Course : Creating cross-platform mobile applications

Practical course - 4d - 28h00 - Ref. LIJ
Price : 2100 € E.T.

NEW

Cette formation s'adresse aux développeurs et chefs de projet ayant des bases en web (HTML, CSS, JavaScript, XML) salariés des branches professionnelles relevant de l'OPCO Atlas. Elle vise à permettre la création d'applications mobiles multiplateformes à partir d'un seul code source. Le programme couvre les principes du développement mobile multiplateforme, la découverte des frameworks courants, la conception d'une application, la création d'interfaces responsives, et les tests multi-appareils.

Teaching objectives

At the end of the training, the participant will be able to:

- ✓ Understand the principles of cross-platform mobile development
- ✓ Discover the most popular mobile development frameworks
- ✓ Designing a mobile application
- ✓ Developing a responsive and interactive user interface
- ✓ Testing the application on different devices

Intended audience
For OPCO Atlas members: developers, project managers.

Prerequisites
Knowledge of the Web, HTML, CSS, JavaScript and XML.

Practical details

Hands-on work
Discussions, practical work.

Teaching methods
To optimize the learning experience, e-learning modules can be provided before and after the classroom session or virtual class, at the participant's request.

Course schedule

(1) HTML and CSS - Creating modern, semantic websites - Content digital learning pre-training

- HTML basics.
- Structure and organize an HTML page.
- Formatting HTML with CSS.
- Use semantic HTML.
- Create HTML forms.

Digital activities

This online training course introduces the basics of HTML and CSS for designing modern websites. Participants will learn how to structure a page with text, media and interactive elements, how to use *id* and *class* attributes, and how to create a first stylesheet. The course also covers semantic HTML and the new features of HTML5, before concluding with the creation of a complete contact form.

(2) A reminder of basic technologies

- Presentation of terminal families: Smartphones, tablets, their specific features.
- HTML, CSS and JavaScript technologies.
- Review of modern development tools (NodeJS, NPM, Bower, Gulp...).
- Introduction to Webkit and Gecko engines...

Hands-on work

Set up the mobile development environment.

(3) On-board solutions

- Overview of existing technologies and their development environments: limitations, cost, portability.
- Objective-C and Swift (iPhone, iPad), Java (Android), C# .Net (Windows 10, Xamarin), Tizen, Firefox OS.
- Method of publishing an application in official operator catalogs (Apple Store, Google Play, etc.).
- Administrative procedures (iPhone Developer Program, Certificates...).

Hands-on work

Solving definition puzzles.

(4) Differences between traditional web and mobile applications

- Browser differences: JavaScript and CSS engines, plug-in limitations (Flash, PDF...), other specificities.
- Different screen sizes and solutions.
- Hardware differences (CPU, memory).

Hands-on work

End-of-day MCQs.

(5) Page structure

- META and specific tags: Viewport and dynamic CSS settings.
- Traditional page construction tags.
- Table construction.
- Links and dedicated accesskey options.
- Images and the importance of images on terminals (format, image weight)
- p and span text. HTML 5 div and canvas blocks.

Hands-on work
Create a mobile first page with Scalable Vector Graphics (SVG) integration and manipulation.

(6) Form and CSS

- Forms. New data types (daterange, slider...). Numeric keyboard activation.
- CSS, importance in multi-channel mode. Element positioning, navigation between elements (zindex, display...).
- Bringing CSS3 to HTML5. Simplify CSS creation and maintenance with Bootstrap, Sass, Compass and Less.
- Importance of DOM for multi-terminal porting.

Hands-on work
Implement an input form, buttons and components, and rotate the terminal using CSS.

(7) Events and protocols

- Mobile environment (sensors, battery, network detection, etc.).
- Terminal-specific multitouch events (gesture, touch, drag & drop, etc.).
- Communication protocols (WebSocket, Ajax), advantages of Node.js.
- Graphics library: drawing API (rectangle, line...), color palette, image manipulation.
- W3C compliance testing: validation tools.

Hands-on work
Setting up a WebSocket server with Node.js. Network outage detection.

(8) Overview of cross-platform frameworks

- Advantages of using a framework.
- Choosing a mobile framework.
- Capacitor (Cordova) and its publishing tools.
- Angular, a framework for mobile development.
- Ionic, building native applications with Capacitor and Angular.

Hands-on work
Implement and discover the different frameworks available on the market.

(9) Implementation with Ionic

- Creating a project.
- Integration with Android Studio and Xcode.
- Plug-in creation.
- Debug a multiplatform application.
- Push notifications.
- Material Design frameworks (Polymer, Materialize CSS...).

Hands-on work
Create a mobile project with Ionic and Material Design, integrate Capacitor and generate a multi-OS application.

(10) Data storage on the mobile

- Set up a strategy "disconnected mode".
- Manifest and caching of mobile site resources.
- Browser-integrated SQLite and indexedDB databases.
- SQL language and administration tools.
- JavaScript management and control (table creation, queries, etc.).
- Use of Google Gears for non-HTML 5 devices.
- Client-side cache management for offline work (localStorage, sessionStorage).

Hands-on work
Create a note manager with storage in the embedded database.

(11) Cartography and geolocation

- Google Maps management.
- Options to support mobile geolocation.

Hands-on work
Display the map in relation to the mobile's location and display ded markers on the map.

(12) Testing and deploying a mobile application

- Test on simulator.
- Test and deploy on mobile devices.
- Deploy on Google Play.
- Deploy on App Store.
- CI/CD integration.

Hands-on work
Test and deploy on simulators and mobiles. Deployment procedures on blinds.

(13) Kotlin - Les bases indispensables - Post-training digital learning content

- Introducing Kotlin.
- Kotlin fundamentals.
- Functions.
- Coroutines.
- Object-oriented programming.
- Android development.
- Native development.
- JavaScript development.
- Server-side development.

Digital activities

This online training course presents the Kotlin language, its fundamentals, its various functions, the notion of coroutine, object-oriented programming with a demonstration of developing an application in Android Studio, how to use Kotlin for native or server-side development and how Kotlin can generate JavaScript code.

Dates and locations

REMOTE CLASS
2026 : 31 Mar., 23 June, 6 Oct., 15 Dec.

PARIS LA DÉFENSE
2026 : 24 Mar., 16 June, 29 Sep., 8 Dec.

LILLE
2026 : 31 Mar., 23 June, 6 Oct., 15 Dec.