Using HTML

Using HTML & CSS

Cascading Style Sheets

What is CSS?

Cascading Style Sheets (CSS) is a simple mechanism for adding style to Web documents such as

- Fonts
- Layouts
- Spacing
- Colors

Syntax:

Selector Declaration

```
{  color:  red  }
```

Property Value

Kinds of CSS

Inline

Inline CSS is used to style a specific HTML element. For this CSS style, you'll only need to add the **style** attribute to each HTML tag

Internal

Internal/Embedded CSS requires us to use style element in head section of HTML document.

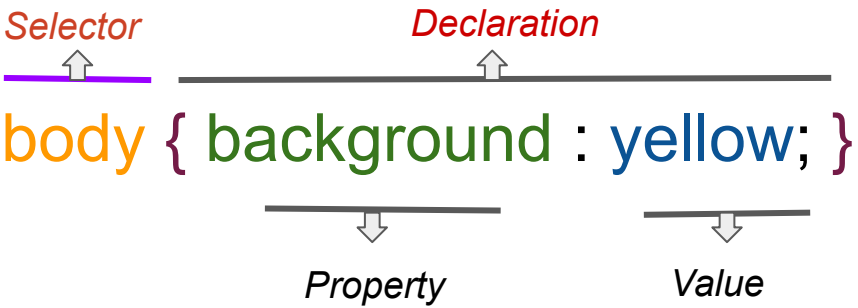
External

With External CSS, you'll link your web pages to an external **.css** file.

Inline CSS	Internal CSS	External CSS
Benefits : ● We need not to use any selectors for selecting a specific HTML element. ● We can use styles directly with in the HTML tag by using style attribute	**Benefits** : ● We can use different selectors in same HTML documents. ● We need not to use multiple files for implementing this kind of CSS	**Benefits** : ● We can implement the structure of HTML code and CSS code. ● We can reuse the styles for various web documents.
Limitations : ● Adding styles to every element is a time consuming process. ● Combining of HTML and CSS makes the code messy	**Limitations** : ● We have to implement style properties along with with the HTML document it affects the performance of web site. ● We can't reuse the styles which were defined for one file in multiple files.	**Limitations** : ● By using multiple files, it affects the download time. ● Designed part not fully loaded until the css file loads

Selector in CSS

A CSS selector allows us to select the content we want to style



Types of Selectors in CSS

Basic selectors

- id
- class
- element
- universal

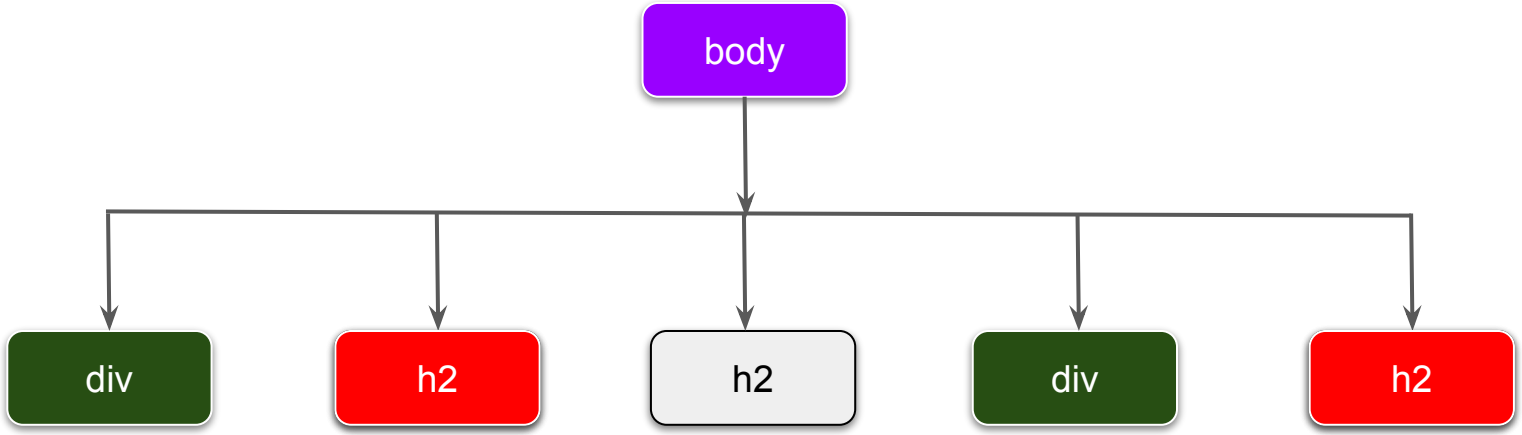
Relational selectors

- descendant combiner
- child combiner
- adjacent sibling
- general sibling

Attribute selectors

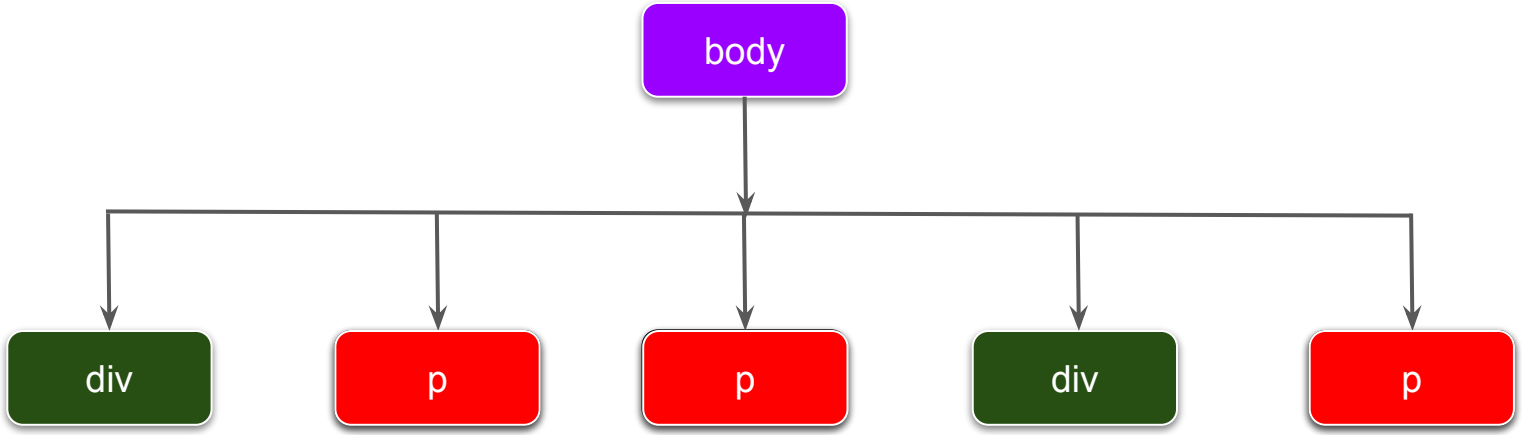
Pseudo selectors

Adjacent sibling selector



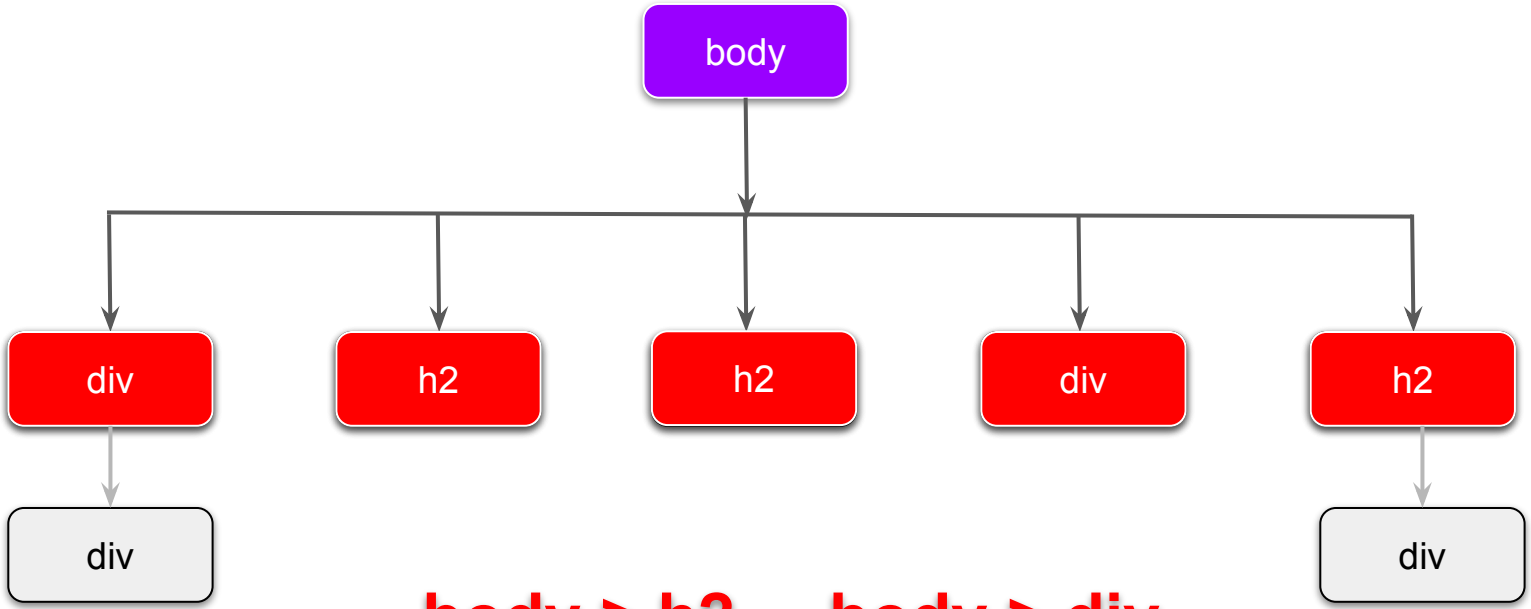
div + h2

General sibling selector



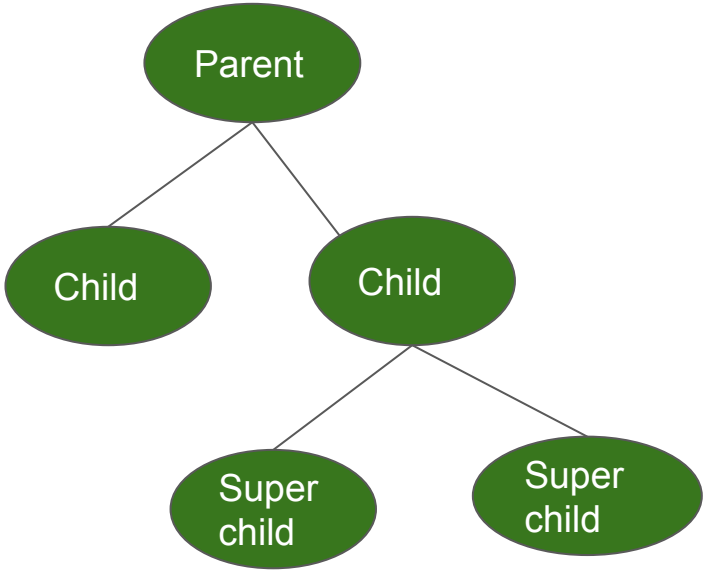
div ~ p

Child combinator in CSS

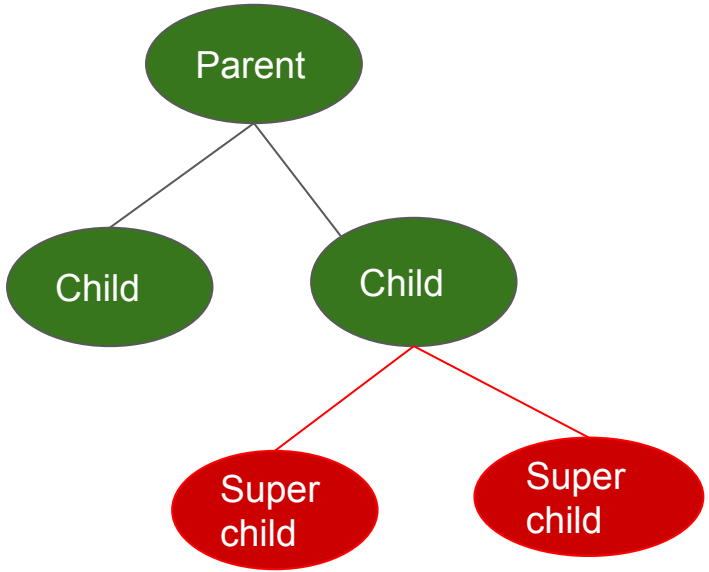


body > h2 **body > div**

Descendant combiner vs Child combiner



Descendant combiner



Child combiner

CSS units

Absolute lengths

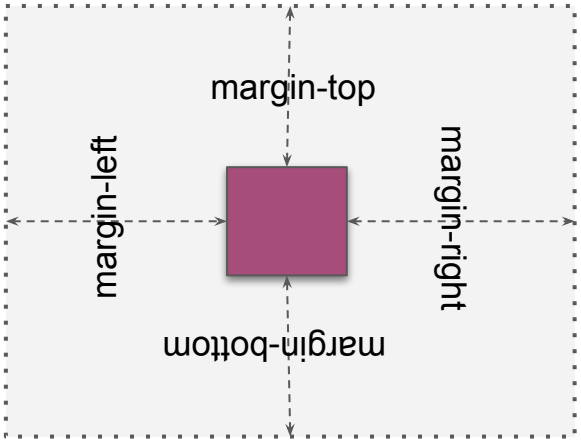
cm => 1 cm = 96px / 2.54
mm => 1 mm=1 cm/10
Q => 1 Q = 1 / 40 of 1 cm
in => 1 in = 96px or 2.54cm
pc => 1 pc = 1 in / 6
pt => 1 pt = 1 in / 72
px => 1 px = 1 / 96 of 1 in

Relative lengths

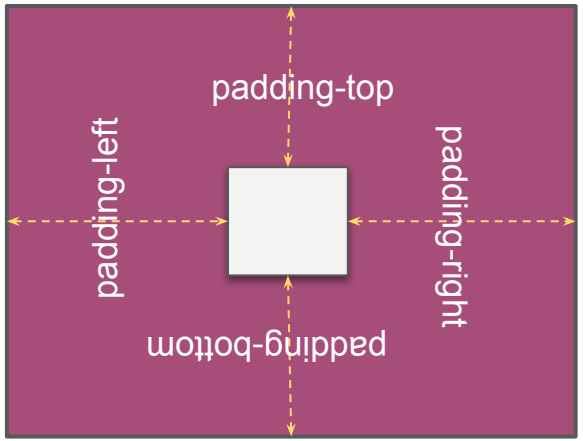
em => Font size of the parent, in the case of typographical properties like font-size, and font size of the element itself, in the case of other properties like width.
ex => x height of the element's font
rem => Font size of the root element
lh => Line height of the element
vh => 1% of viewport height
vw => 1% of viewport width
vmin => 1% of viewport smaller dimension
vmax=> 1% of viewport larger dimension
% => Depends on parent,s size

Margin vs Padding

Margin properties are used to create space around the element



Padding properties are used to create space around the element's content



Margin vs Padding

```
margin:     1px     1px     0px     2px;
```

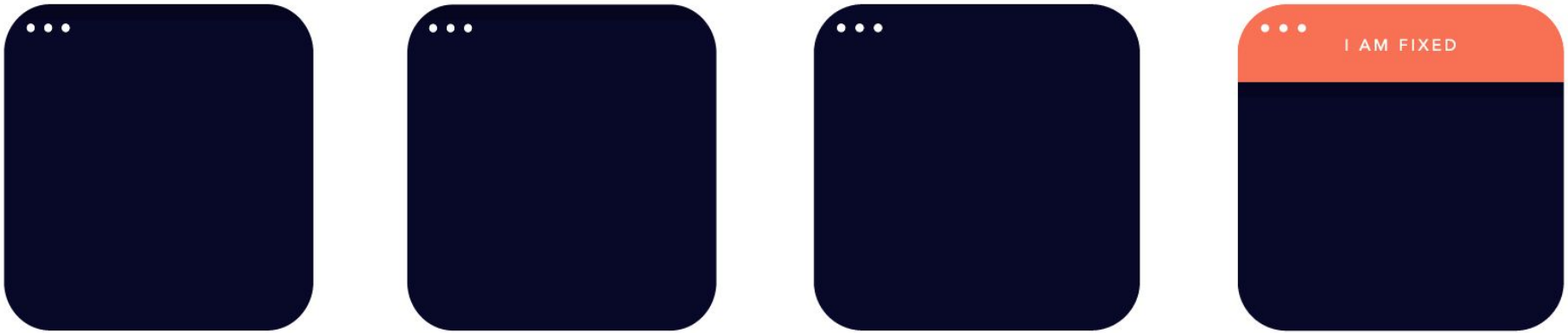

```
margin:    1px;
```

margin-top : 1px;
margin-right : 1px;
margin-bottom : 1px;
margin-left : 1px;

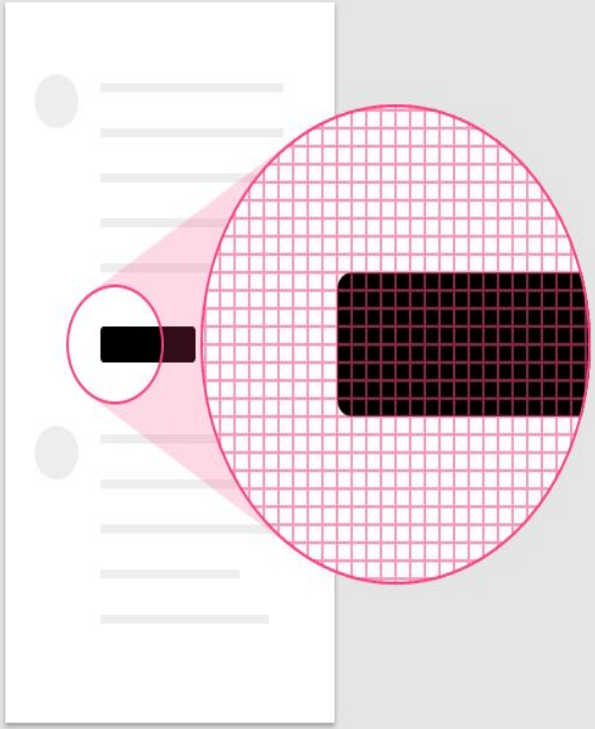
```
margin:    1px    auto;
```

margin-top : 1px;
margin-right : auto;
margin-bottom : 1px;
margin-left : auto;

Positions in CSS



Responsive web design



```
<meta name="viewport"
content="width=device-width,
initial-scale=1.0">
```

The **viewport** is the user's visible area of a web page. It varies with the device

The **width=device-width** sets the width of the page to follow the screen-width of the device

The **initial-scale=1.0** sets the initial zoom level when the page is first loaded by the browser

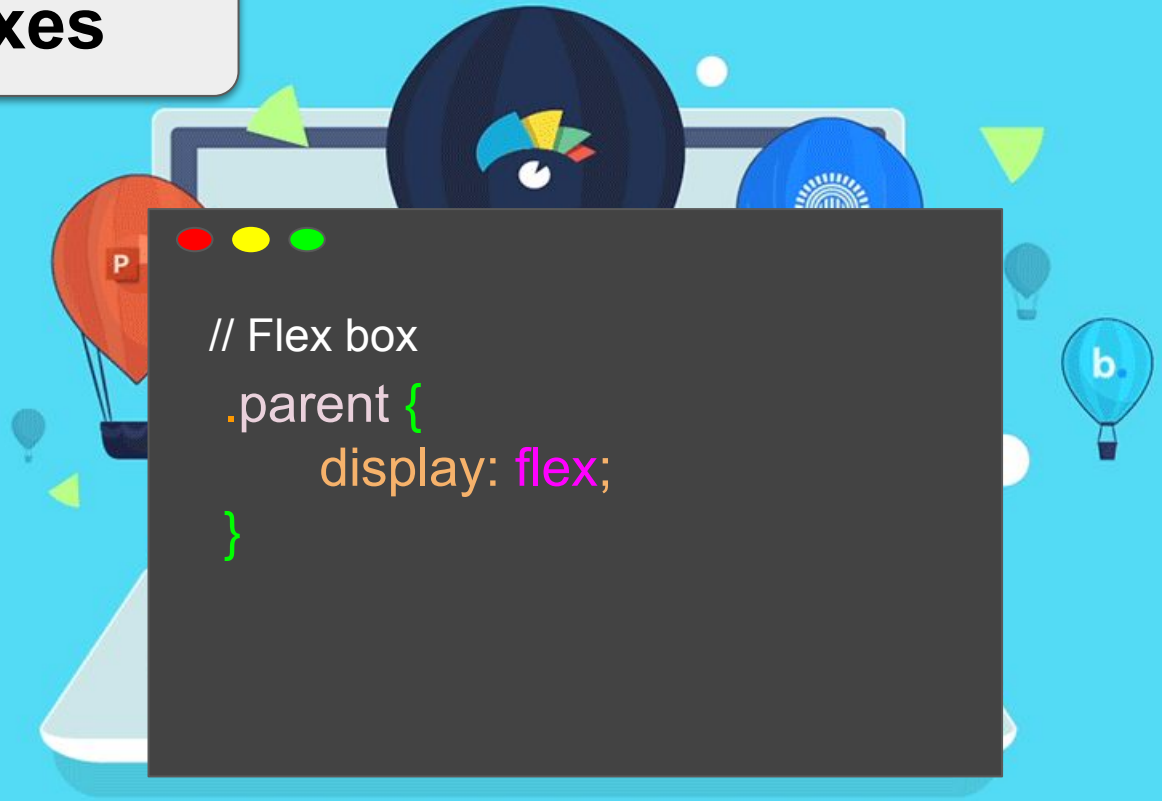
Mobiles => (>=320px to < 768px)

Tablets => (>=768px to <1024px)

Desktops => (>=1024px)

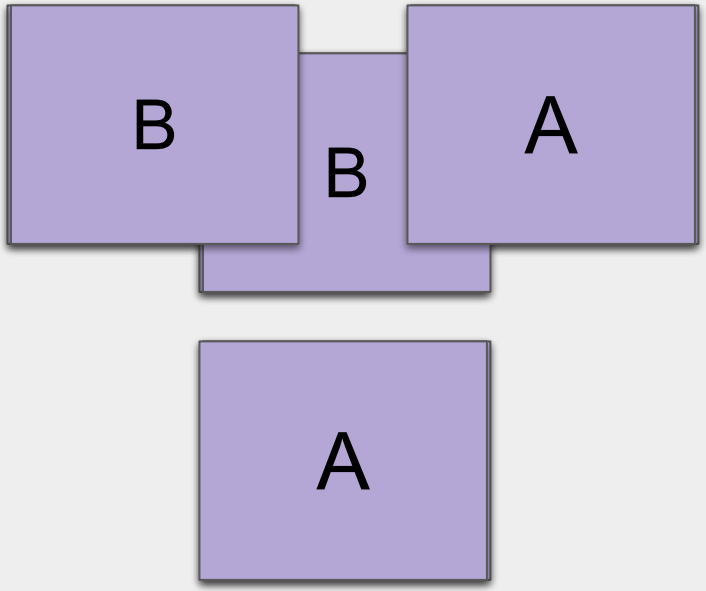
SCREEN RESOLUTIONS

Flexboxes



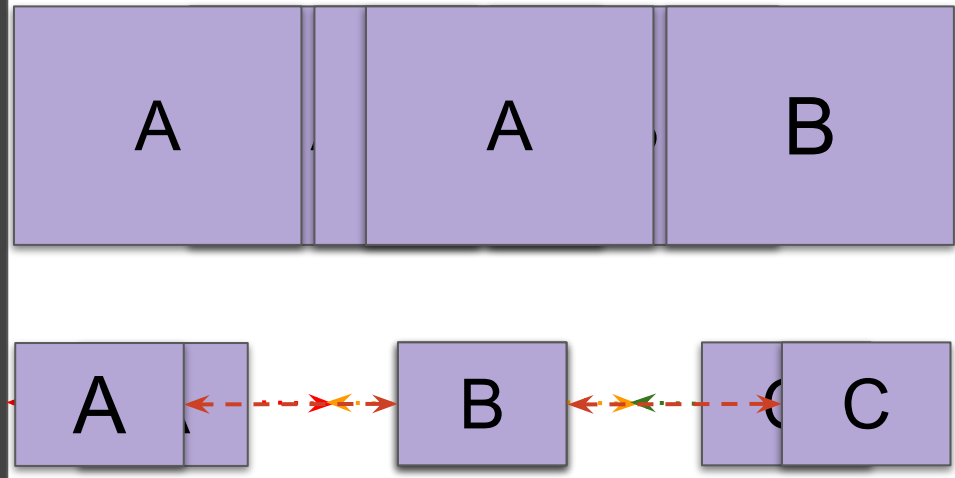
flex-direction : row | column | row-reverse | column-reverse;

```
// Flex box

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



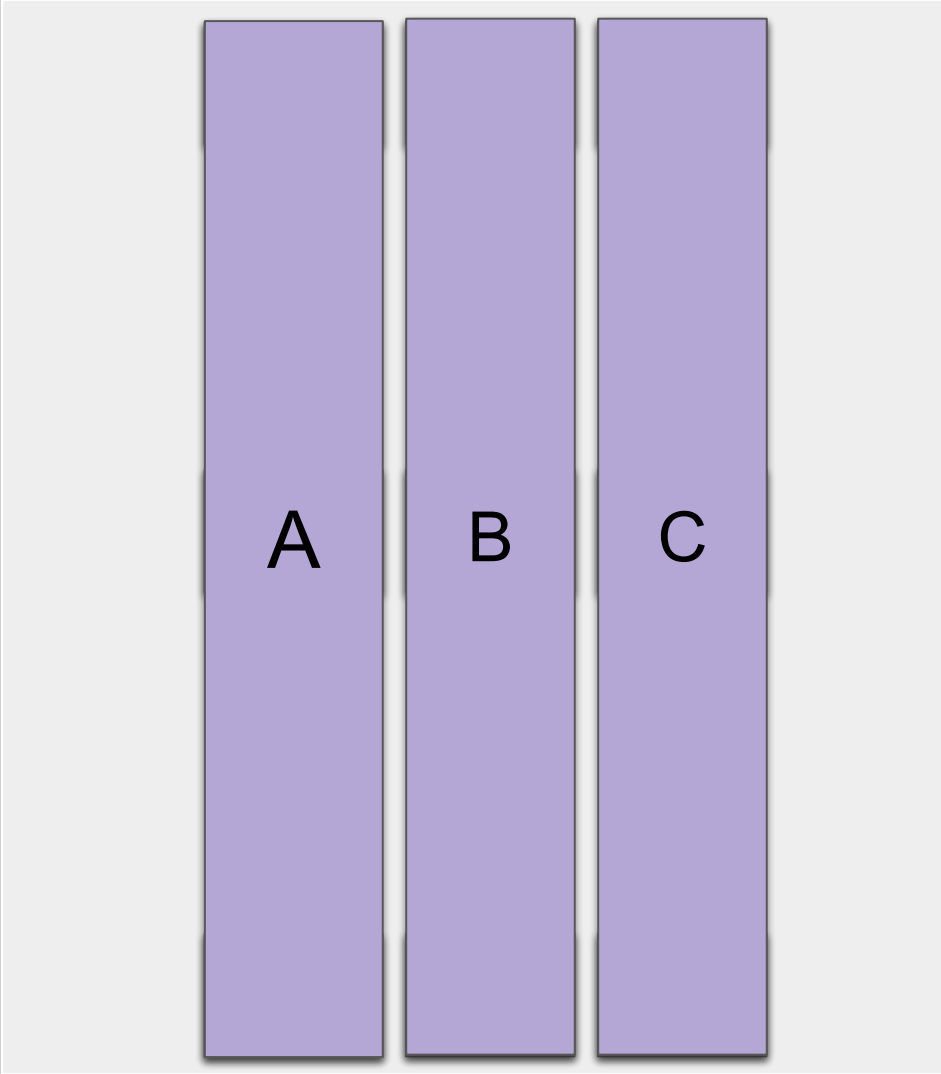
justify content : center | flex-start | flex-end |
space-around | space-between;

```
// Flex box

 .parent {
        display: flex;
        flex-direction: row;
        justify-content: space-between;
 }
```



```
align-items : flex-start | flex-end | center |
              stretch ;
```

```
// Flex box

 .parent {
        display: flex;
        flex-direction: row;
        align-items: flex-start;
 }
```



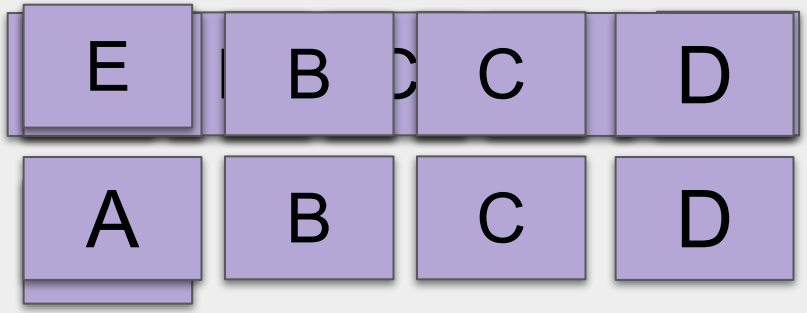
flex-wrap : wrap | nowrap | wrap-reverse |

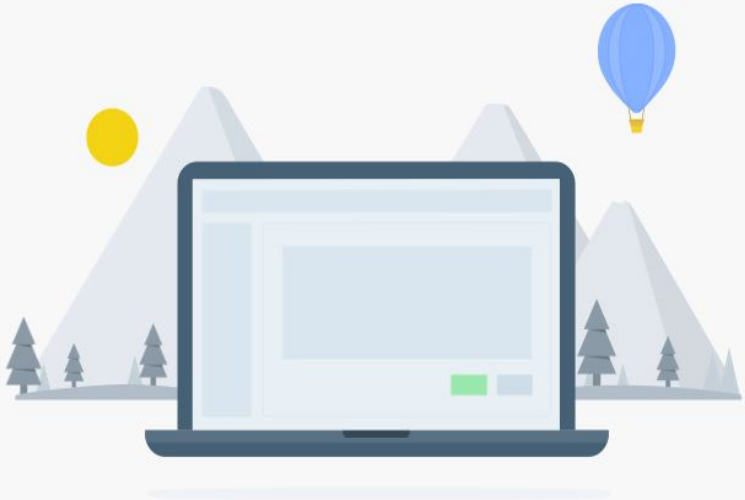
// Flex box

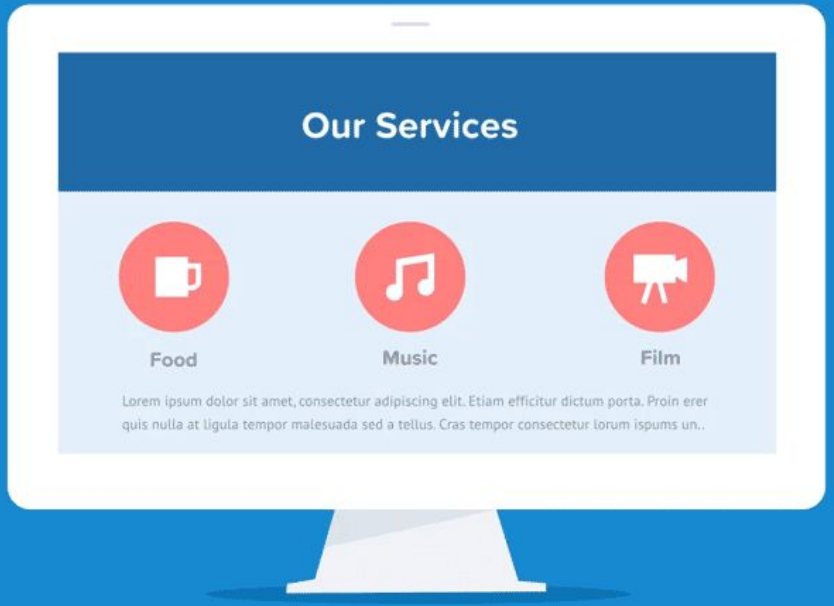
```
.parent {
    display: flex;
    flex-direction: row;
    flex-wrap: wrap-reverse;
}
```

flex- direction + flex-wrap = flex-flow

flex-flow: row wrap;

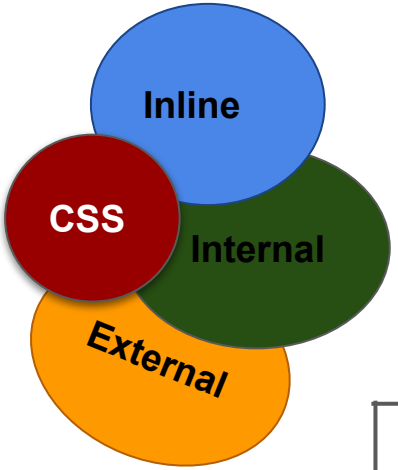





Our Services
Food
Music
Film
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam efficitur dictum porta. Proin erer
quis nulla at ligula tempor malesuada sed a tellus. Cras tempor consectetur lorum ispums un..

CSS Positions

Static
Relative
Fixed
Absolute
Sticky



Selectors
- Basic (#, ., tagName, *)
- Relational
  - Descendant (m n)
  - Child combiner (m>n)
  - Adjacent sibling (m + n)
  - General Sibling (m ~ n)
- Attribute
- Pseudo (:)

Layout Properties
- Margin & Padding
- Shadows
- Box model
- Positions
  - Static
  - Relative
  - Absolute
  - Fixed
  - Sticky

Backgrounds
Font styles

Grid system
Responsive Web Design (meta)
FlexBoxes
- Display
- Grid-template-rows
- Grid-template-columns
- Grid-column-gap
- Grid-row-gap

- Display
- Flex-direction
- Flex-wrap
- Flex flow
- Justify-content
- Align-items

GRID LAYOUT

DISPLAY

GRID-TEMPLATE-ROWS

GRID-TEMPLATE-COLUMNS

GRID-COLUMN-GAP

GRID-ROW-GAP

Grid container