

Session: 8

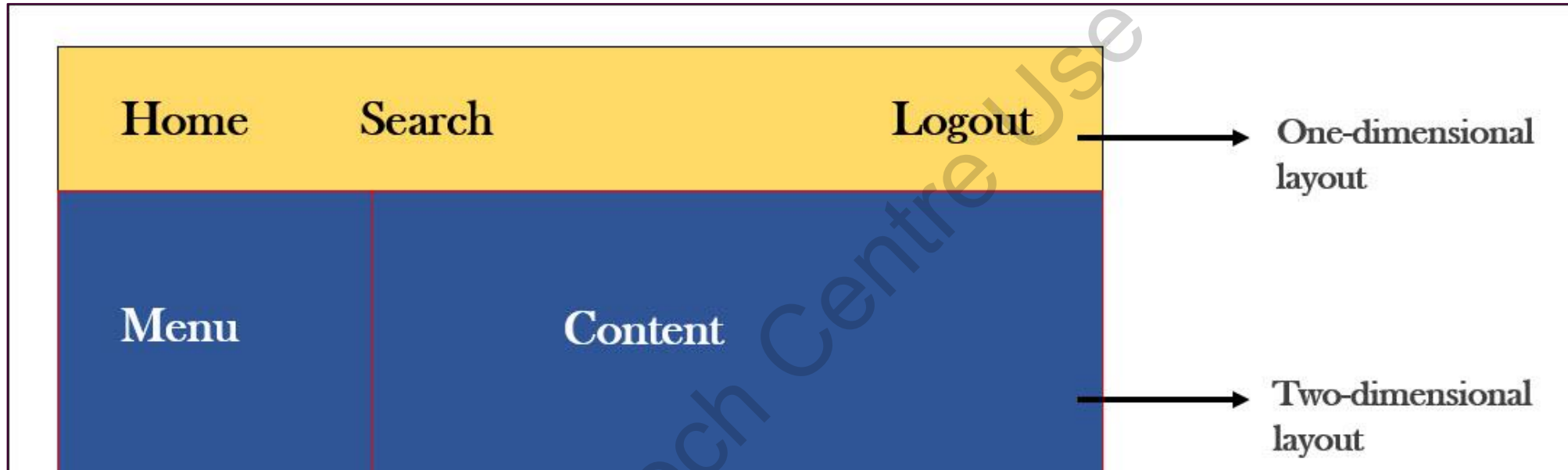
Understanding Layouts in CSS3

For Aptech Centre Use Only

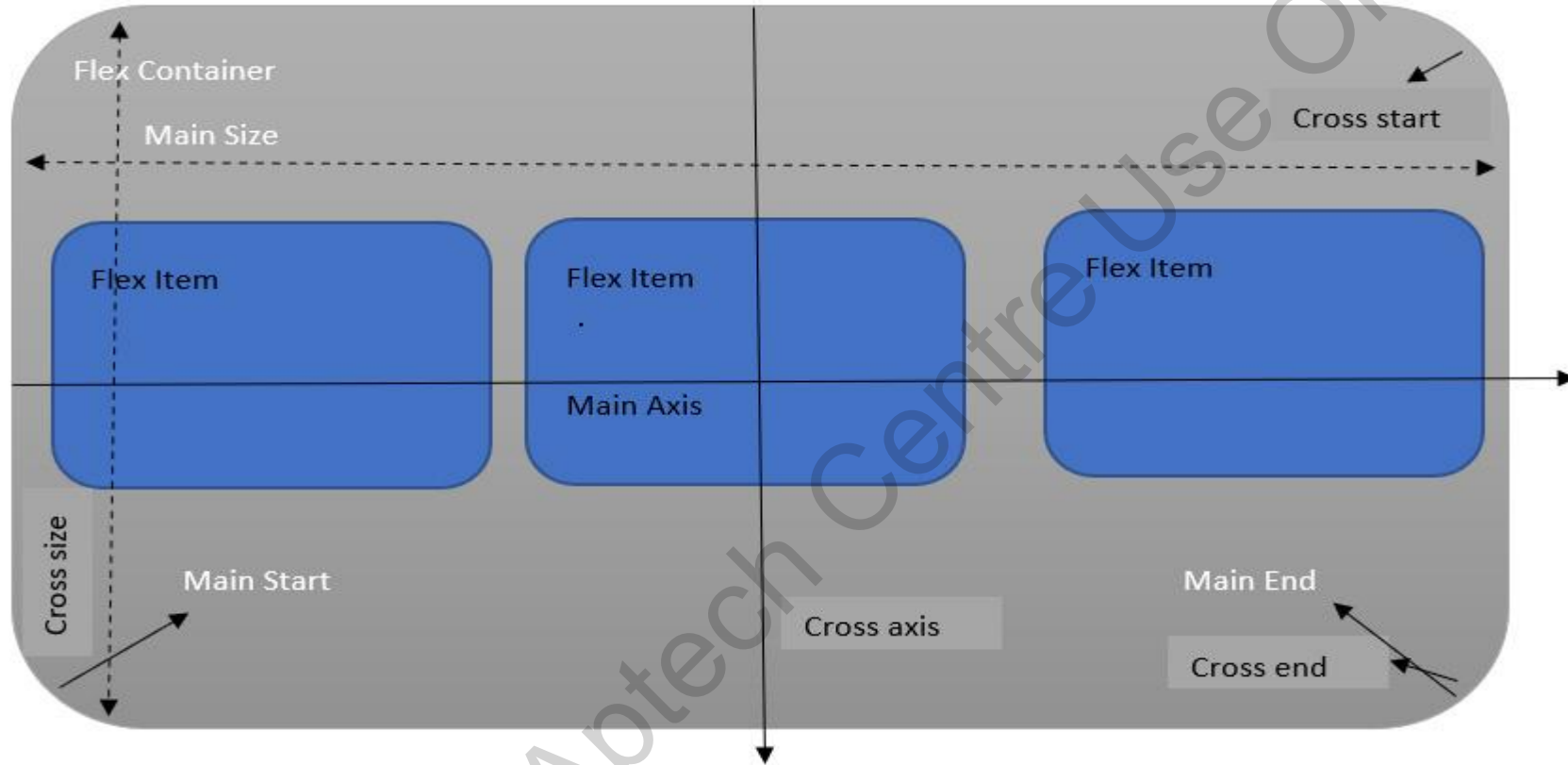
Objectives

- Describe flexbox and grid layouts
- Describe different flexbox properties
- Explain different grid properties

Introduction



CSS Flexbox Layout Module



Properties of Flexbox Container 1-3

`display`

`flex-
direction`

`flex-wrap`

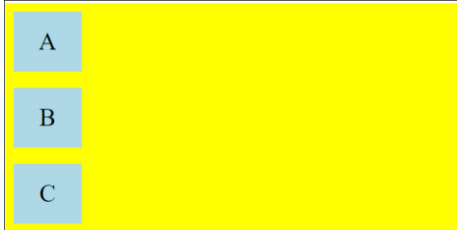
`flex-flow`

`justify-
content`

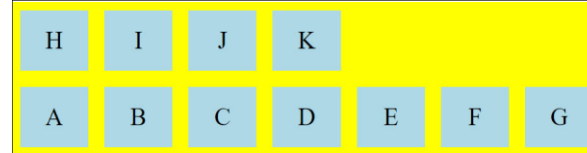
`align-
items`

`align-
content`

Properties of Flexbox Container 2-3



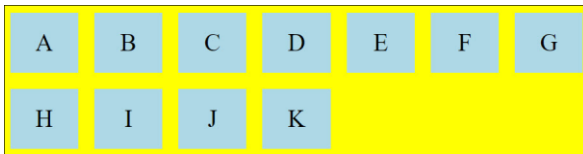
Example of Flex Direction - column



flex-wrap Property: wrap-reverse



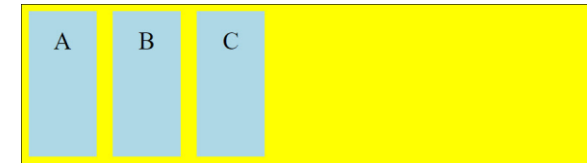
space-around Attribute Of justify-content Property



flex-flow Property: row wrap

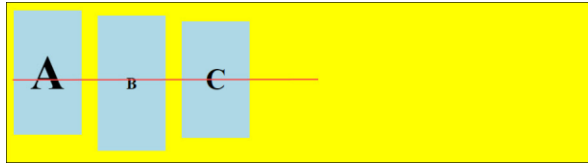


flex-end Attribute Of justify-content Property

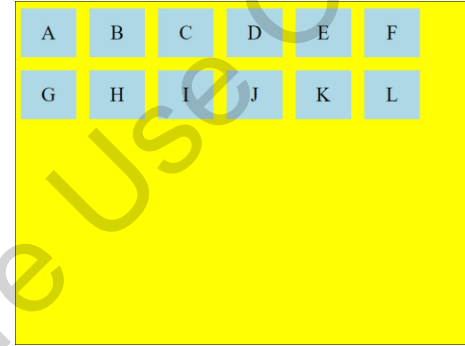


stretch Attribute of align-items Property

Properties of Flexbox Container 3-3



**baseline Attribute of
align-items Property**



**flex-start Attribute of
align-content Property**

Properties of Flexbox Item 1-4

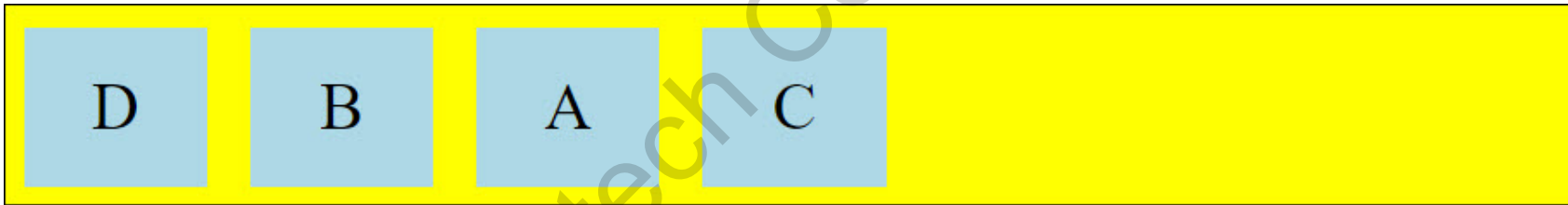
`order`

`flex-grow`

`flex-shrink`

Properties of Flexbox Item 2-4

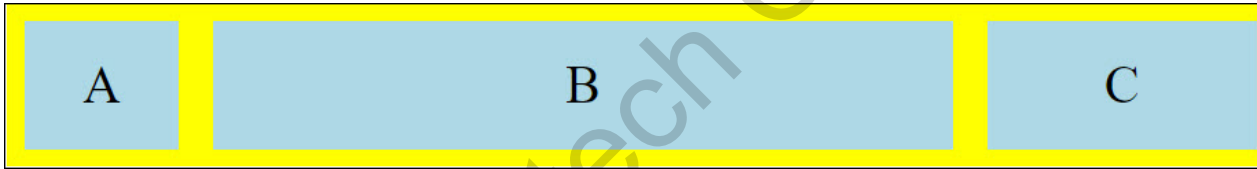
```
<div class="flex-container">  
  <div style="order: 3">A</div>  
  <div style="order: 2">B</div>  
  <div style="order: 4">C</div>  
  <div style="order: 1">D</div>  
</div>
```



order Property

Properties of Flexbox Item 3-4

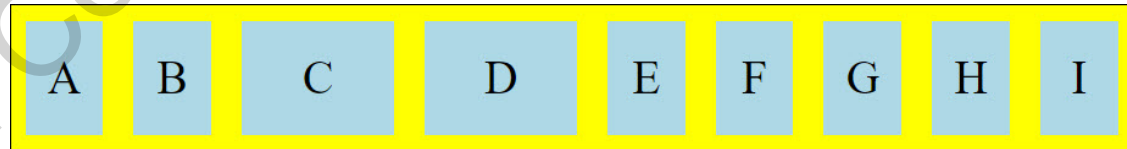
```
<div class="flex-container">  
  <div style="flex-grow:  
  1">A</div>  
  <div style="flex-grow:  
  6">B</div>  
  <div style="flex-grow:  
  2">C</div>  
</div>
```



flex-grow Property

Properties of Flexbox Item 4-4

```
<div class="flex-container">  
  <div>A</div>  
  <div>B</div>  
  <div style="flex-shrink: 0">  
    C</div>  
  <div style="flex-shrink: 0">  
    D</div>  
  <div>E</div>  
  <div>F</div>  
  <div>G</div>  
  <div>H</div>  
<div>I</div></div>
```



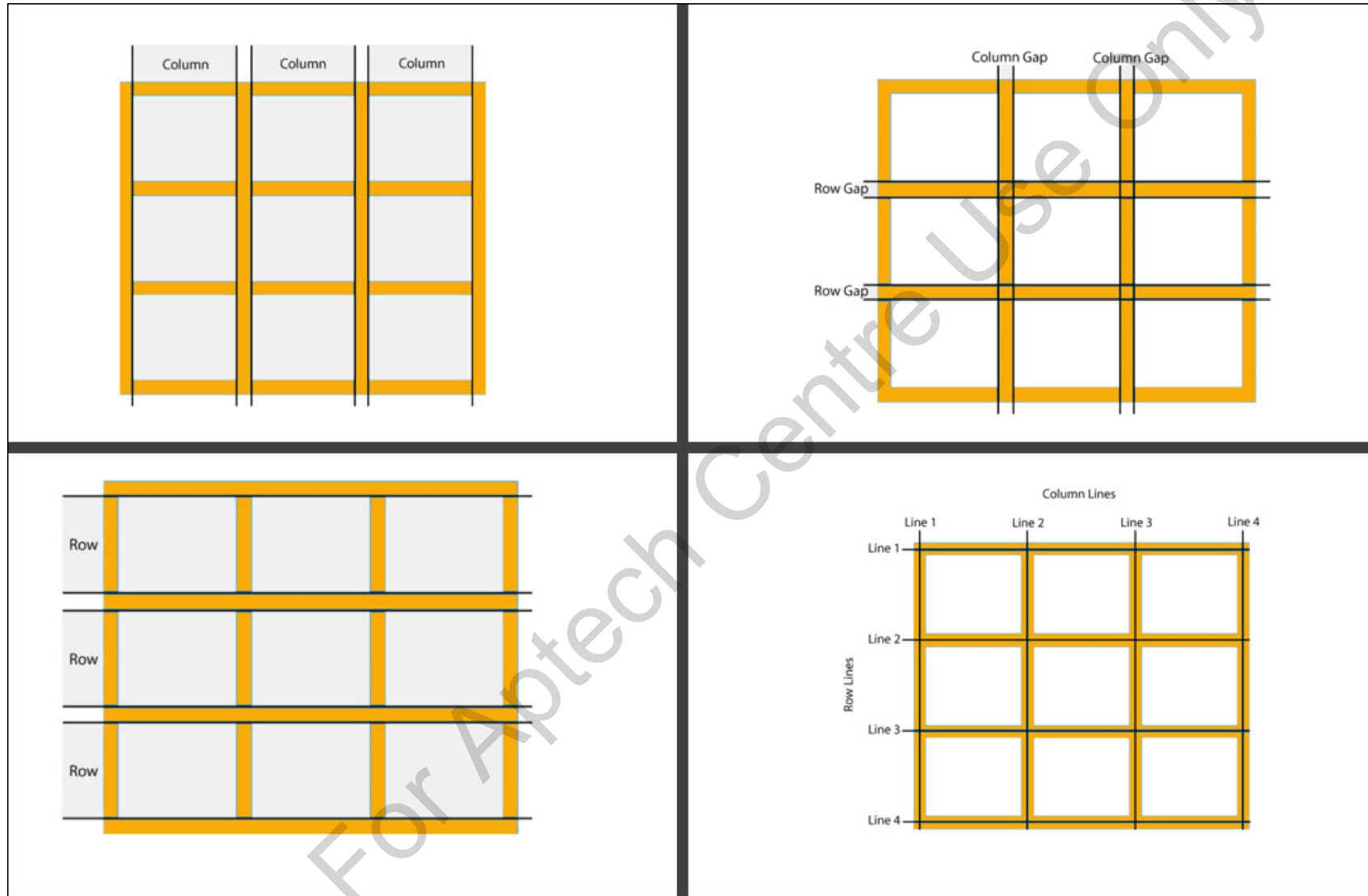
flex-shrink Property

CSS Grid Layout Module 1-2

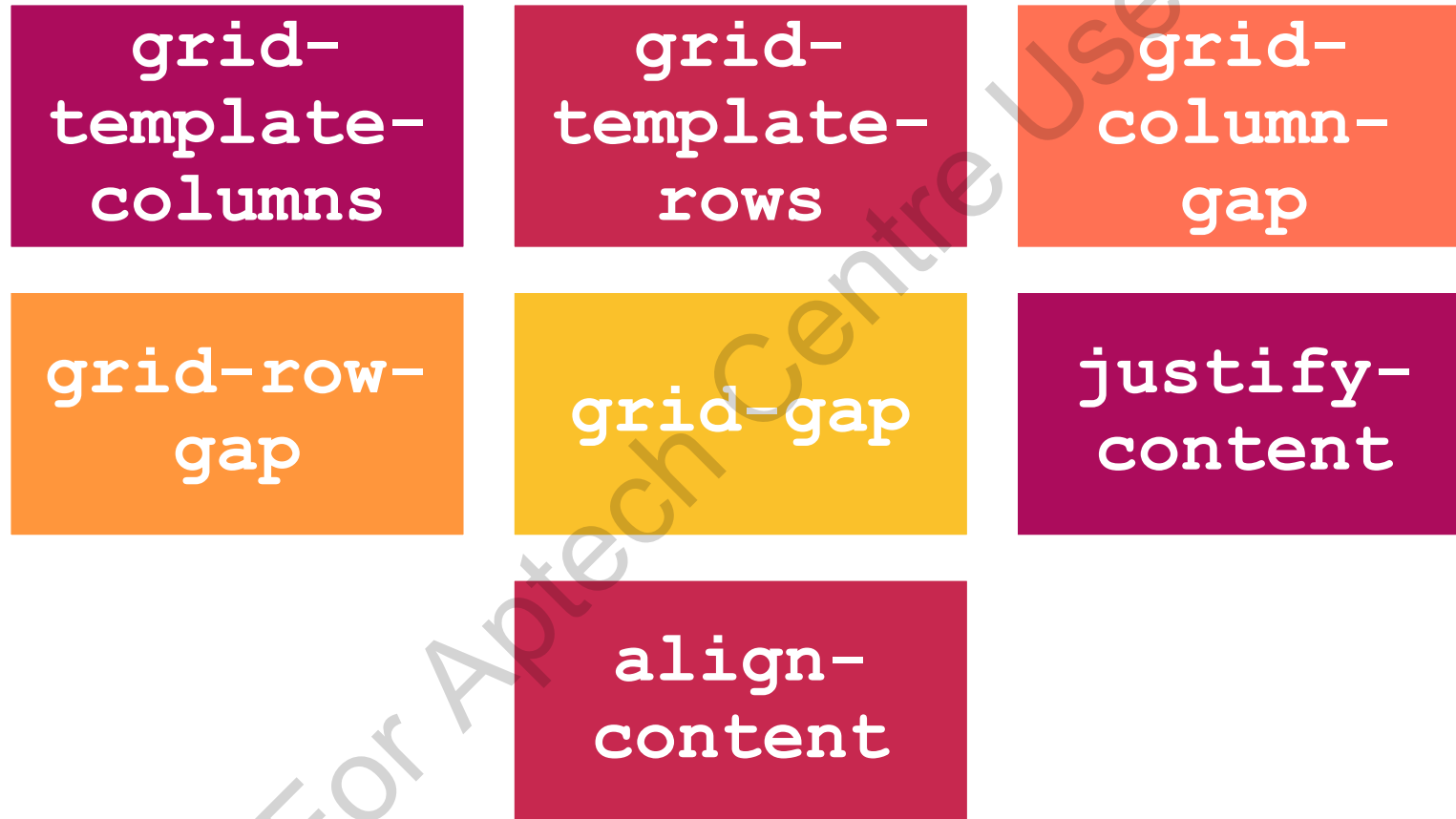
A	B	C
D	E	F

Grid Layout

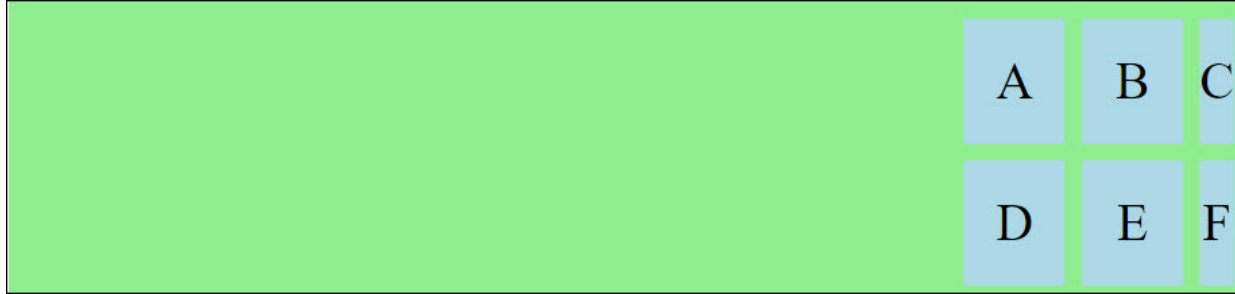
CSS Grid Layout Module 2-2



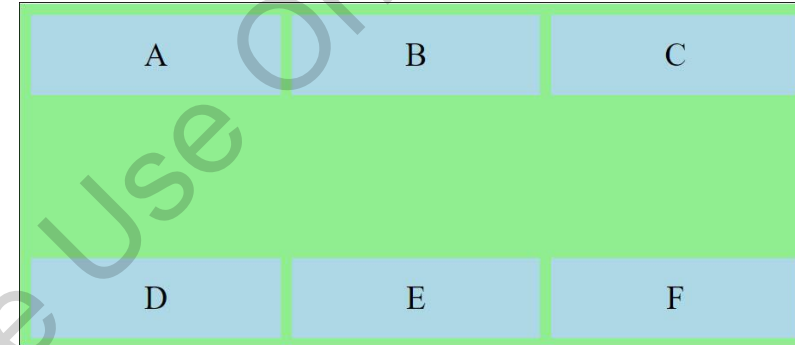
Grid Container Properties 1-2



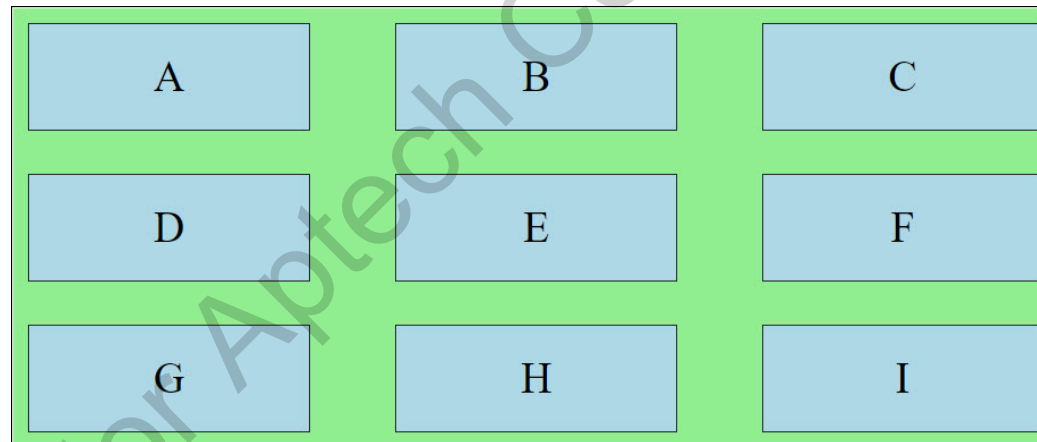
Grid Container Properties 2-2



`justify-content`, `grid-template-columns`,
and `grid-gap` **Properties**

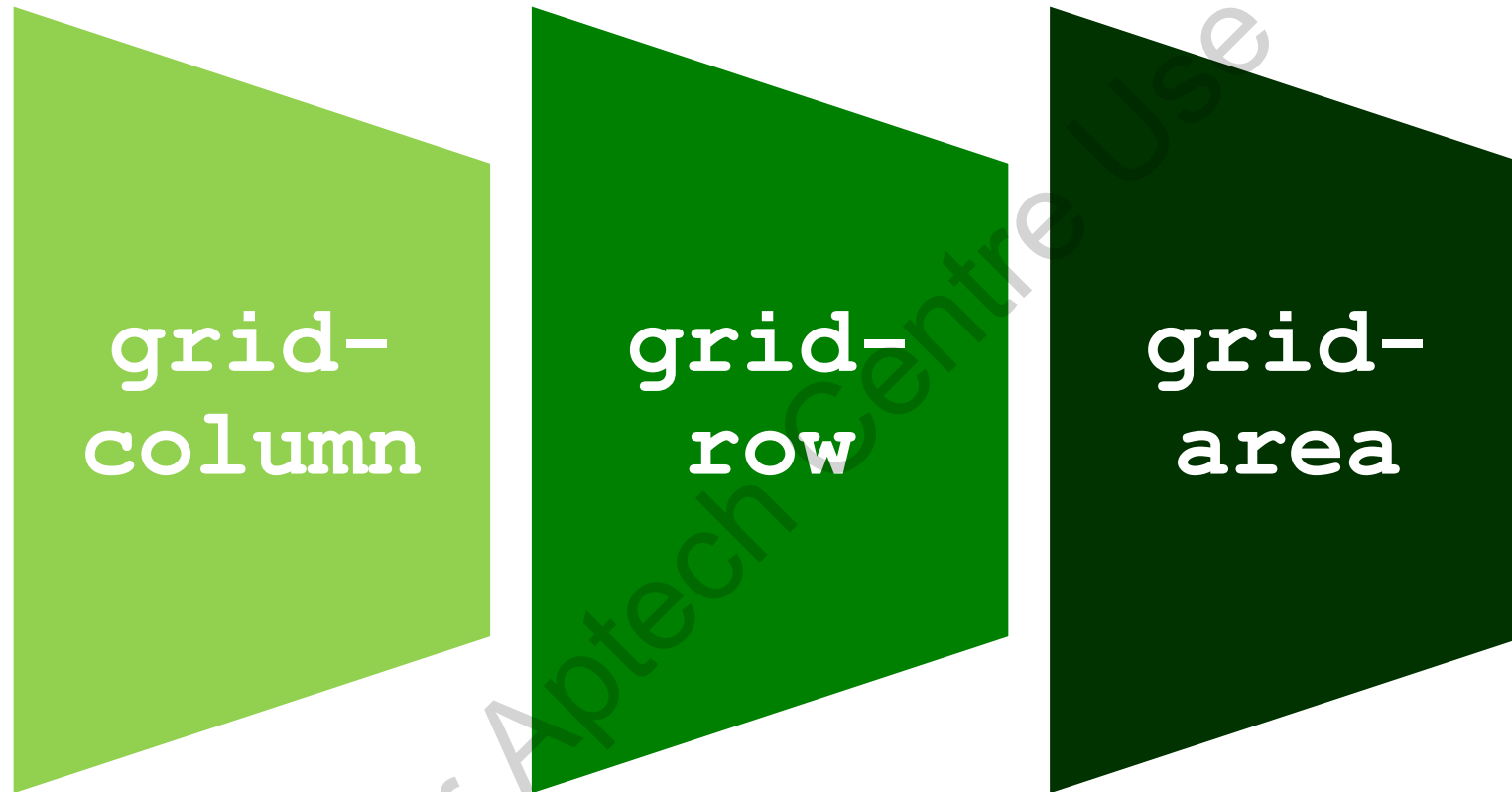


`align-content` **Property**



`grid-gap` **Property**

Grid Items 1-4



Grid Items 2-4

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
display: grid;
grid-template-columns: auto auto auto auto auto auto;
grid-gap: 10px;
background-color: lightgreen;
padding: 10px;
}
.grid-container > div {
background-color: lightblue;
text-align: center;
padding: 20px 0;
font-size: 30px;
}
.item2 {
grid-column: 2 / span 5;
}
</style>
</head>
```

```
<body>
<div class="grid-container">
<div class="item1">A</div>
<div class="item2">B</div>
<div class="item3">C</div>
<div class="item4">D</div>
<div class="item5">E</div>
<div class="item6">F</div>
<div class="item7">G</div>
<div class="item8">H</div>
<div class="item9">I</div>
<div class="item10">J</div>
<div class="item11">K</div>
<div class="item12">L</div>
</div>
</body>
</html>
```

A	B				
C	D	E	F	G	H
I	J	K	L		

grid-column Property

Grid Items 3-4

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
display: grid;
grid-template-columns: auto auto auto auto auto auto;
grid-gap: 10px;
background-color: lightgreen;
padding: 10px;
}
.grid-container > div {
background-color: lightblue;
text-align: center;
padding: 20px 0;
font-size: 30px;
}
.item1 {
grid-row: 1 / span 2;
}
</style>
</head>
```

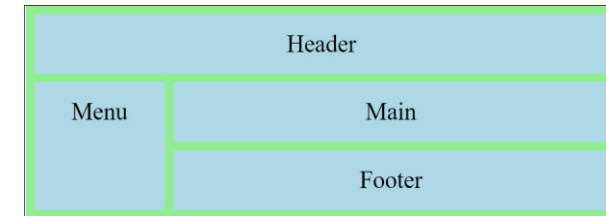
```
<body>
  <div class="grid-container">
    <div class="item1">A</div>
    <div class="item2">B</div>
    <div class="item3">C</div>
    <div class="item4">D</div>
    <div class="item5">E</div>
    <div class="item6">F</div>
    <div class="item7">G</div>
    <div class="item8">H</div>
    <div class="item9">I</div>
    <div class="item10">J</div>
    <div class="item11">K</div>
    <div class="item12">L</div>
  </div>
</body>
</html>
```

A	B	C	D	E	F
G	H	I	J	K	
L					

grid-row Property

Grid Items 4-4

```
<!DOCTYPE html>
<html>
<head>
<style>
.item1 { grid-area: header; }
.item2 { grid-area: menu; }
.item3 { grid-area: main; }
.item4 { grid-area: footer; }
.grid-container { display:
  grid;
  grid-template-areas:
    'header header header header header header' 'menu
    main main main main main'
    'menu footer footer footer footer footer';
  grid-gap: 10px;
  background-color: lightgreen;
  padding: 10px;
}
```



grid-area Property

Summary

- ❖ Flexbox provides one-dimensional layout.
- ❖ Grid provides two-dimensional layout.
- ❖ A container is the parent element and child elements of the container are flex items.
- ❖ A grid layout has one parent element with one or more child elements.
- ❖ Items in a grid container are called grid items.
- ❖ All grid items are placed inside columns and rows.
- ❖ Layout of grid, flexbox, flex items, and grid items can be controlled using different properties.
- ❖ A data attribute helps store custom data or extra information on an HTML element.