Celebrating 13 Years

lexbox

← Float based layout CSS Grids→

vinash Malhotra

05-June-2023

sed layouts

ox is **CSS Layout Design** build using **display:flex** property. **Flexbox** is used to build one-dimensional layout in css. One l means flexbox can build layout in one dimension (either row or column) at one time. For two-dimensional layouts, use hat can handle both row and column.

or inline-flex is used to build flexbox. Flex can build one dimension layout which is better than float based layout.

In this page

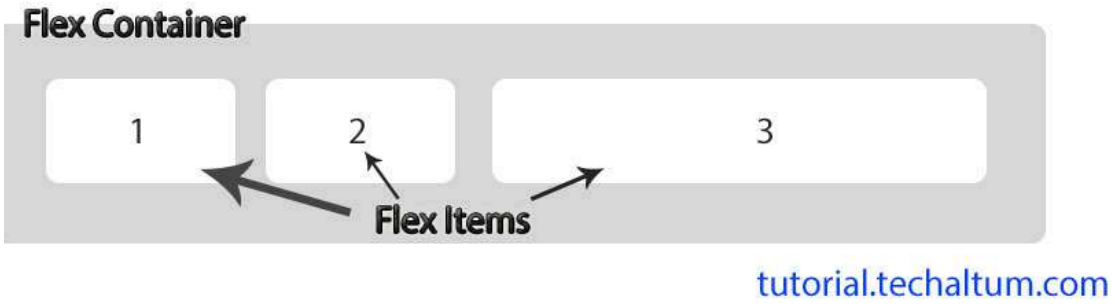
advantages

layout designs
ntage of flexible layouts.
orts IE 10 and above browsers.
height and width option available.
o change direction of flex columns.

nd below, please use **css float based layouts**.

properties

alue of css display . By using display flex in parent element, child elements automatically align like column or row with auto width and auto



es of flex container

flex / inline-flex	compulsory
ction	
p	
v	
ontent	
ems	

Celebrating 13 Years

Properties of flex container

/ Flex

compulsory

ex is the property of flex container to use flexbox. **CSS Display** property can have value **flex** or **inline-flex**. By using display flex or inline-ent container, the children automatically enable flex content.

display flex

Flex

MPLE

Item 2 Item 3 Item 4

```
in:0; box-sizing:border-box}
ex-container{
  display: flex;

ex-item{
  padding:10px;
  border: 1px solid #ccc;
  margin: 5px

>

ass="flex-container">
v class="flex-item">Item 1</div>
v class="flex-item">Item 2</div>
v class="flex-item">Item 3</div>
v class="flex-item">Item 4</div>
```

rection

tion property is given to **flex container** to **change direction of flex items**. By default, flex direction is row.

ections value

default)

Celebrating 13 Years

n-reverse

ge Flex directions

row-reverse | column | column-reverse

MPLE

Item 2 | Item 3 | Item 4

```
in:0; box-sizing:border-box}
ex-container{
 display:flex;
 flex-direction: row;

ex-item{
 padding:0 10px;
 border: 1px solid #ccc;
 margin: 5px

>

ass="flex-container">
v class="flex-item">Item 1</div>
v class="flex-item">Item 2</div>
v class="flex-item">Item 3</div>
v class="flex-item">Item 4</div>
```

rap

will always fit in one row even if content is more. **flex wrap** is used to allow items to wrap in next line or wrap in reverse direction.

ap values

p *(default)*
wrap to next line)
reverse *(multiple line but in reverse from bottom to top)*

ge Flex wrap

ap | wrap | wrap-reverse

MPLE

Item 2 | Item 3 | Item 4 | Item 5 | Item 6 | Item 7 | Item 8 | Item 9 | Item 10 | Item 11 | Item 12 | Item 13 | Item 14 | Item 15 | Item 16

```
in:0; box-sizing:border-box}
ex-container{
 display:flex;
```

Celebrating 13 Years

```
.ex-item{
  padding:0 10px;
  border: 1px solid #ccc;
  margin: 5px

:>

.ass="flex-container">
.v class="flex-item">Item 1</div>
.v class="flex-item">Item 2</div>
.v class="flex-item">Item 3</div>
.v class="flex-item">Item 4</div>
.v class="flex-item">Item 5</div>
.v class="flex-item">Item 6</div>
.v class="flex-item">Item 7</div>
.v class="flex-item">Item 8</div>
.v class="flex-item">Item 9</div>
.v class="flex-item">Item 10</div>
.v class="flex-item">Item 11</div>
.v class="flex-item">Item 12</div>
.v class="flex-item">Item 13</div>
.v class="flex-item">Item 14</div>
.v class="flex-item">Item 15</div>
.v class="flex-item">Item 16</div>
```

ɔw

s the shorthand for **flex-direction** and **flex-wrap** properties. The default value is *row nowrap*.

MPLE

```
;in:0; box-sizing:border-box}
ontainer{
play:flex;
x-flow: column wrap;

tem{
lding:0 10px;
der: 1px solid #ccc;
gin: 5px

:>
.ass="flex-container">
.v class="flex-item">Item 1</div>
```

Celebrating 13 Years

```
.v class="flex-item">Item 4</div>
```

Content

ntent property is used to justify content in main axis. This can adjust items to left, center, right or add space in between.

:ontent values

art *(default)*
ıd

-between
-around
-evenly

ıge justify-content

start | flex-end | center | space-between | space-around | space-evenly

MPLE

Item 2 | Item 3

```
;in:0; box-sizing:border-box}
.ex-container{
  display:flex;
  flex-flow:row wrap;
  justify-content: flex-start;

.ex-item{
  padding:0 10px;
  border: 1px solid #ccc;
  margin: 5px

:>

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

tems

ns property define the behavior of how flex item laid out across horizontal axis. By-default, the value is stretch.

of align-items

ı
art
ıd

ıe

Celebrating 13 Years

MPLE

3

```
;in:0; box-sizing:border-box}
.ex-container{
  display:flex;
  flex-flow:row wrap;
  align-items: stretch;

.ex-item{
  padding:0 10px;
  border: 1px solid #ccc;
  margin: 5px;

:>

.ass="flex-container">
.v class="flex-item"> 1 </div>
.v class="flex-item"> 2 </div>
.v class="flex-item"> 3 </div>
```

erty of Flexbox and Grid add gutters or gap in multi column layout, flex items or grid items. Gap is shortcut or row-gap and column-gap.

rgin, gap property add gaps between elements only, but margin add gap even outside last or first item.

MPLE

2 3

```
;in:0; box-sizing:border-box}
.ex-container{
  display:flex;
  gap: 1rem;

.ex-item{
  padding:0 10px;
  border: 1px solid #ccc;

:>

.ass="flex-container">
.v class="flex-item"> 1 </div>
.v class="flex-item"> 2 </div>
```

Celebrating 13 Years

Properties of flex items

perty of flex item set the order in which they appear in parent container. The default order is 0. value of order is always a number. order sitive or negative.

all flex items are in order of document flow.

MPLE

3

```
in:0; box-sizing:border-box}
ex-container{
  display:flex;
  flex-flow:row wrap;

ex-item{
  padding:0 10px;
  border: 1px solid #ccc;
  margin: 5px;
  order:0;

>

ass="flex-container">
v class="flex-item"> 1 </div>
v class="flex-item"> 2 </div>
v class="flex-item"> 3 </div>
```

ow

property of flex item defines the growth ability. The default value for all flex items is 0. Flex grow when not zero can use the remaining ontainer

w value

ult)
er

0 means width of flex item as per content only.

1 means width of flex item will be distributed equally to all items.

: 0 1

Celebrating 13 Years

3

```
;in:0; box-sizing:border-box}
.ex-container{
  display:flex;
  flex-flow:row wrap;

.ex-item{
  padding:0 10px;
  border: 1px solid #ccc;
  margin: 5px;
  flex-grow: 0;

:>

.ass="flex-container">
.v class="flex-item"> 1 </div>
.v class="flex-item"> 2 </div>
.v class="flex-item"> 3 </div>
```

ow 2

MPLE

2 3

```
;in:0; box-sizing:border-box}
.ex-container{
  display:flex;
  flex-flow:row wrap;

.ex-item{
  padding:0 10px;
  border: 1px solid #ccc;
  margin: 5px;
  flex-grow: 1;

;2{
ex-grow:2;

:>

.ass="flex-container">
.v class="flex-item"> 1 </div>
.v class="flex-item fg2"> 2 </div>
.v class="flex-item"> 3 </div>
```

Celebrating 13 Years

ink 0 means flex item will not shrink.

ink values

ult)

MPLE

ık: ○ 0 ◉ 1

	Column	Column	Column	Column	Column	Column	Column

```
in:0; box-sizing:border-box}
ex-container{
 display:flex;

ex-item{
 padding:0 10px;
 border: 1px solid #ccc;
 margin: 5px;
 flex-shrink: 1;

>

ass="flex-container">
v class="flex-item">Column</div>
v class="flex-item">Column</div>
v class="flex-item">Column</div>
v class="flex-item">Column</div>
v class="flex-item">Column</div>
v class="flex-item">Column</div>
v class="flex-item">Column</div>
v class="flex-item">Column</div>
```

sis

property of flex item defines the initial main size (width or height) of flex item. The flex basis value could be auto, %, px, em or rem. The ue of **flex-basis** is 0. We can assign value to flex-basis based on the percentage required. For example, 20% means the initial width of flex e 20%, and sum of width of other items will be 80%.

is auto

: auto will apply automatic or equal width to flex items. If flex items are two, then its 50%, if flex items are 3, then 33.33% and if its 4, then

MPLE

	auto	auto

Celebrating 13 Years

```
;in:0; box-sizing:border-box}
.ex-container{
  display:flex;
  flex-flow:row wrap;

.ex-item{
  padding:0 10px;
  border: 1px solid #ccc;
  margin: 5px;
  flex-grow: 1;
  flex-shrink:0;
  flex-basis:auto;

:>

.ass="flex-container">
.v class="flex-item"> auto </div>
.v class="flex-item"> auto </div>
.v class="flex-item"> auto <//div>
```

ic px or %

px or % defines initial width of flex-items in px or %. Like 25%, 20px, 4em, 5rem etc.

MPLE

50%

```
yle>
margin:0; box-sizing:border-box}
 .flex-container{
     display:flex;
     flex-flow:row wrap;
 }
 .flex-item{
     padding:0 10px;
     border: 1px solid #ccc;
     margin: 5px;
     flex-grow: 1;
     flex-shrink:0;
     flex-basis:50%;
 }
tyle>

.v class="flex-container">
 <div class="flex-item"> 50% </div>
 <div class="flex-item"> 50% </div>
liv>
```

sis Vs Width

overwrites css width property. But max-width can be used with flex-basis.

Celebrating 13 Years

max-width > flex-basis > width

s, **flex-basis** can overwrite **width** property, but **max-width** can overwrite **flex-basis**.

shorthand for flex-grow, flex-shrink and flex-basis. The default value of **flex** is *0 1 0%*. The first value, i.e flex-grow is compulsory and
are optional.

efer **flex shorthand** property for flex-grow, flex-shrink and flex-basis for simplicity.

MPLE

	2

```
in:0; box-sizing:border-box}
ex-container{
  display:flex;
  flex-flow:row-wrap;

ex-item{
  padding:8px;
  border: 1px solid #ccc;
  flex: 1 0 auto;

>

ass="flex-container">
v class="flex-item"> 1 </div>
v class="flex-item"> 2 </div>
```

n row-wrap

MPLE

	Col 2	Col 3	Col 4
	Col 6	Col 7	Col 8

```
in:0; box-sizing:border-box}
ontainer{
play:flex;
x-flow:row wrap;

tem{
lding:8px;
der: 1px solid #ccc;
x: 1 0 25%;

>
```

Celebrating 13 Years

```
ass="flex-item">col 1 </div>
ass="flex-item">col 2 </div>
ass="flex-item">col 3 </div>
ass="flex-item">col 4 </div>
ass="flex-item">col 5 </div>
ass="flex-item">col 6 </div>
ass="flex-item">col 7 </div>
ass="flex-item">col 8 </div>
```

n column-wrap

MPLE

	Col 5
	Col 6
	Col 7
	Col 8

```
ontainer{
play:flex;
x-flow:column wrap;
ght:200px;

tem{
lding:8px;
der: 1px solid #ccc;
x: 1 0 25%;

>

ass="flex-container">
ass="flex-item">col 1 </div>
ass="flex-item">col 2 </div>
ass="flex-item">col 3 </div>
ass="flex-item">col 4 </div>
ass="flex-item">col 5 </div>
ass="flex-item">col 6 </div>
ass="flex-item">col 7 </div>
ass="flex-item">col 8 </div>
```

elf

is the property of flex items. It is used to align an individual flex item on y-axis.

lf Values

art

nd

Celebrating 13 Years

MPLE

2

3

```
.ex-container{
  display:flex;
  flex-flow:row wrap;
  height:150px;

.ex-item{
  padding:0 10px;
  border: 1px solid #ccc;
  margin: 5px;
  flex:1 0 auto;

.ex-item:nth-child(1){align-self:flex-start}
.ex-item:nth-child(2){align-self:center}
.ex-item:nth-child(3){align-self:flex-end}
:>

.ass="flex-container">
.v class="flex-item"> 1 </div>
.v class="flex-item"> 2 </div>
.v class="flex-item"> 3 </div>
```

sed layout

CSS Grids→

nd Tutorial

- itend Tutorial
- 1L Tutorial
- 1L5 Tutorial
- Tutorial
- 3 Tutorial
- Script
- ery Tutorial
- S Tutorial
- to HTML
- tstrap

Backend Tutorial

1. Node JS
2. Asp.net
3. MVC
4. Linq
5. WCF
6. Entity Framework
7. PHP
8. Java
9. Python
10. Ruby

Celebrating 13 Years

ech Altum Tutorial is run and maintained by Tech Altum, IT Training Institute in Noida (An IIT Alumni Institute) established in 2012.
For Video Tutorials, Subscribe our Youtube Channel. Tech Altum Youtube Channel.

Our Popular Courses

Development

itend Web Development
Stack Web Development
ance JavaScript with ES6 to ES14
Designing & UI Development
t
ular

Backend Development

1. Node JS
2. Java
3. Python
4. Php
5. Asp.net

room and Online classes are conducted by 12+ Exp and Certified Alumni of IIT or Corporate Trainers having minimum 10+ experience in Teaching and Development.