

Internet and Web Principal

Week 8

Responsive Web Design

Content

1. Why RWD is important?
2. Fluids Layouts
3. Media Queries
4. Design strategies and patterns
5. Testing

Why RWD is important?

A website may be conveniently read and utilized on a variety of devices thanks to responsive web design, often known as RWD. The fundamental tenet is that the same HTML code is sent on all devices at the same URL, but various styles are applied dependent on viewport size to reorder elements and improve usability.

Why RWD is important?

An major milestone was reached in 2016 when mobile internet usage exceeded desktop usage. The proportion of online traffic that originates from platforms other than desktop browsers is rising continuously. Younger users may choose to just utilize mobile devices. In addition, the great majority of us access the internet throughout the day from a variety of devices (phone, tablet, computer). And guess what? Regardless of how we approach your site, we anticipate having a comparable experience with your content or service.

Why RWD is important?

RWD fits into that situation. You can guarantee that mobile users receive the same material as regular visitors by using a single source (although it might be organized differently). The mere fact that a user is using a smartphone does not subject them to material or services that are diminished. You can guarantee a consistent experience for visitors who could start using your site on one device and conclude it on another.

In reality, "responsive design" is now simply "web design" for many web designers. It is replacing the specialty approach as the standard method for creating websites that are responsive to the needs of our present multidevice environment.

Why RWD is important?

We were unsure of how we could handle screens that fit in the palm of your hand because we were used to designing primarily for giant desktop screens. Using the phone's built-in web display feature was one option. Mobile devices often show a full web page that has been compressed to fit on the available screen space. Users may browse to different portions of the page and zoom in on certain information by pinching the screen. Even if it is theoretically possible, the experience is not ideal. Another strategy was to design a unique mobile website specifically for handheld devices and consumers who are "on the go."

Why RWD is important?

Viewport

Mobile browsers render pages on a canvas called the viewport, which is subsequently shrunk to fit the width of the screen in order to accommodate regular websites into small devices (device width). A web page is displayed NOTE as though it were on a desktop browser window that is 980 pixels wide, for instance, on iPhones where mobile Safari sets the viewport width to 980 points. This depiction is compressed to fit inside the confines of the screen, which can be anywhere from 320 and 414 points wide depending on the iPhone model.

Why RWD is important?

Mobile Safari introduced the viewport **meta** element, which allows us to define the size of that initial viewport. Soon, the other mobile browsers followed suit. The following **meta** element, which goes in the **head** of the HTML document, tells the browser to set the width of the viewport equal to the width of the device screen (**width=device-width**), whatever that happens to be ([FIGURE 17-2](#)). The **initial-scale** value sets the zoom level to 1 (100%).

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

With the viewport **meta** element in place, if the device's screen is 320 pixels wide, the rendering viewport on that device will also be 320 pixels across (not 980) and will appear on the screen at full size. That is the width we test for with media queries, so setting the viewport is a crucial first step.

Why RWD is important?

By default, the viewport shrinks to the size of the screen.



With the viewport meta tag, the viewport is created at the same size as the screen.

```
<meta name="viewport"  
content="width=device-width,  
initial-scale=1">
```

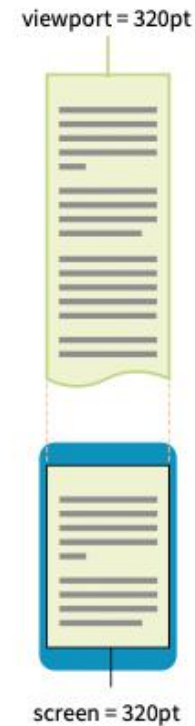


FIGURE 17-2. The viewport **meta** element matches the resolution of the device's browser viewport to the resolution of its screen.

Fluids Layouts

In a **fluid layout**, the page area and its grid resize proportionally to fill the available width of the screen or window (**FIGURE 17-3**, top). That is easily accomplished with **fr** and **minmax()** units in CSS Grid layouts and with **flex** property settings in Flexbox. If you need to also target older browsers that don't support CSS layout standards, you can use percentage values for horizontal measurements so elements remain proportional at varying sizes. On the downside, fluid layouts may allow text line lengths to become uncomfortably long, so that is something to watch out for.

Fluids Layouts

Fluid layouts fill the viewport proportionally.



w3c.org

Fixed layouts stay the same size and may get cut off or leave extra space.



kexp.org

FIGURE 17-3. Fluid and fixed layout examples.

Fluids Layouts

Making Images Flexible

Every now and then a solution is simple. Take, for example, the style rule required to make images scale down to fit the size of their container:

```
img {  
  max-width: 100%;  
}
```

That's it! When the layout gets smaller, the images in it scale down to fit the width of their respective containers. If the container is larger than the image—for example, in the tablet or desktop layouts—the image does not scale larger; it stops at 100% of its original size ([FIGURE 17-4](#)). When you apply the **max-width** property, you can omit the **width** and **height** attributes in the **img** elements in the HTML document. If you do set the **width** attribute, be sure the **height** attribute is set to **auto**; otherwise, the image won't scale proportionately.

Fluids Layouts

```
img { max-width: 100%; }
```

In this book, Philadelphia Ice Creams, comprising the first group, are very palatable, but expensive. In many parts of the country it is quite difficult to get good cream. For that reason, I have given a group of creams, using part milk and part cream, but it must be remembered that it takes smart "chugging" to make the best of your milk. For the better use condensed milk, with enough water or milk to make out the cream.

Ordinary fruit creams may be made with condensed milk at a cost of about fifteen cents a quart, which, of course, is cheaper than ordinary milk and cream.



If pure raw cream is stirred rapidly, it swells and becomes frothy, like the beaten whites of eggs, and is "whipped cream." To prevent this in making Philadelphia Ice Cream, one-half the cream is scalded, and when it is very cold, the remaining half of raw cream is added. This gives the smooth, light and rich consistency which makes these creams so different from others.

The time for freezing varies according to the quality of water or milk or water. Water will require a longer time than ice creams. If it is too hot, to freeze the mixture too rapidly, they are apt to be coarse, not smooth, and if they are churned before the mixture is too cold they will be coarse or broken.

In this book, Philadelphia Ice Creams, comprising the first group, are very palatable, but expensive. In many parts of the country it is quite difficult to get good cream. For that reason, I have given a group of creams, using part milk and part cream, but it must be remembered that it takes smart "chugging" to make the best of your milk. For the better use condensed milk, with enough water or milk to make out the cream.

Ordinary fruit creams may be made with condensed milk at a cost of about fifteen cents a quart, which, of course, is cheaper than ordinary milk and cream.



If pure raw cream is stirred rapidly, it swells and becomes frothy, like the beaten whites of eggs, and is "whipped cream." To prevent this in making Philadelphia Ice Cream, one-half the cream is scalded, and when it is very cold, the remaining half of raw cream is added. This gives the smooth, light and rich consistency which makes these creams so different from others.

In this book, Philadelphia Ice Creams, comprising the first group, are very palatable, but expensive. In many parts of the country it is quite difficult to get good cream. For that reason, I have given a group of creams, using part milk and part cream, but it must be remembered that it takes smart "chugging" to make the best of your milk. For the better use condensed milk, with enough water or milk to make out the cream.

Ordinary fruit creams may be made with condensed milk at a cost of about fifteen cents a quart, which, of course, is cheaper than ordinary milk and cream.



If pure raw cream is stirred rapidly, it swells and becomes frothy, like the beaten whites of eggs, and is "whipped cream." To prevent this in making Philadelphia Ice Cream, one-half the cream is scalded, and when it is very cold, the remaining half of raw cream is added. This

FIGURE 17-4. Setting the **max-width** of inline images allows them to shrink to fit available space but not grow larger than actual size.

Media Queries

Media queries apply different styles based on characteristics of the browser: its width, whether it is vertically or horizontally oriented, its resolution, and more. They are what make it possible to send a one-column layout to small screens and a multicolumn layout to larger screens on the fly.

The query itself includes a media type followed by a particular feature and a value for which to test. The criteria are followed by a set of curly brackets that contain styles to apply if the test is passed.

```
@media type and (feature: value) {  
/* styles for browsers that meet this criteria */  
}
```

Let's clarify that with an example. The following media queries look at whether the viewport is on a screen and in **landscape** (horizontal) or **portrait** (vertical) orientation. When the query detects that the viewport is in landscape mode, the background color of the page is "skyblue"; when it is in portrait orientation, the background is "coral" ([FIGURE 17-5](#)).

Media Queries

```
@media screen and (orientation: landscape) {  
  body {  
    background: skyblue;  
  }  
}  
@media screen and (orientation: portrait) {  
  body {  
    background: coral;  
  }  
}
```



When the viewport is in portrait mode, the background color is "coral."



When the viewport is in landscape mode, the background color is "skyblue."

FIGURE 17-5. Changing the background color based on the orientation of the viewport with media queries.

Media Queries

TABLE 17-1. Media features you can evaluate with media queries

Feature	Description
width	The width of the display area (viewport). Also min-width and max-width .
height	The height of the display area (viewport). Also min-height and max-height .
orientation	Whether the device is in portrait or landscape orientation.
aspect-ratio	Ratio of the viewport's width divided by height (width/height). Example: aspect-ratio: 16/9 .
color	The bit depth of the display; for example, color: 8 tests for whether the device has at least 8-bit color.
color-index	The number of colors in the color lookup table.
monochrome	The number of bits per pixel in a monochrome device.
resolution	The density of pixels in the device. This is increasingly relevant for detecting high-resolution displays.
scan	Whether a tv media type uses progressive or interlace scanning. (Does not accept min-/max- prefixes.)
grid	Whether the device uses a grid-based display, such as a terminal window. (Does not accept min-/max- prefixes.)

Media Queries

How to use Media Queries

Media queries may be used to conditionally load external style sheets or inside a style sheet itself. Inline styles and media queries cannot be used together.

Use of a `@media` ("at- media") rule directly in the style sheet is the most popular method for using media queries.

The sequence of rules is crucial when using media queries within a style sheet. Your media query must appear after any rules with the same declaration because rules later in the style sheet override rules that come before them.

In order to optimize for different viewing environments, the idea is to first establish the baseline styles that act as a default and then override individual rules as necessary.

Media Queries

With external style sheets

Developers may decide to segregate the styles for various devices on big or complex websites into their own style sheets and only call the entire.css file when certain criteria are satisfied. One approach is to conditionally load different.css files by using the media property in the link element. In this example, a style sheet that will only be utilized if the device is more than 1,024 pixels wide (and if the browser supports media queries) is requested after the basic styles for the website:

```
<head>  
<link rel="stylesheet" href="styles.css">  
<link rel="stylesheet" href="2column-styles.css" media="screen and  
(min-width:1024px)"> </head>
```

Similarly, you can carry out media queries with **@import** rules that pull in external style sheets from within a style sheet. Notice that the word “media” does not appear in this syntax, only the type and Query

```
<style>  
@import url("/default-styles.css");  
@import url("/wide-styles.css") screen and (min-width: 1024px); /* other styles */  
</style>
```

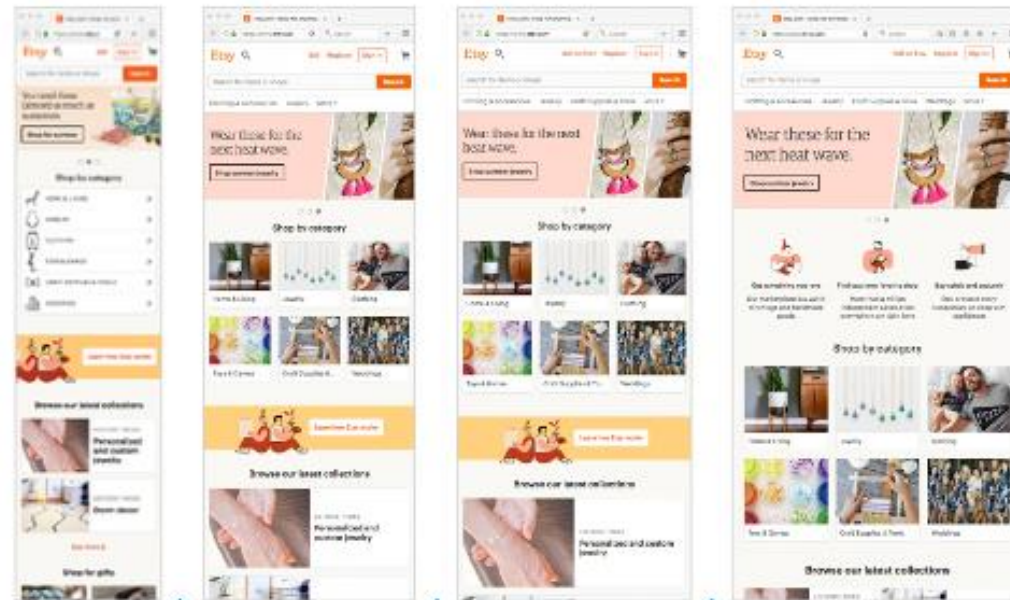
Media Queries

CHOOSING BREAKPOINTS

A **breakpoint** is the point at which we use a media query to introduce a style change. When you specify **min-width: 800px** in a media query, you are saying that 800 pixels is the “breakpoint” at which those particular styles should be used. **FIGURE 17-6** shows some of the breakpoints at which Etsy.com makes both major layout changes and subtle design tweaks on its home page.

When RWD was first introduced, there were only a handful of devices to worry about, so we tended to base our breakpoints on the common device sizes (320 pixels for smartphones, 768 pixels for iPads, and so on), and we created a separate design for each breakpoint. It didn’t take long until we had to deal with device widths at nearly every point from 240 to 3,000+ pixels. That device-based approach definitely didn’t scale.

Media Queries



At the 480-pixel breakpoint, the category navigation changes from a list to photos. “Register” is added to the top navigation bar.

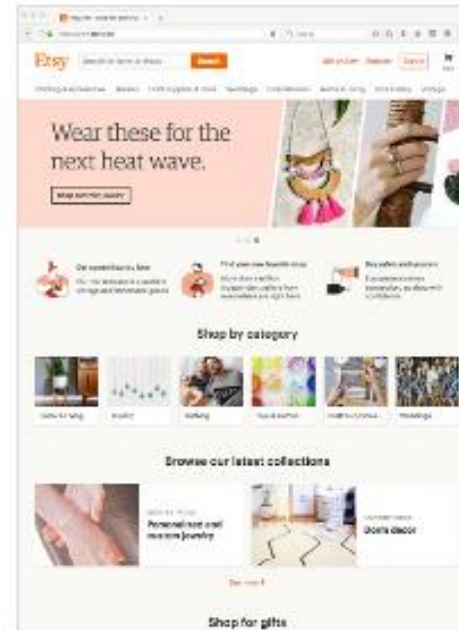
At 501 pixels, “Sell” becomes “Sell on Etsy” (a very subtle adjustment). You can also see more links in the navigation bar under the search field.

At 640 pixels, the “How Etsy Works” images and messages move above the categories. In smaller views, they were accessible via the “Learn how Etsy works” link in a yellow bar.

Media Queries



At 901 pixels, the search input form moves into the top header.



At 981 pixels, the word "Cart" appears under the shopping cart icon. We now see the full list of navigation options in the header (no "More" link). At this point, the layout expands to fill larger windows until it reaches its maximum width of 1400 pixels. Then margins add space equally on the left and right to keep the layout centered.

Media Queries

Module-Based Breakpoints

Instead of changing the entire page at once, it is preferable to define breakpoints for each section of a page (although for some pages that may work just fine). Creating the design for small screens first, resizing the browser to a larger size, and then paying attention to the point at which each component of the website starts to become unsatisfactory are frequent practices. The one-column layout could be OK up until it reaches 800 pixels, at which time a two-column design might be added. However, the navigation might become too difficult and need a breakpoint at 400 pixels wide.

Design strategies and patterns

DESIGNING RESPONSIVELY

In our tests, we've encountered a few instances where content seems goofy when the browser becomes extremely narrow or extremely wide. A three-column layout simply does not work, and text resized to fit a 320 pixel wide screen may become illegible. On the other hand, when the viewport occupies a high-resolution desktop display, the line length in single-column layouts becomes too lengthy to read comfortably. One size does not fit all when it comes to many parts of a web page. To "preserve the integrity" of the elements, designers must be aware of where things break down and designate breakpoints.

Design strategies and patterns

Content Hierarchy

Because content is king on the web, it is vital that content is thoroughly evaluated and arranged before any code is produced. These are the kinds of duties that Information Architects and Information Strategists perform in order to organize, categorize, plan, and manage digital content.

A major challenge is the hierarchy and organization of the site's numerous perspectives, with an emphasis on the small-screen user experience. For the greatest browsing experiences across all platforms, it is preferable to start with an inventory of prospective information and narrow it down to what is most crucial and helpful. You may choose the order in which the content modules show on various screen sizes once you know what they are.

Design strategies and patterns

Layout

With reason, rearranging material into multiple layouts may be the first thing that comes to mind when you think about responsive design. The layout influences our first impressions of the content and usefulness of a website.

As previously said, responsive design is built on fluid layouts that expand and contract to fill the viewport's available area. However, one fluid layout is frequently insufficient to suit all screen sizes. More frequently than not, two or three layouts are created to suit device requirements, with minor tweaks in between layout shifts.

Design strategies and patterns

Layout and line length

Text line lengths are an excellent indicator for when to change the layout. Text lines that are too short or too lengthy are difficult to read, therefore strive for ideal line lengths of 45 to 75 characters, including spaces. If your text lines are much longer, it's time to make layout modifications, such as extending the margins or adding a column. To maintain the character count within the acceptable range, you might alternatively raise the font size of the content.

Design strategies and patterns

In this book, Philadelphia Ice Creams, comprising the first group, are very palatable, but expensive. In many parts of the country it is quite difficult to get good cream. For that reason, I have given a group of creams, using part milk and part cream, but it must be remembered that it takes smart "juggling" to make ice cream from milk. By far better use condensed milk, with enough water or milk to rinse out the cans.

FIGURE 17-8. Highlight the 45th to 75th characters to test for optimal line lengths at a glance.

Design strategies and patterns

Responsive layout patterns

The manner in which a site transitions from a small-screen layout to a wide-screen layout must make sense for that particular site, but there are a few **patterns** (common and repeated approaches) that have emerged over the years.

Mostly fluid

This pattern uses a single-column layout for small screens, and another fluid layout that covers medium and large screens, with a maximum width set to prevent it from becoming too wide. It generally requires less work than other solutions.

Column drop

This solution shifts between one-, two-, and three-column layouts based on available space. When there isn't room for extra columns, the sidebar columns drop below the other columns until everything is stacked vertically in the one-column view.

Design strategies and patterns

Layout shifter

If you want to get really fancy, you can completely reinvent the layout for a variety of screen sizes. Although expressive and potentially cool, it is not necessary. In general, you can solve the problem of fitting your content to multiple environments without going overboard.

Tiny tweaks

Some sites use a single-column layout and make tweaks to type, spacing, and images to make it work across a range of device sizes.

Off canvas

As an alternative to stacking content vertically on small screens, you may choose to use an “off-canvas” solution. In this pattern, a page component is located just out of sight on the left or right of the screen and flies into view when requested. A bit of the main content screen remains visible on the edge to orient users as to the relationship of moving parts. This was made popular by Facebook, wherein Favorites and Settings were placed on a panel that slid in from the left when users clicked a menu icon.

Design strategies and patterns

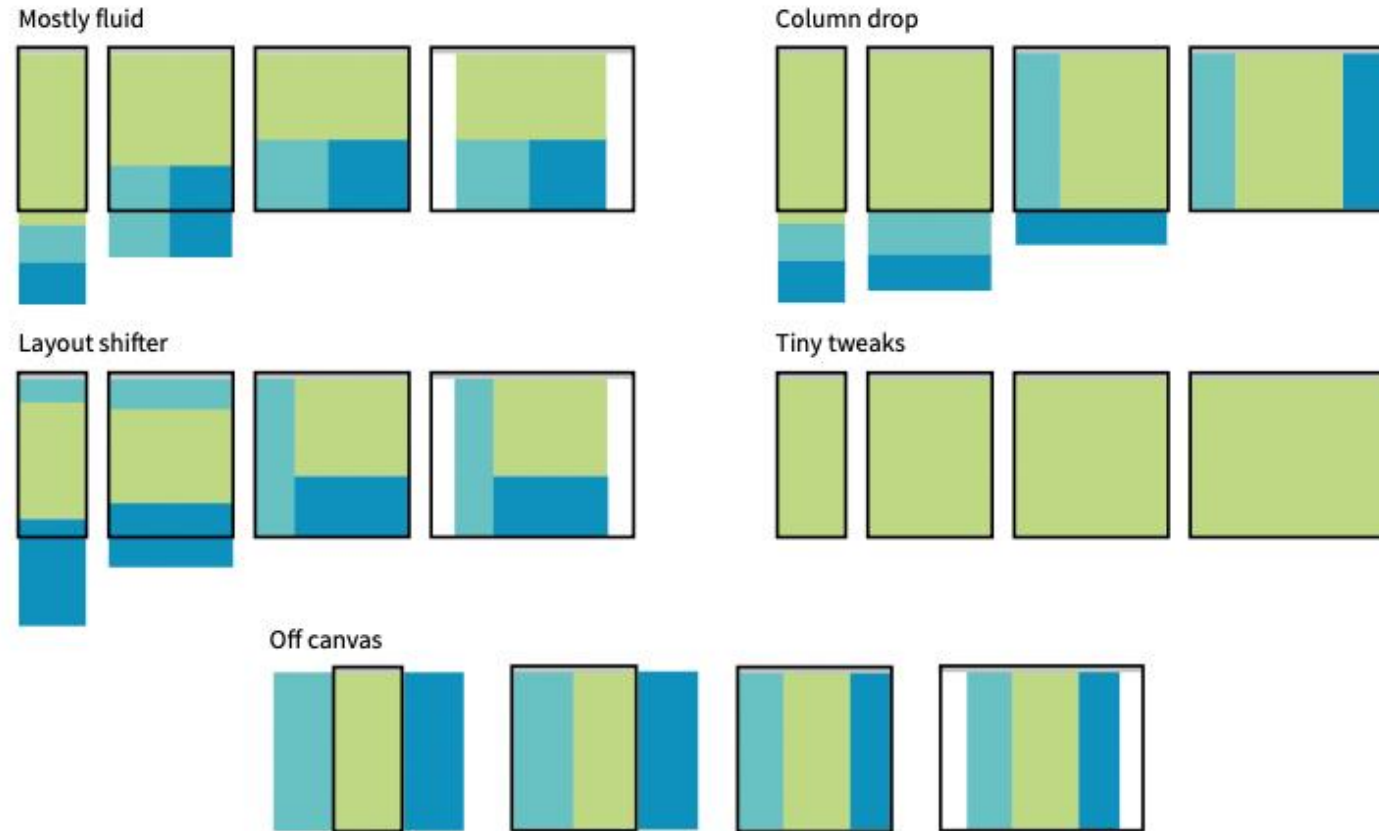


FIGURE 17-9. Examples of the responsive layout patterns identified by Luke Wroblewski.

Design strategies and patterns

Typography

Typography requires fine-tuning along the spectrum from small-screen to wide-screen views in order to keep it legible and pleasant to read. Here are a few typography-related pointers

Font face

Be careful about using fancy fonts on small screens and be sure to test for legibility. At small sizes, some fonts become difficult to read because line strokes become too light or extra flourishes become little blobs. Consider also that small screens may be connecting over cellular, so taking advantage of locally available fonts may be better for performance than requiring a web font to download.

Font size

Varying viewport widths can wreak havoc on line lengths. You may find that you need to increase the font size of text elements for wider viewports to maintain a line length of between 45 and 75 characters.

Design strategies and patterns

Line height

Line height is another measurement that you may want to tweak as screens get larger. On average, line height should be about 1.5 (using a number value for the **line-height** property); however, slightly tighter line spacing (1.2 to 1.5) is easier to read with the shorter line lengths on small screens.

Margins

On small screens, make the most of the available space by keeping left and right margins on the main column to a minimum (2–4%). As screens get larger, you will likely need to increase side margins to keep the line lengths under control and just to add some welcome whitespace to the layout. Remember to specify margins above and below text elements in em units so they stay proportional to the type.

Design strategies and patterns

Navigation

Navigation feels a little like the Holy Grail of Responsive Web Design. It is critical to get it right. Because navigation at desktop widths has pretty much been conquered, the real challenges come in re-creating our navigation options on small screens.

Top navigation

If your site has just a few navigation links, they may fit just fine in one or two rows at the top of the screen.

Priority +

In this pattern, the most important navigation links appear in a line across the top of the screen alongside a More link that exposes additional options. The pros are that the primary links are in plain view, and the number of links shown can increase as the device width increases. The cons include the difficulty of determining which links are worthy of the prime small-screen real estate.

Design strategies and patterns

Select menu

For a medium list of links, some sites use a **select** input form element. Tapping the menu opens the list of options using the select menu UI of the operating system, such as a scrolling list of links at the bottom of the screen or on an overlay. The advantage is that it is compact, but on the downside, forms aren't typically used for navigation, and the menu may be overlooked.

Link to footer menu

One straightforward approach places a Menu link at the top of the page that links to the full navigation located at the bottom of the page. The risk with this pattern is that it may be disorienting to users who suddenly find themselves at the bottom of the scroll.

Accordion sub-navigation

When there are a lot of navigation choices with sub-navigation menus, the small-screen solution becomes more challenging, particularly when you can't hover to get more options as you can with a mouse. Accordions that expand when you tap a small arrow icon are commonly used to reveal and hide sub-navigation.

Design strategies and patterns

Push and overlay toggles

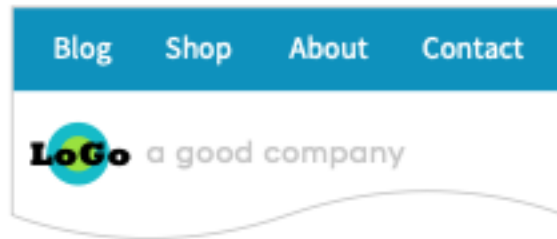
In toggle navigation, the navigation is hidden but expands downward when the menu link is tapped. It may push the main content down below it (push toggle) or slide down in front of the content (overlay toggle).

Off-canvas/fly-in

This popular pattern puts the navigation in an off-screen panel to the left or right of the main content that slides into view when you tap the menu icon.

Design strategies and patterns

Top navigation



Priority +

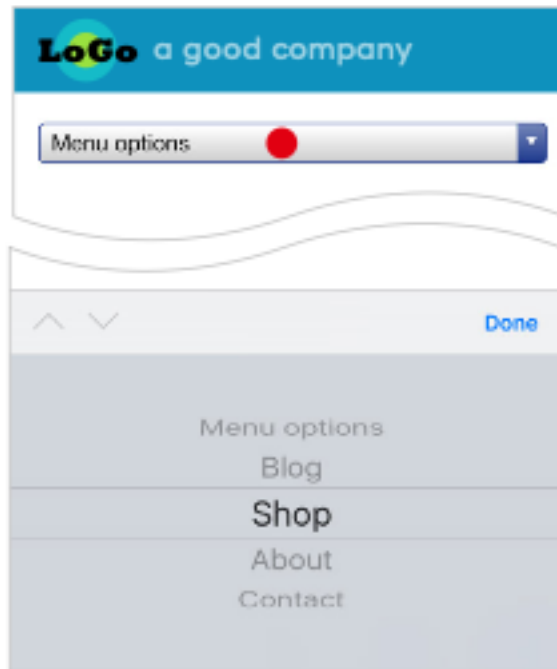


KEY

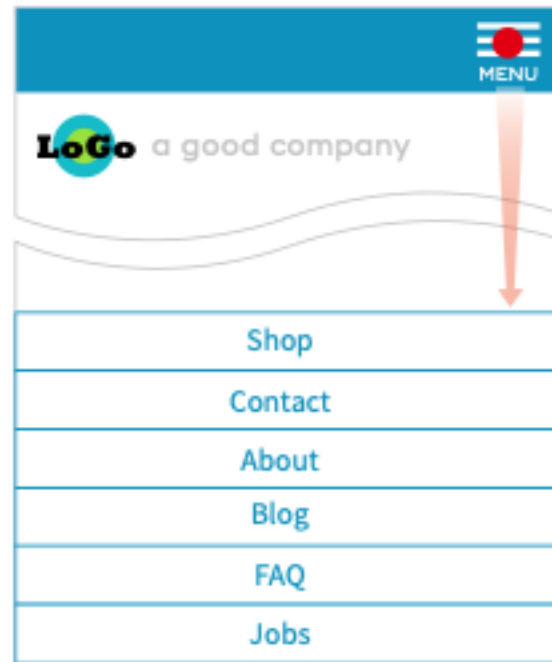


Design strategies and patterns

Select menu



Link to footer menu



Accordion sub-navigation



Design strategies and patterns

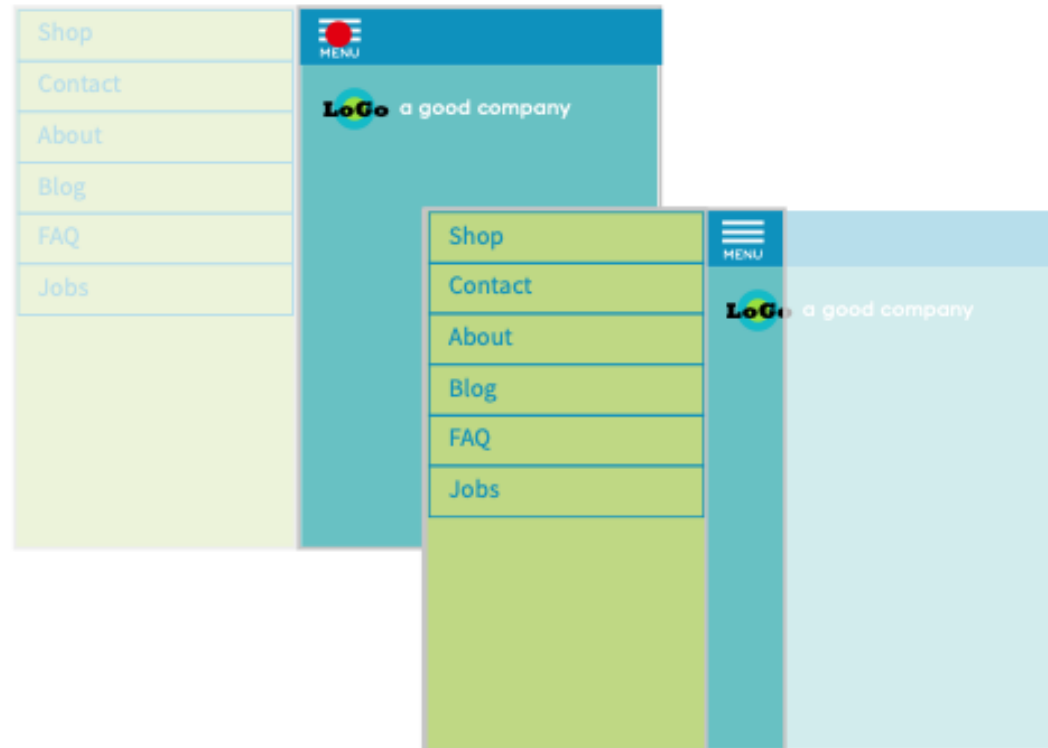
Overlay toggle (covers top of screen)



Push toggle (pushes content down)



Off-canvas/fly-in



Design strategies and patterns

Images

Images demand particular consideration in responsive designs. Here's a short rundown of some of the major concerns, the most of which should be familiar:

- Make use of responsive image markup techniques to deliver numerous copies of crucial pictures in different sizes and resolutions.
- As the default, serve the shortest version to save needless data downloads.
- Check that critical image detail is not lost at lesser sizes. For tiny displays, consider using a cropped version of the image.
- Avoid using text in visuals, but if you must, give additional versions with bigger font for tiny displays.

Design strategies and patterns

Special Content

Some of our popular page components face issues when it comes to fitting on smaller devices since we don't have the benefit of wide-open, desktop viewports:

Forms

Forms frequently require some tinkering to suit the given area. Flexbox is an excellent tool for providing conditional wrapping and flexibility to form fields and labels. A simple web search will yield several excellent tutorials. Also, for every screen size, ensure that your form is as efficient as possible, with no unnecessary information. Finally, keep in mind that form inputs will be utilized with fingers rather than mouse pointers, so make the target size larger by including adequate padding or margins and making labels tappable to pick an input.

Design strategies and patterns

Tables

One of the most difficult aspects of small-screen design is dealing with massive data tables. Not surprisingly, there are several options because there are numerous table forms.

Interactive Elements

A large embedded map is fantastic on a site's desktop view, but it is less useful when it is the size of a postage stamp. Consider if any interactive elements should be replaced by other means of accomplishing the same objective. In the case of maps, providing a link to a map might cause the device's native mapping software to launch, which is optimized for small-screen viewing. Other interactive elements, such as carousels, can be scaled down to fit smaller viewports.

Testing The RWD

Real Device

There is no replacement for testing a site across a wide range of real-world devices and operating systems. Testing on actual devices tells you how your site functions in addition to how it appears. How quickly does it load? Are the links simple to access? Do all of the interactive features function properly? Do they actually work?

Web development firms may have a device lab with iPhones and iPads of various sizes, Android smartphones and tablets of various sizes, and Macs and PCs with current operating systems (Windows and Linux) that designers and developers may use for testing sites. The size of the device lab is determined by the budget.

Testing The RWD

Emulators

If a particular device is out of your reach, you could use an emulator, a desk- top application that emulates mobile device hardware and operating systems. The emulator presents a window that shows exactly how your site would behave on that particular device. Emulators require a lot of space on your computer and they can be buggy, but it is certainly better than not testing on that device at all.

Testing The RWD



The Android Emulator lets you set up a wide variety of phones, televisions, wearables, and tablets for testing. I chose a Nexus 5X.



The Nexus 5X emulator displays an image of the device at actual size. All of the buttons work as they would on the phone.



The Bakery page viewed on the Nexus 5X emulator.

FIGURE 17-18. Examples of the Android Emulator (download at developer.android.com/studio/index.html).

Thank You

Alfred Tenggono, S.Kom., M.Kom.

alfred.tenggono@jiu.ac

Reference

- Learning Web Design, Jennifer Niederst Robbins, O'Reilly Media, Inc., 2018, ISBN: 978-1-491-96020-2