Changes:

8/14/2017 - Changed sw-precache/sw-toolbox module to Workbox module

|  |  |
| --- | --- |
| PWA Instructor’s Guide *Last updated: April 2017*  [OVERVIEW & COURSE PREP](#5x1hxkdufh15)  [COLLECTING FEEDBACK](#bvwbncnjchik)  [TIPS AND TRICKS FOR STRUCTURING YOUR DAYS](#q621fy5fqres)  [DETAILED LESSON PLANS](#kix.k35vaoe7zvm7) | |
| **OVERVIEW & COURSE PREP** | |
| **Overview** | Building Progressive Web Apps is an Instructor-Led Training course that teaches web developers how to convert web pages and single-page web apps to Progressive Web Apps (PWAs). A PWA is not an API or a technology, but it is a web development approach that uses a combination of tools, technologies, and techniques already available to create targeted, ideal mobile user experiences.  This course quantifies the trend toward mobile device usage and the business benefits of building satisfying mobile web experiences. It assesses the challenges web teams face in creating quality user experiences for a variety of mobile device types and platforms. It prescribes an effective mobile development methodology for fast, engaging, and reliable web experiences using a collection of techniques, open web standards, and development and testing tools to construct web applications that are great for users and great for businesses.  This course is aimed at:   * **Audience:** Beginner-to-intermediate web developers. * **Prerequisites:** Comfortable using HTML, JavaScript, and CSS. * **Suggested Duration:** 2-day or 3-day accelerated formats. * **Venue:** Medium-sized training centers (3+ courses a week). Larger training organizations and universities, or industry workshops may also be considered. |
| **Overall Learning Objectives** | The PWA course is designed to get students thinking about how to adopt the PWA philosophy and build mobile web apps that match native applications in functionality, especially in offline behavior, application performance, user engagement, and many other types of functionality that normally require a native application. The PWA credo is to build the core of a web site using the simplest technologies and only add technologies incrementally when they enhance the experience. That’s the “progressive” in progressive web apps!  Examples of PWA technologies include HTTPS, service workers, app shell, web app manifest, home-screen install banners, IndexedDB, push notifications, responsive design, an assortment of API’s (e.g. window.fetch, Cache Storage API, sw-toolbox and sw-precache, Promises), and many other features. PWAs can (and should) use and build on these technologies to enhance web experiences and performance both online and offline.  The ultimate goal of PWA is to enhance the experience of the web for all users anywhere on the planet, including where users cannot (or will not) install native apps after an initial purchase; where networks are slow, expensive or poorly available. Another important goal is not only to make it easier to get to your app in the first place, but to make it easy for users who are engaged with your app to return and re-engage in the experience.  After students complete this course, they will:   * Understand the benefits of progressive web apps * Be familiar with the PWA design principles, technologies, and tools * Learn how to build a resilient mobile web app and integrate tooling * Register and install a service worker for offline functionality and high performance * Improve user engagement/re-engagement with a homescreen presence and push notifications * Understand the Web Payments API, discovery (that's SEO), and analytics * Build (minimally) a baseline PWA: * Served using HTTPS for security * A web app manifest to controls how the application appears * Service worker for offline functionality and high performance * Use sw-precache and sw-toolbox to automate the creation of service workers and create custom caching * Test PWAs |
| **Course Capacity** | Recommended: Up to 25 students per instructor; can add one lab assistant per 25 additional students |
| **Estimated Required Time** | * The course should be scheduled for two or three 8-hour days with 1 lunch break and 2+ mini-breaks (10 mins), typically running from 9:30a-5:00p. * 3 sessions per day, scheduled in ~1.5 hour blocks of lecture + lab exercises. * Use the [DETAILED LESSON PLANS](#kix.k35vaoe7zvm7) later in this doc to plan your course delivery. |
| **Room Setup & Equipment Required** | **Room setup:**   * Classroom or theater style auditorium with 2 whiteboards and video conferencing (GVC or similar) * Power required, as laptops will be used throughout the day   + Supply power at desks or power strips laid out on floor so that everybody can plug in   + For Google venues, coordinate with facilities to get this done * Arrange for lunch, snacks, and drinks   **Equipment required:**   * Participants must bring a laptop and charger cable * Participants should be encouraged to bring mobile devices (phones, notebooks, iPads, and so on)   **Labs:**   * Teacher and teaching assistants should help students and answer questions during exercises, as needed * If people finish early, encourage them to try the optional exercises |
| **Instructor Preparation** | * Set up the instructor’s machine following the “[Setting Up the Labs](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/setting_up_the_labs.html)” instructions. * Create an evaluation form using [Google Forms](https://www.google.com/forms/about/) and the [COLLECTING FEEDBACK](#bvwbncnjchik) guidelines later in this document |
| **One Week Before Class** | **Send this email to enrolled learners:**  Hi <*student-name> (supplied via* [*Google Apps Script*](https://www.google.com/script/start/))  Thank you for registering for Google's [Building Progressive Web Apps](https://pr-4202-dot-web-central.appspot.com/web/ilt/pwa/) course! A few reminders before the course next week:   1. The workshop runs from 9:30am-5:30pm with a lunch break. By signing up, you committed to attending the full course. 2. Bring your laptop and charger. 3. Bring mobile devices (phones, pads/iPads, notebooks, etc) for testing your app. 4. Bring your existing PWA projects (if any) for feedback and suggestions. 5. Arrive with the following software already installed:    1. A programming editor (for example, Atom, Sublime Text, Notepad++)    2. The latest version of Chrome.    3. A stable copy of [Node.JS](https://nodejs.org/en/download/) (we recommend the 6.X LTS release).   If you have time, then look at the “[Your First Progressive Web App](https://developers.google.com/web/fundamentals/getting-started/codelabs/your-first-pwapp/)” codelab to get a jump on the course content.  See you in Class! |
| **First Day of Class** | * Talk to everyone as they enter the classroom, especially on the first day. Find out their interests and what they're trying to build. Take notes. * Refer to these personal examples as you're teaching. This makes the course relevant to each person in the room.   1. If it is an especially large class, then have learners write their interests on sticky notes and post them for reference during the class.   2. If the class is small, then ask individuals to stand up and share information about where they’re from, what they want to build, and what is one thing they want to learn about. This helps you relate the modules to students’ concerns, and that makes the course more interesting and relevant for the students. * To pull these ideas into the course, capture them on a white board and periodically review them. This reassures the students that they're getting what they want, and when they get what they want then you get good ratings! * Encourage learners to enter evaluations at the end of each day and not wait until the last day. |
| **After Class To-Dos** | **Immediately after class wrap up:**   * Ask learners to fill out the evaluation forms |
| **Materials Provided** | All training materials are kept in GitBook and GitHub:   * [Codelabs](https://www.gitbook.com/book/google-developer-training/progressive-web-apps-ilt-codelabs/details) * [Concepts](https://www.gitbook.com/book/google-developer-training/progressive-web-apps-ilt-concepts/details) * Slides ([3-day](https://drive.google.com/open?id=0B5Kg0X0yIQ1PTDd3RlRzYU1UMzQ)) ([2-day](https://drive.google.com/open?id=0B5Kg0X0yIQ1PNzRsSGJfU3dkZGc))   [Videos](https://drive.google.com/corp/drive/folders/0B1r_calG-wwqZ25ESVFELXdidVk) are available for teachers to bring themselves up to speed. |

# 

# COLLECTING FEEDBACK

A good place to start is with [Google Forms](https://www.google.com/forms/about/) because they are easy to use and everything is set up for you. You just pick the type of questions you want to show (such as text questions versus a 1-to-5 scale) and write the content. In general, it is best to keep your survey questions open ended. For example, focus on text responses rather than 1-to-5 ratings.

**TIPS AND TRICKS FOR STRUCTURING YOUR DAYS**

To set up a training schedule:

* Print a copy of the [2-day](#kix.how0j4oabve) or [3-day](#kix.pihs89nrojba) schedule template, and use it during class to track the actual time it takes to present each module as training is going along.
  + Record actual start and end times in the columns provided.

The "planned time" and “Planned Duration” columns in the detailed templates provide scheduling guidelines but instructors should adjust the schedule to meet the needs or restrictions of the students.

* + Use your gathered statistics to see how far behind or ahead you are with each module and make adjustments to the start and end times each day.
  + Determine how many hours are available each day (minus a lunch break and morning and afternoon breaks).
  + Do not go longer than 1.5 hours without a break. The rule of thumb is to assume a minimum of 1.5 hours for lunch and breaks, although it is probably more realistic to schedule 2 hours.
  + During breaks, people often use break time to finish labs but encourage everyone to also get up and stretch to refresh themselves.
  + Avoid scheduling a lecture after lunch. It’s more productive to have students do actual work such as a lab that can flow into a break. Learners who finish early get a longer break.
  + For longer courses, schedule more lectures at the beginning of the day and more labs toward the end.
* Make sure everyone takes breaks. It’s important learners (and instructors) take time to rest and refresh.

**DETAILED LESSON PLANS**

**2-Day Course Plan and Schedule**

Print this lesson plan and use it to track the time it takes to present each module as training is going along; use your gathered statistics to make adjustments to the start and end times for each day.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Module** | **Planned Duration** | **Planned**  **Start** | **Actual**  **Start** | **Actual**  **Duration** | **Notes** |
| **Day 1** | | | | | |
| **Preparation Time** | 0:15 | 9:00 |  |  | Arrive early to test video conferencing and confirm room setup |
| **Buffer Time** | 0:15 | 9:15 |  |  | Allow people to get settled in the room: get coffee and snacks |
| **Intro and Agenda** | 0:15 | 9:30 |  |  | **Introductions:**   * Facilitator introduction: background/experience, why participants will trust your advice * Participant introduction around the room: name, role and background (experience) with security, what they want to get out of the course today * Write objectives on the board to have throughout the day as a reminder   **Manage participants’ expectations:**   * Summarize the agenda for the day and the aim of the program * Ask to limit phone/email use only at breaks to get the most out of the training |
| **Lecture:**  Why build Progressive Web Apps? | 0:30 | 9:45 |  |  |  |
| **Lecture:**  New techniques for responsive design | 0:30 | 10:15 |  |  |  |
| **Lab 1:** Laying Out a Responsive App | 0:30 | 10:45 |  |  | **Codelabs:**   * [Lab: Responsive Design](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_responsive_design.html) |
| **Break 1** | 0:15 | 11:00 |  |  | Encourage everyone to stretch and refresh |
| **Lecture:**  Working with Responsive Images | 0:30 | 11:15 |  |  |  |
| **Lab 2:**  Laying Out a Responsive App | 0:30 | 11:45 |  |  | **Codelabs:**   * [Lab: Responsive Images](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_responsive_images.html) |
| **Lecture:**  Core technologies (fetch and promises intro) | 0:15 | 12:15 |  |  |  |
| **Break 2: Lunch** | 1:00 | 12:30 |  |  | Ask everyone to come back on time |
| **Lecture:**  Introduction to Service Workers | 0:30 | 13:30 |  |  |  |
| **Lecture:**  Offline Quickstart | 0:15 | 14:00 |  |  |  |
| **Lab 3:**  Offline support for an existing content site | 0:30 | 14:15 |  |  | **Codelabs:**   * [Lab: Offline Quickstart](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_offline_quickstart.html) |
| **Lecture:**  Audit an existing site with Lighthouse | 0:15 | 14:45 |  |  |  |
| **Lab 4:**  Lighthouse | 0:30 | 15:00 |  |  | **Codelabs:**   * [Lab: Auditing with Lighthouse](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_auditing_with_lighthouse.html) * [Optional: Run Lighthouse from the command line](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_auditing_with_lighthouse.html#7) |
| **Break 3** | 15 min | 15:30 |  |  | Encourage everyone to stretch and refresh |
| **Lecture:**  Intro to PWA architectures | 0:15 | 15:45 |  |  |  |
| **Lab 5:**  Gulp Setup | 0:15 | 16:00 |  |  | **Codelabs:**   * [Lab: Gulp Setup](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_gulp_setup.html) |
| **Lecture:**  Workbox | 0:15 | 16:15 |  |  |  |
| **Lab 6:**  Workbox | 0:30 | 16:30 |  |  | **Codelabs:**   * [Lab: Workbox](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_workbox.html) * [Optional: Creating the service worker in the command line](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_sw-precache_and_sw-toolbox.html#6) |
| **Wrap Up/End Q&A** | 0:15 | 17:00 |  |  | **Reminders:**   * Ask them to fill out course evaluation |
|  | | | | | |
| **Module** | **Planned Duration** | **Planned**  **Time** | **Actual**  **Time** | **Actual**  **Duration** | **Notes** |
| **Day 2** | | | | | |
| **Preparation Time** | 0:15 | 9:15 |  |  | Arrive early to test the equipment and confirm room setup |
| **Intro to Day 2** | 0:15 | 9:30 |  |  | **Manage participants’ expectations:**   * Summarize the agenda for the day * Remind to limit phone/email use   **Morning review, Q&A:**  Ask learners what was the most interesting (useful) thing they learned from the previous day. This has multiple benefits:   * It helps learners think back through the day before and reawakens that information in their minds. * It immediately breaks the ice. Any questions learners have immediately rise to the surface and discussion begins! On the other hand, if you just start out the day by asking them if they have any questions, it typically causes brain freeze, very few questions, or even silence. |
| **Lecture:**  Working with Promises | 0:15 | 9:45 |  |  |  |
| **Lab 1:**  Promises | 0:15 | 10:00 |  |  | **Codelabs:**   * [Promises](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_promises.html) * [Optional: Using Promise.all and Promise.race](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_promises.html#4) |
| **Lecture:**  Working with the Fetch API | 0:15 | 10:15 |  |  |  |
| **Lab 2:**  Fetch API practice | 0:30 | 10:30 |  |  | **Codelabs:**   * [Fetch API](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_fetch_api_practice.html) * [Optional: CORS and custom headers](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_fetch_api_practice.html#7) |
| **Break 1** | 0:15 | 11:00 |  |  | Encourage everyone to stretch and refresh |
| **Lecture:**  Caching files with the Service Worker | 0:30 | 11:30 |  |  |  |
| **Lab 3:**  Caching Files with Service Worker | 0:30 | 12:00 |  |  | **Codelabs:**   * [Caching files with the Service Worker](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_caching_files_with_the_service_worker.html) |
| **Break 2: Lunch** | 1:00 | 12:30 |  |  | Ask everyone to come back on time |
| **Lecture:**  Working with IndexedDB | 0:30 | 13:30 |  |  |  |
| **Lab 4:**  IndexedDB | 0:30 | 14:00 |  |  | **Codelabs:**   * [IndexedDB](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_indexed_db.html) * [Optional: Processing orders](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_indexed_db.html#5) |
| **Lecture:**  Working with live data in the service worker | 0:15 | 14:15 |  |  |  |
| **Lecture:**  Discoverability & Analytics | 0:15 | 14:30 |  |  |  |
| **Break 3** | 0:15 | 14:45 |  |  | Encourage everyone to stretch and refresh |
| **Lecture:**  Introduction to Web Push & Notifications | 0:45 | 15:30 |  |  |  |
| **Lecture:**  Introduction to Payment Request API | 0:15 | 15:45 |  |  |  |
| **Conclusion and Feedback** | 0:15 | 16:00 |  |  | Reminders:   * Ask students to fill out course evaluation |

**3-Day Course Plan and Schedule**

Print this lesson plan and use it to track the time it takes to present each module as training is going along.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Module** | **Planned Duration** | **Planned**  **Start** | **Actual**  **Start** | **Actual**  **Duration** | **Notes** |
| **Day 1** | | | | | |
| **Preparation Time** | 0:15 | 9:00 |  |  | Arrive early to test video conferencing and confirm room setup |
| **Buffer Time** | 0:15 | 9:15 |  |  | Allow people to get settled in the room: get coffee and snacks |
| **Intro and Agenda** | 0:15 | 9:30 |  |  | **Introductions:**   * Facilitator introduction: background/experience, why participants will trust your advice * Participant introduction around the room: name, role and background (experience) with security, what they want to get out of the course today * Write objectives on the board to have throughout the day as a reminder   **Manage participants’ expectations:**   * Summarize the agenda for the day and the aim of the program * Ask to limit phone/email use only at breaks to get the most out of the training |
| **Lecture:**  Why build Progressive Web Apps? | 0:30 | 9:45 |  |  |  |
| **Lecture:**  Your audience, your content | 0:30 | 10:15 |  |  |  |
| **Break 1** | 0:10 | 10:45 |  |  | Encourage everyone to stretch and refresh |
| **Introductions via post-it** | 0:10 | 10:55 |  |  |  |
| **Lecture:**  Design for all your users | 0:25 | 11:05 |  |  |  |
| **Lecture:**  New techniques for responsive design | 0:30 | 11:30 |  |  |  |
| **Lab 1:** Laying Out a Responsive App | 0:30 | 12:00 |  |  | **Codelabs:**   * [Lab: Responsive Design](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_responsive_design.html) |
| **Break 2: Lunch** | 0:45 | 12:30 |  |  | Ask everyone to come back on time |
| **Lecture:**  Working with Responsive Images | 0:30 | 13:15 |  |  |  |
| **Lab 2:**  Laying Out a Responsive App | 0:30 | 13:45 |  |  | **Codelabs:**   * [Lab: Responsive Images](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_responsive_images.html) |
| **Lecture:**  Core technologies (fetch and promises intro) | 0:15 | 14:15 |  |  |  |
| **Lecture:**  Introduction to Service Workers | 0:30 | 14:30 |  |  |  |
| **Lab 3:**  Scripting the Service Worker | 0:45 | 15:00 |  |  | **Codelabs:**   * [Lab: Scripting the Service Worker](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_scripting_the_service_worker.html) |
| **Break 3** | 0:15 | 15:45 |  |  | Encourage everyone to stretch and refresh |
| **(Service Worker lab extension)** | 0:15 | 16:00 |  |  | Learners might need this time to reproduce examples if they didn’t have enough time do all parts of the exercise, or to try the optional labs. |
| **Lecture:**  Offline Quickstart | 0:15 | 16:15 |  |  |  |
| **Lab 4:**  Offline support for an existing content site | 0:30 | 16:45 |  |  | **Codelabs:**   * [Lab: Offline Quickstart](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_offline_quickstart.html) |
| **Wrap Up/End Q&A** | 0:15 | 17:00 |  |  | **Reminders:**   * Ask students to fill out course evaluation |
|  | | | | | |
| **Module** | **Planned Duration** | **Planned**  **Time** | **Actual**  **Time** | **Actual**  **Duration** | **Notes** |
| **Day 2** | | | | | |
| **Preparation Time** | 0:15 | 9:15 |  |  | Arrive early to test the equipment and confirm room setup |
| **Intro to Day 2** | 0:15 | 9:30 |  |  | **Manage participants’ expectations:**   * Summarize the agenda for the day * Remind to limit phone/email use   **Morning review, Q&A:**  Ask learners what was the most interesting (useful) thing they learned from Day 1. This has multiple benefits:   * It helps learners think back through the day before and reawakens that information in their minds. * It immediately breaks the ice. Any questions learners have immediately rise to the surface and discussion begins! On the other hand, if you just start out the day by asking students if they have any questions, it typically causes brain freeze, very few questions, or even silence. |
| **Lecture:**  Working with Promises | 0:15 | 9:45 |  |  |  |
| **Lab 1:**  Promises | 0:15 | 10:00 |  |  | **Codelabs:**   * [Lab: Promises](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_promises.html) * [Optional: Using Promise.all and Promise.race](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_promises.html#4) |
| **Break 1** | 0:15 | 10:15 |  |  | Encourage everyone to stretch and refresh |
| **Lecture:**  Working with the Fetch API | 0:15 | 10:30 |  |  |  |
| **Lab 2:**  Fetch API practice | 0:30 | 10:45 |  |  | **Codelabs:**   * [Lab: Fetch API](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_fetch_api_practice.html) * [Optional: CORS and custom headers](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_fetch_api_practice.html#7) |
| **Lecture:**  Caching files with the Service Worker | 0:30 | 11:15 |  |  |  |
| **Lab 3:**  Caching Files with Service Worker | 0:30 | 11:45 |  |  | **Codelabs:**   * [Lab: Caching files with the Service Worker](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_caching_files_with_the_service_worker.html) |
| **Break 2: Lunch** | 0:45 | 12:15 |  |  | Ask everyone to come back on time |
| **(Cache API lab extension)** |  |  |  |  | Learners might need this time to reproduce examples if they didn’t have enough time do all parts of the exercise, or to try the optional labs. |
| **Lecture:**  Intro to PWA Architecture | 0:15 | 13:00 |  |  |  |
| **Lecture:**  Working with IndexedDB | 0:30 | 13:15 |  |  |  |
| **Lab 4:**  IndexedDB | 0:30 | 13:45 |  |  | **Codelabs:**   * [Lab: IndexedDB](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_indexed_db.html) * [Optional: Processing orders](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_indexed_db.html#5) |
| **Lecture:**  Working with live data in the service worker | 0:15 | 14:15 |  |  |  |
| **Exercise:**  Real Estate App exercise (data management strategies) | 0:45 | 14:30 |  |  |  |
| **Break 3** | 0:15 | 15:15 |  |  | Encourage everyone to stretch and refresh |
| **Lecture:**  Auditing an existing site with Lighthouse | 0:15 | 15:30 |  |  |  |
| **Lab 4:**  Lighthouse | 0:30 | 15:45 |  |  | **Codelabs:**   * [Lab: Auditing with Lighthouse](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_auditing_with_lighthouse.html) * [Optional: Run Lighthouse from the command line](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_auditing_with_lighthouse.html#7) |
| **Q&A, Wrap-up of day 2** | 0:45 | 16:15 |  |  | Reminders:   * Ask students to fill out course evaluation |
|  | | | | | |
| **Module** | **Planned Duration** | **Planned**  **Time** | **Actual**  **Time** | **Actual**  **Duration** | **Notes** |
| **Day 3** | | | | | |
| **Preparation Time** | 0:15 | 9:15 |  |  | Arrive early to test the equipment and confirm room setup |
| **Intro to Day 3** | 0:15 | 9:30 |  |  | **Manage participants’ expectations:**   * Summarize the agenda for the day * Remind to limit phone/email use   **Morning review, Q&A:**   * Ask learners what was the most interesting (useful) thing they learned from Day 2. |
| **Lecture:**  Gulp Setup | 0:15 | 9:45 |  |  |  |
| **Lab 1:**  Gulp Setup | 0:30 | 10:00 |  |  | **Codelabs:**   * [Lab: Gulp Setup](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_gulp_setup.html) |
| **Lecture:**  sw-toolbox and sw-precache | 0:15 | 10:30 |  |  |  |
| **Lab 2:**  sw-toolbox and sw-precache | 0:30 | 10:45 |  |  | **Codelabs:**   * [Lab: sw-precache and sw-toolbox](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_sw-precache_and_sw-toolbox.html) * [Optional: Creating the service worker in the command line](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_sw-precache_and_sw-toolbox.html#6) |
| **Break 1** | 0:15 | 11:15 |  |  | Encourage everyone to stretch and refresh |
| **Lecture:**  Introduction to Web Push & Notifications | 0:45 | 11:30 |  |  |  |
| **Lab 3:**  Integrating Web Push | 0:45 | 12:15 |  |  | Codelabs:   * [Lab: Integrating Web Push](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_integrating_web_push.html) * [Optional: Identifying your service with VAPID](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_integrating_web_push.html#vapid) * [Optional: Best practices](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_integrating_web_push.html#best) |
| **Break 2: Lunch** | 0:45 | 13:00 |  |  | Ask everyone to come back on time |
| **Lecture:**  Discoverability & Analytics | 0:15 | 13:45 |  |  |  |
| **Lab 4:**  Integrating Analytics | 0:30 | 14:00 |  |  | **Codelabs:**   * [Integrating Analytics](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_integrating_analytics.html) |
| **Lecture:**  Payment Integration | 0:15 | 14:30 |  |  |  |
| **Lecture:**  e-Commerce app walkthrough | 0:30 | 14:45 |  |  |  |
| **Lab 5:**  e-Commerce: Create a service worker | 0:15 | 15:15 |  |  | **Codelabs:**   * [E-Commerce Lab 1: Create a Service Worker](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/e-commerce-lab-1-create-a-service-worker.html) |
| **Lab 6:**  e-Commerce: Add to the home screen | 0:15 | 15:30 |  |  | **Codelabs:**   * [E-Commerce Lab 2: Add to Homescreen](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/e-commerce-lab-2-add-to-homescreen.html) |
| **Lab 7:**  e-Commerce: Gulp setup | 0:15 | 15:45 |  |  | **Codelabs:**   * [Lab: Gulp Setup](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/lab_gulp_setup.html) |
| **Lab 8:**  e-Commerce: Implementing the payments API | 0:30 | 16:00 |  |  | **Codelabs:**   * [E-Commerce Lab 3: PaymentRequest API](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/e-commerce-lab-3-paymentrequest-api.html) |
| **Lab 9:**  e-Commerce: Using sw-precache and sw-toolbox | 0:15 | 16:30 |  |  | **Codelabs:**   * [E-Commerce Lab 4: sw-precache and sw-toolbox](https://google-developer-training.gitbooks.io/progressive-web-apps-ilt-codelabs/content/docs/e-commerce-lab-4-sw-precache-and-sw-toolbox.html) |
| **Conclusion and Feedback** | 0:15 | 16:45 |  |  | Reminders:   * Ask students to fill out course evaluation |