FALL 2021 / ONLINE

INTERACTIVE DEVELOPMENT

Friday, September 24

WARM-UP QUESTION:

WHAT'S YOUR FAVORITE BOOK?

“HELLO WORLD”

HW1.0 GITHUB HOMEWORK RECAP

A03.1 RECAP

DEMOS + DISCUSSION

NOTE: WE’RE GONNA TACKLE

JAVASCRIPT

NEXT FRIDAY.

DISPLAY MODES

INLINE vs BLOCK-LEVEL

HTML elements are usually "inline" or "block-level" elements.

An **inline** element occupies only the space bounded by the tags that define it.

A **block-level** element occupies the entire space of its parent element (container), thereby creating a "block".

SPAN vs DIV

Both elements are generic containers for content.

- A `span` tag is inline.
- A `div` (*document **div**isions*) tag is block-level.

Neither has any effect on the content or layout until styled in some way using CSS besides its inherited display value.

See this demo: http://maxx.link/inline-block

OTHER DISPLAY MODES

`display:none`

Turns off the display of an element so that it has no effect on layout (the document is rendered as though the element did not exist). All descendant elements also have their display turned off.

`display:flex`

The element behaves like a block element and lays out its content according to the flexbox model.

`display:grid`

The element behaves like a block element and lays out its content according to the grid model.

`display:table`

These elements behave like HTML <table> elements. It defines a block-level box.

FLEXBOX

Flexbox is a one-dimensional layout method for arranging items in rows or columns.

Items **flex** (expand) to fill additional space or shrink to fit into smaller spaces.

Refer to these resources:

- MDN: Basic Concepts of Flexbox
- CSS Tricks: Guide to Flexbox

POSITIONS

CSS POSITIONING

`position:static (default)`

The element is positioned according to the normal flow of the document. The top, right, bottom, left, (TRBL) and z-index properties have **no effect**.

`position:relative`

TRBL only affect the element. The offset does not affect the position of any other elements; thus, the space given for the element in the page layout is the same as if position were static.

`position:absolute`

The element is removed from the normal document flow, and no space is created for the element in the page layout. Its final position is determined by the values of top, right, bottom, and left.

`position:fixed`

Similar to absolute, but its position is relative only to the viewport. It will not move regardless of screen position.

See this demo: http://maxx.link/css-positions

TOP/LEFT/RIGHT/BOTTOM

The top and bottom CSS properties specifies the vertical position of a positioned element.

The left and right CSS properties specifies the horizontal position of a positioned element.

Note that none of these props have any effect on non-positioned elements.

```
.class {
    top: 1rem;
    left: 100%;
    right: 0;
    bottom: 50px;
}
```

See this demo: http://maxx.link/css-positions

RESPONSIVE/ MULTI-DEVICE LAYOUT PATTERNS

QUICK NOTE:

Understanding responsive design patterns will make you a better web *designer*.

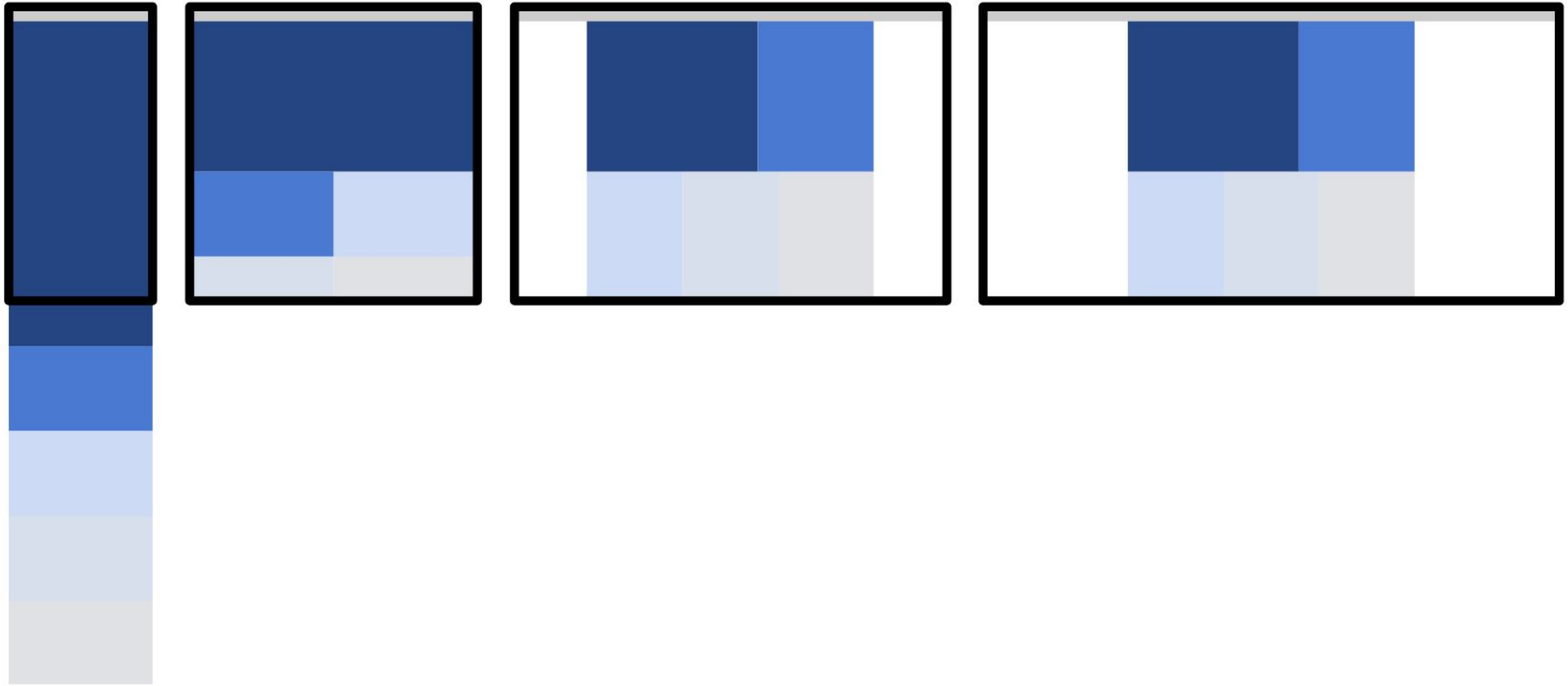
MULTI-DEVICE LAYOUT PATTERNS

Most layouts used by responsive web pages can be categorized into one of five patterns:

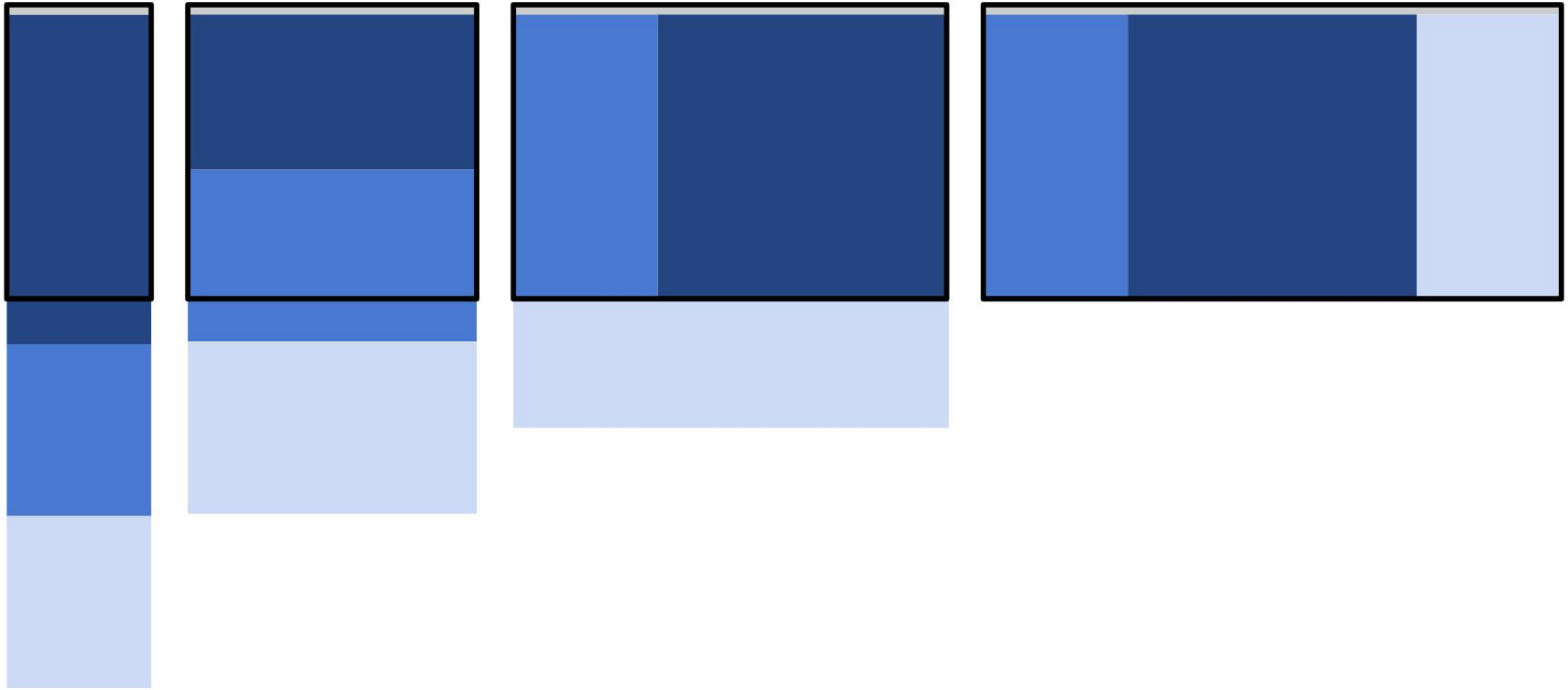
- Mostly fluid
- Column drop
- Layout shifter
- Tiny tweaks
- Off canvas

In some cases, a page may use a combination of patterns, for example column drop and off canvas. These patterns, originally identified by Luke Wroblewski, provide a solid starting point for any responsive page.

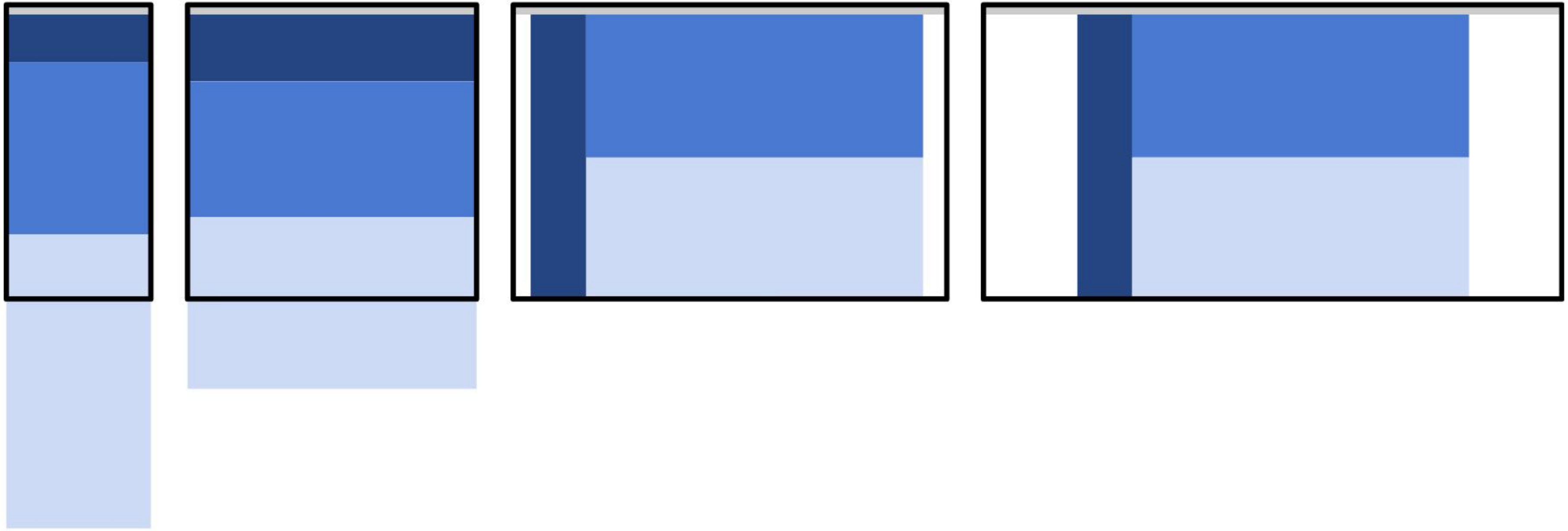
MDLP: MOSTLY FLUID



MDLP: COLUMN DROP

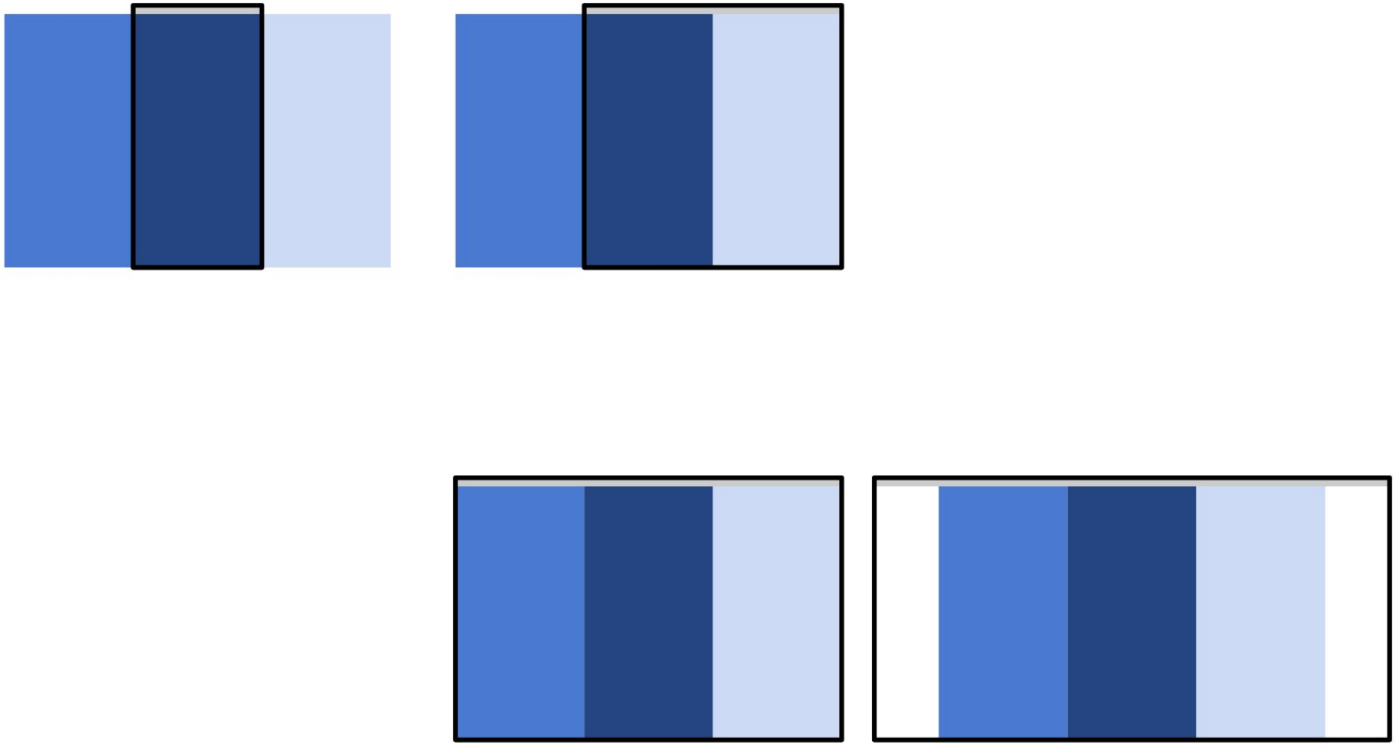


MDLP: LAYOUT SHIFTER



MDLP: TINY TWEAKS

MDLP: OFF CANVAS



GitHub Homework (HW2.0)

- Use the following GitHub repo as a template:
  - https://github.com/maxxcrawford/multi-device-layout-patterns
  - Pick **one** of the three layouts:
    - Mostly Fluid, Column Drop, Layout Shifter
  - Add CSS to the media queries in the stylesheet to change the layout on tablet/desktop, per the responsive guides
  - Tips: Investigate flex-wrap and order properties.
- Due Tuesday, Sept 28, 5:00 PM

MULTI-DEVICE LAYOUT PATTERNS

GITHUB REPO WALKTHROUGH

WEBSITE DISTILLERY PROJECT

XD Link/Desktop + XD Link/Mobile

ASSIGNMENT 04:
HIGH-DEF TO WIREFRAME

- Part 1 in a multi-week project
- Select website to distill down to a wireframe
- Use GitHub Pages to host your project
- Deadlines:
  - Website to distill: Monday, Sept 27, 12:00 PM
  - Wireframes due: Wednesday, Sept 29, 5:00 PM
  - Responsive wireframe website: Due Friday, Oct 1, Beginning of class

GitHub Homework (HW2.0)

- Use the following GitHub repo as a template:
    - https://github.com/maxxcrawford/multi-device-layout-patterns
    - Pick **one** of the three layouts:
        - Mostly Fluid, Column Drop, Layout Shifter
    - Add CSS to the media queries in the stylesheet to change the layout on tablet/desktop, per the responsive guides
    - Tips: Investigate flex-wrap and order properties.
- Due Tuesday, Sept 28, 5:00 PM

© MAXX CRAWFORD 2021. ALL RIGHTS RESERVED.