



shadows



# CSS Text Columns

The CSS3 multi-column layout allows easy definition of multiple columns of text - just like in newspapers.

To create multiple columns all we need is to set **column-count** property to a numerical value:

**column-count: 3;**

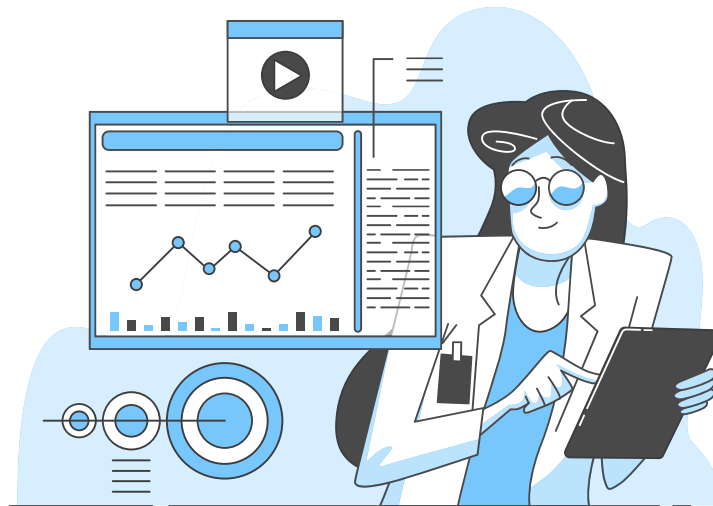


A decorative network diagram consisting of several blue circular nodes connected by thin black lines. The nodes are arranged in a non-linear fashion, with some having concentric circles. Ellipses (...) are used to indicate that the network continues beyond the visible nodes.

# Flexbox

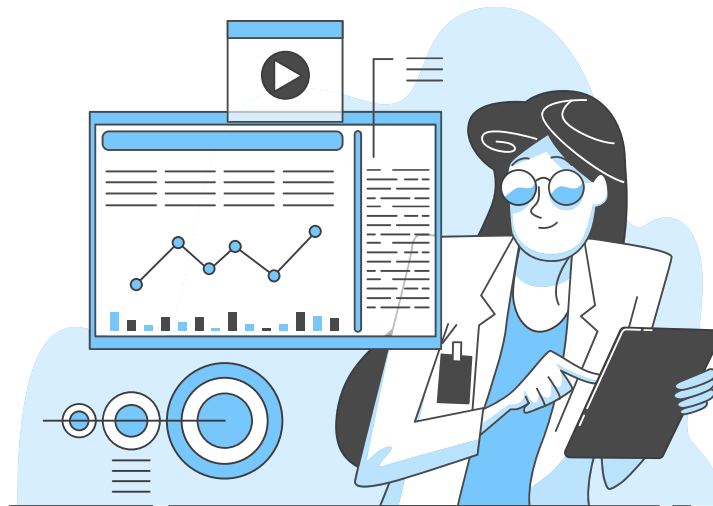
# Flexbox

- **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.
- **Flexbox** layout allows you to easily format HTML. Flexbox makes it simple to align items vertically and horizontally using rows and columns. Items will "flex" to different sizes to fill the space. It makes responsive design easier.



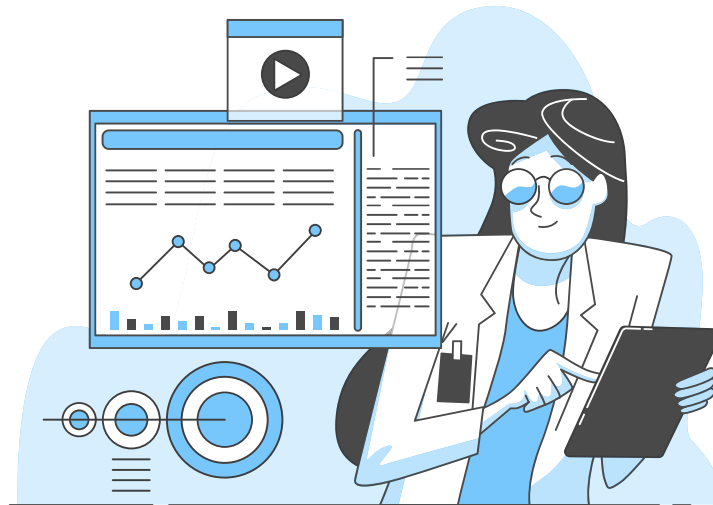
# Flexbox

- It's a new layout mode in CSS3.
- Flexbox consists of flex containers and flex items.
- A flex container is declared by setting the **display** property of an element to **flex** or **inline-flex**.
- Inside a flex container there is one or more flex items.
- Everything outside a flex container and inside a flex item is rendered as usual. Flexbox defines how flex items are laid out inside a flex container.

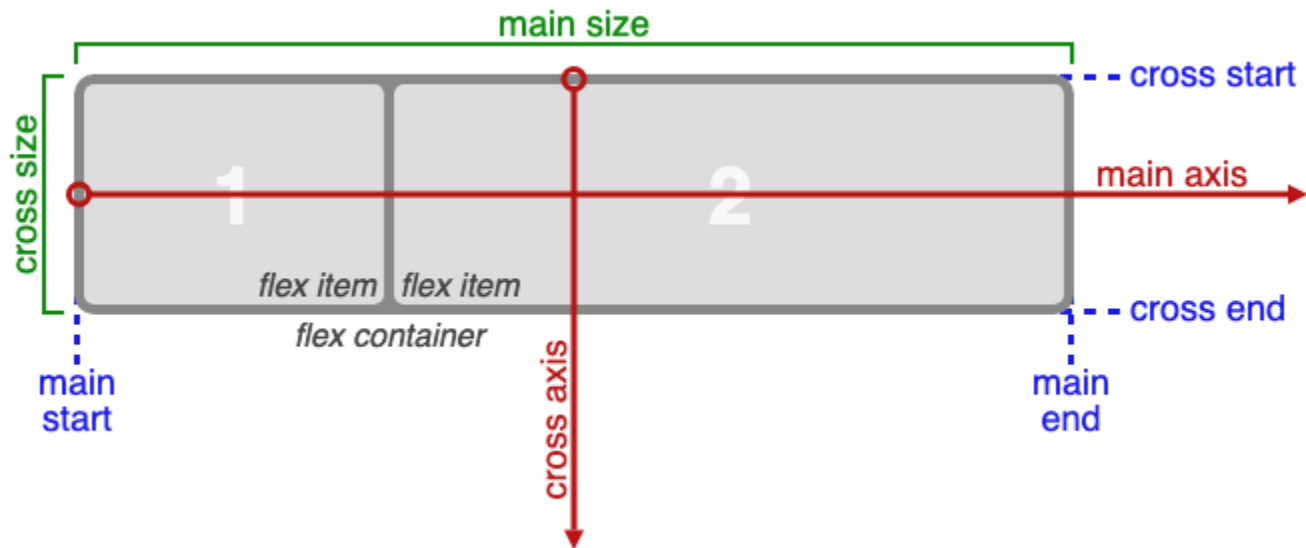


# Why Flexbox?

- Vertically centering a block of content inside its parent.
- Making all the children of a container take up an equal amount of the available width/height, regardless of how much width/height is available.
- Making all columns in a multiple-column layout adopt the same height even if they contain a different amount of content.



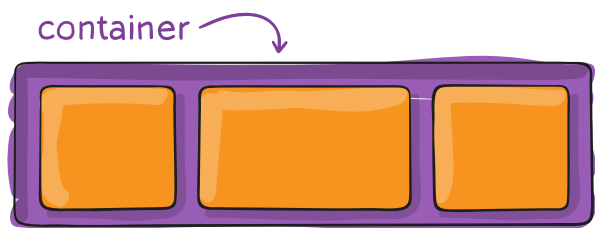
# Axes



# Flexbox Properties

A decorative network diagram consisting of several blue circular nodes connected by thin grey lines. The nodes are arranged in a non-linear fashion, with some having concentric circles. Ellipses (...) are used to indicate that the network continues beyond the visible nodes.





## For Container

**flex-direction**

**Flex-wrap**

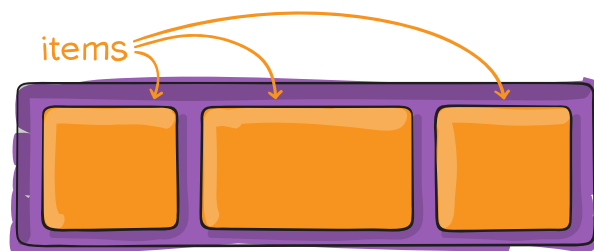
**Flex-flow**

**Justify-content**

**Align-items**

**Align-content**

**gap**



## For Items

Order

Flex-grow

Flex-shrink

Flex-basis

Align-self

Flex

**display: block;**

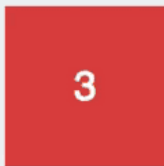
1

2

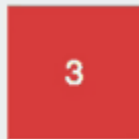
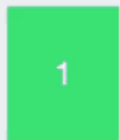
3

4

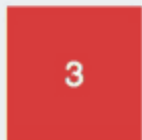
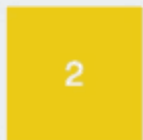
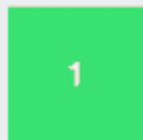
**flex-direction: row;**



**flex-direction: column;**



**justify-content: flex-start;**



**align-items: flex-start;**

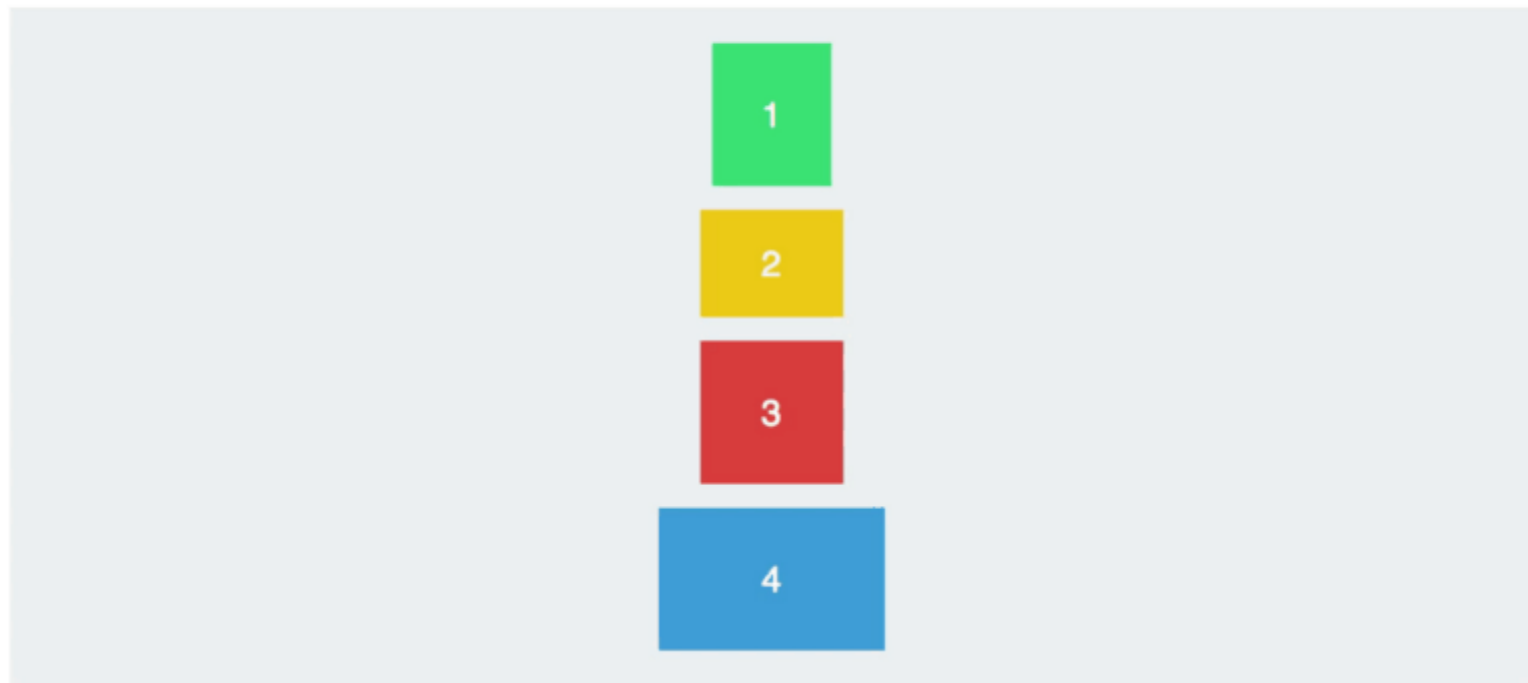
1

2

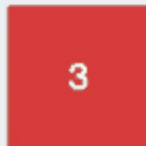
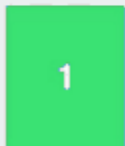
3

4

**flex-direction: column;**  
**justify-content: center; align-items: center;**



**align-self: flex-start;**





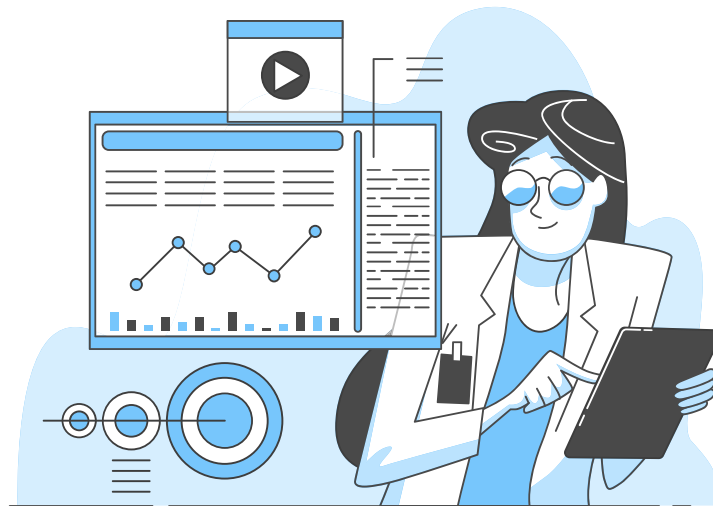
# Flexbox

If you want to learn flexbox while playing  
you can check:

<http://flexboxfroggy.com/>

You can learn more about flexbox here:

<https://css-tricks.com/snippets/css/a-guide-to-flexbox/>



# Play and Learn

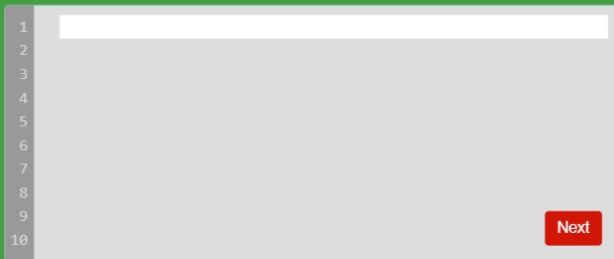
## FLEXBOX FROGGY

◀ Level 1 of 1 ▶

Welcome to Flexbox Froggy, a game where you help Froggy and friends by writing CSS code! Guide this frog to the lilypad on the right by using the `justify-content` property, which aligns items horizontally and accepts the following values:

- `flex-start`: Items align to the left side of the container.
- `flex-end`: Items align to the right side of the container.
- `center`: Items align at the center of the container.
- `space-between`: Items display with equal spacing between them.
- `space-around`: Items display with equal spacing around them.

For example, `justify-content: flex-end;` will move the frog to the right.



Flexbox Froggy is created by [Codepen](#) • [GitHub](#) • [Twitter](#) • [Settings](#)

Want to learn CSS grid? Play [Grid Garden](#)