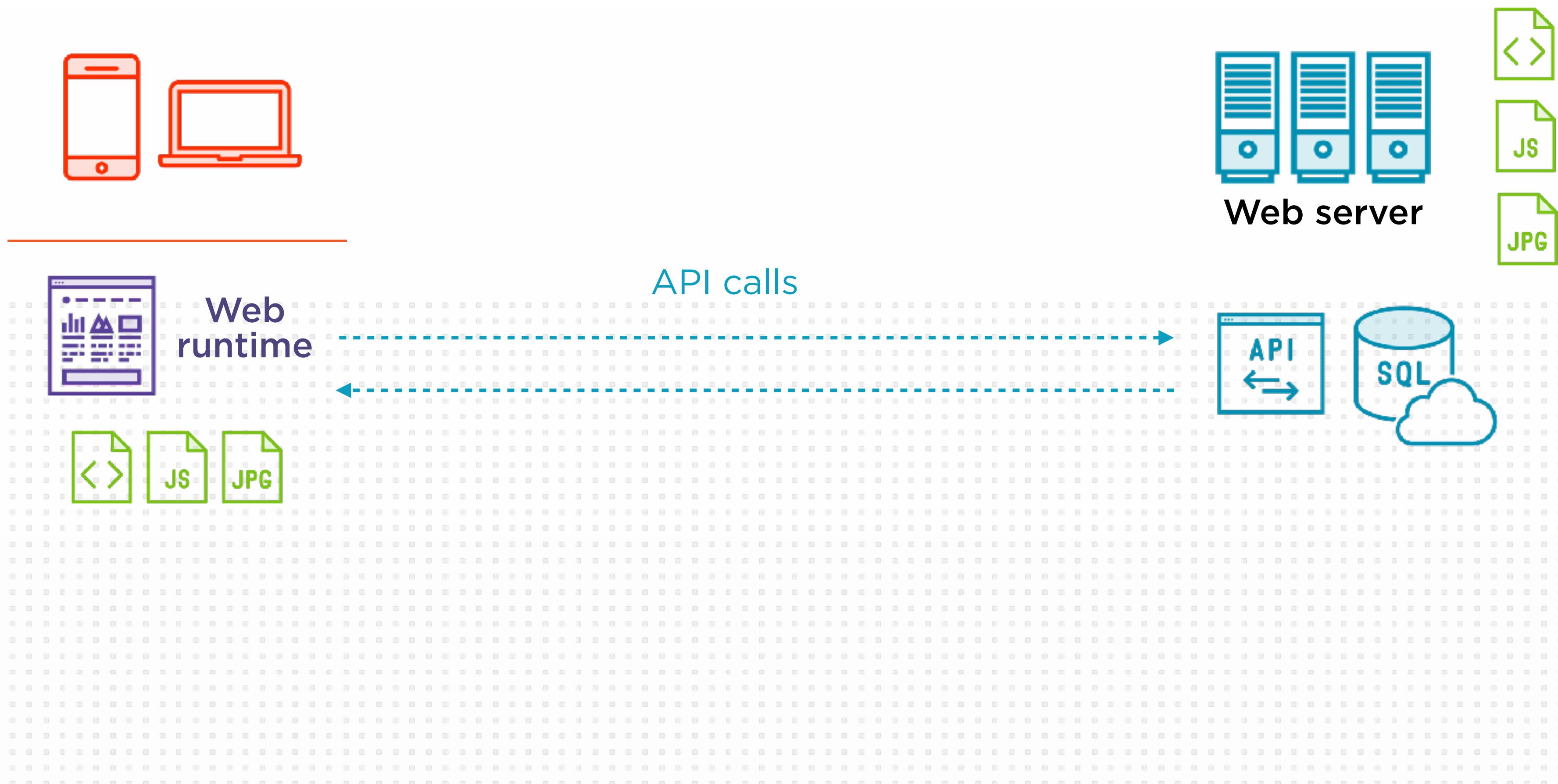
Web server



**Web
runtime**



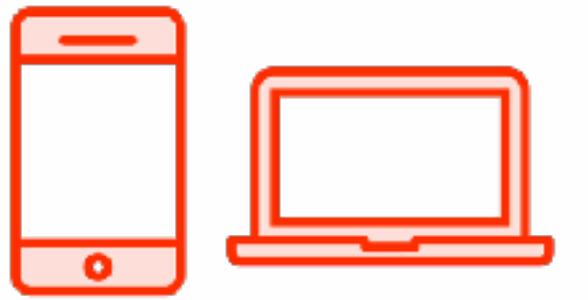
Classic Web Apps



Classic Web Apps



Progressive Web Apps



Web
runtime



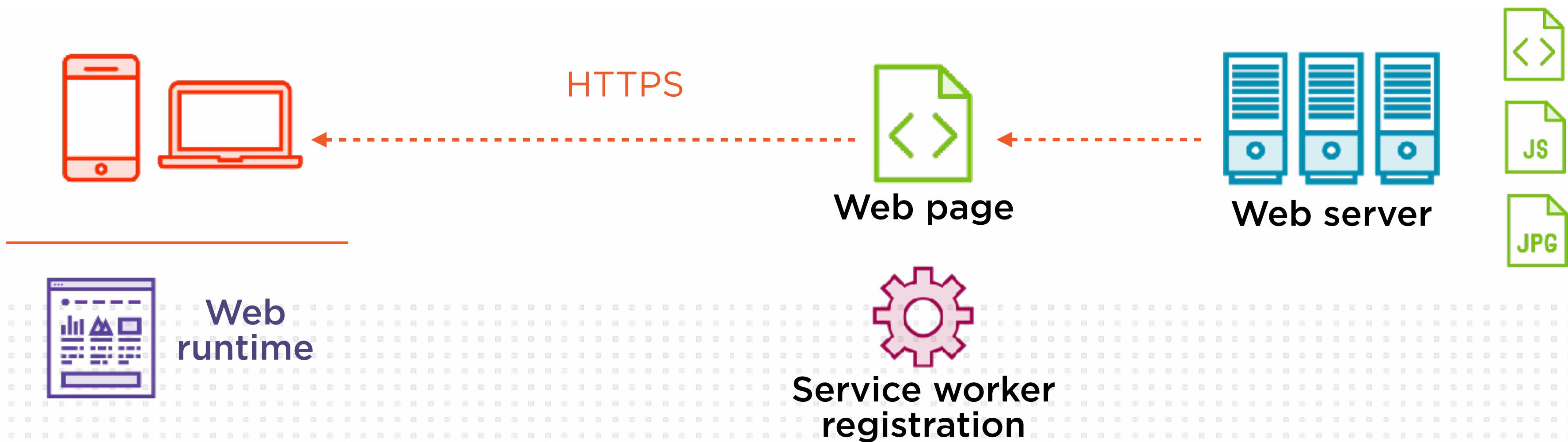
Web server



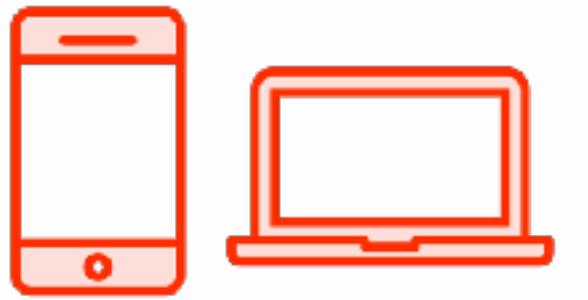
Progressive Web Apps



Progressive Web Apps



Progressive Web Apps



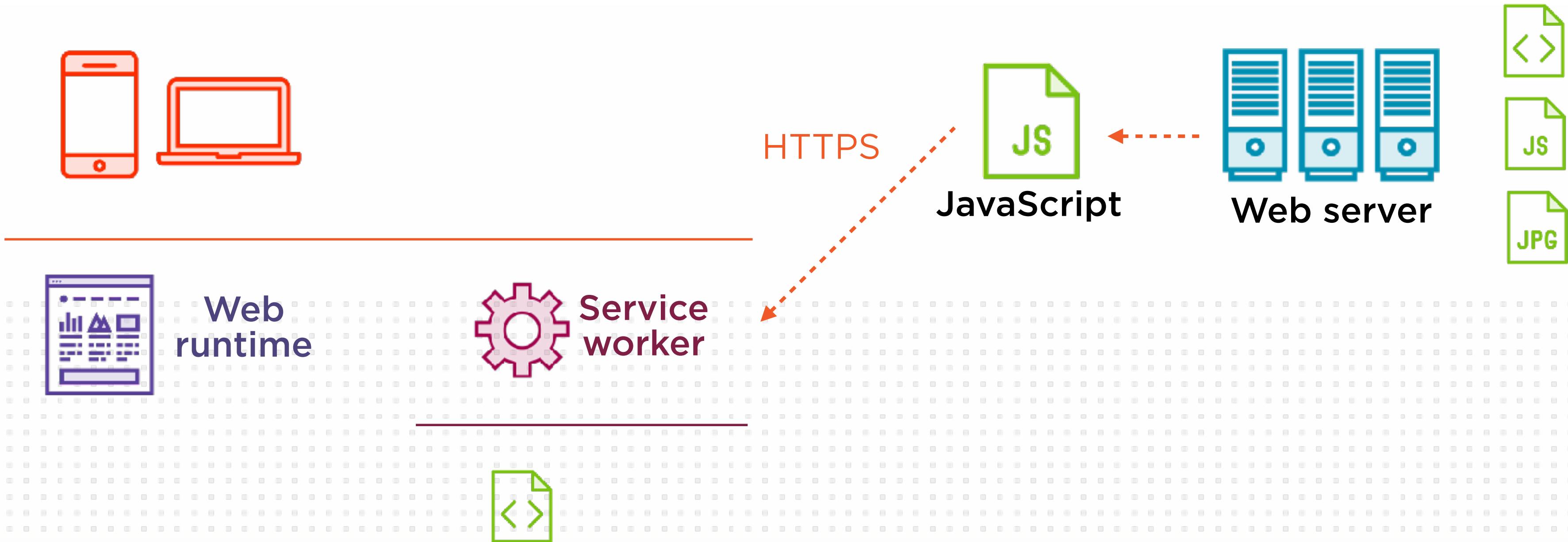
Web
runtime



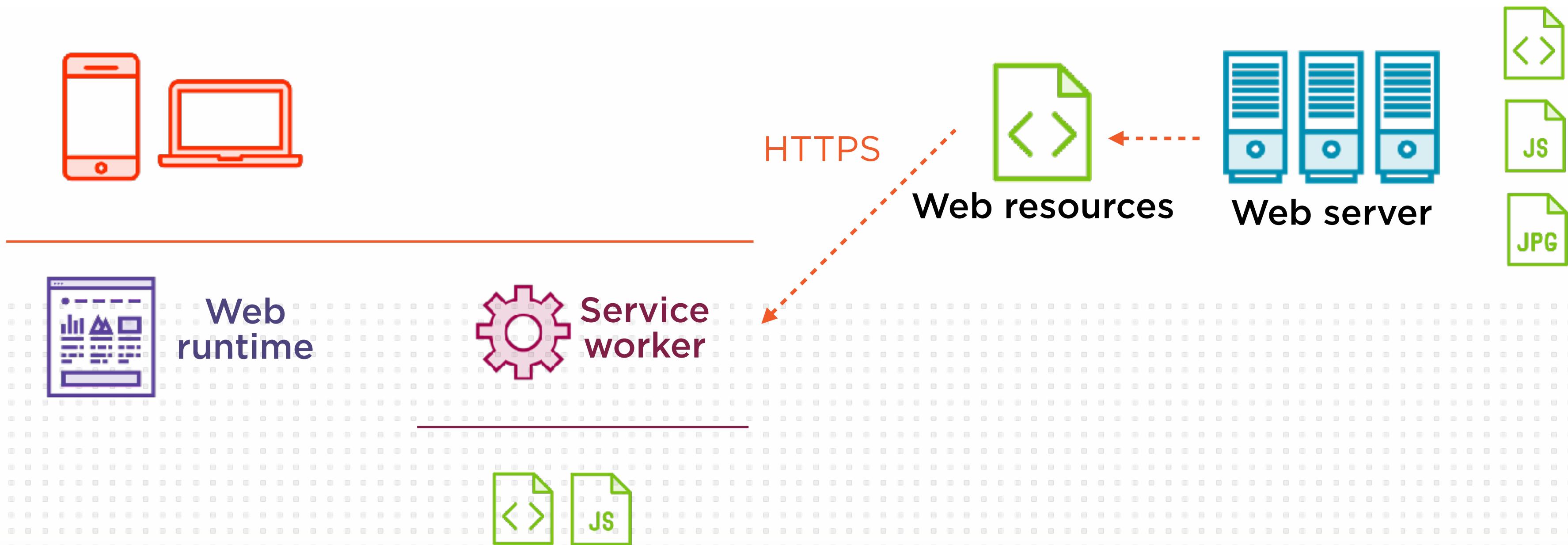
Service
worker

Web server

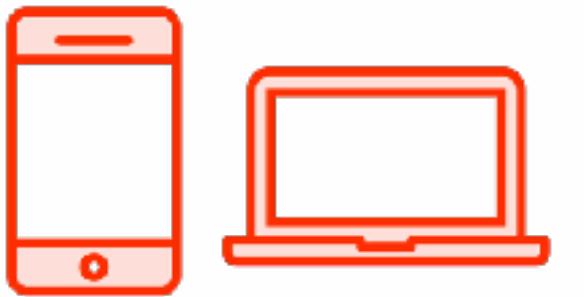
Progressive Web Apps



Progressive Web Apps



Progressive Web Apps



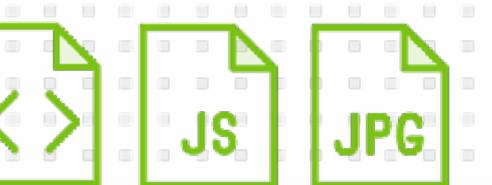
Web server



**Web
runtime**



**Service
worker**



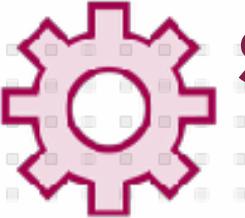
Progressive Web Apps



Progressive Web Apps

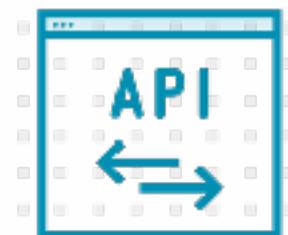


Web
runtime

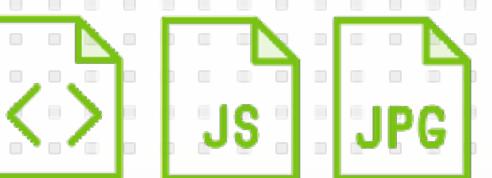


Service
worker

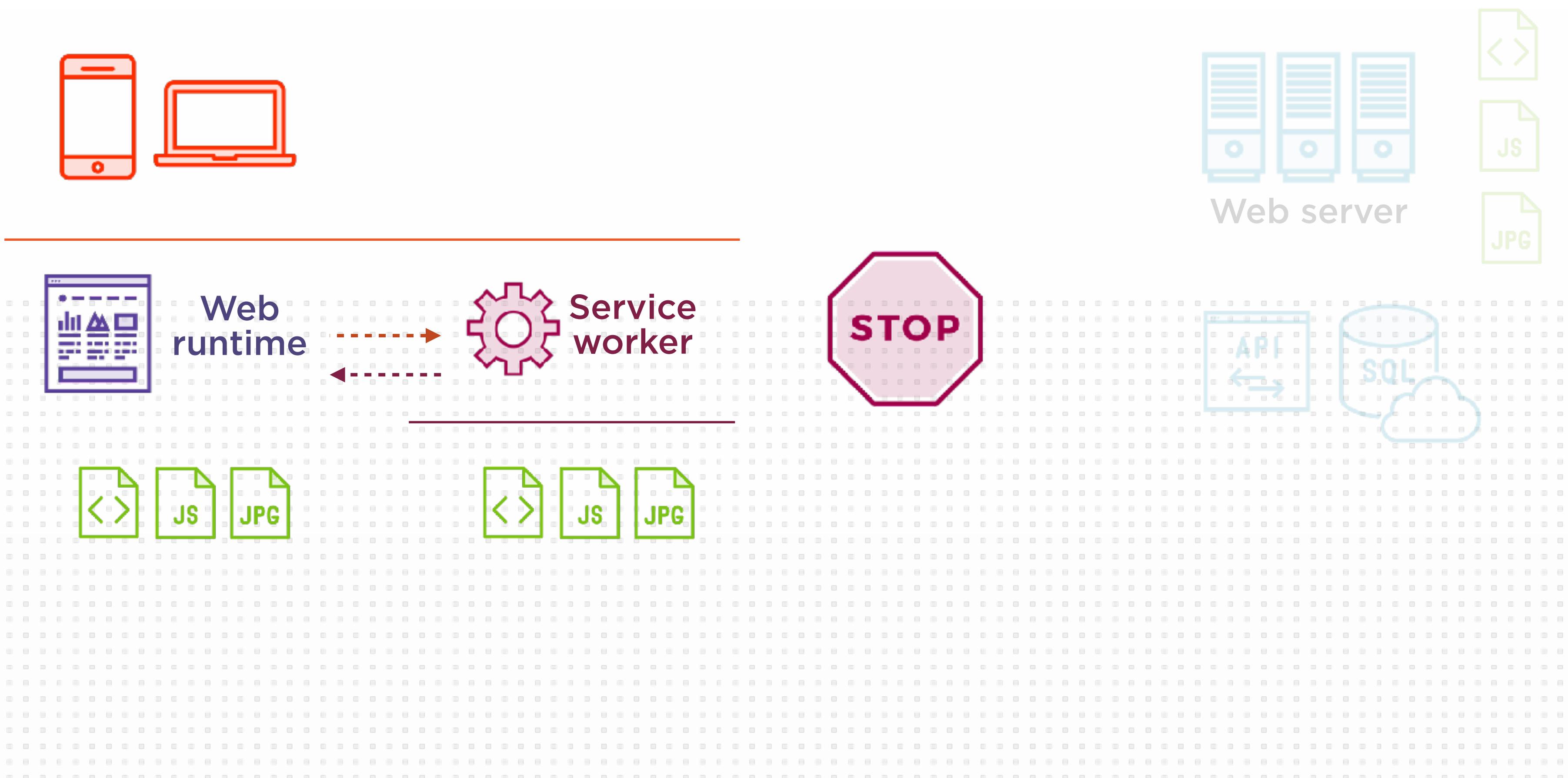
API calls



Web server



Progressive Web Apps

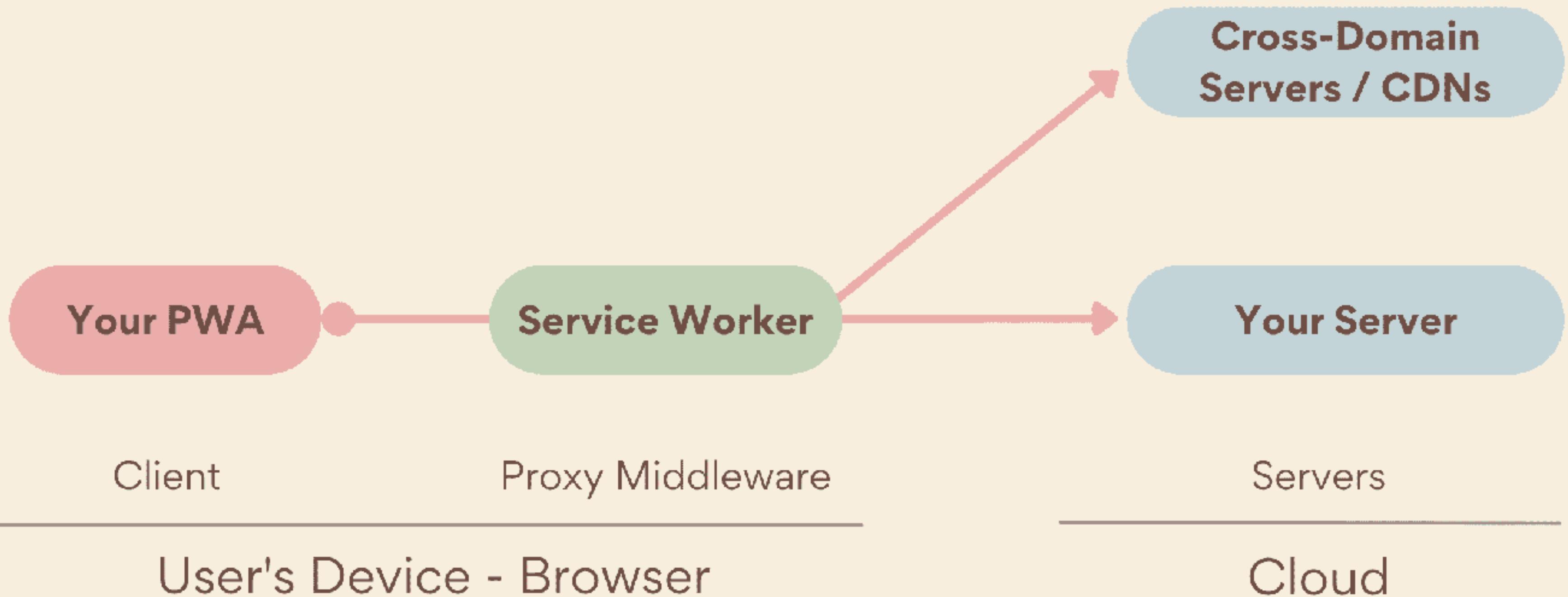


Definition

Service Worker



A JavaScript file running in its own thread
that will act as a middleware offering a local
installed web server or web proxy for your
PWA, including resources and API calls



Service Worker

- Runs client-side in browser's engine
- HTTPS required
- Installed by a web page
- Own thread and lifecycle
- Acts as a network proxy or local web server
in the name of the real server
- Abilities to run in the background
- No need for user's permission

Definition

Scope



An origin (host and port) and a path

`https://mydomain.com` or `https://domain.com/myapp`

Service Worker and Scopes

- It manages all pages within browser and within installed app from scope
- It's installed by any page in the scope
- After installed, it can serve all files requested from the scope
- Only one service worker is allowed
- WebKit adds partition management

Caching resources

- Service worker has a local cache
- We can cache all or some resources
- JavaScript promise
- Prefetch on installation
- Cache on request
- App Shell pattern

Serving resources

- The service worker will respond for every request the PWA make
- It can serve from the cache
- It can forward the request to the network
- It can synthesize a response
- Any mixed algorithm is possible

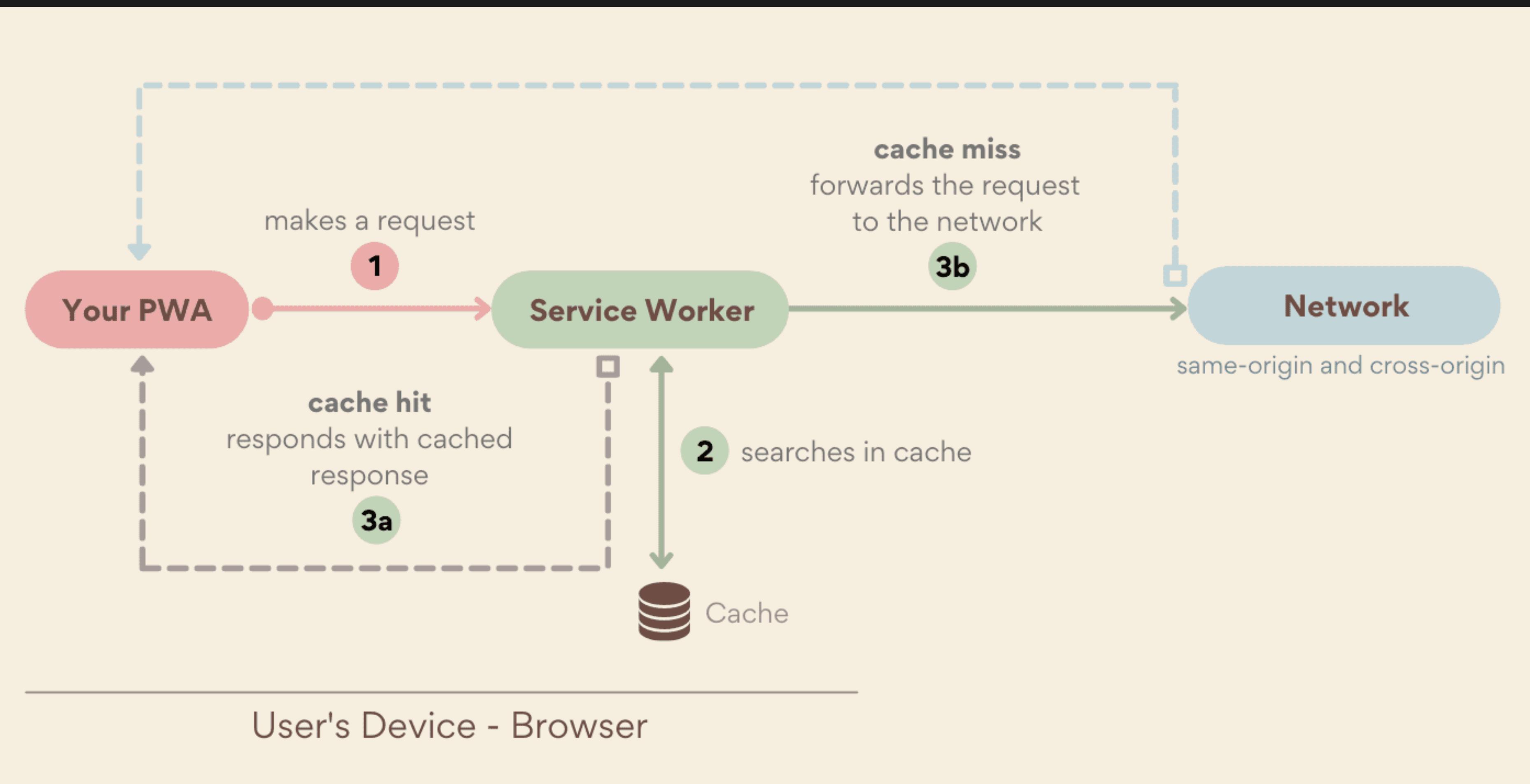
Cache Serving Strategies

Cache First

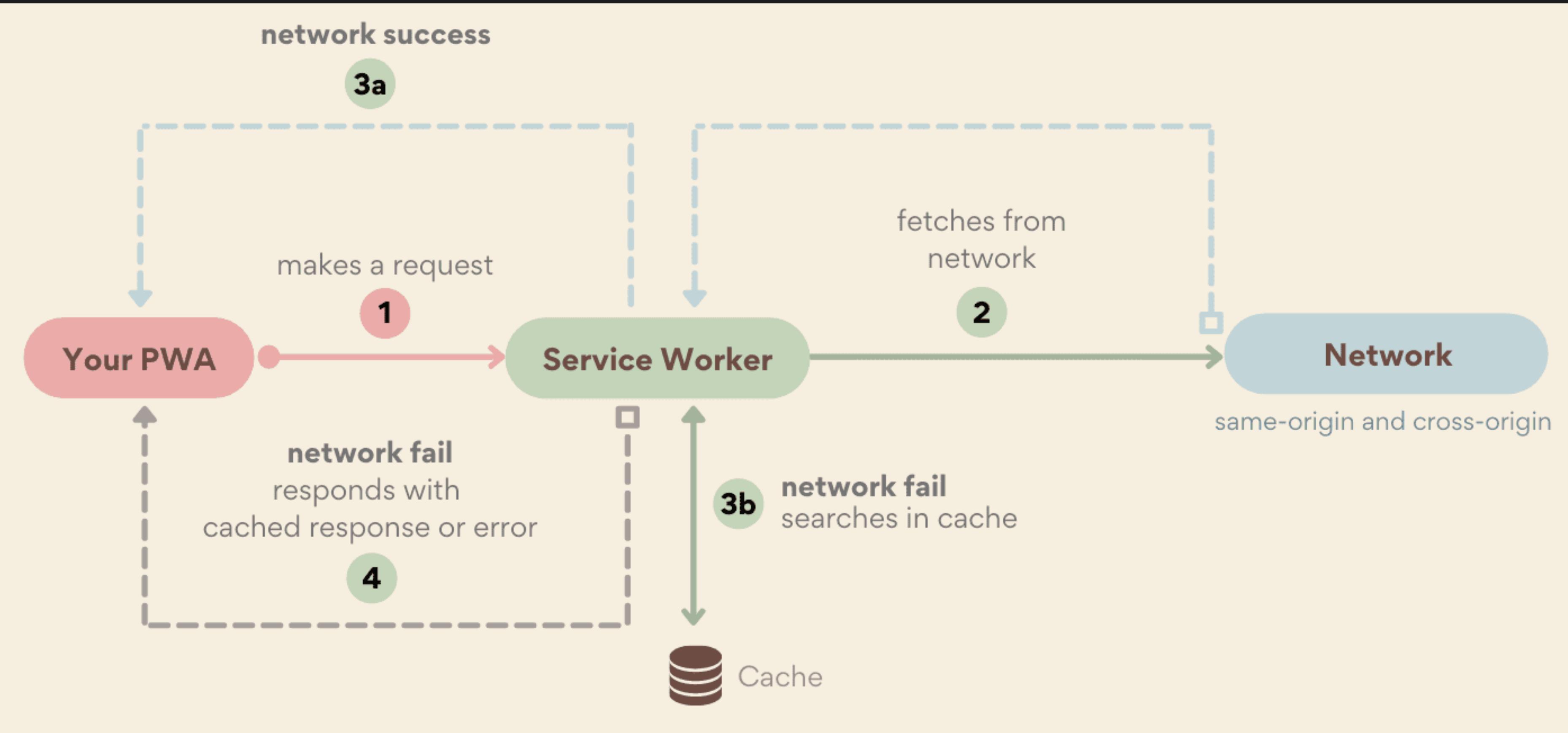
Network First

Stale while
Revalidate

Cache First

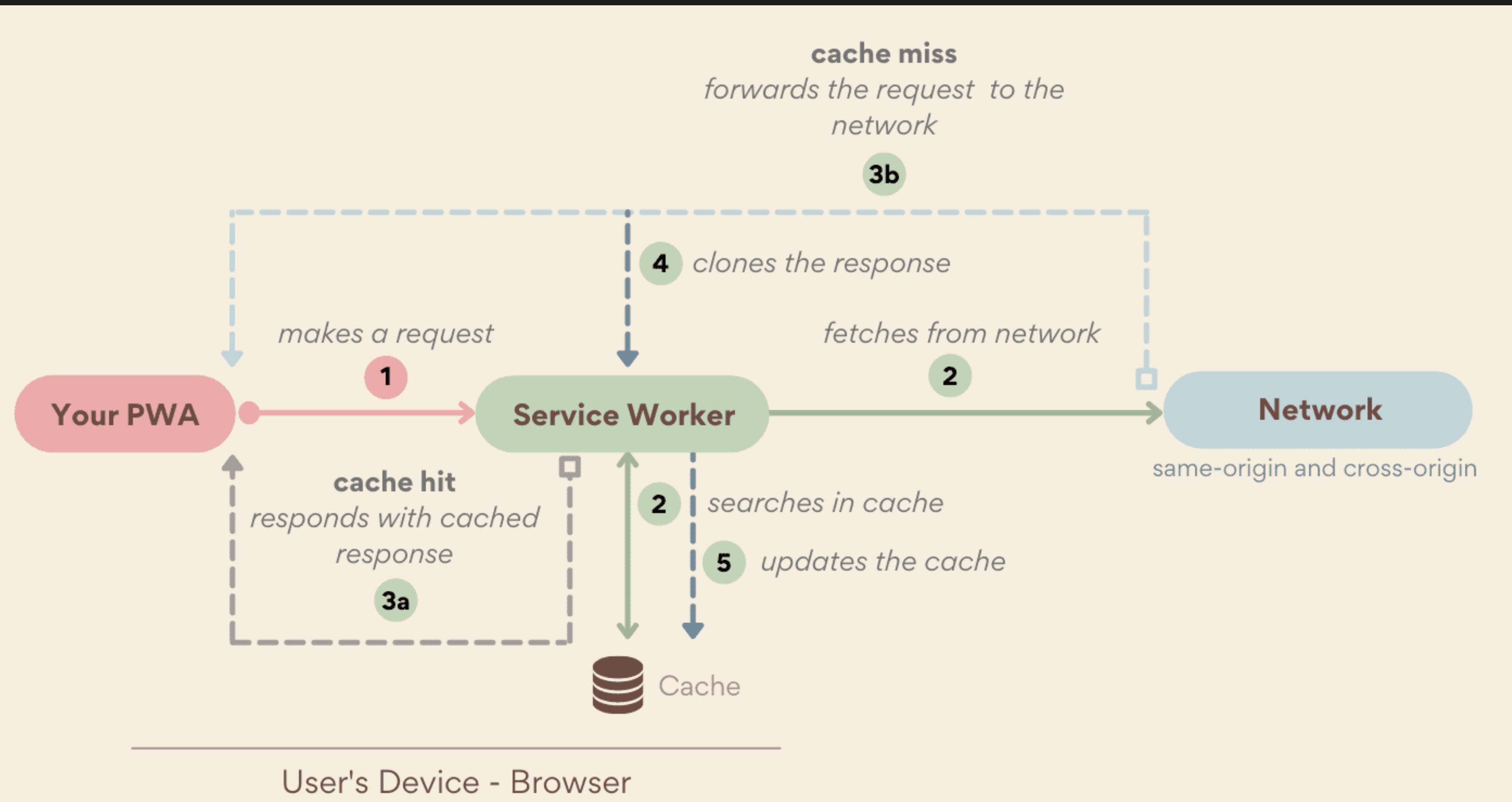


Network First



User's Device - Browser

Stale while Revalidate



Updating resources

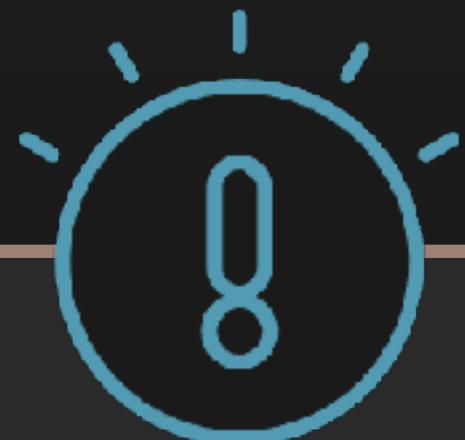
- Files are saved in the client
- Updating files in the server won't trigger any automatic change in the client
- We need to define and code an update algorithm
- It will need a process within your build system for hashing or versioning files

Important



Developer is in full control of how to cache and serve the resources of the PWA, and how to manage API calls.

Important



Updating the app doesn't require full reinstallation;
we can just replace the updated files silently or
with a user's notification.

Updating the Service Worker

- Change the file on the server
- Browser will check on new instances
- Work with HTTP Cache headers (Max 24 hours)
- Call `registration.update()`

Idea



For more information on this topic, check the course
JavaScript in the Background at Frontend Masters



Advanced PWA UI techniques

Avoid Content Selection

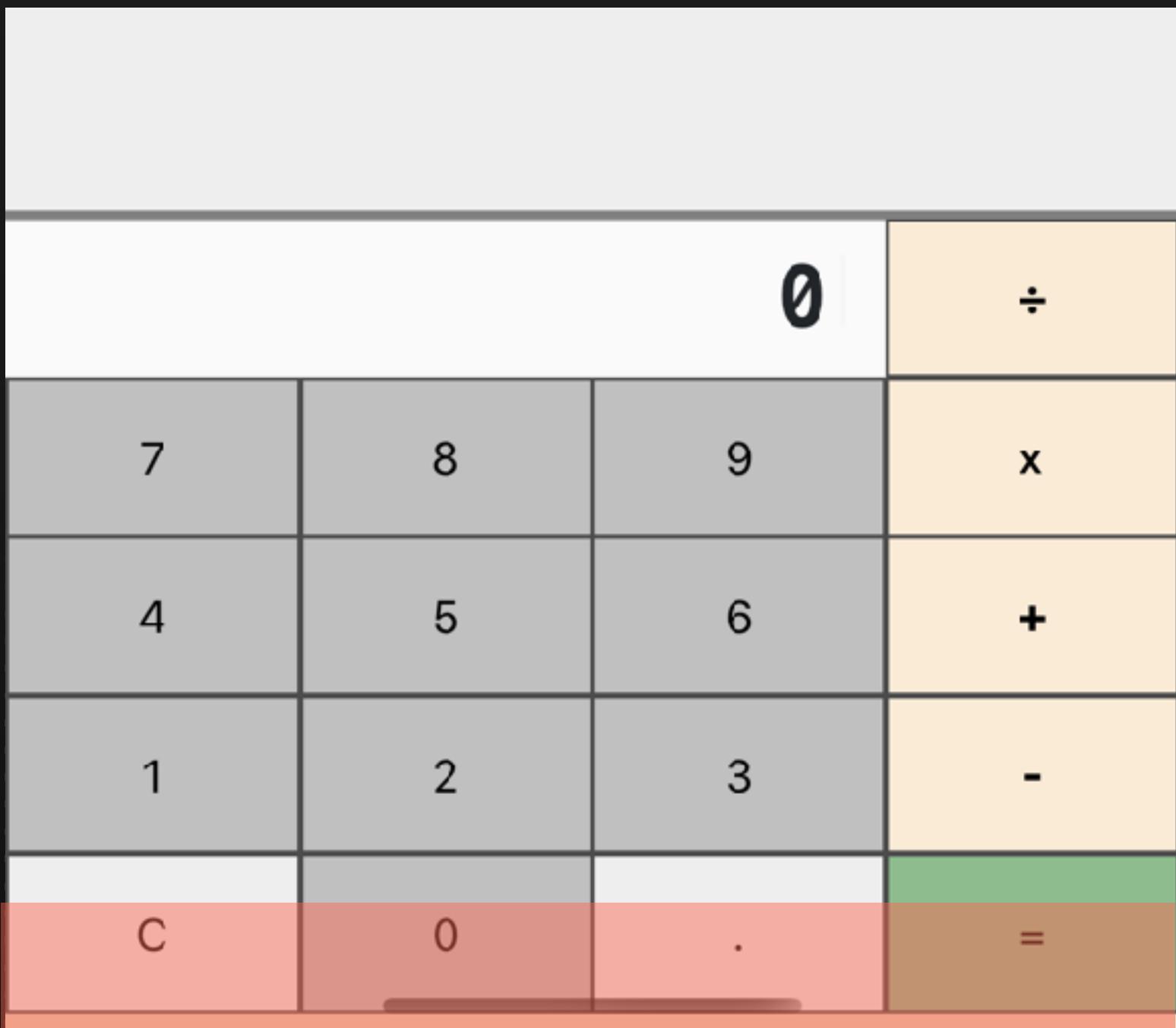
We use user-select to disable selection on UI elements. Use this with responsibility!

styles.css

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

X

Safe Areas



Non-safe area

Use Safe Areas

With CSS Environmental Variables support through the env() function and four variables

styles.css

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

X

Use Safe Areas

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

styles.css

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

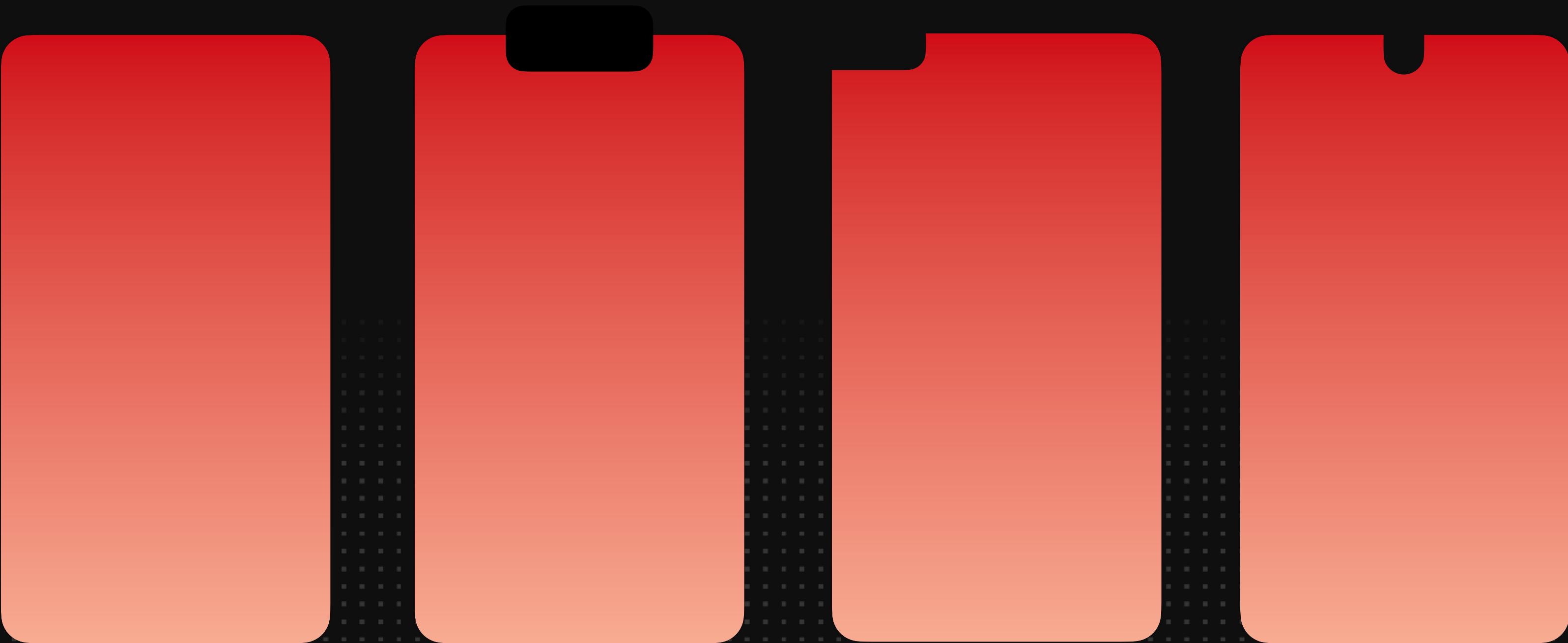
Use Safe Areas

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

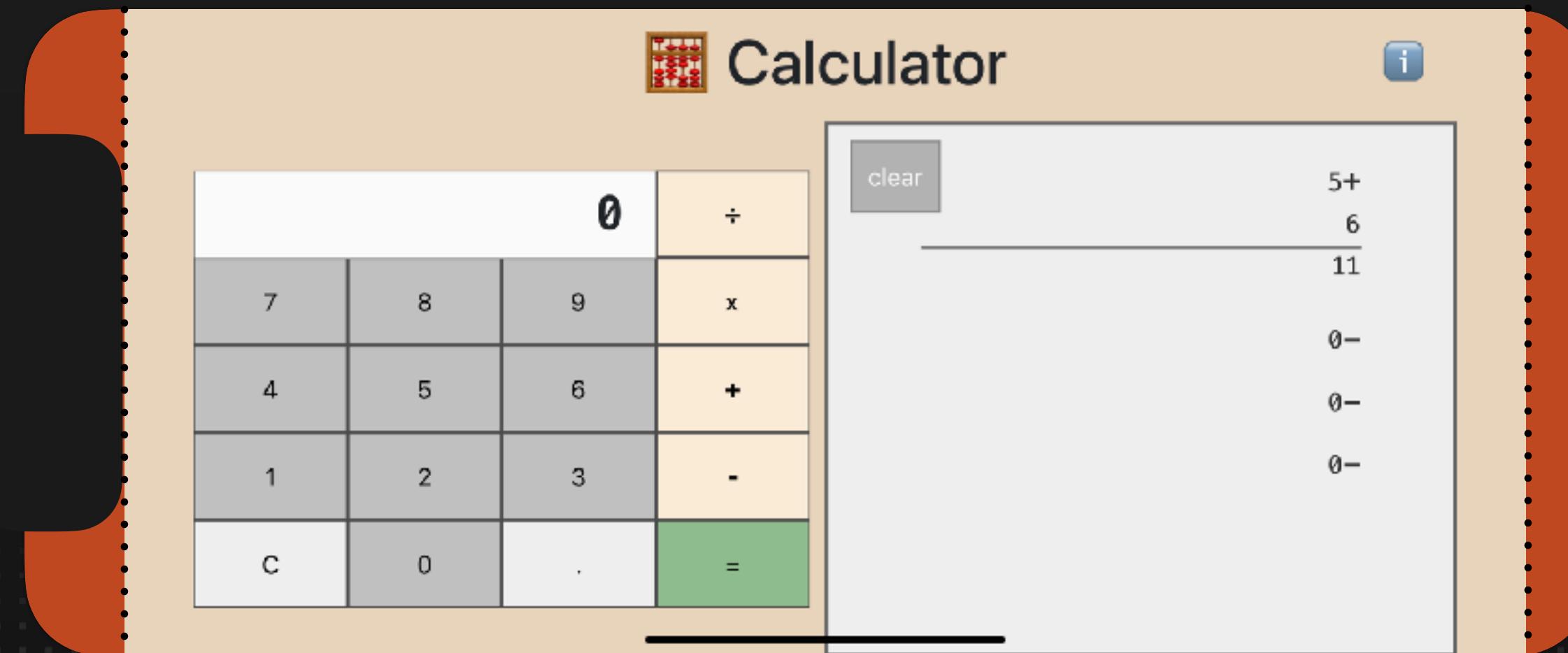
styles.css

```
.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  
    }  
}
```

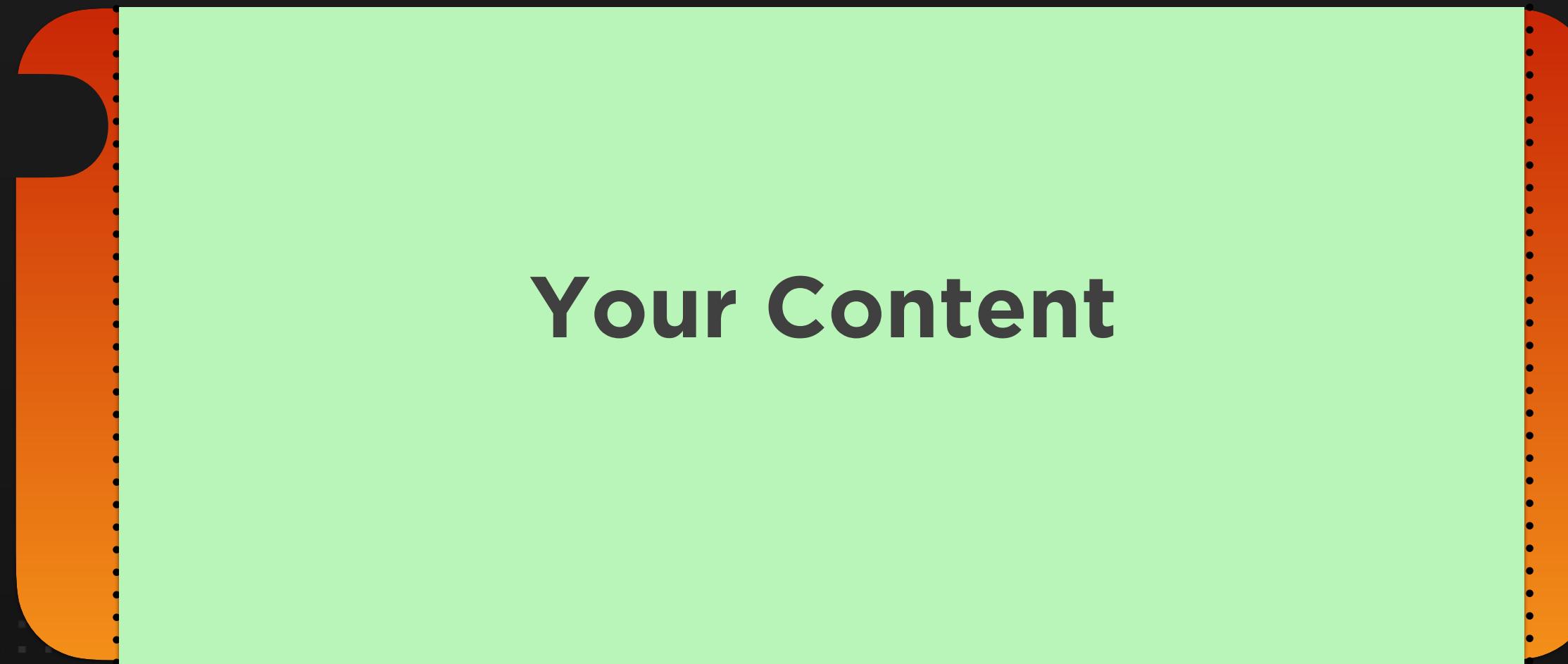
Devices with Notch



Viewport Area



Default Viewport definition



PWAs in standalone for devices with notch

Ask for Cover Viewport

Use Safe Areas

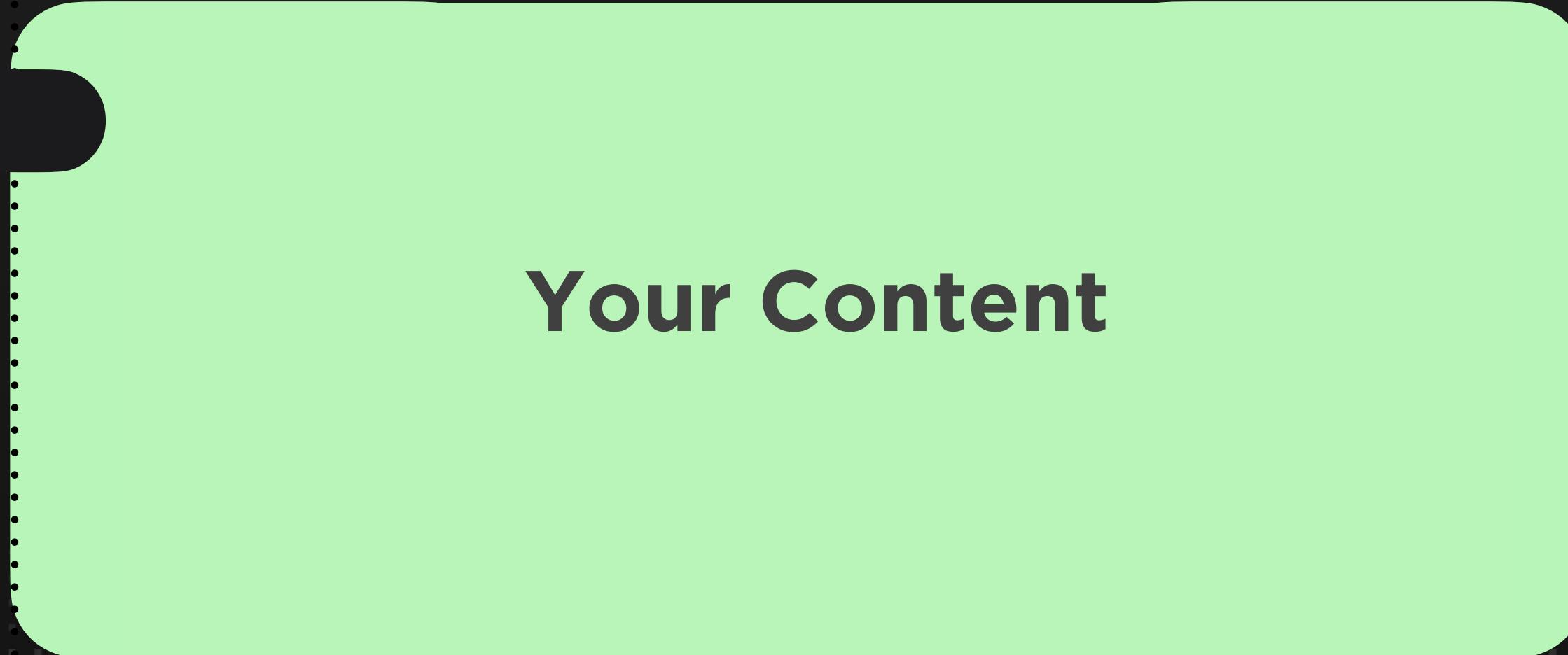
Cover Viewport

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

styles.css

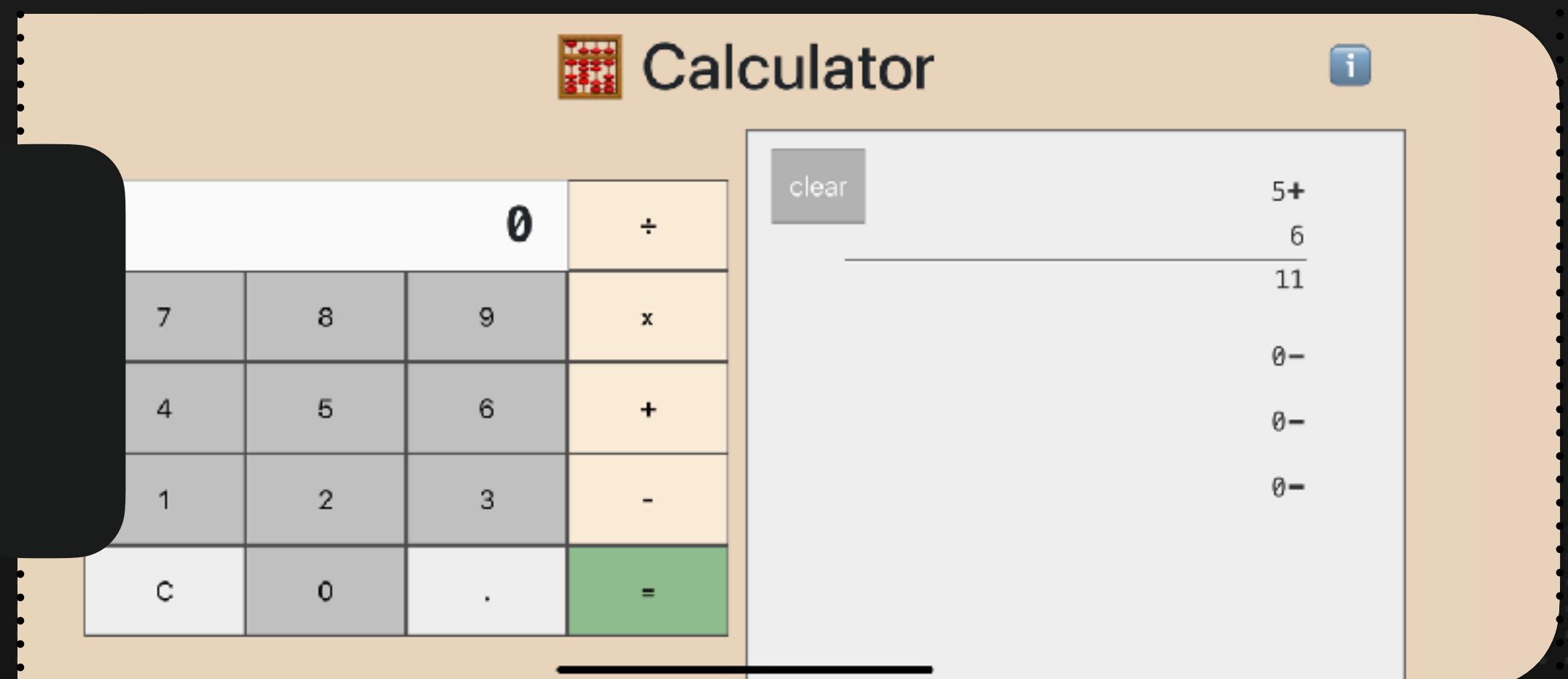
```
<meta name="viewport" content="width=device-width,  
initial-scale=1.0, viewport-fit=cover">
```

Viewport-fit=cover



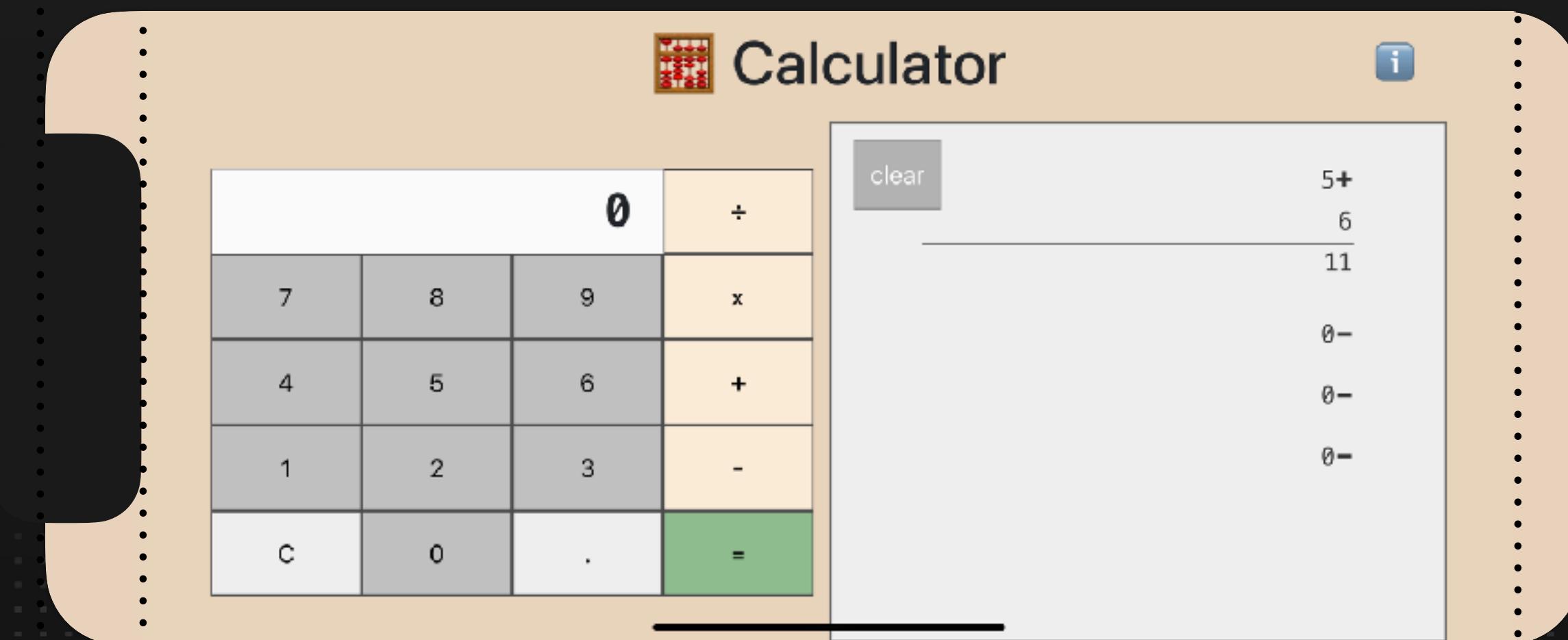
Your Content

Fit Cover Viewport Area

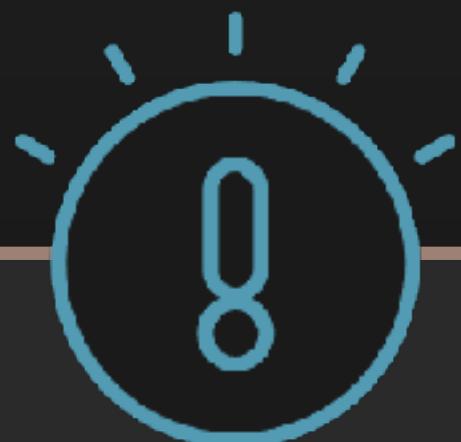


Viewport area with viewport=fit

Using safe areas



Important



Apple had many meta tags available for web apps before the web app manifest spec was created

Warning



Do not add any meta tag that you
don't know what it's actually doing

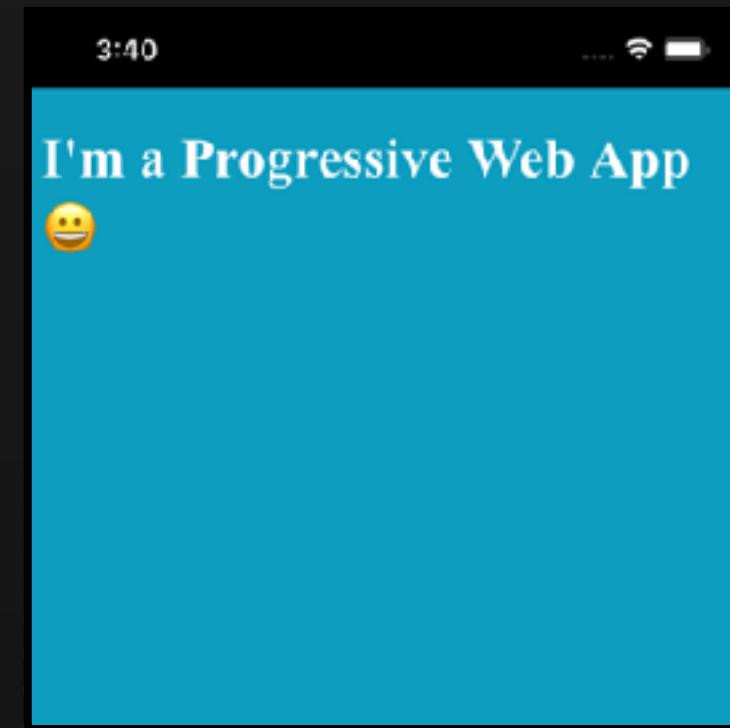
Fullscreen on iOS

styles.css

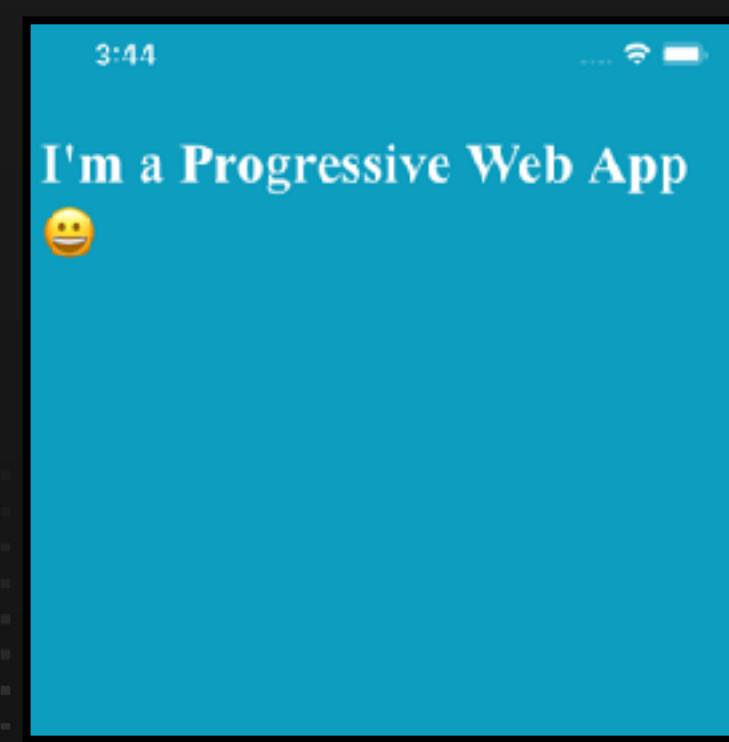
X

```
<meta name="apple-mobile-web-app-status-bar-style"  
      content="black-translucent">
```

apple-mobile-web-app-status-bar-style



black



black-translucent

Pull to Refresh

The screenshot displays the Financial Times mobile website. At the top, the title "FINANCIAL TIMES" is visible. Below it is a profile picture of Gideon Rachman and his name. A sub-headline reads "The cruelty of conspiracy theories". To the right, there are links for "Fast FT" and "Zimbabwe's Mnangagwa". The main content includes an article about Apple becoming the first trillion-dollar company, featuring a photo of an Apple logo. Another article discusses Germany becoming Donald Trump's European punchbag, with a photo of Donald Trump and Angela Merkel. A third article mentions US spy chiefs warning of Russian meddling, with a photo of senators. A red banner at the bottom left says "Daily Briefing Friday" and "Today's top global stories". The footer contains icons for "myFT", search, refresh, and settings, along with the text "Updated 11:10pm".

Pull to Refresh

We use CSS Overscroll Behavior.

styles.css

```
body {  
  overscroll-behavior-y: contain;  
}
```

Idea



Use the new View Transition API to transition between page changes, if you are a SPA or a MPA

Idea



For more information on this topic, check the course
Practical Web App Patterns at Frontend Masters

Use System Fonts

On some platforms, only system fonts are available.

styles.css

X

```
font-family: -apple-system,  
BlinkMacSystemFont, "Segoe UI", Roboto,  
Oxygen-Sans, Ubuntu, Cantarell, "Helvetica  
Neue", sans-serif;
```

Use Web Share API

index.js

X

```
navigator.share({  
  title: "Recommendation",  
  text: "Check my website",  
  url: "https://..."  
});
```

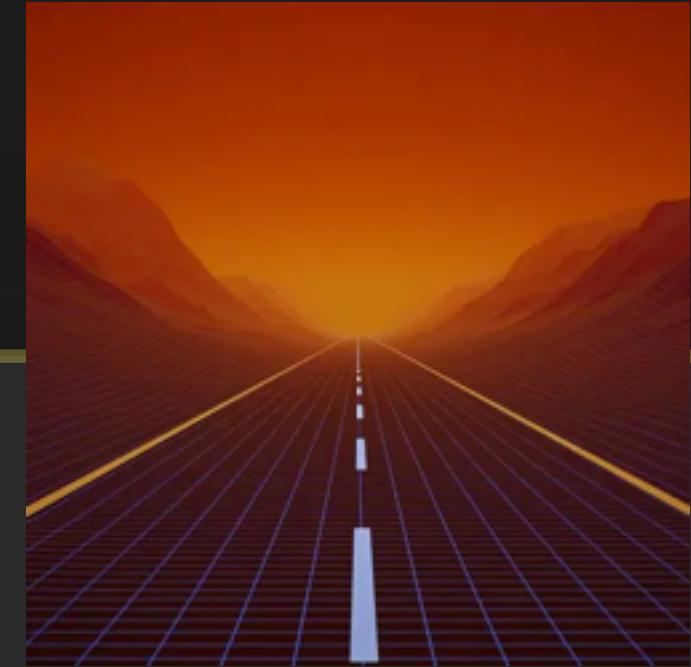
Disable Touch Callout

iOS context menu when long pressing a link; disabling is useful for javascript: links

styles.css

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

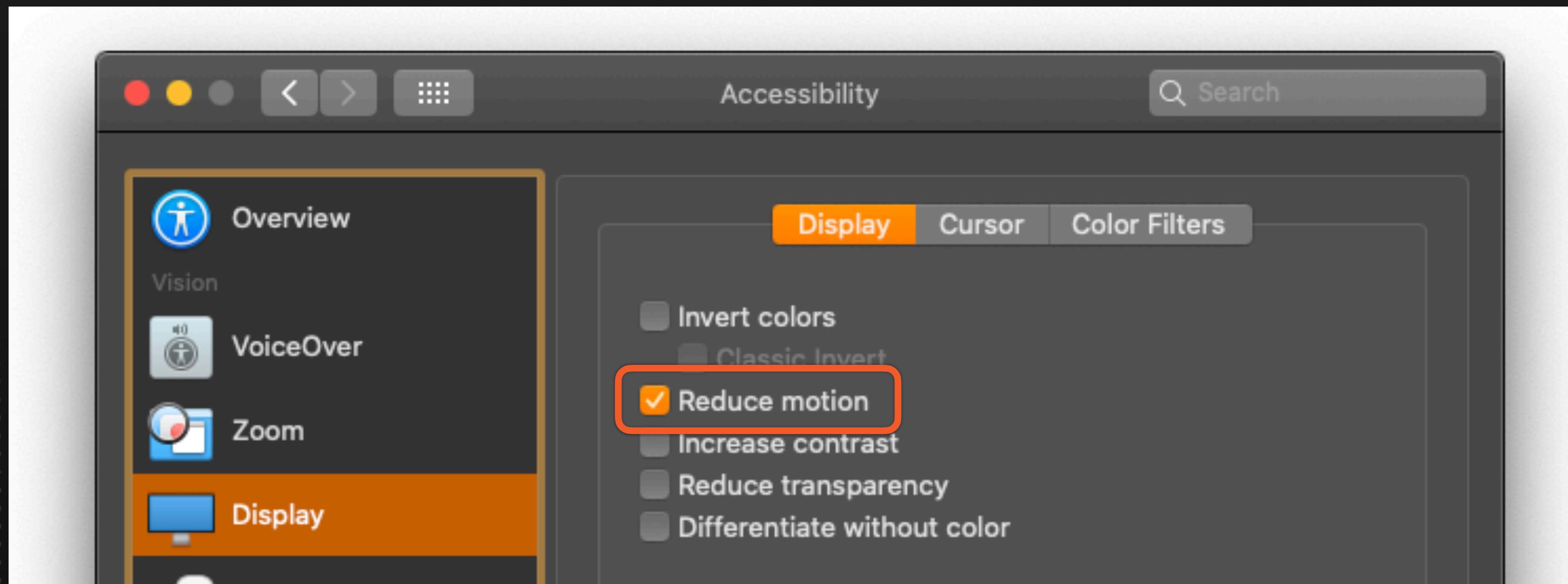
Idea



Check the course
A Tour of Web Capabilities at Frontend Masters
for examples of using hardware and sensors APIs

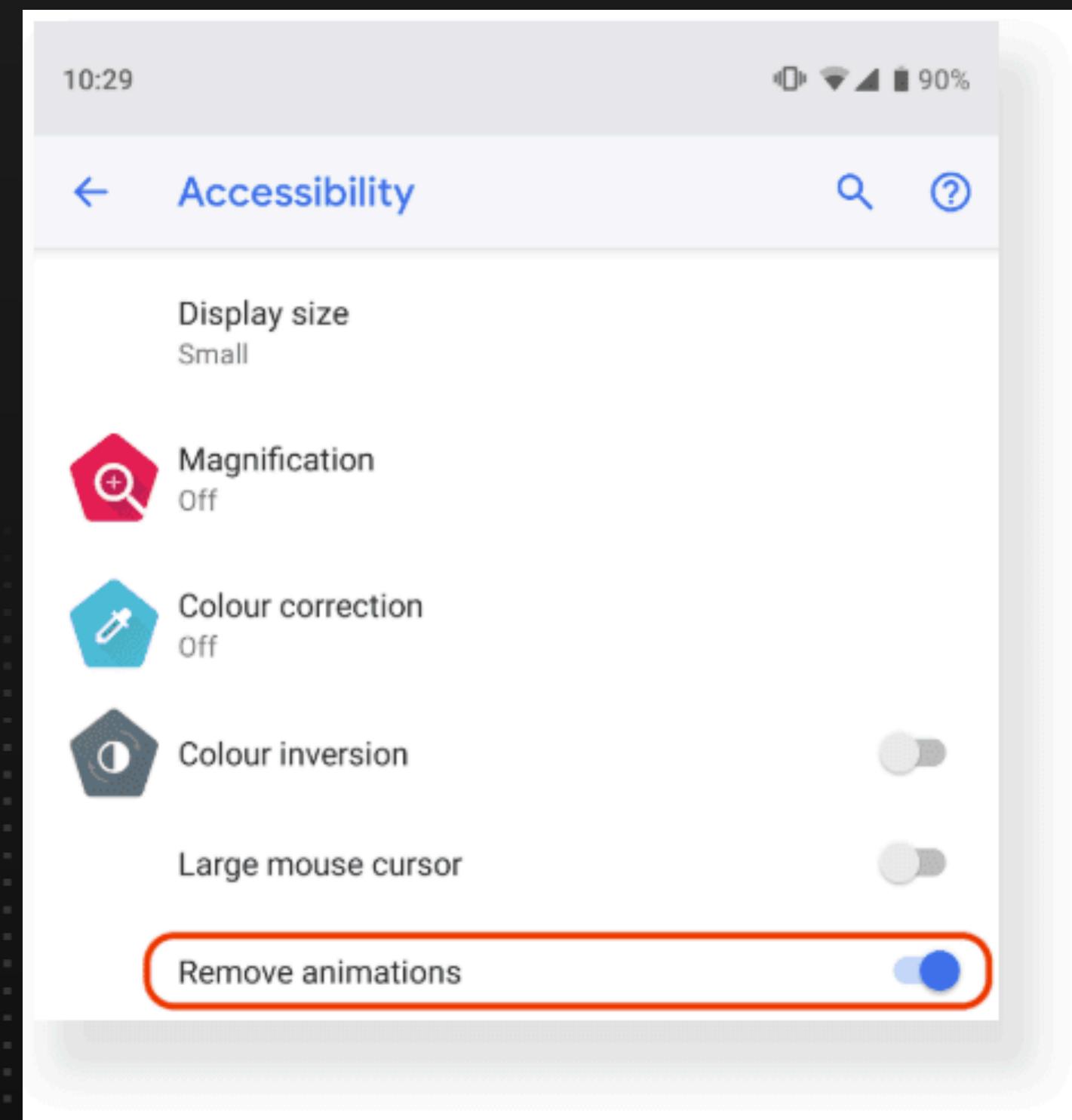
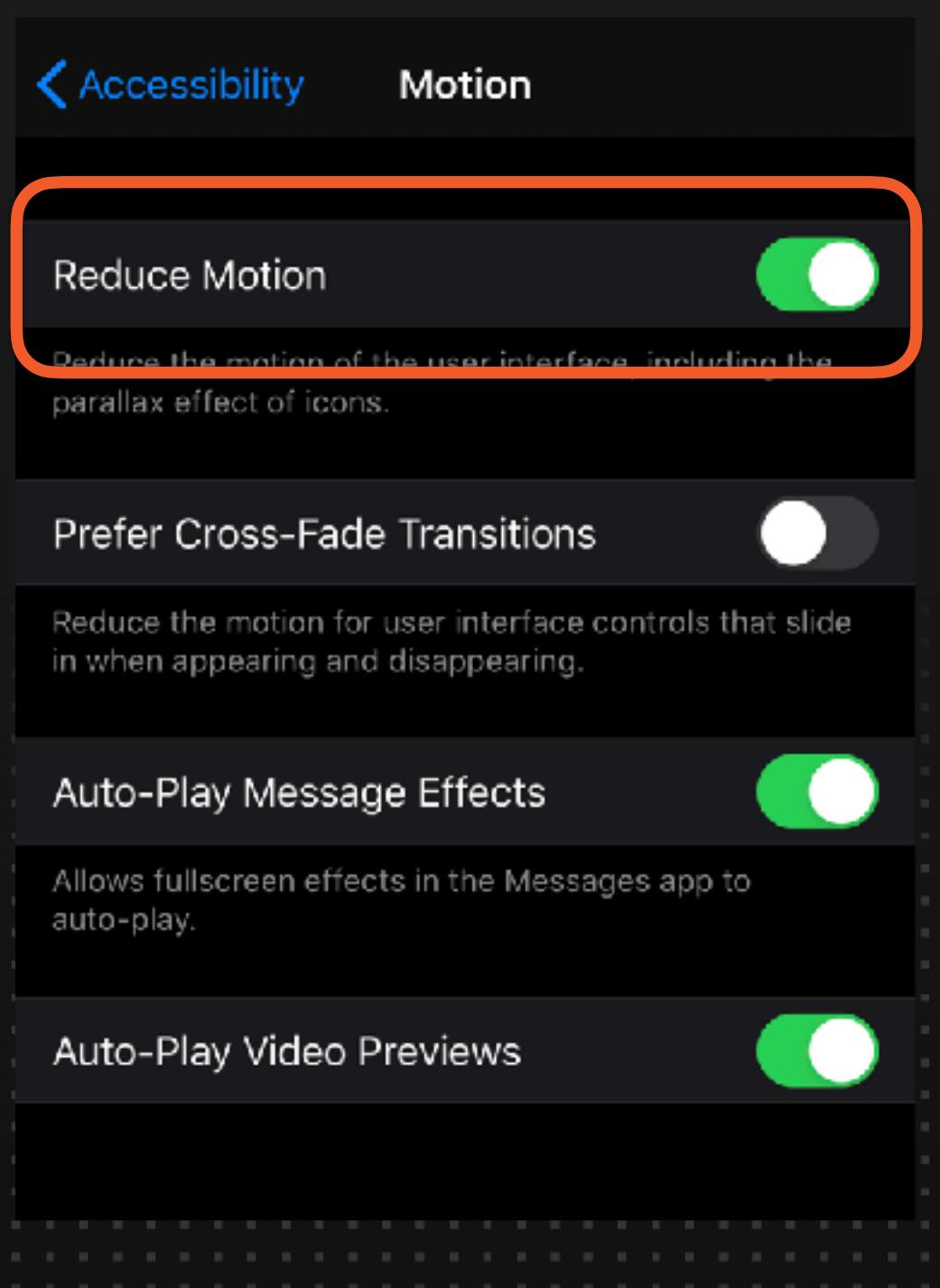
Reduced Motion

Accessibility matters



Reduced Motion

Accessibility matters



Reduced Motion Media Query

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

styles.css

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

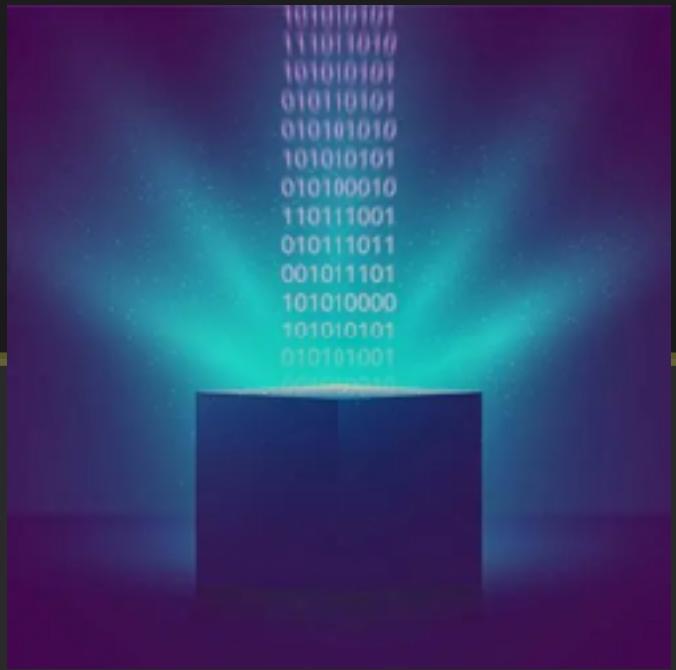
Persistent Storage Request

Firefox will ask the user, Chromium will grant or deny based on criteria

styles.css

```
if (navigator.storage && navigator.storage.persist) {  
  const granted = await navigator.storage.persist();  
  track('storage-persist-request', granted);  
}
```

Idea



For more information on this topic, check the course
Web Storage APIs at Frontend Masters



Deploying the App for User Installation

Important



Deploying a PWA is just deploying a web app

Detect Installation API Availability

Chromium-only

styles.css

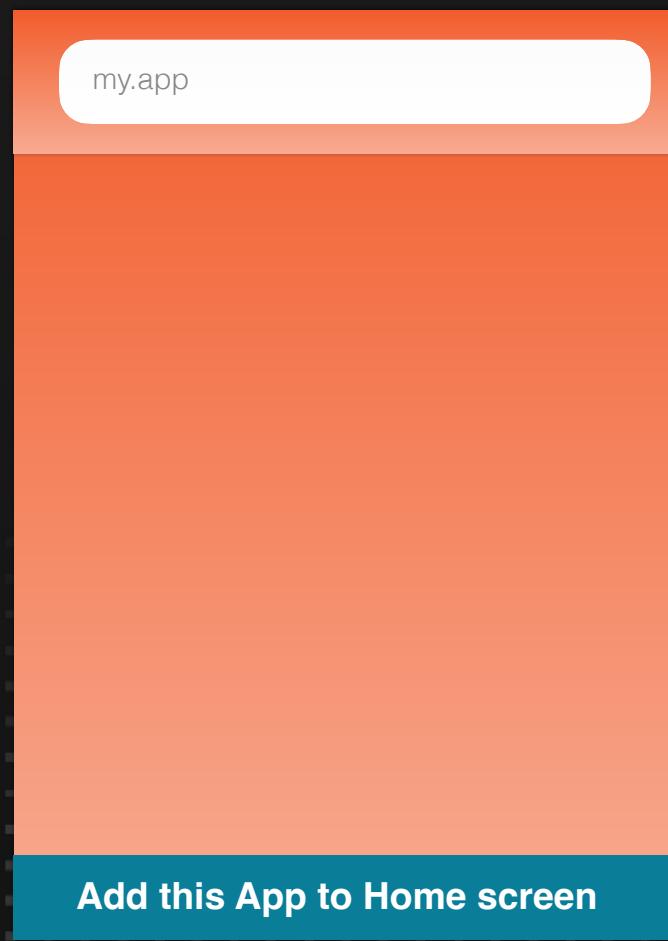
```
// Tracking installation availability
window.addEventListener('beforeinstallprompt', event => {
  track('install', 'available');
});
```

X

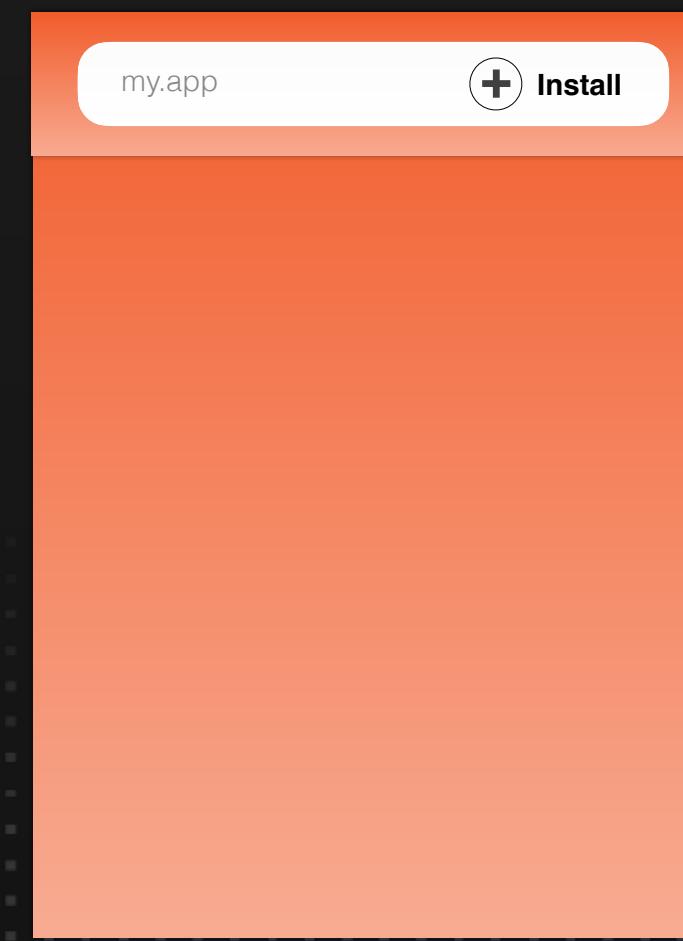
Before Install Prompt event

- It was part of Web App Manifest spec
- Now implemented by Chromium only
- Event is fired when:
 - Current URL passes the PWA criteria
 - The PWA is not already installed

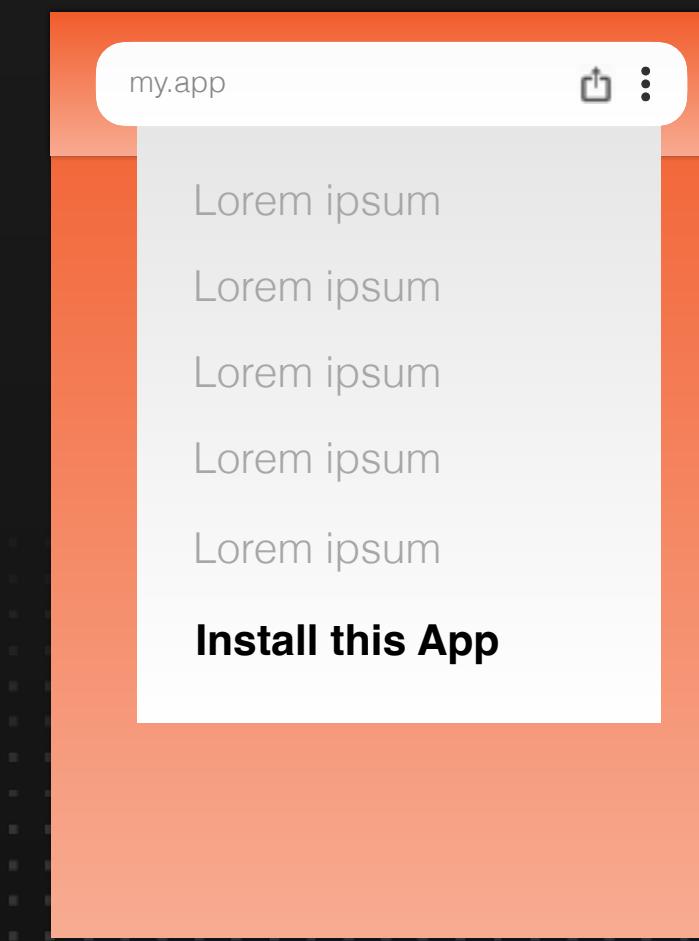
When the beforeinstallevent fires



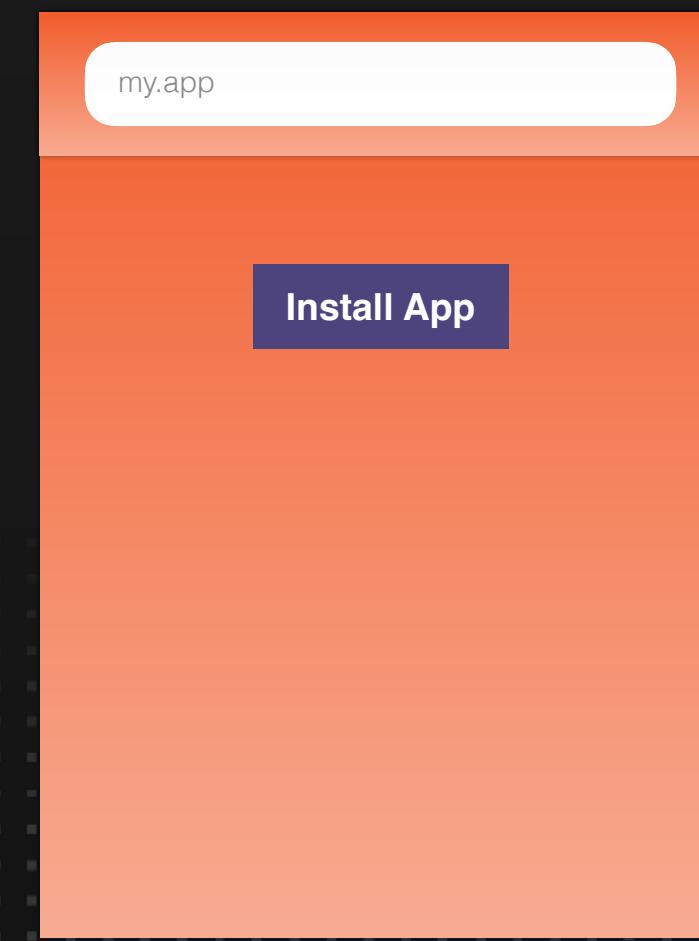
Render infobar or
banner
(cancellable)



Add a badge



Create a desktop
menu item



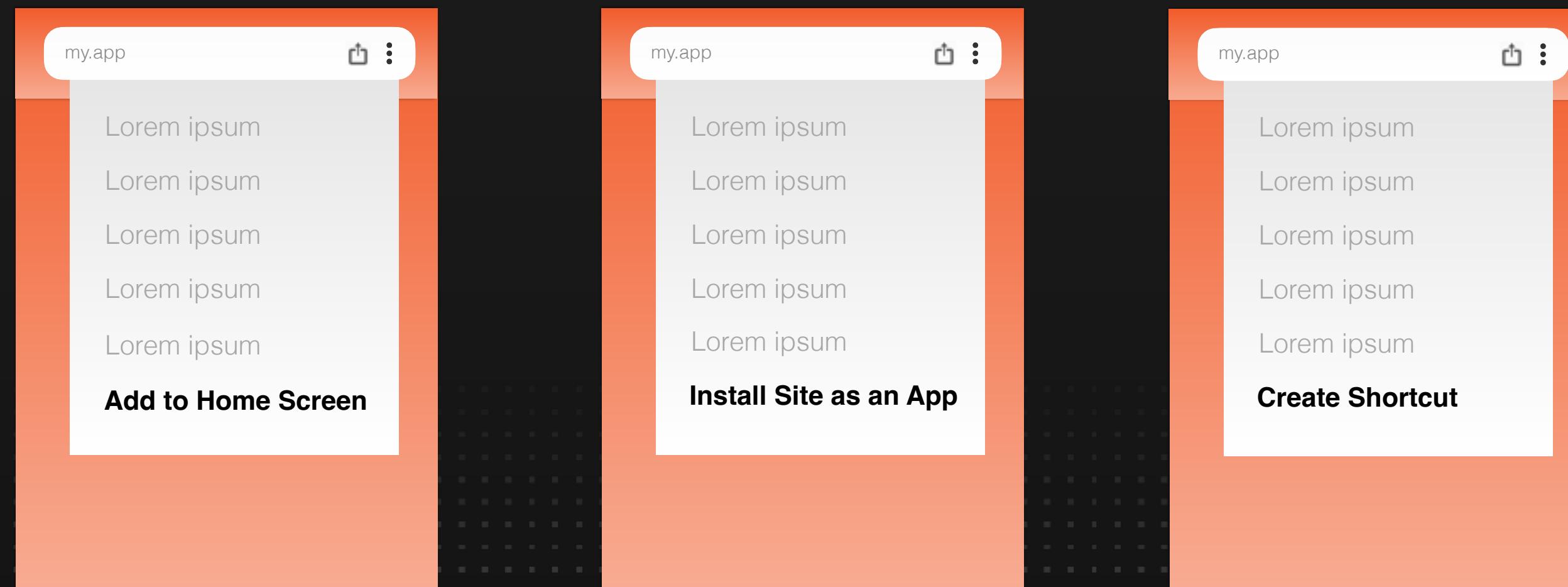
Allow developer to
provide a custom
install button

Warning



There are always-available
installation menu items that can't be
tracked

Always-available Installation Menu Items



Detect Installation

Chromium-only

styles.css

```
// Tracking PWA installation on Chromium
window.addEventListener('appinstalled', event => {
  track('install', 'installed');
});
```

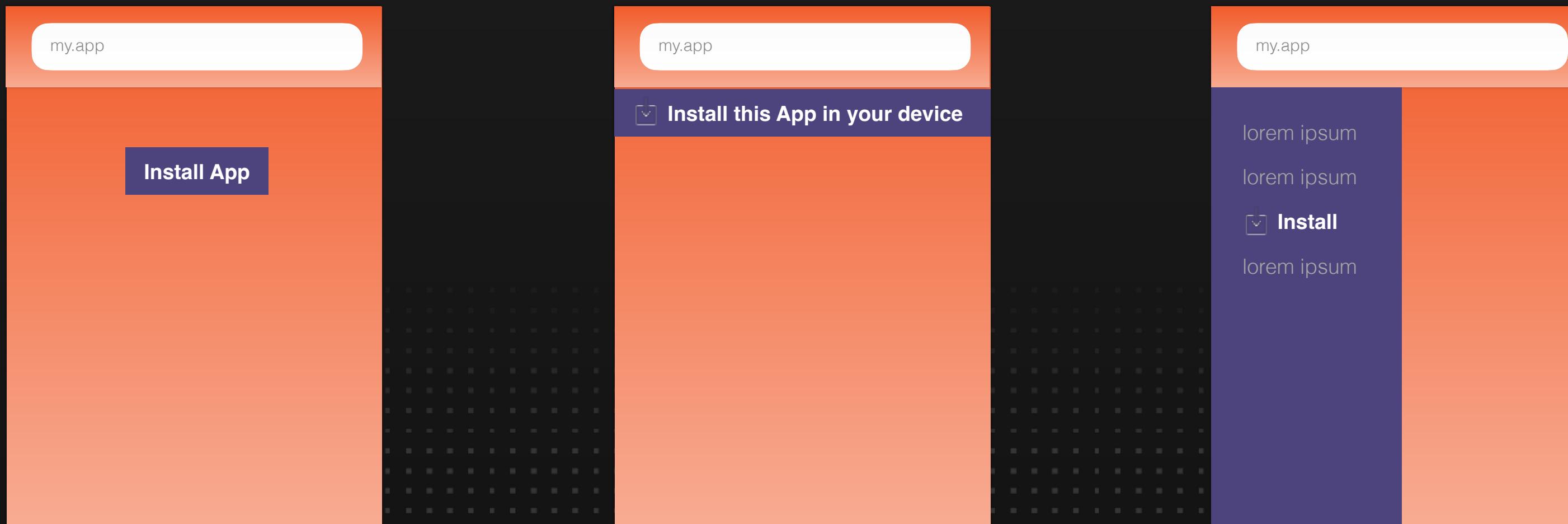
X

Warning



All these APIs are not available in
Safari or Firefox on Android.

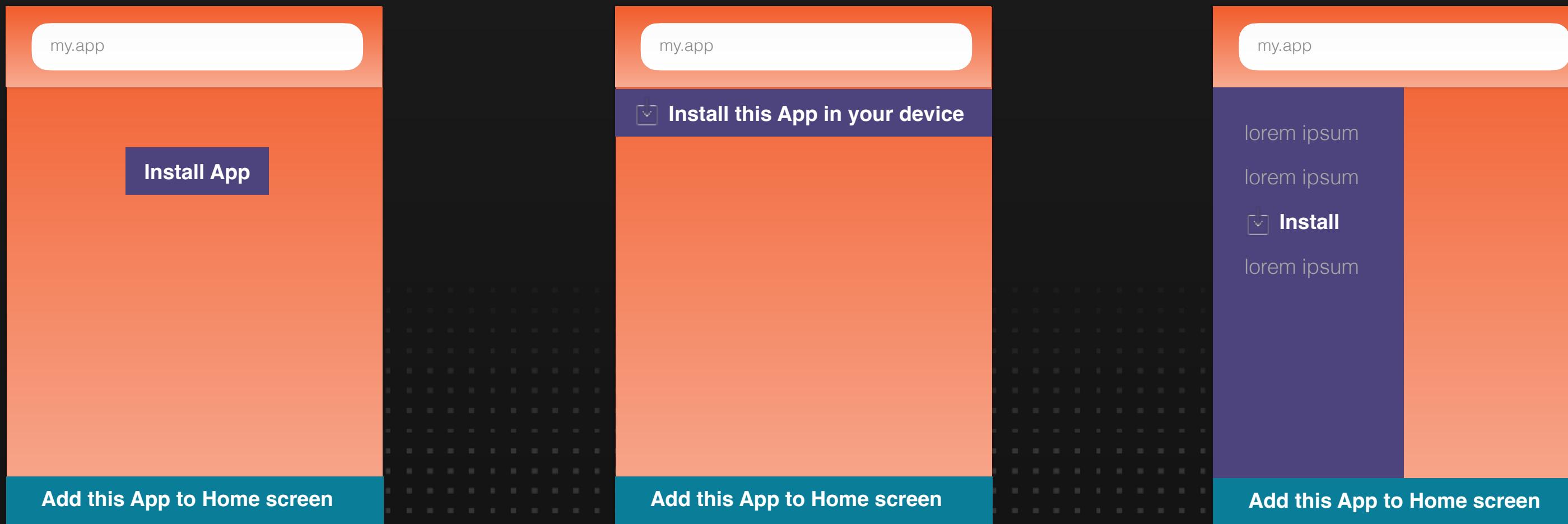
Create a Custom Install Promotion



Custom Install Promotion

- It's based on `beforeinstallprompt` event
- We need to wait for the event
- Hide the promotion if:
 - The user is not in browser display mode
 - You know the app is installed
 - You know an alternative app is installed
- While no event is fired you can:
 - Hide the promotion
 - Fallback to manual instructions

Avoid user experience duplications



Promote Installation from API

Chromium-only

styles.css

```
const bipEvent;

window.addEventListener('beforeinstallprompt', event => {
  bipEvent = event;
});

function install() {
  if (bipEvent) bipEvent.prompt();
}
```

X

Rich Installation

- Adding promotional fields we can get a better installation UX on Android and desktop
- screenshots
- description
- categories

Idea



You can still publish in App Stores!

Definition

Web App Launcher Package



Native package that renders a browser runtime in standalone mode that will load our PWA from the cloud.

Tools to create native app launcher packages

Native IDEs

CLI

PWA Builder

Helping developers build and publish PWAs

[Start a new PWA →](#)

Looking to build a new Progressive Web App? Checkout all the documentation here.

[Use dev tools →](#)

Use our VS Code extension to create, improve, and package your PWA directly in your code editor.

[Ship your PWA to app stores](#)

Start

Try a [demo url](#)



Apps packaged

Companies of all sizes—from startups to Fortune 500s—have used PWABuilder to package their PWAs.



Glass

TikTok



Warning



Not every web app can be published in app stores. Check their rules and requirements.



Recap

Best of Both Universes

Links and discoverability

Easy to deploy

Easy to update

Standards and tools

Offline access

Installed icon and standalone

OS integration

Performance and UX

Much more...

- Web Push Notifications
- Badge API
- Shortcuts
- Share API 2.0
- Share Target
- File Handling
- URL Handling

Frontend *Masters*

Web App Performance



Maximiliano Firtman
Independent Consultant

Learn to identify and resolve performance issues in your web apps! Learn about performance issues, metrics, HTTP, browser cache, and service workers. Discover basic to advanced optimization techniques and performance APIs.

5 hours, 17 minutes 

A Tour of Web Capabilities



Maximiliano Firtman
Independent Consultant

Explore what is possible with new Web APIs! Browsers can do so much now, including speech recognition, geolocation, OS integration, face detection, augmented reality, and reading from sensors, external hardware, and devices!

4 hours, 10 minutes 

Vanilla JS: You Might Not Need a Framework



Maximiliano Firtman
Independent Consultant

Explore Vanilla JavaScript, understand core concepts, work with the DOM API, handle events, and build a web application from scratch. Delve into advanced topics like SPA routing, web components, and reactive programming.

5 hours, 52 minutes 

Web Storage APIs



Maximiliano Firtman
Independent Consultant

Go beyond localStorage to use newer and more performant APIs like IndexedDB for storing JSON, CacheStorage for caching requests, and the FileSystem API for accessing the local file system.

4 hours, 8 minutes 

Web Authentication APIs



Maximiliano Firtman
Independent Consultant

Learn basic password logins to using Google to log in. Then, use WebAuthn API and Passkeys to use FaceID and biometrics to make passwordless web apps!

5 hours, 4 minutes 

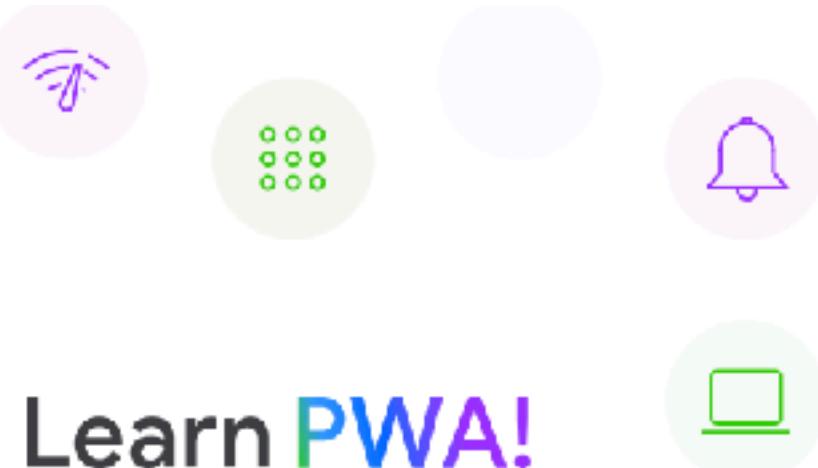
JavaScript in the Background



Maximiliano Firtman
Independent Consultant

Explore new web app capabilities to detect when your web app is in the background to execute code later, even when your web app isn't in focus.

4 hours, 59 minutes 



Learn PWA!

Search

000 Learn PWA



001 Progressive Web Apps



002 Getting started



003 Foundations



004 App design



005 Assets and data



006 Service workers

000

Learn PWA

A course that breaks down every aspect of modern progressive web app development.

Welcome to Learn Progressive Web Apps!

Welcome to Learn Progressive Web Apps!

This course covers the fundamentals of Progressive Web App development in easy-to-understand pieces. Over the following modules, you'll learn what a Progressive Web App is, how to create one or upgrade your existing web content, and how to add all the pieces for an offline, installable app. Use the



Thanks!

Maximiliano Firtman

firt.dev

X x.com/firt

in linkedin.com/in/firtman

github.com/firtman