



HTML

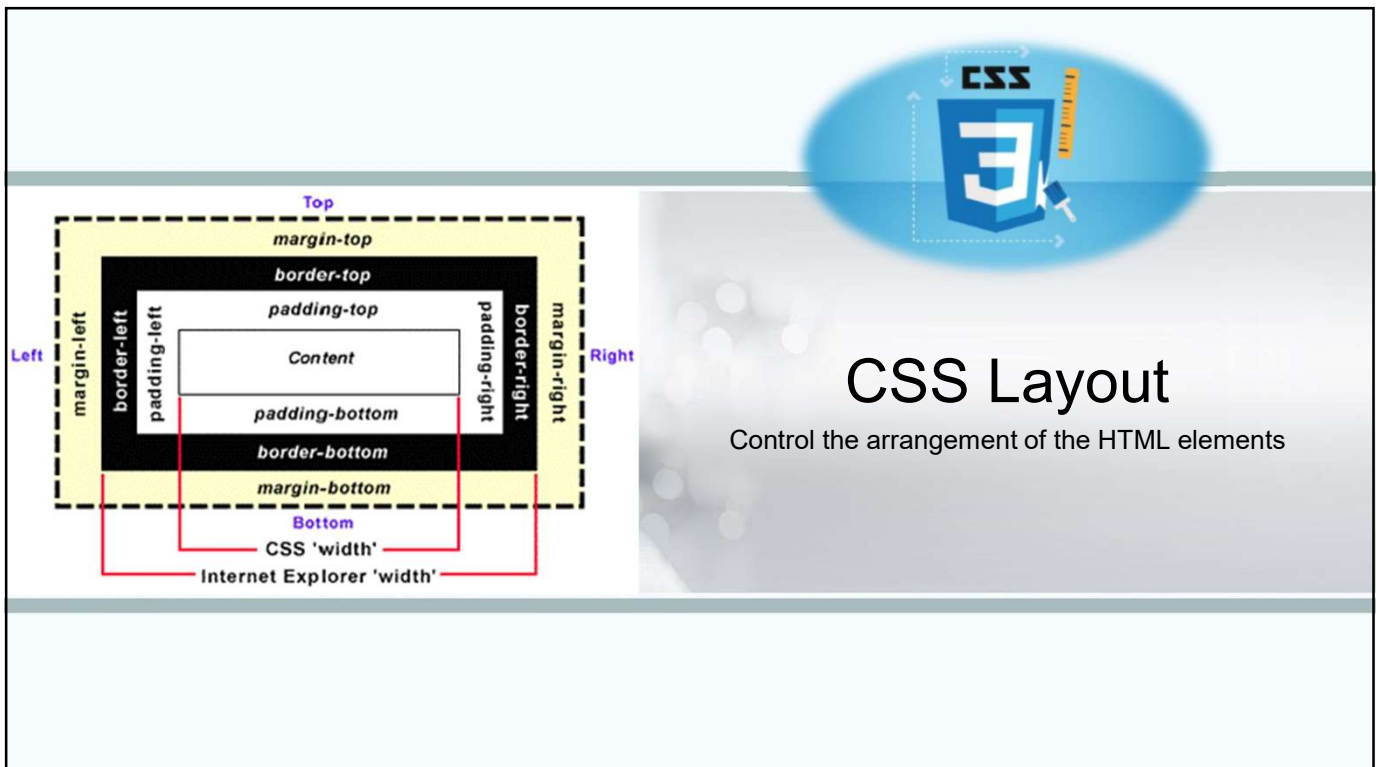
CSS

JavaScript

Web Programming

Lecturer: Ung Văn Giàu
Email: giâu.ung@eiu.edu.vn

1



CSS

CSS Layout

Control the arrangement of the HTML elements

Diagram illustrating the CSS Box Model layers (from outermost to innermost):

- margin-left, margin-top, margin-right, margin-bottom
- border-left, border-top, border-right, border-bottom
- padding-left, padding-top, padding-right, padding-bottom
- Content

Labels: Top, Bottom, Left, Right.

Annotations:

- CSS 'width' (indicated by a red line from the padding-right and padding-left layers)
- Internet Explorer 'width' (indicated by a red line from the border-right and border-left layers)

2

Contents

01 Display

02 Width / Height

03 Overflow

04 Margin and Padding

05 Box Model

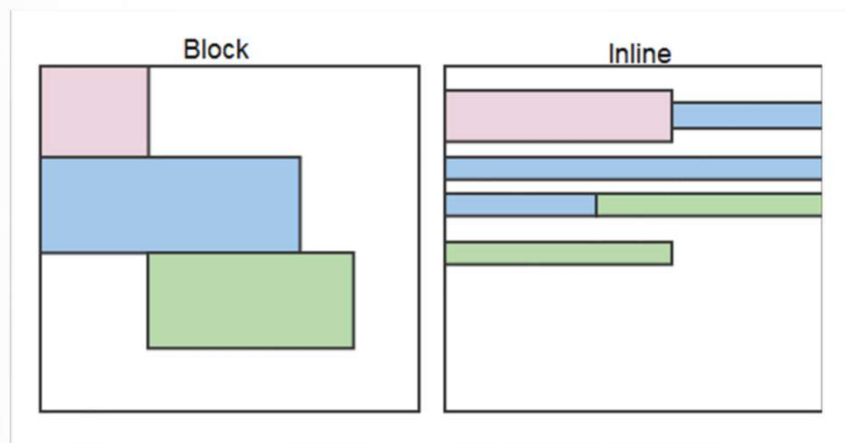
06 Positioning

07 Floating

08 Flexbox

3

3



1. Display

4

Display

- **display** controls the display of the element and the way it is rendered and if **breaks** should be placed before and after the element
- Display **values**:
 - **inline**: **no breaks** are placed before or after (is an inline element)
height and width **depend on** the **content**
 - **block**: **breaks are placed before AND after** the element (is a block element)
height and width may **not depend on** the size of the **content**

5

5

Display Values

- Display **values**:
 - **none**: element is **hidden** and its dimensions are not used to calculate the surrounding elements rendering
differs from **visibility: hidden**
 - **inline-block**: no breaks are placed before and after (like inline)
height and width can be **applied** (like block)
 - **table**, **table-row**, **table-cell**: the elements are arranged in a table-like layout

6

6

Display Values

```
<!DOCTYPE html>
<html>
<head>
<style>
  p {
    color: red;
  }
  p.ex1 {
    display: none;
  }
  p.ex2 {
    display: inline;
    width: 300px
  }
  p.ex3 {
    display: block;
  }
  p.ex4 {
    display: inline-block;
    width: 300px;
  }
</style>
</head>
```

```
<body>
  <div>
    <p class="ex1">HELLO WORLD!</p>
    EIU-Eastern International University
  </div>
  <div>
    <p class="ex2">HELLO WORLD!</p>
    EIU-Eastern International University
  </div>
  <div>
    <p class="ex3">HELLO WORLD!</p>
    EIU-Eastern International University
  </div>
  <div>
    <p class="ex4">HELLO WORLD!</p>
    EIU-Eastern International University
  </div>
</body>
</html>
```

EIU-Eastern International University
HELLO WORLD! EIU-Eastern International University

HELLO WORLD!

EIU-Eastern International University

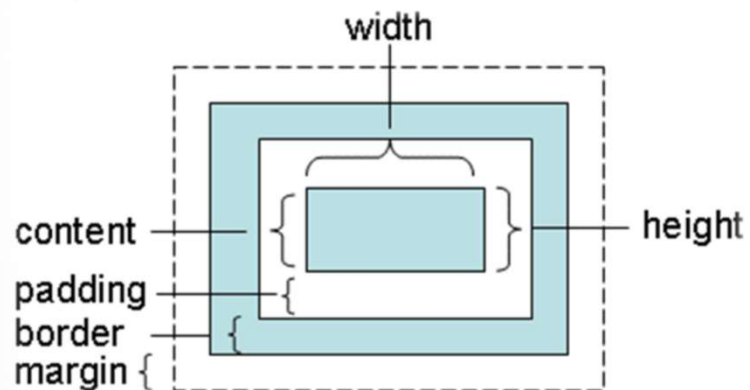
HELLO WORLD!

EIU-Eastern International University

<https://codepen.io/Vn-T-i-the-styleful/pen/vYbXaNO?editors=1000>

7

7



2. Width/Height

8

Width

- **width** – defines numerical value for the width of element, e.g. 200px, 50%
- **width applies only for block elements**
 - The width is **100% by default**
 - The width of inline elements is always the width of their content, by concept
- **min-width** - defines the minimal width
min-width overrides width if (width < min-width)
- **max-width** - defines the maximal width
max-width overrides width if (width > max-width)

9

9

Height

- **height** – defines numerical value for the height of element, e.g. 100px, 100vh
- **height applies only on block elements**
The height of inline elements is always the height of their content
- **min-height** - defines the minimal height
min-height overrides height
- **max-height** - defines the maximal height
max-height overrides height

10

10

Height

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  max-width: 200px;
  min-width: 100px;
  background-color: powderblue;
}
</style>
</head>
<body>
<h2>Set the max-width and min-width of an element</h2>
<p>Resize the browser window to see the effect.</p>
<div>This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.</div>
</body>
</html>
```

Set the max-width and min-width of an element

Resize the browser window to see the effect.

This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.

<https://codepen.io/Vn-T-i-the-styleful/pen/dyapjBO?editors=1010>

11

11

Width and Height Values

The values of the width and height properties are numerical:

- Pixels (px)
- Percentages

A percent of the available width

- Centimeters (cm)

12

12



13

Overflow

- **overflow** defines the behavior of element when content needs more space than the available
- overflow values:
 - **visible** (default) – content spills out of the element
 - **auto** – show scrollbars if needed
 - **scroll** – always show scrollbars
 - **hidden** – any content that cannot fit is clipped

14

14

Overflow

```

<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
  background-color: lightblue;
  width: 110px;
  height: 50px;
  overflow: scroll;
}
div.ex2 {
  background-color: lightblue;
  width: 110px;
  height: 50px;
  overflow: hidden;
}
div.ex3 {
  background-color: lightblue;
  width: 110px;
  height: 50px;
  overflow: auto;
}
div.ex4 {
  background-color: lightblue;
  width: 110px;
  height: 50px;
  overflow: clip;
}
div.ex5 {
  background-color: lightblue;
  width: 110px;
  height: 50px;
  overflow: visible;
}
</style>
</head>
<body>
<h4>overflow: scroll:</h4>
<div class="ex1">EIU-Eastern International University,EIU-Eastern
International University</div>
<h4>overflow: hidden:</h4>
<div class="ex2">EIU-Eastern International University, EIU-Eastern
International University,EIU-Eastern International University</div>
<h4>overflow: auto:</h4>
<div class="ex3">EIU-Eastern International University,EIU-Eastern
International University,</div>
<h4>overflow: clip:</h4>
<div class="ex4">EIU-Eastern International University,EIU-Eastern
International University,EIU-Eastern International University</div>
<h4>overflow: visible (default):</h4>
<div class="ex5">EIU-Eastern International University,EIU-Eastern
International University,EIU-Eastern International University</div>
</body>
</html>

```

overflow: scroll:

overflow: hidden:

overflow: auto:

overflow: clip:

overflow: visible (default):

overflow: visible (default):

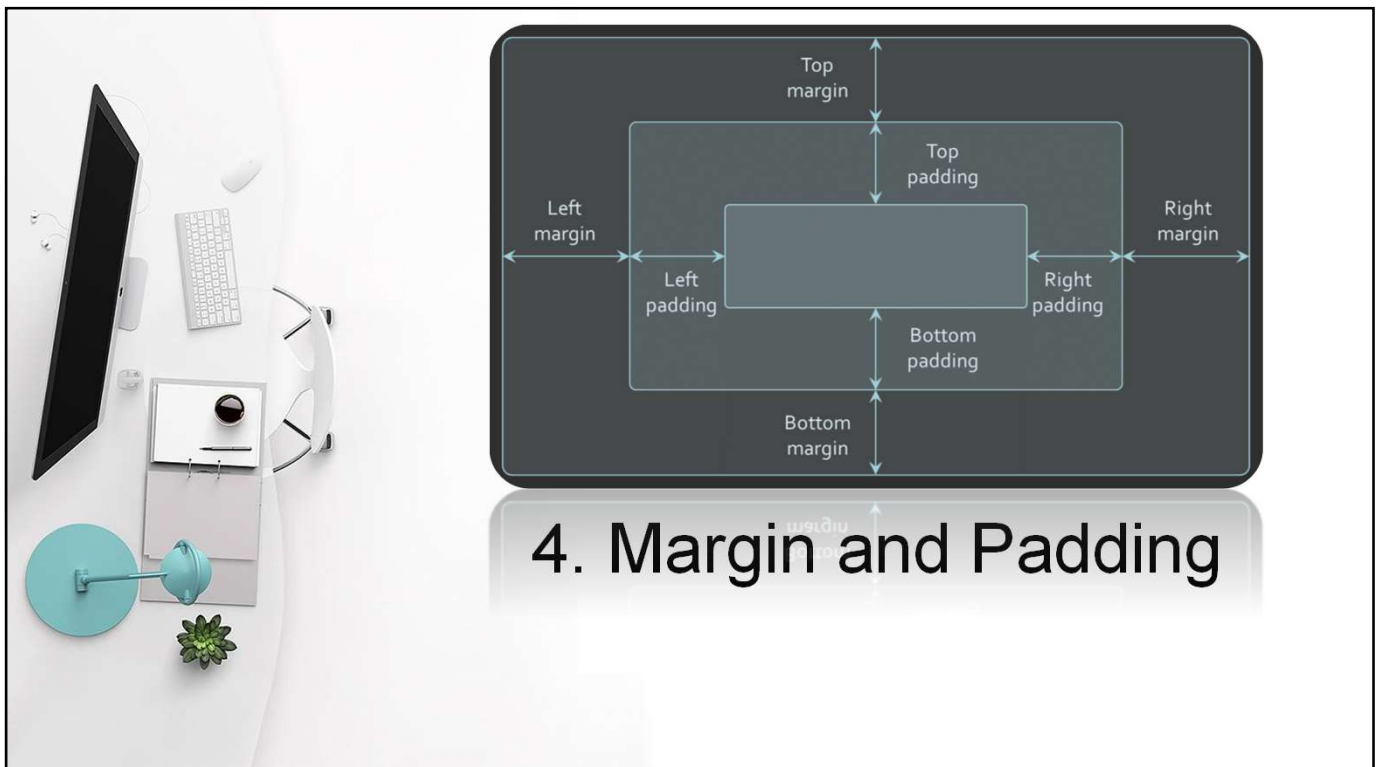
overflow: visible (default):

overflow: visible (default):

overflow: visible (default):

<https://codepen.io/Vn-T+the-styleful/pen/oNmzPvY?editors=1000>

15



16

Most Common Attributes

- margin and padding define the spacing around the element
 - Numerical value, e.g., 10px or -5px
 - Can be **defined** for each of **the four sides separately** – margin-top, padding-left,...
 - **margin** is the **spacing outside** of the border
 - **padding** is the **spacing between** the **border** and the **content**
- Collapsing margins

When the vertical margins of two elements are touching, only the margin of the element with **the largest margin value** will be honored

17

17

Margin and Padding: Short Rules

- **margin: 5px;**
Sets all four sides to have margin of 5 px;
- **margin: 10px 20px;**
top and bottom to 10px, left and right to 20px;
- **margin: 5px 3px 8px;**
top 5px, left/right 3px, bottom 8px
- **margin: 1px 3px 5px 7px;**
top, right, bottom, left (clockwise from top)
- **Same for padding**

18


18

Margin and Padding: Short Rules

```
<!DOCTYPE html>
<html>
<head>
<style>
  .center {
    margin: auto;
    background: lime;
    width: 66%;
  }
  .outside {
    margin: 3px 50px 0 -30px;
    background: cyan;
    width: 66%;
  }
  h4 {
    background-color: lime;
    padding: 20px 50px;
  }
  h3 {
    background-color: cyan;
    padding: 110px 50px 50px 110px;
  }
</style>
```

```
</style>
</head>
<body>
  <h2 style="color:green">Margin</h2>
  <p class="center">EIU-Eastern International University</p>
  <p class="outside">EIU-Eastern International University</p>
  <h2 style="color:green">Padding</h2>
  <h4>EIU-Eastern International University</h4>
  <h3>EIU-Eastern International University</h3>
</body>
</html>
```

Margin


 EIU-Eastern International University

Padding


 EIU-Eastern International University

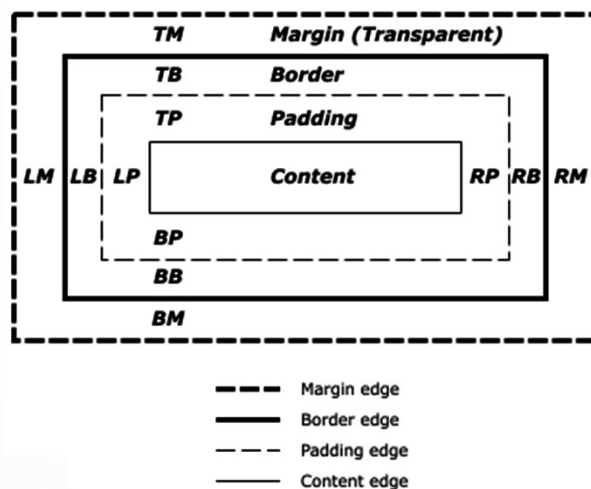

 EIU-Eastern International University

<https://codepen.io/Vn-T-i-the-styleful/pen/PoVGdqR?editors=1010>

19

19

5. Box Model



20

20

box-sizing

CSS 3

- Determine whether you want an element to render its **borders and padding within its specified width**, or **outside** of it.
- Possible **values**:
 - **box-sizing: content-box**
 - ✓ Default
 - ✓ box width: 288px + 10px padding + 1px border on each side = 300px
 - **box-sizing: border-box**
 - box width: 300px, including padding and borders

21

21

box-sizing

Example

```
width: 300px;
border: 1px solid black;
padding: 5px;
box-sizing: border-box;

/* Firefox */
-moz-box-sizing: border-box;
/* WebKit */
-webkit-box-sizing: border-box;
```

22

22

box-sizing

Example

```
<!DOCTYPE html>
<html>
<head>
<style>


```



<https://codepen.io/Vn-T-i-the-styleful/pen/MWLjgeN?editors=1000>

23

23

Chuột Không Dây Logitech M187 -
Hàng chính hãng

259.000 đ -19%
~~319.000 đ~~

6. Positioning

24

Positioning

- **position:** defines the positioning of the element in the page content flow
- The **value** is one of:
 - **static** (default): Elements render in order, as they appear in the document flow
 - **relative** – relative position according to where the element would appear with static position
 - **absolute** – relative to the first parent element that has a position other than static
 - **fixed** – relative to the browser window, but ignores page scrolling
 - **sticky** – The element is positioned based on the user's scroll position

25

25

sticky Example

```
<!DOCTYPE html>
<html>
<head>
<style>
div.sticky {
  position: -webkit-sticky;
  position: sticky;
  top: 0;
  padding: 5px;
  background-color: #cae8ca;
  border: 2px solid #4CAF50;
}
</style>
</head>
```

Try to **scroll** inside this frame to understand how sticky positioning works.

Note: IE/Edge 15 and earlier versions do not support sticky position.

I am sticky!

In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.

I am sticky!

In this example, the sticky element sticks to the top of the page (top: 0), when you reach its scroll position.

Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, malisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et.

https://www.w3schools.com/css/tryit.asp?filename=trycss_position_sticky

26

26

Positioning

- **Fixed** and **absolutely** positioned elements do **not influence** the page normal flow and usually stay on top of other elements
 - Their **position** and **size** are **ignored** when calculating the size of parent element or position of surrounding elements
 - Overlaid according to their **z-index**
 - Inline fixed or absolutely positioned elements can apply height like block-level elements

27

27

Demo

```
<!DOCTYPE html>
<html>
<head>
<style>
.container {
  position: relative;
  background: lightgray;
  width: 50%;
  margin: 0 auto;
  height: 1000px;
}
.container p {
  text-align: center;
  font-size: 20px;
}
.box-orange {
  background: orange;
  width: 100px;
  height: 100px;
  position: fixed;
  right: 5px;
}
.box-blue {
  background: lightblue;
  width: 100px;
  height: 100px;
}
</style>
```

```
<body>
  <div class="container">
    <div class="box-orange"></div>
    <div class="box-blue"></div>
    <p>Scroll down the page</p>
  </div>
</body>
</html>
```

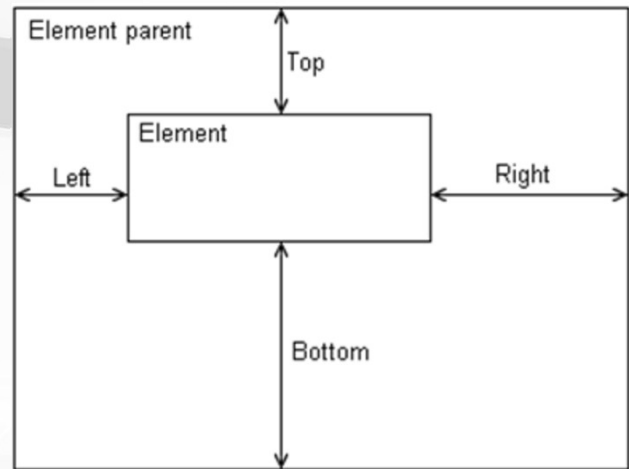
<https://codepen.io/Vn-T-i-the-styleful/pen/yLZaxEy?editors=1010>

28

28

Positioning

top, left, bottom, right: specifies **offset** of **absolute/fixed/relative** positioned element as numerical values



29

29

How to center an absolute block

Horizontal align

```
.center-absolute-block {
  position: absolute;
  right: 0;
  left: 0;
  margin-left: auto;
  margin-right: auto;
}
```

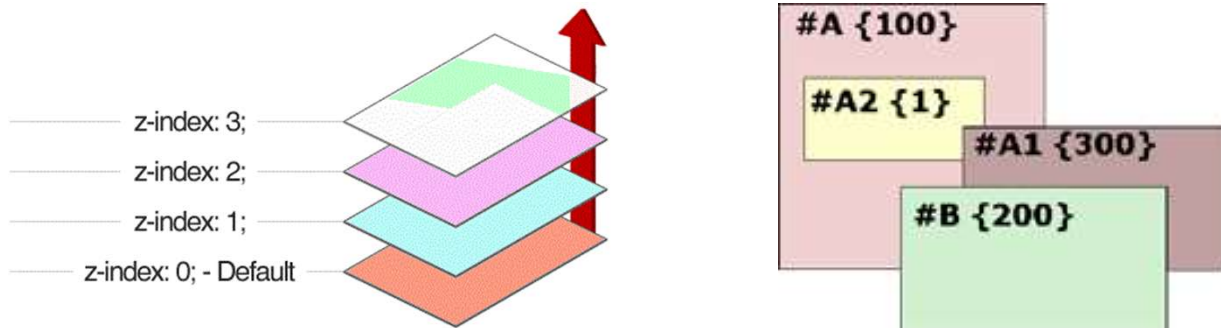
30

30

Positioning

z-index: specifies the stack level of positioned elements

Understanding stacking context



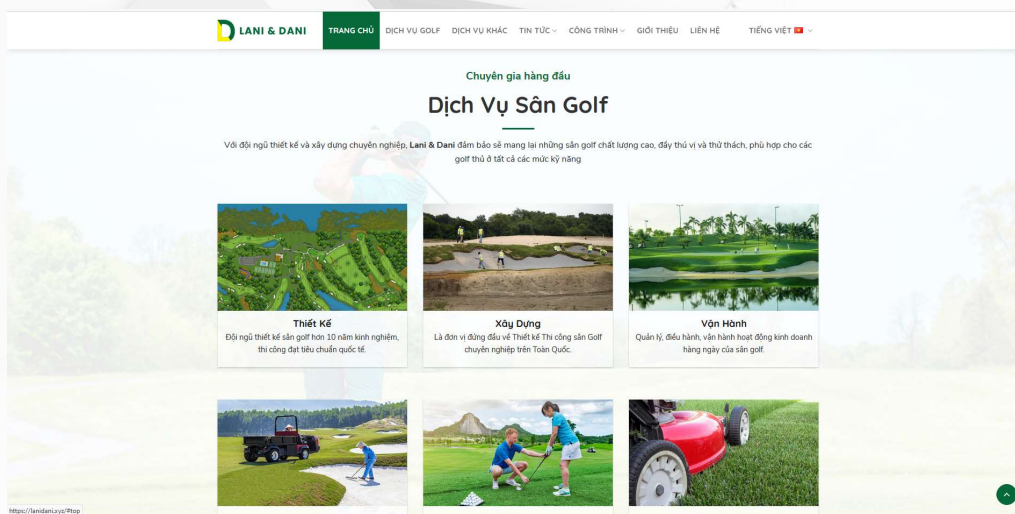
https://www.w3schools.com/cssref/tryit.php?filename=trycss_zindex

31

31

Exercise

Create a “Go to top” button

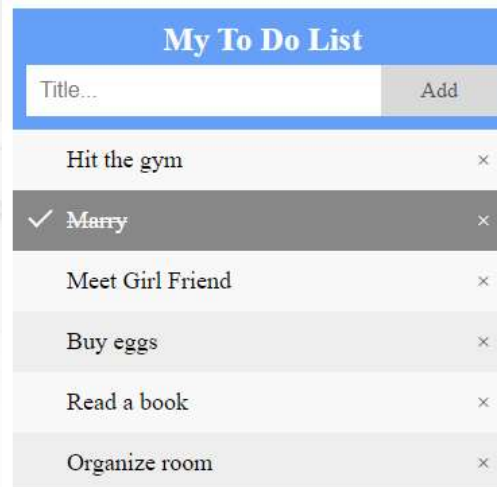


32

32

Exercise

Create a To-do List like the following.



My To Do List	
Title...	Add
Hit the gym	x
✓ Marry	x
Meet Girl Friend	x
Buy eggs	x
Read a book	x
Organize room	x

33

33

Inline element positioning

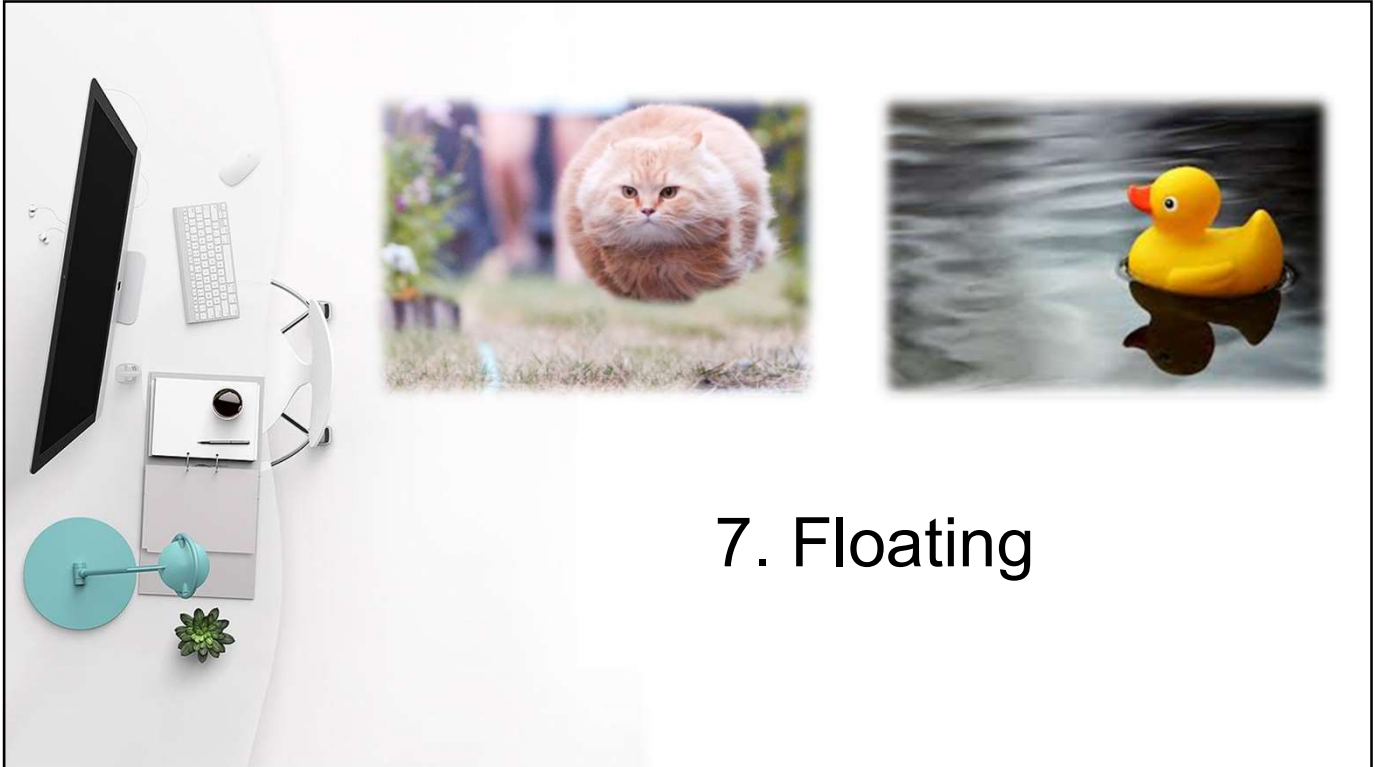
vertical-align: sets the vertical-alignment of an inline, inline-block element, **according to the line height**

- **Values:** baseline, sub, super, top, text-top, middle, bottom, text-bottom or numeric
- Also used for content of **table cells** (which apply **middle** alignment by default)

https://www.w3schools.com/cssref/tryit.php?filename=trycss_vertical-align

34

34



7. Floating

35

Float

float: the element “floats” to one side

- **left:** places the element on the left and following content on the right
- **right:** places the element on the right and following content on the left
- **Floated elements should come before the content** that will wrap around them in the code
- **Margins** of floated elements do **not collapse**
- Floated inline elements can apply height

36

36

CSS – Float

Menu

[Trang chủ](#)
[Giới thiệu](#)
[Tin tức](#)
[Bảo hành](#)
[Liên hệ](#)

Ordered and Unordered List: list-style-type values

none	No marker is shown
disc	Default value. The marker is a filled circle
square	The marker is a square
circle	The marker is a circle
decimal	The marker is a number

https://www.w3schools.com/cssref/pr_list-style-type.asp

37

37

Float

How floated elements are positioned?

```
<style>
img { float: left; }
p.clear { clear: both; }
</style>

<p>This is some text...</p>
<p>This is also some text...</p>
<p>...Remove the "clear" class...</p>
```



This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.

This is also some text. This is also some text. This is also some text. This is also some text. This is also some text. This is also some text.

Remove the "clear" class to see the effect.



This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.

This is also some text. This is also some text. This is also some text. This is also some text. This is also some text. This is also some text.

Remove the "clear" class to see the effect.

38

38

Clear

- **clear**
 - Sets the sides of the element where other floating elements are NOT allowed
 - Used to **"drop" elements below floated ones** or expand a container, which contains only floated children
 - **Values:** left, right, both
- Clearing floats
 - **Clear using pseudo-class :after**
 - Additional element (<div>) with a clear style
Deprecated - semantically unused div
- Link:
 - <https://css-tricks.com/all-about-floats/>
 - IE: positioniseverything.net/easyclearing.html

39

39

Float

How floated elements are positioned?

```
<style>
  img { float: left; }
  p.clear { clear: both; }
</style>

<p>This is some text...</p>
<p>This is also some text...</p>
<p>...Remove the "clear" class...</p>
```



This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.

This is also some text. This is also some text. This is also some text. This is also some text. This is also some text. This is also some text.

Remove the "clear" class to see the effect.

```
<style>
  img { float: left; }
  p.clear { clear: both; }
</style>

<p class="clear">This is some text...</p>
<p>This is also some text...</p>
<p>...Remove the "clear" class...</p>
```



This is some text. This is some text. This is some text. This is some text. This is some text. This is some text.

This is also some text. This is also some text. This is also some text. This is also some text. This is also some text. This is also some text.

Remove the "clear" class to see the effect.

40

40

Clear

New modern Clearfix

Put the following class to a float container (a <div> or <section> covers all floated elements)

```
.clearfix::after {
  content: "";
  clear: both;
  display: block;
}
```

41

41

CSS – Layout

Exercise

EIU-Eastern International University

[Home](#)
[About](#)
[Contact](#)

Menu

Ngành nghề đào tạo

Tin tuyển sinh

Tin tức

Các thông tin chung

Cuộc thi “Lập trình xe đua tự động EIU MCR” lần hai chính thức khởi động!

Thời gian: 08:00, 29/10/2023 Địa điểm: EIU EIU MCR trở lại, hứa hẹn sẽ là đường đua khoa học, công nghệ: Trí tuệ Bô ích Kịch tính Hấp dẫn Cuộc thi không chỉ thúc đẩy phong trào học tập – nghiên cứu cho học sinh, sinh viên mà còn giúp các bạn tiếp cận với kỹ thuật, công nghệ và kiến thức lập trình, điều khiển tự động; nâng cao kỹ năng lập trình, khả năng chế tạo kỹ thuật, phát huy khả năng sáng tạo, khả năng làm việc nhóm v.v.. EIU MCR lần một đã hút sự tham gia của hơn 40 đội thi đến từ các trường đại học, cao đẳng, trường THPT khu vực phía Nam. Hứa hẹn lần hai sẽ bùng nổ với những đường đua căng thẳng và hấp dẫn hơn nữa! Nhanh tay đăng ký tham gia cuộc thi để vừa học vừa chơi vừa nhận được nhiều phần thưởng giá trị bạn nhé!

Copyright 2023

Đường Nam Kỳ Khởi Nghĩa, phường Hòa Phú, thành phố Thủ Dầu Một, tỉnh Bình Dương. (+84) 0274 222 0372

42

42



43

Flexbox

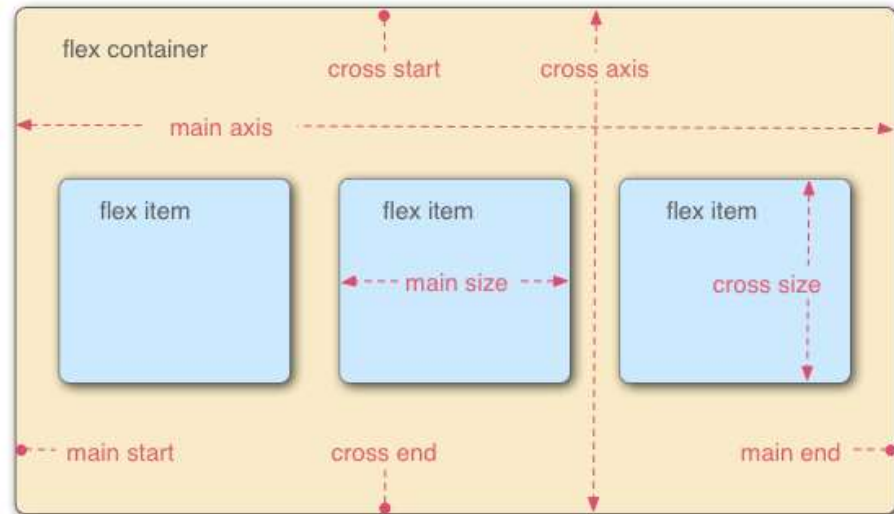
- Flexbox Layout
 - Layout mode for the **arrangement of elements** on a page
 - The elements behave predictably on **different screen sizes** and different display devices
- Browser compatibility
[Compatibility table](#)
- Complete guide
[Guide](#)

44

44

Flexbox vocabulary

- Flex container
- Flex item
- Axes
 - Main axis
 - Cross axis
- Directions
- Lines
- Dimensions



45

45

Parent properties

- **display** - enables flex for all children


```
.container {
  display: flex;
}
```
- **flex-direction** - establishes the main-axis

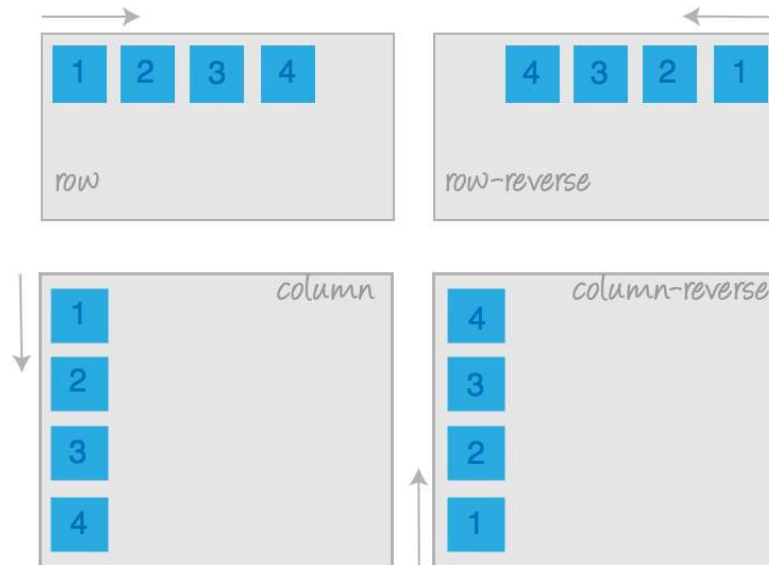

```
.container {
  flex-direction:
    row | row-reverse | column | column-reverse;
}
```
- **flex-wrap** - flex items will all try to fit onto one line by default


```
.container {
  flex-wrap: nowrap | wrap | wrap-reverse;
}
```

46

46

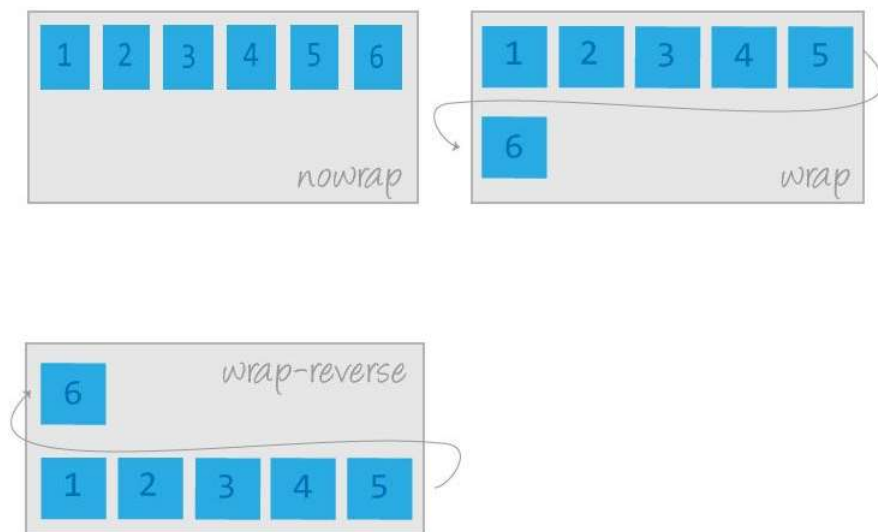
Flex-direction



47

47

Flex-wrap



48

48

Parent properties

- **flex-flow** - shorthand for flex-direction and flex-wrap

```
.container {
  flex-flow: <'flex-direction'> <'flex-wrap'>
}
```

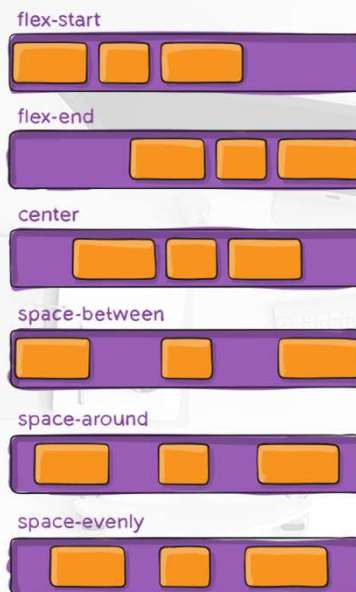
- **justify-content** - align the items on the main axis

```
.container {
  justify-content: flex-start | flex-end
                  | center | space-between | space-around;
}
```

49

49

Justify-content



50

50

Parent properties

- **align-items** - align the items on the cross axis

```
.container {
  align-items: flex-start | flex-end |
    center | baseline | stretch;
}
```

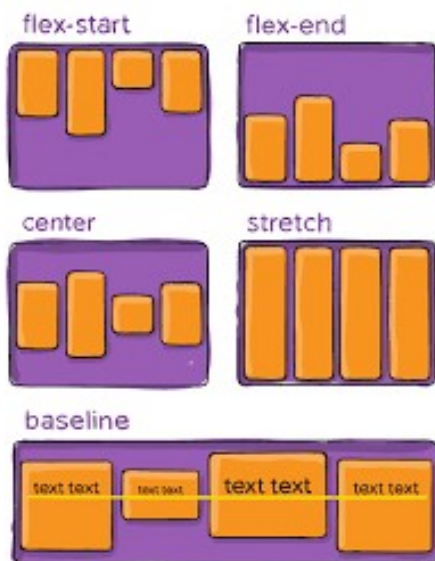
- **align-content** - flex container's lines within when there is extra space in the cross-axis

```
.container {
  align-content: flex-start | flex-end | center |
    space-between | space-around | stretch;
}
```

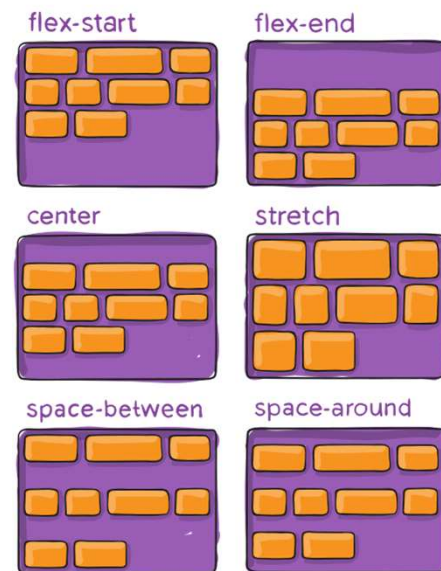
51

51

Align-items



Align-content



52

52

Children properties

- **order** - controls the order in which the children appear in the flex container

```
.item {
  order: <integer>;
}
```

- **flex-grow** - defines the ability for a flex item to grow if necessary

```
.item {
  flex-grow: <number>; /* default 0 */
}
```

53

53

Order

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

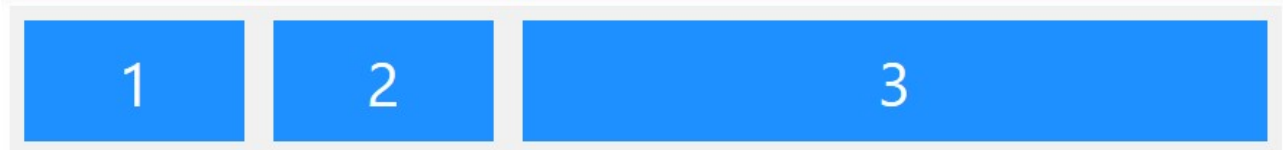


54

54

flex-grow

```
<div class="flex-container">
  <div style="flex-grow: 1">1</div>
  <div style="flex-grow: 1">2</div>
  <div style="flex-grow: 8">3</div>
</div>
```



55

55

Children properties

- **flex-shrink** - defines the ability for a flex item to shrink if necessary

```
.item {
  flex-shrink: <number>; /* default 1 */
}
```

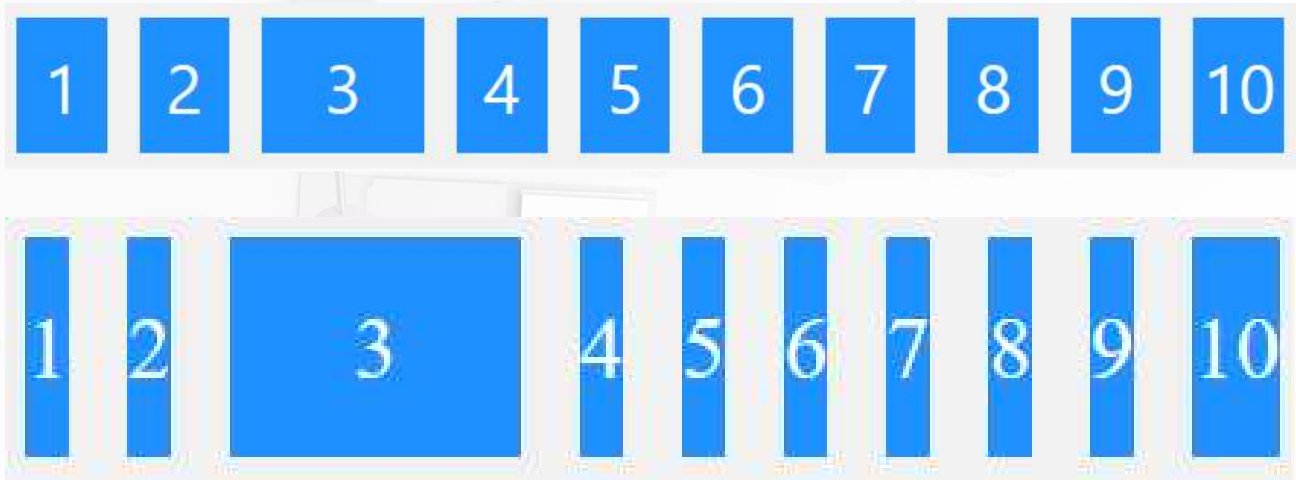
- **flex-basis** - defines the default size of an element before the remaining space is distributed

```
.item {
  flex-basis: <length> | auto; /* default auto */
}
```

56

56

Flex-shrink



57

57

Children properties

- **flex** - shorthand for **flex-grow**, **flex-shrink** and **flex-basis** combined (**recommended**)

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

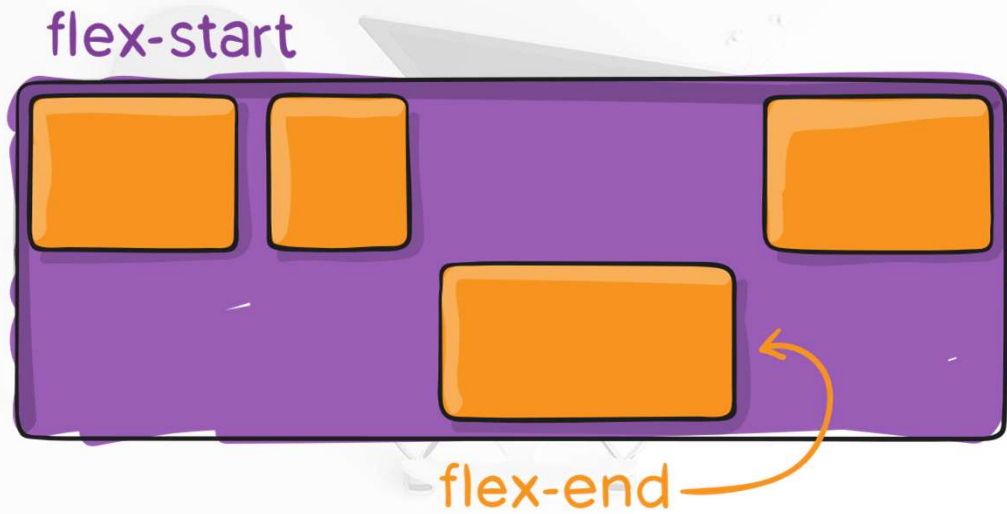
- **align-self** - allows the default alignment (or the one specified by align-items) to be overridden for individual flex items

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

58

58

Align-self



59

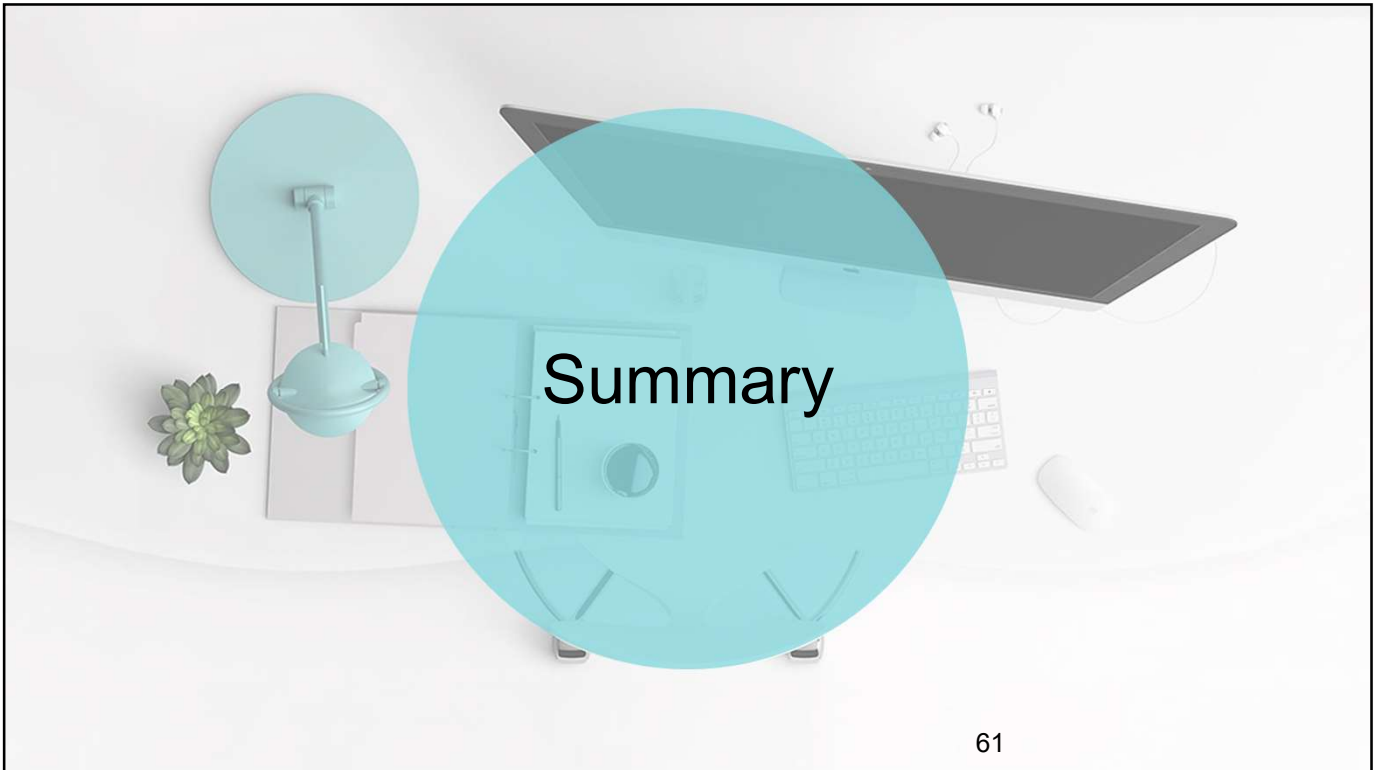
59

Example

<https://codepen.io/Vn-T-i-the-styleful/pen/WNPGgKq?editors=1000>

60

60

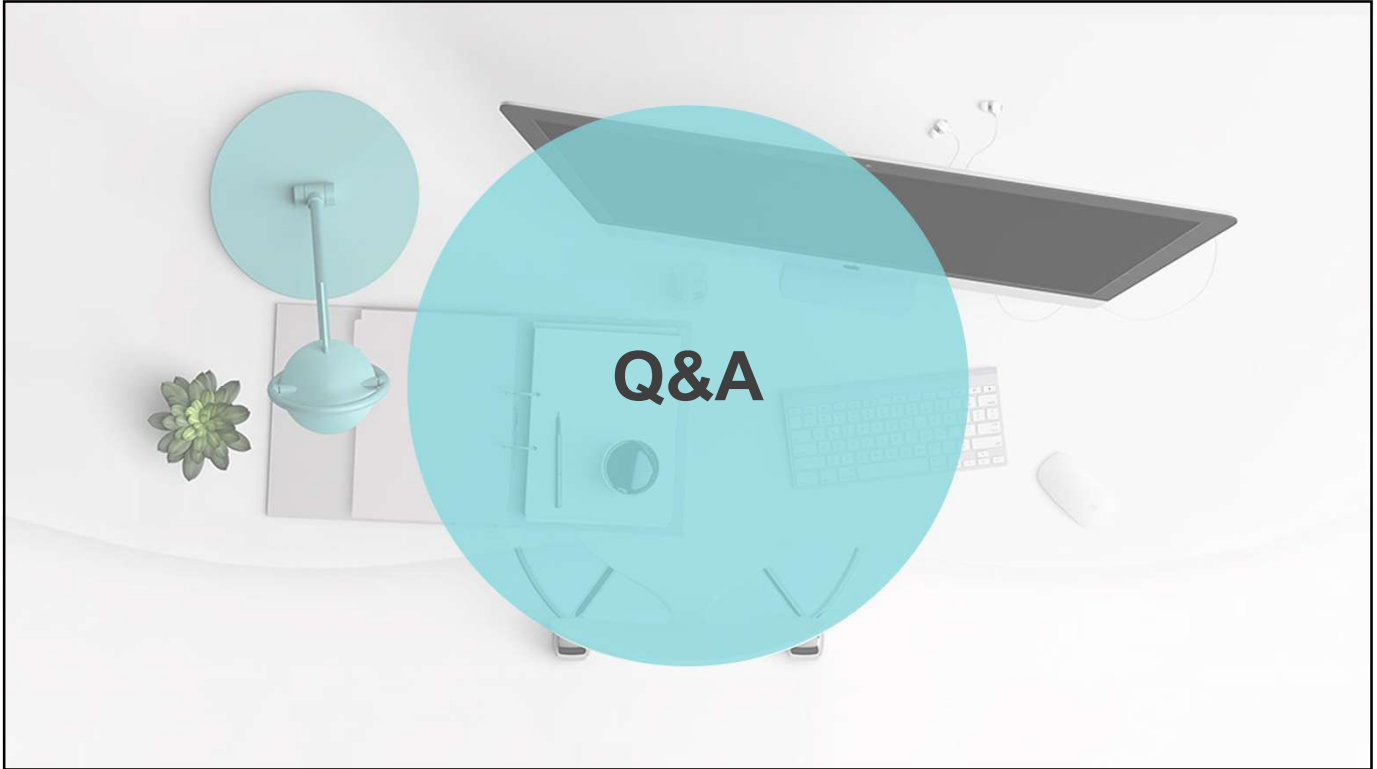


61



Complete the Lani & Dani webpage (only Desktop UI).

62



63



64

Visibility

- **Visibility** determines whether the element is visible
- **Values:**
 - **hidden:** element is not rendered, but still occupies place on the page
similar to **opacity: 0**
 - **visible:** element is rendered normally
 - **collapse:** collapse removes a row or column, but it does not affect the table layout
 - ✓ only for table elements
 - ✓ The space taken up by the row or column will be available for other content

65