# Today's Agenda

- Accessibility
- Grids layout
- Responsive design
- Figma Activity
- **Q&A**



# Accessibility

(Design for Everyone)



# What is Accessibility?

Accessibility is not only the right thing to do, but often also brings benefits to all users.

# Types of Accessibility Issues:

- **Visual** (e.g., color blindness)
- Motor/mobility (e.g., wheelchair-user concerns)
- Auditory (hearing difficulties)
- **Seizures** (especially photosensitive epilepsy)
- **Learning/cognitive** (e.g., dyslexia)

#### 1- Provide Text Alternatives

Alternative text (alt text) is intended as an alternative for non-text content such as images

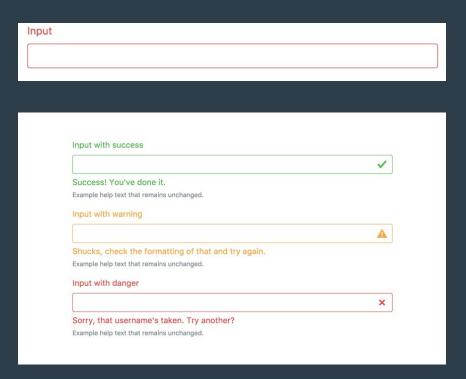
#### Affected audiences

- People who are blind
- People with certain cognitive disabilities
- People who rely on assistive technology for any reason not listed here
- People with slow internet connections or limited mobile data plans
- Search engines



# 2. Avoid Using Only Color to Convey Information

Color only is not the best way to convey information, as shown by the following example. Below, a red border is used to highlight form fields that have not been filled in correctly. Consequently, if a user is color blind, they won't be able to see this indication, so it becomes difficult to understand which fields have errors:



#### 3. Add Transcripts and captions

Provide text transcripts and captions whenever a sound appears (synchronized with video content, live captions, cues in addition to sound in apps).

Helps people with hearing difficulties



# 4. Ensure sufficient contrast between text and its background.

According to the WCAG, the contrast ratio between text and a text's background should be at least 4.5 to 1. If your font is at least 24 px or 19 px bold, the minimum drops to 3 to 1.

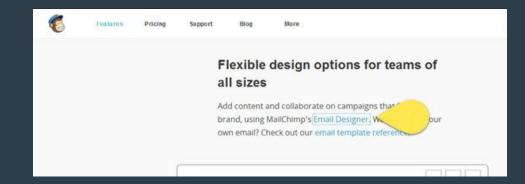
This helps users with low vision, color blindness, or worsening vision see and read the text on your screen.

Check Color Contrast tool



5. Anything that can be done with a mouse, should also work with a keyboard.

Certain users navigate the internet **using the keyboard rather than the mouse**. For example, people with a **motor impairment** 



- Obvious Keyboard Focus
- All Interactive Elements Are Accessible
- Consider a Skip Navigation Link



## **Grid Layout**

**a grid** is a structure comprising a series of lines (vertical or intersecting) that divide a page into columns or modules. This structure helps designers to arrange content on the page.

grids improve page readability and scannability and allow people to quickly get where they need to go.



Learn more about Grids

### **Main Elements of Grid**

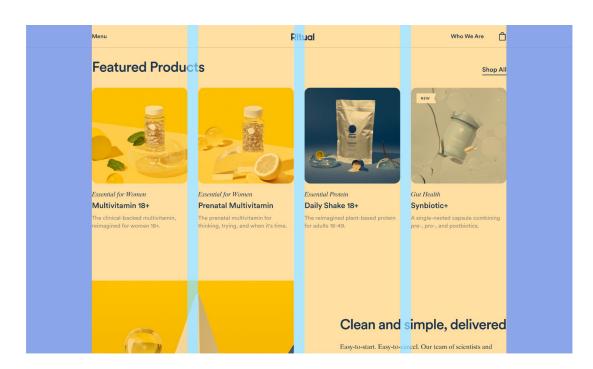
- **1. Columns:** Columns take up most of the real estate in a grid. Elements and content are placed in columns.
- **2. Gutters:** The gutter is the space between columns that separates elements and content from different columns.
- **3. Margins:** this refers to the left and right outermost areas on the screen. Content does not live in the margins of a grid.



**Modules** are the building blocks of any grid. They are the spaces created between the horizontal lines and vertical lines.

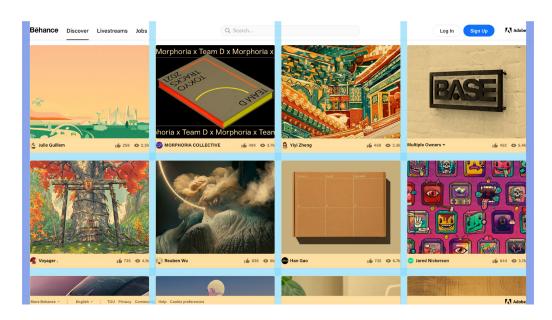
### 1. Column Grid

involves dividing a page into vertical columns. UI elements and content are then aligned to these columns.



### 2. Modular Grid

extends the column grid further by adding rows to it. This intersection of columns and rows make up modules to which elements and content are aligned. Modular grids are great for ecommerce and listing pages, as rows are repeatable to accommodate browsing.



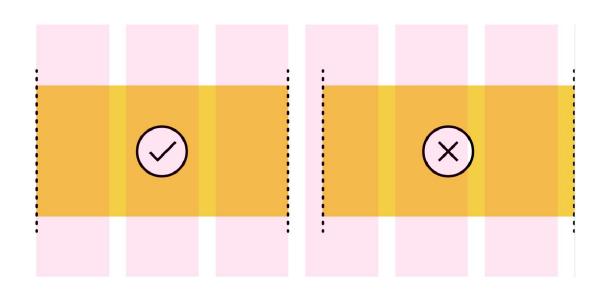
### 3. Hierarchical Grid

Content is organized by importance using columns, rows, and modules. The most important elements and pieces of content take up the biggest pieces of the grid.



### **Tips**

- Always place content within columns, not gutters.
- Consider using an 8px grid system. For most common devices, the screen size in pixels is a multiple of 8. Keeping grid-component values at a multiple of 8 will generally make it easier to scale and implement a grid.
- Tip: Not every element in the component has to align with the vertical column. Align the bounds of the components with the columns.



# What type is this?



# What type is this?



## What type is this?

