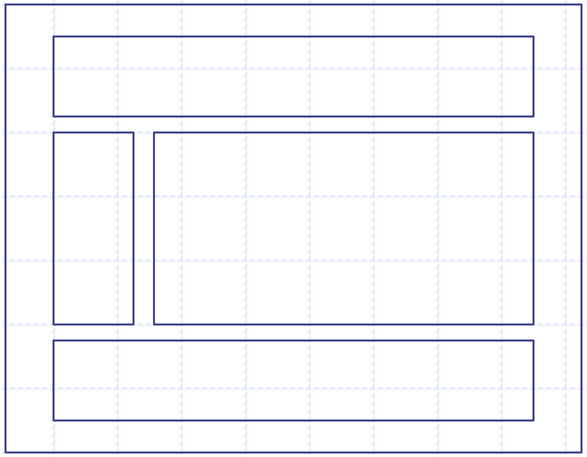
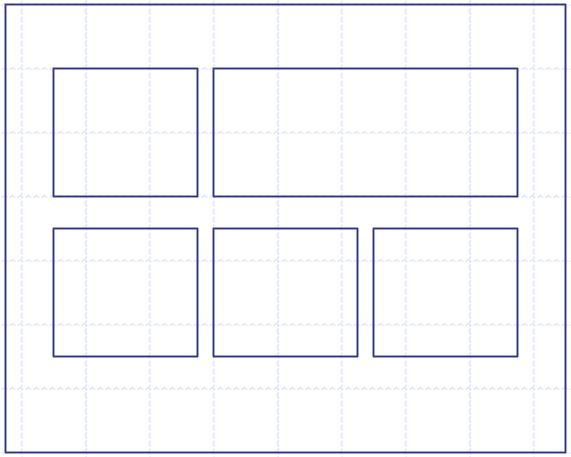
CS3220 Web and Internet Programming

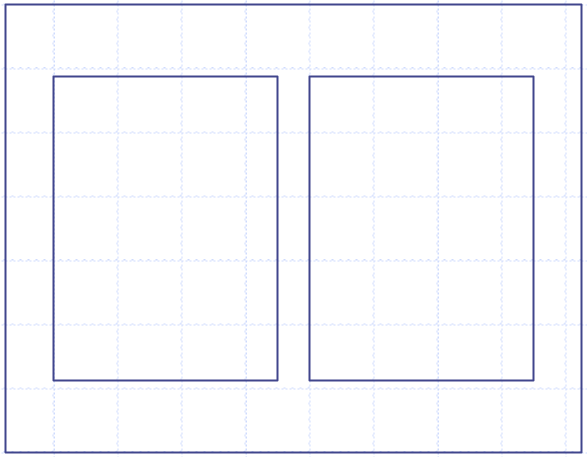
Page Layout with CSS

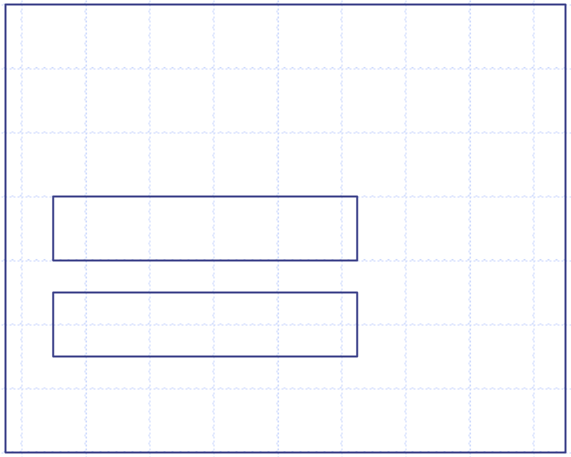
Chengyu Sun
California State University, Los Angeles

Some Page Layouts We May Want





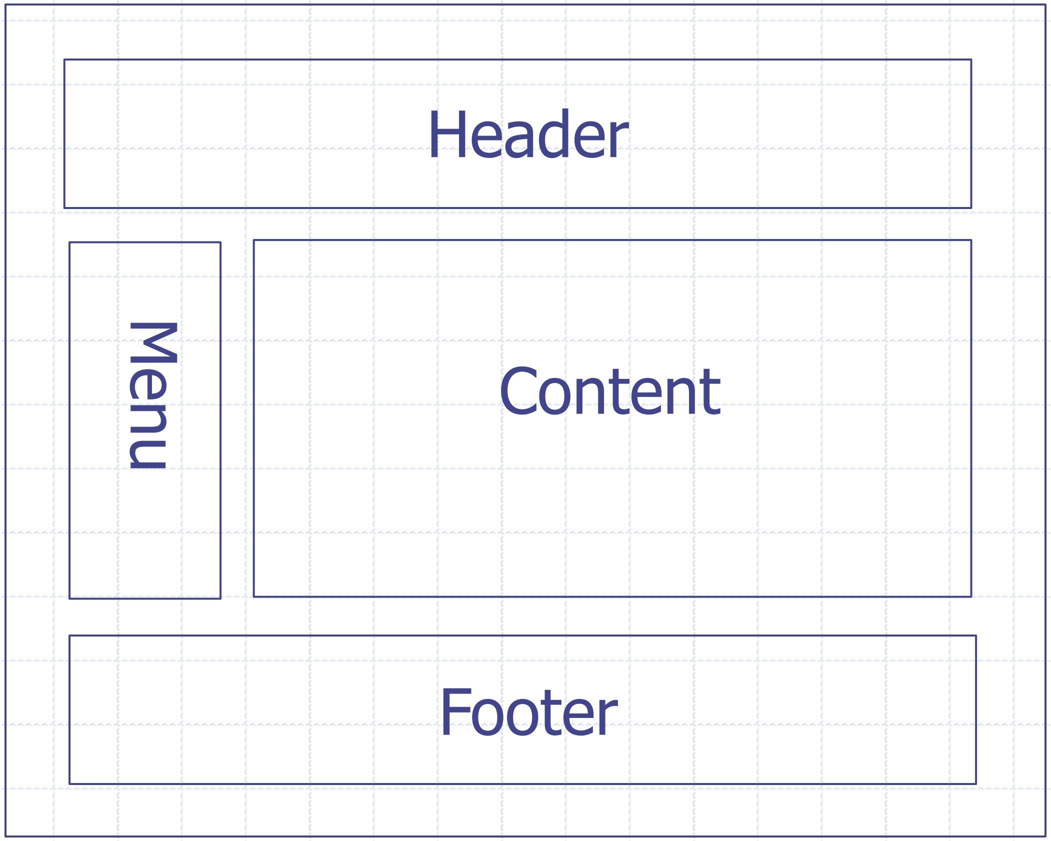




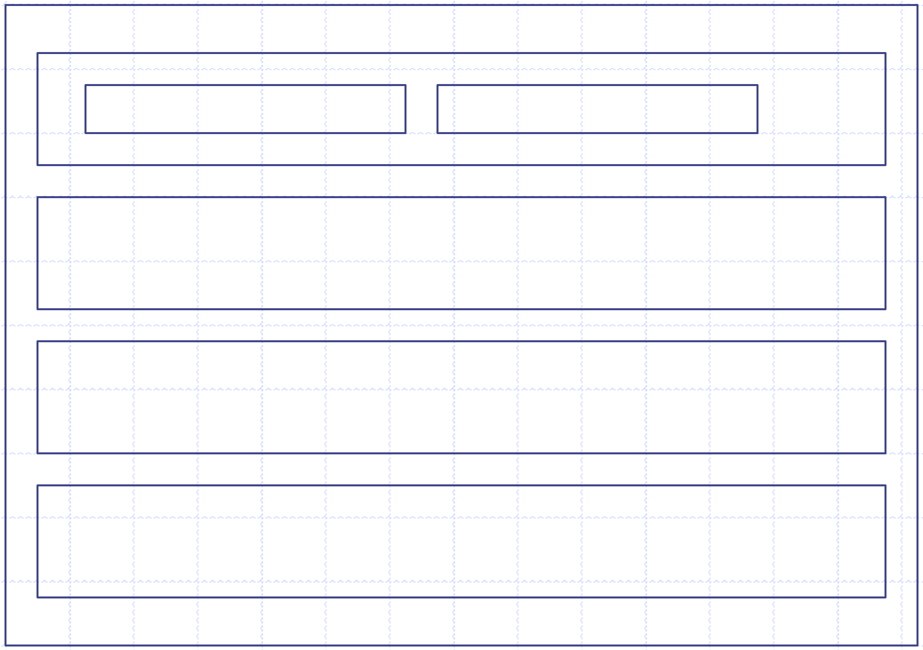
Using CSS Over Table for Layouts?

- Separation of structure and presentation
  - Better readability
  - Easier to change and maintain
- More flexible and powerful
- Accessibility support
- Better performance

Example



Regular HTML Page Layout



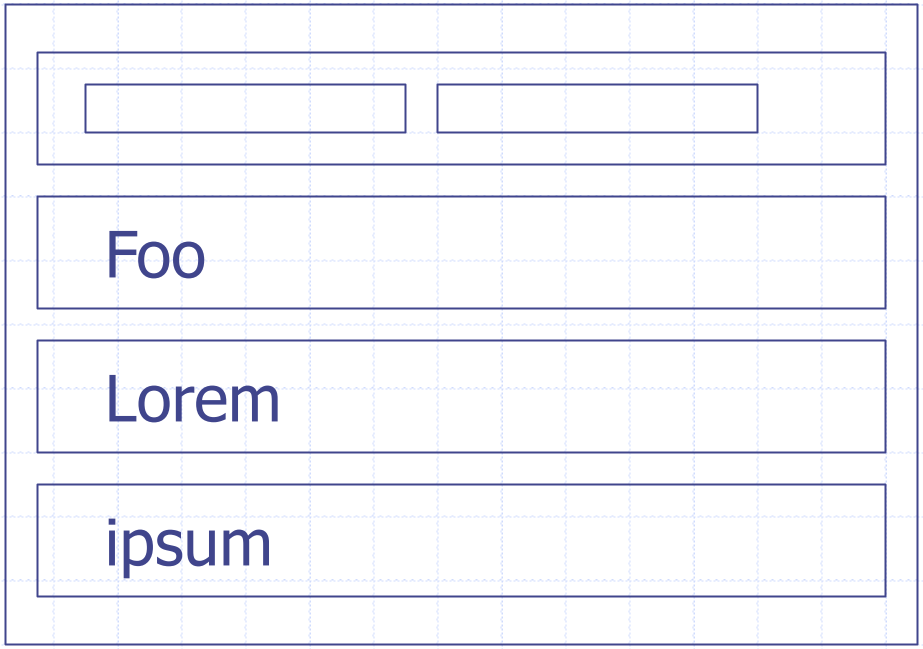
- The elements “flows” from left to right and top to bottom

The `float` Property

- "Float" an element to either the `left` or the `right` inside the element that contains it
- A floated block element no longer take up the full width of the containing element
- <u>A floated element is taken out of the normal flow</u>
  - The elements below will be moved up and overlap with the floated element
  - The content of the element(s) overlapping with the floated element will wrap around it

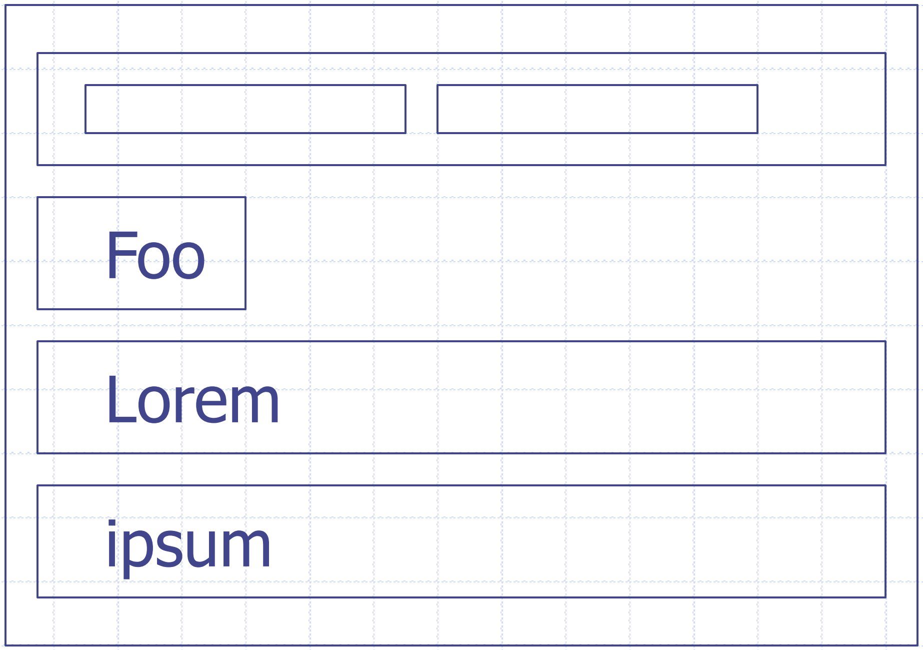
A `float` Example ...

- Before float



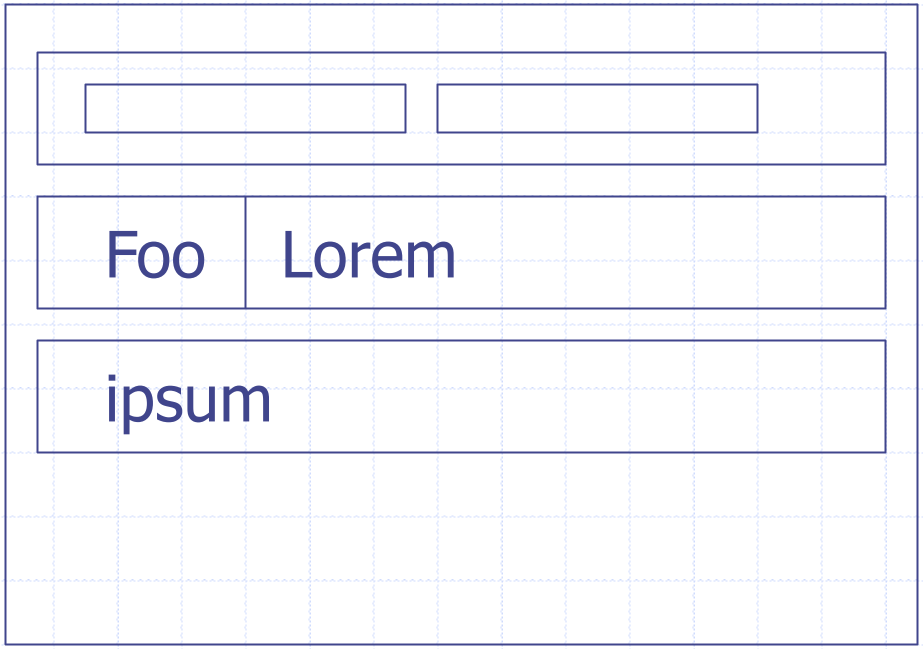
… A `float` Example …

- Float "Foo" block to left – it only takes up the width needed by its content

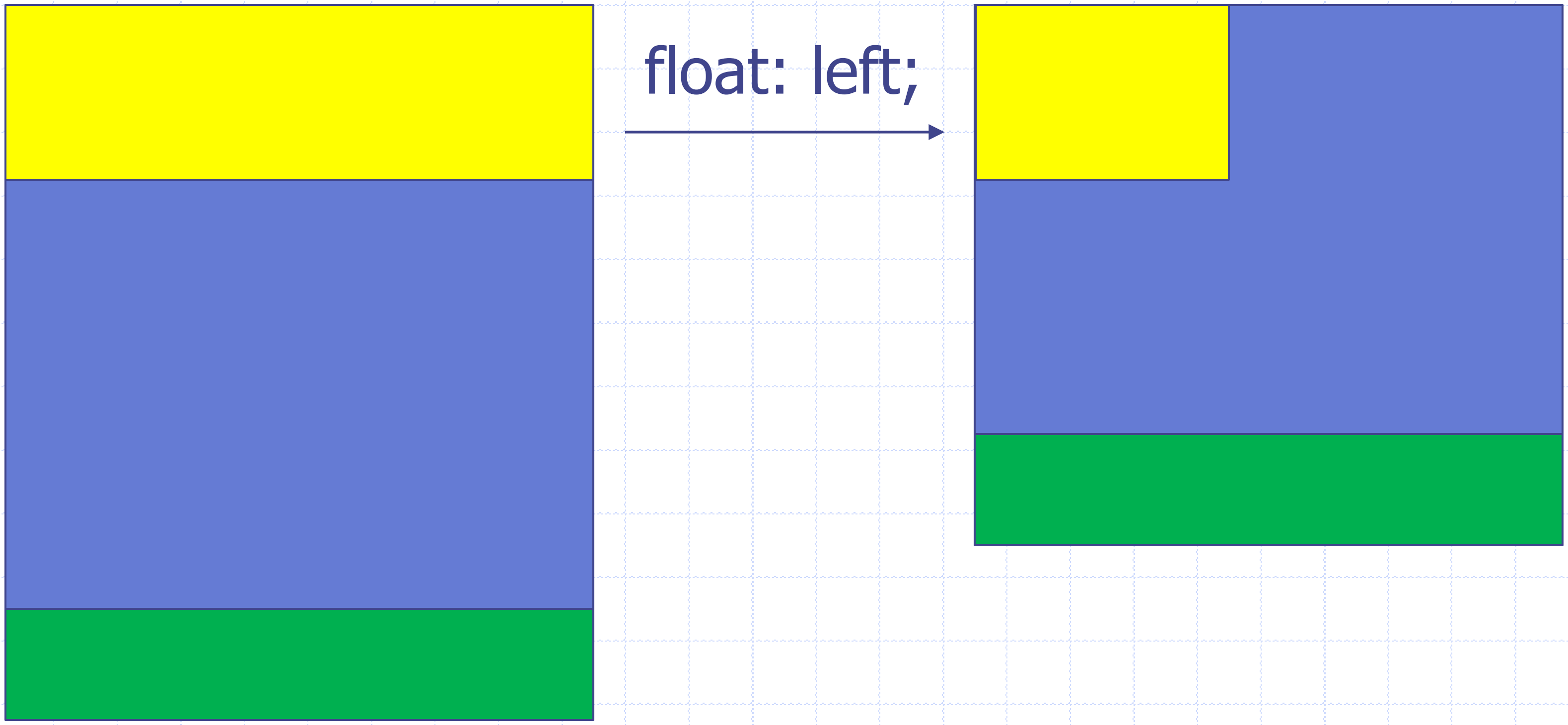


… A `float` Example …

- "Lorem" and "ipsum" blocks are moved up
- "Lorem" block is overlapping with "Foo" block
- The content of "Lorem" block wraps around "Foo" block



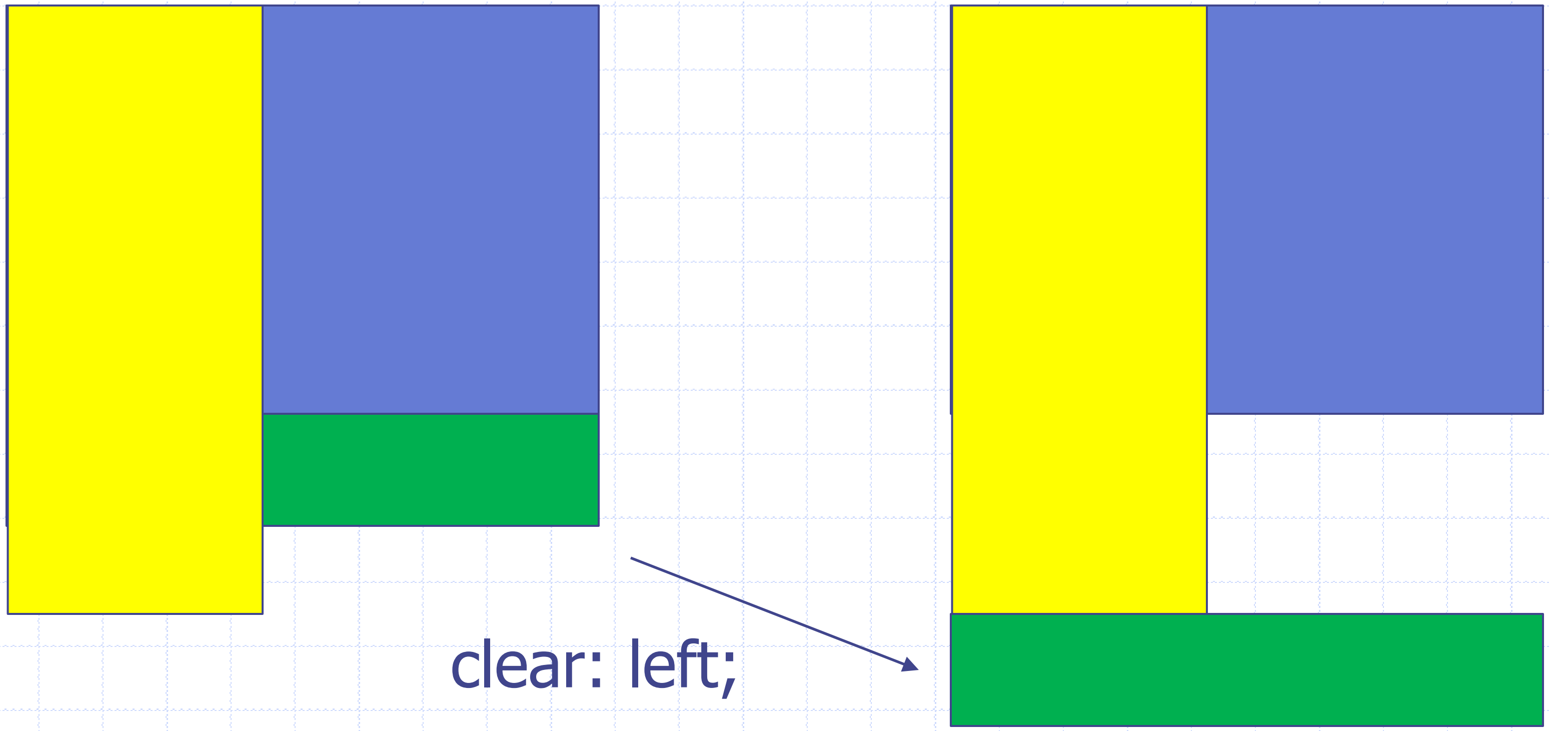
… A `float` Example



The `clear` Property

- Whether an element can be next to floating elements that precede it or must be moved down below them
  - `left`: no floating elements on the left
  - `right`: no floating elements on the right
  - `both`: no floating elements on either side

A `clear` Example



clear: left;

The Positional Properties …

- `position` – how an element is positioned
  - `static`: normal position
  - `relative`: relative to the normal position
  - `absolute`: relative to the containing block
  - `fixed`: relative to the browser window

The element is taken out of the normal flow

… The Positional Properties

- `top`, `bottom`, `left`, `right` – the positions of the top, bottom, left, and right side of the element
- `z-index`
  - An element with higher z-index is displayed in front of elements with lower z-index
  - Only works on positioned elements

CSS Layout Modes

- Block layout
  - float; multiple columns
- Inline layout
- Table layout
- Positioned layout
- Flexible box (i.e. *flexbox*) layout
- Grid layout (experimental)

Multiple Columns Example

- Designed for displaying text – *not for creating multi-column layouts*
- Properties
  - `column-count`
  - `column-width` (suggested optimal width)
  - `column-gap`
  - `column-rule` (can be set like borders)

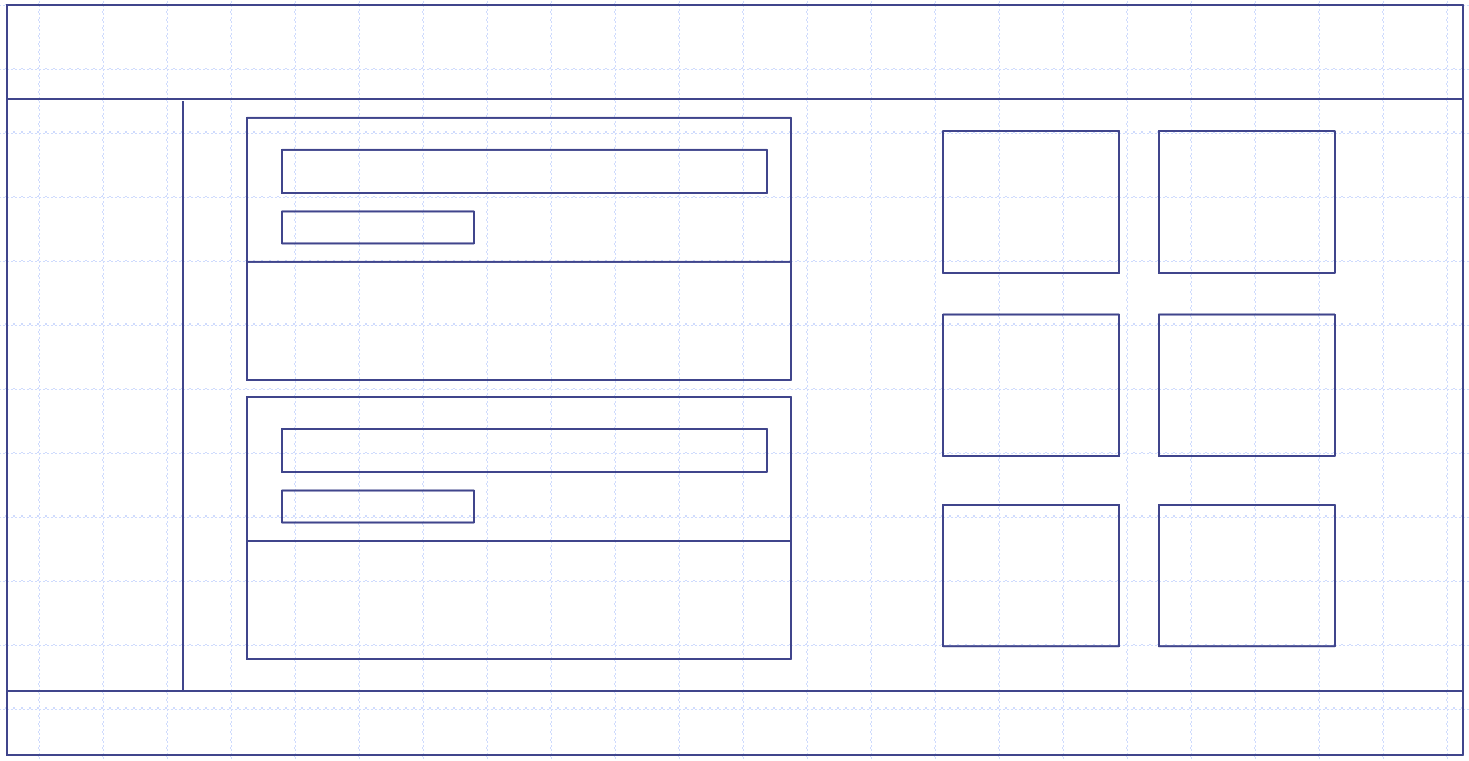
Flexbox Example

- `display: flex;`
- `flex-direction`
- `flex-wrap`
- `align-items`
- `align-content` and `justify-content`

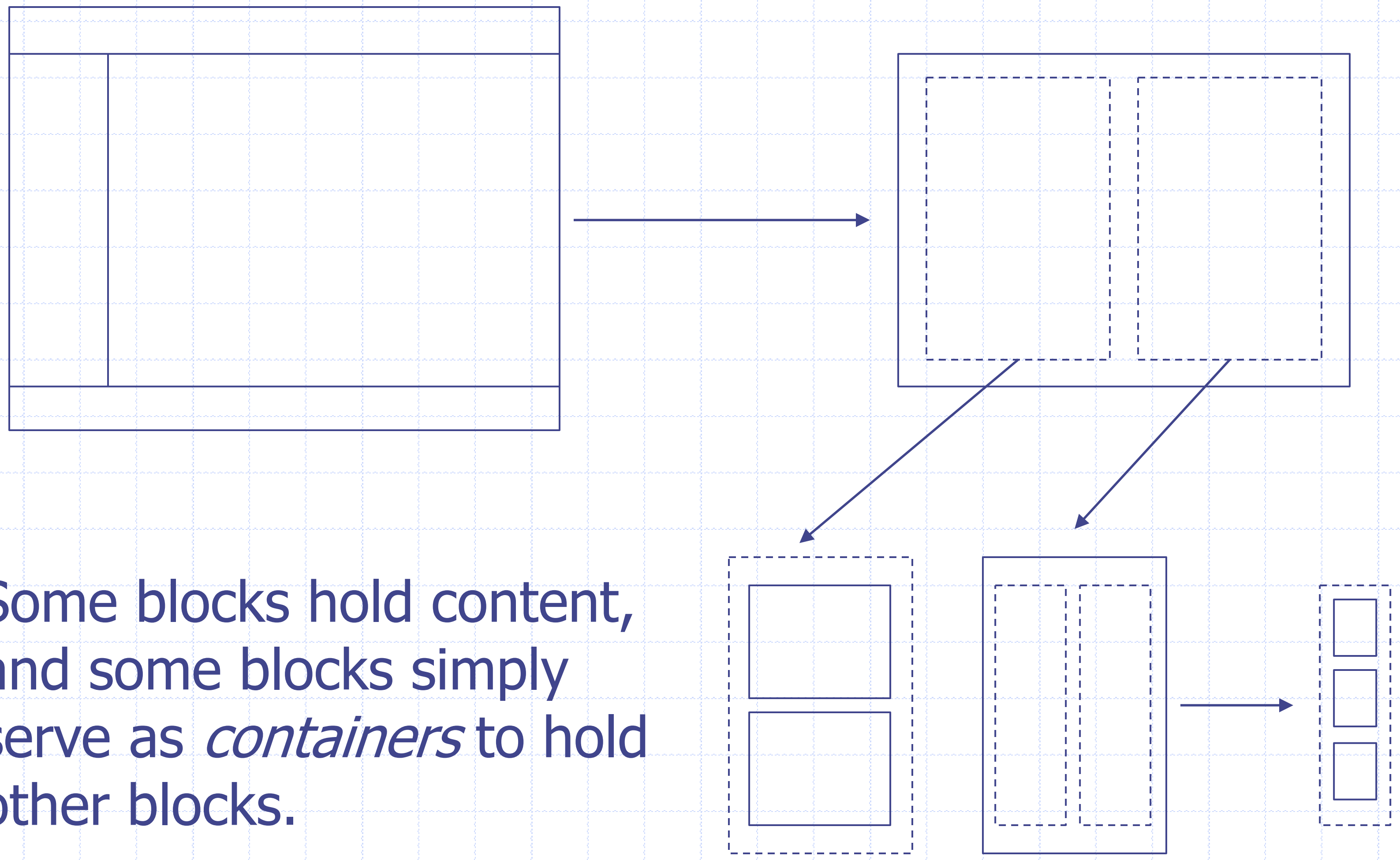
For more on flexbox, please see A Complete Guide to Flexbox.

Tip About Creating Complex Layouts

- Start with the top-level blocks, then create "sub-layout" in each block



Complex Layout Example



Some blocks hold content, and some blocks simply serve as *containers* to hold other blocks.