Mobile First Design

Developing Responsive Web Sites

1

Smart Phones vs. Feature Phones

- **Smartphones now outstrip feature phones**
  - Smartphones Accounted for 57.6 Percent of Total Sales in 4th qtr 2013
  - Sales of Android Phones to Approach One Billion in 2014

- Nokia have shelved their Symbian OS
  - Symbian phones and the typical Nokia experience is a feature device

- Smartphones run an OS – they are often hardware programmed
  - Rarely open to a developer community

V Phone Grande
Launched July 2014
in India

2

http://indianexpress.com/article/technology/mobile-tabs/videocon-launches-feature-phone-vphone-grande-at-rs-1950/

What Are Smartphones

- **Smartphones run an operating system**
  - They by the large tend to provide a limited number of physical buttons
  - Enhanced with a touch environment
- **They provide personal computing**
  - High end devices now run 64bit OS
- **They provide an ‘app store’ eco-culture**
  - Allowing developers to build for the device
- **They are littered with amazing sensor capabilities**
  - Bridging the digital divide

The prototype Android handset - 2007

3

Smartphones

- **Smartphones are now the primary global phone type**
  - Worldwide sales of smartphones exceed feature phone sales
- **Device is likely to be operated with one hand**
- **Devices are dropped and bashed about**
- **Incredibly variable specification**

4

Are now the primary global phone type – worldwide sales of smartphones exceed feature phones

Much more limited space to work with

Device is more likely to be operated with one hand – accidental touching and brushing against the screen is much more likely, and even using light sensors and proximity sensors it is going to be difficult to tell if that is a purposeful touch or not.

These devices get dropped and bashed about a lot – how will you take that into account and bear that in mind when designing smartphone applications?

These devices suffer a lot of abuse – sensors start to go a bit awry, buttons start to wear out, the screen gets cracked, the file system might even get so cluttered with the remnants of uninstalled apps that didn't clean up properly that performance takes a massive hit

Many devices have very limited internal memory available

Buttons will eventually wear out or break – does your app still have a way of working that does not rely on a button? (If that button is integral to the way the OS works. i.e. if that button ceases to work the core functionality of the phone is inaccessible or unusable, then you may find it acceptable to design a system where you still rely on that button) – this may change, but in the current market climate where many users will hold on to their phone for two or more years as part of the contract, even though the phone ceases to function optimally after 12 months, then it is your job to make sure that your product is not associated with the ire that they may feel towards their hardware or service provider.

Smartphones – Interactions

- **A different type of user**
  - User is generally on the go
  - They want clear, concise, simple and pain free experiences
    - The Apple ‘it just works’ mantra
  - Get to the point
    - They want minimal form fields and clicks
    - They want concise and optimised information
  - Optimise for the small screen

5

Mobile:

This user is generally on the go, commuting to and from work, waiting at a doctor’s office or perusing Twitter while sitting in a meeting. They are the surface users trying to get quick-hit information like purchasing a Groupon or setting a dinner reservation on OpenTable.

How to Approach them: Offer an experience that is concise, clear and simple. These people are doing a million things at once and capturing their attention is considered a victory.

Quick offers- Provide a pain-free way to purchase or get more information with minimal form fields and clicks required.

Clear messaging- What are they getting out of this? Is the offer different because they are on a mobile device?

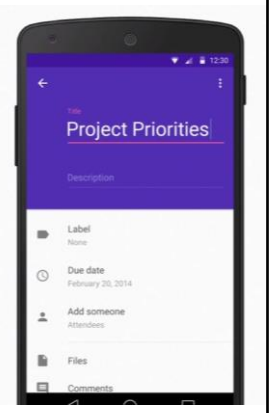
Optimized for the screen- Small screen means less real estate. Be sure to have clear imagery and content that is easily read or able to be zoomed in on with big enough calls to action.

What is an App?

- **An App is a sovereign application running on the smartphone OS**
  - Each mobile OS has its own mobile UI design guidelines
  - These can change rather significantly over time
    - iOS7 and Google Design Material

- **Apps are native programs running on the OS**
  - Access to the full device capabilities
    - Sensors etc…

- **Deployed through an App Store**
  - Must follow design and UX for the platform

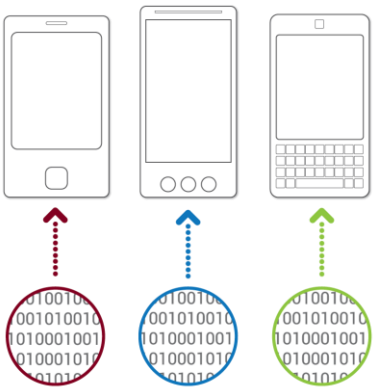


Android Design Material
June 2014

Native Apps

Native apps are written to work on a specific mobile OS

- **PROS:**
  - Faster, more responsive
  - No network connection required
  - Can have icon on home screen
  - Can use phone hardware (camera, accelerometer, compass, etc.)
- **CONS:**
  - Different development skills required for each platform
  - Different, often unclear, requirements for each platform's store
  - Harder to maintain and update app



7

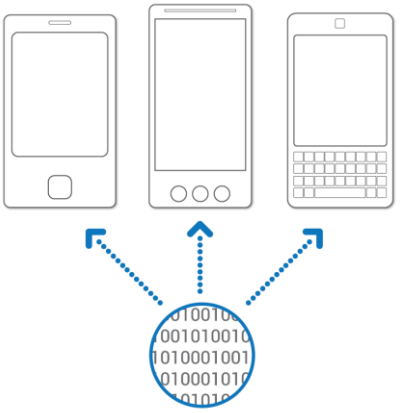
Native applications should be used when you want to gain maximum power from the native embedded code compiles application will run directly on the hardware giving you greatest interaction with the device capabilities including accelerometers, cameras and GPS systems.

However any native app must also be added to the vendors app store, this can be a minefield in its own way. It is possible to run enterprise level applications that can be installed on suitably registered devices.

Apps That Use Embedded HTML

Package HTML / CSS / JS into a native app

- **PROS:**
  - Looks like a native app to the end users
  - Same source code can be used across multiple platforms
  - Works offline
  - Can use phone hardware (compass, camera, accelerometer, etc.)
- **CONS**
  - Slower and less capable than a native app
  - More effort to deploy and maintain than a website
  - Requires users to download, update, etc.



8

Embedded HTML uses frameworks such as PhoneGap. Using HTML CSS and JavaScript the application developer will then leveraged the framework to embed the application. Phone get as assistant libraries for wide variety of devices including iPhone, Android, Windows phone and Blackberry.

The key advantage to this approach is reduced need to maintain separate workflows and teams have different hardware and OS layers. However it is operating through an abstraction layer and lacks the raw power of native applications.

Conceptualising a Smartphone App

- **Interacting with the user:**
  - User activities
    - What will your app do?
    - Who is your app for?
    - How does your app fit in?
    - Where and When will your app be used?
    - What kind of content will be displayed?
  - UI Elements
    - Mobile devices emulate traditional interactions methods
      - E.G. Spinboxes, sliders, switches, keyboards

Discussion – Our mobile app will allow users to select a cinema, film and screening. We need to consider what information is needed from the user and the most effective UI interactions to achieve this.

9

What will your app do?

Think in fairly general terms at first, and don't get into specific features just yet, but do be clear about the purpose of the app and why people will find it useful.

List tasks or operations that the app will perform, or list the controls you imagine on the screen. Alternatively, make a list of a hypothetical user's goals: what are they trying to accomplish with your app? (For example, "send an e-card" or "take a panoramic photo.") Try to arrange this list in order of importance, so that if you need to pare down your feature list later, you'll know what to cut.

When an app is well-focused on a certain task or group of tasks, and those tasks are clear and valuable, the value of the app to the user becomes immediately apparent. It also makes it easier to assign your app a meaningful name and category in an app store.

Concentrating on a few important tasks will also allow you to support them more thoroughly. Make a quick list of the options or attendant functions you'll want to enable in each screen of your app. Does the user have access to everything he or she needs inside your app?

Users may not stick around to learn the ins and outs of a complex app.

Extend your app's clarity of purpose to the title, description, and icon.

Who is your app for?

It's important to imagine who will use your app to better tailor it to their tastes. While there are plenty of devices in the hands of all types of people, keep in mind that apps that appeal to a narrow audience are more likely to go unnoticed in the competitive market for apps.

How does your app fit in?

Find similar apps and compare and contrast them with yours. Reading about competing apps may lead you to make changes to your approach, or spark new ideas the marketplace hasn't yet seen.

If applicable, consider using third-party services, especially social networking and enterprise software services, to help users get friends and content in and out of their phone to the wider web. This allows them to share work and media they've created and consumed on their phone with their network of friends and associates.

Where and when will your app be used?

For example, say your app will be used "at the gym" or "in the car." Evaluate these scenarios for common distractions, and note ways that your app can prevent users from making simple mistakes. The highest-quality apps use design to anticipate user error and compensate for it, even in simple ways; for example, making buttons large and easy to tap, or positioning controls under the user's thumb for more comfortable single-handed use.

What kind of content will be displayed?

Content deserves centre stage in your app, so consider what content your app will highlight, and how users will want to see it presented on a mobile device. E.g.. a weather app might want to present on-screen animations that show the current meteorological conditions in the region.

Interesting and dynamic content can help bring users back to your app time and again. Don’t distract them with too many controls in a given screen. When users are first learning your app, they’ll probably have the impulse to operate most of the controls out of curiosity. There should be few enough controls on the screen that to learn the function and effect of each doesn’t require more than a few moments.

Follow the design advice of making the content the control whenever possible. This means that you don't have separate controls that manipulate content when a gesture can accomplish the same thing. For example, if you have an image of a musical album, you don't need a separate control button to play it. Just a tap on the album image should play it. Look at how images are grabbed and expanded all without the need for any space-wasting controls.

Don’t leave users with the impression that your app has wide areas of functionality that they’ll never explore. Examine navigation, interactions, and controls of other apps. Are there lessons you can learn from their designs? How can you improve upon or revise them to fit your purposes? Sometimes it’s strategically advantageous to ship a simplified version of your app to see how it fares in the market before investing more time and effort in extra features.

Understanding Mobile Devices - iOS

- **All iOS owners start their experience in the same place**
  - This is a ‘5 second learn’ capturing the users attention
  - It imparts some critical experiences
    - The screen is touch based
    - UI elements are touch based
    - The use uses fluid gestures
    - The hardware buttons are secondary to the touch experience

10

Understanding Mobile Devices - iOS

- **iOS has core interface elements referred to as the standard controls**
  - This is a library of buttons, navigation controls, key boards etc…

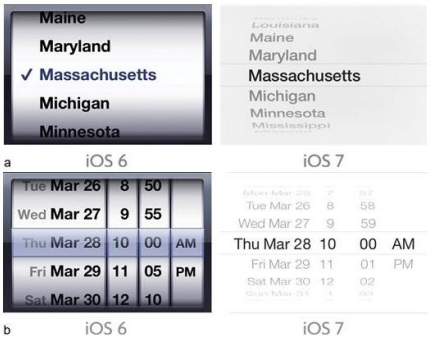
Understanding Mobile Devices - iOS

- **The virtual keyboard is context aware using HTML5 it will provide a better UX**

```
<input type="text"/>

<input type="email"/>

<input type="url"/>

<input type="tel"/>
<input type="
```

Understanding Mobile Devices - iOS

- **We should only ever use standard behaviour for select and date behaviours**
  - The system UI takes control and provides a spinwheel
  - Some web develops like using custom controls
    - These rarely work with mobile UI well



Understanding Mobile Devices - iOS

- **The tab bar is a familiar set of icons to any iOS user**
  - The tab bar appears to the bottom of the screen

iOS 6 iOS 7

- **It provides a consistent UI experience**
  - Clear selected state shown through colour
  - Consistent placement looked to the bottom of the screen
  - Large active touch area
  - Maximum width – 5 icons only further icons shown as more

Understanding Mobile Devices - iOS

- **The Navigation Bar also provides a consistent UI interaction type**
  - Used with descendent level screens to allow back button navigation

1. Button to go back a step or page view
2. Title of the current page/tile
3. Action buttons for current context

Understanding Mobile Devices - iOS

- **The final core control is the Action Menu**
  - It is equivalent to a dialog box in desktop design

Understanding Mobile Devices - Android

- **Welcome to Android – there are over 400 different Android devices**
  - It is an open source OS created by Google
  - Any handset developer can use it for free
  - Each manufacturer usually customises the skin
  - Delivering an inconsistent UI/UX

- **There are a bewildering number of form factors and capabilities**

17

Understanding Mobile Devices - Android

- **Take a look at the following landing screens:**

- **The above are skins for:**
  a) HTC (Sense)
  b) Google
  c) Samsung (TouchWiz)
  d) Motorola

Android – Hardware to Software Interface

- **Most Android phones share some hardware buttons**
  - Back button – as per iOS back nav button
  - Menu button – opens the device settings
  - Search button – opens device search feature
  - Home button – returns to home/landing screen

19

Understanding Mobile Devices - Android

- **Tabs are unique to the Android app UI and a common affordance**
  - Highly customisable for text and icons it has core functionality
    - Hover state
    - Active colour state
  - The Android skin has the default colours and behaviours set
    - Green for HTC
    - Blue for Samsung
  - The tab bar can be placed anywhere on the screen

20

Google Know There is a Problem

- **Jellybean (Android 4) is the first version to add UX/UI guidelines**
  - Brining some consistency to the market
  - The menu button is replaced with an action bar
  - Search is gone
  - A library of design UI elements added

  - http://developer.android.com/design/index.html

Understanding Mobile Devices – Windows Phone

- **Windows repositioned itself with WP7 with a new design paradigm**
  - With Windows 8 the Live Tile UI is seen on every Microsoft device
    - Windows Phone 8
    - Windows 8
    - Xbox One

UX/UI Guidelines

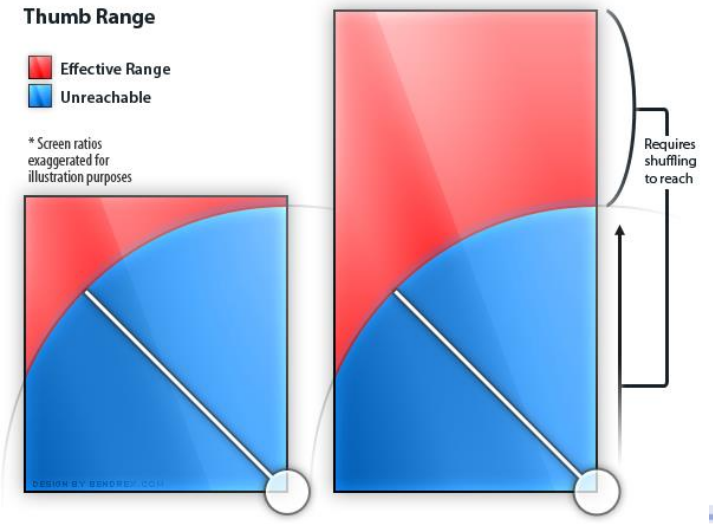
- **Android Jellybean**
  - http://developer.android.com/design/index.html
- **Android Material Design**
  - http://www.google.com/design/spec/material-design/introduction.html
- **iOS**
  - https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/index.html
- **Windows Phone 8**
  - http://devcenter.windowsphone.com/en-us/design
- **Windows 8**
  - http://devcenter.windowsphone.com/en-us/design

23

http://www.mobilexweb.com/blog/ui-guidelines-mobile-tablet-design

Reach

- **Core components such as navigation and core functionality need to consider the reach of the UI and touch ergonomics**



24

http://www.umpcportal.com/2011/08/a-critical-look-at-the-interface-and-ergonomics-of-android/

Exploring Device Differences

Split into groups and consider the different devices available on the market and draw up a list of considerations and solutions for the various screen sizes available.

Share you findings with the class.

In your groups, discuss the implications of different platforms on any design decisions you might make

Share you findings with the class.

25

Smartphone Design Guidelines (1)

- **Functionality**
  - Prioritise and present core features that have special relevance in a mobile environment
  - Offer relevant mobile-only functionality (like barcode scanning and image recognition), and enhance functionality using the capabilities of mobile devices where possible
  - Ensure that fundamental features and content are optimized for mobile
  - Include features that are relevant to the business category
    - E.G. In retail – orders, checkout etc…
  - Offer key capabilities across all channels – phone, tablet and desktop

www.fastcompany.com/1816610/sharing-app-bump-30-slashes-most-features-proves-less-really-can-be-more

26

Mobile users and mobile usage are growing. With more users doing more on mobile, the spotlight is on how to improve the individual elements that together create the mobile user experience.

The mobile user experience encompasses the user's perceptions and feelings before, during and after their interaction with your mobile presence — be it through a browser or an app — using a mobile device that could lie anywhere on the continuum from low-end feature phone to high-definition tablet.

Creating mobile user experiences that delight users forces us to rethink a lot of what we have taken for granted so far with desktop design. It is complicated in part by mobile-specific considerations that go hand in hand with small screens, wide variations in device features, constraints in usage and connectivity, and the hard-to-identify-but-ever-changing mobile context.

Dissecting the mobile user experience into its key components gives us a conceptual framework for building and evaluating good mobile experiences, within the context of a user-centred approach to designing for mobile. These components shape the mobile user experience — including functionality, context, user input, content and marketing, among others.

The relevance of these elements will change depending on the type of device (feature phone versus smartphone versus tablet) and the presentation interface

(app versus Web). This article briefly describes each of these elements and elaborates on each with selected guidelines.

Functionality

This has to do with tools and features that enable users to complete tasks and achieve their goals.

GUIDELINES

- Prioritize and present core features from other channels that have special relevance in a mobile environment. For an airline, this includes flight statuses and flight check-ins. For an online store, it includes supporting in-store shopping via easy access to product reviews on mobile devices.
- Offer relevant mobile-only functionality (like barcode scanning and image recognition), and enhance functionality using the capabilities of mobile devices where possible to engage and delight users.
- Ensure that fundamental features and content are optimized for mobile. For example, make sure the store locator shows the nearest stores based on the device's location, and make the phone numbers click-to-call.
- Include features that are relevant to the business category. For retail websites and apps, this would include product search, order status and shopping cart.
- Offer key capabilities across all channels. Users who sign in should see their personalized settings, irrespective of the device or channel being used. If certain functionality is not offered on mobile, then direct users to the appropriate channel.

Further reading: http://www.fastcompany.com/1816610/sharing-app-bump-30-slashes-most-features-proves-less-really-can-be-more

Smartphone Design Guidelines (2)

- **Information Architecture**

- Present links to the main features and content on the landing page, prioritized according to the user's needs
- Use concise, clear, consistent and descriptive labels for navigation items and links

- Enable mobile users to navigate to the most important content and functionality in as few taps or key presses as possible
- Address the navigation needs of both touchscreen and non-touchscreen users

- Provide navigational cues to let users know where they are, how to get back and how to jump back to the start

27

Information Architecture

This has to do with arranging the functionality and content into a logical structure to help users find information and complete tasks. This includes navigation, search and labelling.

GUIDELINES

- Present links to the main features and content on the landing page, prioritized according to the user's needs.
- Enable mobile users to navigate to the most important content and functionality in as few taps or key presses as possible. Navigation optimized for small screens is usually broad and shallow instead of deep. While three clicks (or taps) is not the magic number, users need to be able to recognize that each tap is helping them complete their task. Every additional level also means more taps, more waiting for a page to load and more bandwidth consumed.
- Address the navigation needs of both touchscreen and non-touchscreen users. When designing for touch, make sure the tap size of the navigation item is at least 30 pixels wide or tall. Provide keypad shortcuts for feature phones, so that users can enter, say, a number (0 to 9) to quickly access a link:
- Provide navigational cues to let users know where they are, how to get back and how to jump back to the start. Mobile breadcrumbs are often implemented by replacing the "Back" button with a label showing users the section or category that they came from. For mobile websites, use standard conventions, such as a home icon that links back to the start screen,

especially when navigation is not repeated on every screen.

- Use concise, clear, consistent and descriptive labels for navigation items and links. While always a good practice, it becomes even more important on tiny mobile devices.

Smartphone Design Guidelines (3)

- **Visual Design**

  - Mobile design should not just rehash the desktop design.
    - “Mobilize, don’t miniaturize”
    - “Don’t shrink, rethink”
    - Go Mobile First!

  - Design for glanceability and quick scanning

  - Maintain visual consistency with other touch points and experiences through the use of colour, typography and personality
  - Guide users from the initial and most prominent element of the design to other elements to help them complete their tasks
  - Consider both portrait and landscape orientations in the design process

28

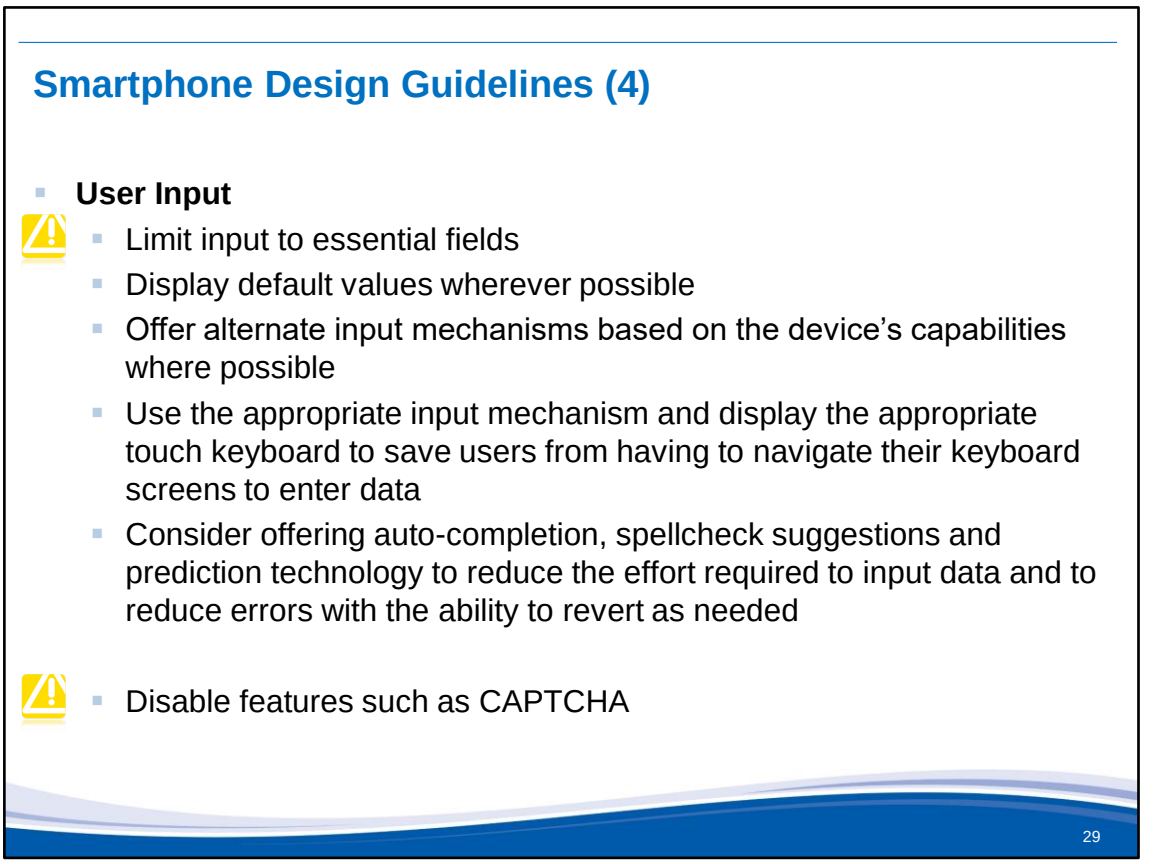
Visual Design

This has to do with the visual presentation and interactive experience of mobile, including graphic design, branding and layout.

GUIDELINES

- Remember the sayings “Mobilize, don’t miniaturize” (popularized by Barbara Ballard) and “Don’t shrink, rethink” (of Nokia). Both make the point that mobile design should not just rehash the desktop design.
- Design for glance ability and quick scanning. Glanceability refers to how quickly and easily the visual design conveys information.
- Maintain visual consistency with other touch points and experiences (mobile, app, Web, print and real world) through the use of colour, typography and personality. Identifying an Amazon web page is easy enough, even when the brand name is not visible.
- Guide users from the initial and most prominent element of the design to other elements to help them complete their tasks. This is known as visual flow. A good design brings together visual elements as well as information architecture, content and functionality to convey the brand’s identity and guide the user.
- Consider both portrait and landscape orientations in the design process. Devices increasingly support multiple orientations and automatically adjust to match their physical orientation. Maintain the user’s location on the page when they change orientation. Indicate additional or different functionality in

the new orientation if applicable



User Input

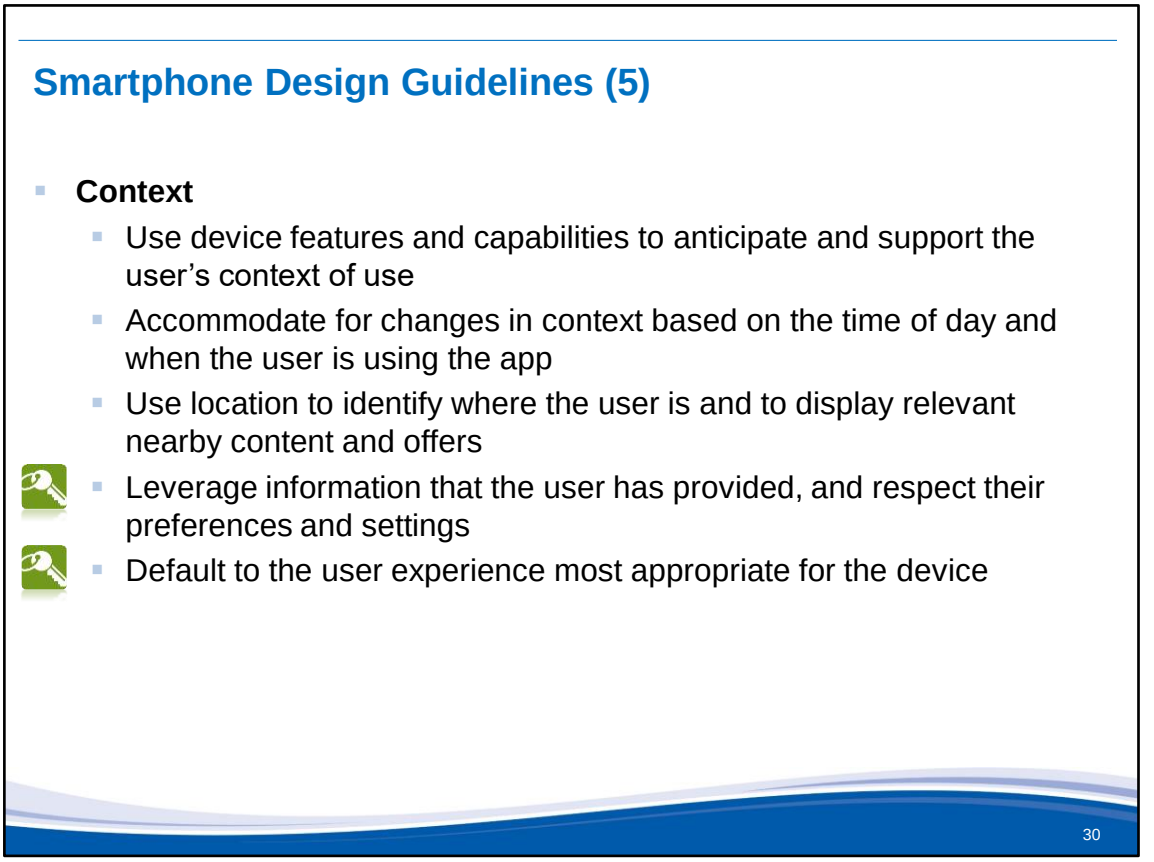
This has to do with the effort required to enter data, which should be minimized on mobile devices and not require the use of both hands.

GUIDELINES

- Limit input to essential fields. Or, as Luke Wroblewski says in his book Mobile First, "When it comes to mobile forms, be brutally efficient and trim, trim, trim." Limit registration forms to the minimum fields required, and use shorter alternatives where possible, such as a postcode instead of city and state.
- Display default values wherever possible. This could be the last item selected by the user (such as an airport or train station) or the most frequently selected item (such as today's date when checking a flight's status)
- Offer alternate input mechanisms based on the device's capabilities where possible. Apps take advantage of quite a few input mechanisms built into devices, including motion, camera, gyroscope and voice, but mobile websites are just starting to use some of these features, particularly geolocation.
- Use the appropriate input mechanism and display the appropriate touch keyboard to save users from having to navigate their keyboard screens to enter data. Keep in mind that inputting data is more tedious on feature phones that have only a numeric keypad. For non-sensitive applications, allow users to stay signed in on their mobile device; and save information such as email address and user name because mobile phones tend to be personal devices, unlike tablets, which tend to be shared between multiple people.
- Consider offering auto-completion, spellcheck suggestions and prediction

technology to reduce the effort required to input data and to reduce errors — with the ability to revert as needed.

- Disable features such as CAPTCHA where not appropriate.



Mobile Context

A mobile device can be used at anytime, anywhere. The mobile context is about the environment and circumstances of usage — anything that affects the interaction between the user and the interface, which is especially important for mobile because the context can change constantly and rapidly. While we often focus on distractions, multitasking, motion, low lighting conditions and poor connectivity, it also includes the other extreme — think using a tablet in a relaxed setting over a fast Wi-Fi connection.

GUIDELINES

- Use device features and capabilities to anticipate and support the user's context of use. The iCookbook app allows users to walk through a recipe using voice commands — a nice feature when your hands are covered in batter!
- Accommodate for changes in context based on the time of day and when the user is using the app. A GPS app should automatically switch from day to night mode, showing low-glare maps for safer night-time driving.
- Use location to identify where the user is and to display relevant nearby content and offers. A Google search for "movies" on a mobile device brings up movies playing nearby and that day's show times, with links to buy tickets online if available.
- Leverage information that the user has provided, and respect their preferences and settings. After the first leg of a multi-leg flight, TripIt shows the flight and gate information for the next flight, as well as how much time you have.

- Default to the user experience most appropriate for the device (i.e. a mobile experience for small screens, and perhaps a desktop-like experience for tablets), but give users the option to have enhanced features. A big discussion on how to present this to the user recently took place, with Jakob Nielsen recommending a separate mobile website and Josh Clark arguing instead for a responsive design; yet others believe that Nielsen and Clark are both wrong.

Smartphone Design Guidelines (6)

- **Usability**
  - Make it clear to the user what can be selected, tapped or swiped (affordance), especially on touchscreen devices
  - For touchscreen devices, ensure that touch targets are appropriately sized and well spaced to avoid selection errors
  - Follow conventions and patterns to reduce the learning curve for users and to make the mobile experience more “intuitive”
  - Ensure usability in variable conditions
  - Do not rely on technology that is not universally supported by your audience’s devices

31

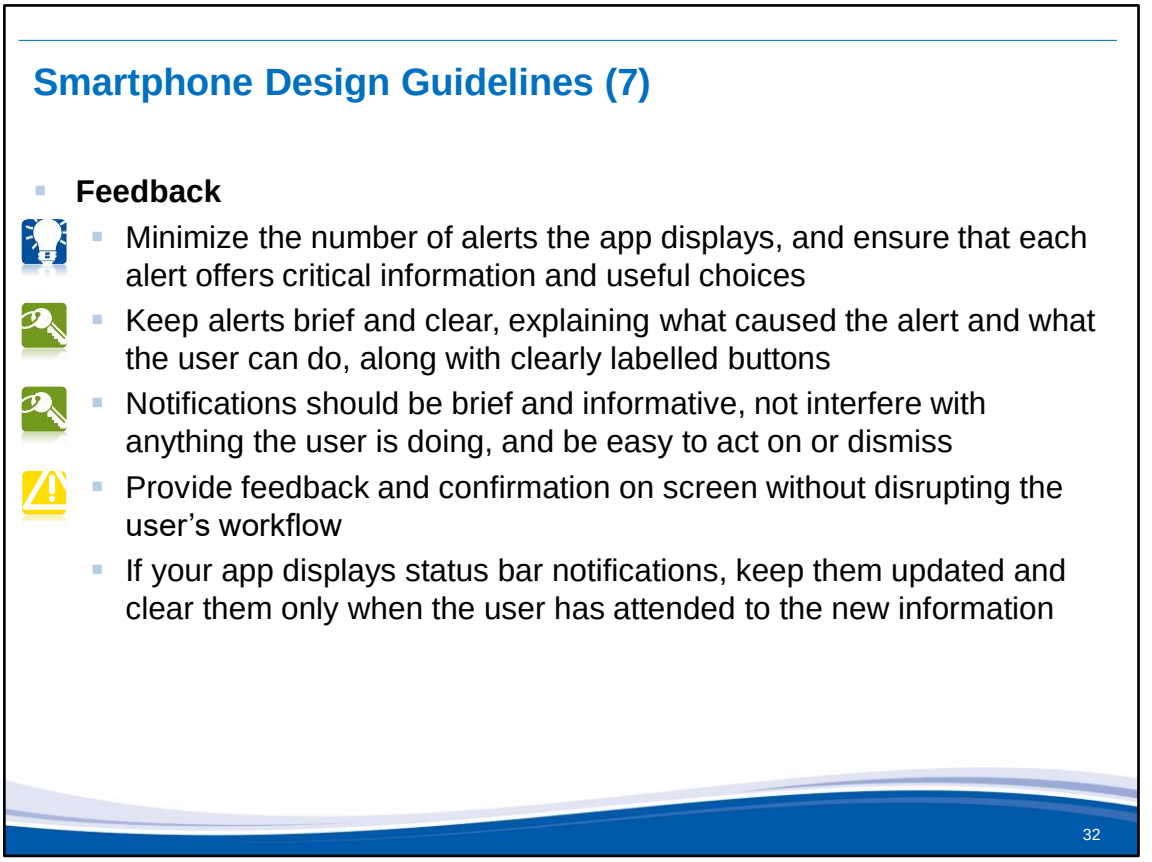
Usability

This is the overall measure of how well the information architecture, design, content and other elements work together to enable users to accomplish their goals.

GUIDELINES

- Make it clear to the user what can be selected, tapped or swiped (this is known as affordance), especially on touchscreen devices. One of the big findings of Nielsen Norman Group’s usability studies of the iPad was that users didn’t know what was touchable or tappable. Another issue was swipe ambiguity: when the same swipe gesture means different things in different areas of a screen. Ensure that touchability is clear and that items such as links, icons and buttons are visibly tappable.
- For touchscreen devices, ensure that touch targets are appropriately sized and well spaced to avoid selection errors. Also, place touch targets in the appropriate screen zones; for example, put destructive actions such as those for deletion in the “Reach” zone, where the user has to actively reach.
- Follow conventions and patterns to reduce the learning curve for users and to make the mobile experience more intuitive. Dedicated apps should follow platform-specific standards and guidelines.
- Ensure usability in variable conditions, including for daylight glare and changed angle of viewing and orientation, by paying attention to design elements like contrast, colour, typography and font size.
- Do not rely on technology that is not universally supported by your audience’s devices, including Java, JavaScript, cookies, Flash, frames, pop-ups and

auto-refreshing. When opening new windows or transitioning from an app to the browser, warn users to avoid overwriting already open tabs.



Feedback

This has to do with the methods for attracting the user's attention and displaying important information.

GUIDELINES

- Minimize the number of alerts the app displays, and ensure that each alert offers critical information and useful choices.
- Keep alerts brief and clear, explaining what caused the alert and what the user can do, along with clearly labelled buttons.
- Notifications should be brief and informative, not interfere with anything the user is doing, and be easy to act on or dismiss.
- Provide feedback and confirmation on screen without disrupting the user's workflow.
- If your app displays badges and status bar notifications, keep the badges updated and clear them only when the user has attended to the new information. If you clear the notifications badge for a mobile app the moment the user visits the notification section, even before the user has seen which of their multiple accounts triggered the badge, it forces them to hunt through each account to see what triggered it.

Smartphone Design Guidelines (8)

- **Help**

  - Make it easy for users to access help and support options
  - Offer multiple ways to get support, including options relevant in a mobile context
  - Present a quick introduction and short tutorial on using the app when it first launches, with options for the user to skip and view later
  - When introducing new or unique functionality offer contextual help and tips to guide users the first time, and as a refresher for infrequently used functionality

33

Help

This relates to the options, products and services that are available to assist the user in using the website or app.

GUIDELINES

- Make it easy for users to access help and support options. Users commonly look for help in the footer of a mobile website and in the toolbar or tab bar of an app.
- Offer multiple ways to get support, including options relevant in a mobile context, such as self-serve FAQs, live support via click-to-call, and near-real-time Direct Message tweets.
- Present a quick introduction and short tutorial on using the app when it first launches, with options for the user to skip and view later.
- When introducing new or unique functionality (such as when check depositing via mobile apps was first introduced), offer contextual help and tips to guide users the first time, and as a refresher for infrequently used functionality.
- Offer help videos when appropriate, but allow the user to start, pause, stop and control the volume as they wish, and keep in mind the multimedia guidelines mentioned in the “Content” section previously.

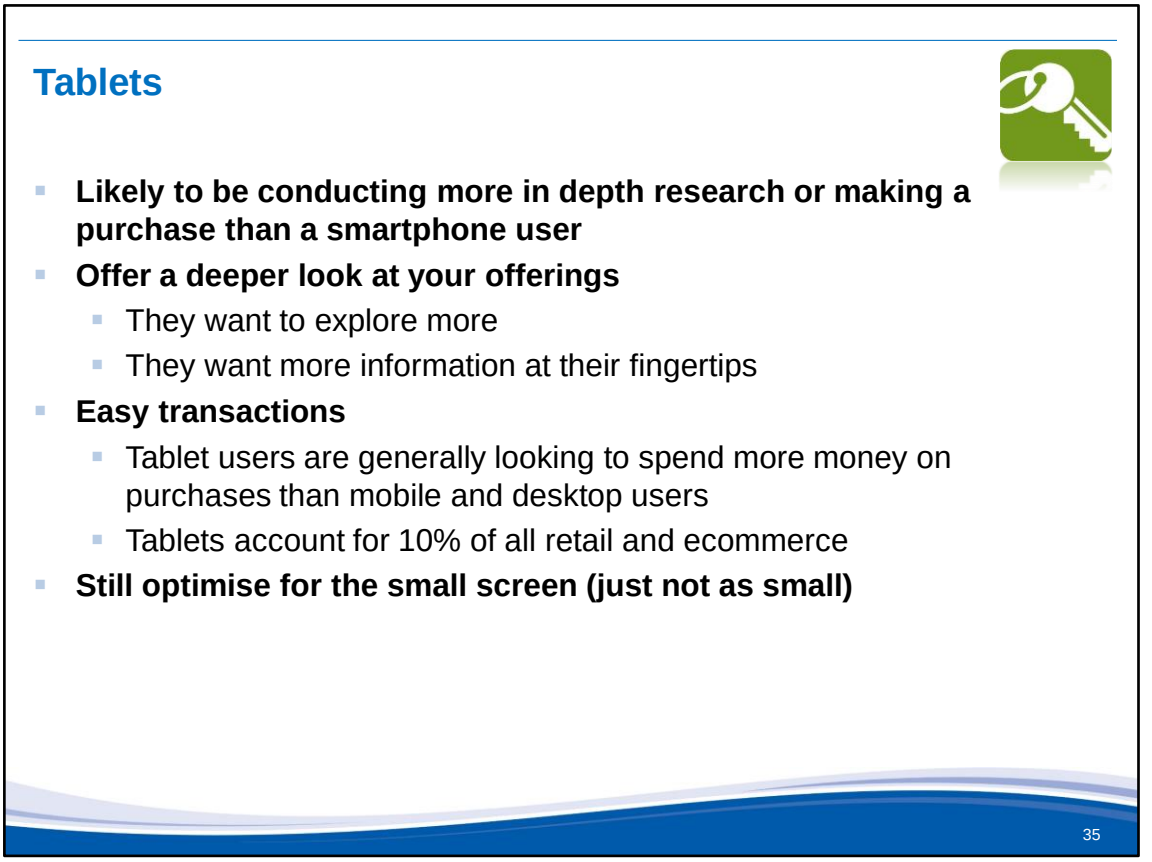
Tablets

- **A single piece mobile computer**
- **Typically uses a touchscreen**
- **Often has supplemental inputs either built in or as peripherals**
- **Used on the move, they tend to be employed for surfing the web and light production work**

34

A tablet is a single-piece mobile computer. Devices typically have a touchscreen, with finger or stylus gestures replacing the mouse. It is often supplemented by physical buttons or input from sensors such as accelerometers. An on-screen, hide able virtual keyboard is usually used for typing. Tablets differentiate themselves by being larger than smart phones or personal digital assistants.

Being used on the move, they tend to have enough power and battery life for surfing the web and light production work, but are not particularly well suited to heavy content production. Some devices bridge the gap between a tablet and a personal computer by having docking stations that allow (or include) keyboards and other peripherals to be attached, or allowing keyboards to be attached and used on the move (e.g. the MS Surface has a keyboard integrated into the cover)



Tablet:

Today, for most brands, the tablet has quickly become the holy grail of customer communication. Brands that can connect with consumers at this touchpoint via a digital catalogue or shopping cart can capitalize on this growing behaviour. What could be better than a slick device that fits perfectly in your lap while you sit on your couch catching up on the latest episode of Mad Men? This trend is quickly becoming referred to as "couch commerce" and "multi-screen experience" as people no longer solely watch TV while unwinding after work.

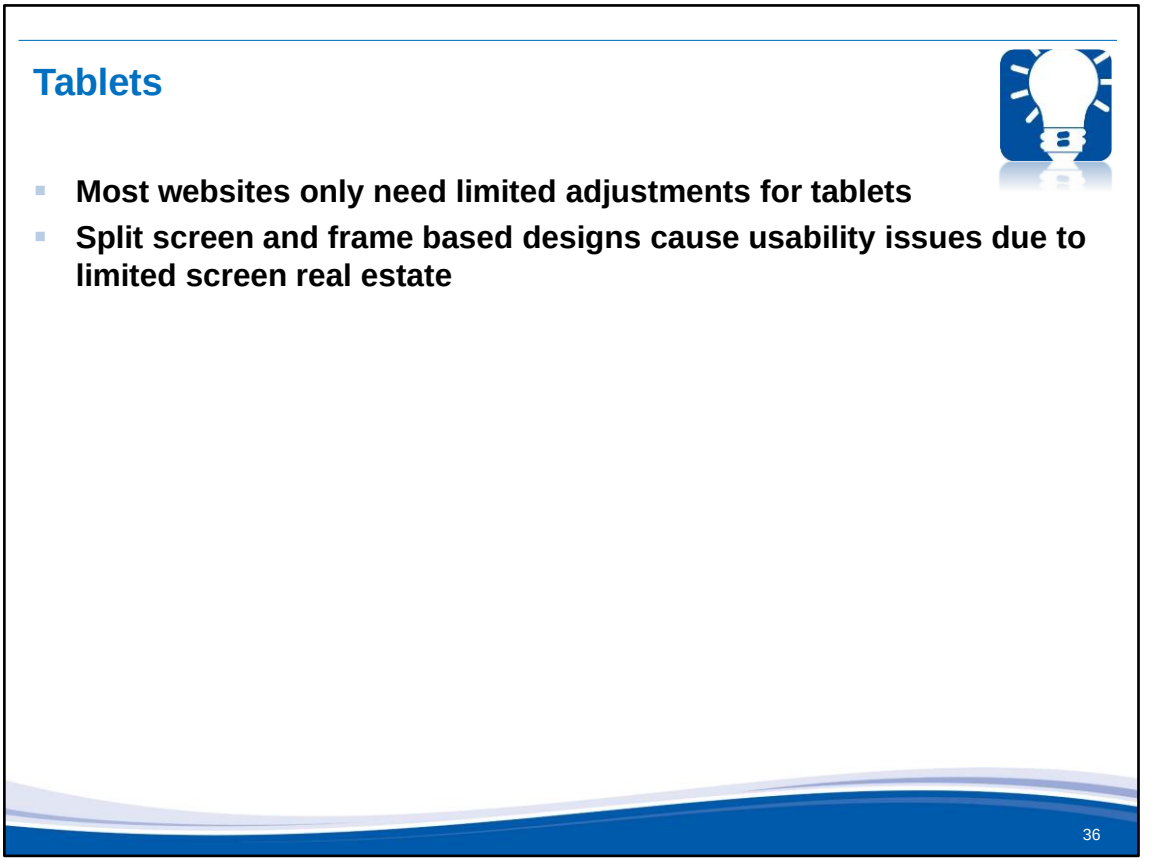
How to approach them: These users are craving a more in depth visit and are most likely conducting research or making a purchase.

Offer more- You can demand that these users take a deeper look at your offering. They want to explore more and want more information at their fingertips.

Easy to transact- These users are spending 54% more on their purchases than their mobile counterparts and 19% more than those on desktops. Encourage them and make it easy for them to buy. Right now tablets are the preferred channel for retail and ecommerce at around 10%, with mobile trailing at around 7%.

Clean content- The screen is still relatively small. No one ever complimented the iPad for offering them the best super bowl watching experience- so keep that in mind. Continue to keep the content clean, uncluttered, with clear calls to action. You want to make sure your visitors are clear on what they should be doing whether its clicking on a link, previewing a video or signing up for something.

End Result?



Tablet Usability

Summary: Flat design and improperly rescaled design are the main threats to tablet usability, followed by poor gestures and workflow.

Most websites are fairly usable on tablets and need only limited adjustments to suit this environment. (In contrast, using websites on mobile phones requires many more design changes to accommodate the smaller screens.)

When people are asked how they use their tablets, web browsing was universally mentioned as a top activity.

Although tablet-specific applications have plenty of usability flaws, the problems are mainly the same as those that plague traditional application design: difficult features, a mismatch with user workflow, and poor instructions that people don't read.

Designing and building any high-usability application involves substantial work, and tablet apps entail a few additional issues, including the need to modify the user interface for different tablet models. This, combined with the popularity and ease of using websites on tablets, begs the question of why companies would have a tablet app in the first place. It is usually better to stick to designing a website and invest the resources in improving web usability, which still suffers

badly in most companies.

Build a tablet app only if you can offer value-added functionality over a website, such as creating an app that is focused on supporting a single main task.

In any case, don't make your tablet app a scaled-up phone app. We've seen hundreds of apps (mainly on Android) that misuse screen space by offering tablet users the same basic design as phone users.

Revenge of the Frames

Frames are rarely seen today; improved design techniques — such as inline frames and parallax scrolling — meet similar goals with better usability.

But frames-like concepts cause usability problems in many modern tablet designs. Two common problems are split-screen designs and temporary frames for search results and the like.

Although a tablet seems big compared to a phone, it's still a small screen and typically shouldn't be subdivided into smaller frames or split views, except when users really need to access two types of information simultaneously. Every time you split off part of the screen, less remains to show content.

Web UX Bleed through

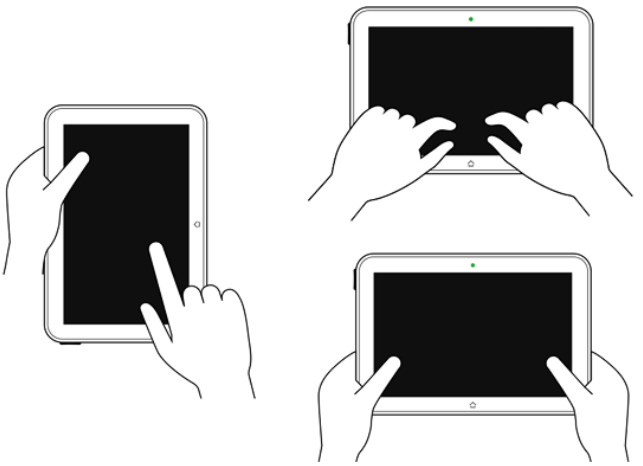
Given the web's dominance in computer use these days, it's not surprising that we found concepts from the web user experience bleeding through the platform divide and influencing people's use of tablet apps. Key examples here include search dominance and heavy reliance on the Back button.

Users frequently want to search on tablets; they also want to return to their search results. Unfortunately, many apps don't provide a proper SERP (search engine results page) as a primary navigation object that users can easily return to. Instead, search results are shown in one of those zombie-attack frames with a fleeting screen presence.

The Back button has long been the user's lifeline on the web; if anything, it's even more important on tablets, where accidental activation is a common

consequence of the touchscreen interface. Unfortunately, even with apps that offer a Back button, the feature is sometimes hard to find, while other times it doesn’t undo the user’s last action as expected.

Reach and Interaction points with Tablets



Differences between Tablets and Phones

Split into groups and discuss what you believe the differences between a the usage patterns and activities of phone and tablet users to be.

Share you findings with the class.

In your groups, discuss the design standards that those users will expect. Where are they the same, and how are they different from each other?

Share you findings with the class.

38

Behavioural differences between tablets and phones

- Usage patterns and activities among mobile users
- Design standards that people expect

Filter at the Server

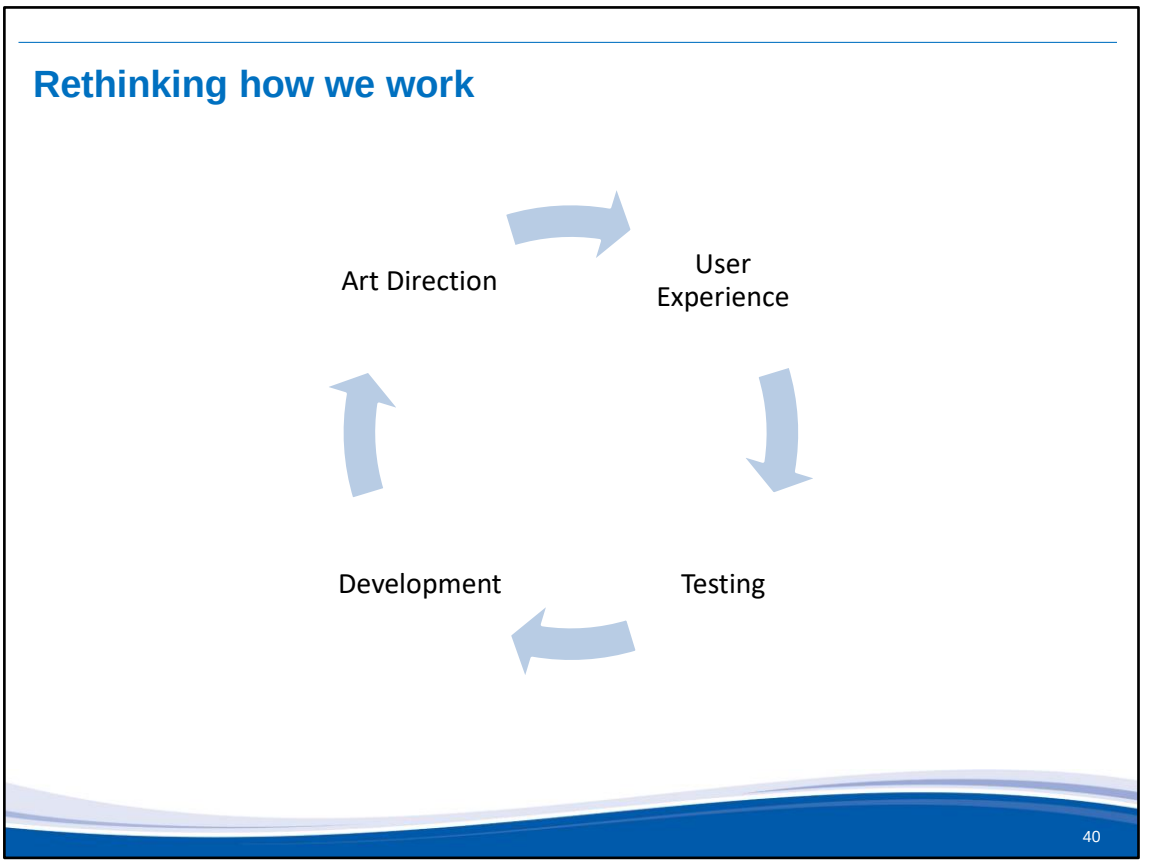
Two separate websites, possibly on separate domains. Good approach for web apps.

- **PROS:**
  - Easier to code / separation of concerns
  - Content can be completely different
  - Interaction style can be different (e.g. touch vs. mouse)
- **CONS:**
  - How to reuse content across both sites?
  - What about intermediate-sized devices?
  - Probably need to automatically redirect users
  - What if the user doesn't want to be redirected?

39

Where an established server orientated architecture such as ASP.NET exists it may not be immediately possible to the utilise mobile first response design. By creating a separate mobile ready website you could filter at the server. This allows you to serve device optimised website.

The key problems with this approach is of the mobile website is usually added as a extension or afterthought of the desktop version. Managing to different workflows is challenging as is ensuring that new devices and new mobile filters are also dealt with on the server.



The UI developer has a significant change of skillset in the basis of modern development lifecycle. With mobile first design the emphasis shifts consider the information and the user experience over pixel perfect creative asset is primary UI goal.

By focusing on adaptivness for fluid layout we gain:

A layout that adapts to screen size – not device

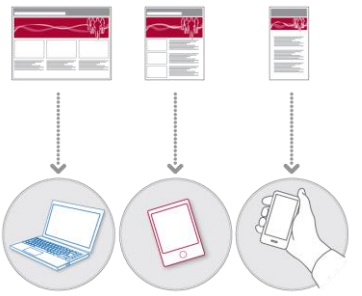
Setting load time as a constraint for development

Adding the cross browser support in a progressive enhancement approach

Responsive Design

Create a single site that re-styles itself to suit the device

- **PROS:**
  - Mobile-first development demands a focus on key use cases
  - Progressive enhancement will benefit all users, not just mobile
  - Relatively future-proof
  - Much less duplication of content
- CONS:
  - A good multi-device experience requires excellent javascript and CSS skills
  - Being efficient for mobile bandwidth requires discipline
  - Downloading large pages, images, scripts, CSS, etc.
  - Require some kind of fallback / poly-fill for older browsers



41

Responsive design involves a website that carries the HTML CSS and JavaScript behaviour to filter at the client. To work responsively involves rethinking from scratch the way the website UI will work. It potentially involves wasted data arriving at the client as you try to solve all possible device proportions.

Going Responsive – Two strategies

- **On a static page mobile devices are challenged**
  - A small device 320x480 shows 20% of a page
  - With 80% needing to be reached via pinch and zoom
  - The content is not optimised for mobile bandwidth
    - Delays kill user interest!

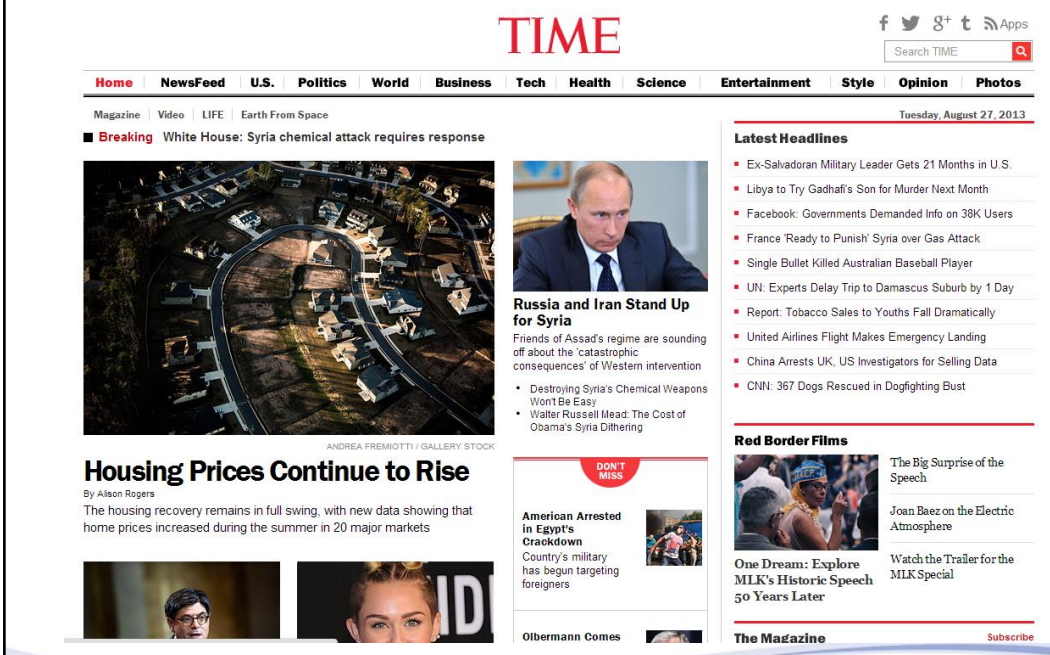
- **There are two strategies for a responsive website either:**
  - To refactor existing solutions progressively to use HTML5 & CSS3
    - Build from 1024x768 down to smaller device settings
  - To rethink development from a mobile first perspective
    - Rebuilding from the smallest device up
    - Considering the information first and presenting it differently

42

A great deal of discussion/argument/fisticuffs is over building responsive web designs or to simply go forward with separate (mobile and desktop) site designs. With this battle there is no right answer here. Because we are looking at the notion of Mobile First design rather than separate vs. responsive web design.

Although the factoring down has been applied successfully there are numerous horror stories littering the web. Deciding what information should survive

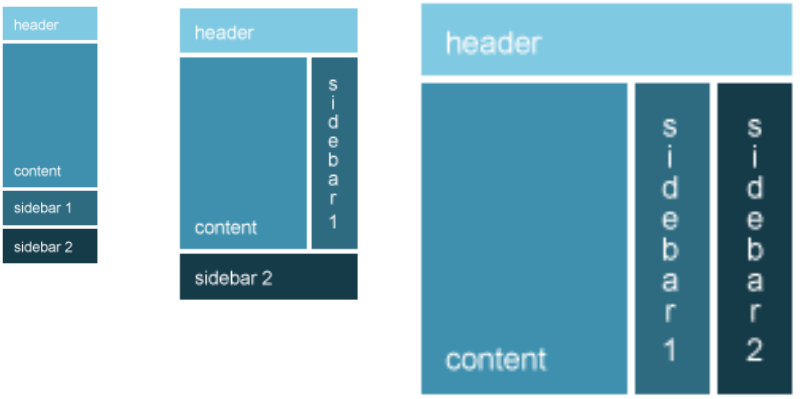
A Mobile First Example – Time Magazine



Time magazine is an excellent example of a mobile first design – the website makes the most of each device proportion and uses some important mobile first design practices restructuring navigation and essential content. The mobile first principal builds on stacks of content, the tablet shifts to a double column layout and the desktop uses a full navigation maximal width multi column page layout like we created earlier.

Responsive is not just smaller text

- **Responsive development is about re-envisioning content**



44

Response design start of the concept of mobile first which means you start with the smallest device and consider the call of your website. You build from this smallest device adding new content and taking advantage of the larger website as you develop.

Working from the mobile first perspective we must consider what information is call essential to the site.

Mobile First and Responsive Design

- **Web design and UI development grew out of DTP concepts**
  - The web has evolved beyond the desktop screen
  - Now a multi-device, multi-platform world

- **This involves a shift in design methodology and paradigm**
  - Mobile-first development demands a focus on key use cases
    - Placing the Mobile experience as a key consideration
  - Responsive development places UI layout in the hand of the client
    - Progressive enhancement will benefit all users, not just mobile
    - Uses CSS3 media queries and is relatively future-proof
    - Much less duplication of content – but still to much!
    - Aim is to create efficient content for mobile bandwidth
      - This requires discipline

45

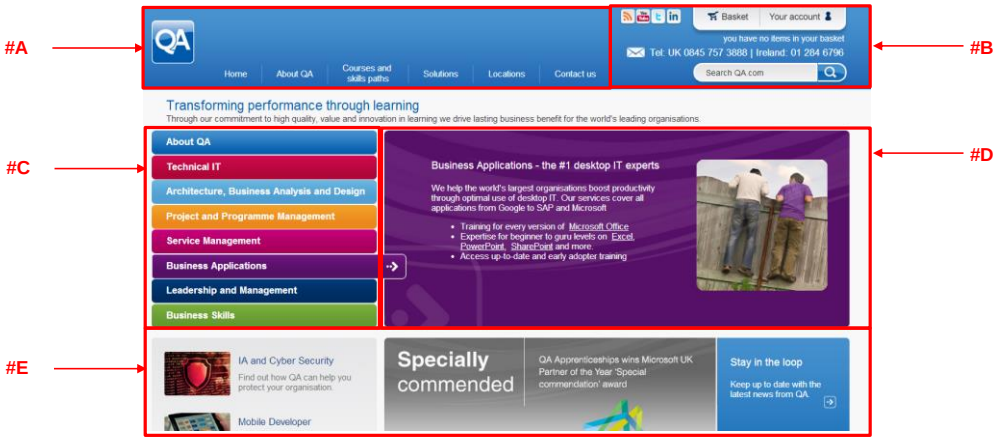
Web UI development evolved out of traditional desktop publishing and print methodologies. In this brave new world of mobile development concepts such as above the fold and fixed screen resolutions become a thing of the past. There are many different methodologies emerging as a way of supporting mobile devices this course will focus on utilising the principles of mobile first and responsive design.

Mobile first places the mobile experience as an equal partner or perhaps the more important partner in the UI design thinking for web project. Instead of the painful swiping across the screen zooming in and out web browser experience of the early days of smart phone technology we build a website to capitalise on the features resolutions and capabilities of mobile users. This is not simply building smaller websites it's about thinking how we reach mobile users what data they want while the interact with this on a mobile device and ensuring the rain which media capabilities supported in the same way you would consider doing so on a desktop device.

Responsive web design is away of implementing a mobile first methodology by using CSS three we can create media queries media queries allows the client to decide how to implement the UI layout. This is an extremely useful methodology meaning that all the CSS information necessary to support potentially multiple layouts is included on the initial visit to the web server to the user was to switch orientation of the device the web page can adapt without needing to go back to the server for more information.

Focusing on Information

- **Consider the following Desktop layout**
  - We need to consider the areas of intent and how to present them



The Fluid Grid

A flexible grid that allows content to be structured consistently and predictably, but also allows for the size and positioning of the elements to change depending on the available size of the rendering device's display

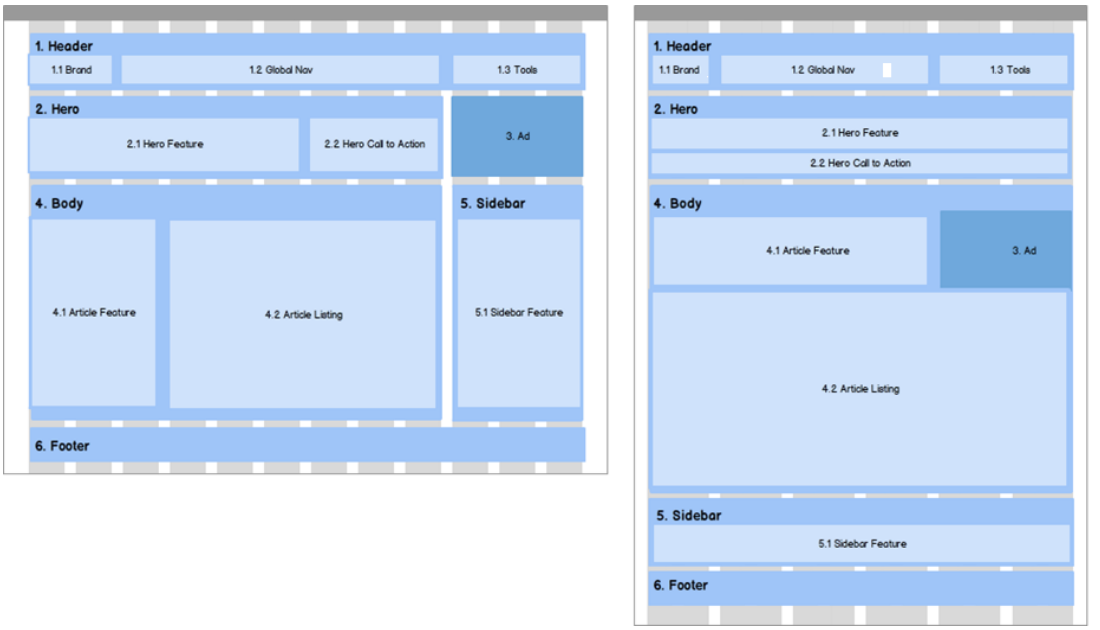
- **It is part of the responsive design toolbox, but is not the only one.**
- **Like a grid, it is made of vertical and horizontal alignments**
- **It concerns itself with reflow of the layout for when the screen size changes rather than changing the size of the grid units or the atomic grid unit**

47

Definition

- A flexible "grid" that allows content to be structured consistently and predictably, but allows for the size and positioning of the elements to change depending on the available size of the rendering device's display.
- It is part of the responsive design toolbox, but is not the only one.
- Like a grid, it is made of vertical and horizontal alignments
- It concerns itself with reflow of the layout for when the screen size changes rather than shrinking the size of the grid units (columns and – if they're being used, rows) or the atomic grid unit

The Fluid Grid



48

Definition

- A flexible “grid” that allows content to be structured consistently and predictably, but allows for the size and positioning of the elements to change depending on the available size of the rendering device’s display.
- It is part of the responsive design toolbox, but is not the only one.
- Like a grid, it is made of vertical and horizontal alignments
- It concerns itself with reflow of the layout for when the screen size changes rather than shrinking the size of the grid units (columns and – if they’re being used, rows) or the atomic grid unit

What are Grid Systems and Frameworks?

- **Grids are typically CSS frameworks designed to simplify**
  - Page layout
  - Cross Browser compatibility
  - Increasingly responsive development patterns
- **They are serious pieces of CSS work put together by developers**
  - Rigorous software principals of reuse and modularisation are applied
- **Providing reusable patterns and tools for:**
  - Layout
  - Navigation
  - Typography
  - RIA plugins
- **When used by CSS developers aware they make everything easier!**

49

The idea of a Grid is not dissimilar to drawing on a piece of graph paper. As a concept in design it became a common part of design and development in the Art Deco movement and give rise to the beloved Helvetica font.

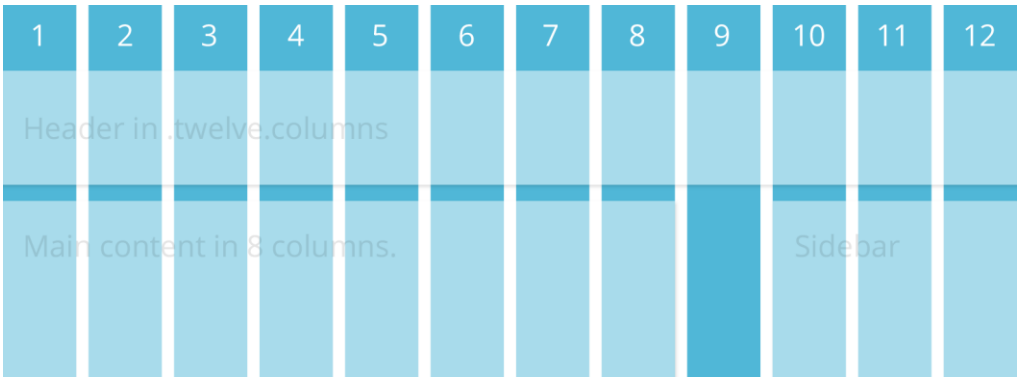
Within web development they are CSS frameworks providing standardised rules for build and shortcuts for site construction using pre-prepared layout classes.

Grids save a lot of time in development, they implement many of the web's best practices for desktop and mobile development. By learning to build a grid the basis of page construction is done for you leaving you to focus on the functionality and/or appearance of the page.

The CSS file that provides the grid provides a good reuse policy similar to the way jQuery has been used in page scripting.

Using Foundation – The Grid

- **The basis of Foundation is its Grid – it consists of 12 columns**
  - The grid is metaphorical, there is no actual grid
  - We use it to space and control areas of content



50

“The grid is the most vivid manifestation of the will to order in graphic design. Units are the basic building block of a grid. They’re all uniform. Columns are the grouping of units that create the visual structure of the page. They are not necessary uniform.”(sic) – Grids are good

Foundation uses the CSS3 border-box and is mobile-first meaning that content is staked by default.

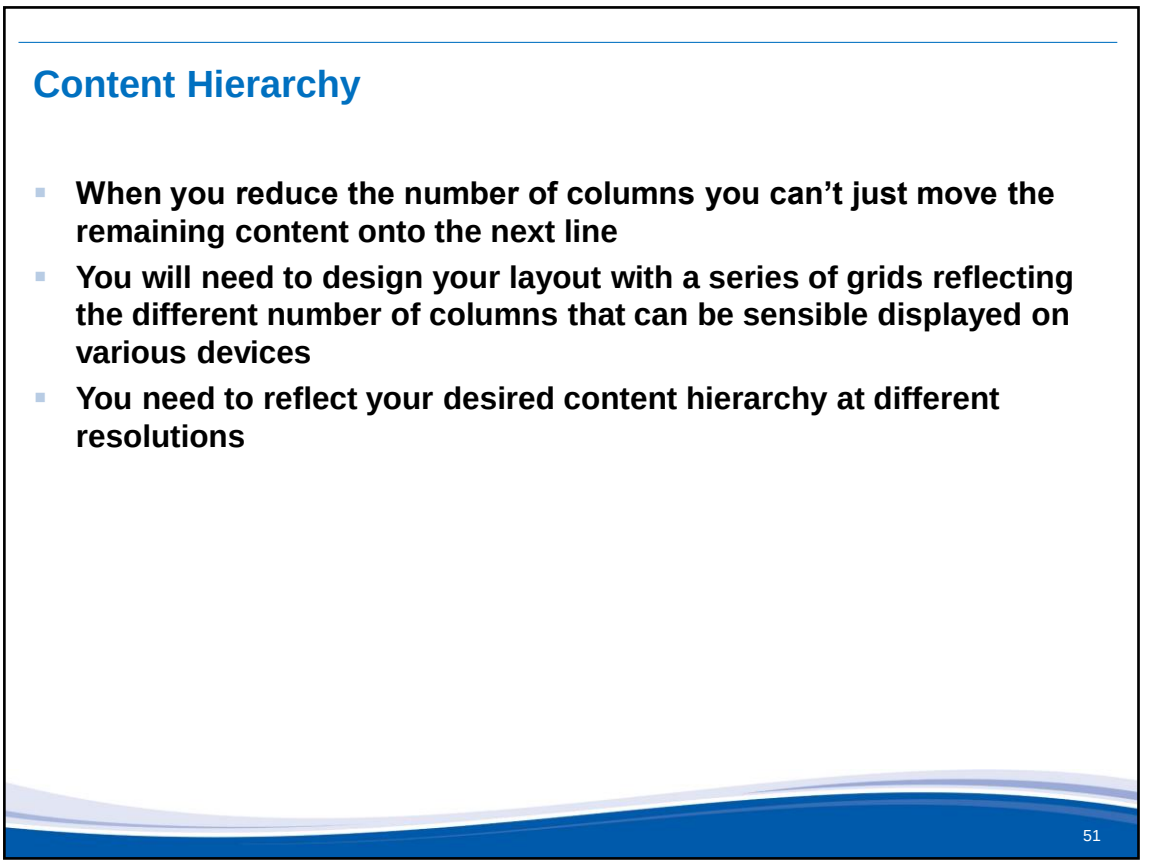
The core of Foundation can be summed up in a few points:

A 12-column, percentage-based grid with an arbitrary maximum width.
The grid can be nested and used for quite complex layouts, and it works all the way back to IE 7. The grid reshuffles itself for smaller devices.

Image styles that disregard pixels.
Images in Foundation are scaled by the grid to different widths.

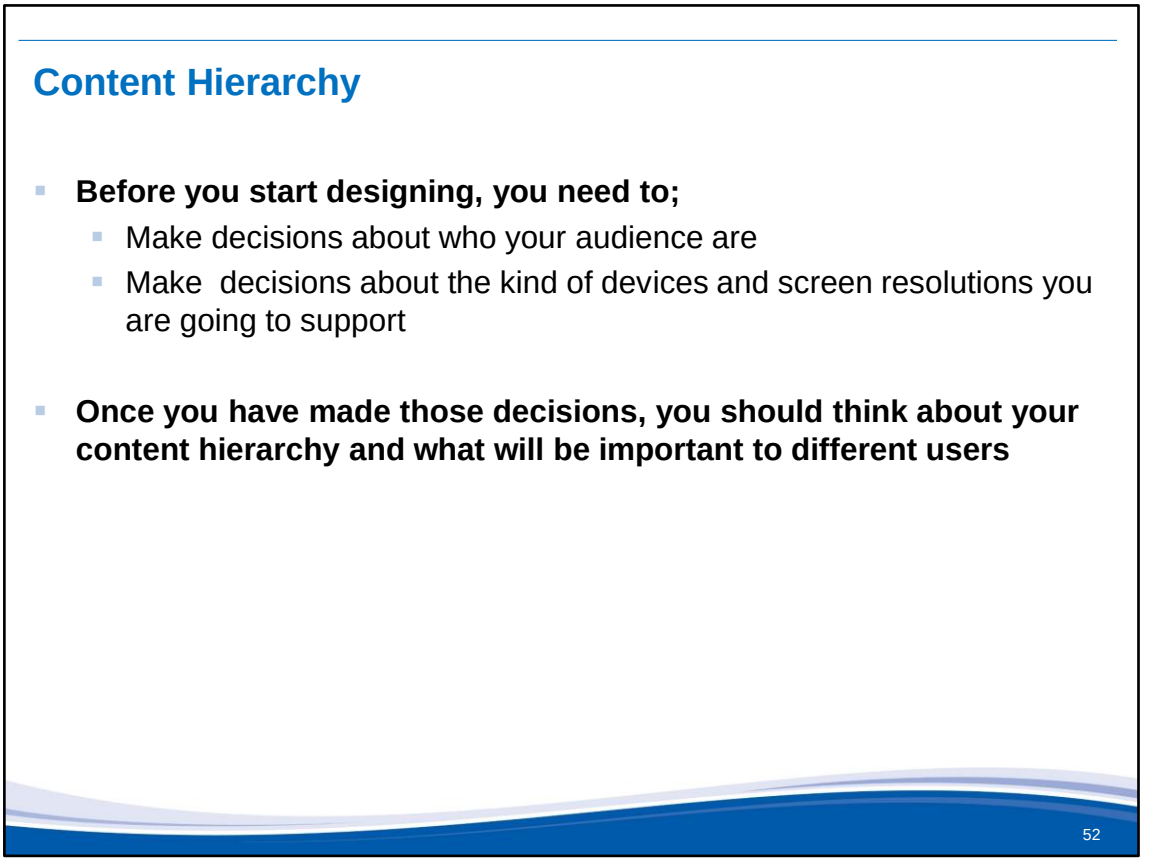
UI and layout elements.
Foundation includes common pieces such as typography and forms, as well as tabs, pagination, N-up grids and more.

Mobile visibility classes.
Rapidly prototyping is partly about having built-in functionality to tailor the experience. Foundation lets you very quickly hide and show elements on desktops, tablets and phones.



Content Hierarchy

- It's not just a case of reducing the number of columns on the screen and just moving the remaining content onto the next line; you need to reflect your desired content hierarchy at different resolutions
- *The content hierarchy is likely to be the cause of much debate; it will be your job to make sure that the hierarchy reflects the importance to the user rather than the importance to the stakeholders; you may need to bring them with you and convince them, especially if they are precious about their "little empire" and are pushing to make it more important for reasons of ego, or because they have a limited view rather than a holistic one.*



Preparing

Decisions about who your audience are (you can't decide what is important to your audience if you do not define your audience, nor can you necessarily determine what devices they are likely to be using)

Decisions about the kind of devices and screen resolutions you are going to support

Before we can make any design decisions we should think about your content hierarchy and what will be important to different users.

Content Hierarchy

- **Example Hierarchy**
  - In this example there are six groups of elements which are listed in order of importance between the header and the footer
    - Group 1: Header
    - Group 2: Featured Content
    - Group 3: Advertising
    - Group 4: Main Content
    - Group 5: Sidebar
    - Group 6: Footer

The order of importance will be debatable, and must be determined in conjunction with various stakeholders

53

Let us imagine that we have 6 groups of elements which are ordered in order of importance

Group 1: Header

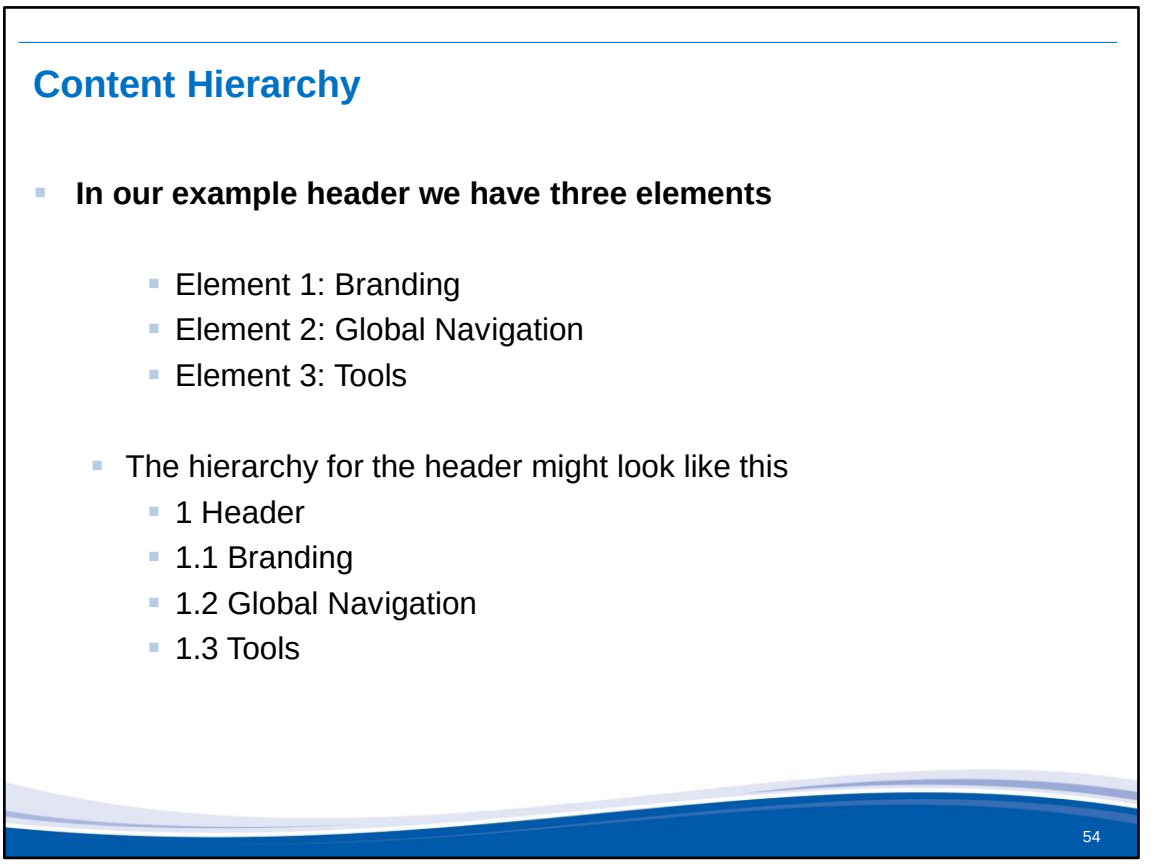
Group 2: Featured Content

Group 3: Advertising

Group 4: Main content

Group 5: Sidebar

Group 6: Footer

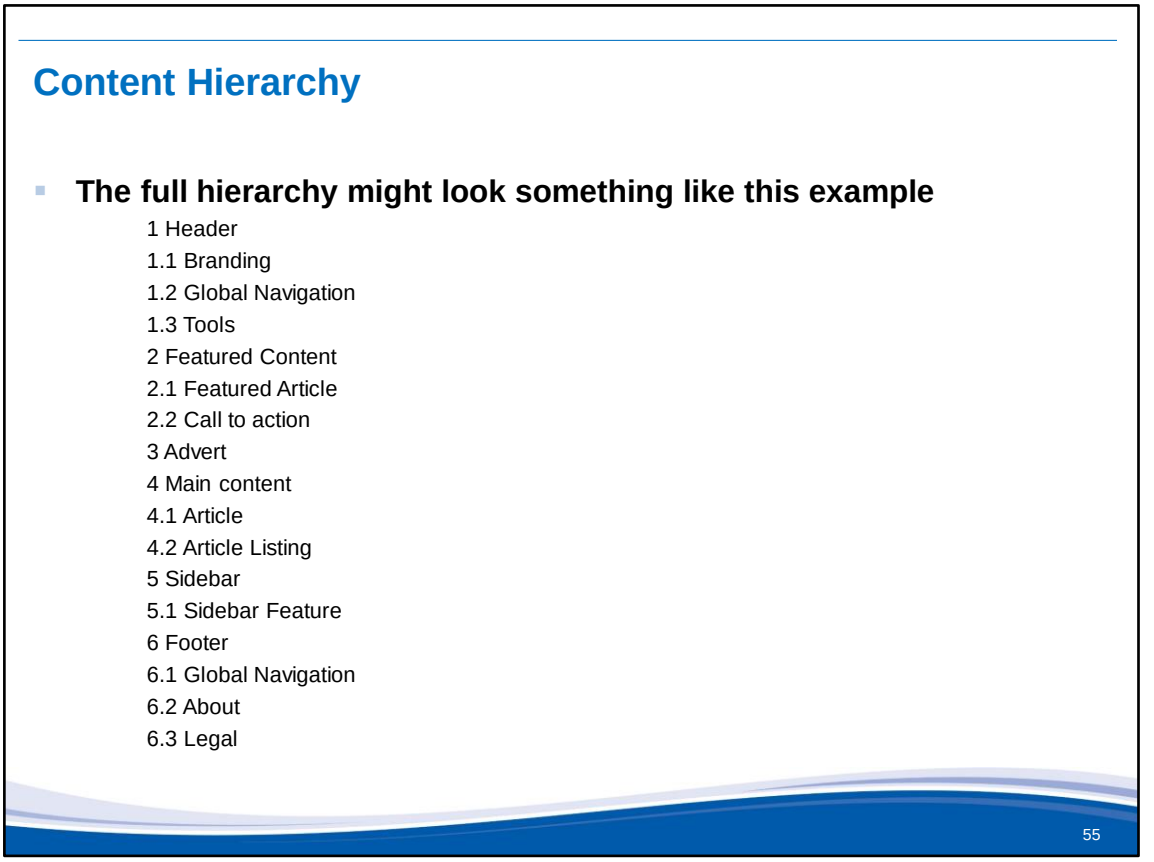


In the header we have three elements

Element 1: Branding

Element 2: Global Navigation

Element 3: Tools

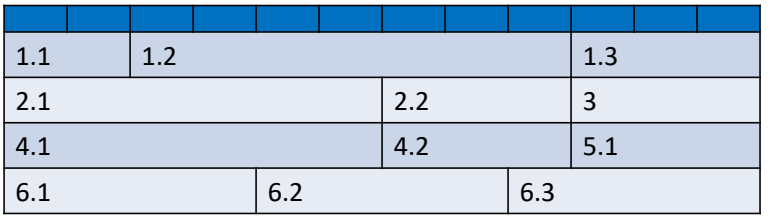


The hierarchy might look like this

1 Header
1.1 Branding
1.2 Global Navigation
1.3 Tools
2 Featured Content
2.1 Featured Article
2.2 Call to action
3 Advert
4 Main content
4.1 Article
4.2 Article Listing
5 Sidebar
5.1 Sidebar Feature
6 Footer
6.1 Global Navigation
6.2 About
6.3 Legal

Grid Layout

Desktop



56

Desktop

On a desktop you might lay it out as such (use table to emulate a grid)

1.1, 1.2, 1.3

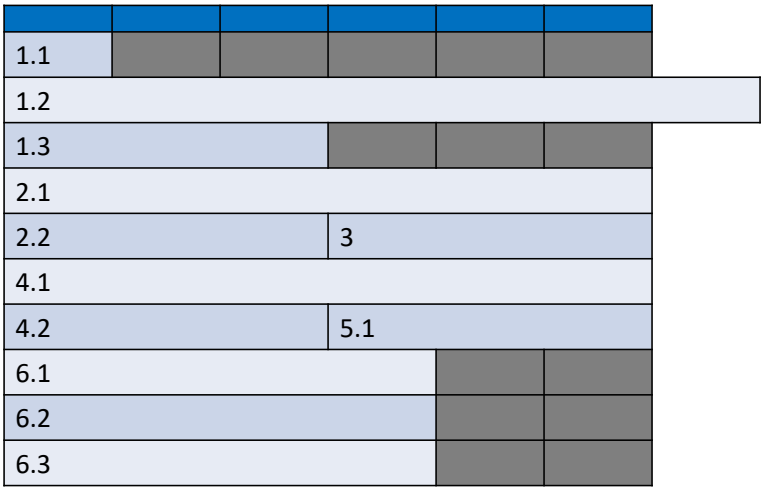
2.1, 2.2, 3

4.1, 4.2, 5.1

6.1, 6.2, 6.3

Grid Layout

Reflow (Tablet) – How not to do it



57

If the number of columns are reduced, you should not just re-shuffle the elements onto a new line when it no longer fits

Whilst you will often see this sort of result, as screen sizes shrink it is the result of a design that is not well optimised for varying resolutions, or does not follow responsive design.

Lots of space is wasted, and some of the elements are wider than grid, and will go off the side of the screen, or make the rest of the content too small for the screen

Grid Layout

Reflow (Tablet)

1.1	1.2			1.3	
2.1		2.2		3	
4.1					
4.2					
5.1					
6.1		6.2		6.3	

1.1	1.2			1.3	
2.1			2.2		
3					
4.1					
4.2					
5.1					
6.1		6.2		6.3	

58

Tablet

As there is less screen real estate on a tablet compared to a desktop, we will need to think about structuring the content differently.

The most important information should still be the most prominent, but we have halved the width of the screen, and thus half the number of grid units

The elements will now be laid out accordingly

1.1, 1.2, 1.3

2.1, 2.2, 3

4.1,

4.2

5.1

6.1, 6.2, 6.3

Note that they may have the same width as they did previously. Most will be narrower, having shrunk to smaller dimensions, but other elements will have expanded to take up more grid units

Without the advertising, nothing would share a line with something from another group, and depending on the project, if there is an advert that must be served it

may be possible to give it a line to itself which would be preferable in most cases so long as the advert does not leave a largely empty line, and does not take up too much vertical space

1.1, 1.2, 1.3

2.1, 2.2

3

4.1,

4.2

5.1

6.1, 6.2, 6.3

Grid Layout

Reflow (Smartphone)

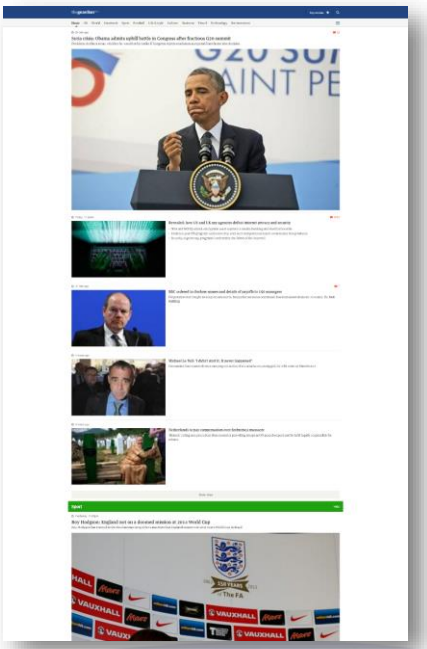
1.1
1.2
1.3
2.1
2.2
3
4.1
4.2
5.1
6.1
6.2
6.3

A mobile screen typically has even less space than a tablet screen

You will want to be cautious about what you show

You will either want to use a single unit wide grid, or (at most) a two unit wide grid

This may change as different kinds of mobile display are developed



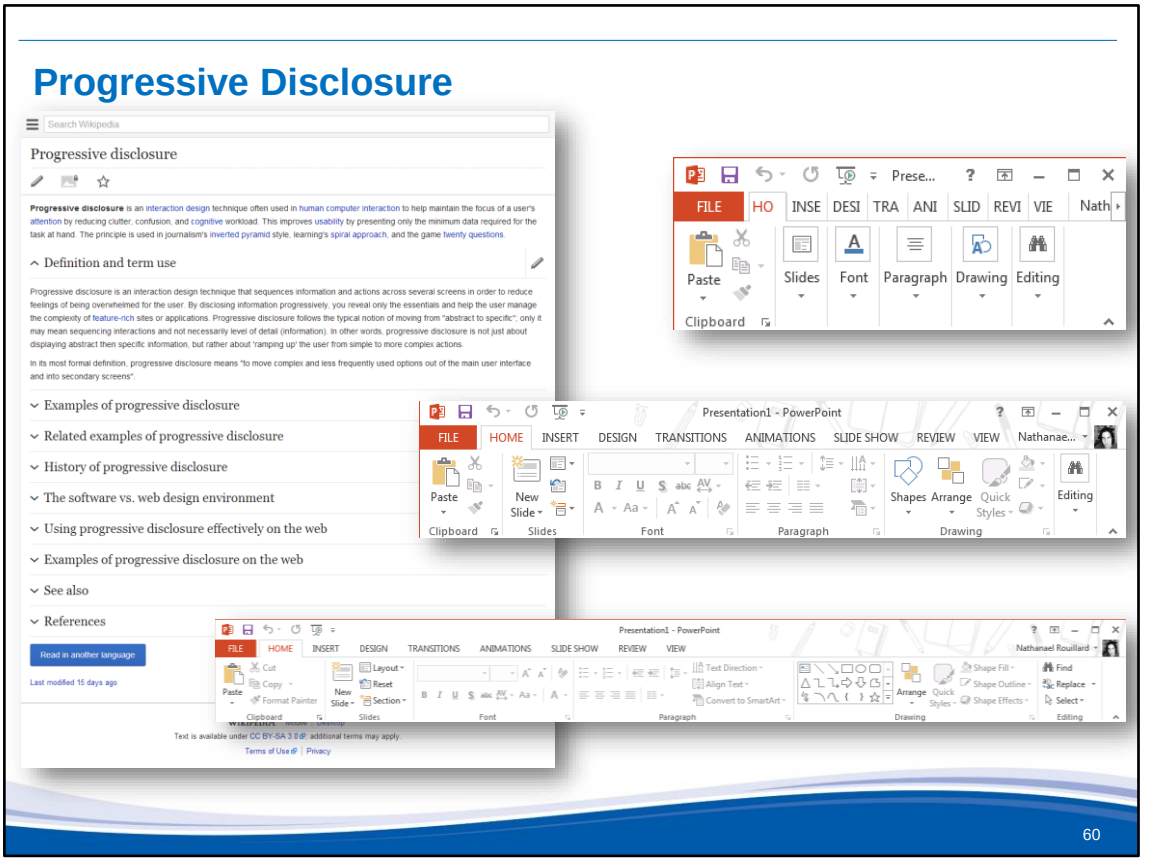
59

Mobile

This will have even less space than the tablet, so you will want to be cautious about what you show.

At the smallest screen size, your layout will be

1.1
1.2
1.3
2.1
2.2
3
4.1
4.2
5.1
6.1
6.2
6.3



You can take advantage of Progressive Disclosure.

At its most extreme you can regress all the way to the top group level

I.e.,

1

2

3

4

5

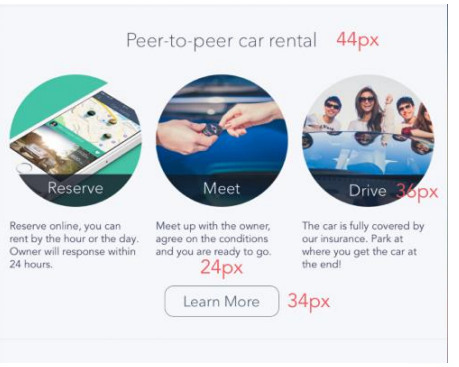
6

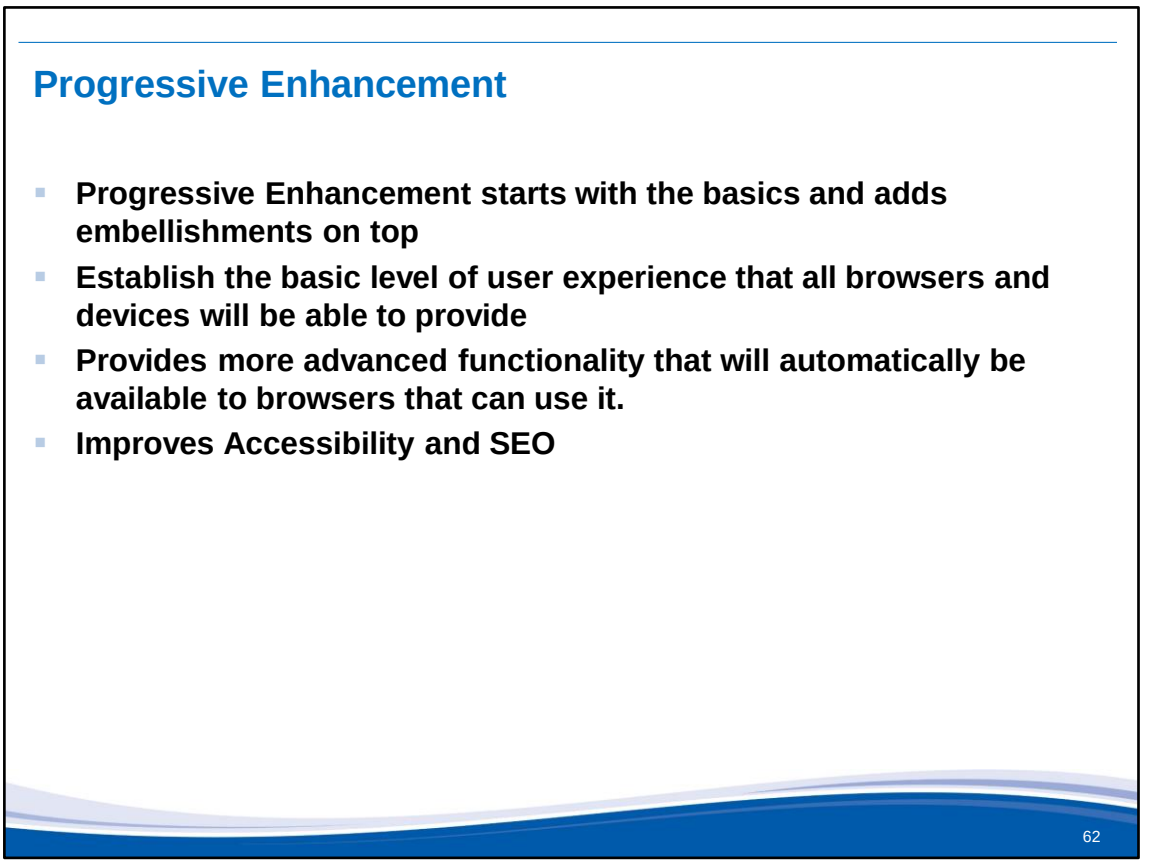
This is level of progressive disclosure is commonly used on mobile sites, but it is possible to use progressive disclosure non mobile sites, with variable levels of disclosure.

E.g. the MS Office ribbon uses a variable level of progressive disclosure, (as well as a fluid grid)

Define Standards

- We need dependable experience across devices – so we need to set standards. The standards bellow work on all major platforms:
  - The minimum sizes for typography (*24px*+, optimal for reading: *32px*),
  - Buttons (*44px* to *88px*)
  - Navigation bar (*72px* to *98px*)





Progressive Enhancement

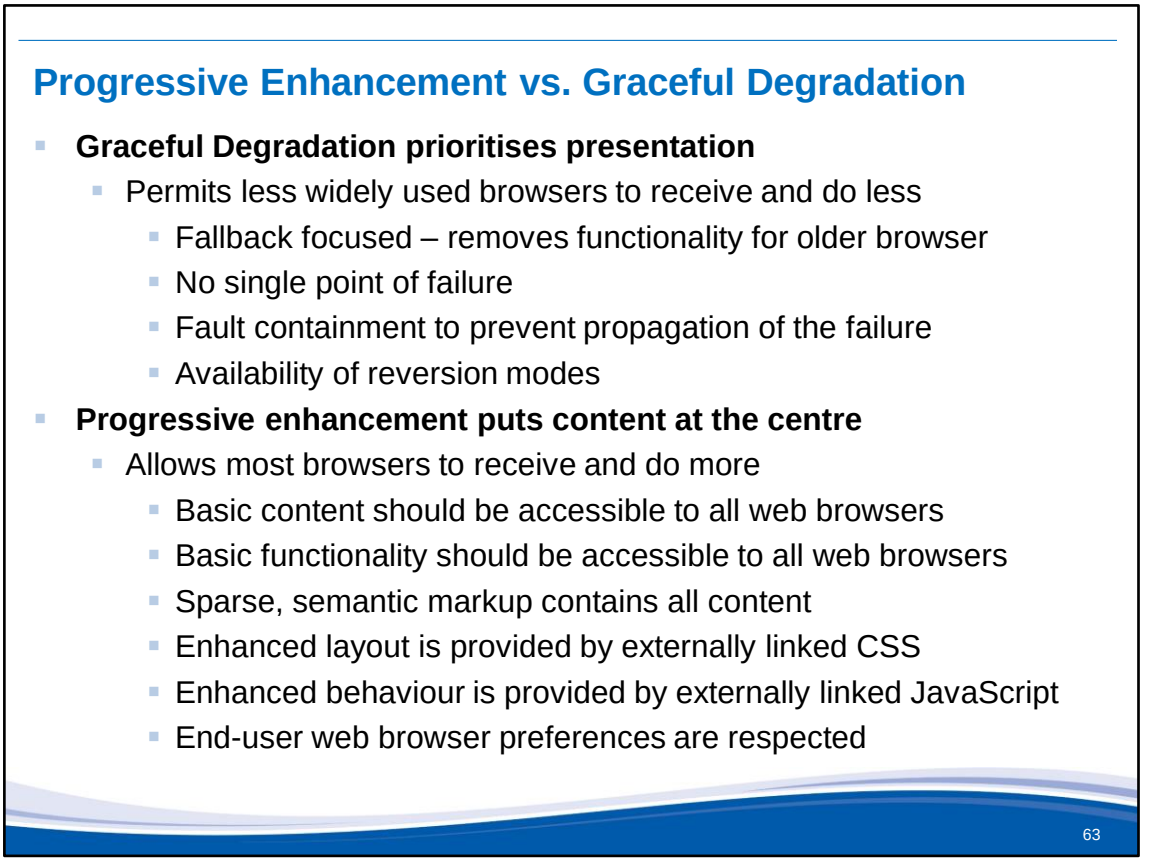
Progressive enhancement does things the other way round. You start by establishing a basic level of user experience that all browsers will be able to provide when rendering your web site, but you also build more advanced functionality that will automatically be available to browsers that can use it.

One of the earliest common uses of progressive enhancement is the external CSS style sheet, which would only be used by browsers that could render CSS, but would be ignored by non-CSS browsers. These non-CSS browsers would only see the plain mark-up, and render it according to their own internal style sheets. The more modern graphical browsers would apply the external CSS style sheet, and thus render a more visually pleasing page with (hopefully) improved usability.

Progressive enhancement is commonly used for JavaScript and Flash as many devices either do not support them, or have had support for them turned off by users.

Web pages created according to the principles of progressive enhancement are naturally more accessible, because the core principles demand that the basic content is always available, and not obstructed by potentially unsupported or disabled scripting. The sparse mark-up principle also makes it easier for accessibility tools such as screen readers to find the content that it is looking for. A side effect of the improved accessibility is that it also provides improved SEO (search engine optimisation) results as the search engine spiders are likewise able to access the basic content that they are looking for, aiding in the task of

search engine indexing.



Progressive Enhancement vs. Graceful Degradation

The concepts of graceful degradation and progressive enhancement are often applied to describe browser support strategies. Indeed, they are closely related approaches to the engineering of "fault tolerance".

These two concepts influence decision-making about browser support. Because they reflect different priorities, they frame the support discussion differently. Graceful degradation prioritizes presentation, and permits less widely-used browsers to receive less (and give less to the user). Progressive enhancement puts content at the center, and allows most browsers to receive more (and show more to the user). While close in meaning, progressive enhancement is a healthier and more forward-looking approach. Progressive enhancement is a core concept of Graded Browser Support.

When to use Graceful Degradation

- **There are a few cases where graceful degradation is the right choice**
  - When maintaining an existing site that uses Graceful Degradation
  - Retrofitting an old product when you do not have the time, access, or insight to replace everything
  - Shortest possible time to initial delivery is of the utmost importance
  - Your product is inherently dependent on scripting for even its most basic functions

64

There are a few cases where graceful degradation is the right choice.

- Retrofitting an old product and you do not have the time, access, or insight to change or replace everything
- You may find that you are unable to rewrite everything, in which case graceful degradation is your answer.
- Shortest possible time to initial delivery is of the utmost importance, for example very high traffic sites where every millisecond of performance means a difference of millions of dollars.
- Your product by definition is so dependent on scripting that it makes more sense to maintain a “basic” version rather than enhancing one (Maps, email clients, feed readers).

Discussion - Graceful Degradation & Progressive Enhancement

You have a form field that contains an email address. You want to make sure that there is an email address included before the form is submitted. This can be achieved with JavaScript, but what other approaches and problems should be taken into consideration?

You are building a product using progressive enhancement and you want to display a map inside the product. What base functionality do you start from?

You have an interface that consists of two dropdown form controls. Setting an option in the first will change the available options in the second. What could be a fall back for this control? What issues could there be with it?

65

Exercises

You have a form field that contains an email address. You want to make sure that there is an email address included before the form is submitted. This can be achieved with JavaScript, but what other approaches and problems should be taken into consideration

You are building a product using progressive enhancement and you want to display a map inside the product. What base functionality do you start from?

You have an interface that consists of two dropdown form controls. Setting an option in the first will change the available options in the second. What could be a fall back for this control? What issues could there be with it?

Text & Proportions

- To some extent, bigger is better when working with mobile
  - But more importantly, it's the proportions that harmonize the content.
  - If your body typography is set at *24px*
  - Make sure that the rest of the site is consistent with it.
    - There are no hard rules to this, but the *line-height* should be from *1.2x to 1.4x* the font size
      - Many grid systems use the Golden Ratio to achieve this
    - Titles are usually larger by *1.5x-2x*, but thinner as well—otherwise they steal too much attention
  - Line length should be between 45 and 90 characters