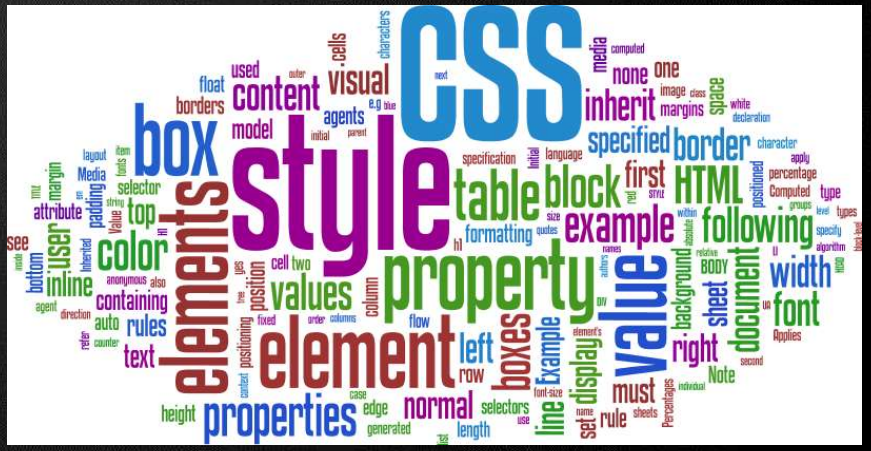
Telerik Academy

CSS Layout

Control the arrangement of the HTML elements



CSS Styling

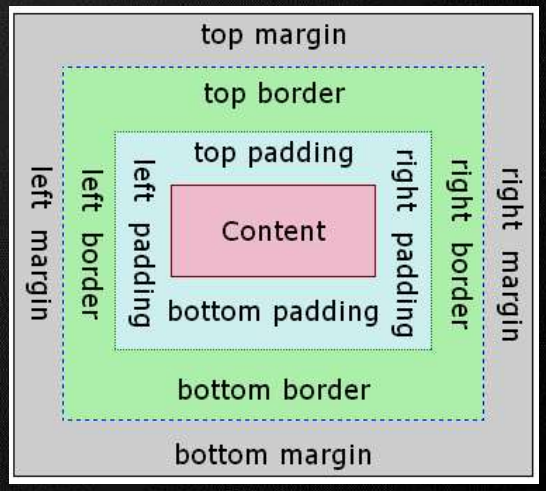
Telerik Software Academy

https://telerikacademy.com

Follow us

Telerik Academy

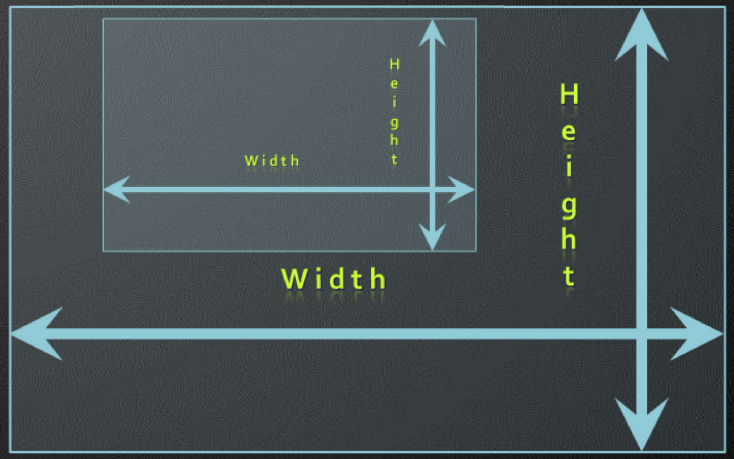
Table of Contents



Follow us

Telerik Academy

Width and Height

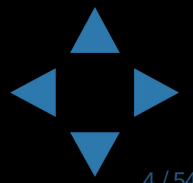


Follow us

Telerik Academy

Width

- The width property defines numerical value for the width of element, e.g. `200px`
- Applies only for block elements
  - Their with is `100%` by default
  - The width of inline elements is always the width of their content, by concept
- min-width
  - defines the minimal width
  - overrides width if width < min-width
- max-width
  - defines the maximal width
  - overrides width if width > max-width

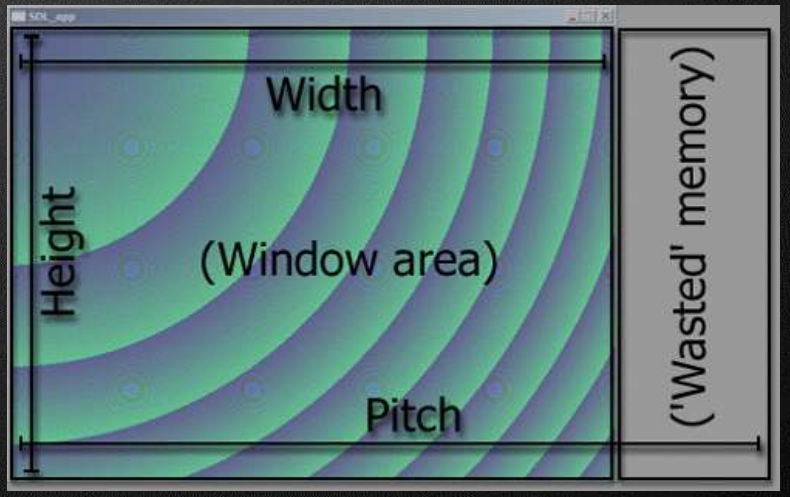


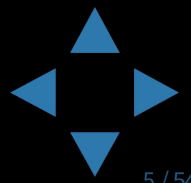
Follow us

Telerik Academy

Width

Demo



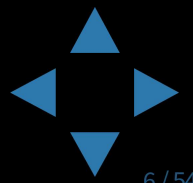


Follow us

Telerik Academy

Height

- The height property defines numerical value for the height of element, e.g. `100px`
- Applies only on block elements
  - The height of inline elements is always the height of their content
- min-height
  - defines the minimal height
  - overrides height if height < min-height
- max-height
  - defines the maximal height
  - overrides height if height < min-height



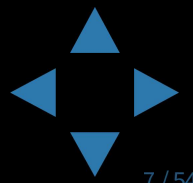
Follow us

Telerik Academy

Width and Height Values

- The values of the width and height properties are numerical:
  - Pixels (px)
  - Centimeters (cm)
  - Ems (em)
  - Or percentages
    - A percent of the available width

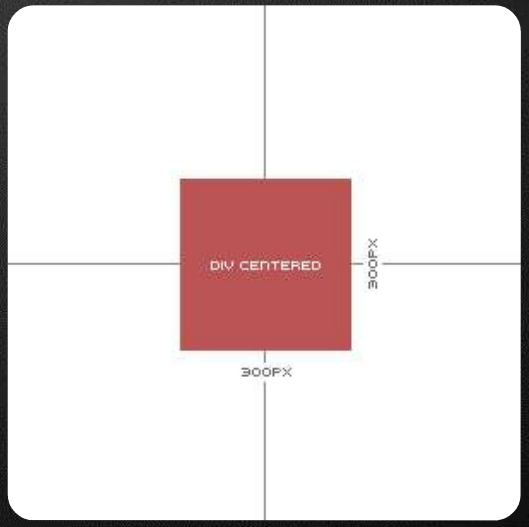
Follow us



Telerik Academy

Height

Demo



Follow us

Telerik Academy

Overflow

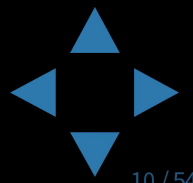
Follow us

Telerik Academy

Overflow

- The overflow property defines the behavior of element when content needs more space than the available
- 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

Follow us



Telerik Academy

Overflow

Demo

Follow us

Telerik Academy

Display

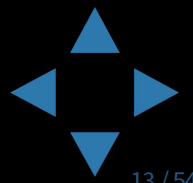
Follow us

Telerik Academy

Display

- The display property controls the display of the element and the way it is rendered and if breaks should be placed before and after the element
- Values:
  - `inline`: no breaks are placed before or after (`<span>` is an inline element)
    - height and width depend on the content
  - `block`: breaks are placed before AND after the element (`<div>` is a block element)
    - height and width may not depend on the size of the content

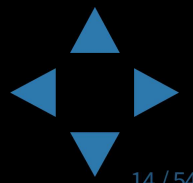
Follow us



Telerik Academy

Display

- More 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



Follow us

Telerik Academy

Display

Demo

Follow us

Telerik Academy

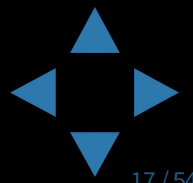
Visibility

Follow us

Telerik Academy

Visibility

- The visibility property
  - Determines whether the element is visible
  - `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



Follow us

Telerik Academy

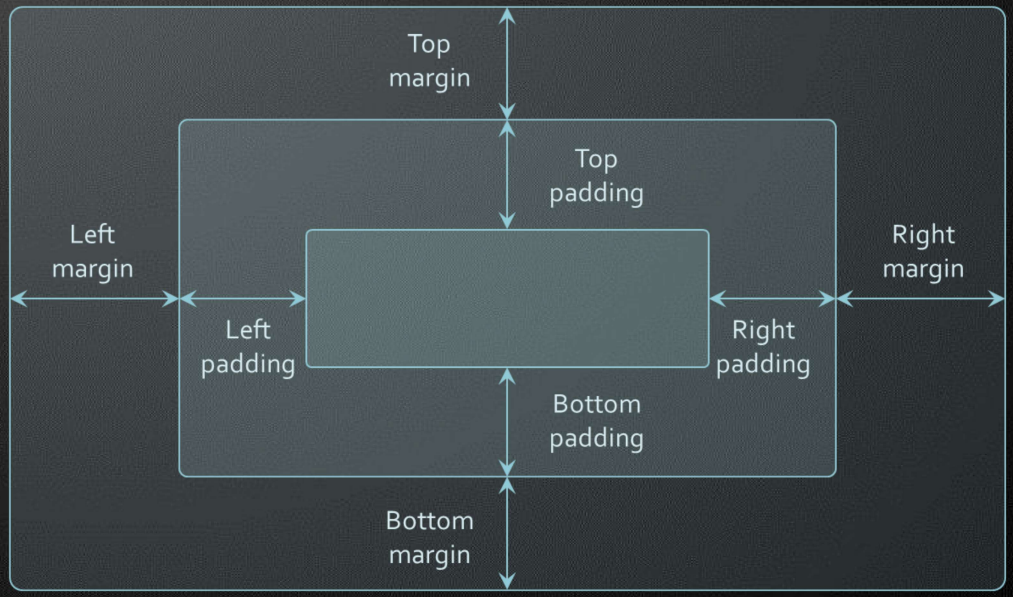
Visibility

Demo

Follow us

Telerik Academy

Margins and Paddings

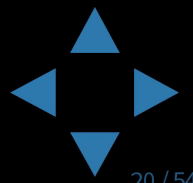


Follow us

Telerik Academy

Margin and Padding

- The margin and padding properties define the spacing around the element
  - They take 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



Follow us

Telerik Academy

Margin and Padding: Short Rules

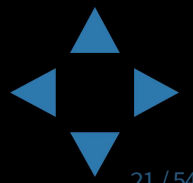
- Sets all four sides to have margin of 5px

```
.container {
    margin: 5px;
}
```

- top and bottom to 10px, left and right to 20px

```
.container {
    margin: 10px 20px;
}
```

Follow us



Telerik Academy

Margin and Padding: Short Rules

- top 5px, left and right 3px, bottom 8px

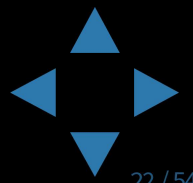
```
.container {
    margin: 5px 3px 8px;
}
```

- top, right, bottom, left (clockwise from top)

```
.container {
    margin: 1px 3px 5px 7px;
}
```

- Same for padding

Follow us

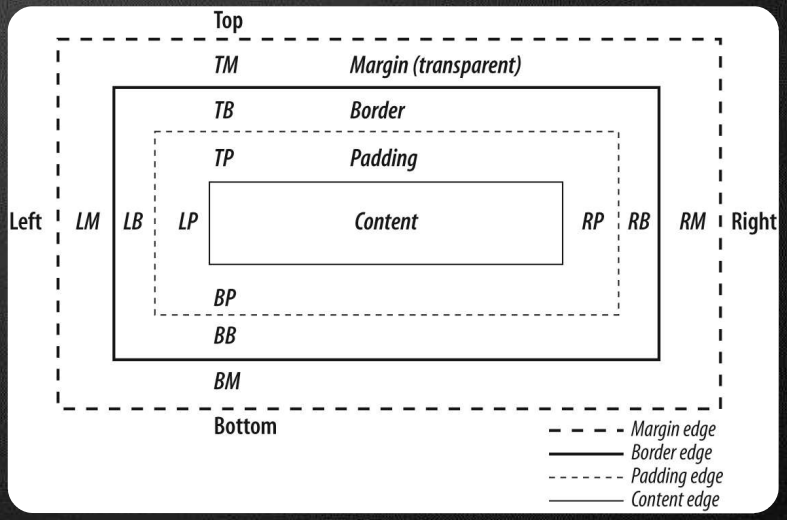


Margins and Paddings

Demo

Telerik Academy

Box Model



Follow us

Telerik Academy

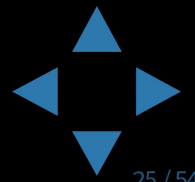
CSS3 box-sizing

- Determine whether you want an element to render it's borders and padding within its specified width, or outside of it.
- Possible values:
- Content-box

```
.container {
    width: 300px;
    box-sizing: content-box;
}
```

  - box width: 288px + 10px padding + 1px border on each side = 300px

Follow us



Telerik Academy

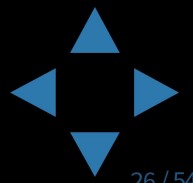
CSS3 box-sizing

- Border-box

```
.container {
    width: 300px;
    box-sizing: border-box;
}
```

  - box width: 300px, including padding and borders

Follow us



Telerik Academy

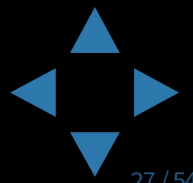
CSS3 box-sizing (*Example*)

- *Example*: Box with total width of 300px (including paddings and borders)

```
.container {
    width: 300px;
    border: 1px solid black;
    padding: 5px;

    /* Firefox */
    -moz-box-sizing: border-box;
    /* WebKit */
    -webkit-box-sizing: border-box;
    /* Opera 9.5+, Google Chrome */
    box-sizing: border-box;
}
```

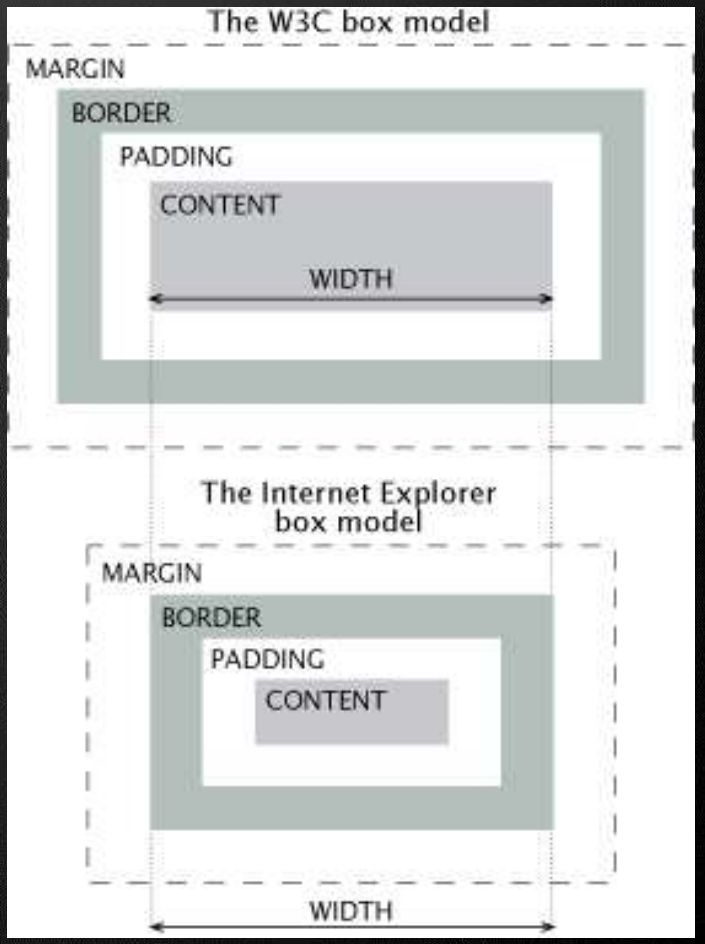
Follow us

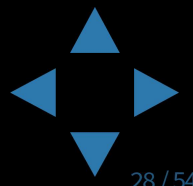


Telerik Academy

IE Quirks Mode

- When using quirks mode (pages with no DOCTYPE or with a HTML 4 Transitional DOCTYPE)
  - Internet Explorer violates the box model standard!





Follow us

Telerik Academy

Box Model

Demo

Follow us

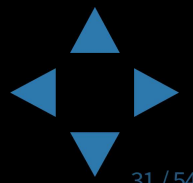
Telerik Academy

Positioning

Telerik Academy

Positioning

- The position property defines the positioning of the element in the page content flow
- The value is one of:
  - `static` is the default value
  - `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



Follow us

Telerik Academy

Positioning

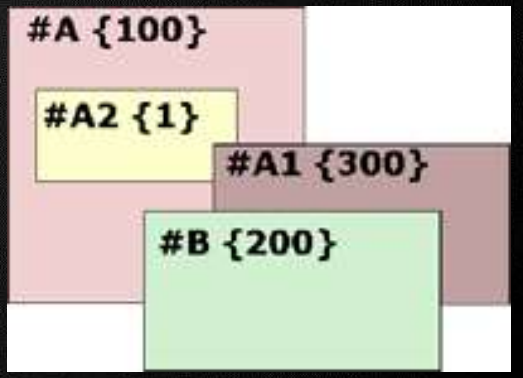
- Margin VS relative positioning
- Fixed and absolutely positioned elements do not influence the page normal flow and usually stay on top of other elements
  - Their position and size are ignored when calculating the size of parent element or position of surrounding elements
  - Overlaid according to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

Follow us

Telerik Academy

Positioning

- `top`, `left`, `bottom`, `right`: specifies offset of absolute/fixed/relative positioned element as numerical values
- `z-index` : specifies the stack level of positioned elements
  - Understanding stacking context

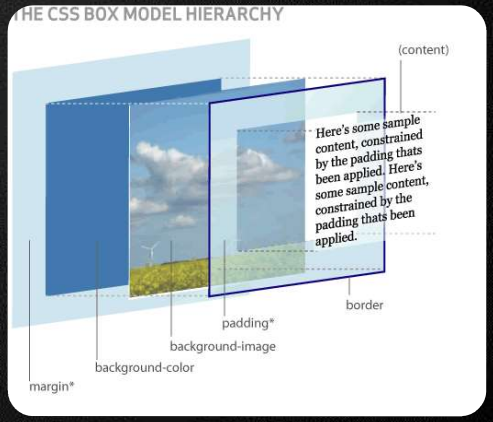


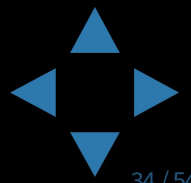
Follow us

Telerik Academy

Positioning

Demo



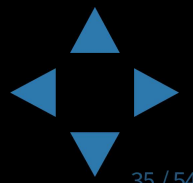


Follow us

Telerik Academy

Inline element positioning

- The vertical-align property sets the vertical alignment of an inline 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)

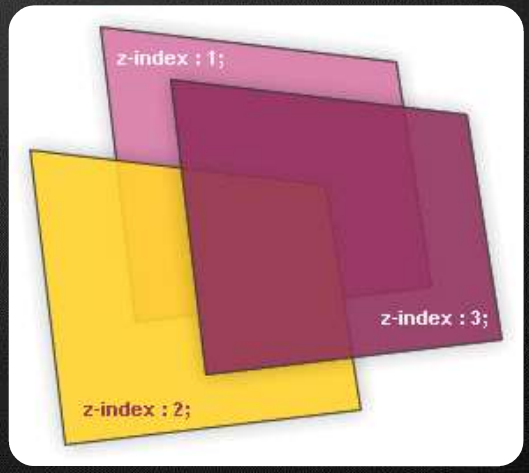


Follow us

Telerik Academy

Alignment and Z-Index

Demo




```
img
{
    position:absolute;
    left:0px;
    top:0px;
    z-index:-1;
}
```

Follow us

Telerik Academy

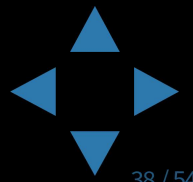
Floating

Follow us

Telerik Academy

Floating

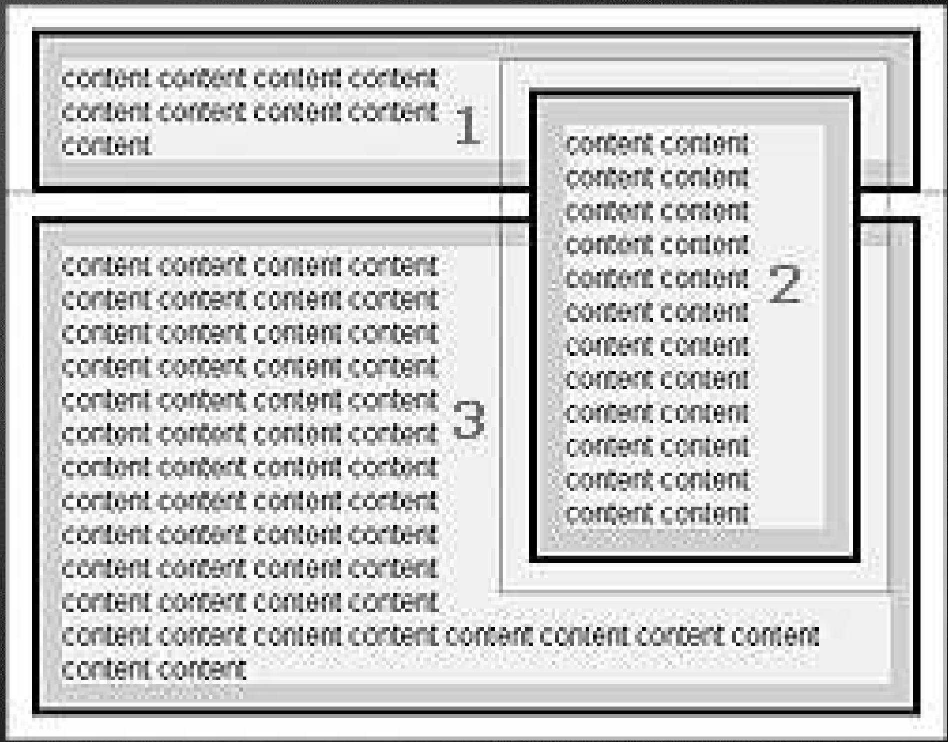
- The float property makes the element “float” to one side
  - `left`: places the element on the left and following content on the right
  - `right`: places the element on the right and following content on the left
  - floated elements should come before the content that will wrap around them in the code
  - margins of floated elements do not collapse
  - floated inline elements can apply height



Follow us

Float

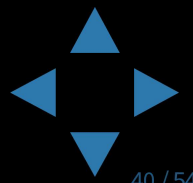
- How floated elements are positioned



Telerik Academy

Clear

- What does clear do?
  - 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`



Follow us

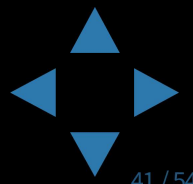
Telerik Academy

Clear

- Clearing floats
  - Clear using pseudo-class `:after`
  - Snippet after IE8

```
.group:after {
content: "";
display: table;
clear: both;
}
```

  - Additional element (`<div>`) with a clear style
    - Deprecated - semantically unused div
  - Additional info about floats



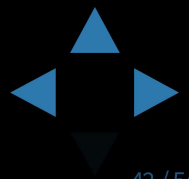
Follow us

Telerik Academy

Floating elements

Demo

Follow us



Telerik Academy

Flexbox

The Next Generation of CSS Layout

Telerik Academy

Flexbox

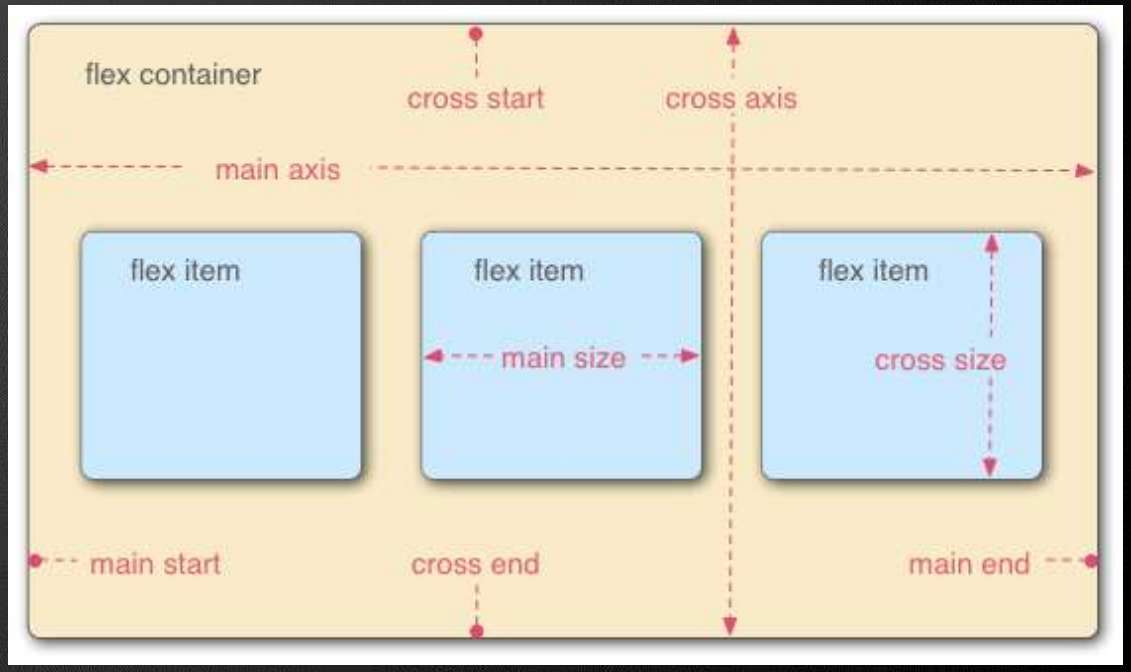
- Flexbox Layout
  - Layout mode for the arrangement of elements on a page
  - The elements behave predictably on different screen sizes and different display devices
- Browser compatibility
  - compatibility table
- Complete guide
  - guide

Follow us

Telerik Academy

Flexbox vocabulary

- Flex container
- Flex item
- Axes
- Directions
- Lines
- Dimensions



Follow us

Telerik Academy

Parent properties

- display - enables flex for all children

```
.container {
  display: flex; /* or inline-flex */
}
```

- flex-direction - establishes the main-axis

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

Follow us

Telerik Academy

Parent properties

- flex-wrap - flex items will all try to fit onto one line by default

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

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

```
.container {
  flex-flow: <‘flex-direction’> || <‘flex-wrap’>
}
```

Follow us

Telerik Academy

Parent properties

- justify-content - defines the alignment along the main axis

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

Follow us

Telerik Academy

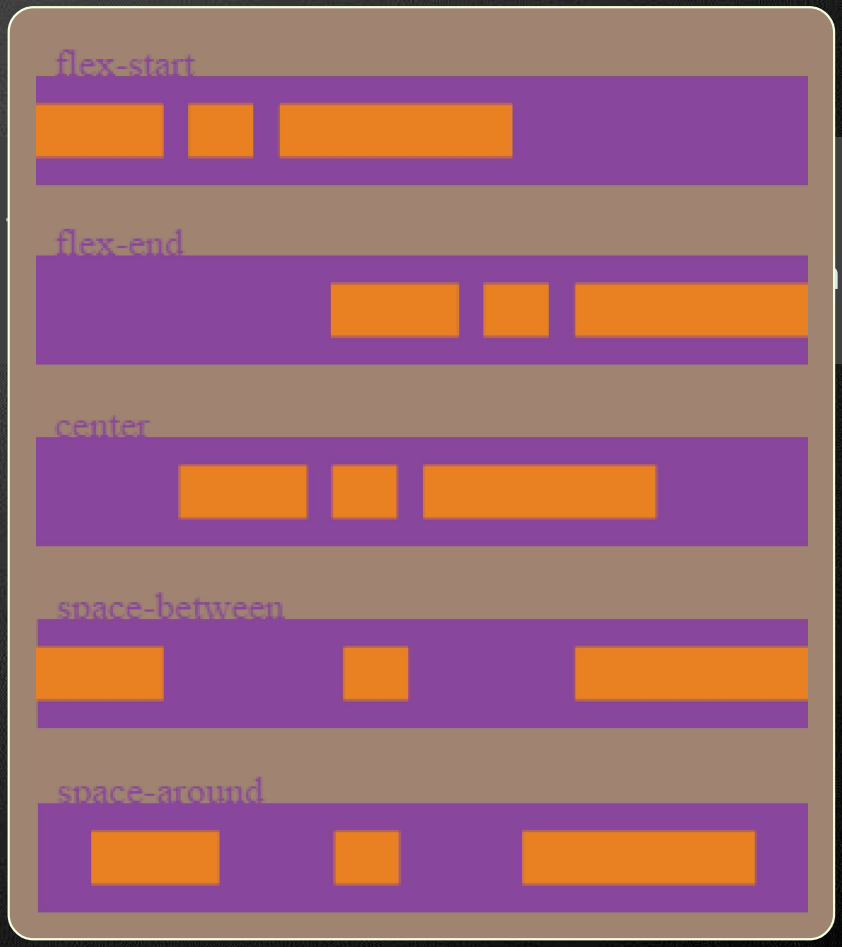
Parent properties

- justify-content - defines the alignment along the main axis

```
.container {
  justify-content:

}
```

```
round;
```



Telerik Academy

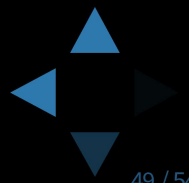
Parent properties

- align-items - justify-content version for the cross-axis

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

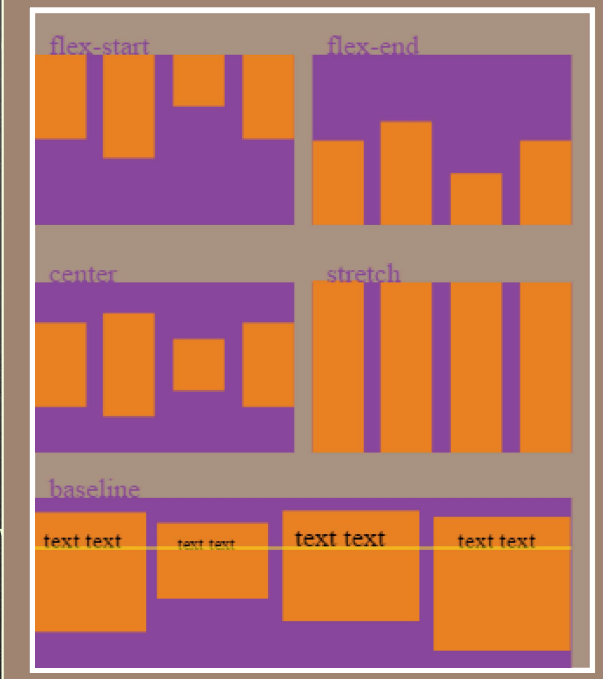
- align-content - flex container's lines within when there is extra space in the cross-axis

```
.container {
  align-content: flex-start | flex-end | center | space-betwee
}
```



Parent properties

- align-items - justify-content version for the




```
end
e | stretch;
```

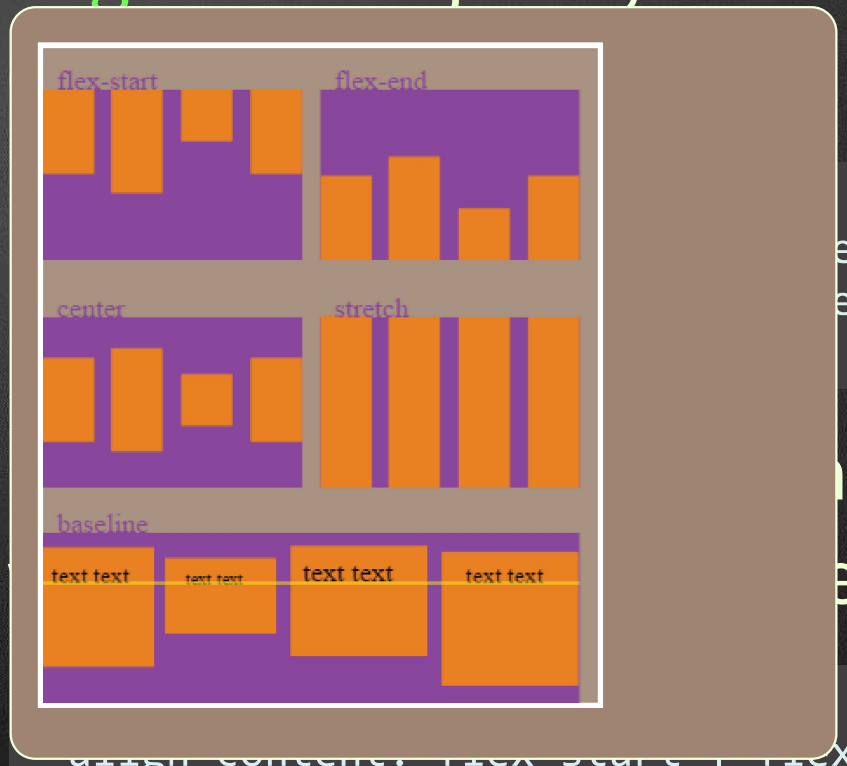
- iner's lines within
e in the cross-axis

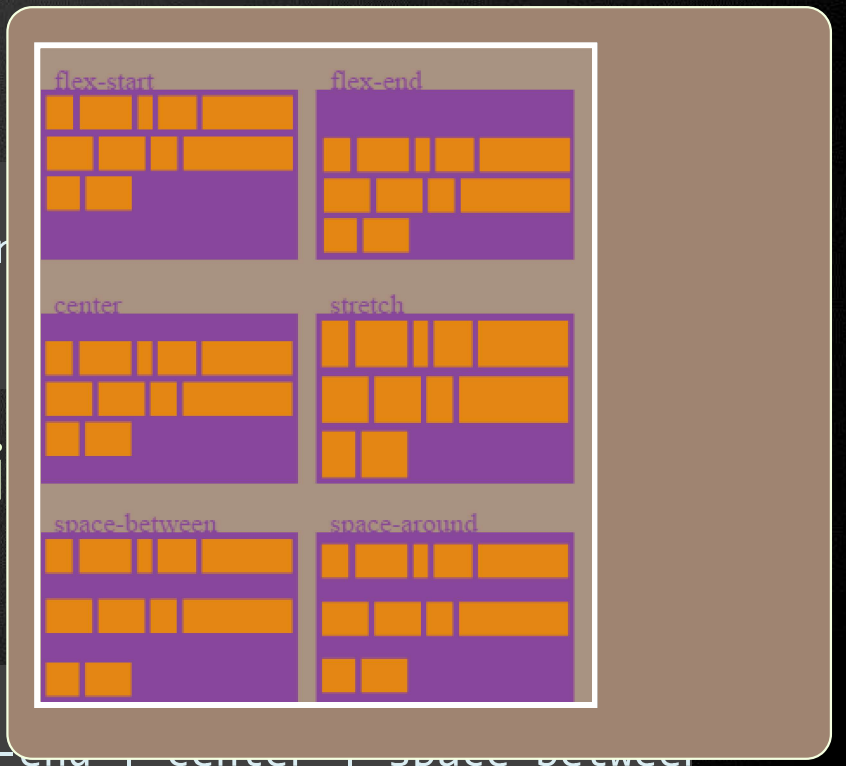
```
x-end | center | space-betwee
}
```

Telerik Academy

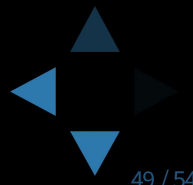
Parent properties

- align-items - justify-content version for the






```
}
```



Follow us

Telerik Academy

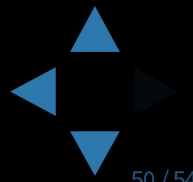
Children properties

- 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 */
}
```



Follow us

Telerik Academy

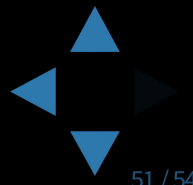
Children properties

- 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 */
}
```



Follow us

Telerik Academy

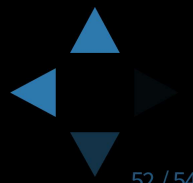
Children properties

- 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 | stre
}
```



Follow us

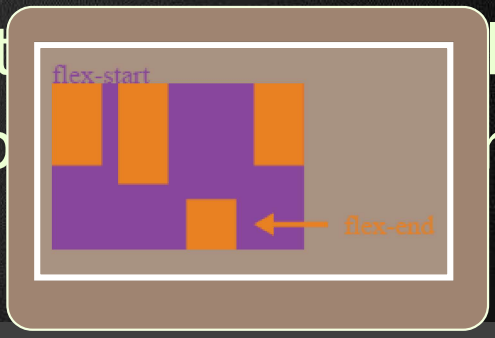
Telerik Academy

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[illegible]he one specified by align-items) to b[illegible] individual flex items




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

Follow us