Progressive Web Applications

The Coding Bootcamp

Class Objectives

- The benefits a progressive web app offers a user over a traditional app.
- Implement and explain the role of a web app manifest.
- Implement and explain the role of a service worker.
- Successfully cache and fetch files to deliver them in an offline experience.
- Install a PWA on both desktop and mobile devices

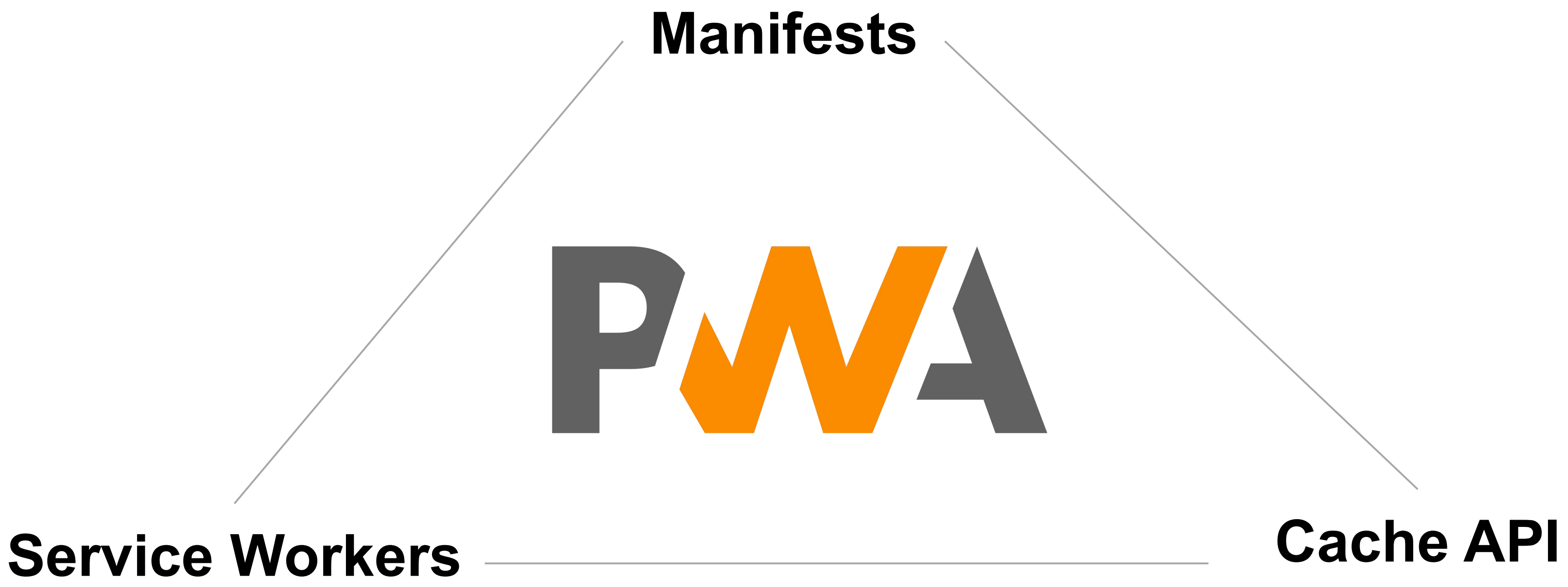
Progressive Web Applications

PWA

Progressive Web Applications

```
# Progressive Web Applications

  In this activity, you will install a progressive web application (PWA) using
your smart phone. You will also research the definition  and production of a PWA.
If you are unable to find the icons mentioned in this activity, try them in
Chrome on your computer.

  ## Instructions

  * Follow these instructions to install a PWA for your specific smartphone OS…
```

Progressive Web Applications 3- Parts



manifest.webmanifest

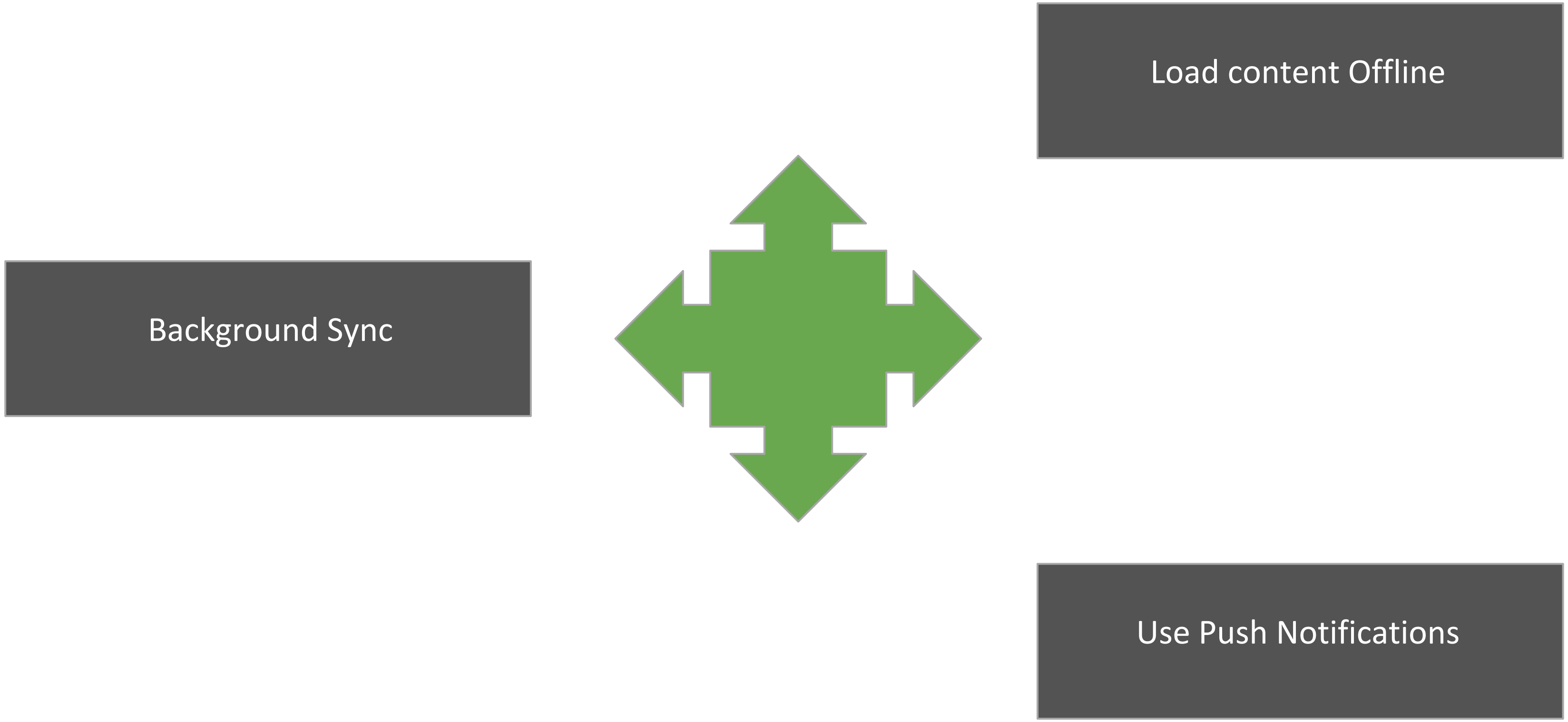
The web app manifest tells the browser about your web application and how it should behave once installed.

```
{
  "name": "Images App",
  "short_name": "Images App",
  "icons": [
    {
      "src": "assets/images/icons/icon-72x72.png",
      "sizes": "72x72",
      "type": "image/png"
    },
   …. Other Sizes…..
  ],
  "theme_color": "#ffffff",
  "background_color": "#ffffff",
  "start_url": "/",
  "display": "standalone"
}
```

Web App Manifest

```
In this activity, you will write your first progressive web application
manifest.

  ## Instructions

  * Using the instructor demo as a guide, create a manifest for the Image
Gallery app.

    * 🤔 Where do you create the `manifest.webmanifest` in the application
architecture?

    * 🤔 How do you deploy a manifest?

  * When finished, run the commands:

    * `npm install`

    * `npm run seed`

    * `npm start`
```

Service Workers Listen…



Service Workers

✅ A service worker is a script that your browser **runs in the background** on a separate thread from your webpage.

✅ **Certain functionality can *only* be implemented from within a service worker**, such as caching assets in order to make the application useable without an internet connection or notifying the browser that the application should be installable.

✅ Cache API Similar to localstorage and indexedDB in that this browser API is used for storing data. However **Cache API can be used to store entire all front end assets such as images, javascript, HTML, CSS, etc. along with API responses**.

Service Workers

✅ Service workers have a lifecycle that consists of 3 main parts.

✅ **Installation**: The service worker creates a *version-specific cache*.
✅ **Waiting**: The updated service worker waits until the existing service worker is no longer controlling clients. *This step is often skipped* with a function, since service workers rarely exist past a new service workers installation.
✅ **Activation**: This event fires *after the service worker has been installed and the previous one has been removed*.

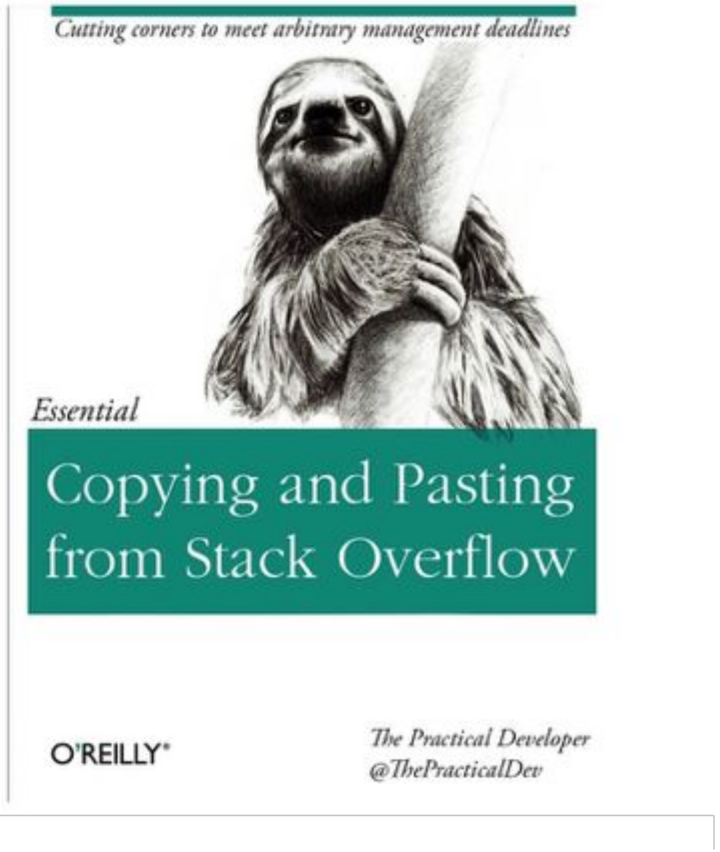
Student Do: Register Service Worker

11-Stu_Service_Workers

In this activity you will be registering your first service worker.

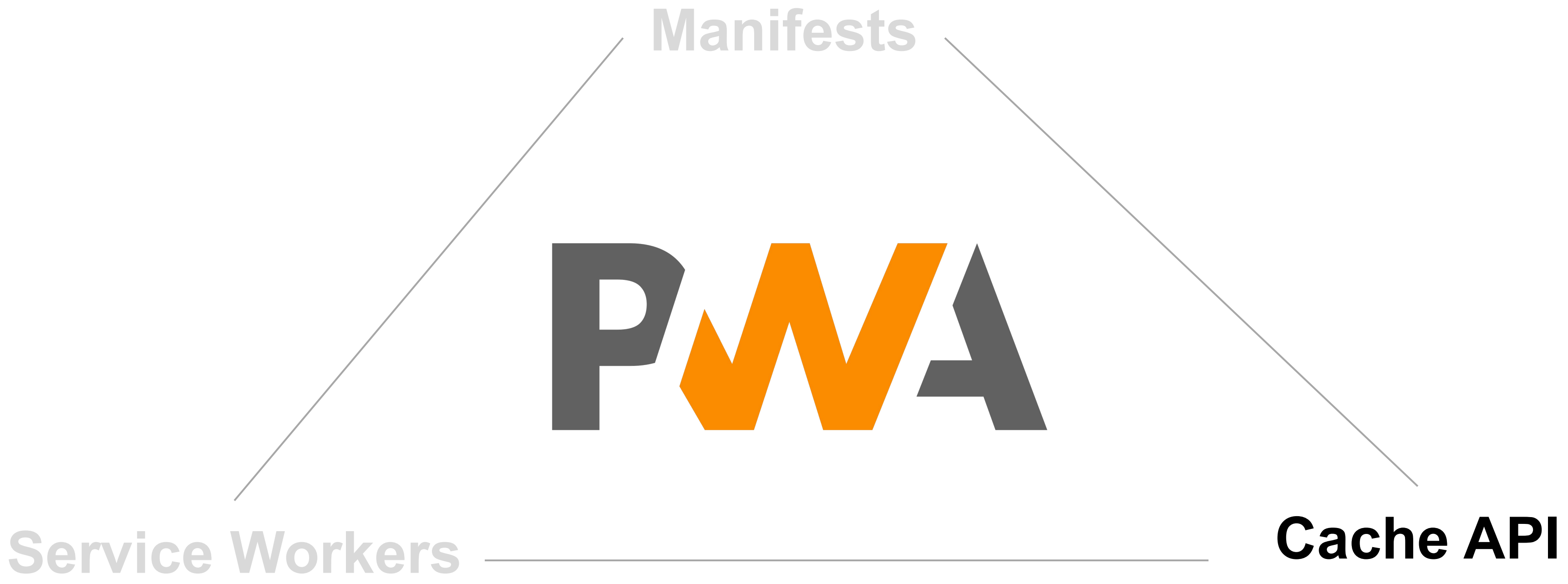
Check Readme for Instructions

Break.. Suggested Reading...


Cutting corners to meet arbitrary management deadlines
Essential
Copying and Pasting from Stack Overflow
O'REILLY
The Practical Developer
@ThePracticalDev

Software can be chaotic, but we make it work
Expert
Trying Stuff Until it Works
O RLY?
The Practical Developer
@ThePracticalDev

Creating An Offline Experience



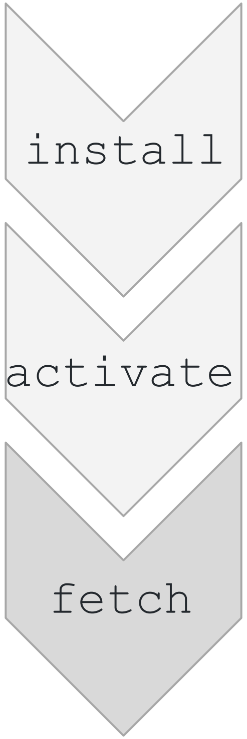
Creating An Offline Experience

✅ All files that need to be cached are stored as strings in an array.

✅ All files that need to be cached are precached in the `install` step.

✅ The `activate` step clears out the all outdated caches.

✅ The `fetch` listener intercepts all fetch requests and uses data from the cache to return a response.



Caching Files (13-Stu_Caching_Fetching_Files)

```
In this activity you will be enabling functionality to allow your application to work offline.

## Instructions

* Add the following code to your `service-worker.js` file. Use Images...

   Set Up Cache Files

   Install and Register Your Service Worker

   If done successfully, you should see your static cache in your Application tab.

   Activate Service Worker

   Fetch Files

  * If done successfully you will see your data cache in your Application tab. At this point
you should be able to put your application in offline mode for an offline experience.
```

Caching Files

☝️ What does a service worker do?

☝️ When using a service worker, can we send POST requests to an API while offline?

☝️ How many times does the install event run for each service worker?

☝️ What does `self.skipWaiting()` do?

Notetaker PWA (14-Stu_Notetaker_PWA)

```md
For this final activity you are going to convert the Notetaker that your
previously worked on into a PWA.

## Instructions

* Refer back to the activities we previously worked through to help you
accomplish the following steps.

  * Create an app manifest.

  * Register and install a service workers.

  * Cache your files and deliver and offline experience.

  * Make your app downloadable.

## BONUS

  * Push your app to heroku!
```

Questions?