Mobile web development

MICC / University of Florence
Daniele Pezzatini

What Is a Mobile Device?

Portable
A mobile device has to be portable, meaning that we can carry it without any special considerations.

Personal
A mobile device is absolutely personal. My mobile is mine; it’s not property of the family, nor is it managed by the company who manufactured it

Companion
Your mobile device can be with you anytime. You may forget to take lots of things with you from your home in the morning, but you won’t forget your wallet, your keys, and your mobile device.

Easy usage
A mobile device needs to be easy and quick to use. I don’t want to wait two minutes for an application to start; I don’t want to sit down. If I’m walking downtown, I want to be able to find out when the next train will be departing without having to stop.

Connected device
A mobile device should be able to connect to the Internet when you need it to.
We can differentiate between *fully connected devices* that can connect any time in a couple of seconds and *limited connected devices* that usually can connect to the network but sometimes cannot.

REF: “Programming the Mobile Web” by Maximiliano Firtman

Mobile Device Characteristics

We can identify many different characteristics among mobile devices

- Physical Dimension
- Screen resolution
- Screen dot density
- Manufacturer
- Operating System
- Connectivity
- Input method (touch / keyboard)
- Sensors and cameras

Resolution

How many pixels (width and height) are available on a given device? This was the only portability problem for many years in the area of mobile development.

Portability refers to the ability of a mobile application to be used on multiple devices with different hardware, software, and platforms.

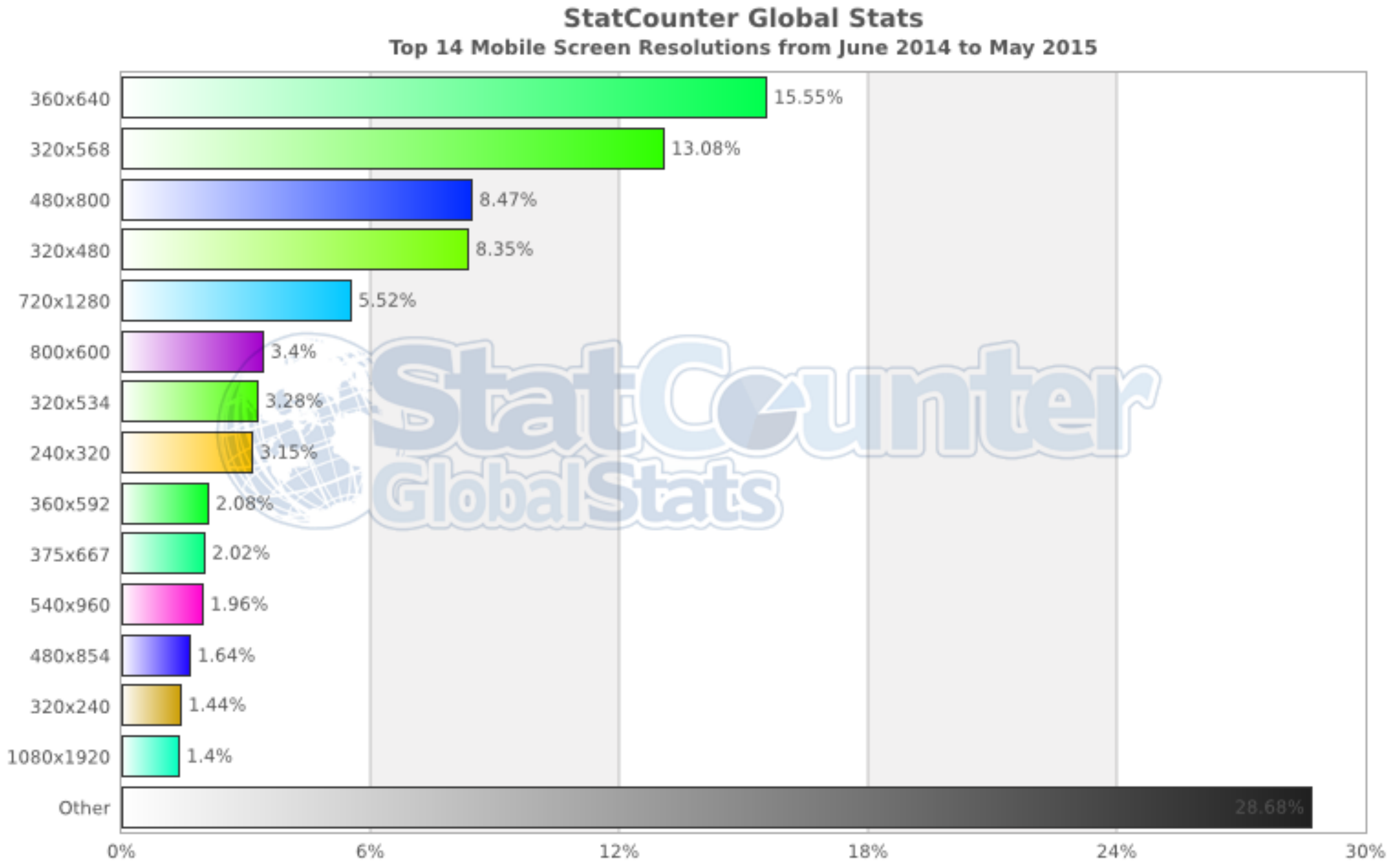
There are **no mobile device standards regarding screen resolution:**

- Typical for smartphones: 240×480, 320×480, 360×480, 480×800, 640×960 or 640×1136 pixels

Touch devices typically have a higher resolution than devices with a keyboard because no space needs to be reserved for the keypad.

Ex. BlackBerry Classic has a 720 x 720 pixels screen, while iPhone 6 display has a resolution of 1334x750

Resolution


StatCounter Global Stats
Top 14 Mobile Screen Resolutions from June 2014 to May 2015
360x640
15.55%
320x568
13.08%
480x800
8.47%
320x480
8.35%
720x1280
5.52%
800x600
3.4%
320x534
3.28%
240x320
3.15%
360x592
2.08%
375x667
2.02%
540x960
1.96%
480x854
1.64%
320x240
1.44%
1080x1920
1.4%
Other
28.68%
0%
6%
12%
18%
24%
30%
StatCounter GlobalStats

Physical dimensions and PPI

One feature as important as the resolution is the physical dimensions of the screen (in inches or centimeters, diagonally or measured as width/height), or **the relation between this measure and the resolution**, which is known as the PPI (pixels per inch) or DPI (dots per inch).

Example: iPhone 4S (retina display)

Display size: 2.91" × 1.94" (7.4cm × 4.93cm)

Resolution: 960 x 640

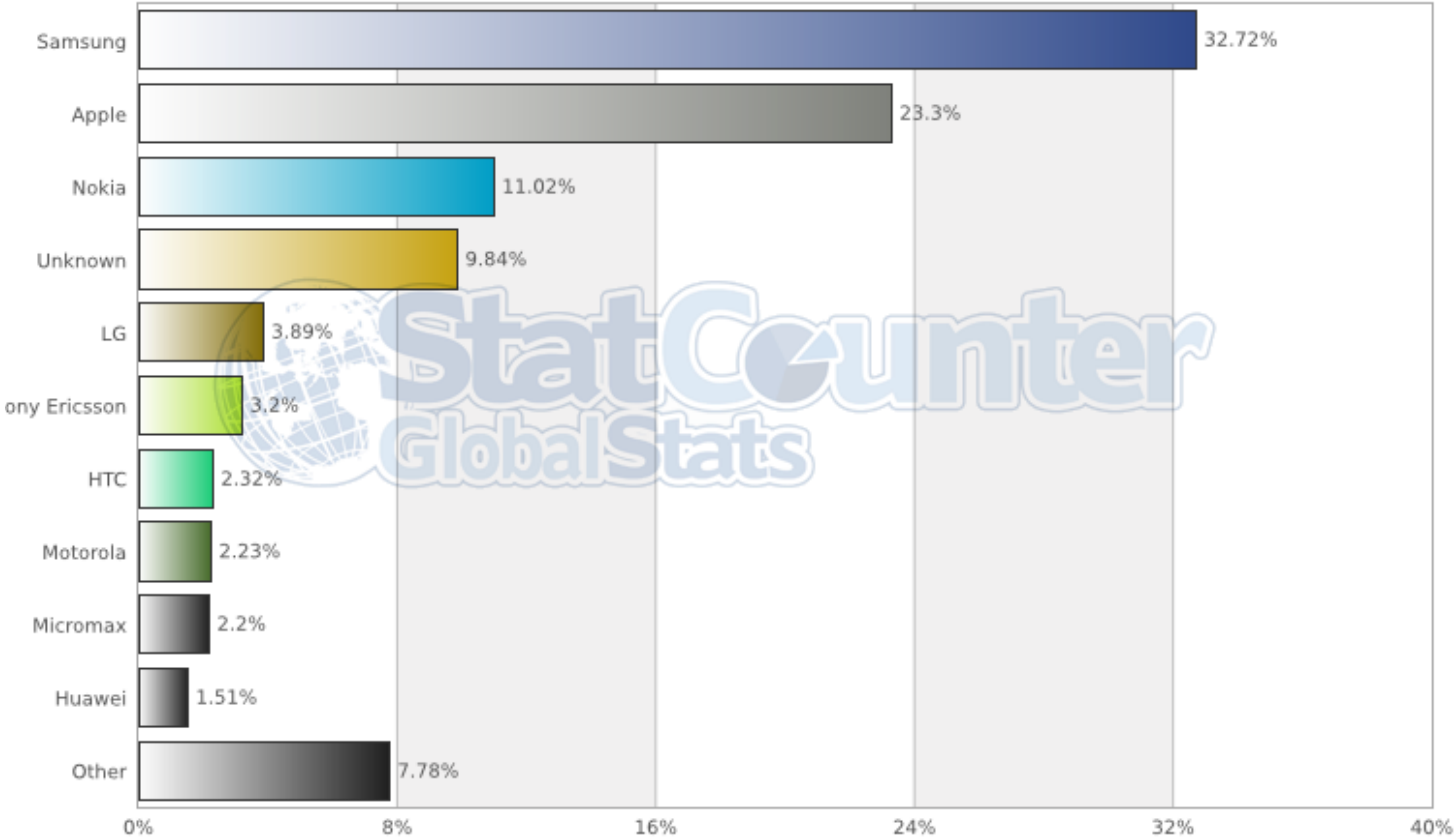
PPI = 329

The human retina has a limit of 300 ppi at a distance of about 30cm, so this device with 960×640 in landscape mode has more pixels per inch that the ones we can really see.

Manufacturer

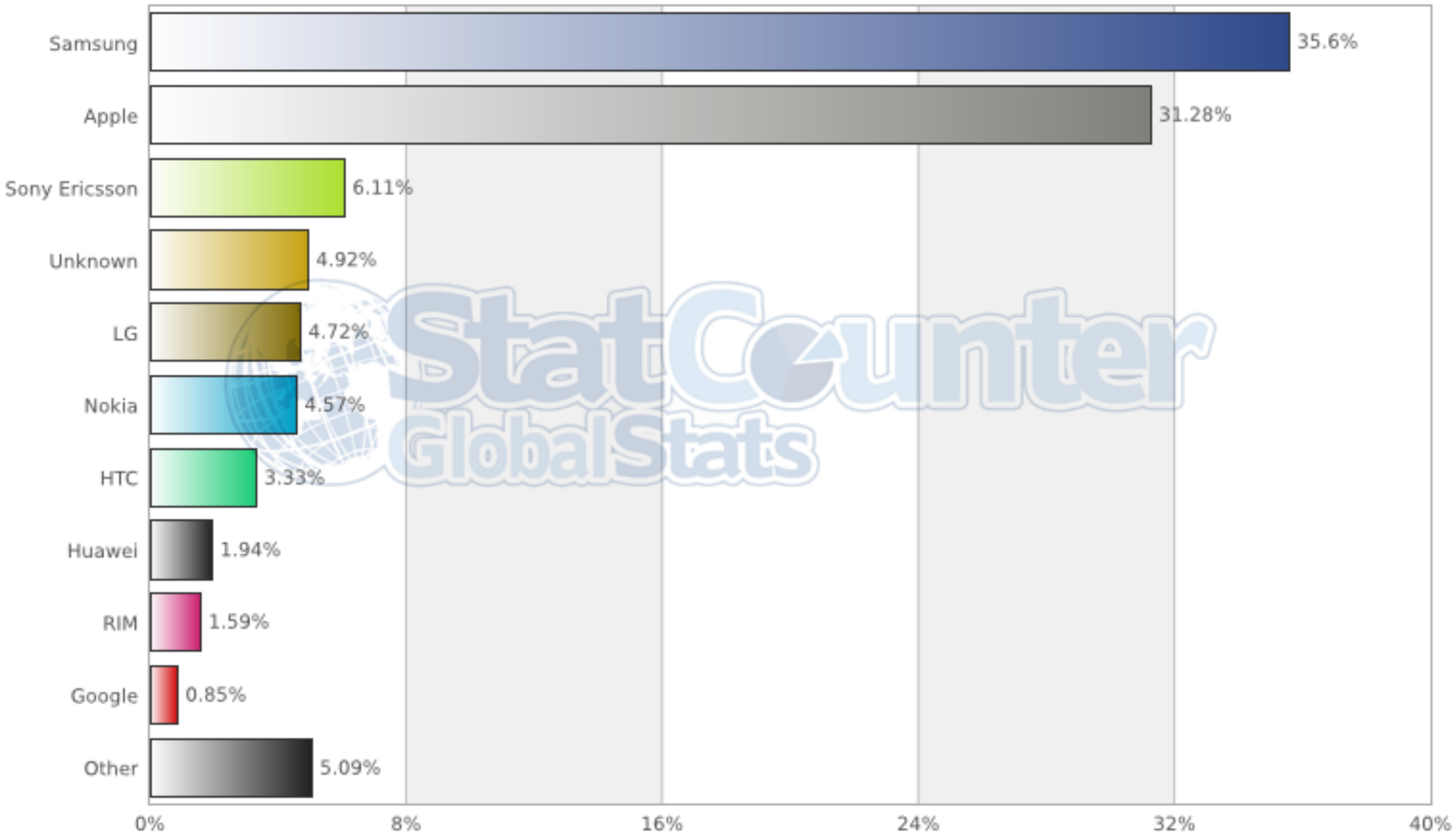
Manufacturer /platform	Developer site URL
Apple	http://developer.apple.com/iphone
Nokia	http://forum.nokia.com
Symbian Foundation	http://developer.symbian.org
Palm webOS	http://developer.palm.com
BlackBerry	http://www.blackberry.com/developers
Sony Ericsson	http://developer.sonyericsson.com
Windows Mobile	http://msdn.microsoft.com/windowsmobile
Motorola	http://developer.motorola.com
Opera Mobile/Mini	http://dev.opera.com
LG	http://developer.lgmobile.com
Samsung	http://innovator.samsungmobile.com
Android	http://developer.android.com
HTC	http://developer.htc.com
Bada (from Samsung)	http://developer.bada.com

Statistics - Mobile Vendor (Worldwide)



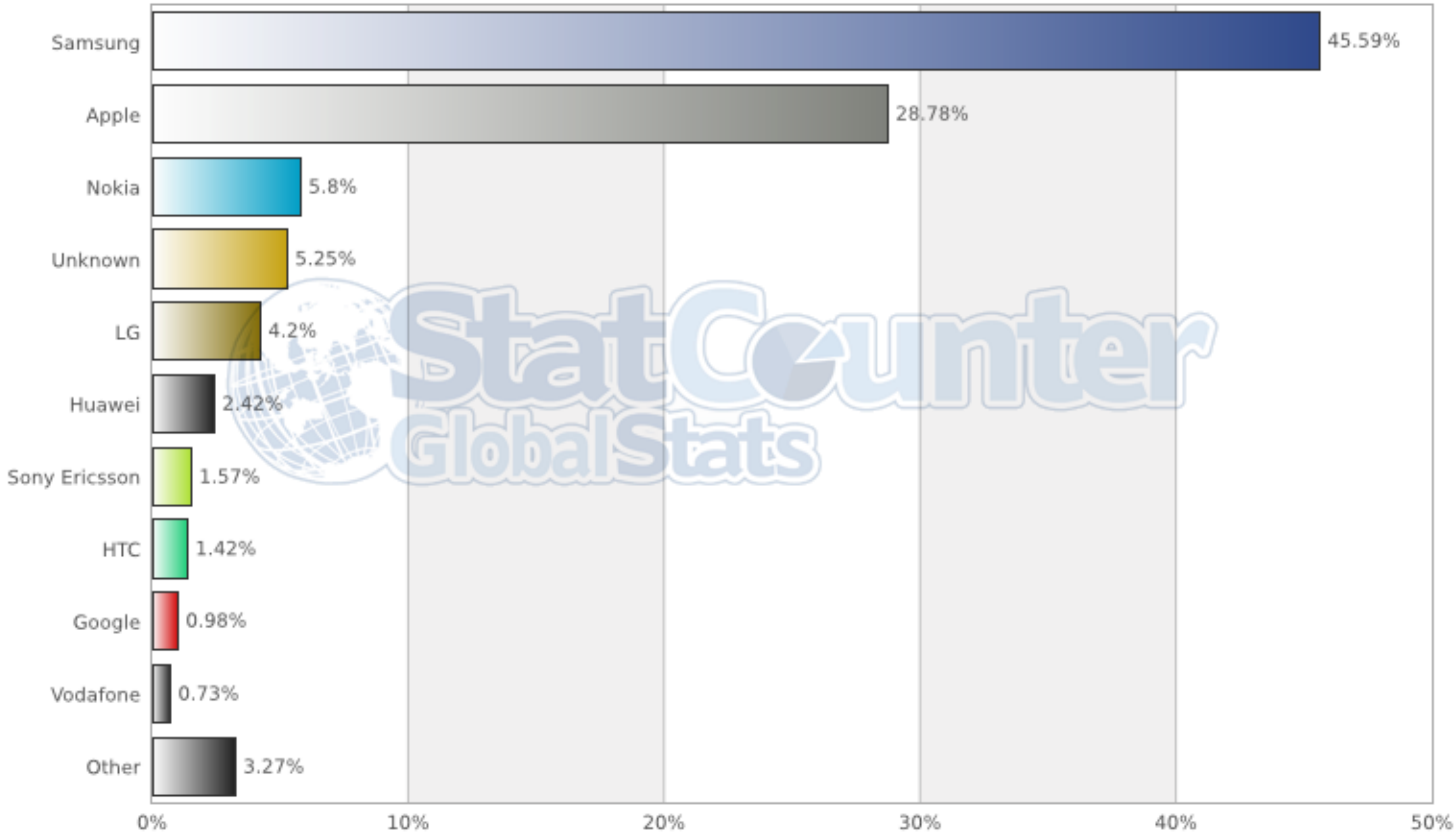
from June 2014 to May 2015

Statistics - Mobile Vendor (Europe)



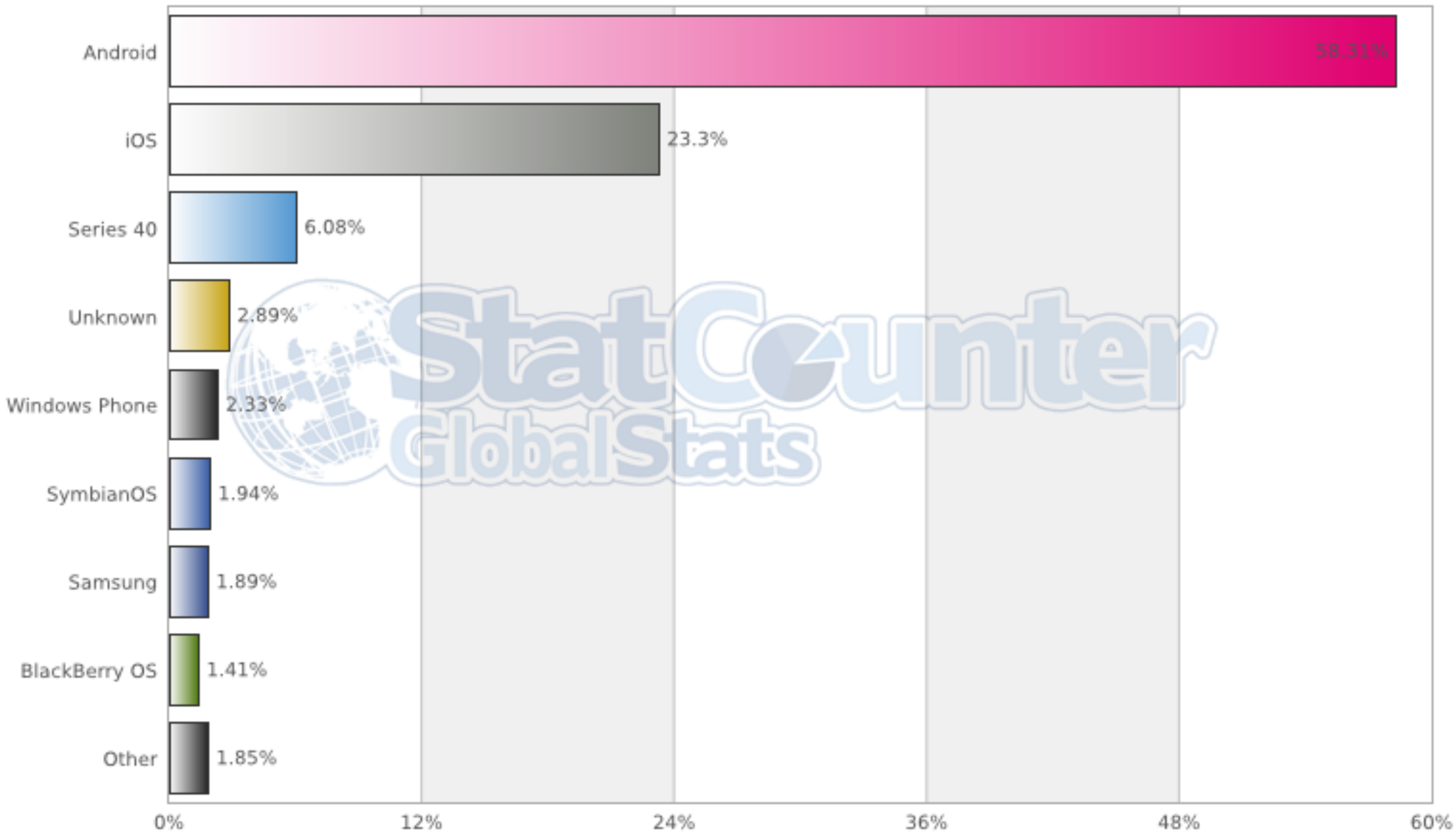
from June 2014 to May 2015

Statistics - Mobile Vendor (Italy)



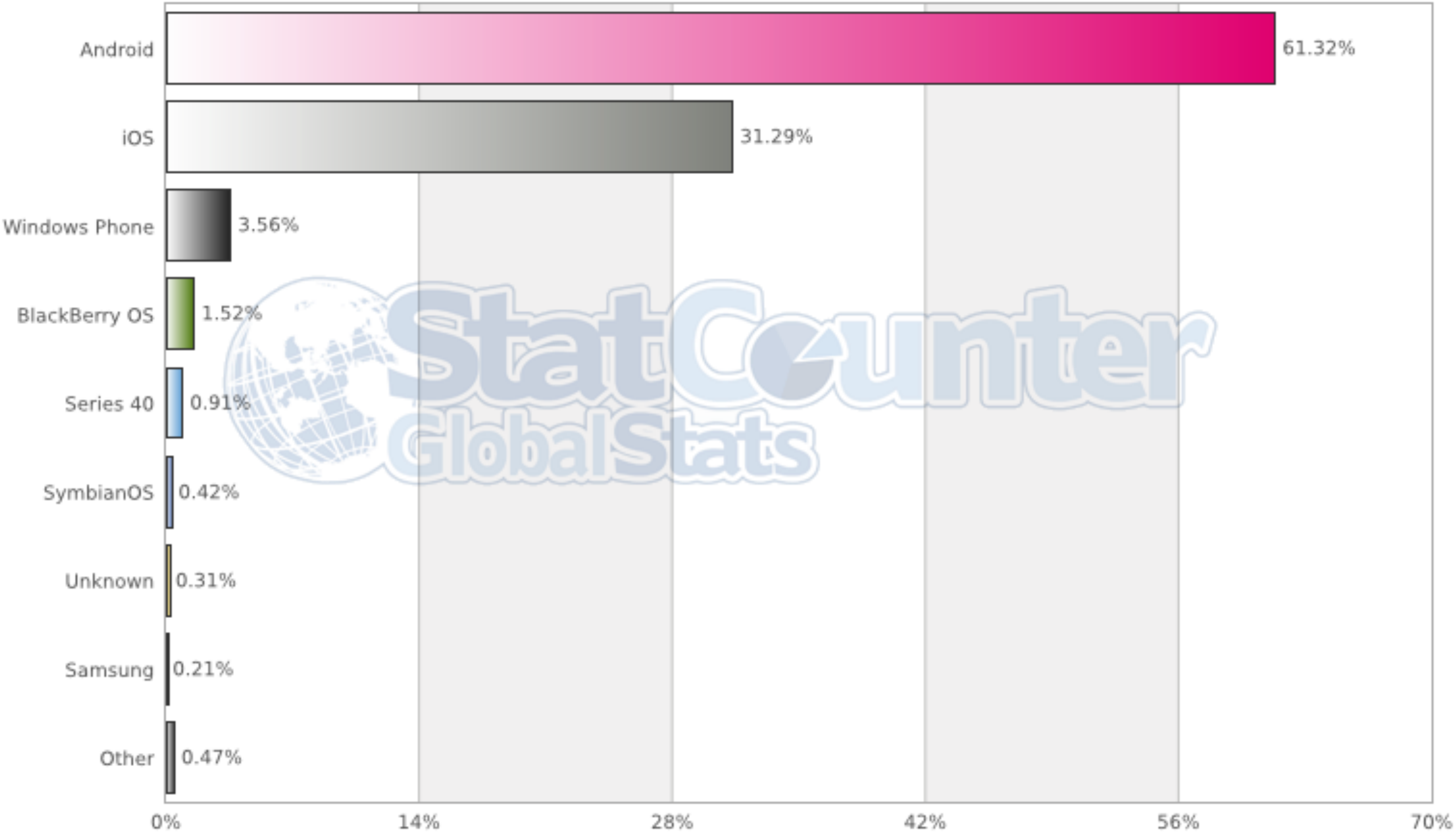
from June 2014 to May 2015

Statistics - Mobile OS (Worldwide)



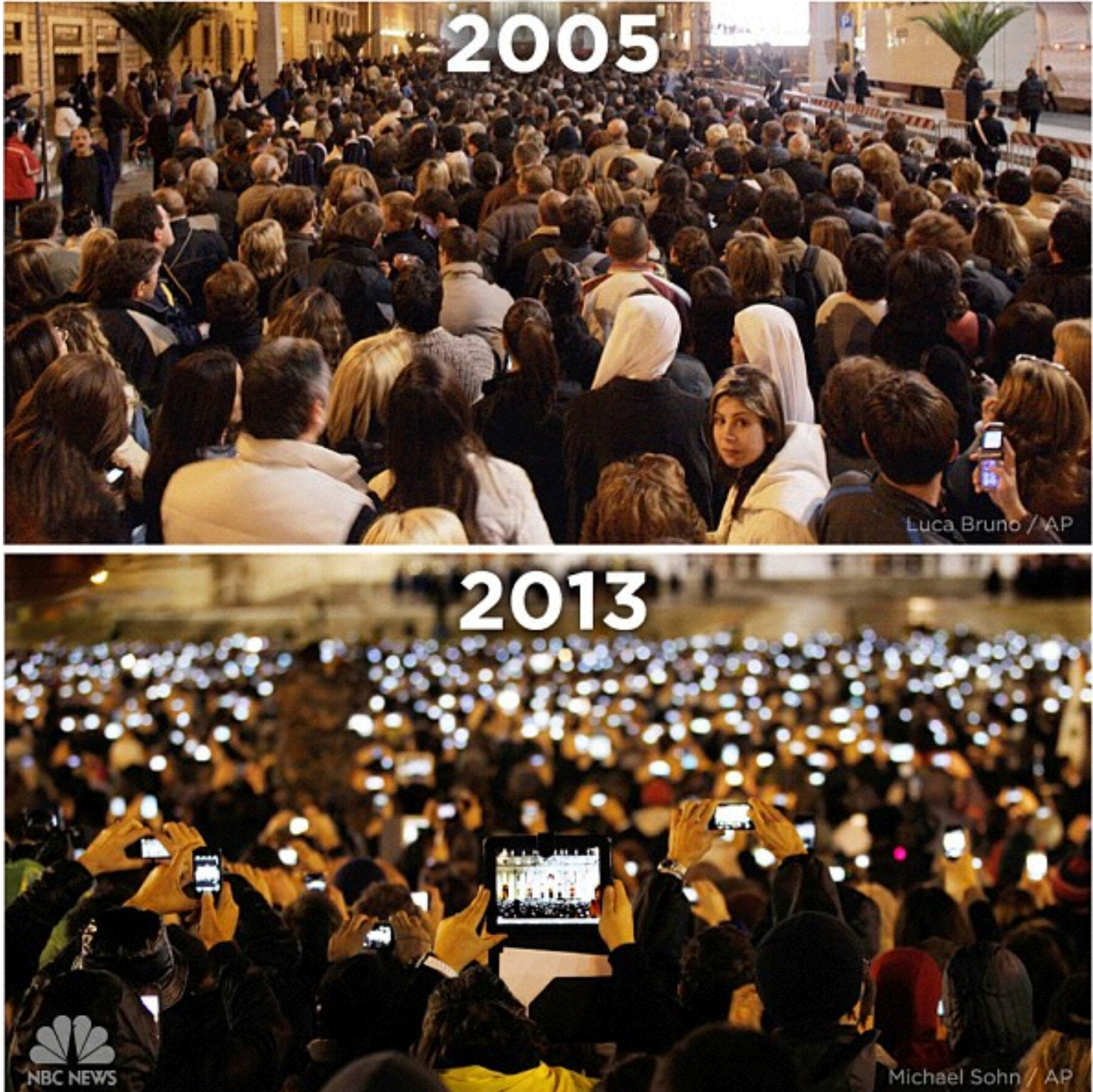
from June 2014 to May 2015

Statistics - Mobile OS (Europe)



from June 2014 to May 2015

Diffusion



2005

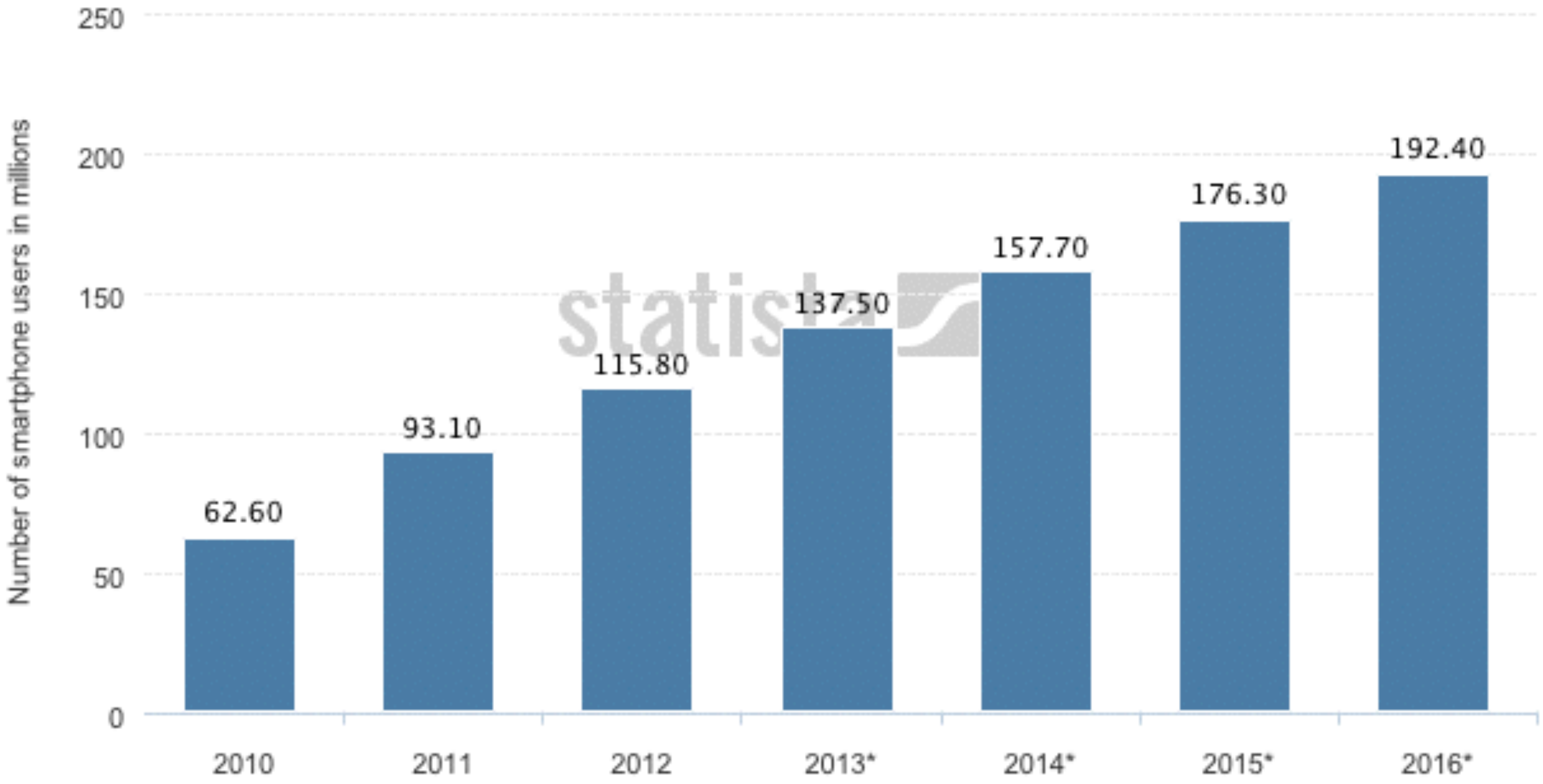
Luca Bruno / AP

2013

NBC NEWS

Michael Sohn / AP

Smartphone users in USA



Today: USA pop. 314 millions, about **43%** own a smartphone, **62%** between aged 25-34

Mobile developing

A **mobile website** is technically the same as a regular website except that it's size is adjusted to the smaller screen and resolutions. It has an adaptive layout. Programming languages are HTML(5)/CSS/JS.

A **web app** is like a regular mobile website but it behaves and is used like a native app. The user interface looks like a native app but technologies used are those of the web. Programming languages are HTML(5)/CSS/JS.

A **native app** is created for a specific platform and uses the required technologies such as an specific SDK or development language. Programming language change based on the platform (e.g. Objective-C for iOS, Java for Android)

A **hybrid app** is a web app or an app developed in a common language that is compiled into several native apps. Additional native features can be added to the web app which is then distributed as a native app.

Native vs Web apps

Web application and **mobile websites** are accessed on the Web via the device's browser

Native application and **hybrid app** are built specifically for a given platform (Android or iOS, for example) and are installed on the device much like a desktop application. These are generally made available to consumers via a platform-specific app marketplace (e.g. Apple's App Store for the iPhone and iPad).

Mobile web and native apps offer different benefits and serve different audiences. You need to look at:

- What experience your app needs to deliver?
- What you are trying to achieve?
- What is your business model?
- What is your budget?
- Who is your target?

Interesting reading: "Native vs mobile apps" by Peter-Paul Koch
http://www.netmagazine.com/opinions/native-vs-mobile-apps

Web apps

Pros:

- A single codebase which can be accessed by any browser-enabled mobile device.
- Uses web technologies (HTML/CSS/Javascript), which are arguably easier to learn than native languages like Objective-C or Java.
- Performance issues are becoming less of an issue as mobile browsers become faster and their Javascript engines keep improving
- No approval process needed, and updates to the app can happen instantaneously
- No revenue sharing with an app store

Cons:

- Using web technologies means interpreted code (as opposed to compiled code for native apps), which some people believe means web apps will always be slower than native apps.
- Don’t have full access to all the methods exposed by the device’s operating system, meaning you are limited to the APIs made available by the browser. As it stands now in Mobile Safari, this means no camera, video capture, microphone, user contacts, file uploading or push/local notifications.
- Can’t be found on the app store. If you’re lucky enough to be a featured app in Apple’s store, for example, it is a huge marketing boost.

Native apps

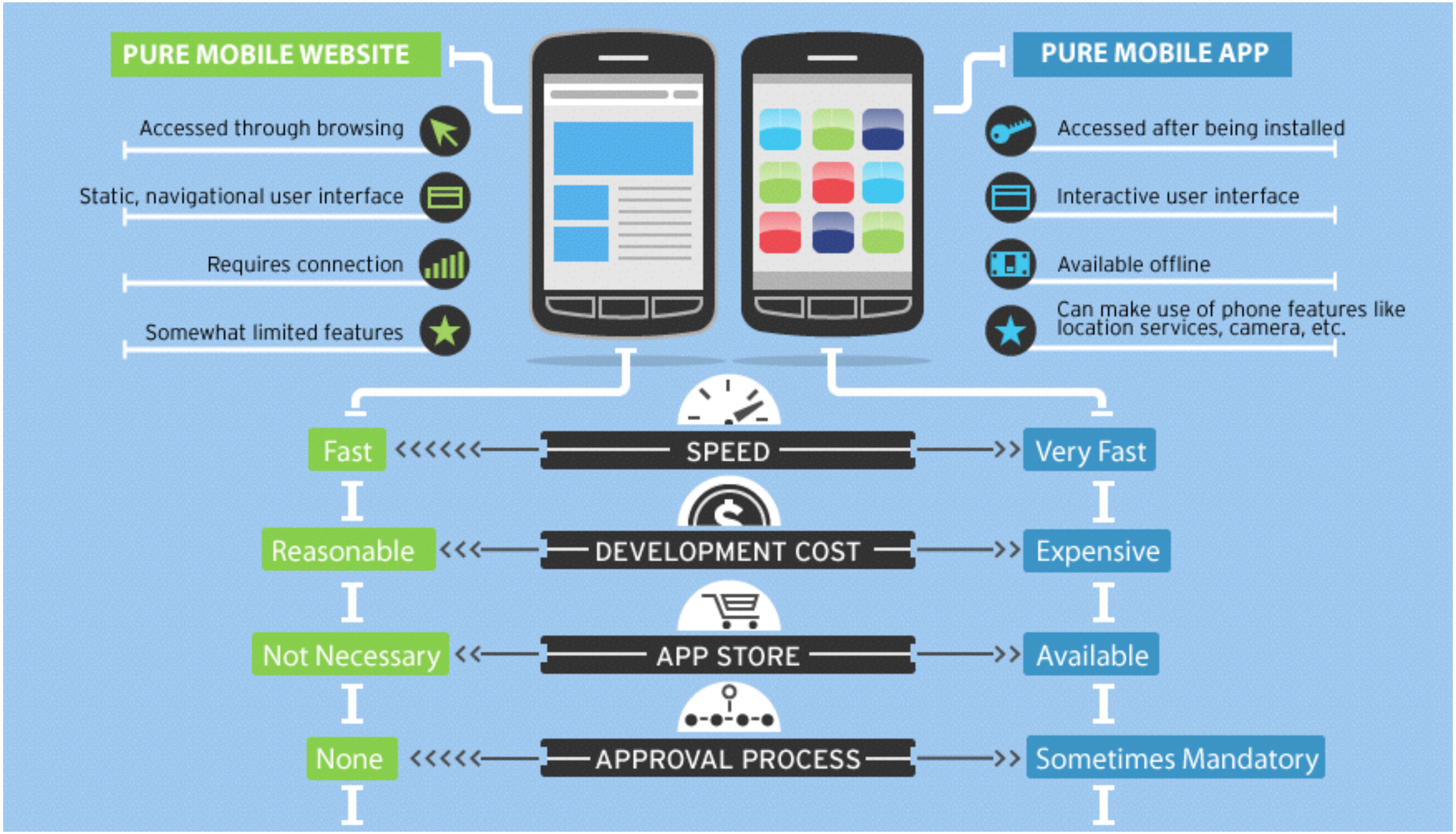
Pros:

- Better performance (at least for now), smoother animations, transitions, and faster load times. The performance difference between native and web apps is far more pronounced on slower devices (e.g. iPhone 3G running iOS4)
- Can store more data offline
- Can be featured and searched for in the app store
- Full access to the device's hardware and OS features
- Implicit installation of an app to the device's home screen. On iOS devices you can add any web app to your home screen, but it's a manual process
- The App Store handles purchase transactions on your behalf

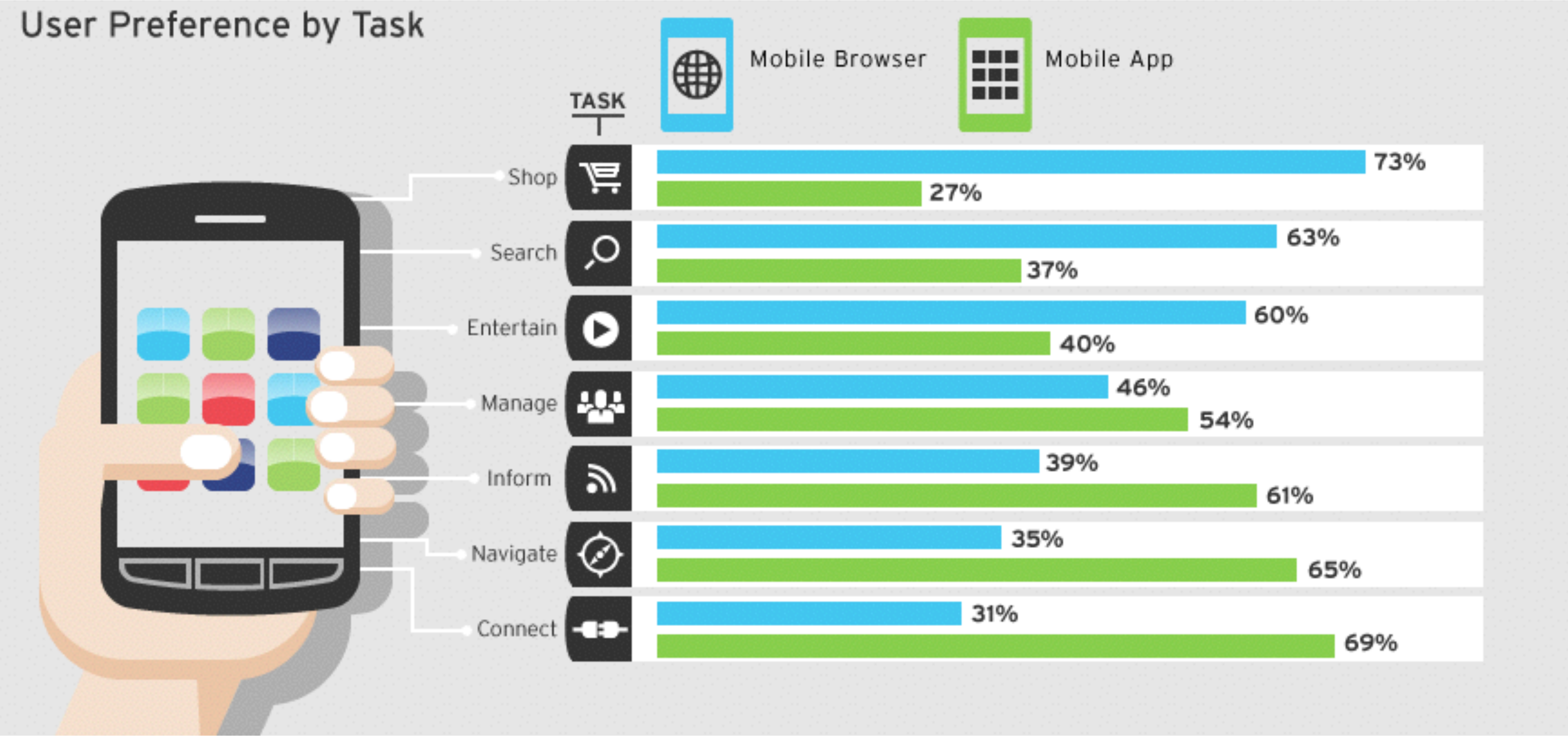
Cons:

- Typically more expensive to build, even for a single platform. Build costs increase significantly for each new platform.
- Because the codebase needs to be re-worked for each OS, the time to build an app for multiple devices can also be quite involved.
- Your app must be accessed through the device's app store, which has two important considerations: your app must go through an approval process, which can be lengthy and arbitrary, and if your app generates revenue you must share a percentage with the store (30% for Apple's App Store, including in-app purchases). App updates must go through a new approval process each time.

Native vs Web apps


PURE MOBILE WEBSITE
Accessed through browsing
Static, navigational user interface
Requires connection
Somewhat limited features
PURE MOBILE APP
Accessed after being installed
Interactive user interface
Available offline
Can make use of phone features like location services, camera, etc.
Fast
SPEED
Very Fast
Reasonable
DEVELOPMENT COST
Expensive
Not Necessary
APP STORE
Available
None
APPROVAL PROCESS
Sometimes Mandatory

Native vs Web apps



Hybrid apps

Pros:

- Allows you to capitalize on the single codebase offered by web technologies yet still market your app in each of the major mobile app stores.
- Gives you APIs to access some, if not all, of the features locked out of the browser, such as camera, compass, contacts.
- Purchases are managed by the App Store.

Cons:

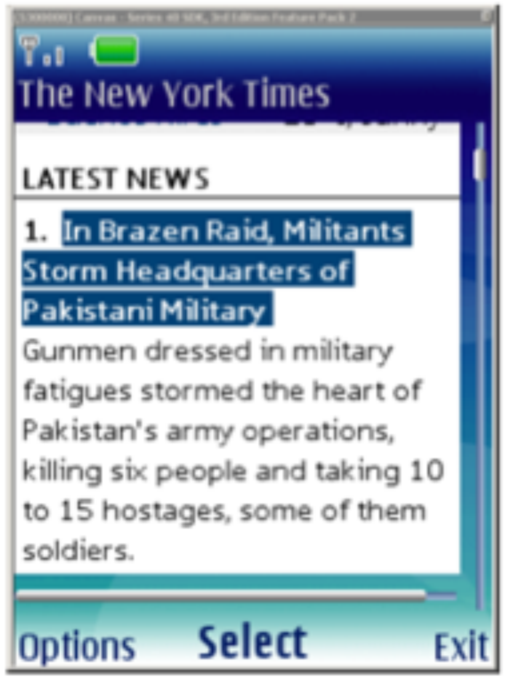
- Still subject to the store's approval process and revenue sharing. No instant updating.
- The app's performance is still dependent on the device's browser capabilities.

mobile websites & web apps

Mobile browser types

A mobile website can be navigated using different techniques. Every mobile browser uses one or many of these modes of navigation.

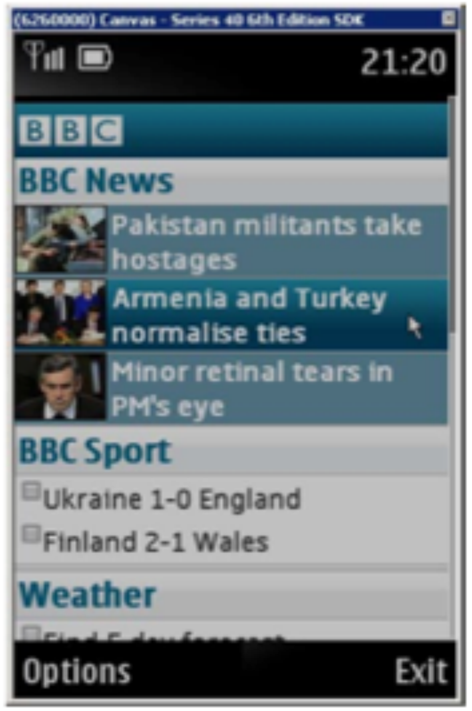
- **Focus navigation** is the most frequent mechanism used for browsing websites on low and mid-end devices. A border or a background color is used to show the user where the focus is. In general it is used in non-touch devices, so the user uses the cursor keypad to navigate between links and scroll the website.
- **Cursor navigation** emulates a mouse cursor over the screen that can be moved using the arrow keys. A mouse click is emulated with the enter key. For a better experience, many browsers jump the cursor to a nearby focusable object to reduce the distance the user has to move the pointer to use a link or a button.
- **Touch navigation** allow users to navigate using a finger or a stylus. The differences in design can be huge; precision is much lower if fingers are used. Touch devices allow the user to use detectable gestures to easily perform some actions. **Multitouch** devices increment the number of gestures that can be detected.


The New York Times

LATEST NEWS

1. In Brazen Raid, Militants Storm Headquarters of Pakistani Military

Gunmen dressed in military fatigues stormed the heart of Pakistan's army operations, killing six people and taking 10 to 15 hostages, some of them soldiers.


(6260000) Canvas - Series 40 6th Edition SDK

21:20

BBC

BBC News

Pakistan militants take hostages

Armenia and Turkey normalise ties

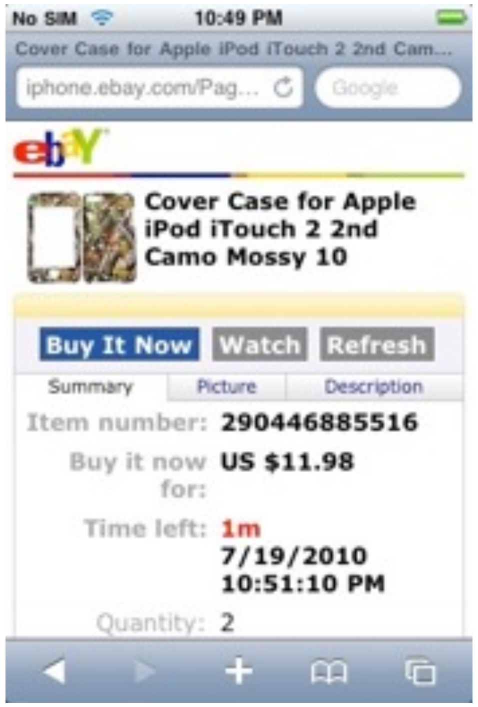
Minor retinal tears in PM's eye

BBC Sport

□Ukraine 1-0 England

□Finland 2-1 Wales

Weather


No SIM 10:49 PM

Cover Case for Apple iPod iTouch 2 2nd Cam...

iphone.ebay.com/Pag...

Cover Case for Apple iPod iTouch 2 2nd Camo Mossy 10

Item number: 290446885516

Buy it now for: US $11.98

Time left: 1m 7/19/2010 10:51:10 PM

Quantity: 2

Website navigation

When creating your mobile web concept, before you do any coding you should define what will be in the navigation tree for the user. To do that, you need to understand what services and information will be available for the mobile user.

Pareto principle application: 80% of your desktop users are interested in 20% of the website content.
Therefore, you need to research the 20% you should be focusing on.

- Define the use cases (for example, find a product price, find a store near you, call us, or perform a search).
- Order the use cases by the most frequent for a mobile user. Use your best guess, statistical information, and usability tests to keep this order updated.
- Try to make every use case successful in no more than three clicks or at a page depth of no more than three.
- Always offer a link to the desktop website.
- Determine whether locating the user is useful for your services.
- Reduce the form pages for user input to the minimum.
- Avoid startup or welcome screens.

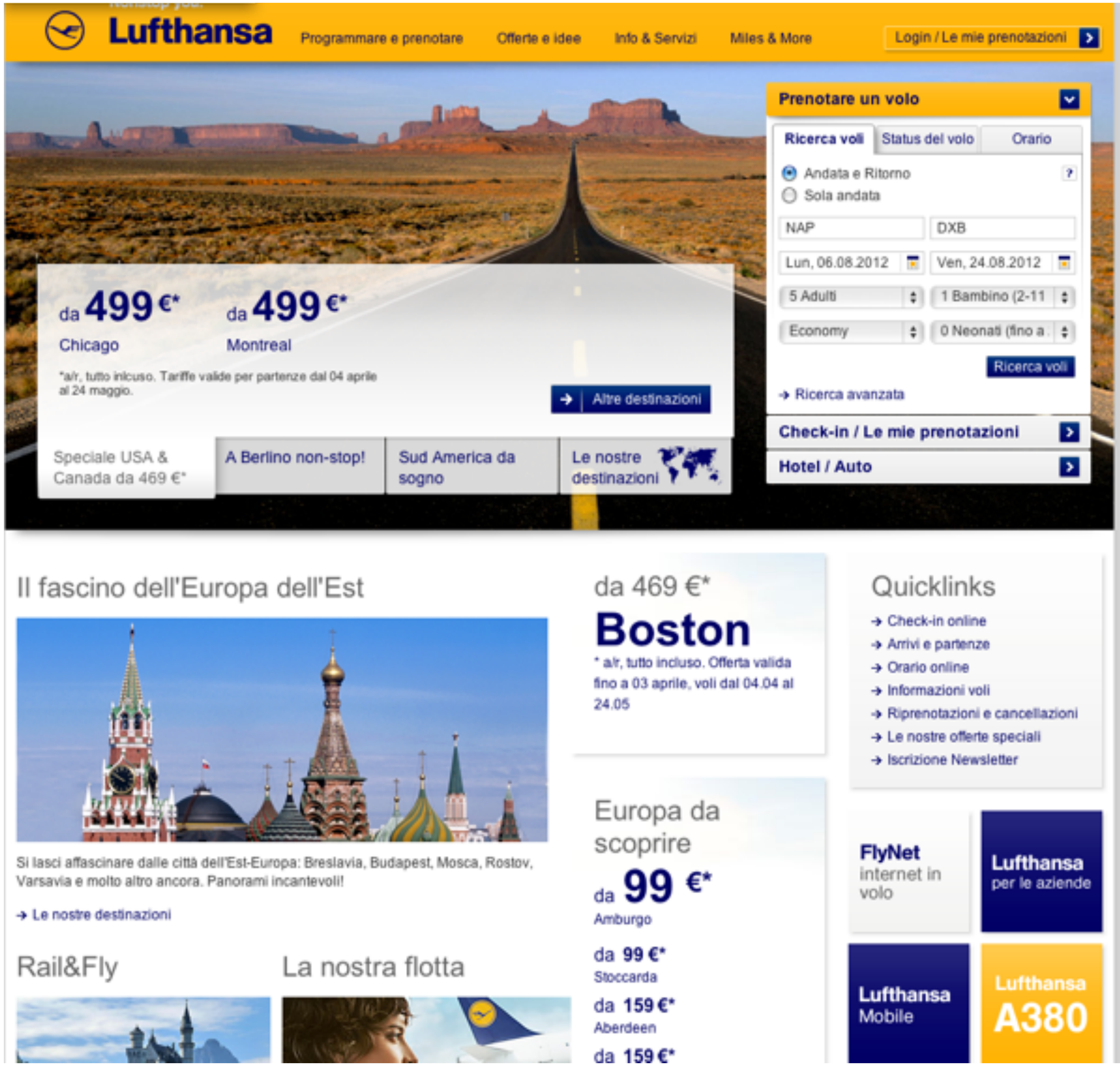
Context

A mobile user has a different context than a desktop user. You should think about and define your users' possible contexts:

- Where is the user?
- Why is the user accessing your mobile website?
- What is the user looking for?
- What can you offer from a mobile perspective to help solve the user's problem?
- Where will the user be when accessing your website? Walking on the street, using public transportation, at the office, traveling as a tourist?

The context will tell you many things about your navigation, use cases, and the usability needs for your mobile site.

Context



● Andata e Ritorno
○ Sola andata

NAP | DXB
Lun, 06.08.2012 | Ven, 24.08.2012
5 Adulti | 1 Bambino (2-11
Economy | 0 Neonati (fino a

da 499 €* Chicago
da 499 €* Montreal

*a/r, tutto incluso. Tariffe valide per partenze dal 04 aprile al 24 maggio.

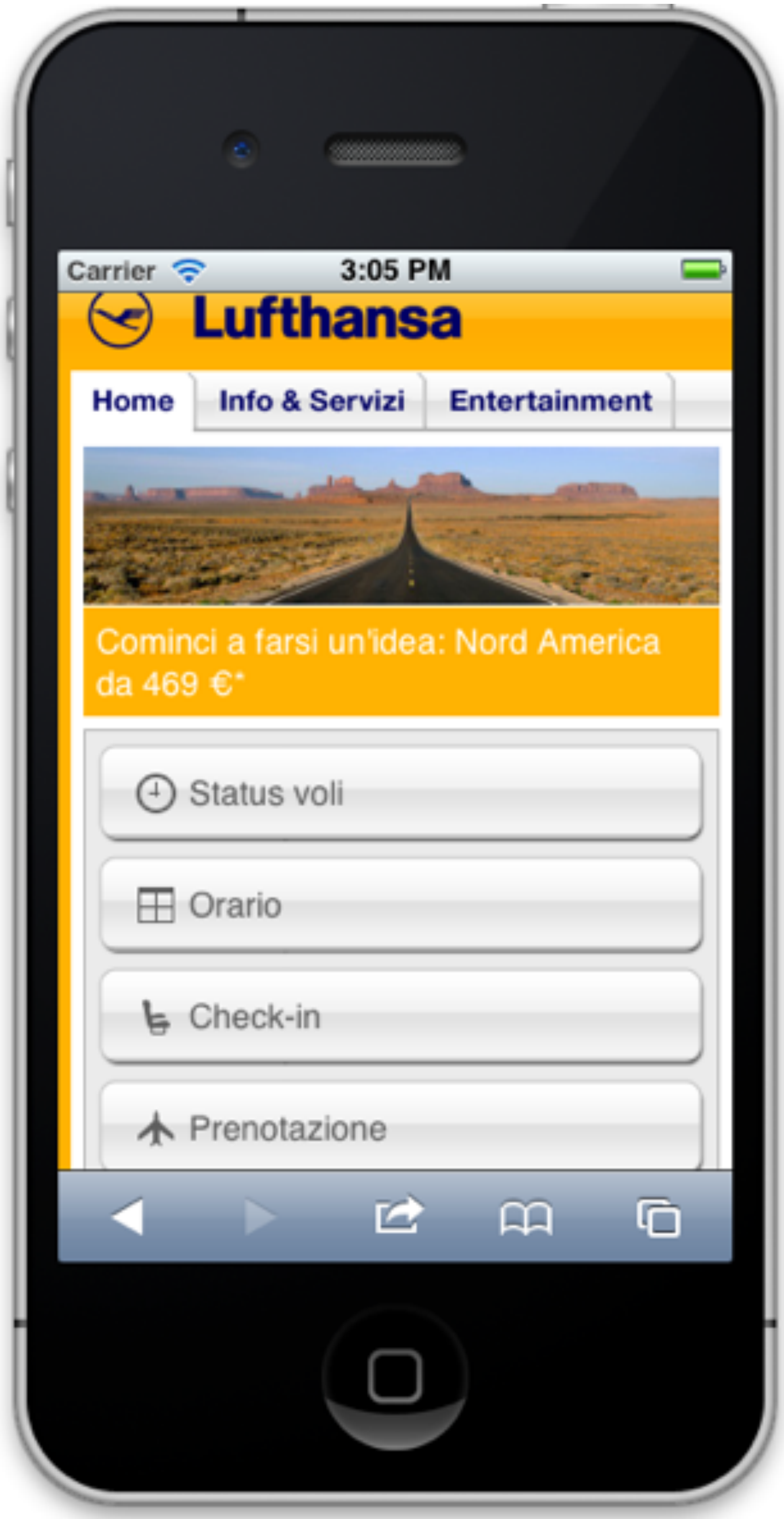
Il fascino dell'Europa dell'Est

Si lasci affascinare dalle città dell'Est-Europa: Breslavia, Budapest, Mosca, Rostov, Varsavia e molto altro ancora. Panorami incantevoli!

Le nostre destinazioni

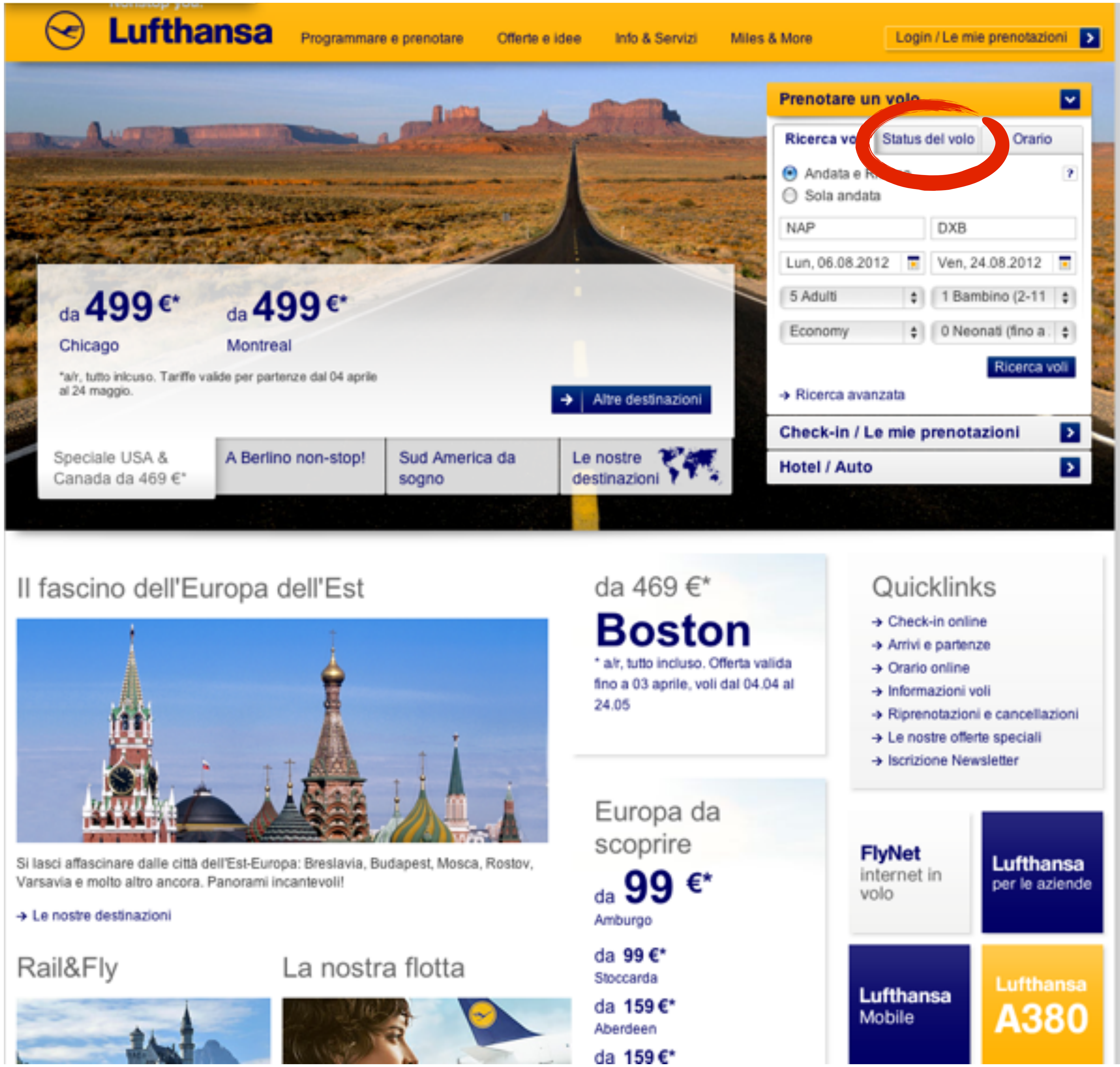
da 469 €*
Boston
* a/r, tutto incluso. Offerta valida fino a 03 aprile, voli dal 04.04 al 24.05

Europa da scoprire
da 99 €* Amburgo
da 99 €* Stoccarda
da 159 €* Aberdeen
da 159 €*



Cominci a farsi un'idea: Nord America da 469 €*

Context



*a/r, tutto incluso. Tariffe valide per partenze dal 04 aprile al 24 maggio.

Il fascino dell'Europa dell'Est

Si lasci affascinare dalle città dell'Est-Europa: Breslavia, Budapest, Mosca, Rostov, Varsavia e molto altro ancora. Panorami incantevoli!

→ Le nostre destinazioni

da 469 €*

Boston

* a/r, tutto incluso. Offerta valida fino a 03 aprile, voli dal 04.04 al 24.05

Europa da scoprire

da 99 €* Amburgo

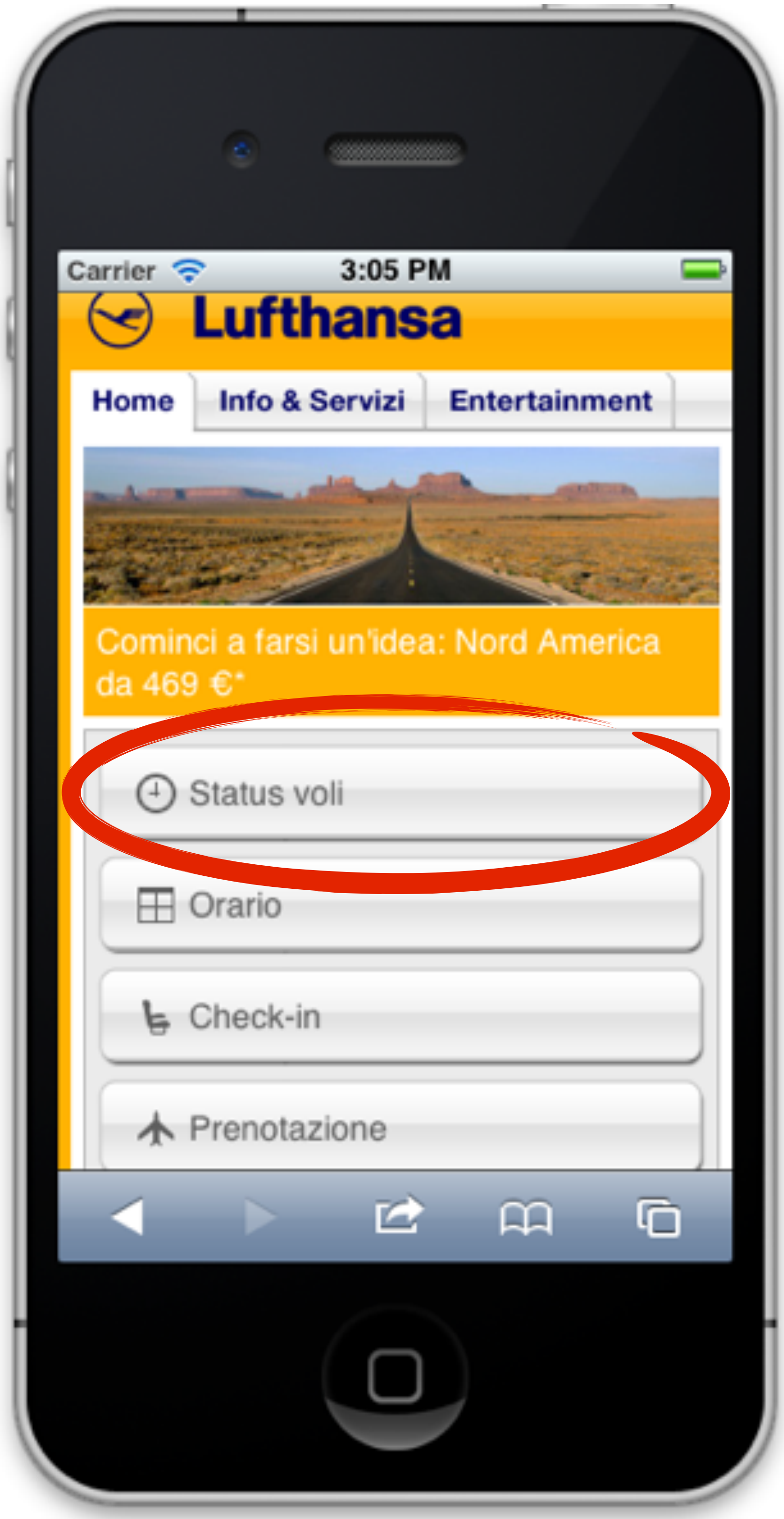
da 99 €* Stoccarda

da 159 €* Aberdeen

da 159 €*

Quicklinks

- → Check-in online
- → Arrivi e partenze
- → Orario online
- → Informazioni voli
- → Riprenotazioni e cancellazioni
- → Le nostre offerte speciali
- → Iscrizione Newsletter



Comincì a farsi un'idea: Nord America da 469 €*

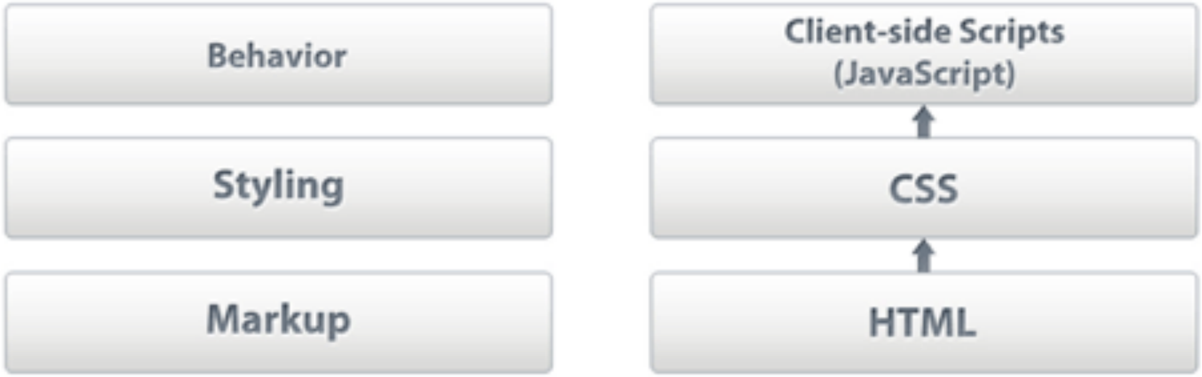
Progressive Enhancement

Progressive enhancement is a simple but very powerful technique used in web design that defines layers of compatibility that allow any user to access the basic content, services and functionality of a web and providing an enhanced experience for browser with better support of standards.

The concept subverts the typical web design strategy, known as “graceful degradation,” where designers develop for the latest technologies and browsers and their designs automatically work with the lesser functions available on older browsers. This technique is not useful for mobile browsers because, as we will see, there are serious compatibility issues in the mobile world. If we develop a website for the latest device (for example, the iPhone), it may not automatically work on other, less advanced devices.

Progressive enhancement has the following core principles:

- Basic content and functionality are accessible to all browsers.
- Semantic markup contains all content.
- Enhanced layout is provided by externally linked CSS.
- Enhanced behavior is provided by externally linked JavaScript.



http://www.alistapart.com/articles/understandingprogressiveenhancement

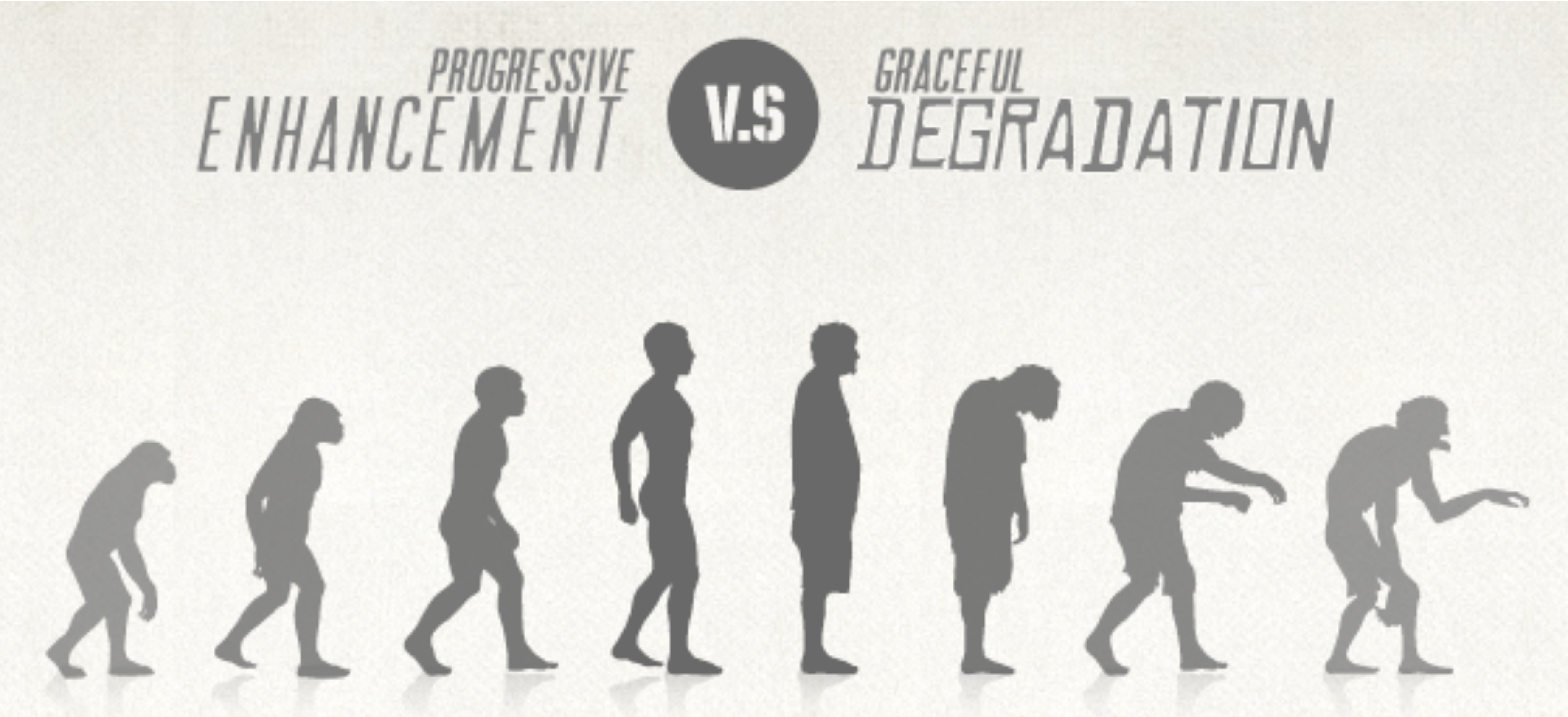
Progressive Enhancement vs Graceful degradation

Progressive enhancement

Starting with a baseline of usable functionality, then increasing the richness of the user experience step by step by testing for support for enhancements before applying them.

Graceful degradation

Providing an alternative version of your functionality or making the user aware of shortcomings of a product as a safety measure to ensure that the product is usable.



Progressive Enhancement

```
<p id="printthis">Thank you for your order. Please
print this page for your records.</p>

<script type="text/javascript">

(function(){

  if(document.getElementById){

    var pt = document.getElementById('printthis');

    if(pt && typeof window.print === 'function'){

      var but = document.createElement('input');

      but.setAttribute('type','button');

      but.setAttribute('value','Print this now');

      but.onclick = function(){

        window.print();

      };

      pt.appendChild(but);

    }

  }

})();
```

Graceful degradation

```
<p id="printthis">

   <a href="javascript:window.print()">Print
this page</a>

</p>

<noscript>

  <p class="scriptwarning">

    Print a copy of your confirmation.

    Select the "Print" icon in your browser,

    or select "Print" from the "File" menu.

  </p>

</noscript>
```

http://www.w3.org/wiki/Graceful_degredation_versus_progressive_enhancement

Responsive web design

The responsive web design solution addresses the ever-changing landscape of devices, browsers, screen sizes and orientations by creating flexible, fluid and adaptive websites.

Instead of responding to today's needs for a desktop web version adapted to the most common screen resolution, along with a particular mobile version (often specific to a single mobile device), the idea is to approach the issue the other way around: use flexible and fluid layouts that adapt to almost any screen.

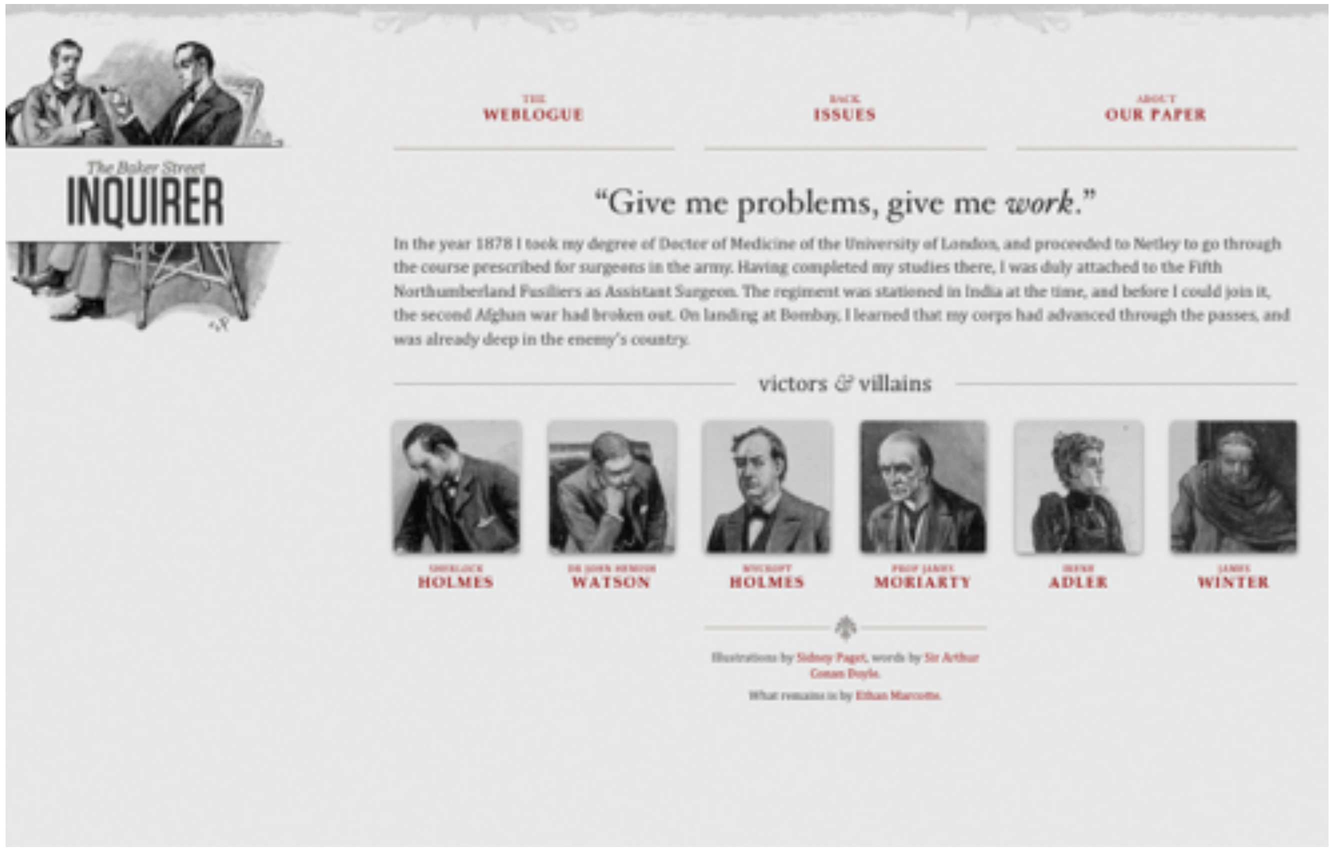
Three key technical features are at the heart of responsive web design:

- Media queries and media query listeners
- A flexible grid-based layout that uses relative sizing
- Flexible images and media, through dynamic resizing or CSS

Responsive web design

Responsive web design tutorial

http://www.alistapart.com/articles/responsive-web-design/

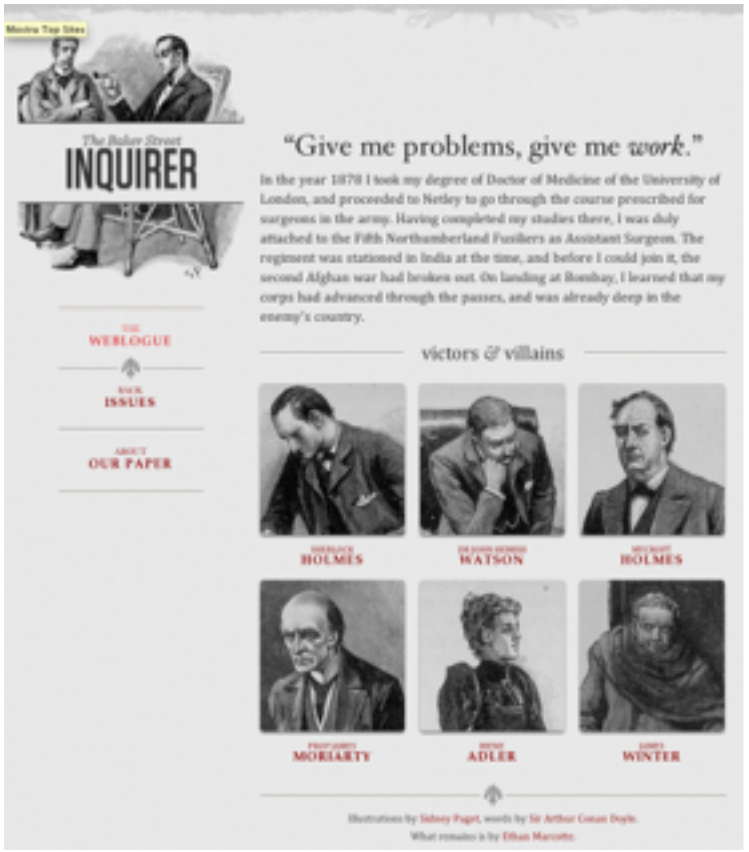


"Give me problems, give me *work*."

In the year 1878 I took my degree of Doctor of Medicine of the University of London, and proceeded to Netley to go through the course prescribed for surgeons in the army. Having completed my studies there, I was duly attached to the Fifth Northumberland Fusiliers as Assistant Surgeon. The regiment was stationed in India at the time, and before I could join it, the second Afghan war had broken out. On landing at Bombay, I learned that my corps had advanced through the passes, and was already deep in the enemy's country.

Illustrations by Sidney Paget, words by Sir Arthur Conan Doyle.

What remains is by Ethan Marcotte.

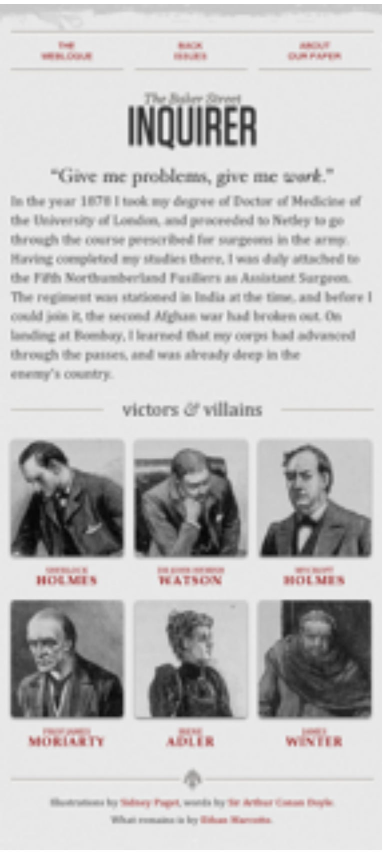


"Give me problems, give me *work*."

In the year 1878 I took my degree of Doctor of Medicine of the University of London, and proceeded to Netley to go through the course prescribed for surgeons in the army. Having completed my studies there, I was duly attached to the Fifth Northumberland Fusiliers as Assistant Surgeon. The regiment was stationed in India at the time, and before I could join it, the second Afghan war had broken out. On landing at Bombay, I learned that my corps had advanced through the passes, and was already deep in the enemy's country.

Illustrations by Sidney Paget, words by Sir Arthur Conan Doyle.

What remains is by Ethan Marcotte.



"Give me problems, give me *work*."

In the year 1878 I took my degree of Doctor of Medicine of the University of London, and proceeded to Netley to go through the course prescribed for surgeons in the army. Having completed my studies there, I was duly attached to the Fifth Northumberland Fusiliers as Assistant Surgeon. The regiment was stationed in India at the time, and before I could join it, the second Afghan war had broken out. On landing at Bombay, I learned that my corps had advanced through the passes, and was already deep in the enemy's country.

Illustrations by Sidney Paget, words by Sir Arthur Conan Doyle.

What remains is by Ethan Marcotte.

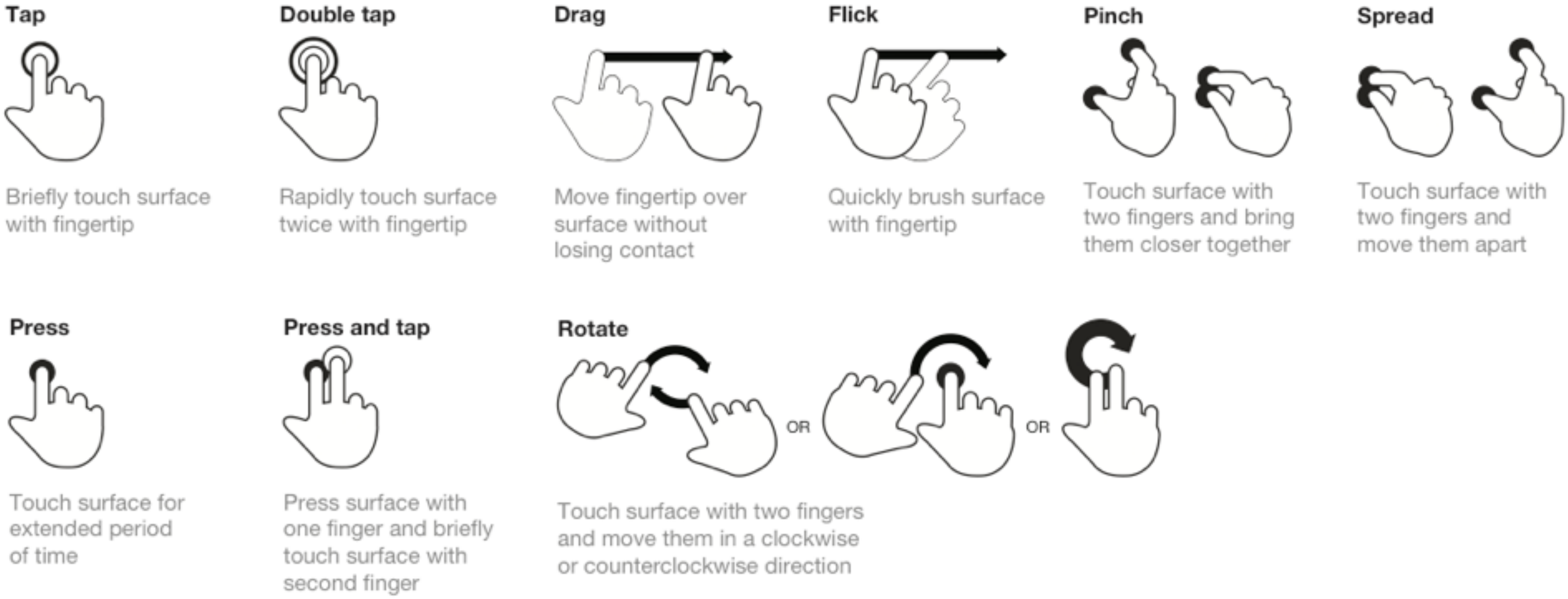
screen width > 1300 px → screen width < 640 px

Touch design patterns

Touch devices have unique features in terms of design and usability. With the same amount of effort, the user can access every pixel on the screen; this encourages a different way of thinking about a design.

Another difference is that the user will use her finger for touch selection (unless it is a stylus-based device). A finger is big compared to a mouse pointer, and the hit zone should reflect this.

Devices with multi-touch support allow user to perform gestures in order to simplify his tasks



Tap
Briefly touch surface with fingertip

Double tap
Rapidly touch surface twice with fingertip

Drag
Move fingertip over surface without losing contact

Flick
Quickly brush surface with fingertip

Pinch
Touch surface with two fingers and bring them closer together

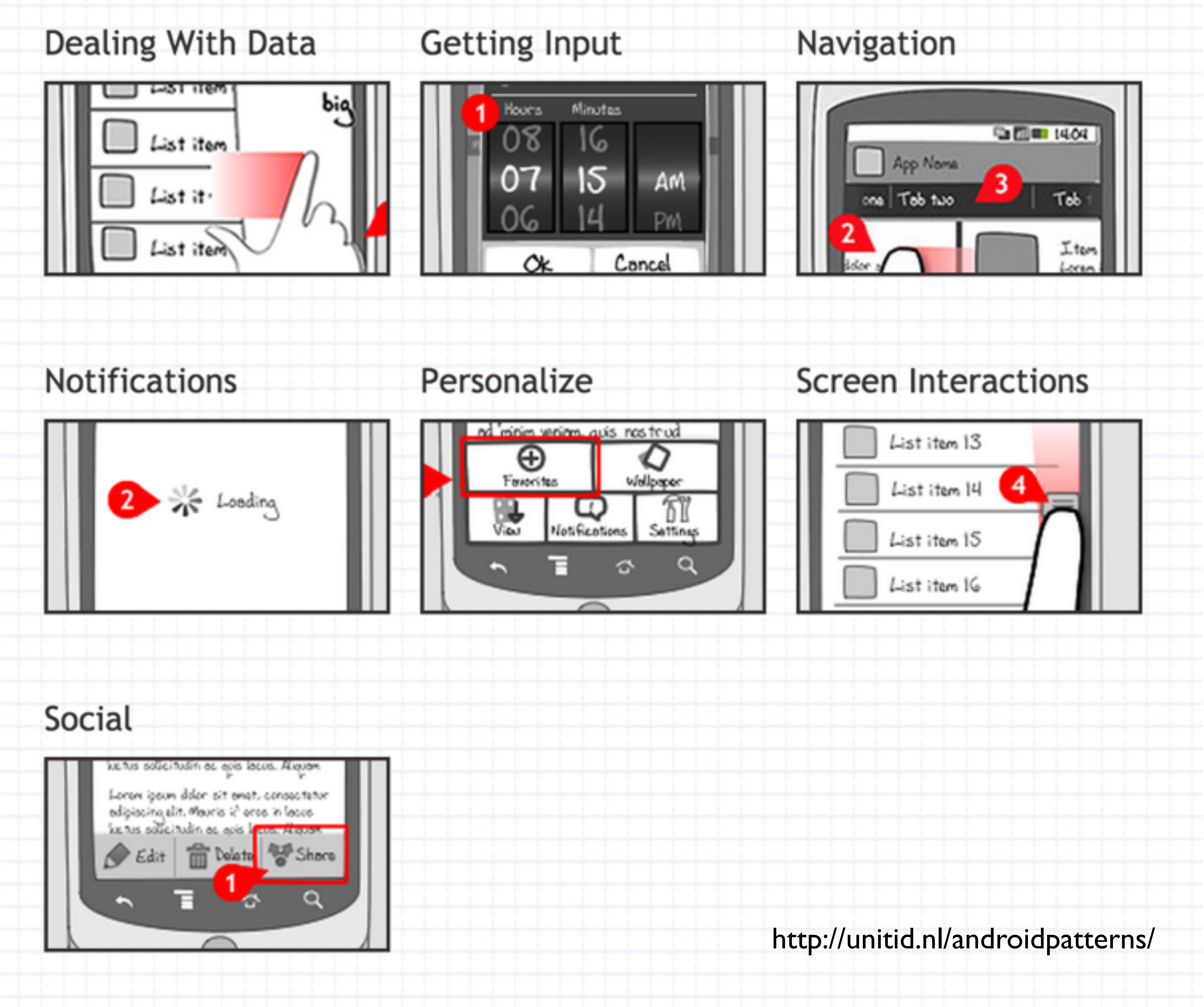
Spread
Touch surface with two fingers and move them apart

Press
Touch surface for extended period of time

Press and tap
Press surface with one finger and briefly touch surface with second finger

Rotate
Touch surface with two fingers and move them in a clockwise or counterclockwise direction

Mobile interaction design patterns



http://unitid.nl/androidpatterns/

Official UI Guidelines

Official user interface guidelines from the manufacturers are another source of inspiration for mobile web design.

You will find guidelines, samples, tips, and descriptions of common mistakes. Many of the guidelines focus on native application development, but we can apply most parts of them to mobile web design, too.

The most important guides are:

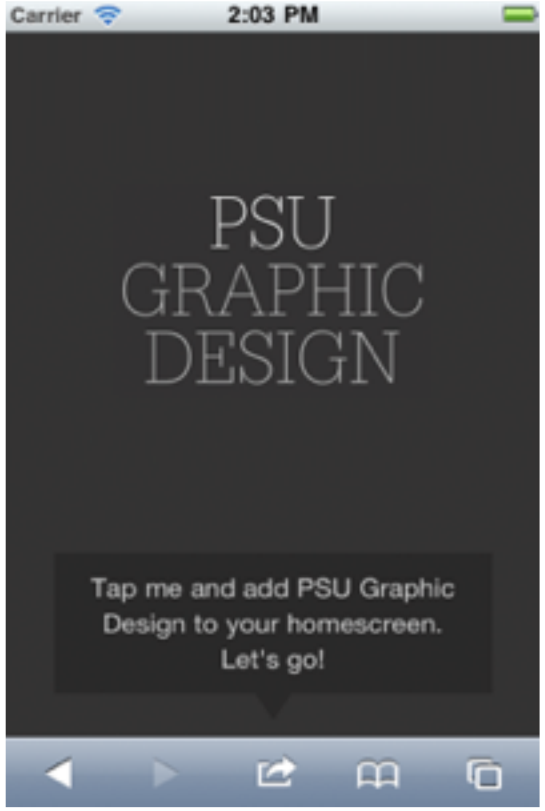
- iOS Human Interface Guidelines
  http://developer.apple.com/library/ios/#documentation/userexperience/conceptual/mobilehig/Introduction/Introduction.html

- UI Guidelines for Android
  http://developer.android.com/guide/practices/ui_guidelines/index.html

- UI Guidelines for Windows Mobile
  http://msdn.microsoft.com/en-us/library/bb158602.aspx

Design inspiration for Mobile

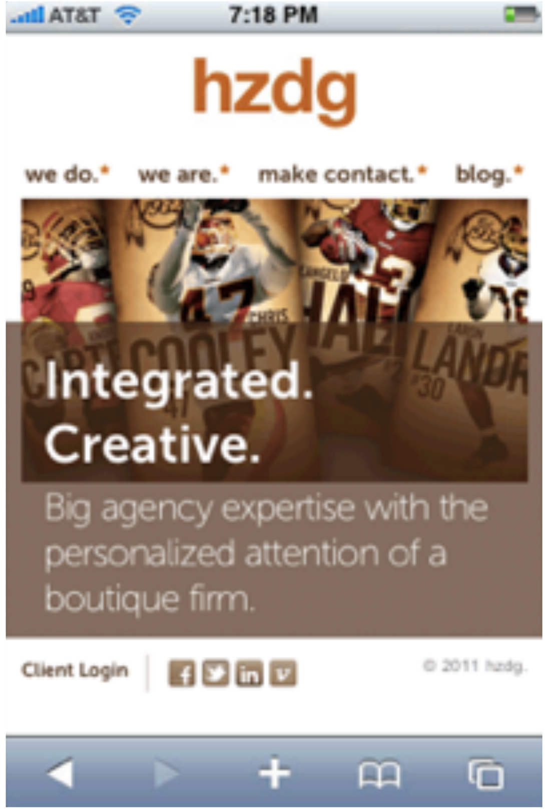
- http://www.mobileawesomeness.com/
- http://cssiphone.com/
- http://dribbble.com/tags/mobile
- http://www.iosinspires.me/
- http://www.tappgala.com/



Carrier 2:03 PM

PSU GRAPHIC DESIGN

Tap me and add PSU Graphic Design to your homescreen. Let's go!



AT&T 7:18 PM

hzdg

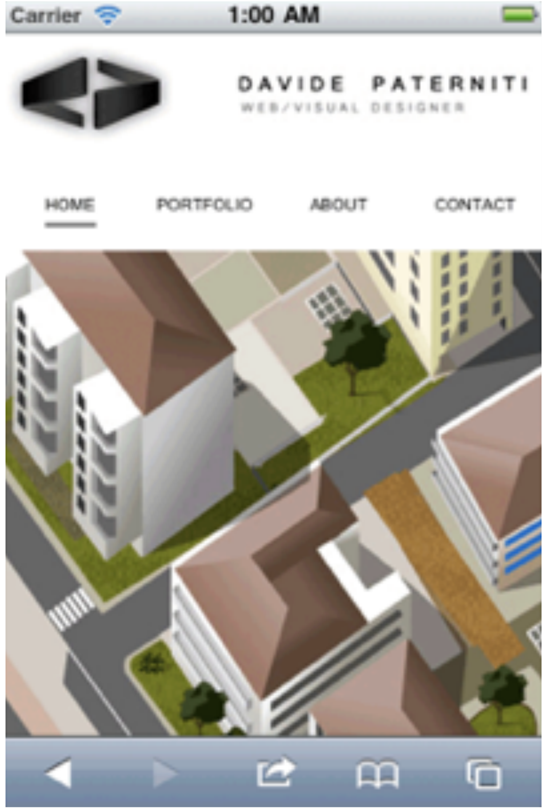
we do.* we are.* make contact.* blog.*

Integrated. Creative.

Big agency expertise with the personalized attention of a boutique firm.

Client Login

© 2011 hzdg.

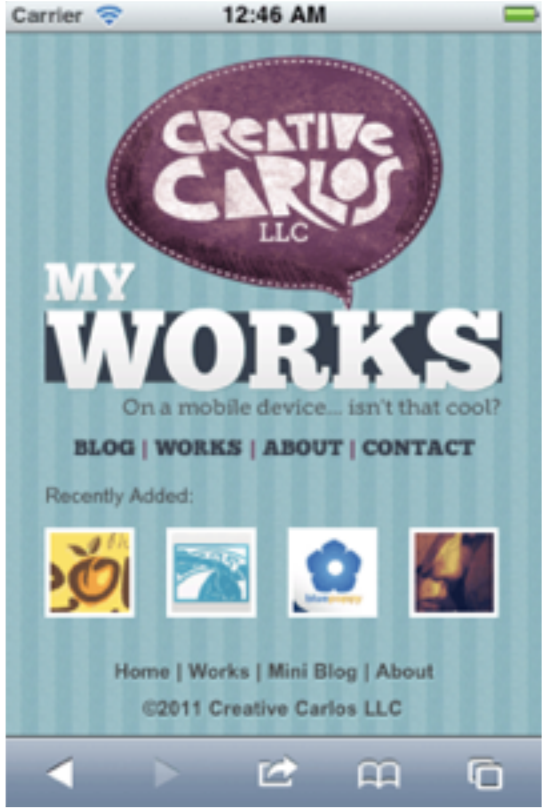


Carrier 1:00 AM

DAVIDE PATERNITI

WEB/VISUAL DESIGNER

HOME PORTFOLIO ABOUT CONTACT



Carrier 12:46 AM

CREATIVE CARLOS LLC

MY WORKS

On a mobile device... isn't that cool?

BLOG | WORKS | ABOUT | CONTACT

Recently Added:

Home | Works | Mini Blog | About

©2011 Creative Carlos LLC

Mobile Web Best Practices

W3C recommendation

The World Wide Web Consortium specifies Best Practices for delivering Web content to mobile devices. The principal objective is to improve the user experience of the Web when accessed from such devices.

The recommendations refer to delivered content and not to the processes by which it is created, nor to the devices or user agents to which it is delivered.

http://www.w3.org/TR/mobile-bp/

1 - Design for One Web

W3C recommendation

Content designed with diverse devices in mind reduces cost, increases flexibility, and reaches the needs of more people.

- **THEMATIC CONSISTENCY**: Ensure that content provided by accessing a URI yields a thematically coherent experience when accessed from different devices.
- **CAPABILITIES**: Exploit device capabilities to provide an enhanced user experience.
- **DEFICIENCIES**: Take reasonable steps to work around deficient implementations.
- **TESTING**: Carry out testing on actual devices as well as emulators.

2 - Rely on Web standards

W3C recommendation

In the highly fragmented market of devices and browsers, standards are the best guarantee for interoperability.

- **VALID MARKUP**: Create documents that validate to published formal grammars.
- **CONTENT FORMAT SUPPORT**: Send content in a format known by the device.
- **CONTENT FORMAT PREFERRED**: Where possible, send content in a preferred format.
- **CHARACTER ENCODING SUPPORT**: Ensure that content is encoded using a character encoding known to be supported by the target device.
- **CHARACTER ENCODING USE**: Indicate in the response the character encoding being used.
- **STYLE SHEETS USE**: Use style sheets to control layout and presentation, unless the device is known not to support them.
- **STRUCTURE**: Use features of the markup language to indicate logical document structure.
- **ERROR MESSAGES**: Provide informative error messages and a means of navigating away from an error message back to useful information.

3 - Stay away from known hazards

W3C recommendation

Thoughtful design can help reduce usability problems due to small screens and keyboards, and other features of mobile devices.

- **POP UPS**: Do not cause pop ups or other windows to appear and do not change the current window without informing the user.
- **TABLES NESTED**: Do not use nested tables.
- **TABLES LAYOUT**: Do not use tables for layout.
- **GRAPHICS FOR SPACING**: Do not use graphics for spacing.
- **NO FRAMES**: Do not use frames.
- **IMAGE MAPS**: Do not use image maps unless you know the device supports them effectively.

4 - Be cautious of device limitations

W3C recommendation

When choosing to use a particular Web technology, consider that mobile devices vary greatly in capability.

- **COOKIES**: Do not rely on cookies being available.
- **OBJECT OR SCRIPT**: Do not rely on embedded objects or script.
- **TABLES SUPPORT**: Do not use tables unless the device is known to support them.
- **TABLES ALTERNATIVES**: Where possible, use an alternative to tabular presentation.
- **STYLE SHEETS SUPPORT**: Organize documents so that if necessary they may be read without style sheets.
- **FONTS**: Do not rely on support of font related styling.
- **USE OF COLORS**: Ensure that information conveyed with color is also available without color.

5 - Optimize navigation

W3C recommendation

Simple navigation and typing become critical when using a small screen and keyboard, and limited bandwidth.

- **NAVBAR**: Provide only minimal navigation at the top of the page.
- **NAVIGATION**: Provide consistent navigation mechanisms.
- **LINK TARGET ID**: Clearly identify the target of each link.
- **LINK TARGET FORMAT**: Note the target file format unless you know the device supports it.
- **ACCESS KEYS**: Assign access keys to links in navigational menus and frequently accessed functionality.
- **URIS**: Keep the URIs of site entry points short.
- **BALANCE**: Take into account the trade off between having too many links on a page and asking the user to follow too many links to reach what they are looking for.

6 - Check graphics & colors

W3C recommendation

Images, colors, and style brighten content, but require care: devices may have low-contrast screens or may not support some formats.

- **IMAGES RESIZING**: Resize images at the server, if they have an intrinsic size.
- **LARGE GRAPHICS**: Do not use images that cannot be rendered by the device. Avoid large or high resolution images except where critical information would otherwise be lost.
- **IMAGES SPECIFY SIZE**: Specify the size of images in markup, if they have an intrinsic size.
- **NON-TEXT ALTERNATIVES**: Provide a text equivalent for every non-text element.
- **COLOR CONTRAST**: Ensure that foreground and background color combinations provide sufficient contrast.
- **BACKGROUND IMAGE READABILITY**: When using background images make sure that content remains readable on the device.
- **MEASURES**: Do not use pixel measures and do not use absolute units in markup language attribute values and style sheet property values.

7 - Keep it small

W3C recommendation

Smaller sites make users happier by costing less in time and money.

- **MINIMIZE**: Use terse, efficient markup.
- **PAGE SIZE LIMIT**: Ensure that the overall size of page is appropriate to the memory limitations of the device.
- **STYLE SHEETS SIZE**: Keep style sheets small.
- **SCROLLING**: Limit scrolling to one direction, unless secondary scrolling cannot be avoided.

8 - Use the network sparingly

W3C recommendation

Web protocol features can help improve the user experience by reducing the impact of network bottlenecks and latencies.

- **AUTO REFRESH**: Do not create periodically auto refreshing pages, unless you have informed the user and provided a means of stopping it.
- **REDIRECTION**: Do not use markup to redirect pages automatically. Instead, configure the server to perform redirects by means of HTTP 3xx codes.
- **EXTERNAL RESOURCES**: Keep the number of externally linked resources to a minimum.
- **CACHING**: Provide caching information in HTTP responses

9 - Help & guide user input

W3C recommendation

Keyboards and other input methods on mobile devices can be tedious to use, so effective designs minimize the need for them.

- **MINIMIZE KEYSTROKES**: Keep the number of keystrokes to a minimum.
- **AVOID FREE TEXT**: Avoid free text entry where possible.
- **PROVIDE DEFAULTS**: Provide pre-selected default values where possible.
- **DEFAULT INPUT MODE**: Specify a default text entry mode, language and/or input format, if the target device is known to support it.
- **TAB ORDER**: Create a logical order through links, form controls and objects.
- **CONTROL LABELLING**: Label all form controls appropriately and explicitly associate labels with form controls.
- **CONTROL POSITION**: Position labels so they lay out properly in relation to the form controls they refer to.

10 - Think of users on the go

W3C recommendation

Web users on the go want compact information when time is short and distractions many.

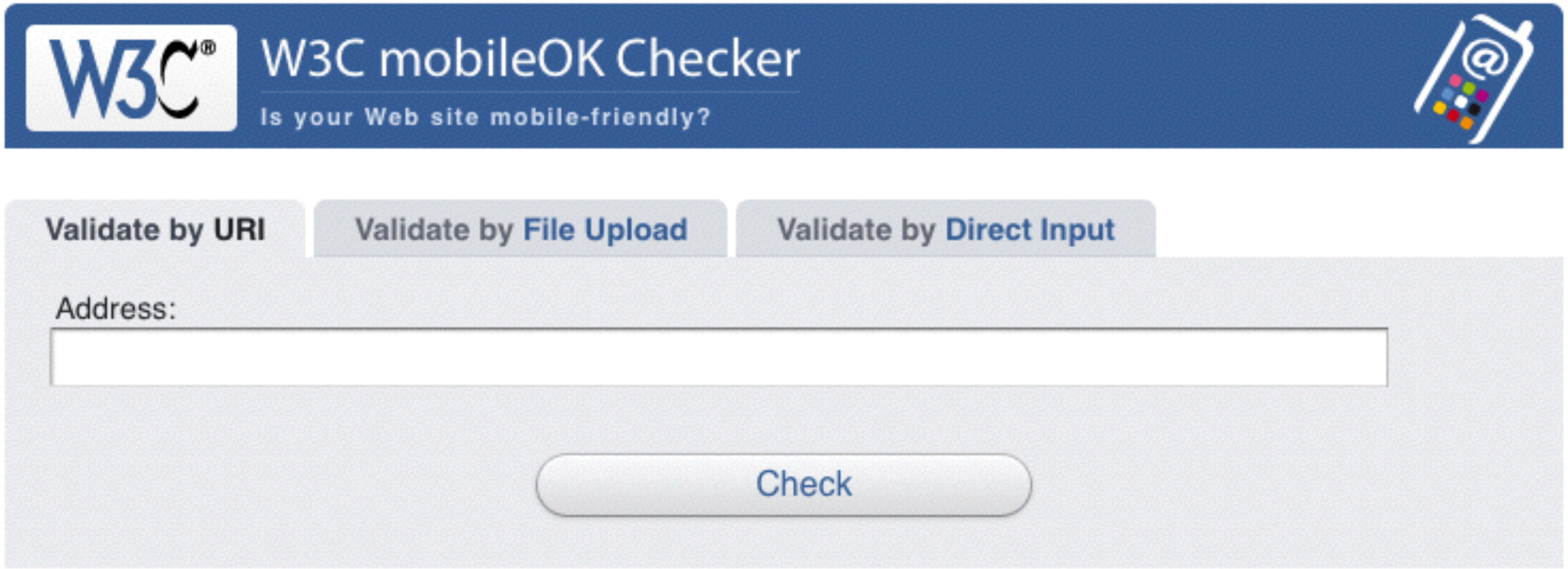
- **PAGE TITLE**: Provide a short but descriptive page title.
- **CLARITY**: Use clear and simple language.
- **CENTRAL MEANING**: Ensure that material that is central to the meaning of the page precedes material that is not.
- **LIMITED**: Limit content to what the user has requested.
- **SUITABLE**: Ensure that content is suitable for use in a mobile context.
- **PAGE SIZE USABLE**: Divide pages into usable but limited size portions.

W3C Mobile OK Checker

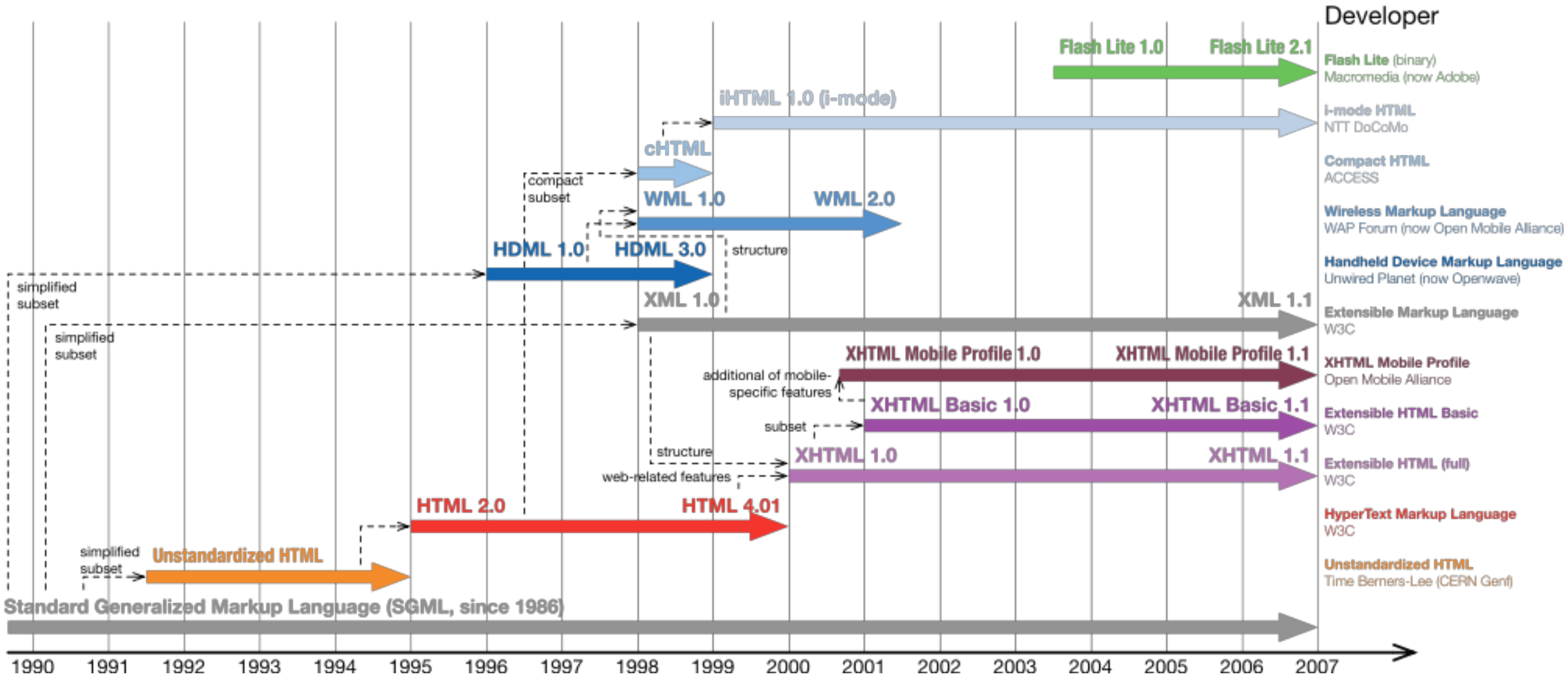
W3C recommendation

The W3C mobile Checker is a free service by W3C that helps check the level of mobile-friendliness of Web documents

http://validator.w3.org/mobile/



Evolution of mobile web markup languages



Mobile Web 2.0

Mobile Web 2.0 began in 2007, with the new smartphones that appeared on the market (iPhone, Nokia N95, Android devices, etc.). These devices introduced great changes for the mobile web: WiFi support, 3G, full desktop browsers (not only WAP 2.0), Ajax or Flash support, and streaming video.

All these ingredients created Mobile Web 2.0, which to date has advanced more than Web 2.0 itself. That's why we can find HTML 5 mobile browsers, but a few HTML 5 desktop browsers on the market today. Clearly, the market really wanted to create better mobile web experiences than the ones delivered only with WAP.

Mobile Web 2.0 sites typically have many of the following features:

- Ajax or Rich Internet Application experience
- Geolocation
- Offline working capability
- Social networking activities
- Contextual ads
- On-demand/live video streaming
- HTML 4/5, CSS 2/3, JavaScript
- Touch/multitouch support

Markup for mobile today

Many of the new breed of phones and tablets, including iOS and Android, have no support for WML. Instead, they have full-featured web browsers that can understand and render HTML on par with their desktop equivalents. Consequently, there's really no need to look to new languages or different standards for the mobile web. For now and into the foreseeable future, HTML is going to be the markup language of choice for mobile web development.

Using standards-based web development techniques will ensure the most consistent presentation and experience for our users, both now and as new devices arrive. A well-structured HTML page with clean, semantic markup will display correctly, and be both usable and accessible on any device—desktop or mobile.

Mobile Emulators

In order to develop and test for mobile devices we could use one or more of the emulator provided by vendors

Model	**Platform**	**Browser testing**	**Compatibility**
Apple iOS	iOS	Safari	Mac
Google Android	Android	Android	Win, Mac, Unix
Nokia Symbian	Symbian	S60 Browser	Win
Windows Phone 7	Windows Phone	Internet Explorer	Win
BlackBerry	RIM OS	RIM Browser	Win

Doctypes

XHTML Mobile Profile (XHTML MP)

It is an XHTML document type defined by the Open Mobile Alliance. XHTML-MP is derived from XHTML Basic 1.0 by adding XHTML Modules, with later versions of the standard adding more modules. However, for certain modules, XHTML-MP does not mandate a complete implementation so an XHTML-MP browser may not be fully conforming on all modules.

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
     <!DOCTYPE html PUBLIC "-//WAPFORUM//DTD XHTML Mobile 1.2//EN"
      "http:// www.openmobilealliance.org/tech/DTD/xhtml-
mobile12.dtd">
```

Doctypes

XHTML Basic

The most common markup language used for mobile web sites is XHTML Basic. This standard ensures specific markup for your web site that works best on mobile devices. For instance, it does not allow HTML frames or nested tables, which perform poorly on mobile devices. Along with the DOCTYPE, be sure to declare the appropriate character encoding for the document (such as UTF-8).

XHTML Basic 1.1 became a W3C Recommendation in July 2008, superseding XHTML-MP 1.2

Example:

```
<?xml version="1.0" encoding="UTF-8"?>
     <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.1//EN"
      "http://www.w3.org/TR/xhtml-basic/xhtml-basic11.dtd">
```

Doctypes

HTML5

HTML5 provides mobile device users richer web applications and improved usability.

The new features of HTML5 standardize the use cases and technologies that are common in smartphone-optimized mobile web applications.

Example:

```
<!DOCTYPE HTML>
```

Viewport

The viewport is the area in which the page fits. You can specify its width and height, and it can be larger or smaller than the total visible area of the screen.

This is where the scale and zoom features of the mobile browser come into play.
You define the viewport using the <meta name="viewport>" tag inside the <head> of the document.
The content of the tag can be a comma-separated list of one or more of the attributes listed below.

Example:

```
<meta name="viewport" content="width=600, initial-scale=1.0"/
```

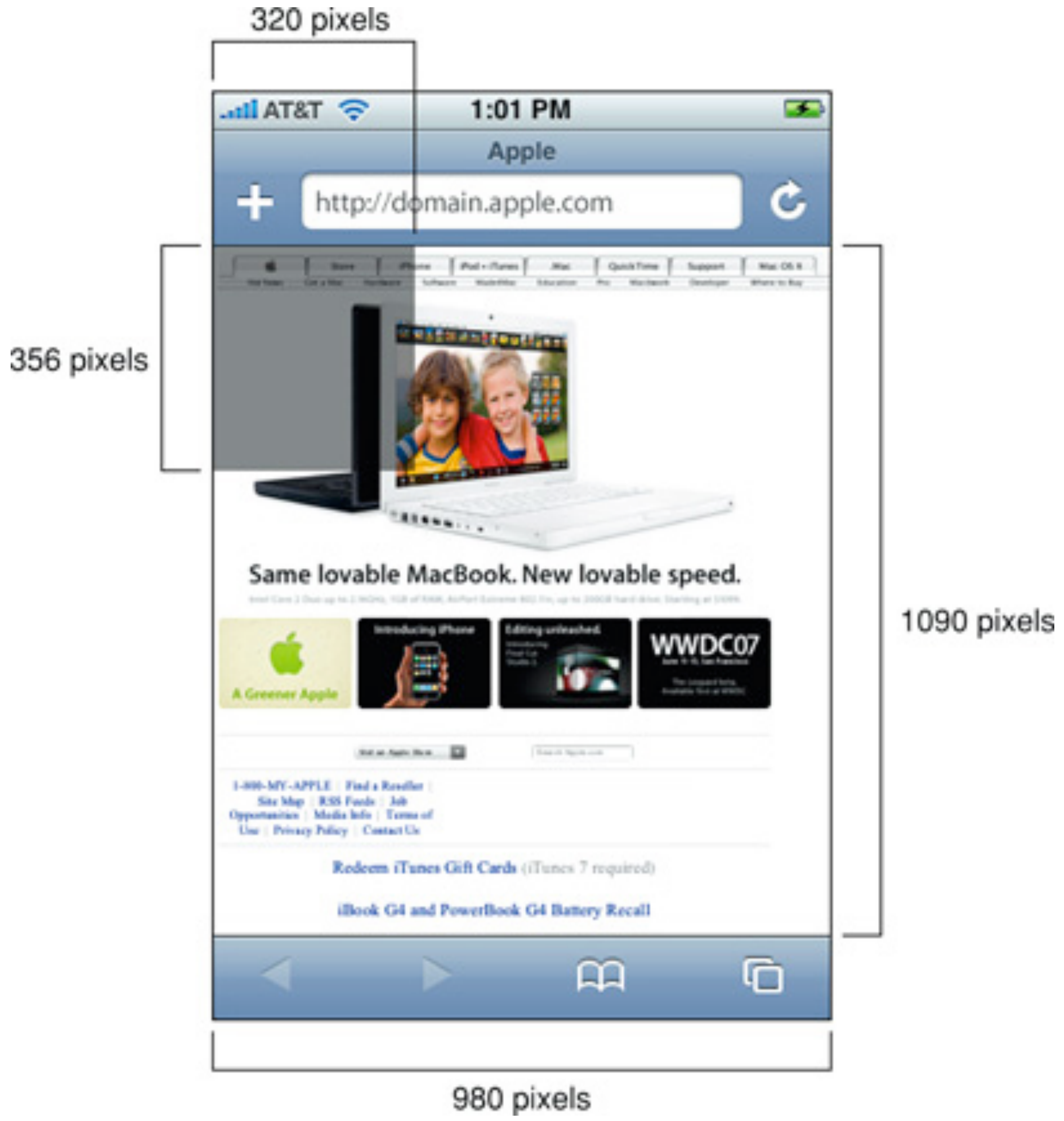
Attribute	**Possible values**	**Description**
width	Integer value (in pixels) or constant device- width	Defines the viewport width
height	Integer value (in pixels) or constant device- height	Defines the viewport height
initial-scale	Floating value (0.1 to n); 1.0 is no scale	Defines the initial zoom scale of the viewport
user-scalable	no or yes	Defines whether we will allow the user to zoom in and out
minimum-scale	Floating value (0.1 to n). 1.0 is no scale	Defines the minimum zoom scale of the viewport
maximum-scale	Floating value (0.1 to n). 1.0 is no scale	Defines the maximum zoom scale of the viewport

Viewport for mobile

Many phone browsers scale web pages down to a wider viewport width to fit them onto the screen.

These browsers allow the user to zoom in and scale the pages up to view the bit they want to see.

For example, although a device screen might have a width of 320px the viewport can have a width of 980px. In this case a web page designed at 980px or less will fit completely on the screen.



Example: Safari on iOS

The majority of webpages fit nicely in the visible area with the viewport width set to 980 pixels in portrait orientation.
If Safari on iOS did not set the viewport width to 980 pixels, then only the upper-left corner of the webpage, shown in gray, would be displayed.

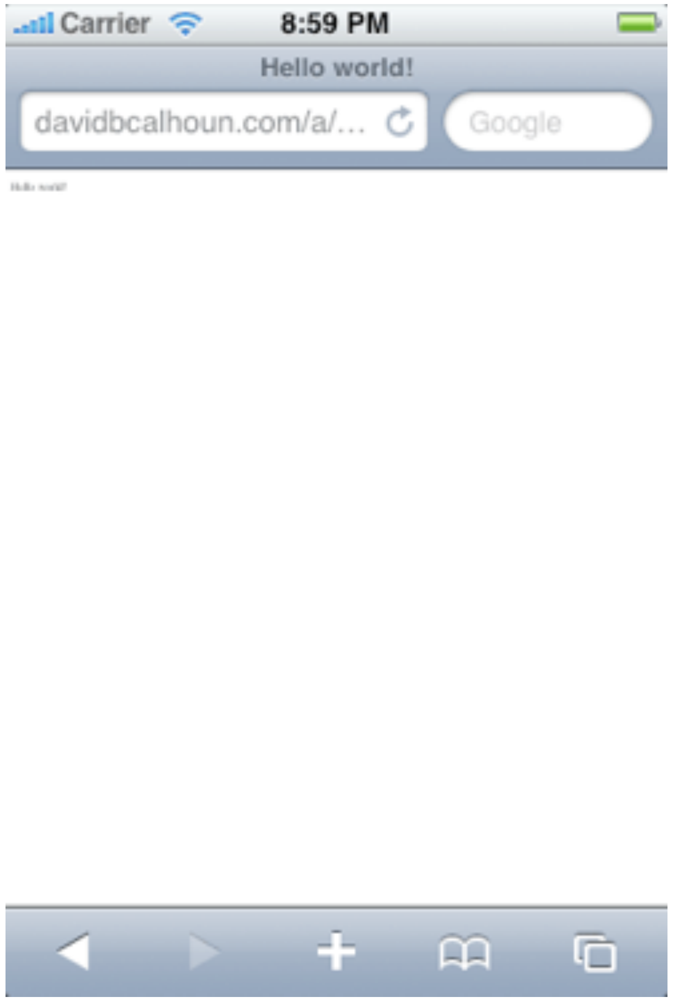
Viewport for mobile

If you open a document without a viewport assigned value on an mobile browser, the default viewport size will be used.

Some default value are 960px for Safari on iOS and 800 for Android

Example

```
<!doctype html>
<html>
<head>
    <title>Hello world!</title>
</head>

<body>

<p>Hello world!</p>

</body>
</html>
```



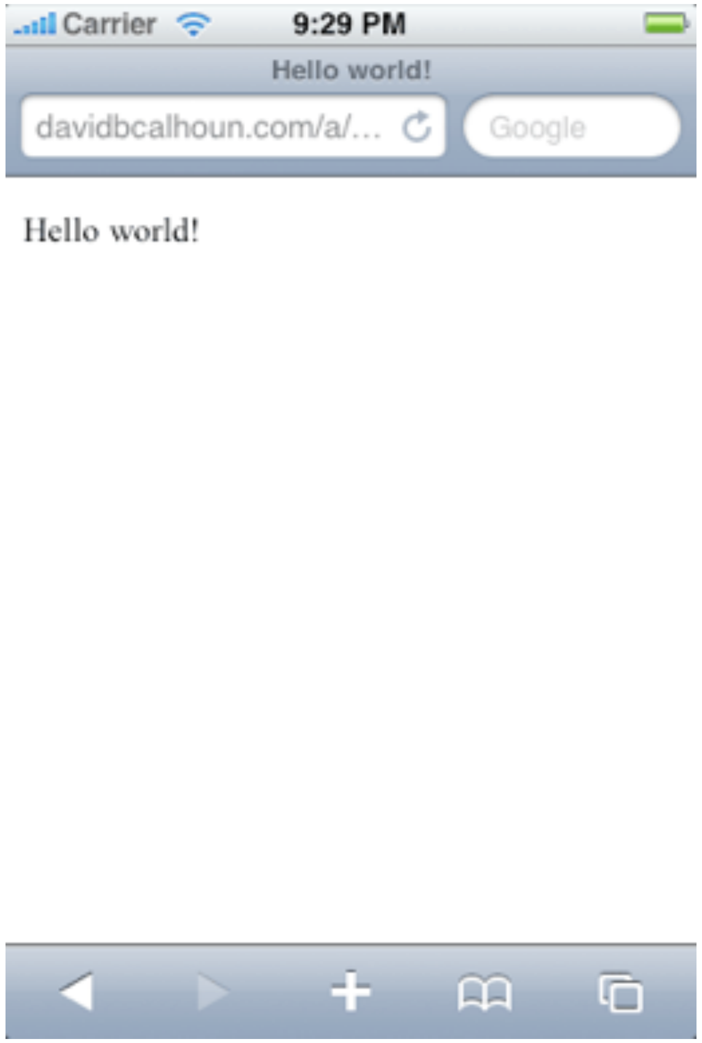
Viewport for mobile

You can define a viewport with a fixed size (in case you are showing a desktop-friendly website), or with a size relative to the visible area.

The most common approach for mobile websites is to define the width as device-width.

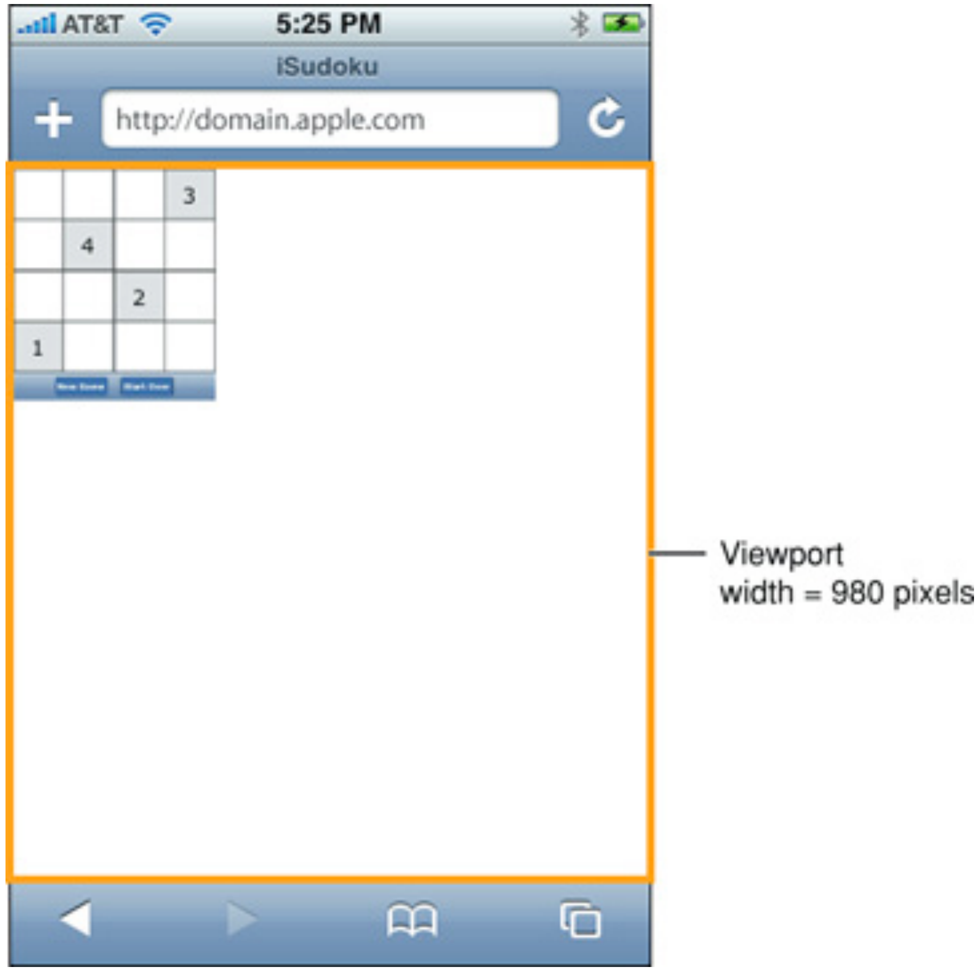
Example

```
<!doctype html>
<html>
<head>
    <title>Hello world!</title>

    <meta name="viewport"
       content="width=device-width"/>
</head>

<body>

<p>Hello world!</p>

</body>
</html>
```

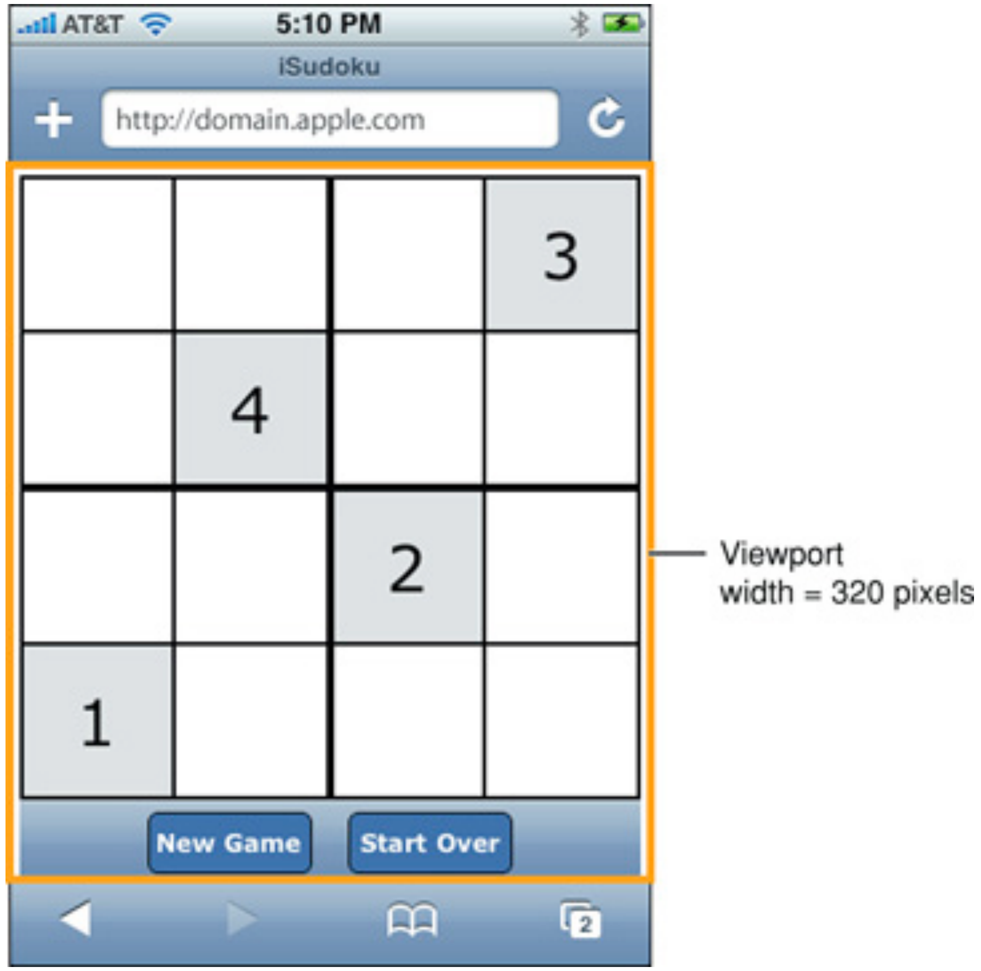


Hello world!

Viewport for mobile



Not specifying viewport properties



Width set to device-width pixels

Interesting reading: "A pixel is not a pixel" by Peter-Paul Koch
http://www.quirksmode.org/blog/archives/2010/04/a_pixel_is_not.html

CSS media filtering

The CSS standard allows us to define more than one stylesheet for the same document, taking into account the possibility of a site being rendered on different types of media.

The most used values for the media attribute are **screen** (for desktops), **print** (to be applied when the user prints the document), and **handheld** (for mobile devices). There are also other values, like tv and braille, but no browsers currently support these.

```
<link rel="stylesheet" type="text/css" media="screen"
href="desktop.css" />

<link rel="stylesheet" type="text/css" media="handheld"
href="mobile.css" />
```

PROBLEM: Many modern mobile browsers rely on screen stylesheets because they can render any desktop website

CSS media queries

A media query consists of a media type and zero or more expressions that check for the conditions of particular media features.

Among the media features that can be used in media queries are ‘**width**’, ‘**height**’, and ‘color’. By using media queries, presentations can be tailored to a specific range of output devices without changing the content itself

For example, to specify a style sheet for iPhone and iPod touch, you can use an expression similar to the following:

```
<link media="only screen and (max-device-width: 480px)"
href="small-device.css" type= "text/css" rel="stylesheet">
```

To specify a style sheet for devices other than iOS, use an expression similar to the following:

```
<link media="screen and (min-device-width: 481px)" href="not-
small-device.css" type="text/css" rel="stylesheet">
```

CSS media queries

Onother way to use media queries is to define a block inside the stylesheet of the page using the @media selector.

(in styles.css)

```
@media screen and (min-device-width: 481px) {

      #text { color: white; background-color: black; }

 }

...
```

specification: http://www.w3.org/TR/css3-mediaqueries/

Using phone features

There are some URL schemes that many mobile browsers understand to communicate with some phone features.

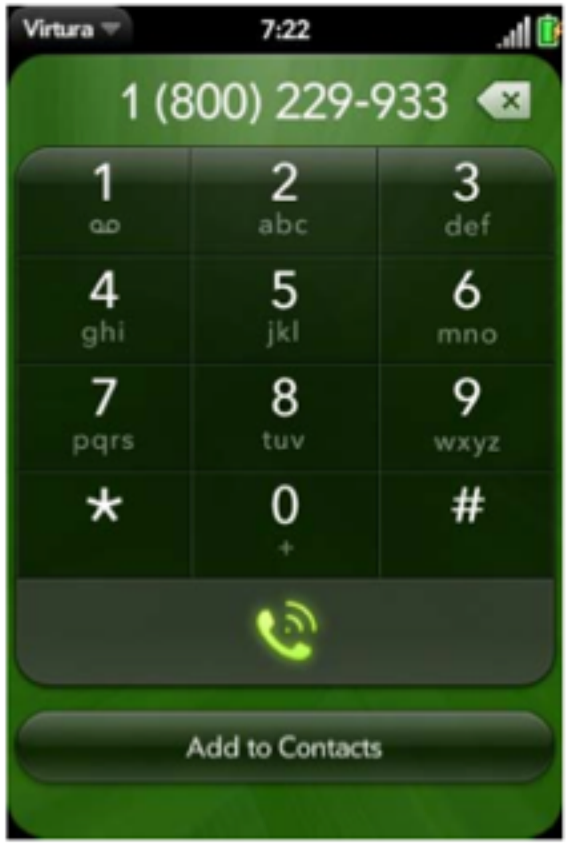
- Making calls
- Sending SMS
- Sending email

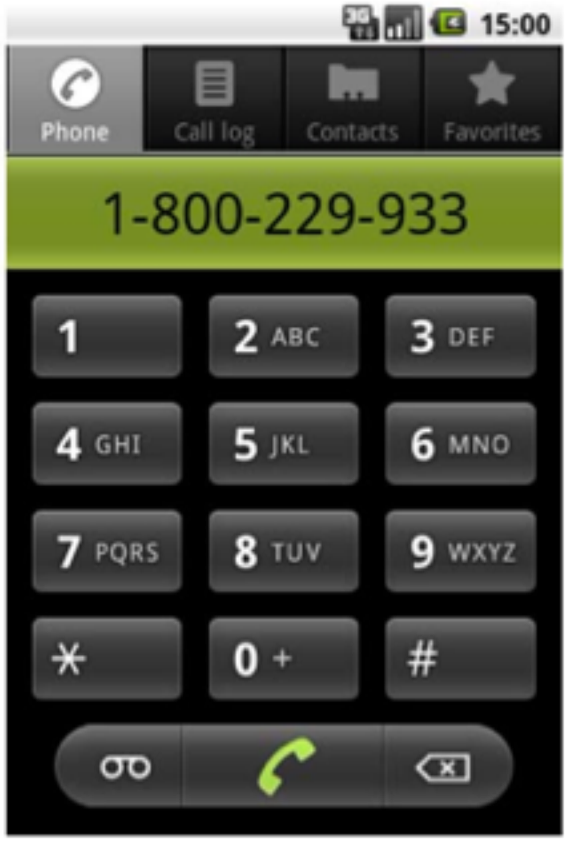
Using phone features - making calls

Remember: most mobile devices are also phones!

The first de facto standard (copied from the Japanese i-Mode standards) is to use the tel:<phone number> scheme. This is called the i-Mode format:

```
<a href="tel:+1800229933">Call us free!</a>
```





Using phone features - sending SMS

To invoke the SMS window from a link we have two possible URI schemes, **sms:** and **smsto:**.

Unfortunately, there is no standard way to know for sure which one is compatible with a user's browser.

The parameters usually define the body, but this property is not compatible with all phones for security reasons

```
<a href="sms:+3490322111?body=Interested%20in%20Product%20AA2">
      More info for this product
</a>
```

Using phone features - sending email

Some modern devices with browsers also have mail applications that can react to the classic web **mailto:** protocol. The syntax is **mailto:<email_destination>[?parameters]**.

The detected parameters can change from device to device but generally include cc, bcc, subject, and body. The parameters are defined in a URL format (key=value&key=value), and the values must be URI-encoded.

Here are some samples:

```
<a href="mailto:info@example.com">Mail us</a>
```

```
<a href="mailto:info@example.com?subject=Contact&body=This%20is
%20the%20body"> Mail us</a>
```

Preventing format detection

In fact, for the iPhone, there’s no need to even wrap the number in a hyperlink.

Mobile Safari includes a feature that **automatically detects phone numbers** in a page and turns them into links for you.
Similarly, there’s a feature that turns address-like text into map links.

But like any automatic feature, it’s not always what you want:
Imagine that Safari tries to turn your product IDs into “dialable” phone numbers...

You can include <meta> tags in the head of your page to disable these features:

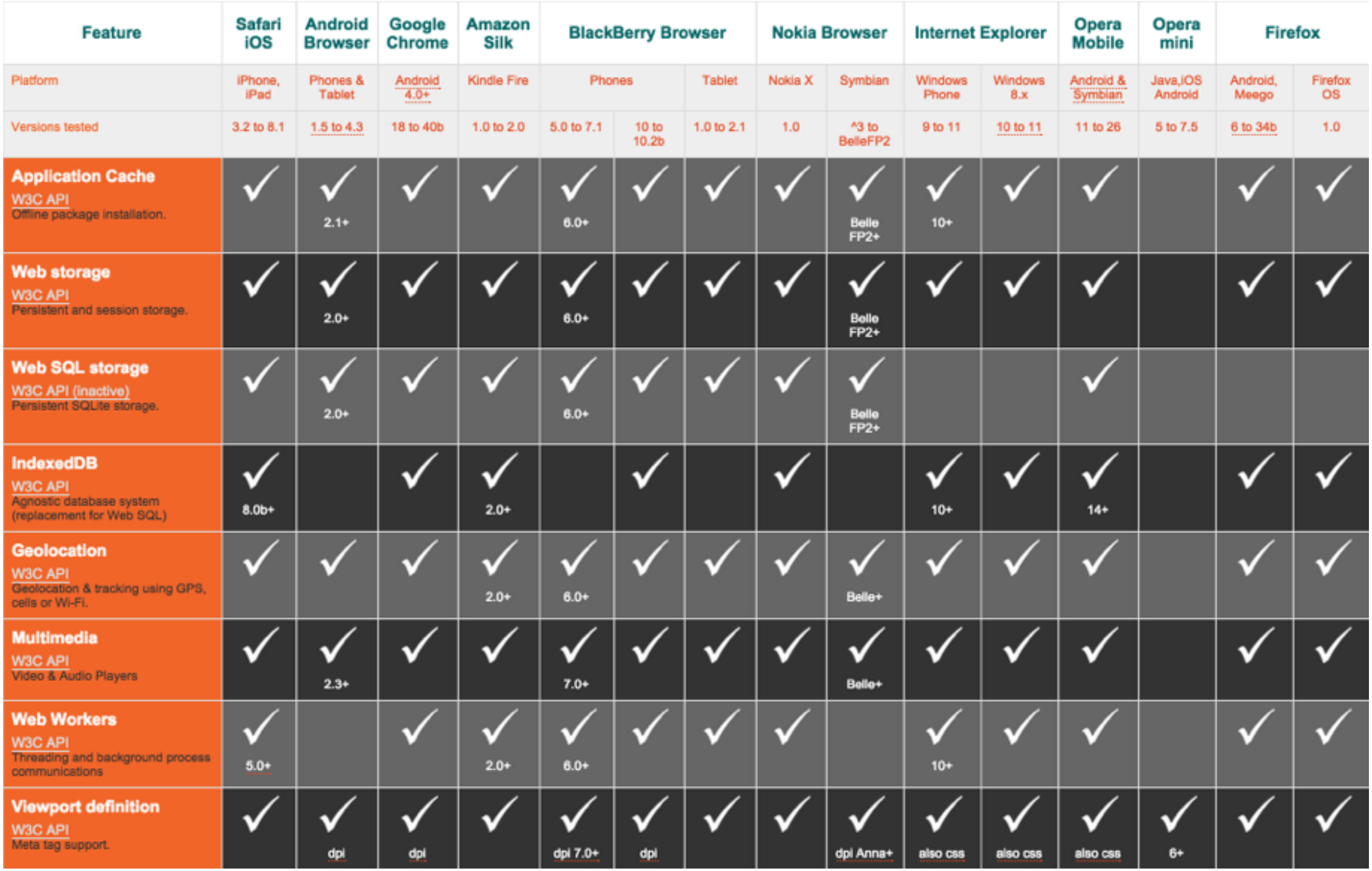
```
<meta name="format-detection" content="telephone=no"/>
<meta name="format-detection" content="address=no"/>
```

HTML5 for mobile

HTML5 provides mobile device users richer web applications and improved usability. The new features of HTML5 standardize the use cases and technologies that are common in smartphone-optimized mobile web applications.

In today's Mobile Web of WML or XHTML or HTML documents, these features are implemented using proprietary device and browser APIs. With HTML5, advanced web application features are available in all mobile browsers supporting the markup language, using the same standard syntax and displaying the same standard behavior

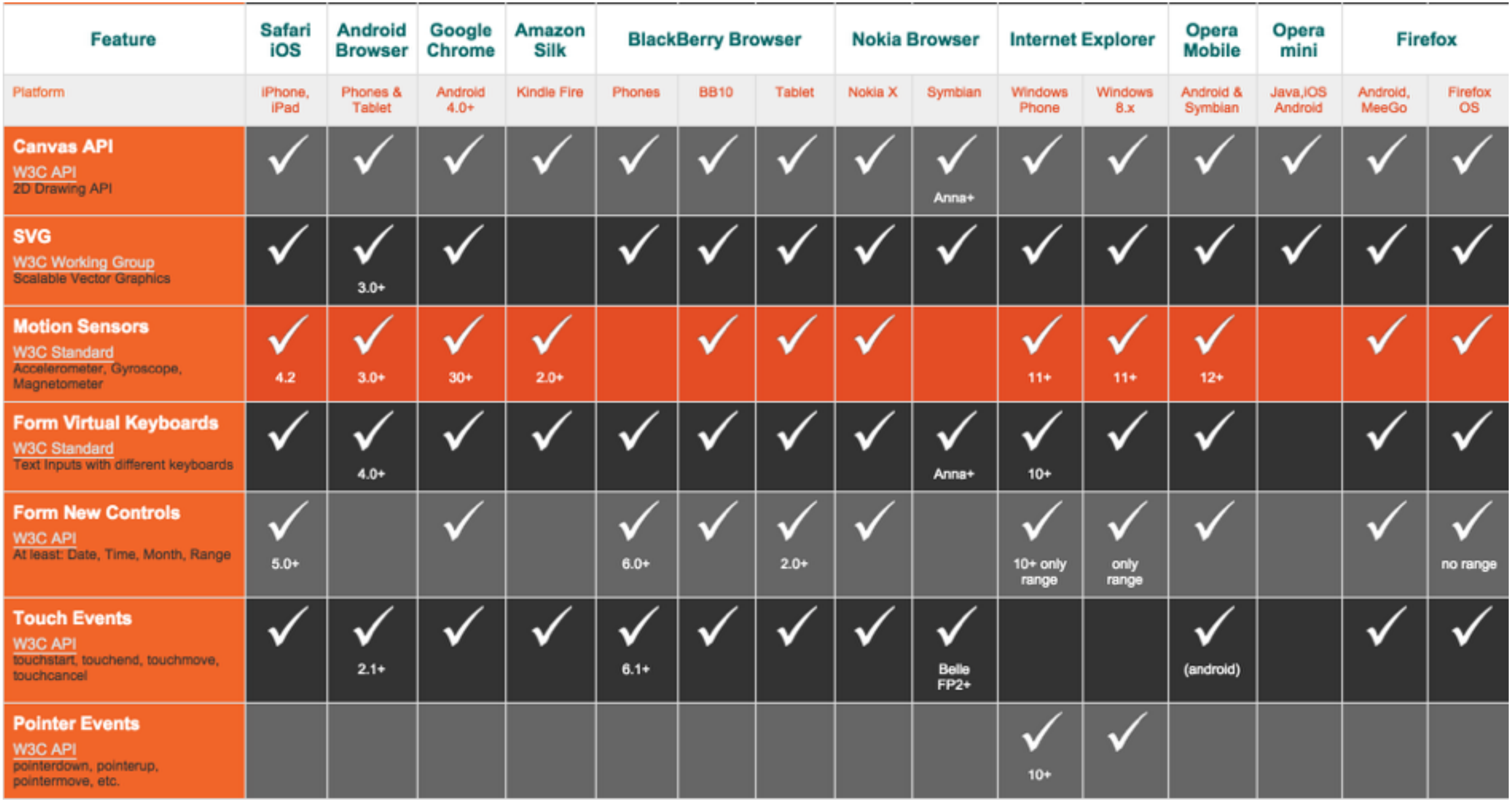
HTML5 for mobile



Feature	Safari iOS	Android Browser	Google Chrome	Amazon Silk	BlackBerry Browser			Nokia Browser		Internet Explorer		Opera Mobile	Opera mini	Firefox	
Platform	iPhone, iPad	Phones & Tablet	Android 4.0+	Kindle Fire	Phones		Tablet	Nokia X	Symbian	Windows Phone	Windows 8.x	Android & Symbian	Java,iOS Android	Android, Meego	Firefox OS
Versions tested	3.2 to 8.1	1.5 to 4.3	18 to 40b	1.0 to 2.0	5.0 to 7.1	10 to 10.2b	1.0 to 2.1	1.0	^3 to BelleFP2	9 to 11	10 to 11	11 to 26	5 to 7.5	6 to 34b	1.0
Application Cache W3C API Offline package installation.	✓	✓ 2.1+	✓	✓	✓ 6.0+	✓	✓	✓	✓ Belle FP2+	✓ 10+	✓	✓		✓	✓
Web storage W3C API Persistent and session storage.	✓	✓ 2.0+	✓	✓	✓ 6.0+	✓	✓	✓	✓ Belle FP2+	✓	✓	✓		✓	✓
Web SQL storage W3C API (inactive) Persistent SQLite storage.	✓	✓ 2.0+	✓	✓	✓ 6.0+	✓	✓	✓	✓ Belle FP2+			✓			
IndexedDB W3C API Agnostic database system (replacement for Web SQL)	✓ 8.0b+		✓	✓ 2.0+		✓		✓		✓ 10+	✓	✓ 14+		✓	✓
Geolocation W3C API Geolocation & tracking using GPS, cells or Wi-Fi.	✓	✓	✓	✓ 2.0+	✓ 6.0+	✓	✓	✓	✓ Belle+	✓	✓	✓		✓	✓
Multimedia W3C API Video & Audio Players	✓	✓ 2.3+	✓	✓	✓ 7.0+	✓	✓	✓	✓ Belle+	✓	✓	✓		✓	✓
Web Workers W3C API Threading and background process communications	✓ 5.0+		✓	✓ 2.0+	✓ 6.0+	✓	✓	✓		✓ 10+	✓	✓		✓	✓
Viewport definition W3C API Meta tag support.	✓	✓ dpi	✓ dpi	✓	✓ dpi 7.0+	✓ dpi	✓	✓	✓ dpi Anna+	✓ also css	✓ also css	✓ also css	✓ 6+	✓	✓

http://mobilehtml5.org/

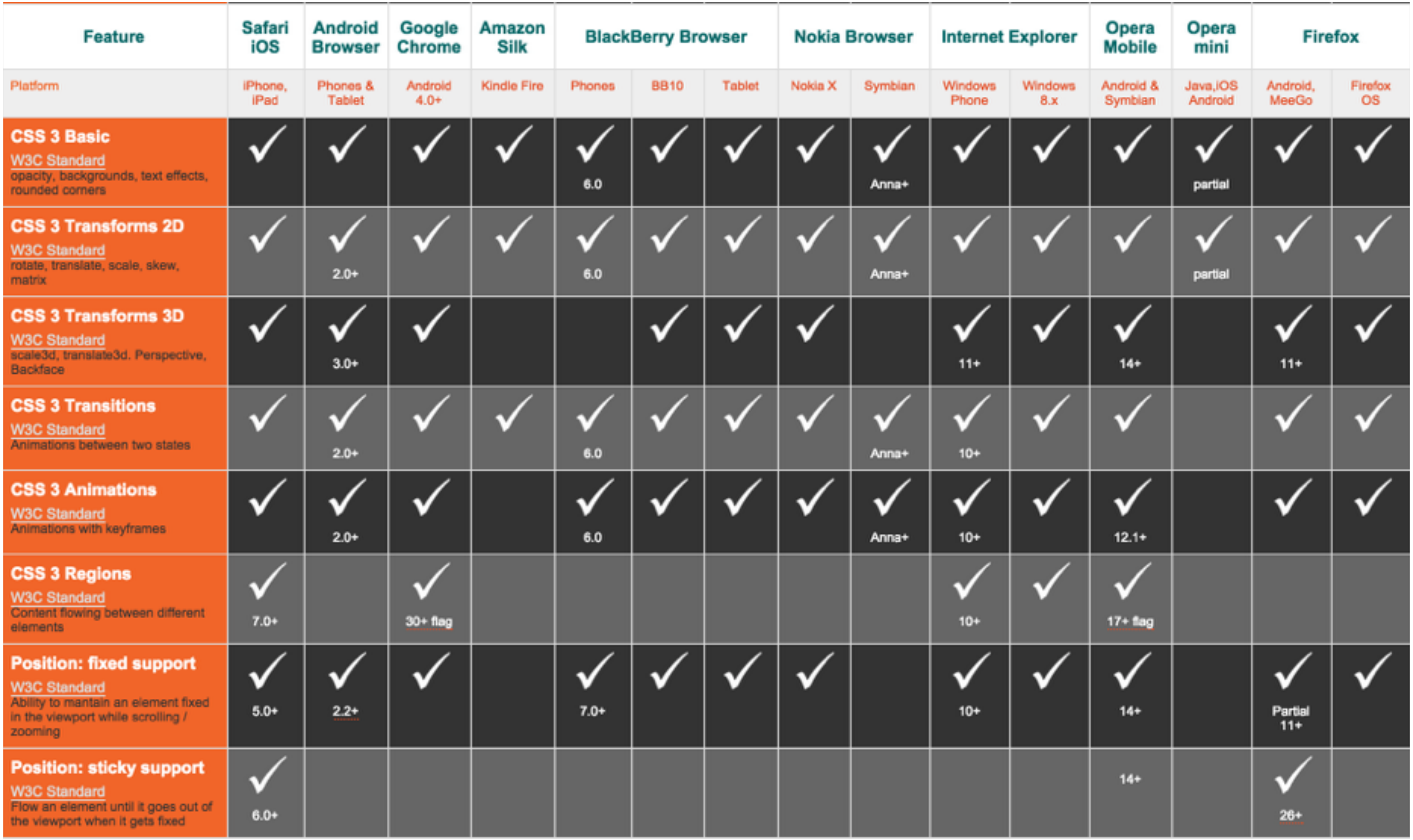
HTML5 for mobile



Feature	Safari iOS	Android Browser	Google Chrome	Amazon Silk	BlackBerry Browser			Nokia Browser		Internet Explorer		Opera Mobile	Opera mini	Firefox	
Platform	iPhone, iPad	Phones & Tablet	Android 4.0+	Kindle Fire	Phones	BB10	Tablet	Nokia X	Symbian	Windows Phone	Windows 8.x	Android & Symbian	Java,iOS Android	Android, MeeGo	Firefox OS
Canvas API W3C API 2D Drawing API	✓	✓	✓	✓	✓	✓	✓	✓	✓ Anna+	✓	✓	✓	✓	✓	✓
SVG W3C Working Group Scalable Vector Graphics	✓	✓ 3.0+	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Motion Sensors W3C Standard Accelerometer, Gyroscope, Magnetometer	✓ 4.2	✓ 3.0+	✓ 30+	✓ 2.0+		✓	✓	✓		✓ 11+	✓ 11+	✓ 12+		✓	✓
Form Virtual Keyboards W3C Standard Text Inputs with different keyboards	✓	✓ 4.0+	✓	✓	✓	✓	✓	✓	✓ Anna+	✓ 10+	✓	✓		✓	✓
Form New Controls W3C API At least: Date, Time, Month, Range	✓ 5.0+		✓		✓ 6.0+	✓	✓ 2.0+	✓		✓ 10+ only range	✓ only range	✓		✓	✓ no range
Touch Events W3C API touchstart, touchend, touchmove, touchcancel	✓	✓ 2.1+	✓	✓	✓ 6.1+	✓	✓	✓	✓ Belle FP2+			✓ (android)		✓	✓
Pointer Events W3C API pointerdown, pointerup, pointermove, etc.										✓ 10+	✓				

http://mobilehtml5.org/

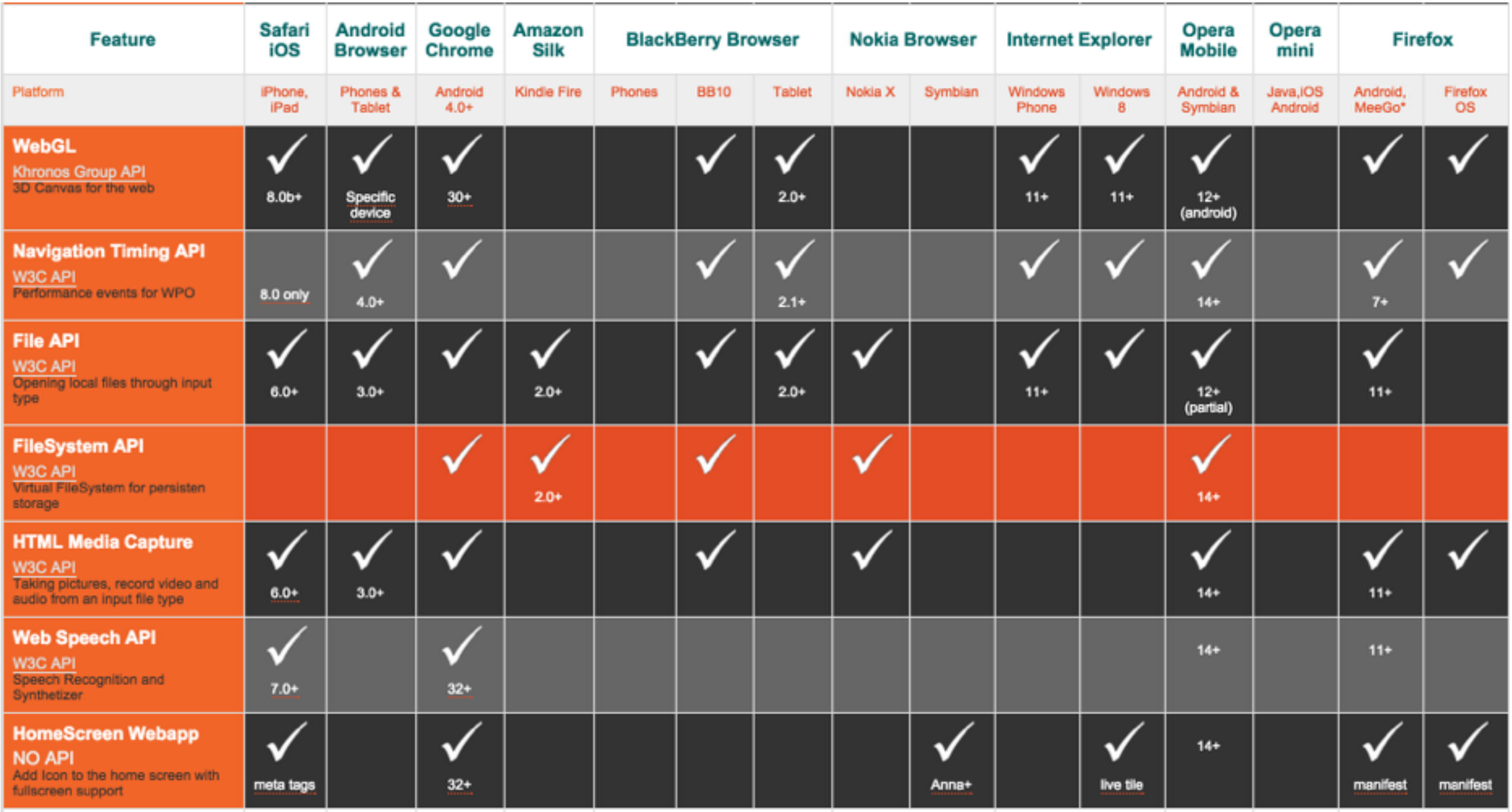
HTML5 for mobile



Feature	Safari iOS	Android Browser	Google Chrome	Amazon Silk	BlackBerry Browser			Nokia Browser		Internet Explorer		Opera Mobile	Opera mini	Firefox	
Platform	iPhone, iPad	Phones & Tablet	Android 4.0+	Kindle Fire	Phones	BB10	Tablet	Nokia X	Symbian	Windows Phone	Windows 8.x	Android & Symbian	Java,iOS Android	Android, MeeGo	Firefox OS
CSS 3 Basic W3C Standard opacity, backgrounds, text effects, rounded corners	✓	✓	✓	✓	✓ 6.0	✓	✓	✓	✓ Anna+	✓	✓	✓	✓ partial	✓	✓
CSS 3 Transforms 2D W3C Standard rotate, translate, scale, skew, matrix	✓	✓ 2.0+	✓	✓	✓ 6.0	✓	✓	✓	✓ Anna+	✓	✓	✓	✓ partial	✓	✓
CSS 3 Transforms 3D W3C Standard scale3d, translate3d. Perspective, Backface	✓	✓ 3.0+	✓			✓	✓	✓		✓ 11+	✓	✓ 14+		✓ 11+	✓
CSS 3 Transitions W3C Standard Animations between two states	✓	✓ 2.0+	✓	✓	✓ 6.0	✓	✓	✓	✓ Anna+	✓ 10+	✓	✓		✓	✓
CSS 3 Animations W3C Standard Animations with keyframes	✓	✓ 2.0+	✓		✓ 6.0	✓	✓	✓	✓ Anna+	✓ 10+	✓	✓ 12.1+		✓	✓
CSS 3 Regions W3C Standard Content flowing between different elements	✓ 7.0+		✓ 30+ flag							✓ 10+	✓	✓ 17+ flag			
Position: fixed support W3C Standard Ability to mantain an element fixed in the viewport while scrolling / zooming	✓ 5.0+	✓ 2.2+	✓		✓ 7.0+	✓	✓	✓		✓ 10+	✓	✓ 14+		✓ Partial 11+	✓
Position: sticky support W3C Standard Flow an element until it goes out of the viewport when it gets fixed	✓ 6.0+											14+		✓ 26+	

http://mobilehtml5.org/

HTML5 for mobile



Feature	Safari iOS	Android Browser	Google Chrome	Amazon Silk	BlackBerry Browser			Nokia Browser		Internet Explorer		Opera Mobile	Opera mini	Firefox	
Platform	iPhone, iPad	Phones & Tablet	Android 4.0+	Kindle Fire	Phones	BB10	Tablet	Nokia X	Symbian	Windows Phone	Windows 8	Android & Symbian	Java,iOS Android	Android, MeeGo*	Firefox OS
WebGL Khronos Group API 3D Canvas for the web	✓ 8.0b+	✓ Specific device	✓ 30+			✓	✓ 2.0+			✓ 11+	✓ 11+	✓ 12+ (android)		✓	✓
Navigation Timing API W3C API Performance events for WPO	8.0 only	✓ 4.0+	✓			✓	✓ 2.1+			✓	✓	✓ 14+		✓ 7+	✓
File API W3C API Opening local files through input type	✓ 6.0+	✓ 3.0+	✓	✓ 2.0+		✓	✓ 2.0+	✓		✓ 11+	✓	✓ 12+ (partial)		✓ 11+	
FileSystem API W3C API Virtual FileSystem for persisten storage			✓	✓ 2.0+		✓		✓				✓ 14+			
HTML Media Capture W3C API Taking pictures, record video and audio from an input file type	✓ 6.0+	✓ 3.0+	✓			✓		✓				✓ 14+		✓ 11+	✓
Web Speech API W3C API Speech Recognition and Synthetizer	✓ 7.0+		✓ 32+									14+		11+	
HomeScreen Webapp NO API Add Icon to the home screen with fullscreen support	✓ meta tags		✓ 32+						✓ Anna+		✓ live tile	14+		✓ manifest	✓ manifest

http://mobilehtml5.org/

Web apps with jQuery Mobile

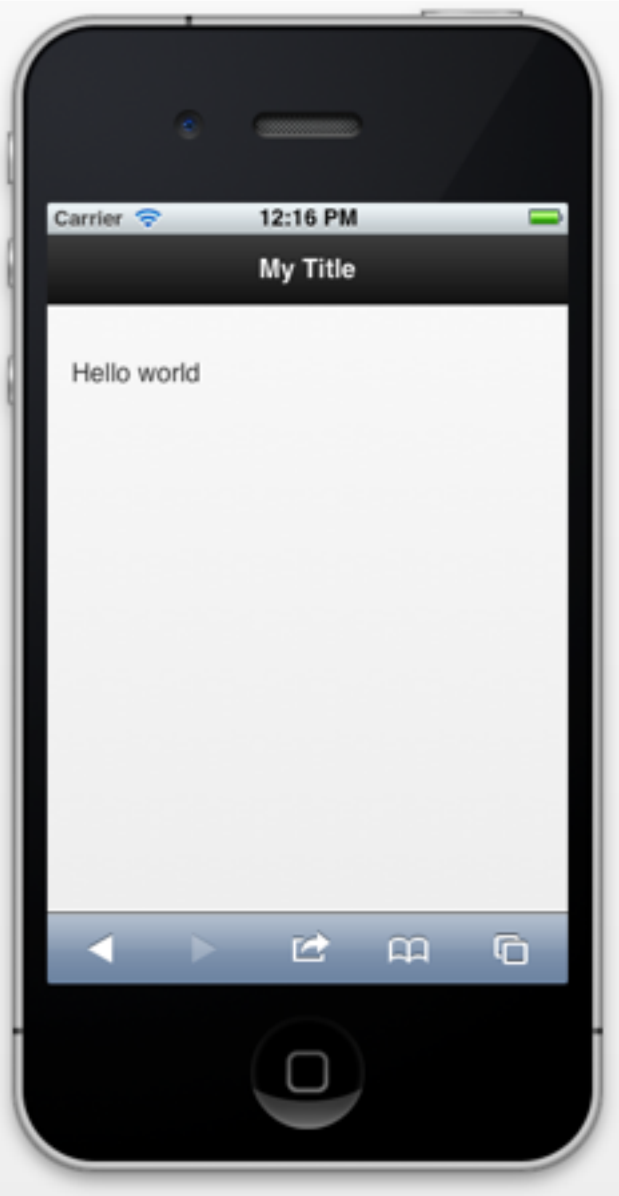
jQuery Mobile is an HTML5-based user interface system for all popular mobile device platforms, built on the rock-solid jQuery and jQuery UI foundation.

Its lightweight code is built with progressive enhancement, and has a flexible, easily themeable design

- Compatible with all major mobile platforms as well as all major desktop browsers, including iOS, Android, Blackberry, Palm, Symbian, Windows Phone 7, and more.
- Built on top of jQuery core so it has a minimal learning curve for people already familiar with jQuery syntax.
- Limited dependencies and lightweight to optimize speed.
- The same underlying codebase will automatically scale to any screen
- Ajax-powered navigation with animated page transitions that provides ability to clean URLs through pushState.

jQuery Mobile page template

```
<!DOCTYPE html>
<html>
    <head>
    <title>My Page</title>
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" href="jquery.mobile.min.css" />
    <script src="jquery.min.js"></script>
    <script src="jquery.mobile.min.js"></script>
</head>
<body>
<div data-role="page">
    <div data-role="header">
        <h1>My Title</h1>
    </div>
    <div data-role="content">
        <p>Hello world</p>
    </div>
</div>
```



My Title

Hello world

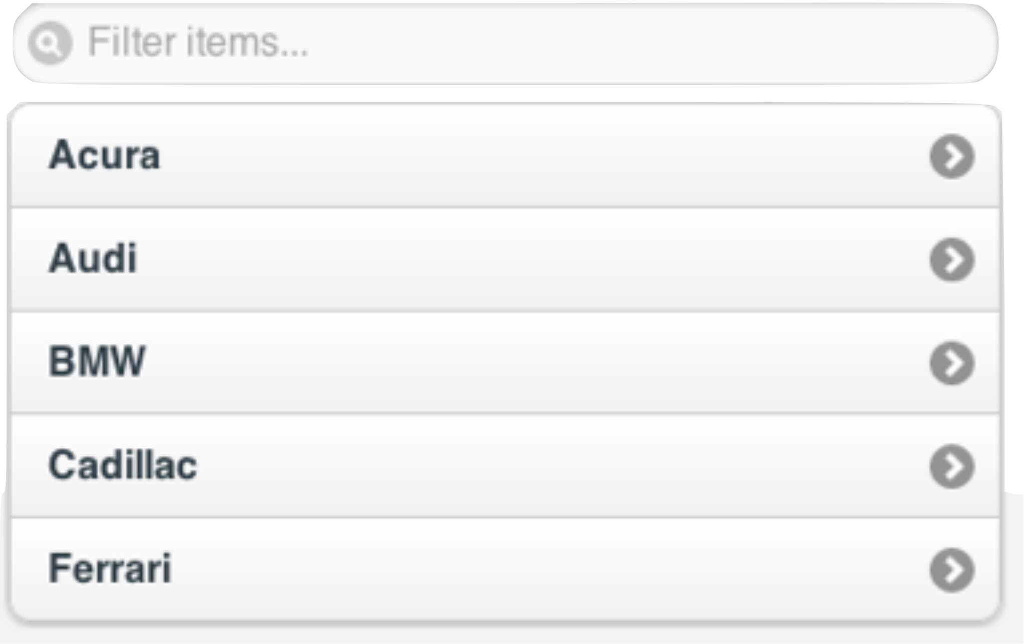
jQuery Mobile: create a listview

jQuery Mobile includes a diverse set of common views that are coded inside the framework.

Here is a simple linked list that has a role of **listview**.

We're going to a dynamic search filter with the **data-filter**="true" attributes.

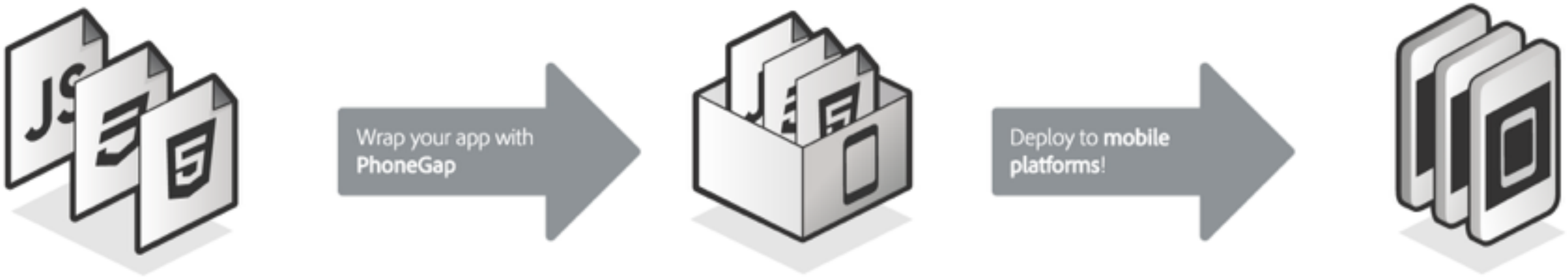
```
<ul data-role="listview" data-filter="true">
    <li><a href="#">Acura</a></li>
    <li><a href="#">Audi</a></li>
    <li><a href="#">BMW</a></li>
    <li><a href="#">Cadillac</a></li>
    <li><a href="#">Ferrari</a></li>
</ul>
```



For documentation and examples: http://demos.jquerymobile.com

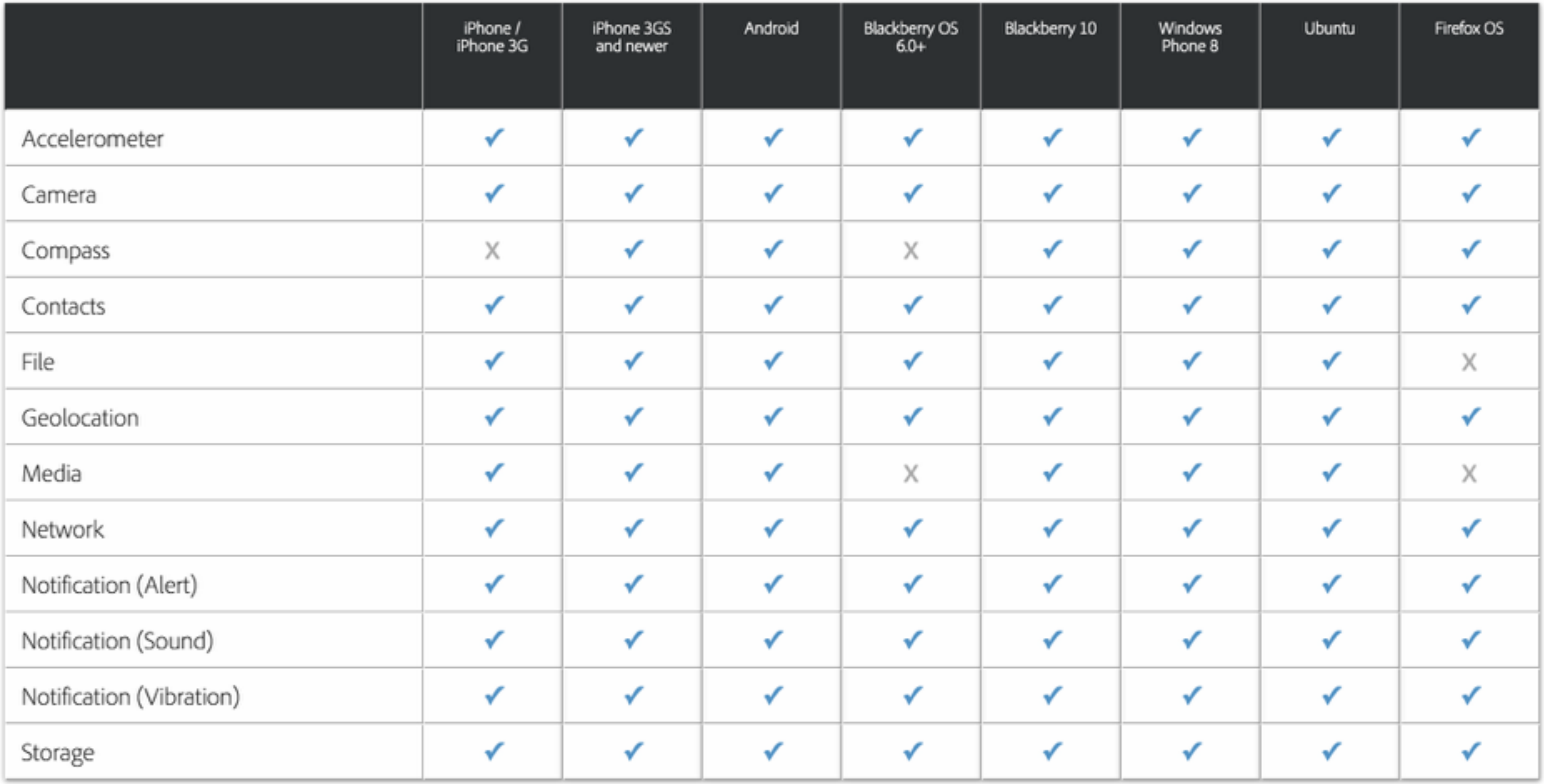
From web to native: PhoneGap

PhoneGap is an open source framework for quickly building cross-platform mobile apps using HTML5, Javascript and CSS.



Official website and docs: **http://phonegap.com**

PhoneGap: supported features



	iPhone / iPhone 3G	iPhone 3GS and newer	Android	Blackberry OS 6.0+	Blackberry 10	Windows Phone 8	Ubuntu	Firefox OS
Accelerometer	✓	✓	✓	✓	✓	✓	✓	✓
Camera	✓	✓	✓	✓	✓	✓	✓	✓
Compass	X	✓	✓	X	✓	✓	✓	✓
Contacts	✓	✓	✓	✓	✓	✓	✓	✓
File	✓	✓	✓	✓	✓	✓	✓	X
Geolocation	✓	✓	✓	✓	✓	✓	✓	✓
Media	✓	✓	✓	X	✓	✓	✓	X
Network	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Alert)	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Sound)	✓	✓	✓	✓	✓	✓	✓	✓
Notification (Vibration)	✓	✓	✓	✓	✓	✓	✓	✓
Storage	✓	✓	✓	✓	✓	✓	✓	✓

PhoneGap plugins

PhoneGap may be extended with plugins that enable the developer to access more device features, including:

- barcode scanning,
- Bluetooth,
- push notifications,
- text to speech,
- calendars,
- Facebook Connect.

Books

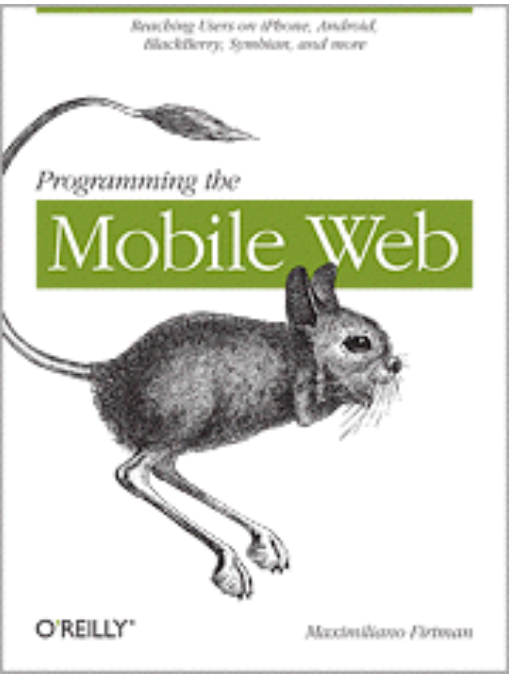
“**Programming the Mobile Web**” - Maximiliano Firtman (O'Reilly Media)

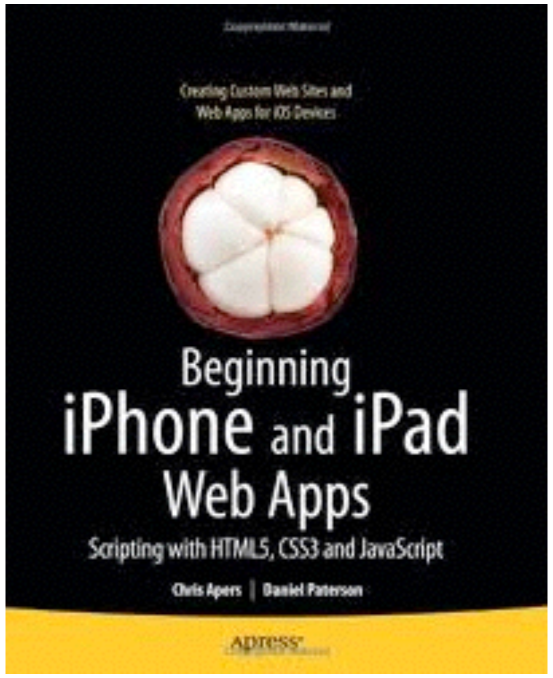
“**Build Mobile Websites and Apps for Smart Devices**” - E. Castledine, et. al. (Sitepoint)

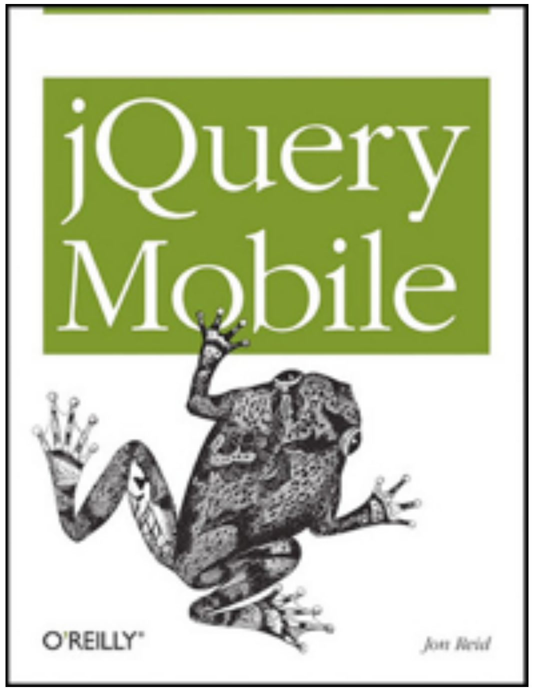
“**Beginning iPhone and iPad Web Apps**” - Chris Apers , Daniel Paterson (Apress)

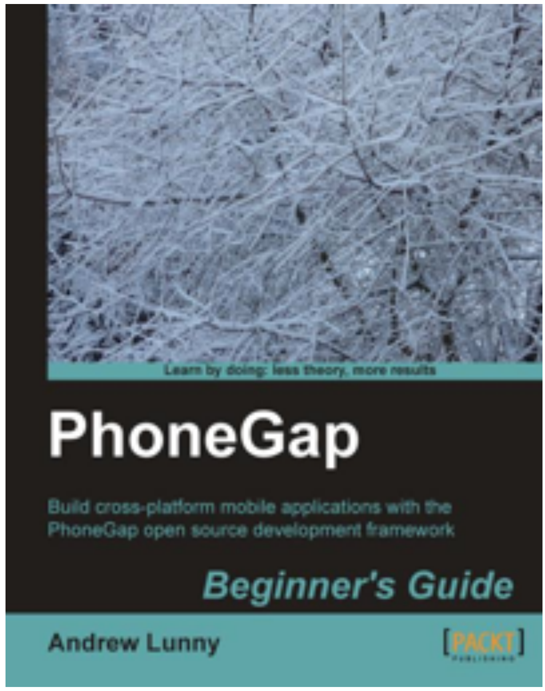
“**jQuery Mobile**” - Jon Reid (O'Reilly Media)

“**PhoneGap Beginner's Guide” -** Andrew Lunny (Packt)


Reaching Users on iPhone, Android, BlackBerry, Symbian, and more
Programming the
Mobile Web
O'REILLY
Maximiliano Firtman


Creating Custom Web Sites and Web Apps for iOS Devices
Beginning
iPhone and iPad
Web Apps
Scripting with HTML5, CSS3 and JavaScript
Chris Apers | Daniel Paterson
Apress




Learn by doing: less theory, more results
PhoneGap
Build cross-platform mobile applications with the PhoneGap open source development framework
Beginner's Guide
Andrew Lunny
[PACKT]