

STUDENT NAME	R.ROJA	
STUDENT REGISTRATION NUMBER	251U1R2064	CLASS: CSE AIML-C
PROGRAM	UG	YEAR and TERM: 1st year & 1st term
SUBJECT NAME	Fundamentals Of Web Development	
NAME OF THE ASSESSMENT	Free Writing-II	
DATE OF SUBMISSION	19.10.2025	

About flex:

The CSS Flexbox (Flexible Box Layout) module is a one-dimensional layout method for arranging items within a container. It provides an efficient and powerful way to distribute space and align items, making it far easier to create responsive web designs compared to older techniques like floats and positioning.

To use Flexbox, you must first create a flex container by applying `display: flex` or `display: inline-flex` to a parent element. All direct children of this container automatically become flex items.

About flex shorthand:

The `flex` property is a shorthand for three individual Flexbox properties: `flex-grow`, `flex-shrink`, and `flex-basis`. It provides a concise and powerful way to control how a flex item will grow, shrink, and be sized within its container. Using the shorthand is highly recommended to avoid common mistakes and write cleaner, more efficient CSS.

Flex-basis:

The `flex-basis` property in CSS Flexbox specifies the initial size of a flex item before any available space is distributed. It essentially sets the item's size along the main axis, but unlike `width` or `height`, it is responsive to the `flex-direction` property.

Example 1: flex-basis with pixels

This is the most direct way to set a specific initial size for an item.

HTML:

html

```
<div class="container">
  <div class="item item-1">Item 1</div>
  <div class="item item-2">Item 2</div>
  <div class="item item-3">Item 3</div>
</div>
```

About flexgrow:

The **flex-grow** property specifies how much a flex item should grow relative to the other items inside the same flex container. It is a unitless, positive number that dictates how any available "free space" in the container is distributed among the flex items that have this property set.

The default value of **flex-grow** is 0, which means items will not grow to fill any extra space.

Example 1: Equal growth

If all items have the same **flex-grow** value, they will grow equally and occupy the same amount of space.

HTML:

```
html  
<div class="flex-container">  
  <div class="item">Item 1</div>  
  <div class="item">Item 2</div>  
  <div class="item">Item 3</div>  
</div>
```

About flex shrink :

The **flex-shrink** property in CSS Flexbox controls how much a flex item will shrink relative to other flex items when there is not enough space in the flex container to fit all the items at their defined **flex-basis** size.

The default value for **flex-shrink** is 1, meaning that items with a **flex-shrink** of 1 will shrink equally if needed. Setting a higher number means the item will shrink more aggressively than its siblings.

Examples

Example 1: Preventing an item from shrinking

In this example, three items are placed in a container that is too small to hold them all at their initial size of 200px. Item 2 has **flex-shrink: 0**, so it will maintain its 200px width, while the other two items shrink.

HTML:

```
html  
<div class="flex-container">  
  <div class="item">Item 1</div>  
  <div class="item item-2">Item 2 (Won't shrink)</div>  
  <div class="item">Item 3</div>  
</div>
```

How to create equal width columns using flexbox:

To create equal-width columns using Flexbox, you need to apply two key properties: `display: flex` to the container and `flex: 1` to each child item. This causes the child items to grow and fill the available space evenly.

- `flex-grow: 1`: Allows the item to grow and take up an equal share of any extra space.
- `flex-shrink: 1`: Allows the item to shrink equally if the container is too small
- Example HTML and CSS
- HTML:

```
html
```
- `<div class="container">`
- `<div class="column">Column 1</div>`
- `<div class="column">Column 2</div>`
- `<div class="column">Column 3</div>`
- `</div>`