FlexBox

Theory :

The FlexBox Layout module aims at providing a more efficient way to lay out, align and distribute space among items in a container.The main idea behind the flex layout is to give the container the ability to alter its items’ width/height (and order) to best fill the available space (mostly to accommodate to all kind of display devices and screen sizes).

Properties of flexbox :

Following are the main two categories for properties of flexbox -

1. Flex container
2. Flex item

1. Flex container :

Flex container have the properties for parent . it is also divided into sub-categories.

Following are sub-categories of flex container -

- Display
- Flex - direction
- Flex - wrap
- Flex - flow
- Justify - content
- Align - item
- Align - content
- Gap

- Display -

  This defines a flex container; inline or block depending on the given value. It enables a flex context for all its direct children.

```
.container {
 display: flex;
}
```

- Flex - Direction -
  It is establish the main axis therefore it defines flex direction

```
.container {
 flex-direction: row | row-reverse | column | column-reverse;
}
```

- Flex - Wrap -
  By default, flex items will all try to fit onto one line. You can change that and allow the items to wrap as needed with this property

```
.container {

 flex-wrap: nowrap | wrap | wrap-reverse;

}
```

- Flex - Flow -

  This is a shorthand for the flex - direction and flex - wrap properties, which together define the flex container’s main and cross axes

```
.container {

 flex-flow: column wrap;

}
```

- Justify - Content -

It helps distribute extra free space leftover when either all the flex items on a line are inflexible, or are flexible but have reached their maximum size

```
.container {

 justify-content: flex-start | flex-end | center | space-between |
space-around | space-evenly | start | end | left | right ... + safe |
unsafe;

}
```

- Align - Item -

  This defines the default behavior for how flex items are laid out along the cross axis on the current line

```
..container {

 align-items: stretch | flex-start | flex-end | center | baseline |
first baseline | last baseline | start | end | self-start | self-end +
... safe | unsafe;

}
```

- Align - Content -

  This property only takes effect on multi-line flexible container

```
.container {
 align-content: flex-start | flex-end | center | space-between |
space-around | space-evenly | stretch | start | end | baseline | first
baseline | last baseline + ... safe | unsafe;
}
```

- Gap -
  It is control space between flex items. It apply space between that item not on their outer eages

```
.container {
 display: flex;
 ...
 gap: 10px;
 gap: 10px 20px; /* row-gap column gap */
 row-gap: 10px;
 column-gap: 20px;
}
```

2. Flex item -

It is an property for children. It is divided into following sub categories.

- Order
- Flex - grow
- Flex - shrink
- Flex - basis
- Flex
- Align - Self

- Order -
  By default, flex items are laid out in the source order. However, the order property controls the order in which they appear in the flex container

```
.item
{
order: 5;
}
```

- Flex - grow -
  It defines the ability to flex item to grow if necessary

```
.item {
 flex-grow: 4;
}
```

- Flex - shrink -
  It shrink the flex item if necessary

```
.item {
 flex-shrink: 3;
}
```

- Flex - basis -
  The flex basis is used to define default size of element before remaining space is distributed

```
.item {
 flex-basis:  | auto;
}
```

- Flex -
  It is short-hand for flex-grow, flex-shrink and flex-basis

```
.item {
 flex: none | [ <'flex-grow'> <'flex-shrink'>? || <'flex-basis'> ]
}
```

- Align - self -
  This allows default alignment to be overridden for individual flex item

```
.item {
 align-self: auto | flex-start | flex-end | center | baseline |
stretch;
}
```