CSS Breakpoints for Responsive Design

© L. Hernández, 2024

Introduction

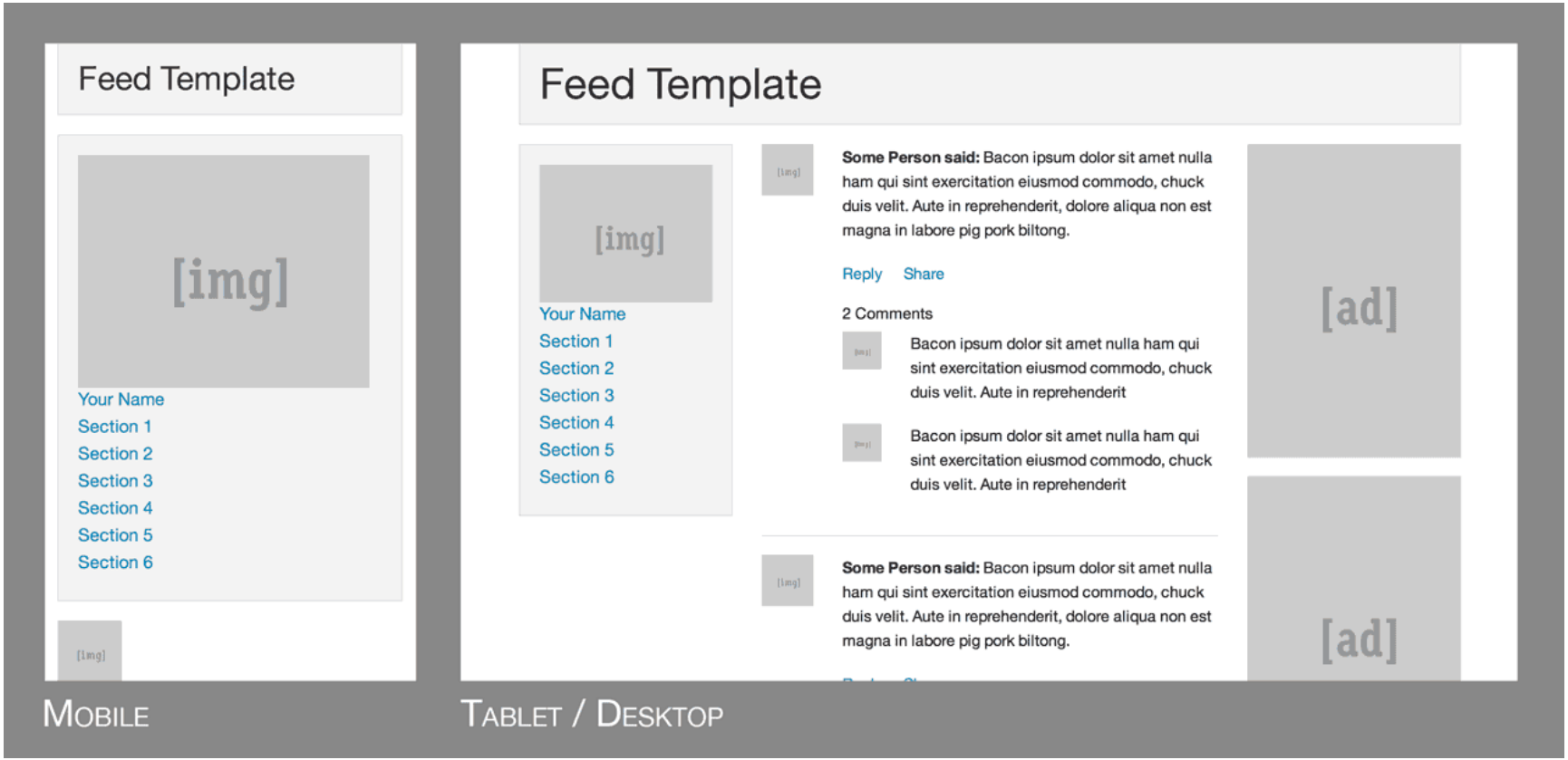
Users access the web with devices that have different screen sizes.

We need to implement a way to display websites properly on those screens to ensure usability.

12
9
6
4
3
2

Introduction

Responsive web design is a CSS-based design technique that ensures webpages render properly across all screen sizes and resolutions while ensuring usability.


Feed Template
[img]
Your Name
Section 1
Section 2
Section 3
Section 4
Section 5
Section 6
[img]
Mobile
Feed Template
[img]
Your Name
Section 1
Section 2
Section 3
Section 4
Section 5
Section 6
[img]
Some Person said: Bacon ipsum dolor sit amet nulla ham qui sint exercitation eiusmod commodo, chuck duis velit. Aute in reprehenderit, dolore aliqua non est magna in labore pig pork biltong.
Reply Share
2 Comments
[img]
Bacon ipsum dolor sit amet nulla ham qui sint exercitation eiusmod commodo, chuck duis velit. Aute in reprehenderit
[img]
Bacon ipsum dolor sit amet nulla ham qui sint exercitation eiusmod commodo, chuck duis velit. Aute in reprehenderit
[img]
Some Person said: Bacon ipsum dolor sit amet nulla ham qui sint exercitation eiusmod commodo, chuck duis velit. Aute in reprehenderit, dolore aliqua non est magna in labore pig pork biltong.
[ad]
[ad]
Tablet / Desktop

Evolution of responsive design

HTML is fundamentally responsive.

If we create a webpage using only HTML and resize the window, the browser will automatically adjust the text to fit the viewport.

But the content won't look good on every screen.

Evolution of responsive design

For example, long lines of text can be difficult to read on a wide monitor.

Similarly, if the line length is reduced with CSS by creating columns or adding a margin, the content may look squashed when viewed on a mobile device.

0 30 40 60

Too narrow

Tokyo, Japan's busy capital, mixes the ultramodern and the traditional, from neon-lit skyscrapers to historic temples. The opulent Meiji Shinto Shrine is known for its towering gate and surrounding woods.

Ideal range

Tokyo, Japan's busy capital, mixes the ultramodern and the traditional, from neon-lit skyscrapers to historic temples. The opulent Meiji Shinto Shrine is known for its towering gate and surrounding woods.

A little wide

Tokyo, Japan's busy capital, mixes the ultramodern and the traditional, from neon-lit skyscrapers to historic temples. The opulent Meiji Shinto Shrine is known for its towering gate and surrounding woods.

Too wide

Tokyo, Japan's busy capital, mixes the ultramodern and the traditional, from neon-lit skyscrapers to historic temples. The opulent Meiji Shinto Shrine is known for its towering gate and surrounding woods.

Evolution of responsive design

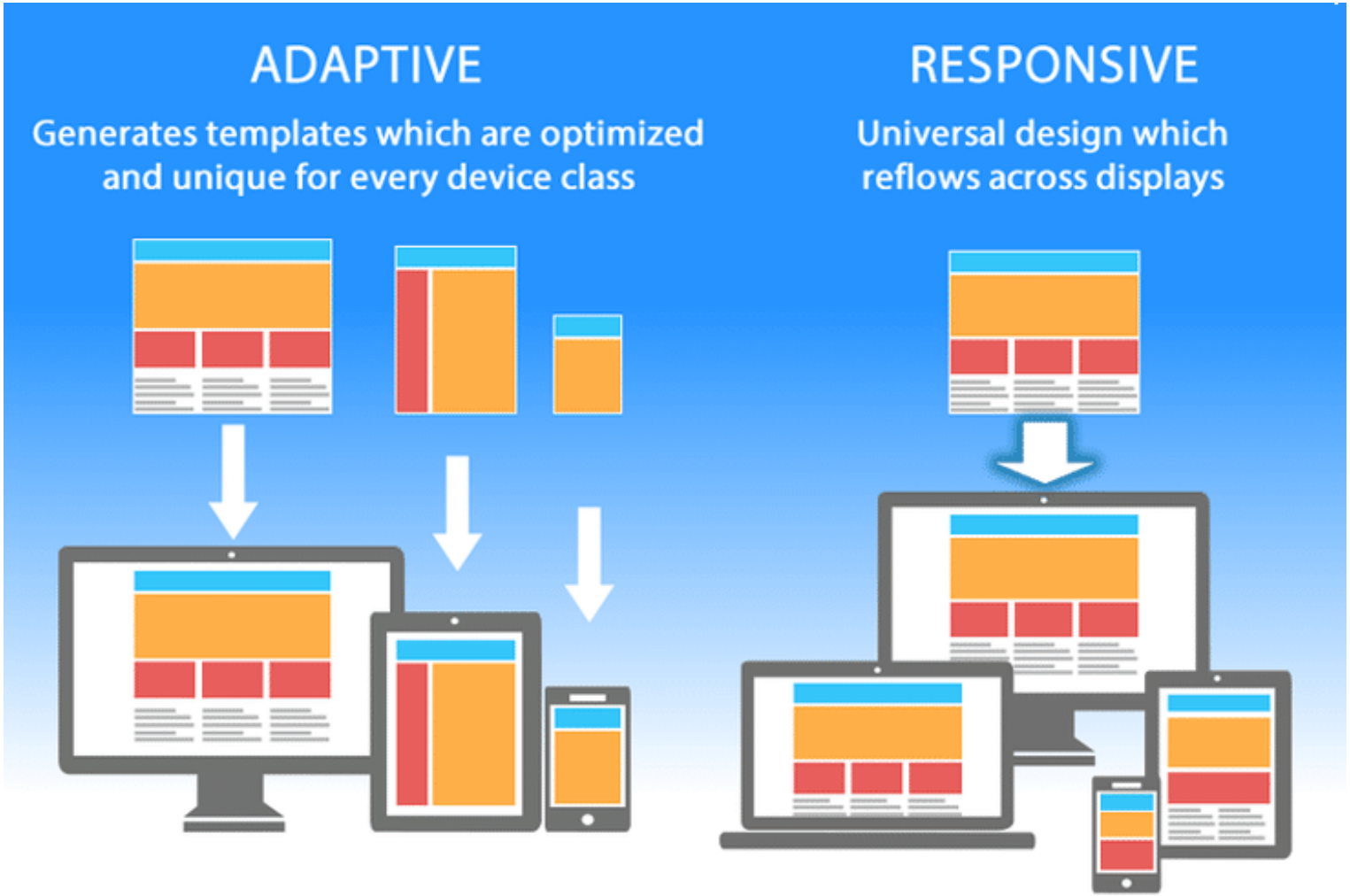
We have to intervene to adapt the style to the screen based on the content and layout of the webpage.

Mobile Tablet Desktop

Responsive web design

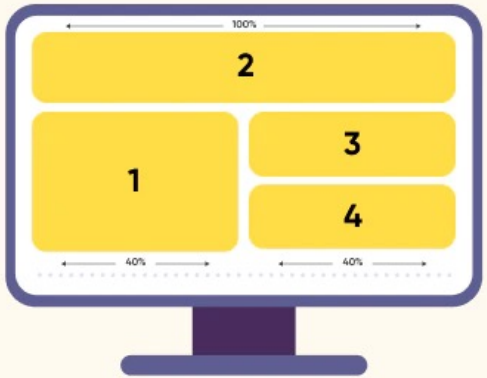
We can use **float** for layouts and media queries to query browser width or height to create layouts for different breakpoints using CSS.

We could also use fluid grids, fluid images, and media queries to create responsive content.


ADAPTIVE
Generates templates which are optimized and unique for every device class
RESPONSIVE
Universal design which reflows across displays

1 Media Queries

A way to apply CSS rules to the page based on the size of the displaying browser.



2 Fluid Grids

A percentage based design that adapts to the screen size accordingly.

Responsive web design

A breakpoint is a point, usually a specific width, at which a webpage's style is changed to ensure best possible user experience.

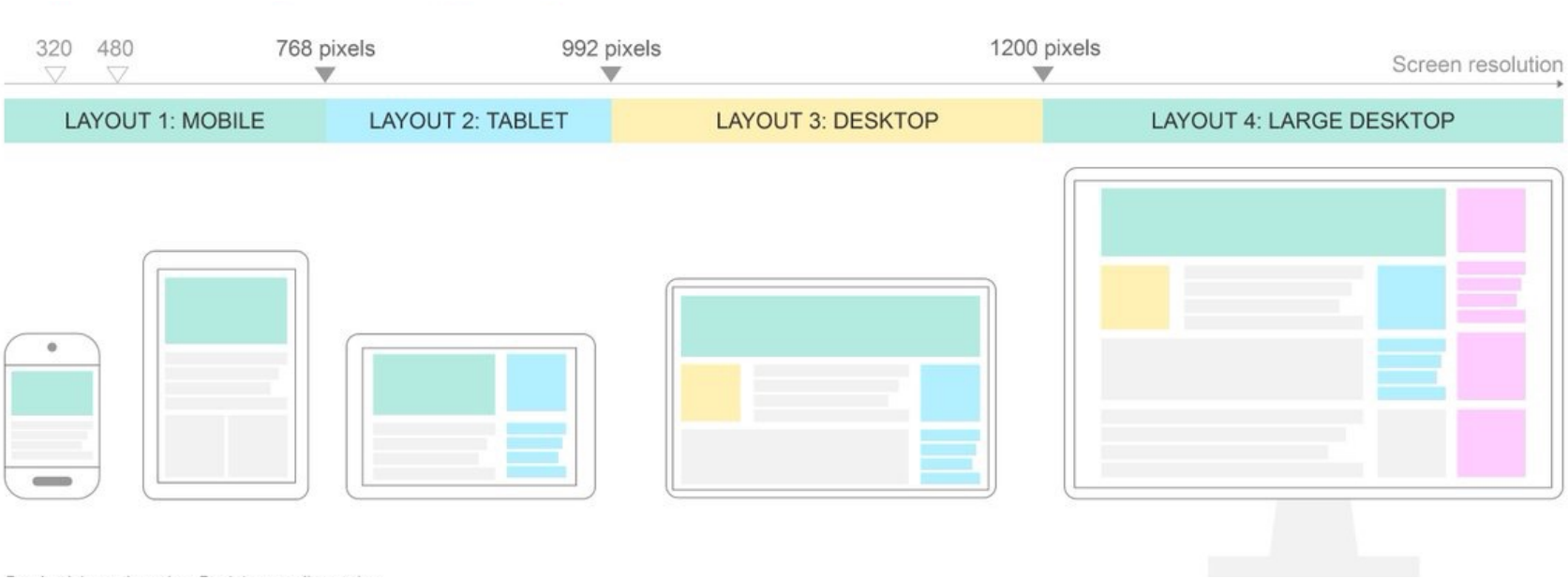
There is no strict rule or standard to define responsive breakpoints because there are so many different screen sizes.

Responsive web design

Creating more device breakpoints offers best results but it increases web design time and delays product delivery.

Creating fewer breakpoints boosts responsive design time but generates fewer layout variations affecting usability.

Responsive web design



0-480px

Smart Phones

481-768px

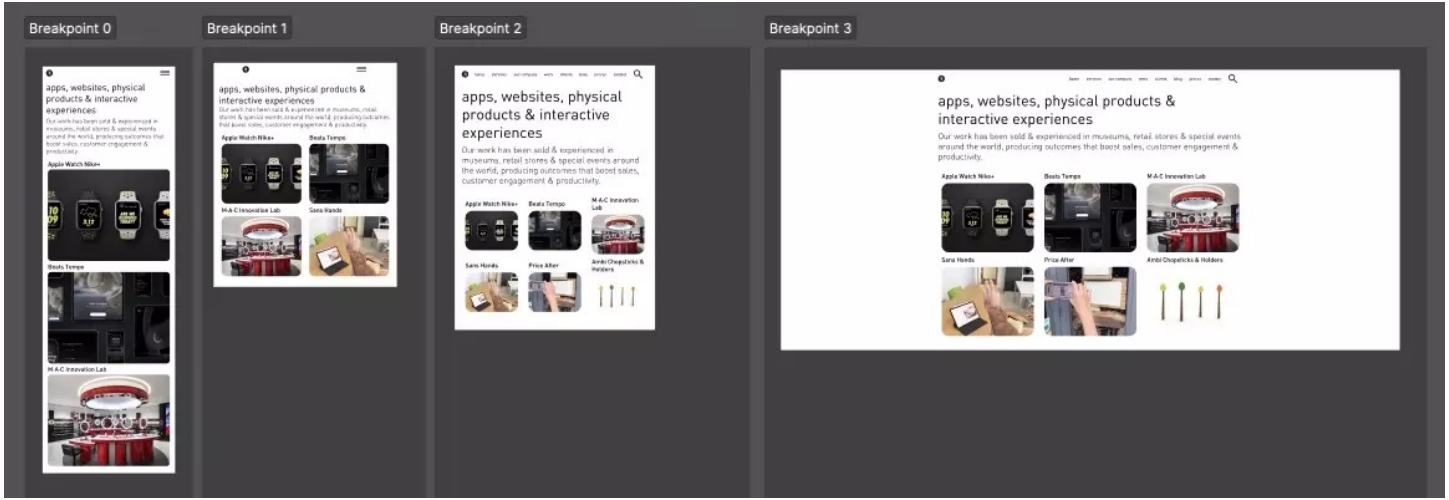
Tablets

769-1279px

Laptops

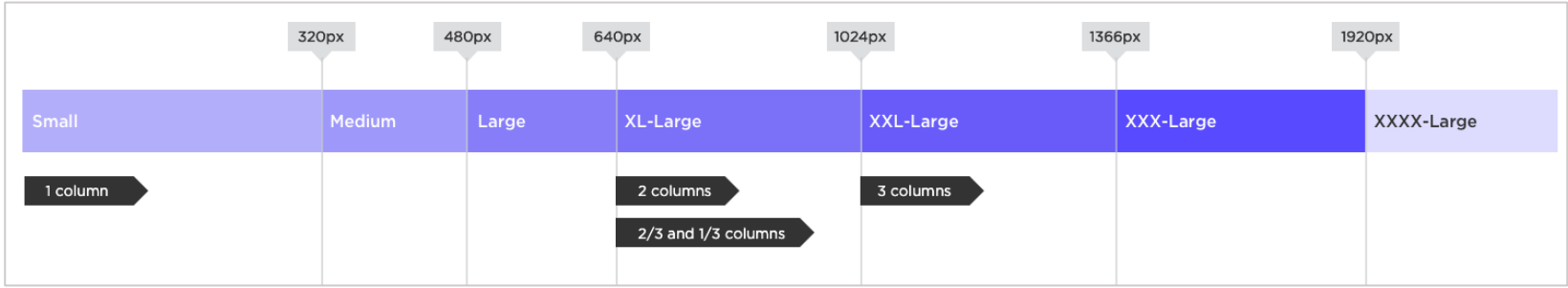
>= 1280px

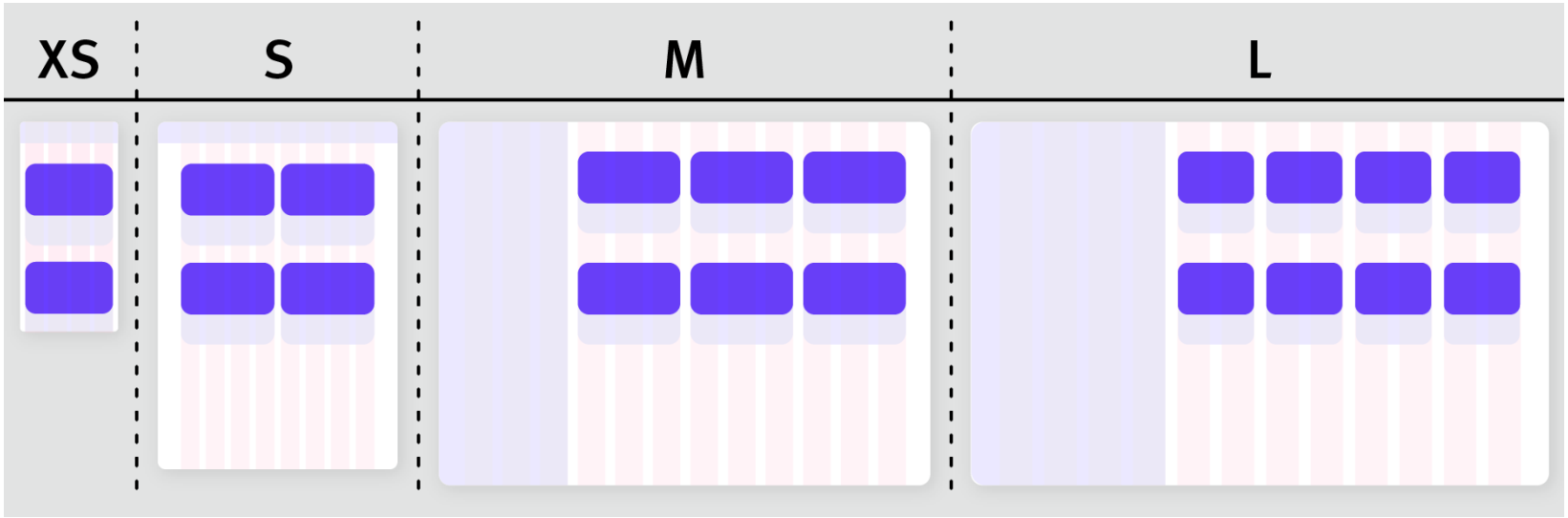
Desktops



Responsive web design

CSS frameworks like Bootstrap rose in popularity because they provided us with responsive grid systems that use pre-defined breakpoints to implement responsive layouts.


320px
480px
640px
1024px
1366px
1920px
Small
Medium
Large
XL-Large
XXL-Large
XXX-Large
XXXX-Large
1 column
2 columns
2/3 and 1/3 columns
3 columns


XS
S
M
L

Responsive web design

Inbuilt CSS layout implementation methods have responsive capabilities such as Flexbox and CSS Grid.

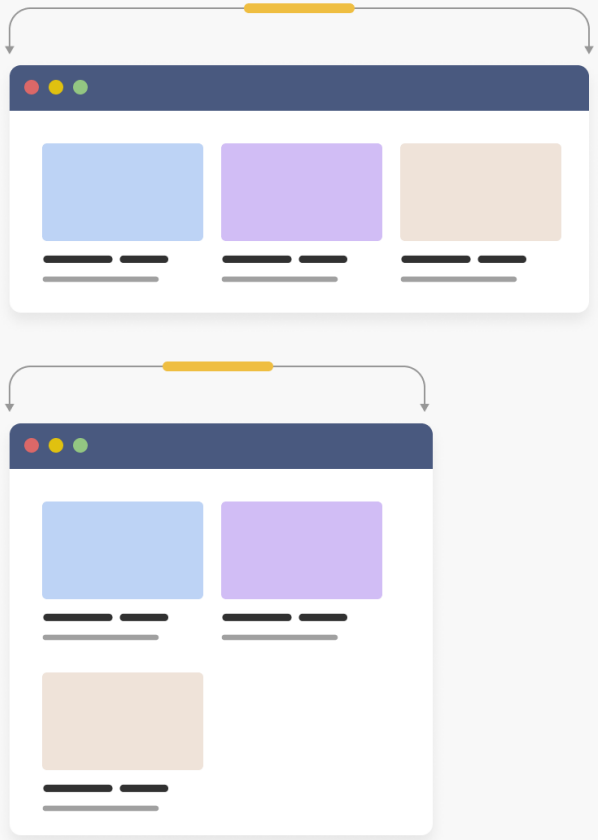
Other modern methods have been developed to make content responsive without even using responsive layout libraries.

Responsive web design

- clamp() function: Allows typography and spacing to be responsive to viewport width.
- Logical properties: Permits spacing to be responsive to website.
- Container queries: Enables an HTML element to be responsive based on its dimensions.

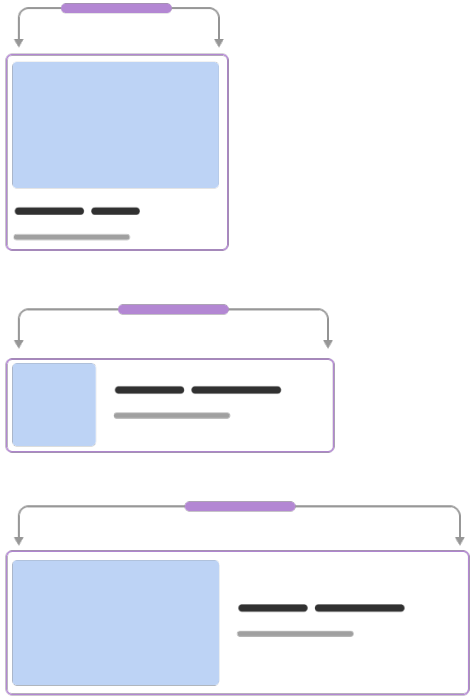
Media Query

Viewport width



Container Query

Container width

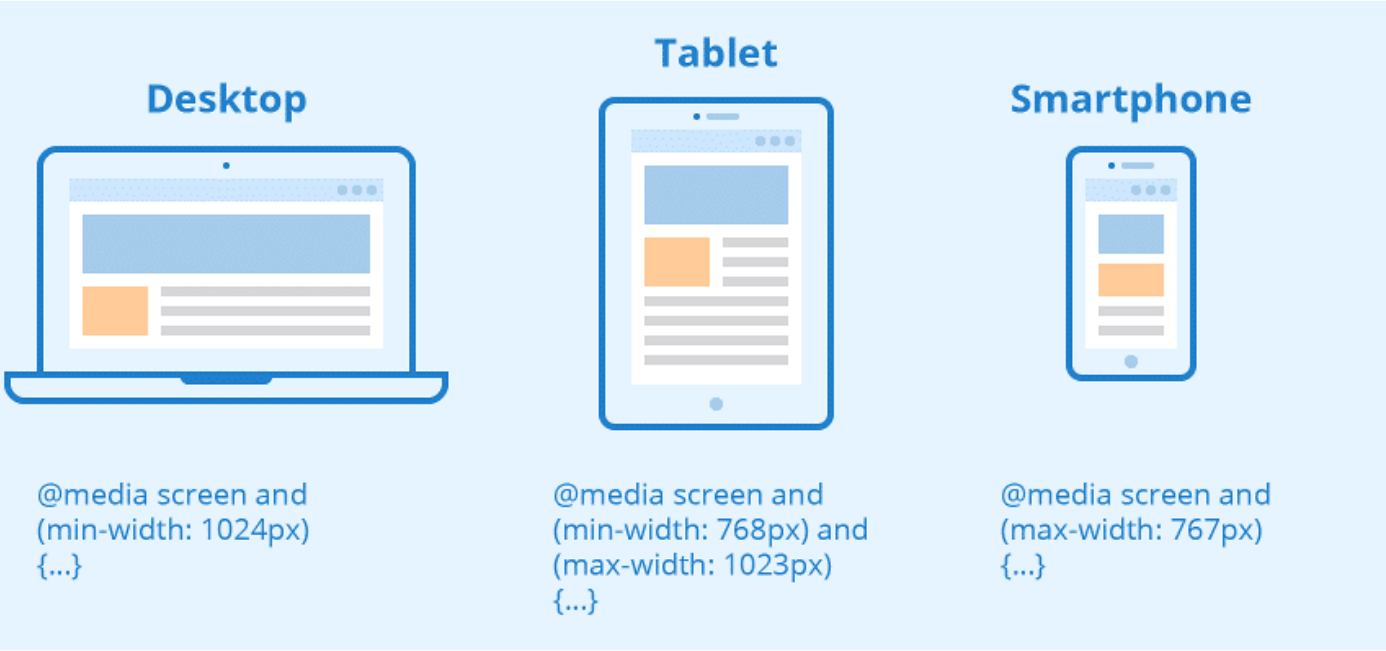


Media queries

Media queries are useful when we want to modify layout or appearance of a site depending on specific system or browser characteristics.

Browser characteristics such as screen resolution of device or browser viewport width/height.

Media queries



Media queries

The media type (all, print, or screen) is optional; it is assumed to be all if omitted.

This is the common syntax for a CSS media query.

The logical operators not, and, only, and or can be used to compose a complex media query.

```
@media <type> <operator> (feature) ... {
  /* CSS rules */
}
```

Media queries

For responsive design, **min-width** and **max-width** are the most commonly used media features.

These media features can help us create responsive breakpoints based on specific width ranges of the viewport.

Media queries

For example, this CSS code will apply styles only if browser's viewport width is equal to or less than 80em.

```
@media (max-width: 80em) {
  /* CSS rules */
}
```

New media queries specification includes some syntax improvements.

```
@media (width <= 80em) { ... }
```

Choose breakpoints

There are two main approaches when choosing CSS breakpoints: one is based on devices and other is based on content.

Breakpoints based on devices

With the variety of devices available, determining breakpoints based on screen sizes is challenging.

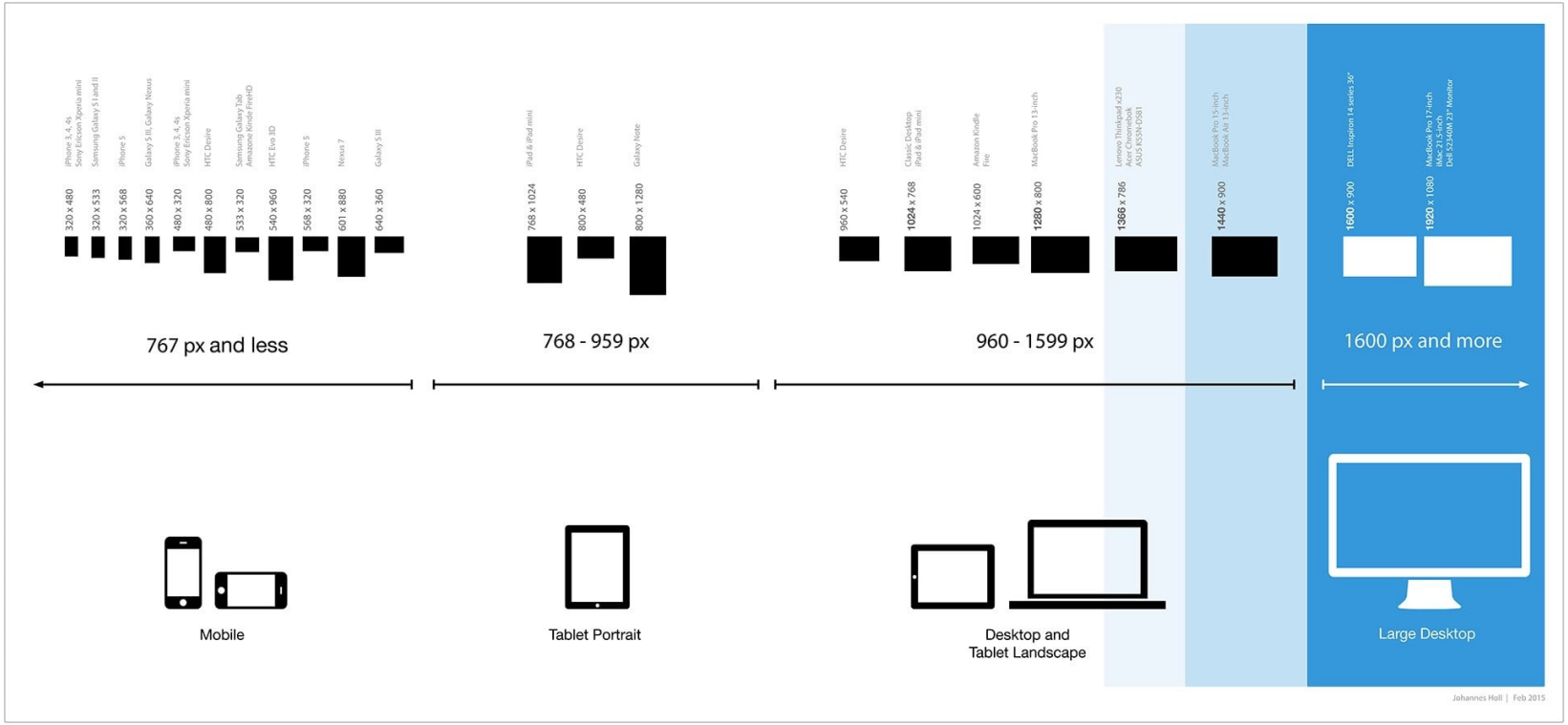
This approach is really not feasible to maintain.

Breakpoints based on devices

To simplify this approach, we could group devices based on a range of sizes.

The most common way is to group devices based on form factor (e.g., mobile devices, tablets, laptops, etc.).

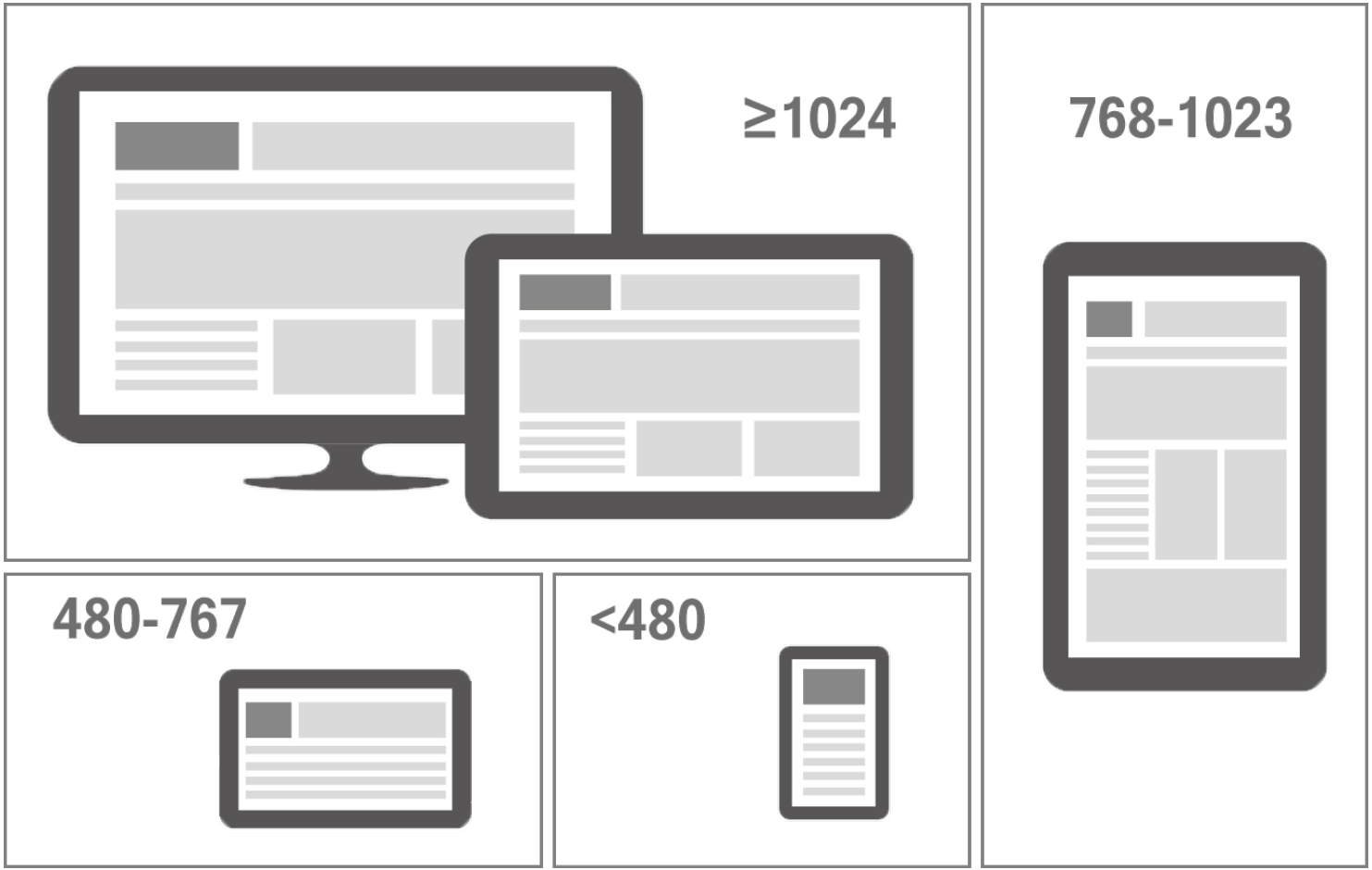
Breakpoints based on devices



Breakpoints based on devices

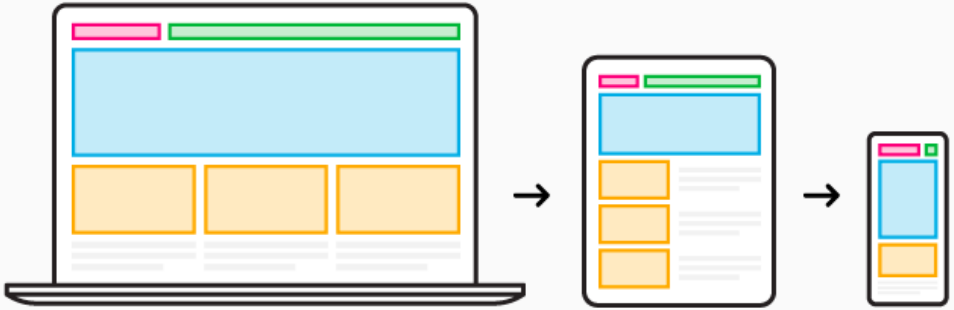
Let's check several common breakpoints that most websites nowadays use.

This CSS example uses some breakpoints with a mobile-first design strategy (default style is for the smallest screen group).

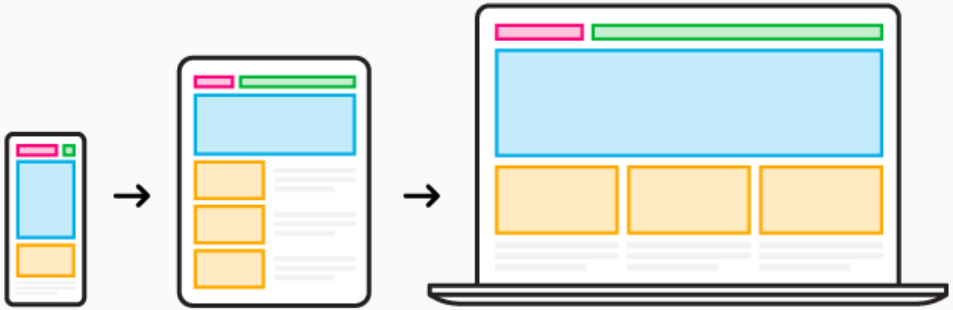

≥1024
768-1023
480-767
<480

```
/* Default: Extra-small devices such as small phones
(less than 640px) */

/* Small devices such as large phones (640px and up) */
@media only screen and (min-width: 640px) {...}

/* Medium devices such as tablets (768px and up) */
@media only screen and (min-width: 768px) {...}

/* Large devices such as laptops (1024px and up) */
@media only screen and (min-width: 1024px) {...}

/* Largest devices such as desktops (1280px and up) */
@media only screen and (min-width: 1280px) {...}
```

RESPONSIVE DESIGN



MOBILE FIRST DESIGN



A CSS example that only defines two breakpoints with a desktop-first design strategy (default style is for the largest screen group).

```
/* Default: Large devices such as laptops, computers (greater than 1024px) *

/* Medium devices such as tablets (1024px or lesser) */
@media only screen and (max-width: 1024px) {...}

/* Small devices such as phones (768px or lesser) */
@media only screen and (max-width: 768px) {...}
```

Breakpoints based on devices

This example renders a responsive login section for desktop, tablet, and mobile screens.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport"
        content="width=device-width, initial-scale=1.0" />
    <meta http-equiv="X-UA-Compatible" content="ie=edge" />
    <title>Responsive design breakpoints example</title>
    <style>
      ...
    </style>
  </head>
  ...
```

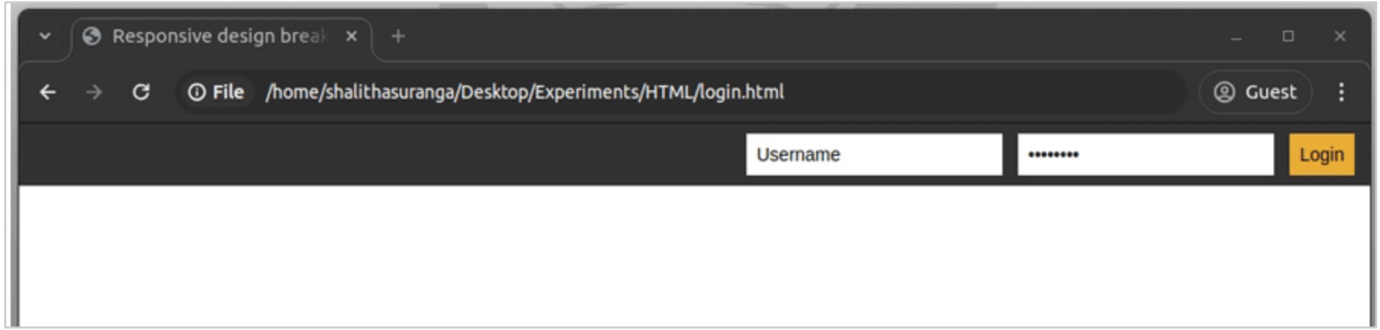
```
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
.form-box {
  display: flex;
  justify-content: flex-end;
  gap: 8px;
  padding: 8px;
  background-color: #333;
  text-align: center;
}
...
```

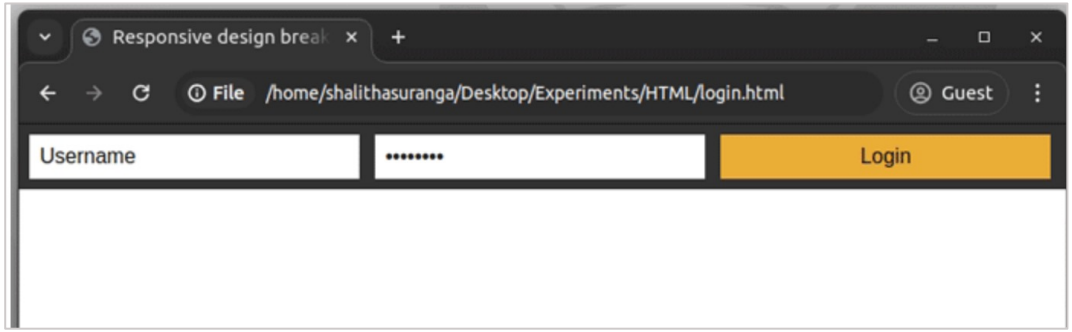
```
...
.form-box input,
.form-box button {
  padding: 8px;
  margin-right: 4px;
  font-size: 14px;
}
.form-box input {
  outline: none;
  border: none;
}
...
```

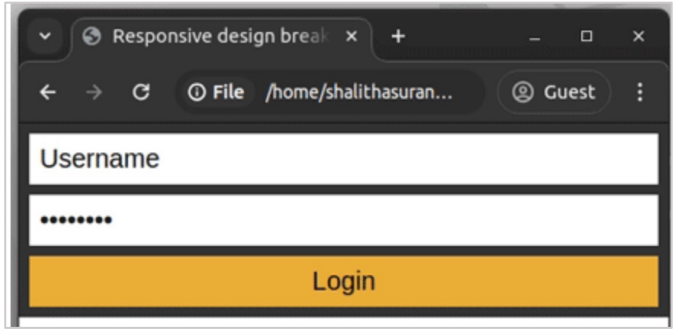
```
...
.form-box button {
  border: none;
  background-color: #edae39;
}
@media only screen and (max-width: 1024px) {
  .form-box input,
  .form-box button {
    display: block;
    width: 100%;
    font-size: 16px;
  }
}
...
```

```
...
@media only screen and (max-width: 768px) {
  .form-box {
    flex-direction: column;
  }
  .form-box input,
  .form-box button {
    display: block;
    width: 100%;
    font-size: 20px;
  }
}
```

```
<!DOCTYPE html>
<html lang="en">
  <head>
    ...
  </head>
  <body>
    <div class="form-box">
      <input type="text" value="Username" />
      <input type="password" value="Password" />
      <button>Login</button>
    </div>
  </body>
</html>
```


Responsive design brea
File
/home/shalithasuranga/Desktop/Experiments/HTML/login.html
Guest
Username
Login


Responsive design brea
File
/home/shalithasuranga/Desktop/Experiments/HTML/login.html
Guest
Username
Login

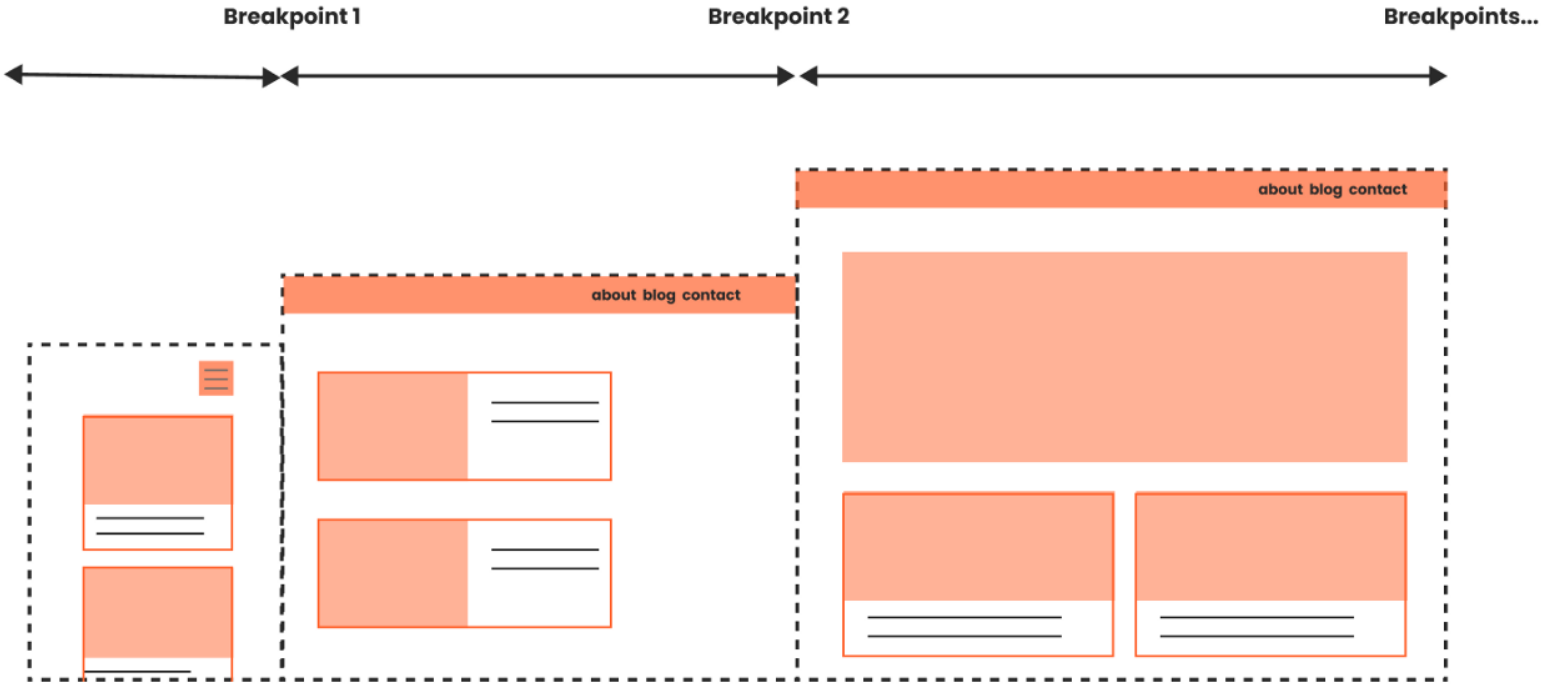

Responsive design brea
File
/home/shalithasuran...
Guest
Username
Login

Breakpoints based on content

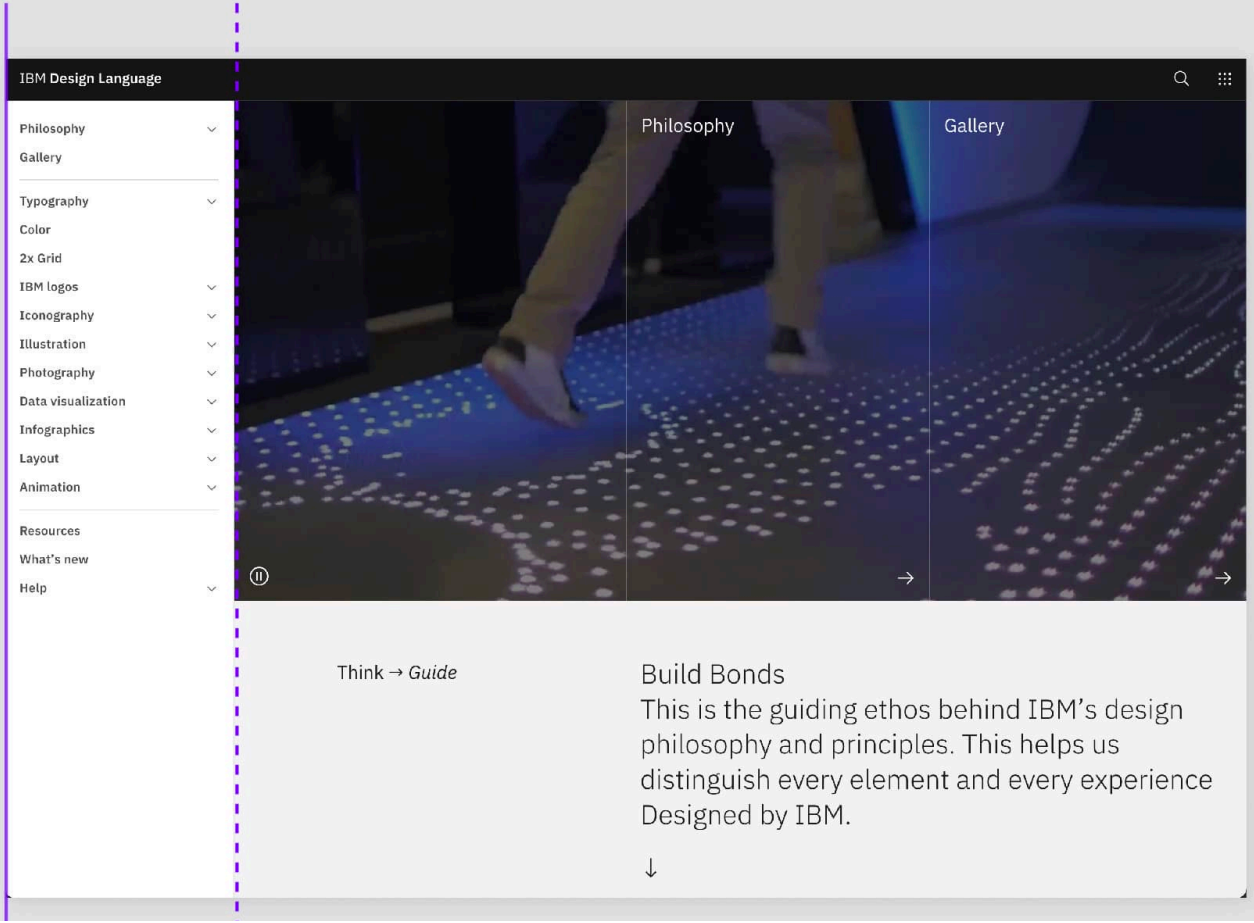
This approach is based on changing the design at the point where the content starts to break in some way.

If line lengths become too long, or if a section becomes too squashed, that's where we need to consider changing the style.

Breakpoints based on content

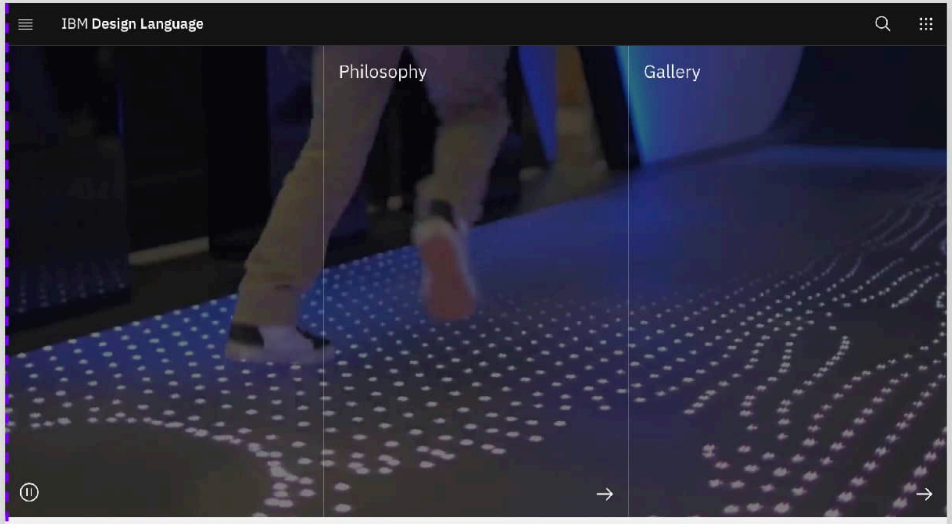


Large Screen Size



Medium Screen Size

COLLAPSED → LEFT NAVIGATION



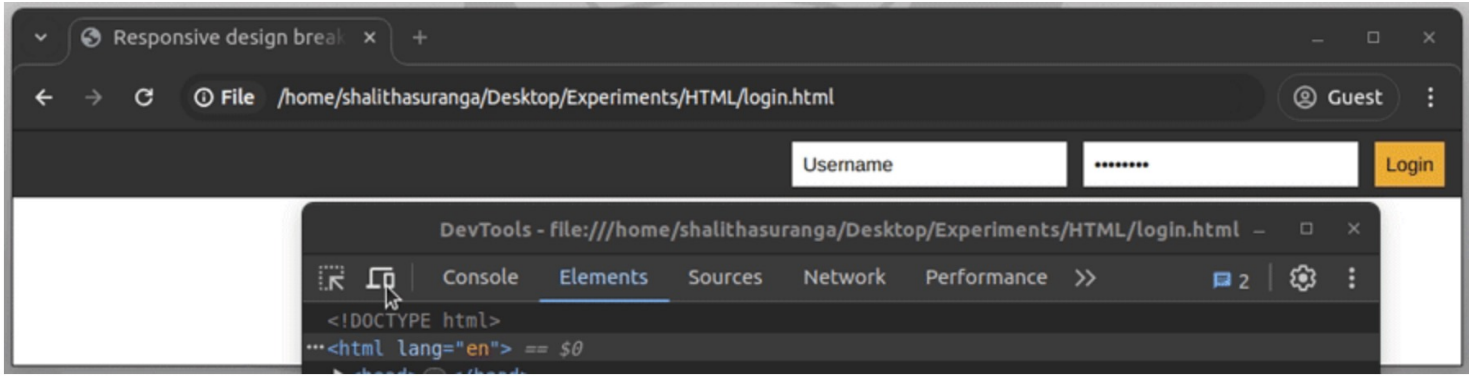
NN/g

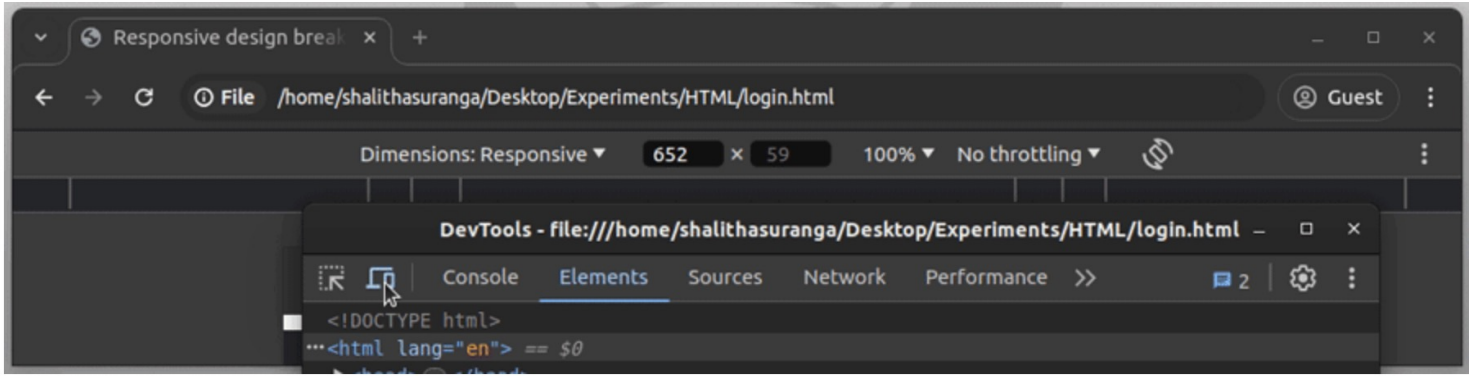
Breakpoints based on content

Responsive mode in browser developer tools (Firefox and Chrome DevTools) is very useful for working out where our breakpoints should go.

We can easily make the viewport smaller or larger to see where the content style could be improved.

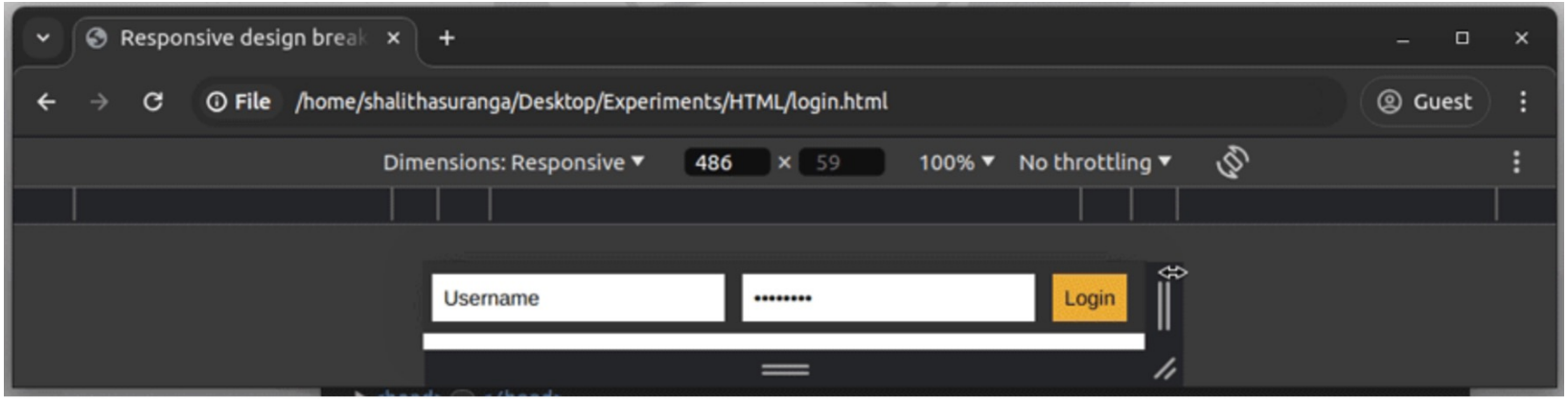
Breakpoints based on content

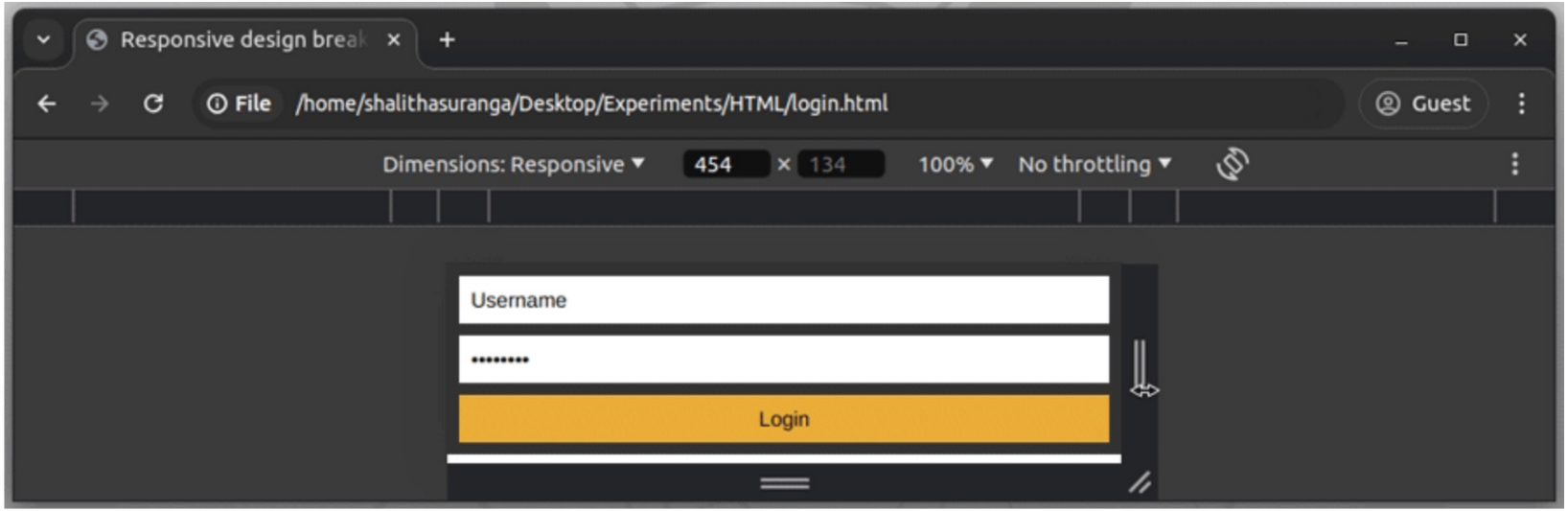




Here, the login form is not correctly getting rendered when the width is less than 486px, so we can create a breakpoint using max-width.

```
@media only screen and (max-width: 485px) {
  .form-box {
    flex-direction: column;
  }
  .form-box input,
  .form-box button {
    display: block;
    width: 100%;
  }
}
```


Responsive design break
File
/home/shalithasuranga/Desktop/Experiments/HTML/login.html
Guest
Dimensions: Responsive
486
59
100%
No throttling
Username
Login


Responsive design break
File
/home/shalithasuranga/Desktop/Experiments/HTML/login.html
Guest
Dimensions: Responsive
454
134
100%
No throttling
Username
Login