

OIM3690 - Web Technologies



CSS Layout

CSS Layout Approaches

- Normal Flow
- Floats
- Positioning
- **Flexbox**
- **Grids**

Normal Flow

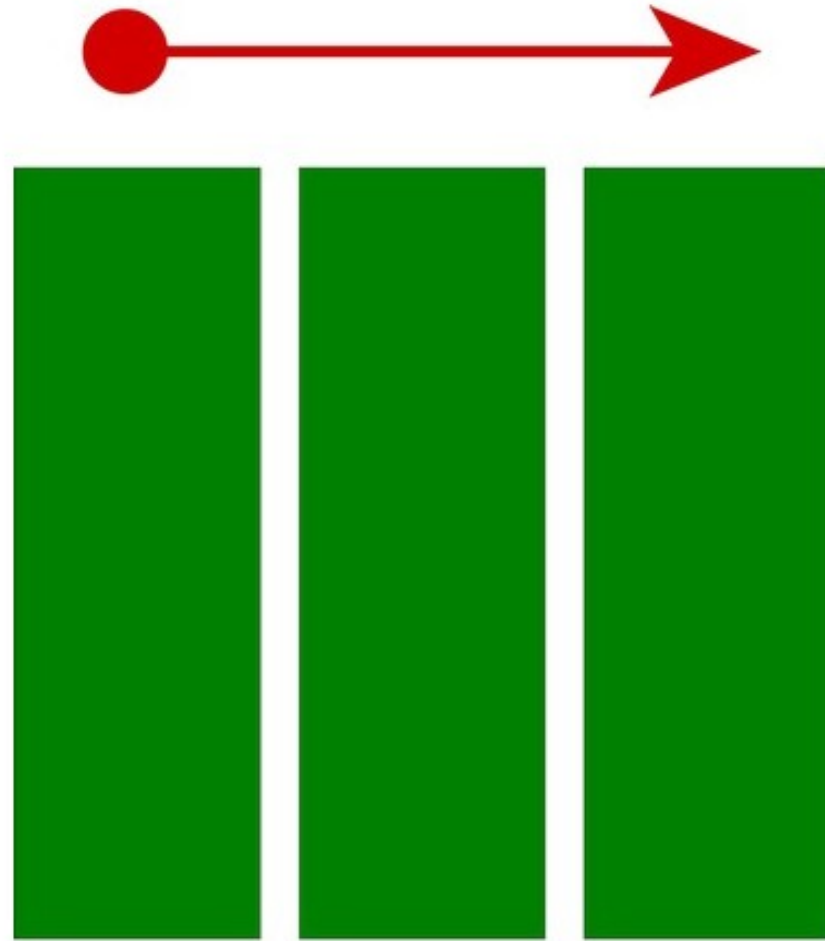
- block direction: mostly vertical
- `display`
- `float`, `clear`, `overflow`
- `position`
 - `static` (default), `relative`, `absolute`, `fixed`, `sticky`

Flexbox

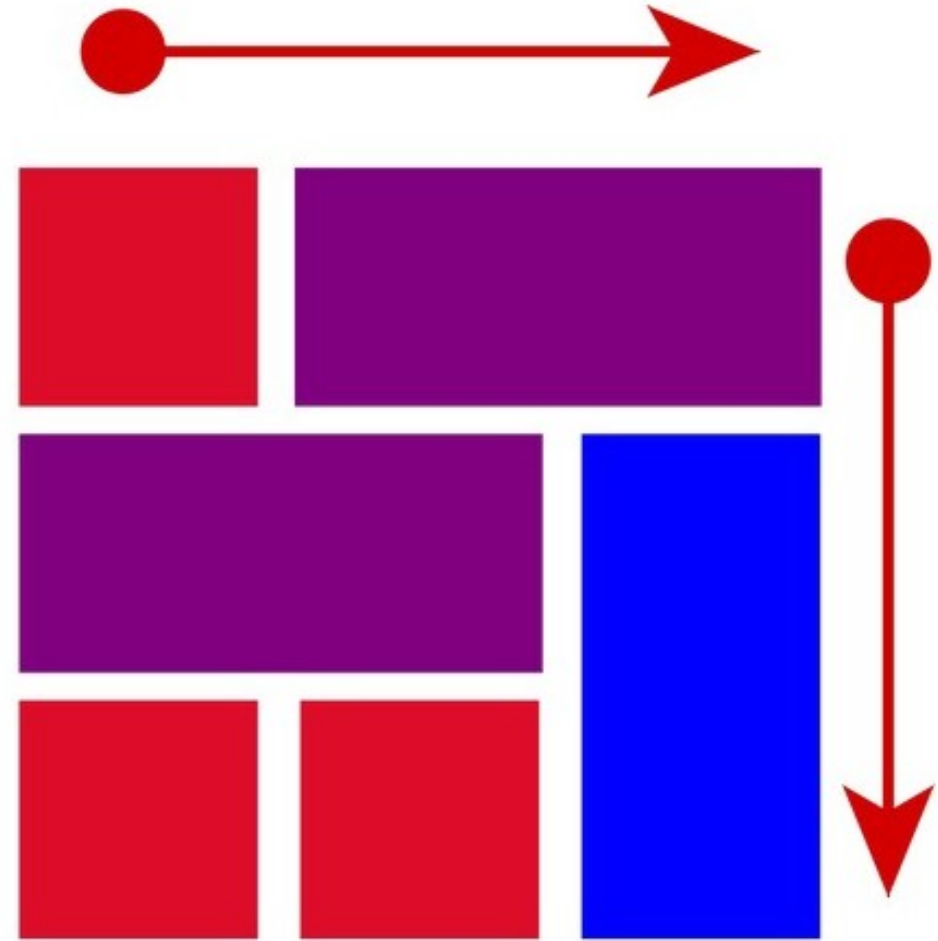
- [MDN](#)
- [W3Schools](#)
- Games: [Flexbox Froggy](#), [Flexbox Defense](#)

(some useful) Flexbox Properties

- **Properties** for the Parent (flex container):
 - `display: flex;`
 - `flex-direction`, `flex-wrap`, `flex-flow`
 - `justify-content`
 - `align-items`
 - `align-content`
- **Properties** for the Children (flex items inside flex container):
 - `order`
 - `flex-grow`, `flex-shrink`, `flex-basis`, `flex`
 - `align-self`



Flexbox
One Dimensions



CSS Grids
Two Dimensions

Grids

- [MDN](#)
- [W3Schools](#)
- Game: [Grid Garden](#)

(some useful) Grids-Related Properties

- **Properties** for the Parent (Grid Container):
 - `display: grid;`
 - `grid-template-columns`, `grid-template-rows`, `grid-template-areas`
 - `column-gap`, `row-gap`, `gap`
 - `justify-items`, `align-items`, `justify-content`, `align-content`
 - `grid-auto-columns`, `grid-auto-rows` ...
- **Properties** for the Children (Grid Items inside Grid Container):
 - `grid-column-start`, `grid-column-end`, `grid-row-start`, `grid-row-end`
 - `grid-area`: (with `grid-template-areas`) or shorthand for above
 - `justify-self`, `align-self`, ...
- **Special units:** `fr`

Exercise: *ex10.html*

Create the elephant "table" in *ex08-2.html* without using `table` element.

References

- MDN: [CSS Layout](#)
- [A Complete Guide to Flexbox](#)
- [A Complete Guide to Grid](#)

Questions?

