

# OIM3690 - Web Technologies



# CSS Layout

# CSS Layout Approaches

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

# Normal Flow

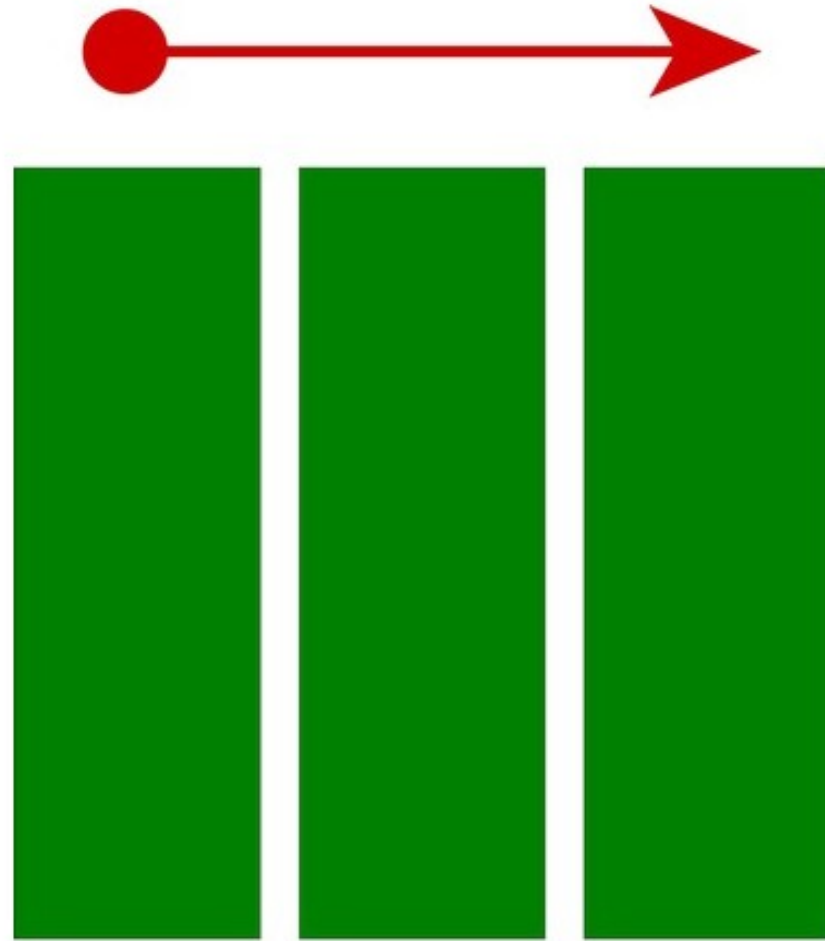
- block direction: mostly vertical
- `display`
- `float`, `clear`, `overflow`
- `position`
  - `static` (default), `relative`, `absolute`, `fixed`, `sticky`

# Flexbox

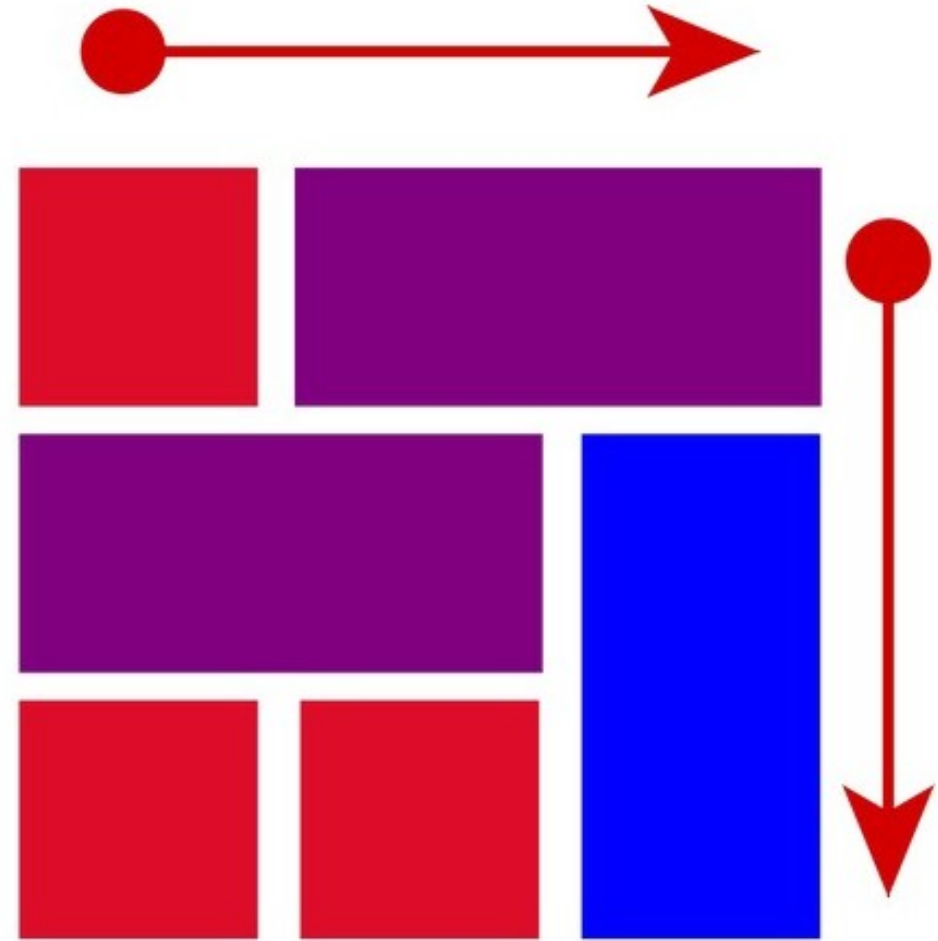
- [MDN](#)
- [W3Schools](#)
- Games: [Flexbox Froggy](#), [Flexbox Defense](#)

# (some useful) Flexbox-Related Properties

- for flex container:
  - `flex-direction`, `flex-wrap`, `flex-flow`
  - `justify-content` : horizontal
  - `align-items`
  - `align-content` : vertical
- for each flex item (inside 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

- for grid container:
  - `grid-template-columns`, `grid-template-rows`, `grid-template-areas`, `gap`, `justify-items`, `align-items`, `justify-content`, `align-content`, `grid-auto-columns`, `grid-auto-rows` ...
- for each grid:
  - `grid-column-start`, `grid-column-end`, `grid-row-start`, `grid-row-end`, `grid-area`, `justify-self`, `align-self`,

## Exercise: *ex11.html*

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

# References

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

# Questions?

