



Mobile Web

A Comprehensive Overview

IT 4403 Advanced Web and Mobile Applications

IT 4213 Mobile Web Development <http://it4213.azurewebsites.net>

IT 5443 Web Tech and App Development <http://it5443.azurewebsites.net>

Jack G. Zheng

Ver. Spring 2020 (Since 2013)

Overview



This lecture notes provides a high level overview of mobile web. This overview is comprehensive to cover as many aspects as possible but keep them at the high level:

- What is mobile web and why do we care about it?
 - Mobile web definition, scope, and context
 - Differences and challenges of using web on mobile devices
 - Key supporting technologies
- Mobile website strategies: independent mobile site, one web (adaptive, responsive, hybrid)
- Mobile web app and web technologies used for native mobile apps
- Development tools, frameworks, and services
- General design considerations and principles
- Learning resources

Data/Reports Relevant to Mobile Web



- <http://gs.statcounter.com>
 - Web traffic by operating systems, device type/vendor, browser type
- <http://netmarketshare.com>
 - Web browsing by device type (mobile accounts for about 50%)
- Internet Trends by Mary Meeker <https://www.bondcap.com/#archive>
 - Mobile usage as % of Web usage
 - Time spent per adult user per day with digital media
- <https://deviceatlas.com/blog>
- Cisco Mobile Visual Networking Index (VNI) Forecast
 - <https://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html#~mobile-forecast>
 - Average data connection speed
 - Total of mobile connected device
 - Average traffic (data usage) per device
 - The percentage of mobile data traffic out of all fixed and mobile data traffic
 - The percentage of mobile connections will be 'smart' connections
 - Mobile traffic per mobile-connected end-user device per month

What is Mobile Web?



Mobile web refers to the use and development of web sites and web applications that are delivered over the Internet to smart mobile devices (and optimized for these devices).

- Web sites vs. web apps
 - Web apps are also web sites, just more functional oriented; they are too delivered via browsers, and share the same web site UI design principles and practices.
 - Mobile web apps look like and are used like native apps.
 - <https://www.scnsoft.com/blog/mobile-web-app-types>
- It's mainly about user experience, particularly with a focus on user interfaces and interactions.
 - Mobile web sites: focus on the optimization of web sites to be used on mobile devices.
 - Mobile web apps: emphasizes functional features specifically designed for mobile devices. Web apps are also focused on functional features besides user interface and interaction.
- It has also impacted the application level architecture:
 - Single page app
 - Progressive web app
 - Serverless architecture

Extended Meaning of Mobile Web



- Narrowly speaking, mobile web (sites or apps) is delivered to browsers. They are dependent on browsers.

CSE 3203 and IT 5443 only provide a high level mobile web overview module.
IT 4213 (the whole course) covers details of mobile web site design.

- Broadly speaking, mobile web also covers the use of web technologies in mobile app development, where browsers are not used. These methods may include:
 - Hybrid web app
 - Use of web components (like WebView) to display web pages in native apps
 - mBaaS, web APIs and services that are used by various kinds of applications.

Mobile App Development Options

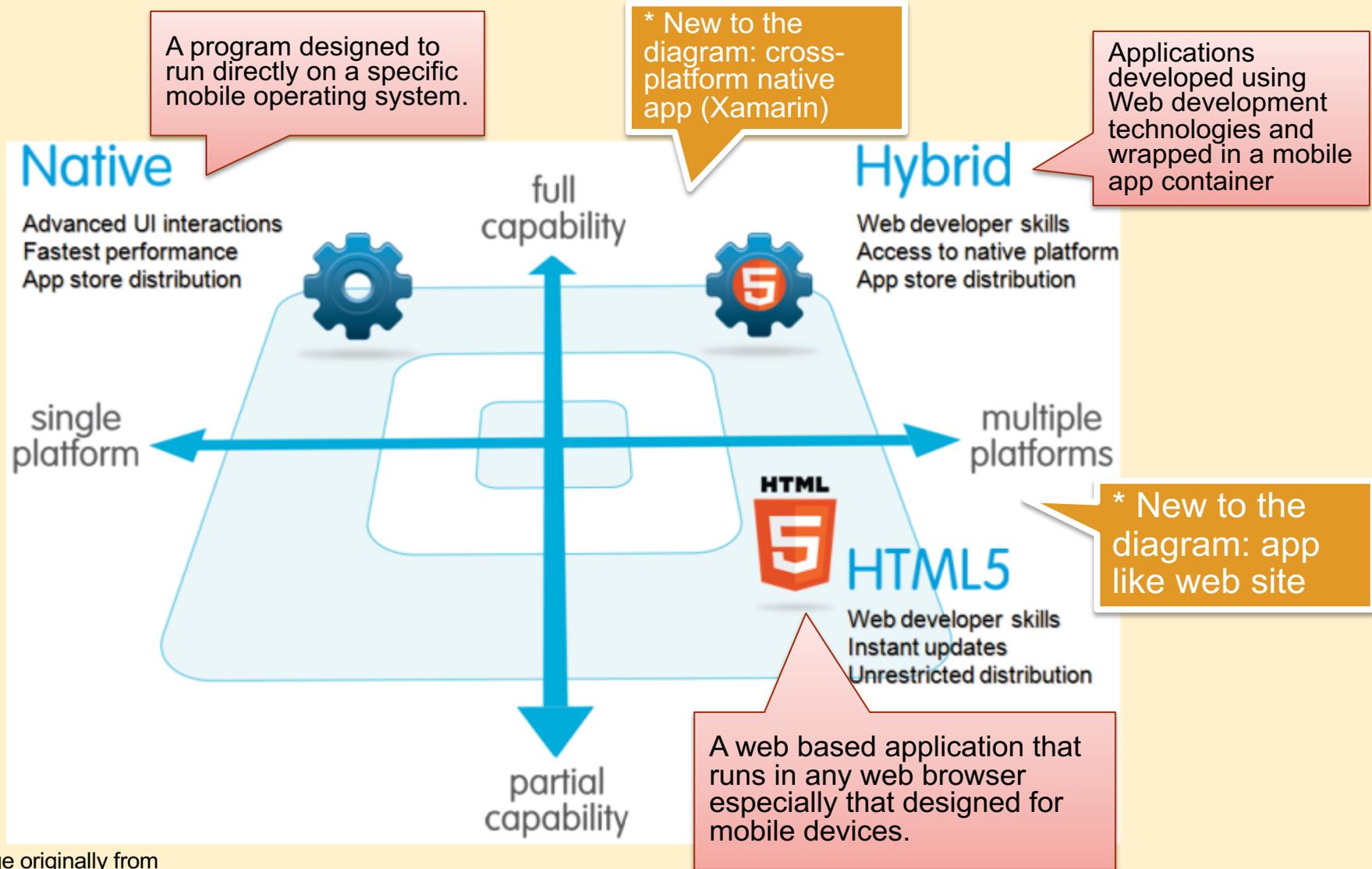


Image originally from https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options

Comparison of the Three Development Options



	Native Apps	Mobile Web	Hybrid
Device features	Can fully utilize platform/OS features if allowed, such as camera, location, sensors, and interactions with file systems and other apps.	Access to devices features are limited to the browser environment.	Can utilize platform/OS features if provided by the SDK, which are usually provided as JavaScript APIs.
UX/UI	Best; smooth interaction as UI is very responsive.	Not as good as native app; may have delays.	Looks like an app, but may have performance issues.
Development Platform	Applications are platform dependent. They have to be written and/or compiled to a specific OS*. Example: <ul style="list-style-type: none">- Android (Java) (Android)- iOS (Objective C, Swift)- Windows (.Net, C#)- * Xamarin (all)	Web applications are cross platform and can use standard languages such as HTML/CSS/JavaScript. Programs runs in browsers, although they may have slightly differences in different browsers.	Hybrid applications are developed using standard web technologies, but compiled to each specific platform.
Distribution / Installation	App Store, or direct downloaded; need installation	Web; no installation	App Store, or direct downloaded; need installation

<https://www.scnsoft.com/blog/mobile-web-app-types>

<https://www.toptal.com/android/developing-mobile-web-apps-when-why-and-how>

<http://www.telerik.com/blogs/everything-you-wanted-to-know-about-native-hybrid-and-web-apps-but-were-afraid-to-ask>

<http://www.sitepoint.com/native-hybrid-or-web-apps/>

<http://www.ymedialabs.com/hybrid-vs-native-mobile-apps-the-answer-is-clear/>

App vs. Web



The old desktop vs. web?

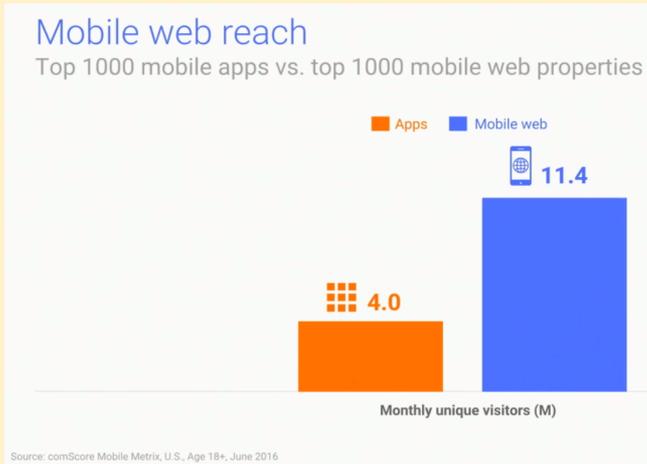


<http://xkcd.com/1367/>

Why Web?



Web has a broader reach



- Mobile Web Is Top Of Funnel, Mobile App Is Bottom Of Funnel
<https://avc.com/2015/09/mobile-web-is-top-of-funnel-mobile-app-is-bottom-of-funnel/>

“The mobile web is the window of your store. Users window shop on your mobile website. Getting them to download and install and use your mobile app is like getting them to come into the store. And that’s where the action is long term.”

<https://www.youtube.com/watch?v=m-sCdS0sQO8>

- What kind of apps is better with Web?
 - Information and content focused (media, magazine, news, forum); simple interaction (mainly tapping for navigation); highly dynamic content; frequent updates.
 - Example: Yahoo, Financial Times, Quora

Consumers Are Choosing To Shop Using The Mobile Web Over Apps Today

<https://www.business.com/articles/mobile-apps-vs-mobile-web-do-you-have-to-choose/>

“How did you perform the following shopping-related activities on your smartphone in the past three months?”



Differences of Web on Mobile



- People use mobile devices differently because of their mobility
 - Prefers short and simple interactions
 - Use for multiple purposes and interactions between apps.
 - Mobile device is not the only platform; usually used together with other devices – multi-device experience
 - <https://www.thinkwithgoogle.com/advertising-channels/mobile-marketing/device-use-marketer-tips/>
- Screen
 - Screen size is smaller <http://www.statista.com/chart/2269/smartphone-shipments-by-screen-size/>
 - Screen resolution is lower and density is higher <http://developer.android.com/about/dashboards/index.html>
 - Fragmented market: properties vary a lot among devices
 - Orientation: vertical screens accounts for 29% of usage overall
- Interaction method: touch is different form pointing device like mouse
 - <https://www.nngroup.com/articles/mouse-vs-fingers-input-device/>
- Connection and speed
 - Mobile device connections are less stable, inconsistent, and slower; variable connectivity; frequent switching among networks; limited bandwidth and usage.
 - <https://queue.acm.org/detail.cfm?id=2441756>
 - <https://www.thinkwithgoogle.com/marketing-resources/experience-design/speed-is-key-optimize-your-mobile-experience/>
- Device capability:
 - Slower processors and memory
 - Limited multitasking and simultaneous browser tabs/windows
 - More device functionalities integrated: cam, mic, sensors, GPS, etc.
- Browser capability
 - Compatibility: browser behaviors are different from desktop browsers
 - Fixed browser size (on the same device)
 - Viewport difference and automatic zooming
 - Delay: a more noticeable delay as the application resides on the web vs. directly on the native platform
 - Weaker plug-in support, no Flash, PDF, etc.

Understand these differences and limitations will help us develop websites and applications tailored for mobile devices.

This subject will be covered in details in IT 4213 module 2.

<https://www.paradoxlabs.com/blog/mobile-vs-desktop-10-key-differences/>

The Problems



- Why make websites mobile friendly?
 - Non-mobile-friendly sites create user experience and functional problems
 - Search engines also include mobile friendliness as in ranking - <https://webmasters.googleblog.com/2018/03/rolling-out-mobile-first-indexing.html>
 - A real story <https://cheekymonkeymedia.ca/blog/mobile-friendly-does-it-really-matter>
- Design/content problem – not optimized for mobile devices
 - Content too small and difficult to view
 - Content hidden
- Interaction problems
 - Links are crowded and difficulty/inaccurate to tap
 - Mouse hovering does not work mobile
 - Slow and error-prone typing
- Performance problems because of connection issues and device capability
- Some interesting problems documented
 - <http://bradfrost.com/blog/post/mobile-web-problems/>
 - <http://wtfmobileweb.com/>

Use a tool to examine the problems!

- <https://testmysite.thinkwithgoogle.com>
- <https://ready.mobi>

Major Problematic Sites



- Not mobile friendly (as of Jan 2020)
 - <https://www.cefconnect.com>
 - <https://www.georgiastandards.org>
 - <http://art.yale.edu>
 - <http://berkshirehathaway.com>
 - <https://www.weather.gov>
 - <http://www.blue-ray.com>
 - <http://www.senate.ga.gov>
 - <https://finviz.com>
 - <http://cobb.com>
 - See more: <http://cubicle-h.blogspot.com/2018/08/website-mobile-friendliness-testing.html>
- KSU web applications
 - <https://owlexpress.kennesaw.edu/prodban/bwckschd.p disp dyn sched>
 - <https://kennesaw.curriculog.com/>
- Particularly problematic for small businesses:
https://medium.com/@dave_sloan/only-30-of-small-businesses-have-mobile-friendly-websites-ad6ef9f71eb5
- See some studies here
 - <https://canonicalized.com/mobile-friendly-study/>
 - <https://www.markbrinker.com/good-bad-medical-website-design>

More and more major sites have been changed mobile friendly since 2013 when this course was developed.

Mobile Web Friendliness Testing



- Mobile Web friendliness
 - Mobile web friendliness can be helpful to think of it in terms of three goals for improving your site's user experience: Presentation, Content, and Performance.
<https://developer.mozilla.org/en-US/docs/Web/Guide/Mobile/Mobile-friendliness>
- Testing tools
 - Website based viewers
 - <https://material.io/tools/resizer/>
 - <http://www.responsinator.com>
 - Automated testing tools
 - <https://search.google.com/test/mobile-friendly>
 - <https://www.bing.com/webmaster/tools/mobile-friendliness>
 - <https://ready.mobi>
 - Loading speed test: <https://testmysite.thinkwithgoogle.com>
 - Browser tools
 - <https://www.digitalcitizen.life/emulate-mobile-device-desktop-browser>
 - <https://developers.google.com/web/tools/chrome-devtools/device-mode/>
 - <https://developers.google.com/web/tools/chrome-devtools/device-mode/emulate-mobile-viewports>
 - https://developer.mozilla.org/en-US/docs/Tools/Responsive_Design_Mode
 - <https://docs.microsoft.com/en-us/microsoft-edge/devtools-guide/emulation>
 - Other tools
 - <https://www.browsersync.io>

Use these tools to test the websites in the previous slides.

This subject will be covered in details in IT 4213 module 11

Key Technologies for Mobile Web



- HTML5/CSS3/JavaScript
 - HTML5 represents a major advancement on the client development
 - More powerful CSS3 and JavaScript make client applications more interactive and responsive
- Mobile browsers
 - Mobile web are delivered through mobile browsers, which have reduced features than their desktop counterparts.
 - Modern web browsers provide some new capabilities to create native-like web apps, such as launch from the home screen, run offline, access internal hardware, etc.

HTML 5 and Mobile Web



- How does HTML 5 play a role in facilitating mobile computing?
- Standardize mobile web development
 - Eliminates the need to create, maintain, and support apps for different mobile platforms (native platforms)
- Enriches client processing and storage
- HTML5 provides a set of features and tools for building mobile apps
 - Mobile friendly tags/attributes (number, phone - activate relevant phone features)

```
<a href="tel:+14155557777">Order Pizza Now!</a>  
<a href="sms:+14155558888?body=Hello">Text me!</a>
```

- JavaScript APIs to interact with device features

Rise of JavaScript



- #1 in GitHub since 2014
 - <https://octoverse.github.com/projects#languages>
- #1 in StackOverflow since 2013
 - <https://insights.stackoverflow.com/survey/2018>
- Most in demand and salaries, 2015
 - <https://gooroo.io/GoorooTHINK/Article/16300/Programming-languages--salaries-and-demand-May-2015>
 - <http://www.techrepublic.com/article/here-are-the-3-most-in-demand-coding-languages-and-where-you-can-find-a-developer-job/>
- Other rankings
 - #1 The RedMonk Programming Language Rankings since 2015
<https://redmonk.com/sogrady/2018/08/10/language-rankings-6-18/>
 - #2 in student programmers in hackathon (#1 is HTML/CSS) <http://studenthackers.devpost.com/2015.html>
 - #2 in Developer Economics survey <https://dashboard.developereconomics.com/?survey=de15#mobile>
- SharePoint/WordPress is moving to JavaScript
 - <https://arc.applause.com/2015/11/24/wordpress-javascript-calypso/>
 - <https://developer.wordpress.com/calypso/>
 - <https://developer.wordpress.com/2015/11/23/the-story-behind-the-new-wordpress-com/>

Mobile Browsers



- Mobile web are mainly delivered through mobile browsers. The use of web through mobile browsers has been increasing and passed the desktop browser use.
 - <http://gs.statcounter.com/platform-market-share/desktop-mobile-tablet>
- Mobile browsers have a number of differences from their desktop counterparts.
 - The bad
 - Mobile browsers run with reduced features
 - Weaker plug-in support, no Flash, PDF, etc.
 - No developer tools
 - Browser behaviors are different, like fixed browser size (on the same device), viewport difference and automatic zooming, switch between tabs, etc.
 - The good
 - Modern web browsers provide some new capabilities to create native-like web apps, such as launch from the home screen, run offline, access internal hardware, etc.
 - Browsers have increase its capabilities to work with mobile devices through browser APIs; many of these are part of HTML5. Common types of APIs:
 - XMLHttpRequest makes it possible to load additional content from the Web without loading a new document, a core component of AJAX.
 - Form field type to work with mobile input
 - URL type
 - The Geolocation API makes the user's current location available to browser-based applications.
 - Offline use, data storage, etc.
 - Touch events, drawing
 - Several APIs make the integration of Web applications with the local file system and storage seamless (Indexed Database).
 - <https://www.w3.org/standards/webdesign/script>
 - <https://www.w3.org/standards/techs/mobileapp>
- However, the standardization of these APIs are more troublesome. Many of these APIs are browser specific.
 - https://developer.chrome.com/apps/api_index
 - https://developer.mozilla.org/en-US/docs/Web/API/Using_the_Browser_API
- Currently, two major mobile browsers are used (together represent over 80% of the market): Chrome and Safari (mostly on iOS devices)
 - Chrome has been growing fast since 2014 to become the top mobile browser
 - <https://netmarketshare.com/browser-market-share.aspx?id=browsersMobile>
 - <http://gs.statcounter.com/browser-market-share/mobile/worldwide>
 - <https://www.statista.com/statistics/272664/market-share-held-by-mobile-browsers-in-the-us/>
- Additional market share data
 - https://en.wikipedia.org/wiki/Usage_share_of_web_browsers
 - <http://www.w3counter.com/globalstats.php>

Mobile browsers will be covered in details
in IT4213 module 2

Mobile Website Delivery Strategy



Two basic approaches

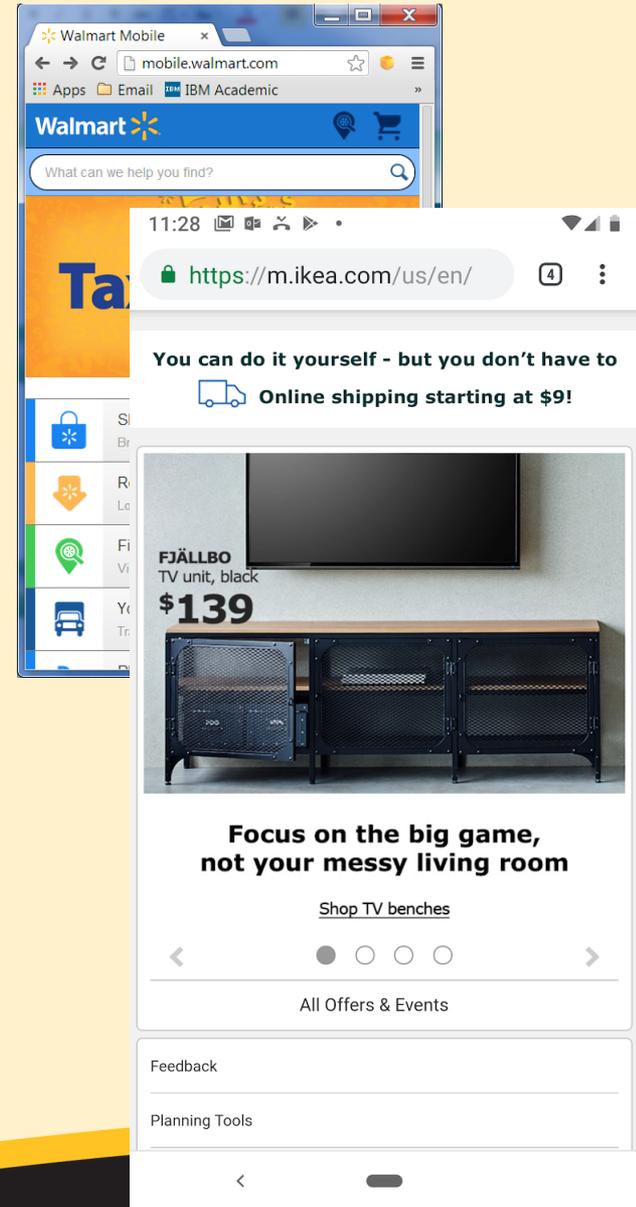
1. A separate mobile version site: has its own domain and address, and is independent from the main site.
 - Ikea: <http://m.ikea.com/us> (jQuery Mobile)
 - Newegg: <https://m.newegg.com>
 - Nike: <https://m.nike.com>
 - Staples: <https://m.staples.com>
 - Wikipedia: <https://en.m.wikipedia.org>
 - Goodwill: <https://mgwdonate.ging.org> (used a third party service, more like an app)

Extended reading: Why Separate Mobile & Desktop Web Pages?

<https://www.nngroup.com/articles/mobile-site-vs-full-site/>
<http://www.lukew.com/ff/entry.asp?1390>

2. One Web

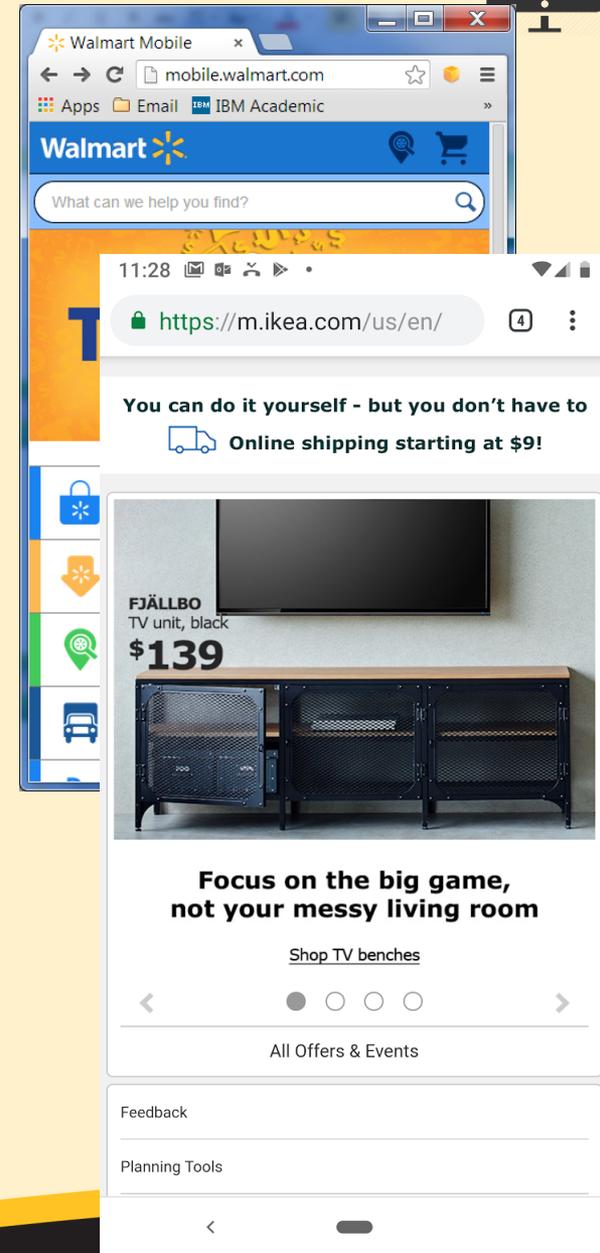
- One Web means making, as far as is reasonable, the same information and services available to users irrespective of the device they are using, but with different presentations (UI)
- <http://www.w3.org/TR/mobile-bp/#OneWeb>



Separate Mobile Site

- Separate mobile site has its own domain and address, and is independent from the main site.
- A separate mobile site can best meet mobile user experience. Build a separate mobile-optimized design (or mobile site) if you can afford it.
 - <https://www.nngroup.com/articles/mobile-site-vs-full-site/>
- Some examples:
 - Ikea: <http://m.ikea.com/us> (jQuery Mobile)
 - Newegg: <https://m.newegg.com>
 - Nike: <https://m.nike.com>
 - Staples: <https://m.staples.com>
 - Wikipedia: <https://en.m.wikipedia.org>
 - Goodwill: <https://mgwdonate.ging.org> (used a third party service, more like an app)
- Some companies has moved away from separate mobile site, such as
 - Costco (m.costco.com)
 - Walmart (mobile.walmart.com)
 - Ebay (m.ebay.com)

Extended reading: Why Separate Mobile & Desktop Web Pages?
<http://www.lukew.com/ff/entry.asp?1390>



One Web



- One Web means making, as far as is reasonable, the same information and services available to users irrespective of the device they are using, but with different presentations (UI)
- It does not mean that the content is available in exactly the same representation across all devices.
- The context of mobile use, device capability variations, bandwidth issues and mobile network capabilities all affect the representation.
- <http://www.w3.org/TR/mobile-bp/#OneWeb>
- Three basic approaches
 - Responsive web design (response to screen size)
 - Adaptive (dynamic serving based on device detection and adaptation)
 - <https://www.wired.com/2013/05/the-two-flavors-of-a-one-web-approach-responsive-vs-adaptive/>
 - Hybrid: RESS (*Responsive Design + Server Side Components*)
<http://www.lukew.com/ff/entry.asp?1392>
- Examples
 - <http://mashable.com> (responsive)
 - <http://google.com> or Wikipdia.org (adaptive)

Key reading: design for multi-screen experiences
<https://www.thinkwithgoogle.com/articles/building-websites-multi-screen-consumer.html>

Responsive Web Design



Responsive Web Design (RWD) is a Web design approach aimed at crafting sites to provide an optimal viewing experience, easy reading and navigation with a minimum of resizing, panning, and scrolling, across a wide range of screen sizes and devices.

- Ethan Marcotte coined the term responsive web design and defined it to mean fluid grid/ flexible images/ media queries in a May 2010 article in A List Apart
 - <http://alistapart.com/article/responsive-web-design>
- Basic principles and practices
 - Fluid grid - no horizontal scrolling
 - Adaptive/flexible image
 - Media query
- See more details at <https://www.edocr.com/v/yxdmelxy/jgzheng/Responsive-Web-Design>



This subject will be covered in details throughout IT 4213

Some interesting infographics about RWD

<https://www.webfx.com/blog/web-design/infographics-learn-responsive-web-design/>

Responsive Examples



- Responsive examples
 - <http://kennesaw.edu>
 - <https://cobbcounty.org>
 - <http://mashable.com>
 - <http://mediaqueri.es> (more examples here)

Adjust the browser width and see the changes or use a tool like responsinator.com or browser developer tools.

Palantir



- Non-responsive examples
 - <https://www.weather.gov> (not mobile friendly)
 - <http://www.atlantawatershed.org/> (adaptive but not responsive)
 - See more: <http://cubicle-h.blogspot.com/2018/08/website-mobile-friendliness-testing.html>

Adaptive (Dynamic Serving)



- Adaptive design aims to deliver website UI (content, style, function, etc.) and user experience that adapts to the capabilities of the device/browser.
 - Also called dynamic serving as in <https://www.thinkwithgoogle.com/marketing-resources/experience-design/building-websites-multi-screen-consumer/>
 - Adaptation can be done either at the server side or at the client side - <https://www.wired.com/2013/05/the-two-flavors-of-a-one-web-approach-responsive-vs-adaptive/>
- Of the Alexa 100 sites, fully 80% use adaptive web design in their delivery.
 - <https://mobiforge.com/news-comment/adaptive-web-design-dominates-in-the-webs-top-brands>
- Some examples (notice the webpage does not respond to size change well):
 - <http://google.com>
 - <https://www.yahoo.com>
 - <http://dealnews.com>
 - <https://gillette.com>
 - <https://github.com/vmware>
- Responsive design can somewhat be considered as a kind of adaptive design. It adapts UI based on screen size (width mainly). But adaptive design considers more features than just screen size.
 - <http://bradfrost.com/blog/post/the-many-faces-of-adaptive-design/>

Feature/Device Detection



- Adaptive design heavily uses feature detection and other sound progressive enhancement techniques.
- These features may include
 - Screen size and capabilities (orientation, touch, etc.)
 - Device/OS type
 - Browser type/version
 - Hardware features
- In the separate mobile site approach, device detection is also often used to redirect users to mobile or desktop version site.
- Feature detection techniques
 - Parsing User-Agent strings: <https://deviceatlas.com/blog/user-agent-parsing-how-it-works-and-how-it-can-be-used>
 - Using JavaScript testing: Modernizr <https://modernizr.com>

Hybrid - RESS



- The two approaches are not mutually exclusive. Many sites combine techniques to achieve the best results.
- RESS: Responsive Design + Server Side Components
 - A single set of page templates define an entire Web site for all devices (responsive) but key components within that site have device-class specific implementations that are rendered server side (server side components).
 - <http://www.lukew.com/ff/entry.asp?1392>
- Comparison of the three:
 - <http://www.lukew.com/ff/entry.asp?1509>

Mobile Web App



- Mobile web apps are still websites and delivered through browsers. They are more focused on a particular functionality without cluttered information and content.
 - <https://www.androidpolice.com/2018/01/26/opinion-mobile-web-apps-finally-becoming-alternative-native-apps/>
 - App-like web site looks and acts like a native mobile app (browser hidden, etc.) but is actually a web site
 - This will give the website an “app” look and feel without installing anything at first
 - But they are still dependent on the browser to drive them
 - See an example “Sencha Fastbook ”here:
<https://www.youtube.com/watch?v=XkDWLcErzGs>
 - Some features:
 - Launch from home screen (like an app): <http://www.howtogeek.com/196087/how-to-add-websites-to-the-home-screen-on-any-smartphone-or-tablet/>
 - Running offline
 - OS level notification
 - Hardware capabilities like GPS, etc.
- Extended reading:
<https://www.sencha.com/blog/create-web-apps-with-the-capabilities-of-native-apps/>
- Take it to another level: progressive web app (optimized app-like web site)

Progressive Web App



- Progressive web applications (PWAs) are web applications that load like regular web pages or websites but can offer the user functionality such as working offline, push notifications, and device hardware access traditionally available only to native applications. PWAs combine the flexibility of the web with the experience of a native application.
 - https://en.wikipedia.org/wiki/Progressive_web_applications
- Examples
 - Twitter <https://buildfire.com/progressive-web-app/>
- More resources (not covered in IT 4213)
 - <https://developers.google.com/web/progressive-web-apps>
 - <http://arc.applause.com/2015/11/30/application-shell-architecture/>
 - <http://developer.telerik.com/featured/what-progressive-web-apps-mean-for-the-web/>
 - <http://gonehybrid.com/are-progressive-web-apps-the-future-of-mobile-apps/>

Apps using Web Technologies



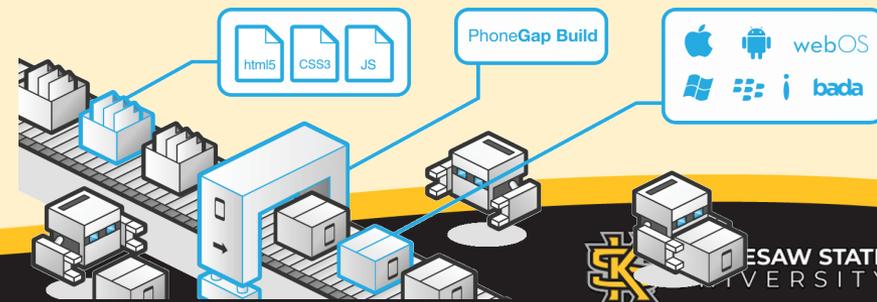
- Mobile web also refers to the use of web technologies for mobile app development that are not dependent on web browsers
- These apps may include:
 - Hybrid web app
 - Use of web components (Web View) to display web pages in native apps
<https://www.addthis.com/blog/2017/04/04/in-app-browsers-explained/#.XEfp7VxKibh>
 - mBaaS, through the use of web APIs and services, used by various kinds of apps

This course will NOT cover these subjects.

Hybrid Web Apps



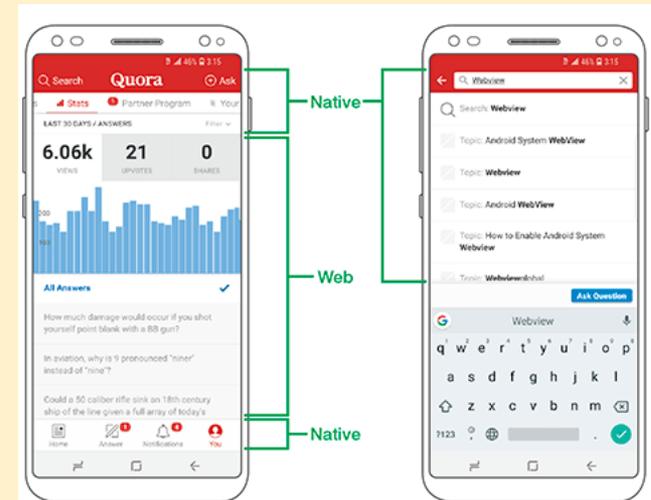
- Hybrid apps are apps developed based on the web technologies (HTML, CSS, JavaScript) and wrapped in a platform specific native container
 - Similar to Java and .Net's vision of Write-Once-Run-Anywhere
 - Device features are exposed through JavaScript API
 - The execution of the app relies on the browser engine of the OS it is installed in
- Gartner Says by 2016, More Than 50 Percent of Mobile Apps Deployed Will be Hybrid
 - <http://www.gartner.com/newsroom/id/2324917>
- Major tools:
 - Apache Cordova
 - PhoneGap: <http://phonegap.com>
 - Appcelerator Titanium
 - IBM MobileFirst



Web View



- All the smartphone operating systems support the use of WebViews in apps. *WebViews* are a small view component to handle web page content (HTML/CSS, etc.). A *WebView* can be a tiny part of the app screen, as shown in Figure 1, a whole page, or anything in between.
- **WebView Apps**
 - The entire app or at least most of it uses one big *WebView*
- **Some example apps**
 - Facebook, Evernote, Instagram, LinkedIn, Uber, Slack, Twitter, Gmail, the Amazon Appstore, and many others are or have been *WebView* apps
 - Quora (figure on the right)



<https://www.uxmatters.com/mt/archives/2018/08/mobile-apps-native-hybrid-and-webviews.php>

Services and Service Providers



- Mobile backend as a service (MBaaS), also known as "backend as a service" (BaaS) is a model for providing mobile app developers (also web apps) with a way to use backend services through Web APIs
 - These services usually include database, file storage, user access and management, push notifications, social networking services, emails, and others
 - Developers can focus on front end development and don't need to develop and maintain a backend server themselves
 - The global The Mobile Backend as a Service Market is expected to grow at USD ~\$87 Billion by 2023, at ~64% of CAGR between 2017 and 2023.
 - <https://www.marketwatch.com/press-release/mobile-backend-as-a-service-market-size-share-market-intelligence-company-profiles-and-trends-forecast-to-2023-2018-06-01>
- Top providers
 - Kinvey
 - Azure <https://docs.microsoft.com/en-us/azure/app-service-mobile/>
 - Sencha
 - IBM

Mobile Application Framework



- A "framework" is a fundamental structure for supporting or enclosing something else, especially a skeletal support used as the basis for something being constructed on.
- The main aim of frameworks is to increase productivity by reducing efforts which eventually saves lot of time for developers to resolve any other important issues in the app. These frameworks provides inbuilt tools for developers to work instantly on difficult and lengthy part of coding.
- Major frameworks for mobile web
 - Responsive UI Frameworks: these frameworks supports UI design on screens of various sizes
 - Bootstrap <http://getbootstrap.com>
 - Foundation <http://foundation.zurb.com>
 - Framework 7
 - <http://materializecss.com/>
 - <http://www.material-ui.com>
 - <http://cssframeworks.org>
 - HTML 5 mobile app: these frameworks mainly supports mobile devices, giving an app-like look and feel, and optimized for touch
 - jQuery Mobile <https://jquerymobile.com>
 - iUI <http://www.iui-js.org>
 - Other mobile frameworks that are not just UI
 - Angular
 - React
 - Other mobile (cross device) app frameworks that utilizes web technologies
 - Ionic <http://ionicframework.com>
 - Ext JS <https://www.sencha.com/products/extjs/>
 - Appcelerator <https://www.appcelerator.com/mobile-app-development-products/>
 - <http://www.gajotres.net/top-7-mobile-application-html5-frameworks/>
- More: <http://mobile-frameworks-comparison-chart.com/>

This subject will be covered in details in
IT 4213 module 10

Development Tools



- Testing and validation
 - <https://search.google.com/test/mobile-friendly>
 - <https://testmysite.thinkwithgoogle.com>
 - <https://ready.mobi>
 - <http://lab.maltewassermann.com/viewport-resizer/>
- Feature check
 - <http://mobilehtml5.org>
 - <http://caniuse.com>
- Live feature detection
 - <http://mydevice.io>
 - <http://detectmobilebrowsers.com>
 - <https://viewportsizer.com> and <https://viewportsizer.com/devices/>
 - <http://whatismyviewport.com>
 - Screen resolution: <http://www.screenresolution.org> or <http://www.whatismyscreenresolution.com>
- Feature detection programming mode
 - Modernizr: <http://modernizr.com>
- Responsive design test
 - <https://material.io/resizer/>
 - <http://www.responsinator.com>
 - <https://www.browsersync.io>
 - <http://mobiletest.me>
- Emulator
 - <https://developers.google.com/web/tools/chrome-devtools/device-mode/>
 - https://developer.mozilla.org/en-US/docs/Tools/Responsive_Design_Mode
 - <http://www.mobilexweb.com/emulators>
 - <http://lab.maltewassermann.com/viewport-resizer/>
- Design tools
 - <https://wireframe.cc>
 - <https://jdittrich.github.io/quickMockup/>
 - <https://pencil.evolus.vn>
- Generator
 - <https://www.duda.co/mobile-website>

Design Considerations and Principles



- Be aware of the difference of user experience on mobile vs desktop
- Web site or app? Consider their pros and cons and select the most appropriate one.
- Follow the general conventions and design principles.
 - Design for small (multi) screen; adapt to screen size and density
 - Design for touch friendly UI
- Reasonably and selectively apply design patterns and best practices
- Other best practices
 - Adopt a design language for consistency
 - Handle browser compatibility and feature detection
 - Design for mobile performance
 - Apply a development strategy like mobile first
 - Use a design tool to assist design and development

This subject will be covered in details
in IT 4213 module 4

Design for Small Screens



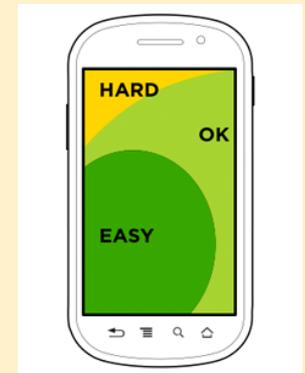
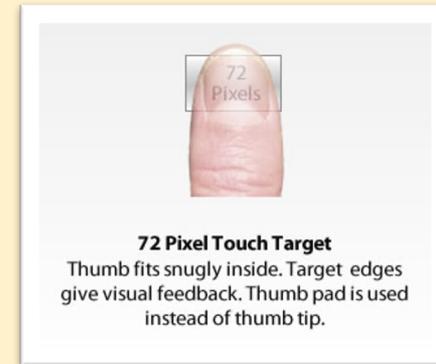
- The grand general principle of small screens is to keep the page short
 - Most mobile websites are viewed in portrait mode with narrow width. Stretching a normal multi column webpage will make the page extra long. Long stretching pages are more likely to get user lost and difficult to navigate.
- Major issues (see <https://www.viget.com/articles/do-responsive-sites-have-to-be-so-tall-on-mobile/>)
 - Scrolling Quickly Becomes Sluggish and Boring
 - Navigation Can be Unfriendly
 - Hindrance for the Footer Elements
- Some general design practices (details on following slides)
 - Content/feature prioritization: identify main content and give more screen area to the main content.
 - Extra on demand: more content with additional actions
 - Layout adjustment: change how content is arranged and grouped
 - <https://www.sitepoint.com/6-ways-to-improve-long-scroll-mobile-websites/>

IT 4213 covers more details on designing for small screens in module 4.

Design for Touch Friendly UI



- Understand the difference of touch oriented input. Some are weaknesses, some are advantages, and some are, just different.
- Major differences
 - Finger touches are generally less precise and less accurate
 - Direct input, generally lack of external input devices like mouse and keyboard
 - No hover, no right click, no scrolling wheel
 - No keyboard shortcut
 - Easy to use gestures, direct action, smoother movement
- Touch UI does offer opportunities to design better interaction methods and enable additional functionalities than traditional UI.
- Finger's advantages vs. mouse
 - Accuracy: less accurate on pointing/tapping, but more accurate in movement (moving, dragging, drawing, etc.)
 - Multi-point input
 - Flexibility: gestures are richer and more advanced
 - Direct action: easy to directly tapping on a target; horizontal scroll is easy
- Avoid touch weaknesses and maximize its advantages
 - Aware of hot touch area – positioning of common commands and actions
 - Set the right touch target size and style
 - Provide interaction feedback
 - Ease user input/typing
 - Take advantage of touch gestures (NOT covered in this class; gestures represent the future UI opportunities - <http://www.lukew.com/touch/>)



IT 4213 covers more details on designing for touch UI in module 4.

Design Patterns and Practices



- Design patterns and best practices are finer level design guides following the general principles.

IT 4213 covers more detailed design patterns and best practices for layout, navigation, data presentation, form, and media (module 5 to 9).

- General patterns and best practices resources
 - Google <https://www.thinkwithgoogle.com/marketing-resources/experience-design/principles-of-mobile-app-design-engage-users-and-drive-conversions/>
 - <http://bradfrost.github.io/this-is-responsive/patterns.html>
 - <https://designmodo.com/content-patterns-mobile/>
 - Android patters: <https://unitid.nl/androidpatterns>
 - <https://www.interaction-design.org/literature/topics/mobile-ux-design>
 - <https://pttrns.com>
 - <http://www.mobile-patterns.com>
 - Books
 - Mobile UI Design Patterns <https://www.uxpin.com/studio/ebooks/mobile-design-patterns/>
 - Mobile Design Pattern Gallery <https://www.amazon.com/dp/1449363636> - one sample chapter <http://www.uxbooth.com/articles/mobile-design-patterns/>
 - Designing mobile interfaces: <http://4ourth.com/wiki/Designing%20Mobile%20Interfaces>

Data, Stats, and Reports



- Key organizational reports
 - Cisco Visual Networking Index <http://www.cisco.com/c/en/us/solutions/service-provider/visual-networking-index-vni/index.html>
 - KPCB Internet Trends <http://www.kpcb.com/internet-trends>
 - <https://deviceatlas.com/blog>
 - <https://www.idc.com/getdoc.jsp?containerId=prUS45636719>
 - <https://www.fcc.gov/reports-research>
 - <https://www.ctia.org/the-wireless-industry/infographics-library>
- General stats
 - <http://www.netmarketshare.com>
 - <http://gs.statcounter.com>
 - <http://marketshare.hitslink.com>
 - Browser: https://en.wikipedia.org/wiki/Usage_share_of_web_browsers
- Other Reports
 - <http://mobiforge.com>
 - <http://www.visionmobile.com/reports/>
 - <http://www.pewinternet.org>
 - <http://www.appcelerator.com/enterprise/resource-center/research/>
 - <https://www.slideshare.net/comScoremarcom/presentations>
 - Android devices: <http://developer.android.com/about/dashboards/index.html>

Key Readings



- This lecture notes <https://www.edocr.com/v/k52p5vj4/jgzheng/Mobile-Web-Overview>
- Mobile web differences
 - <https://www.paradoxlabs.com/blog/mobile-vs-desktop-10-key-differences/>
- Mobile development options
 - Mobile web app: <https://www.scnsoft.com/blog/mobile-web-app-types>
 - Developing Mobile Web Applications: When, Why, and How <https://www.toptal.com/android/developing-mobile-web-apps-when-why-and-how>
- ONE web (responsive), adaptive, and separate mobile site (Google describes the three techniques)
 - <https://www.thinkwithgoogle.com/marketing-resources/experience-design/building-websites-multi-screen-consumer/>
- Additional readings and resources
 - Some history: http://www.phonearena.com/news/Evolution-of-mobile-web-browsing_id9059
 - <http://bradfrost.com/blog/post/the-many-faces-of-adaptive-design/>
 - https://en.wikipedia.org/wiki/Mobile_Web
 - <https://www.wired.com/2013/05/the-two-flavors-of-a-one-web-approach-responsive-vs-adaptive/>
 - Why Separate Mobile & Desktop Web Pages? <http://www.lukew.com/ff/entry.asp?1390>
 - <https://developers.google.com/search/mobile-sites/>
 - One web: <http://adactio.com/journal/1716/>
 - <https://developer.mozilla.org/en-US/docs/Web/Guide/Mobile>
 - Mobile site design for performance: <https://queue.acm.org/detail.cfm?id=2441756>

Good Resources



- Community resources
 - <https://mobiforge.com>
 - <http://www.w3.org/Mobile/>
 - <http://www.w3.org/standards/webdesign/mobilweb>
 - <https://www.linkedin.com/groups?gid=3002798>
- Online learning resources
 - <http://www.quirksmode.org/mobile/>
 - <https://alistapart.com>
 - <https://www.udacity.com/course/mobile-web-development--cs256>
- Vendor resources
 - <https://developers.google.com/search/mobile-sites/>
 - <https://developers.google.com/web/fundamentals/>
 - <http://msdn.microsoft.com/en-us/library/aa286514.aspx>
 - <https://developer.mozilla.org/en-US/docs/Web/Guide/Mobile>
 - <http://www.kinvey.com/native-web-hybrid-developers-ebook>
 - <http://www.ibm.com/mobilefirst/us/en/>
 - <http://www.mobiletuxedo.com>
- Lecture notes collection
 - <https://www.edocr.com/user/jgzheng/collection/mobileandweblecturenotes>
- Influencers
 - <https://ethanmarcotte.com/>
 - Luke Wroblewski <http://www.lukew.com>
 - Maximiliano Firtman <http://www.mobilexweb.com>
 - Steven Hooper <http://www.4ourth.com>
 - <http://www.nngroup.com>
- Books
 - <http://www.amazon.com/dp/1449334970>
 - <https://www.amazon.com/Responsive-Web-Design-HTML5-CSS3/dp/1784398934/>
 - <http://www.amazon.com/dp/1593274874>
 - <http://www.amazon.com/dp/1449311415>
 - <http://www.amazon.com/dp/1449363636>
- Media news and columns
 - <http://mashable.com/category/mobile-web/>
 - <https://www.uxmatters.com/columns/mobile-matters/>
 - <http://www.smashingmagazine.com>
 - <http://www.quirksmode.org/blog/archives/mobile/>

Mobile Site SEO



- <https://developers.google.com/search/mobile-sites/mobile-seo/>
- <https://webmasters.googleblog.com/2018/03/rolling-out-mobile-first-indexing.html>
- <https://www.informaticsync.com/blog/2016/google-boosts-mobile-friendly-algorithm-further-penalizing-non-mobile-sites>
- A real story
<https://cheekymonkeymedia.ca/blog/mobile-friendly-does-it-really-matter>

Design Tools



Going Beyond



- The world is entering a multi-screen multi-device usage environment. The use of mobile devices is not isolated from other type of devices, but rather an important part of it.
 - Watch, phone, tablet, desktop, TV, super size displays
- https://support.google.com/adsense/answer/6051803?hl=en&ref_topic=6051812

Mobile Web Brief History



pocketweb

1994

The first mobile browser for a PDA was PocketWeb for the Apple Newton created at TecO in 1994, followed by the first commercial product NetHopper released in August 1996.



1997

Wireless Application Protocol 1.0

Networking protocol created to allow mobile devices to access the internet for services such as e-mail, stock prices, SMS, etc. It is open, non proprietary. Uses WML (wireless markup language) for text and graphics. Adopted by many vendors such as Sony, Motorola, etc. Allows for an explosion in mobile specific websites and browsers.



2002

WAP 2.0

Released to address 1.0 shortcomings. Uses WML, XHTML, and WCSS to give users a better experience

2005

.**mobi** domain is created for mobile specific websites



2007

first iPhone is released. Native app market explodes.

4G

2010

Wireless companies begin to advertise 4G wireless (10x3G) speeds en masse touting streaming TV, streaming movies, faster web access, etc. as customer benefits. Native app market grows by leaps and bounds as the iPhone and Android systems become household names. Opera Mini, SkyFire, are popular mobile browsers



2013

Major companies try to leverage the power of the mobile phone by making mobile payment solutions common.

HTML5 web development pushes to the forefront allowing for a better mobile web experience but WAP 2.0 still dominates.

HTML5 mobile frameworks such as jQuery mobile become popular

2G

3G

4G



- <https://uxmag.com/articles/framework-for-designing-for-multiple-devices>