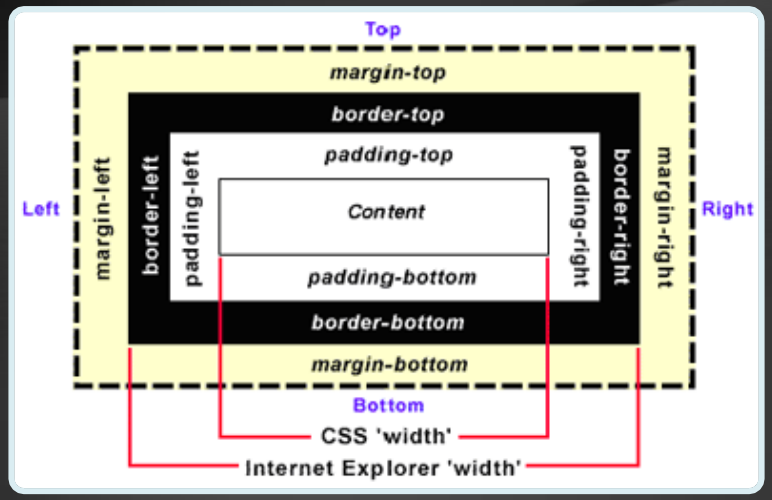
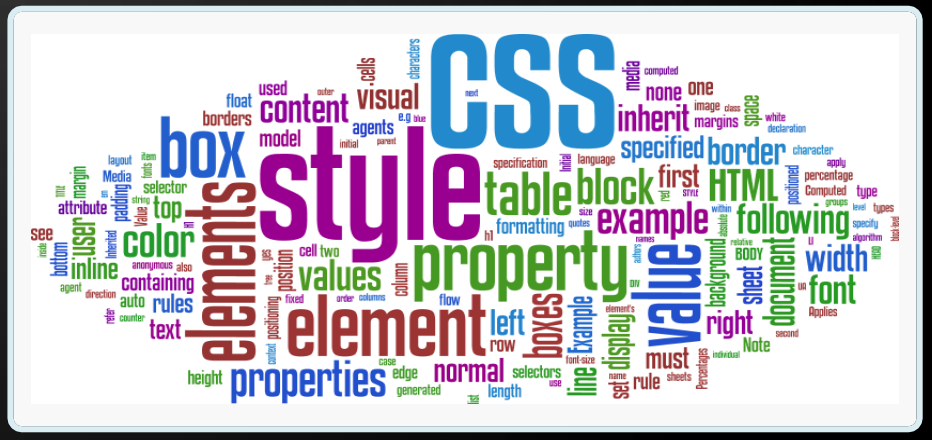
Telerik Academy



CSS Layout

Control the arrangement of the HTML elements



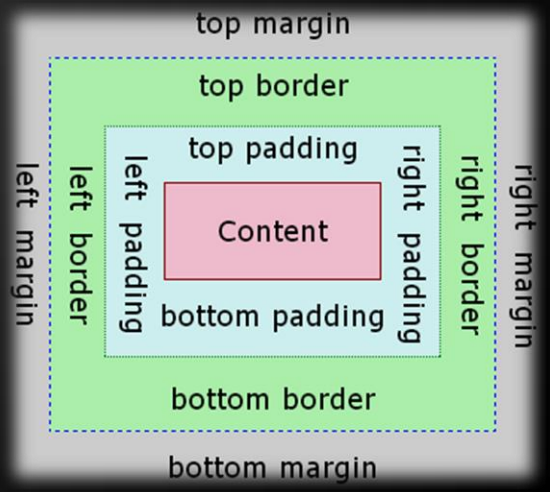
CSS

Telerik Software Academy

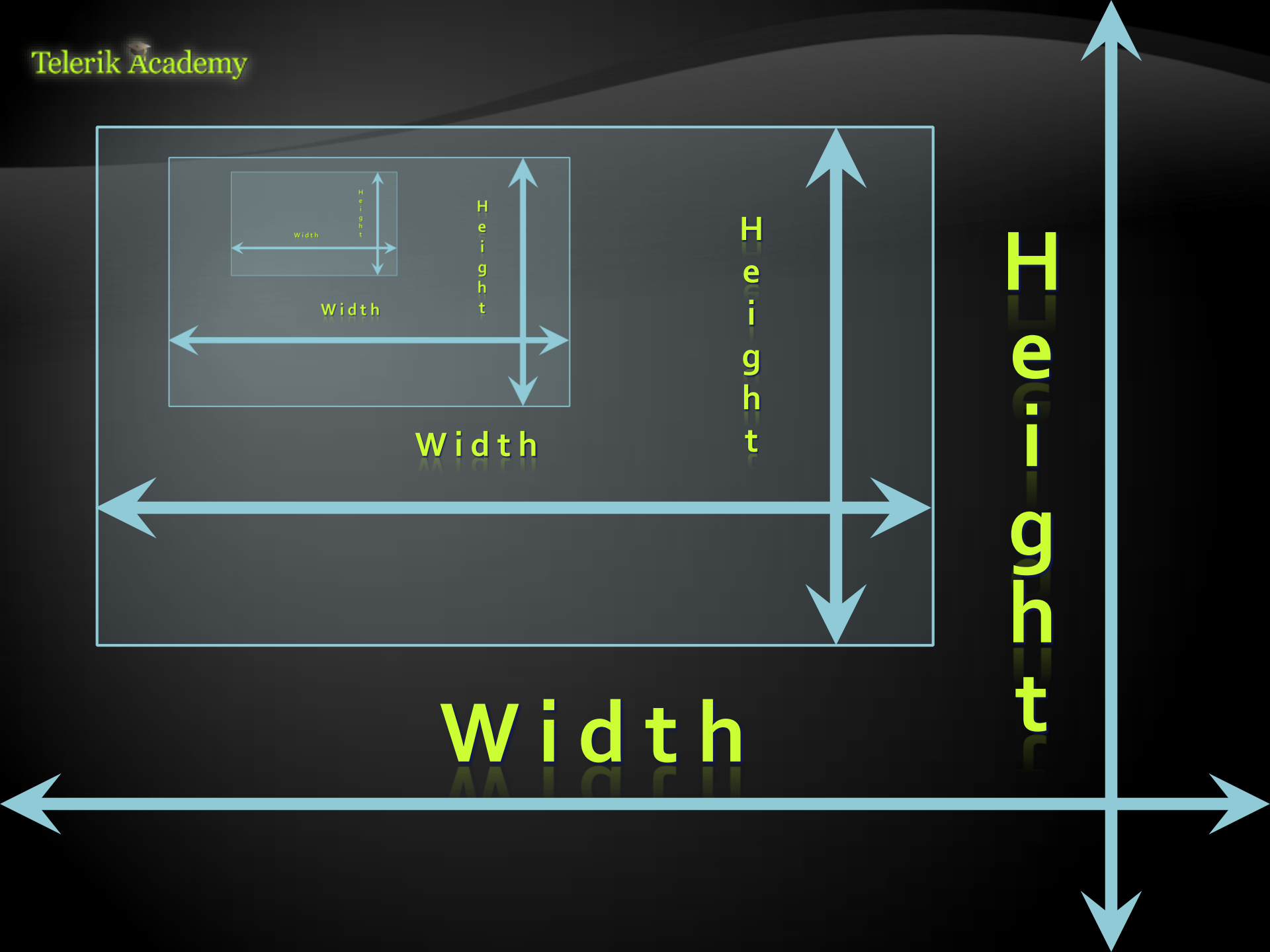
http://academy.telerik.com

Telerik Academy

Table of Contents

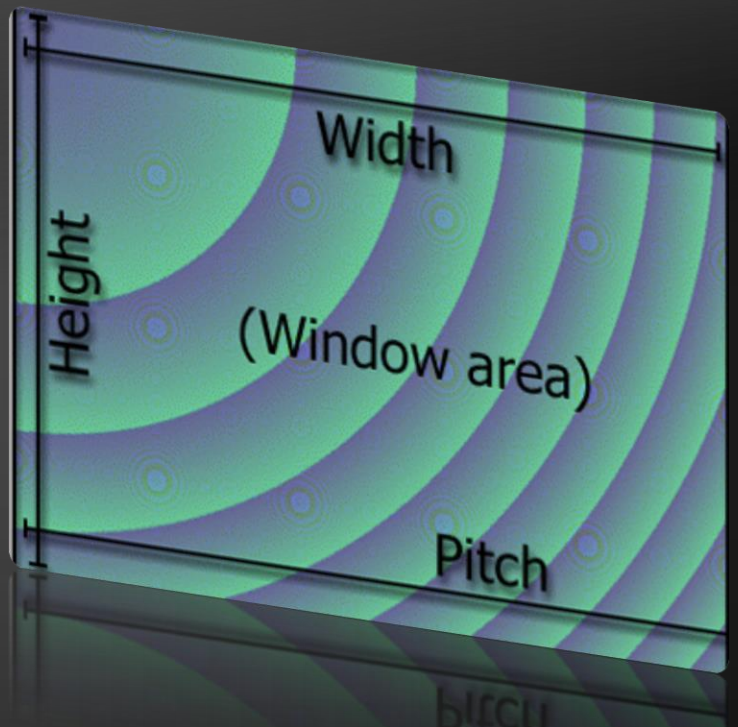


Telerik Academy



Width

- `width` – defines numerical value for the width of element, e.g. `200px`
- `width` 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
  - `min-width` overrides width if (`width<min-width`)
- `max-width` - defines the maximal width
  - `max-width` overrides width if (`width>max-width`)



Width

Live Demo

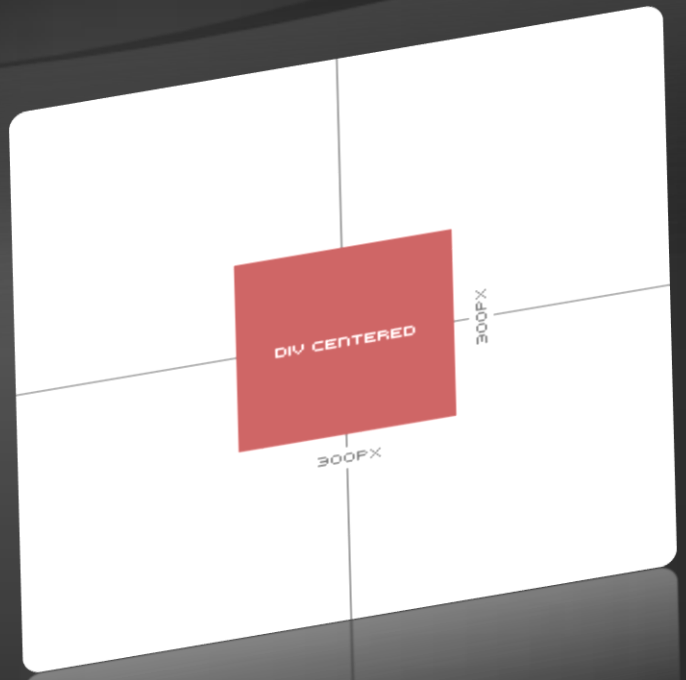
Height

- `height` – defines numerical value for the height of element, e.g. `100px`
- `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

Width and Height Values

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

Telerik Academy



Height

Live Demo

Overflow

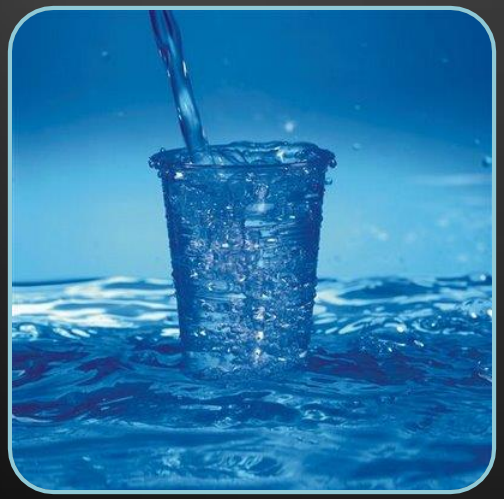
Overflow

- `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

Telerik Academy

Overflow

Live Demo



Telerik Academy

Display

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 (`<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

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

Telerik Academy

Display

Live Demo

Telerik Academy

Visibility

Visibility

- `visibility`
  - 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

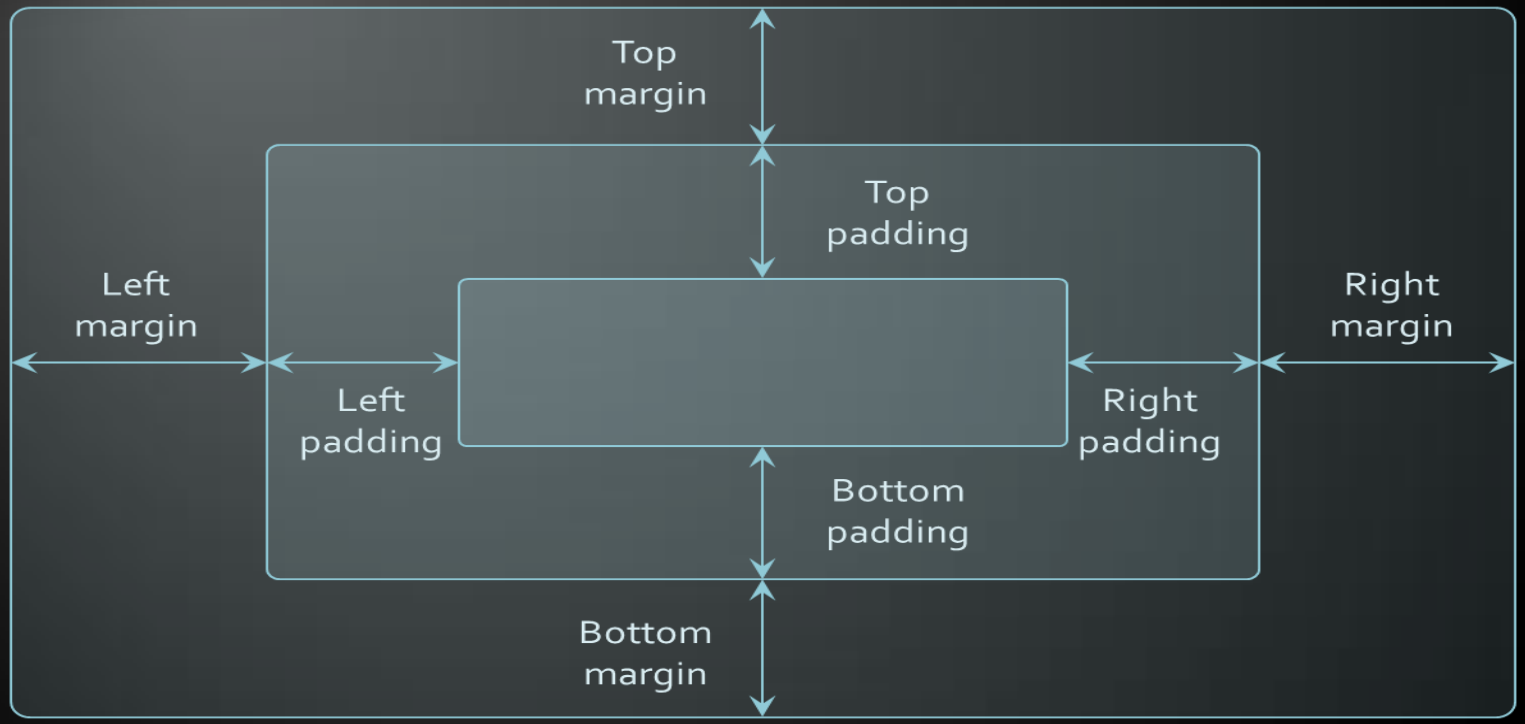
Telerik Academy

Visibility

Live Demo

Telerik Academy

Margins and Paddings



Margin and Padding

- `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

Telerik Academy

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`

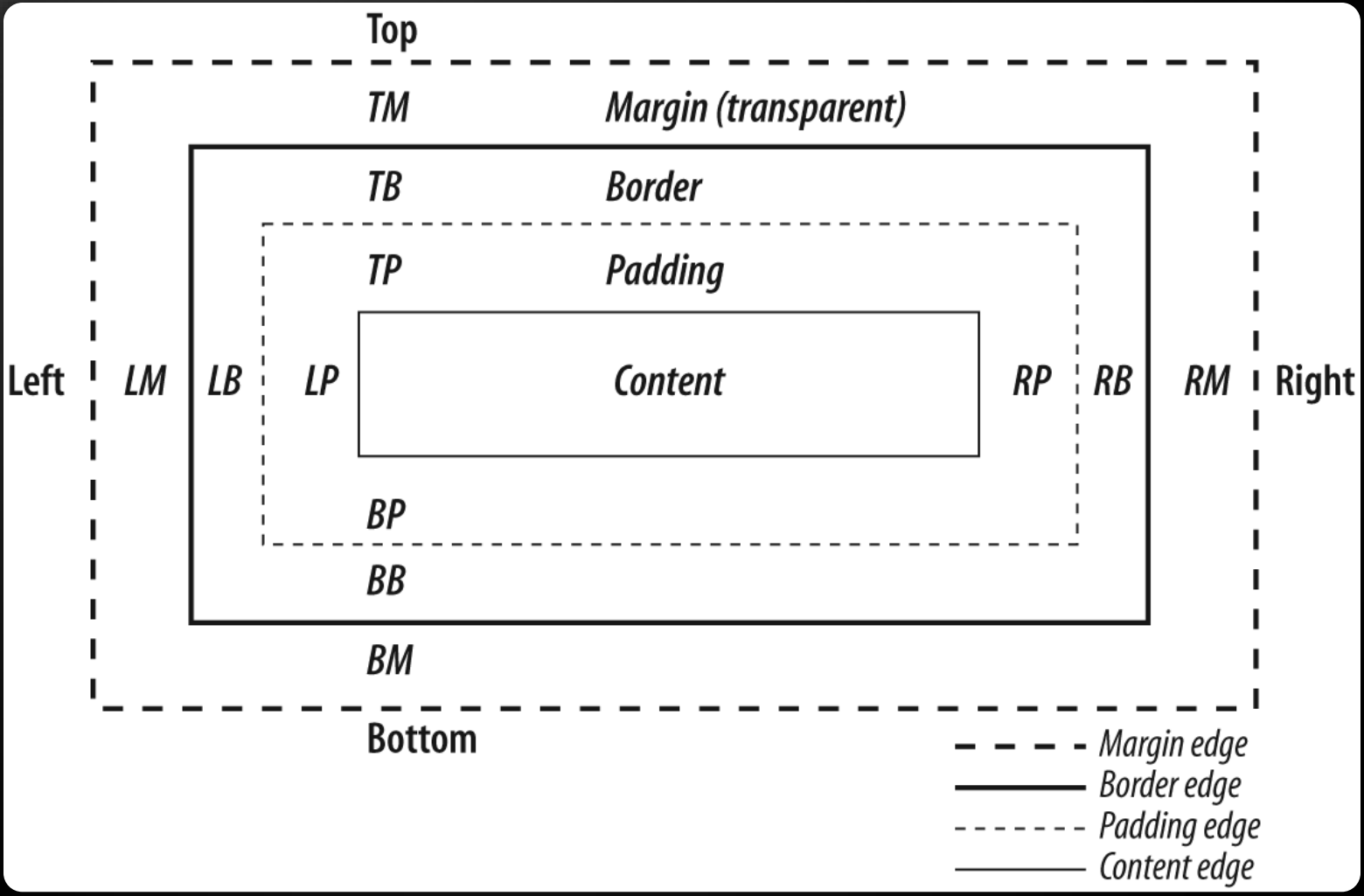
Telerik Academy

Margins and Paddings

Live Demo

Telerik Academy

Box Model



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:
  - `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

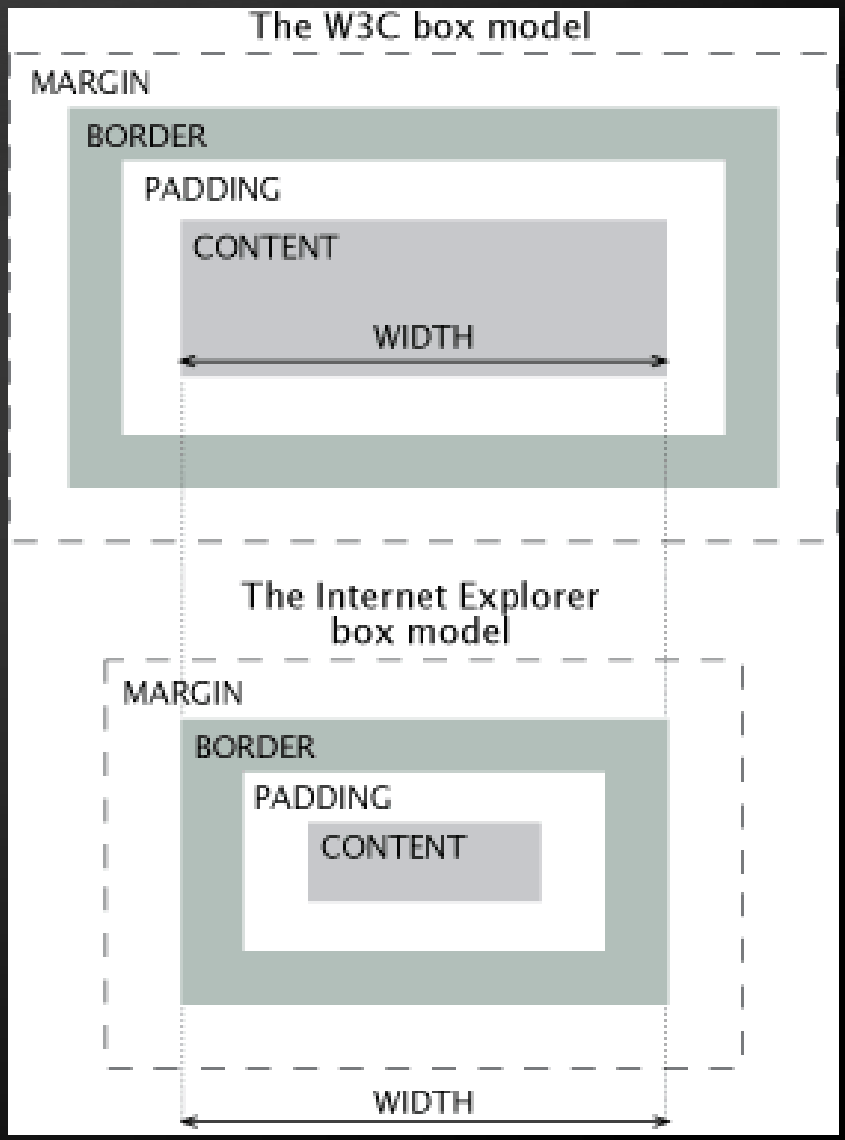
CSS3 box-sizing (Example)

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

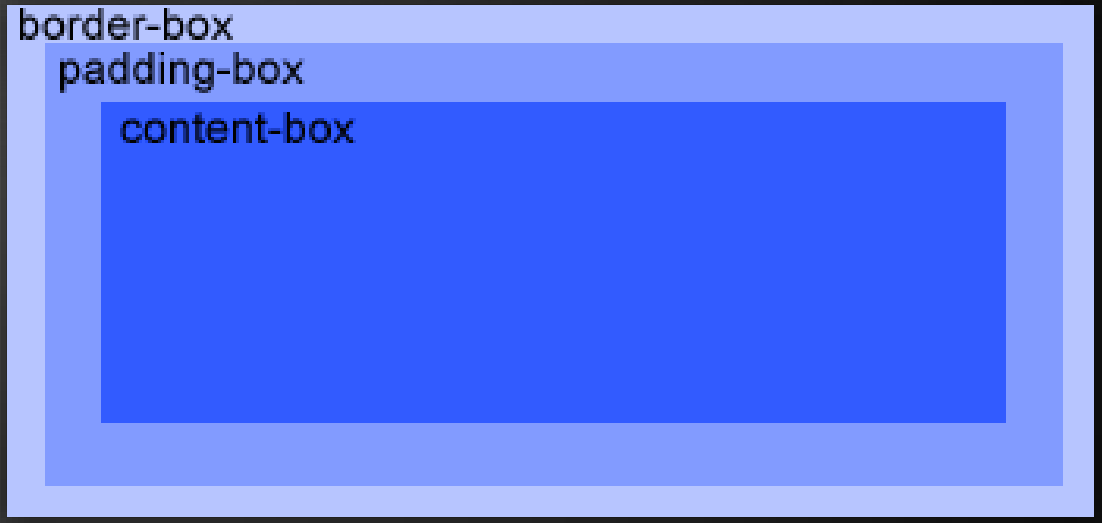
```
width: 300px;
border: 1px solid black;
padding: 5px;

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

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!



Telerik Academy



Box Model

Live Demo

Positioning

Positioning

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

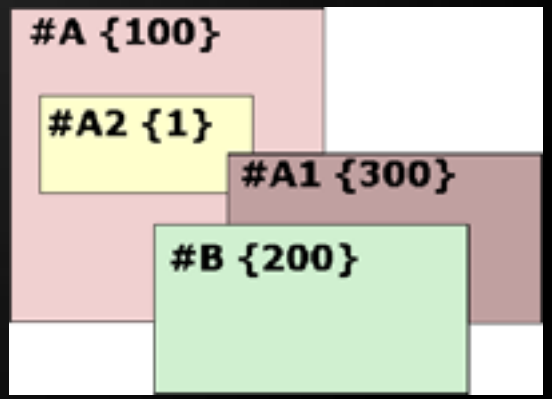
Positioning (2)

- 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

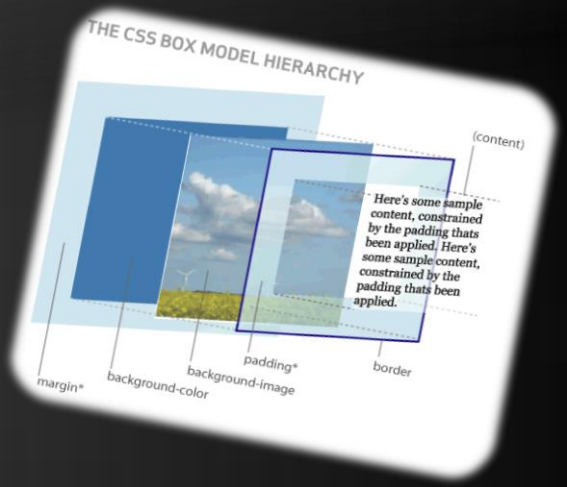
Positioning (3)

- `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

Each positioned element creates a stacking context.
Elements in different stacking contexts are overlapped according to the stacking order of their containers. For example, there is no way for #A1 and #A2 (children of #A) to be placed over #B without increasing the z-index of #A.



Telerik Academy



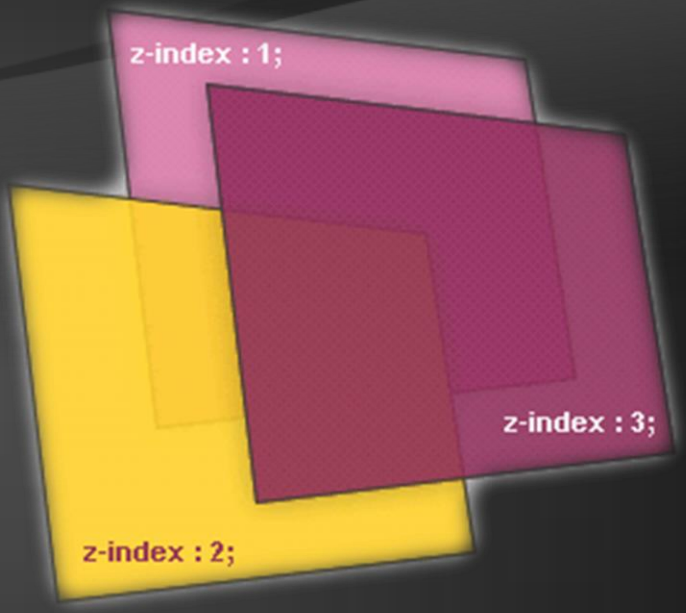
Positioning

Live Demo

Inline element positioning

- `vertical-align`: 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)

Telerik Academy




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

Alignment and Z-Index

Live Demo

Telerik Academy

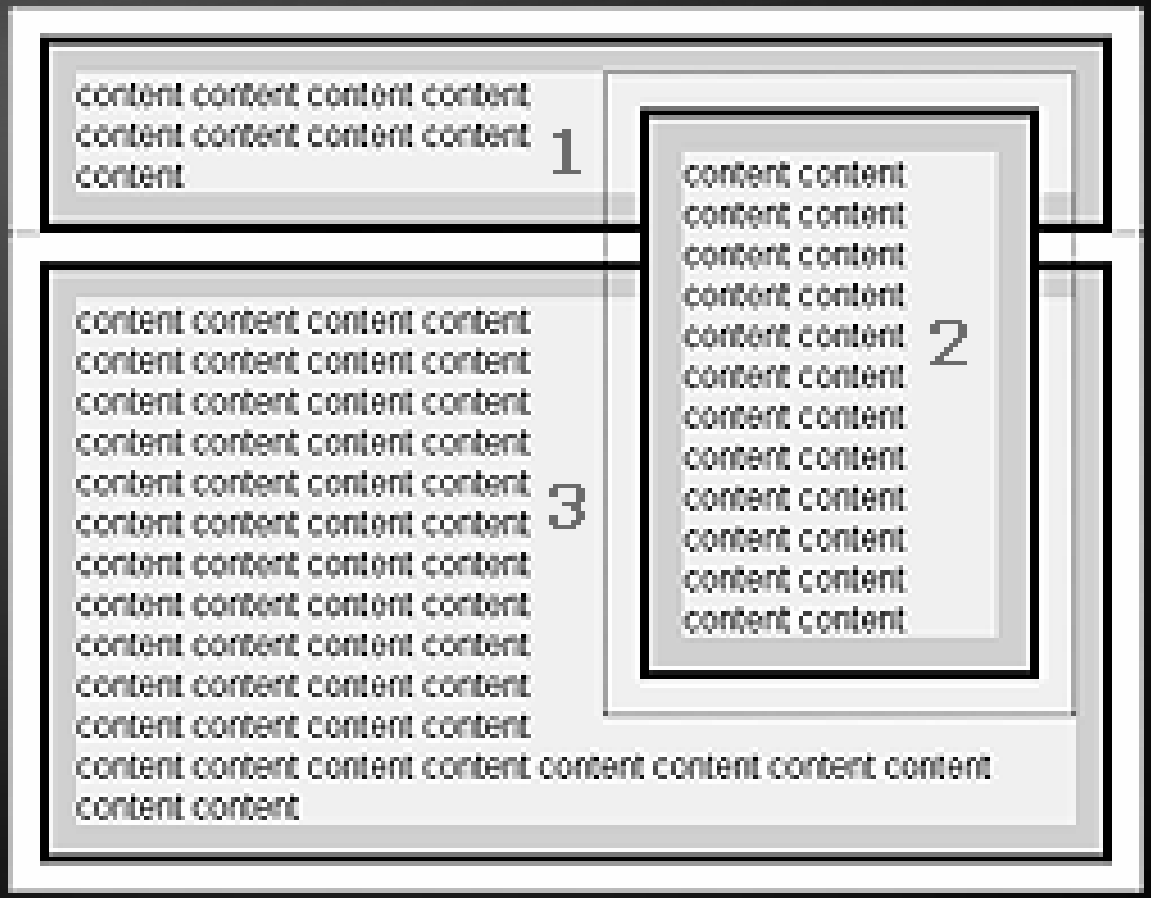
Floating

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

Float (2)

- How floated elements are positioned



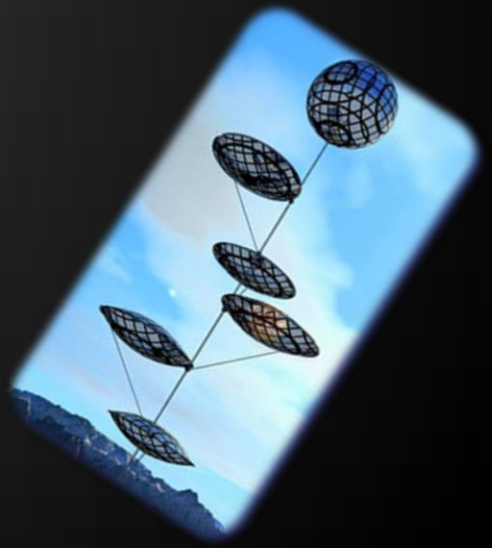
Clear

- `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
- https://css-tricks.com/all-about-floats/
- IE: positioniseverything.net/easyclearing.html

Telerik Academy

Floating Elements

Live Demo



Telerik Academy

CSS Layout

Questions?

http://academy.telerik.com

Free Trainings @ Telerik Academy

- "Web Design with HTML 5, CSS 3 and JavaScript" course @ Telerik Academy
  - html5course.telerik.com
- Telerik Software Academy
  - academy.telerik.com
- Telerik Academy @ Facebook
  - facebook.com/TelerikAcademy
- Telerik Software Academy Forums
  - forums.academy.telerik.com