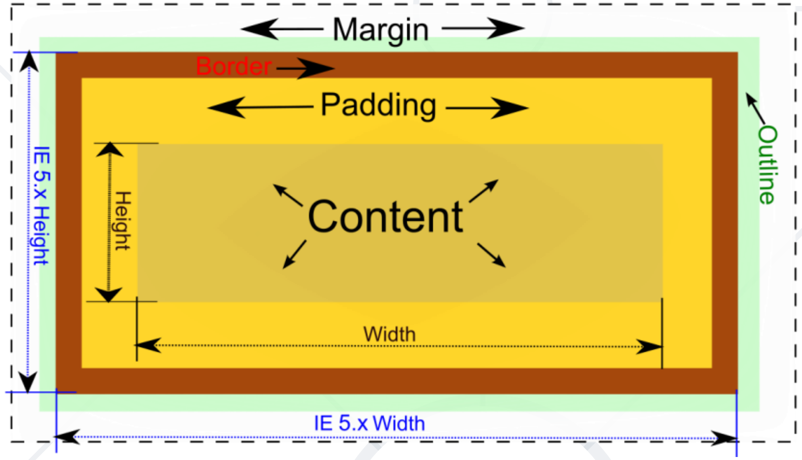
CSS Box Model

Margin, Border, Paddings, Actual Content



SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Table of Contents

- CSS Box Model
- Block and Inline Elements
- Width and Height
- Padding, Margin and Border
- Box Sizing

Have a Question?

Software University

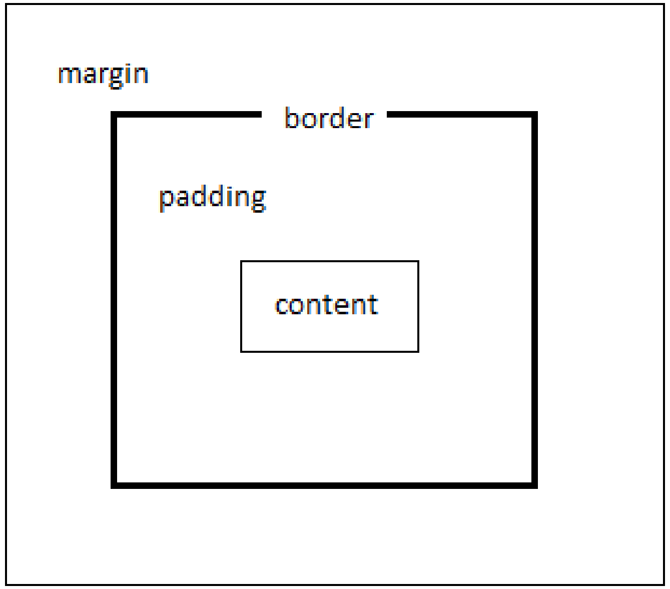
sli.do

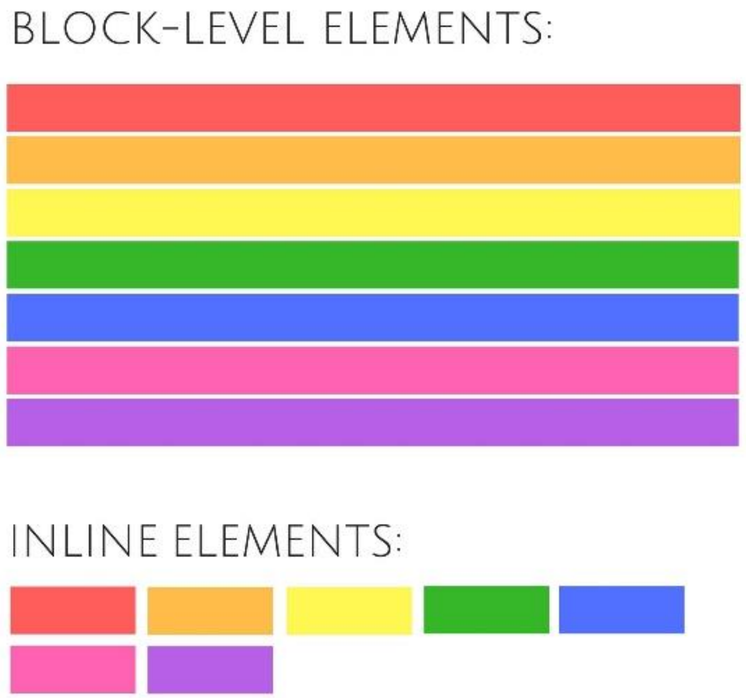
#html-css

What is Box Model?

What is CSS Box Model?

- The **CSS box model** is essentially a **box** that wraps **around every** HTML **element**
  - Visible from the browser Dev Tools: **[F12]**
- HTML elements have:
  - **Content** – the HTML element itself
  - **Padding** – transparent area around the content
  - **Border** – line that goes around the padding
  - **Margin** – transparent area outside the border





Block-level and Inline HTML Elements

Block and Inline Elements

- HTML is made up of various elements that act as the **building blocks** of web pages
- CSS has two different types of boxes - **block** and **inline**
  - **Block** Elements
  - **Inline** Elements
  - **Inline-block** Elements

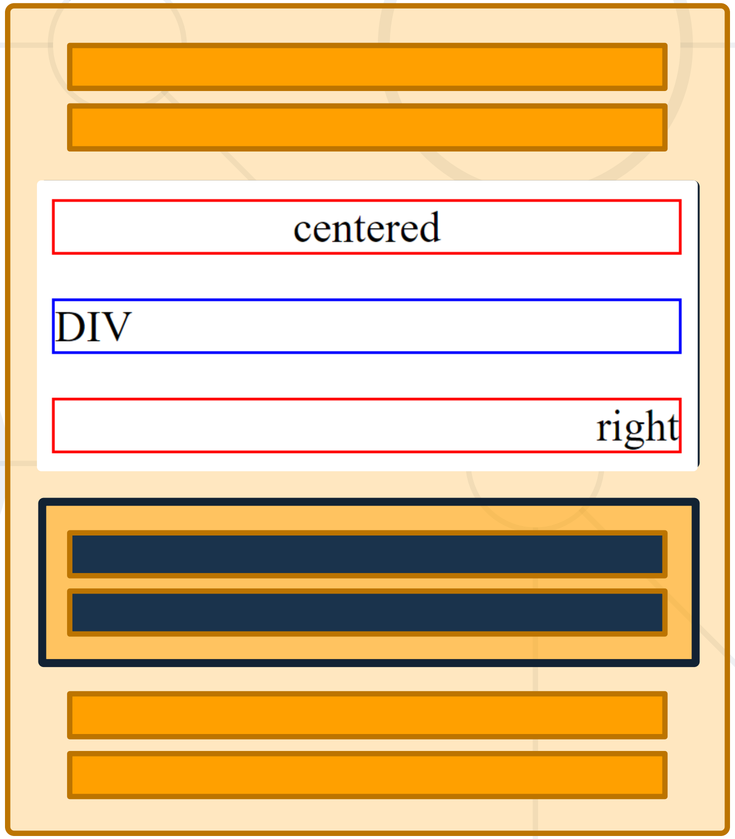
Block Elements

- Block element: starts on a **new line**, and fills up the horizontal space left and right on the web page
- Some examples of block elements are:
  - **`main`**, **`header`**, **`article`**, **`section`**, **`fieldset`**, **`nav`**, **`ul`**, **`ol`**, **`li`**, **`form`**, **`h1-h6`**, **`p`**, **`div`**

Block Elements – Example

```
<p style="border:1px solid red;
text-align:center">centered</p>
<div style="border:1px solid
blue">DIV</div>
<p style="border:1px solid red;
text-align:right">right</p>
```

display: block



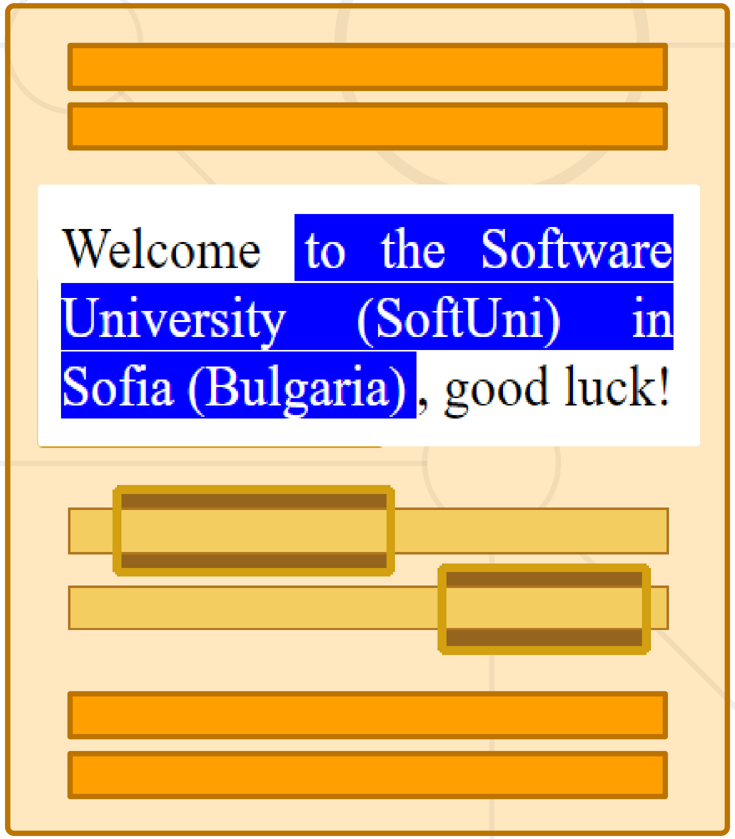
Inline Elements

- Inline element: **don't start** on a new line. They appear on the same line as the content and tags beside them
- Some examples of inline-block elements are:
  - **`a`**, **`label`**, **`map`**, **`span`**, **`strong`**, **`em`**, **`i`**, **`img`**, **`textarea`**, **`input`**, **`button`**, **`select`**
- You can add margins and padding just on **right** and **left** sides of any inline element

Inline Elements – Example

```
<p style="text-align:justify"> Welcome
<span style="color:white;
background:blue; padding-right:3px;
padding-left:3px;">
to the Software University (SoftUni)
in Sofia (Bulgaria)</span>, good
luck!</p>
```

display: inline



Welcome to the Software University (SoftUni) in Sofia (Bulgaria), good luck!

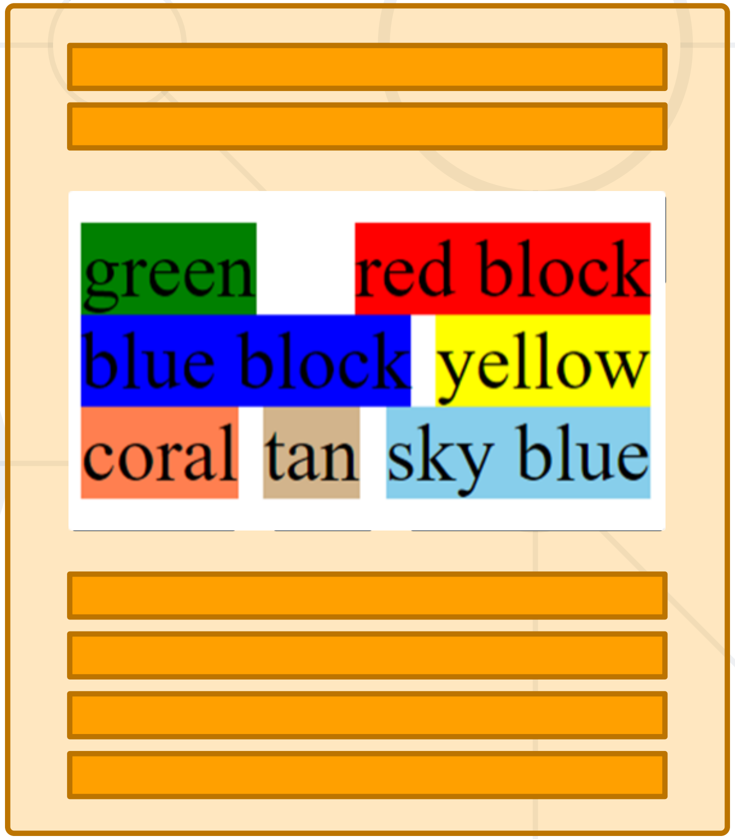
Inline-Block Elements

- Inline-block elements are similar to inline elements
- They can have padding and margins added on **all four sides**
- You have to declare **`display: inline-block`** in your CSS code
- One common use for using inline-block is for creating navigation links horizontally

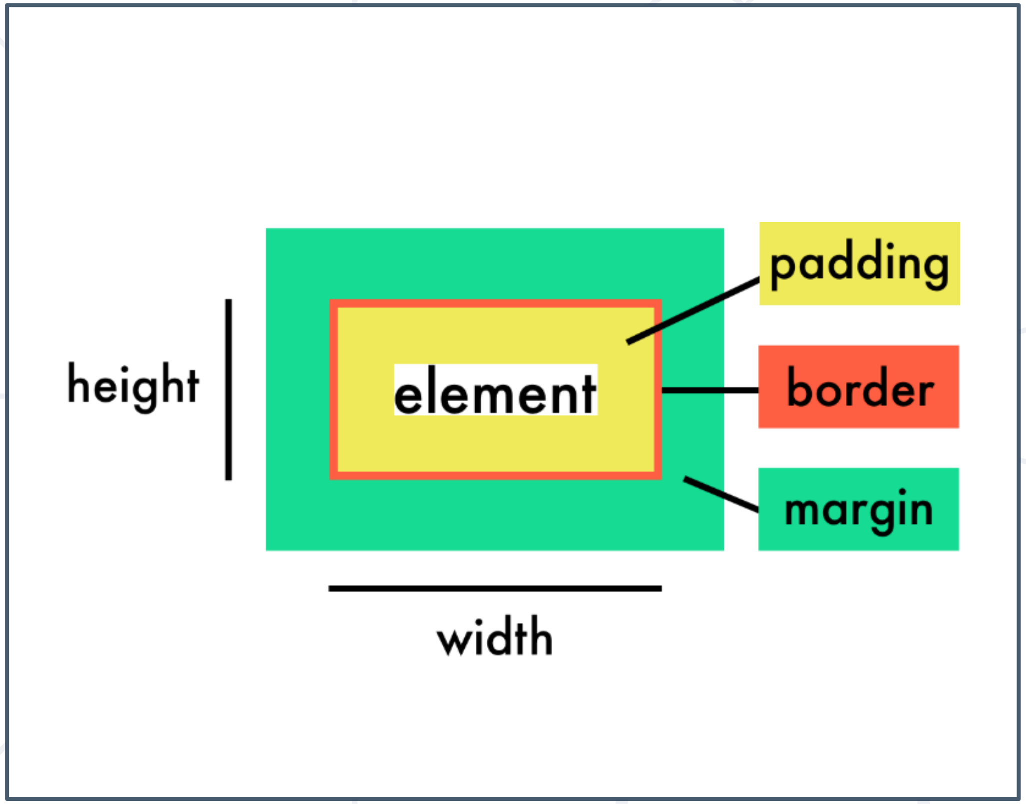
Inline-Block Elements – Example

```
<div style="text-align:justify;">
   <div style="display:inline-block;
   background:green">green</div>
   <div style="display:inline-block;
   background:red">red block</div>
   …
</div>
```

display: inline-block



green red block
blue block yellow
coral tan sky blue

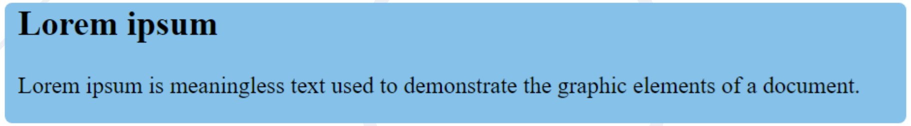


CSS Width and Height Dimensions

Width

- **Width** – defines the width of the element
  - **auto** (default)
  - Auto-calculated width

```
article {
  width: auto;
  background: #8ce;
}
```

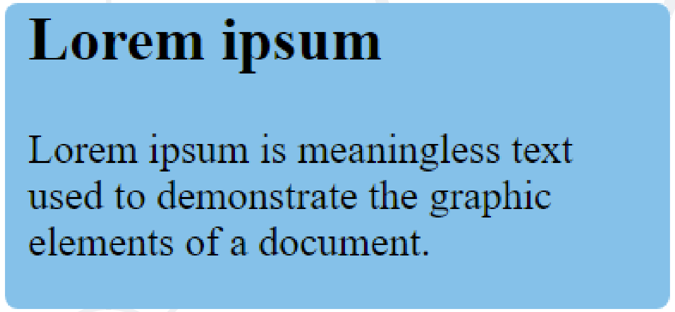


Lorem ipsum

Lorem ipsum is meaningless text used to demonstrate the graphic elements of a document.

  - **pixels** / **em** / **rem**
  - Fixed width

```
article {
  width: 240px;
  background: #8ce;
}
```

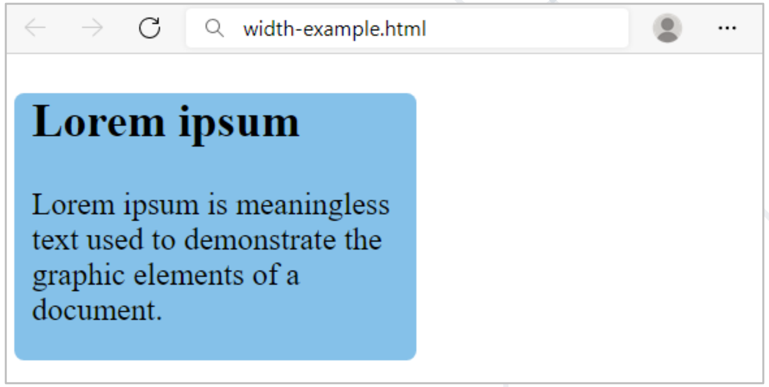


Lorem ipsum

Lorem ipsum is meaningless text used to demonstrate the graphic elements of a document.

  - **percentages**
  - Relative to container's width

```
article {
  width: 50%;
  background: #8ce;
}
```

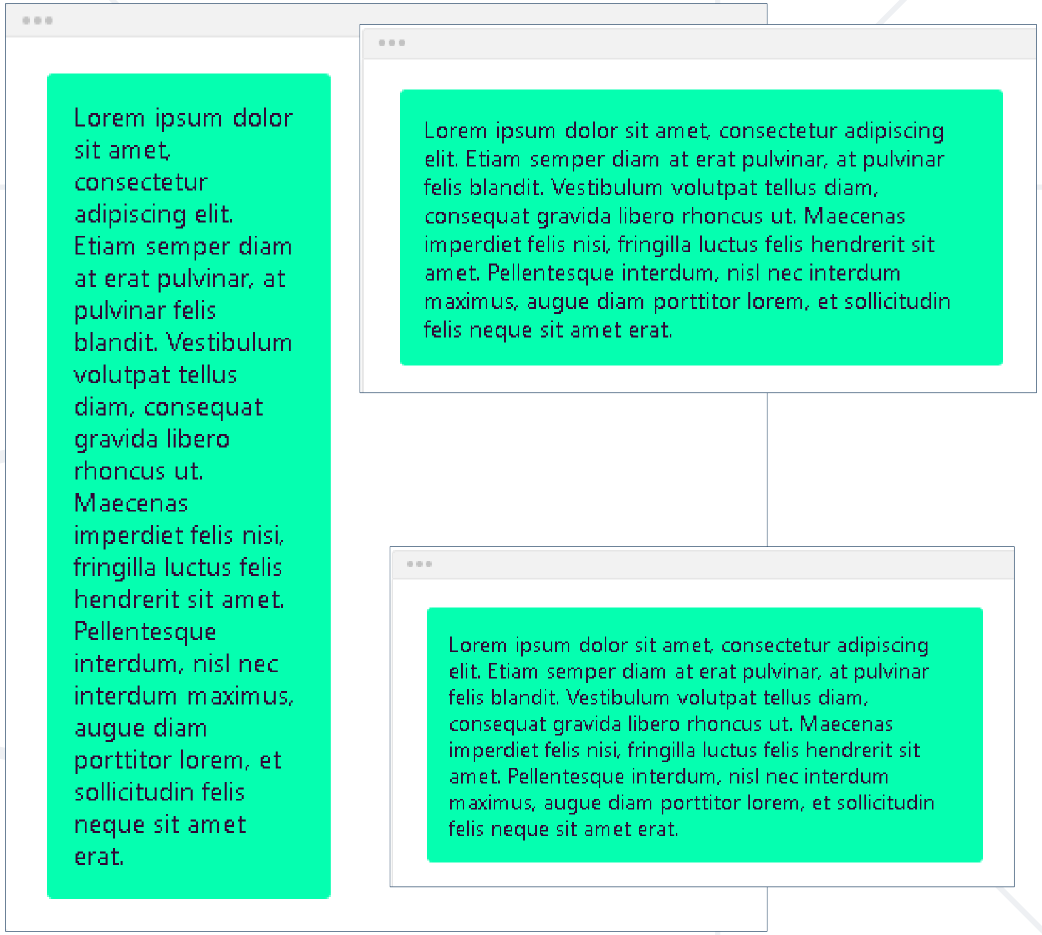


Lorem ipsum

Lorem ipsum is meaningless text used to demonstrate the graphic elements of a document.

Max-width

- Max-width - defines the **maximum** width the element can be
  - **`max-width: none;`** - the element has **no limit** in terms of width
  - **`max-width: 150px;`**
  - **`max-width: 2000px;`** - you can use numeric values like **pixels**, **(r)em**, **percentages**…
- If the **maximum** width is **larger** than the element's **actual** width, the max width has **no effect**



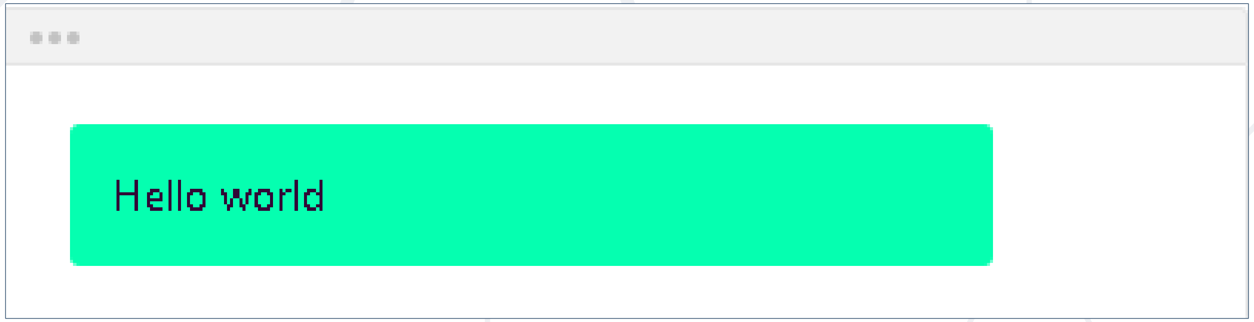
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.

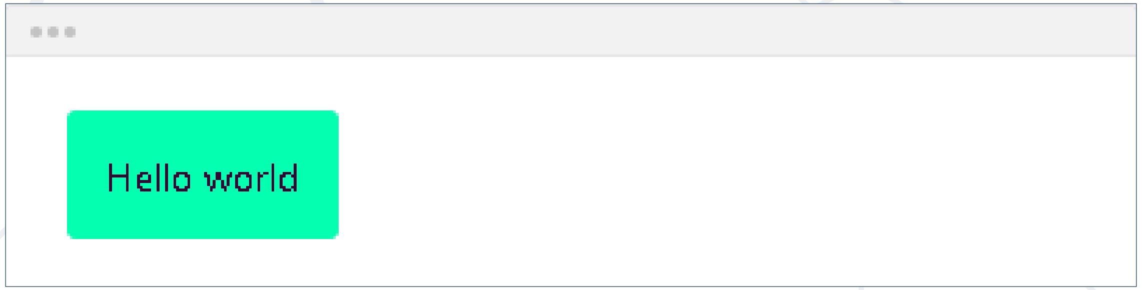
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.

Min-width

- Min-width - defines the **minimum** width the element
  - **`min-width: 300px;`** - if the **minimum** width is **larger** than the element's **actual** width, the min width will be applied
- If the **minimum** width is **smaller** than the element's **actual** width, the min width has **no effect** - **`min-width: 5px;`**



Hello world

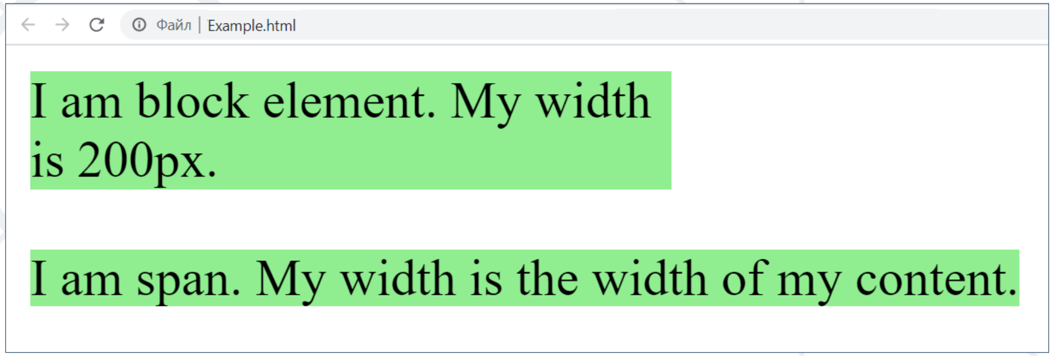


Hello world

Width – Example

```
<body>
    <div>
        I am block element. My width is 200px.
    </div>
    <span>I am span. My width is the width of my content.</span>
</body>
```

```
div, span {
    width: 200px;
    background-color: lightgreen;
}
```



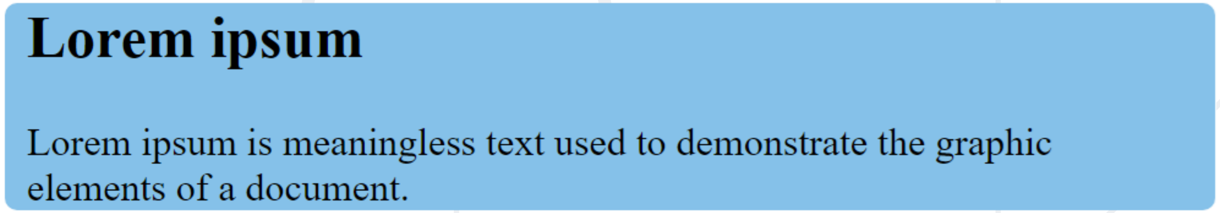
I am block element. My width is 200px.

I am span. My width is the width of my content.

Height

- **Height** – defines the height of the element
  - **auto** (default)
  - Auto-calculated height
  - numeric values like **px** / **pt** / **em** / **rem** / **%**

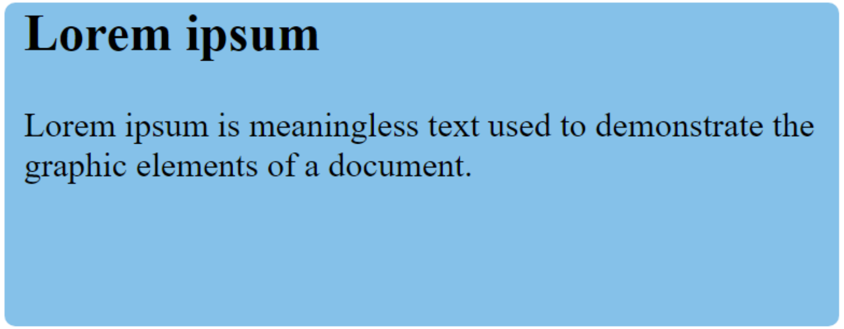
```css
article {
  height: auto;
  background: #8ce;
}
```



Lorem ipsum

Lorem ipsum is meaningless text used to demonstrate the graphic elements of a document.

```css
article {
  height: 100px;
  background: #8ce;
}
```



Lorem ipsum

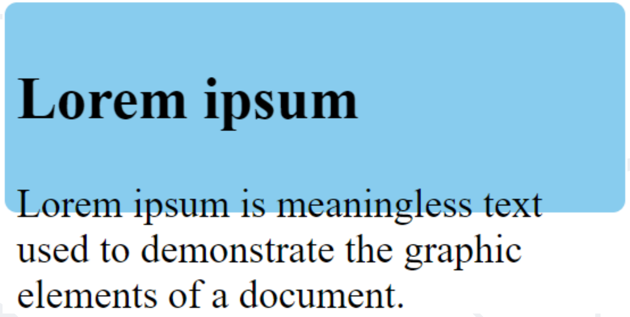
Lorem ipsum is meaningless text used to demonstrate the graphic elements of a document.

Overflow

- The **overflow** property specifies what should happen if **content overflows** an element's box
  - **visible**
  - **auto**
  - **scroll**
  - **hidden**

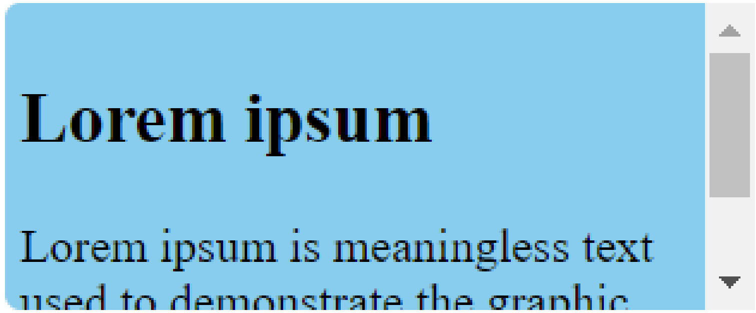
```
article {
  overflow: visible;
}
```

visible


Lorem ipsum

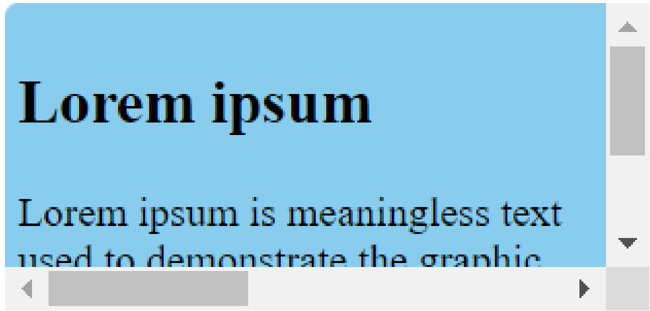
Lorem ipsum is meaningless text used to demonstrate the graphic elements of a document.

auto


Lorem ipsum

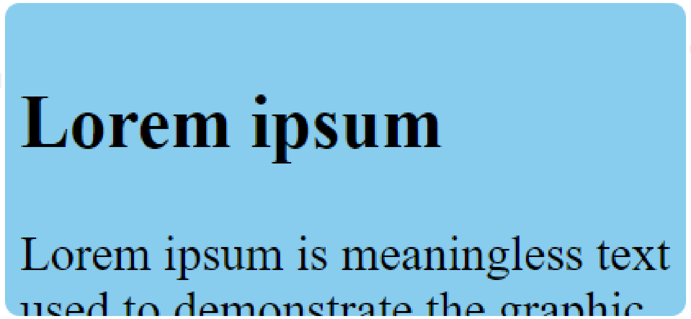
Lorem ipsum is meaningless text used to demonstrate the graphic

scroll


Lorem ipsum

Lorem ipsum is meaningless text used to demonstrate the graphic

hidden

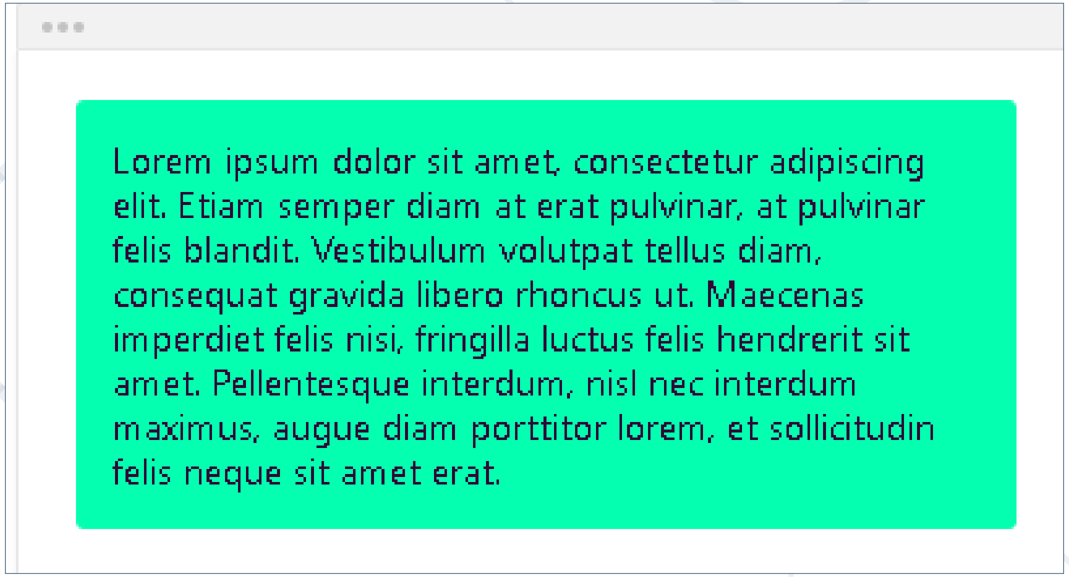

Lorem ipsum

Lorem ipsum is meaningless text used to demonstrate the graphic

Max-height

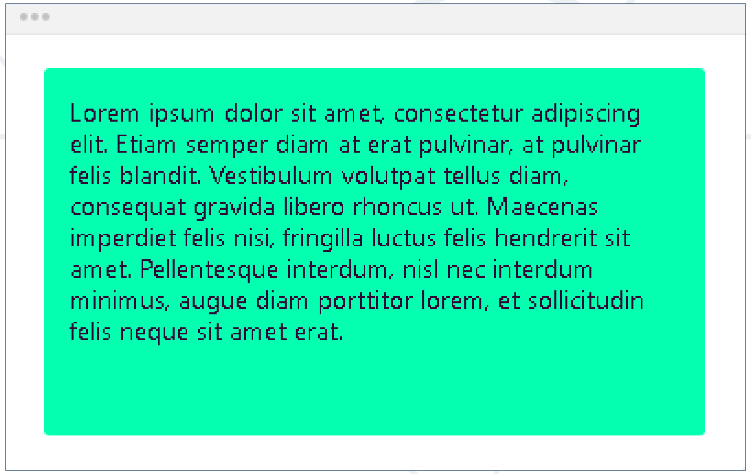
- Max-height - defines the maximum height the element can be
  - **max-height: none;** - the element has **no limit** in terms of height
  - **max-height: 2000px;**

    - if the **maximum** height is **larger** than the element's **actual** height, the max height has **no effect**

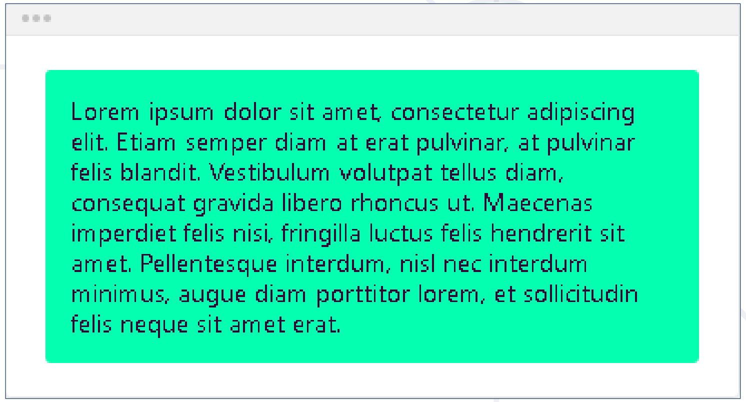

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum maximus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.

Min-height

- Min-height - defines the minimum height the element
  - **min-height: 200px;** - if the **minimum** height is **larger** than the element's **actual** height, the min height will be applied
  - **min-height: 5px;** - if the **minimum** height is **smaller** than the element's **actual** height, the min height has **no effect**



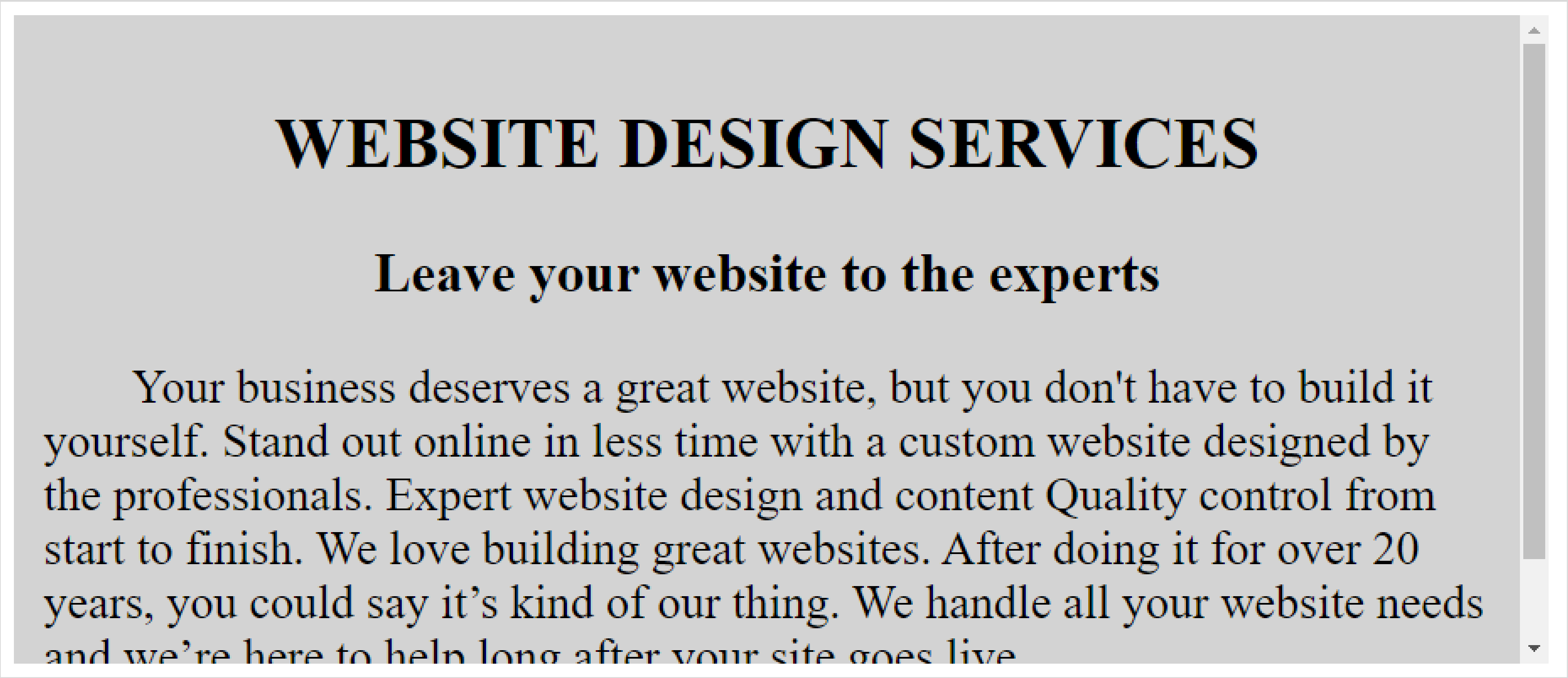
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum minimus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Etiam semper diam at erat pulvinar, at pulvinar felis blandit. Vestibulum volutpat tellus diam, consequat gravida libero rhoncus ut. Maecenas imperdiet felis nisi, fringilla luctus felis hendrerit sit amet. Pellentesque interdum, nisl nec interdum minimus, augue diam porttitor lorem, et sollicitudin felis neque sit amet erat.

Scrolling Article – Exercise

- Using **HTML** and **CSS** create a **section** with **articles** like the following:



WEBSITE DESIGN SERVICES

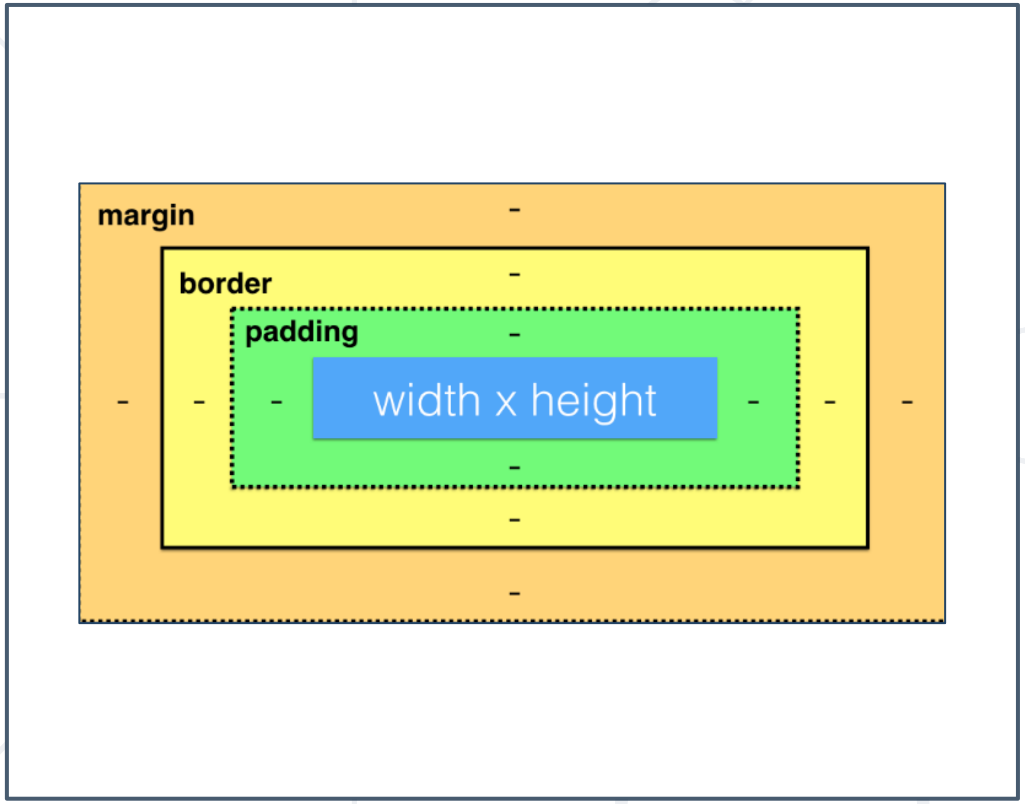
Leave your website to the experts

Your business deserves a great website, but you don't have to build it yourself. Stand out online in less time with a custom website designed by the professionals. Expert website design and content Quality control from start to finish. We love building great websites. After doing it for over 20 years, you could say it’s kind of our thing. We handle all your website needs and we’re here to help long after your site goes live

Scrolling Article – Solution

- **HTML** constraints:
  - Use <section> with <article>, which contains: <h2>, <h3> and <p>
- Hints:

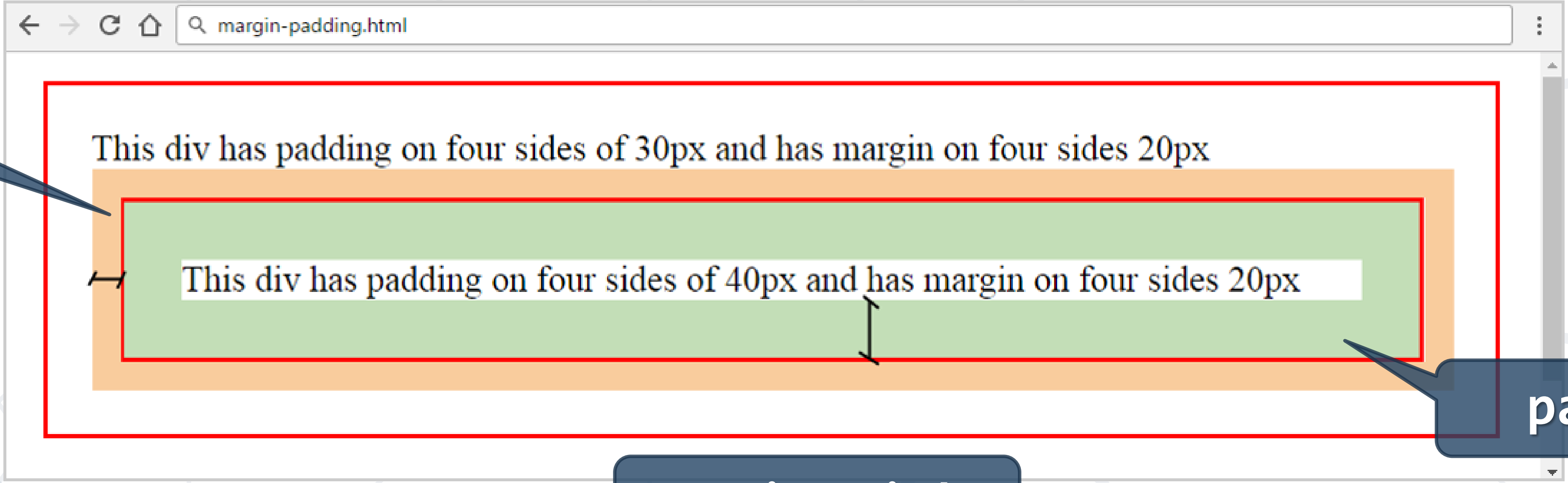
```css
article {
  width: 500px;
  height: 200px;
  background-color: lightgray;
  overflow: auto;
  padding: 10px;
}
```



Margin, Padding and Borders

CSS Box Model Layers

Margins and Paddings



This div has padding on four sides of 30px and has margin on four sides 20px

This div has padding on four sides of 40px and has margin on four sides 20px

margin

padding

margin-right

margin-top

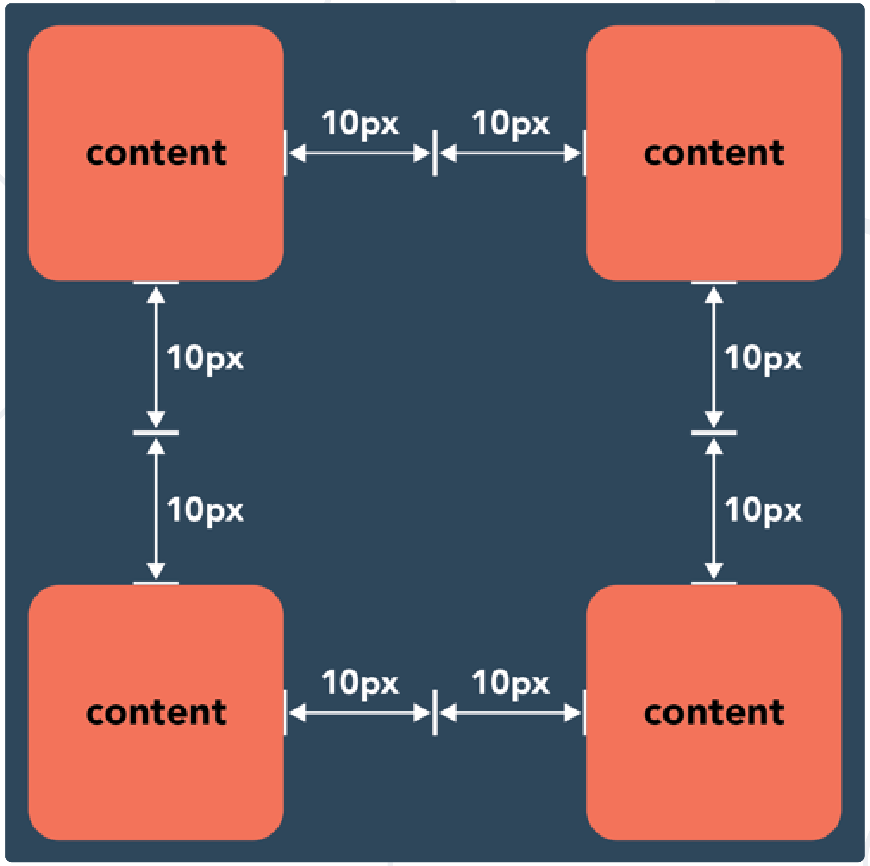
margin-bottom

margin-left

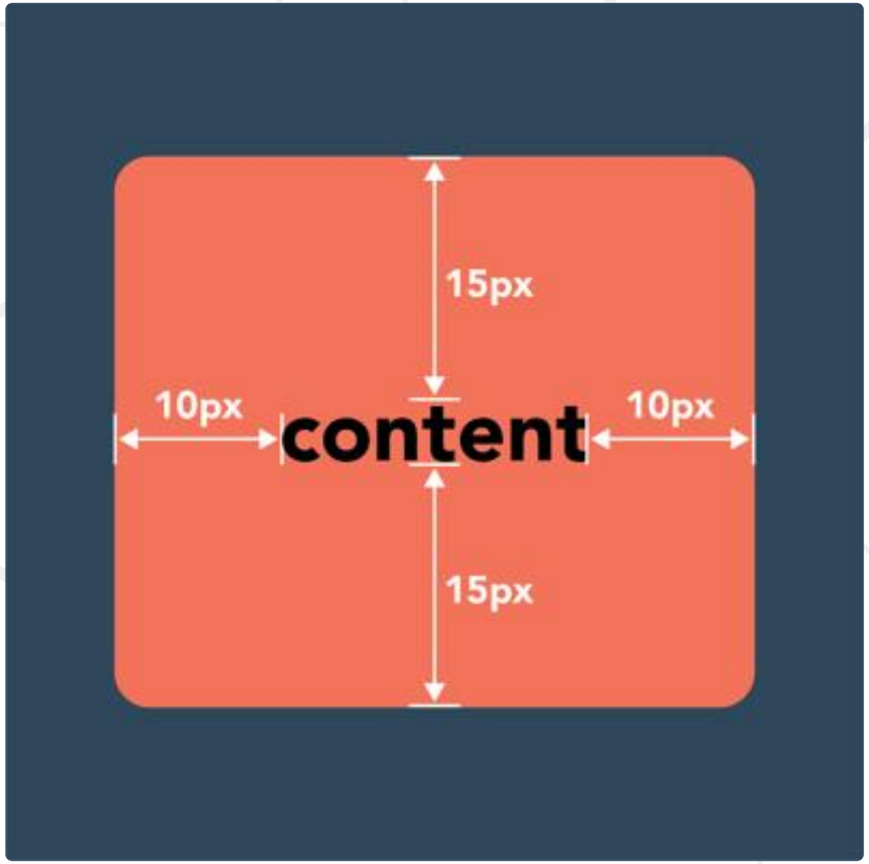
```
div {
    margin: 20px 20px 20px 20px;
    padding: 30px 30px 30px 30px;
}
```

Margins and Paddings

- **Margin** – defines the space **outside** the element



- **Padding** – defines the space **inside** the element

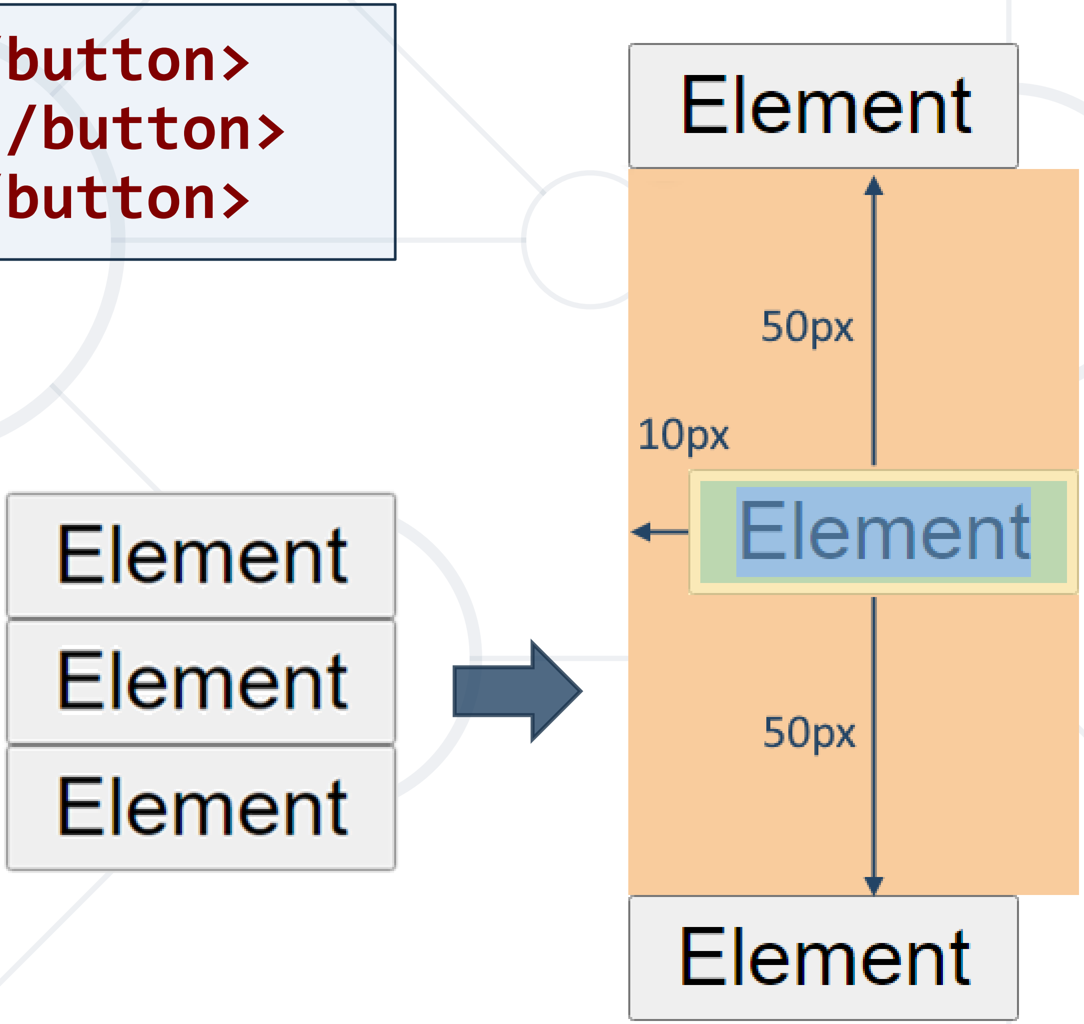


Margin

```html
<button class="first">Element</button>
<button class="second">Element</button>
<button class="third">Element</button>
```

```css
button {
  display: block;
}

button.second {
  margin-top: 50px;
  margin-left: 10px;
  margin-bottom: 50px;
  margin-right: 0;
}
```

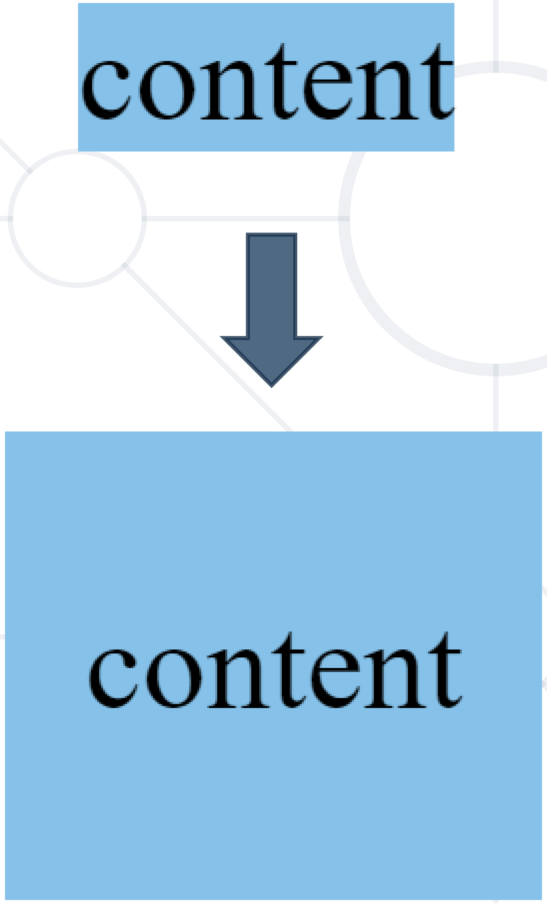


Padding

```
<div>content</div>
```

```
div {
  background-color: #85c1e9;
  display: inline-block;
  text-align: center;

  padding-top: 20px;
  padding-left: 10px;
  padding-bottom: 20px;
  padding-right: 10px;
}
```



Shorthand Margin / Padding

- Shorthand margin rules:

```
button.second {
  margin: 10px 20px 10px 20px;
}
```

top

right

bottom

left

```
button.second {
  margin: 20px 10px;
}
```

top & bottom

left & right

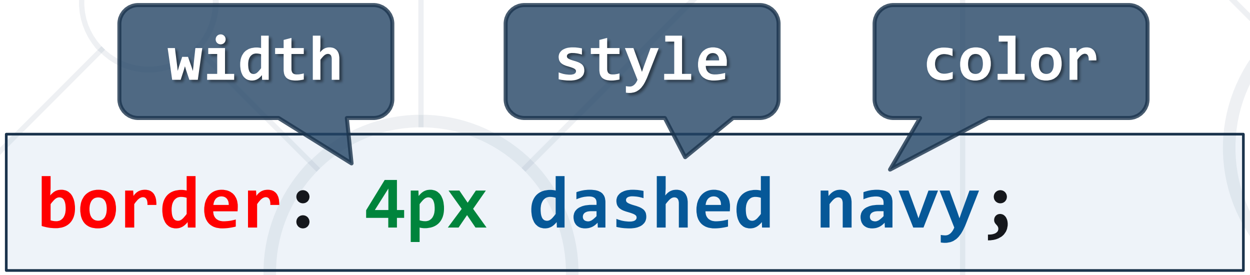
- Shorthand padding rules:

```
div {
  padding: 5px 10px 8px 15px;
}
```

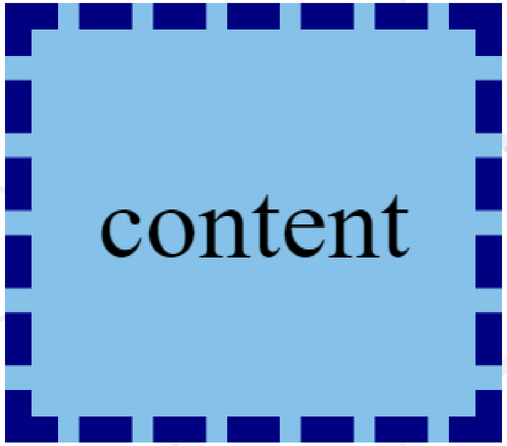
```
li {
  padding: 5x 10px;
}
```

Border

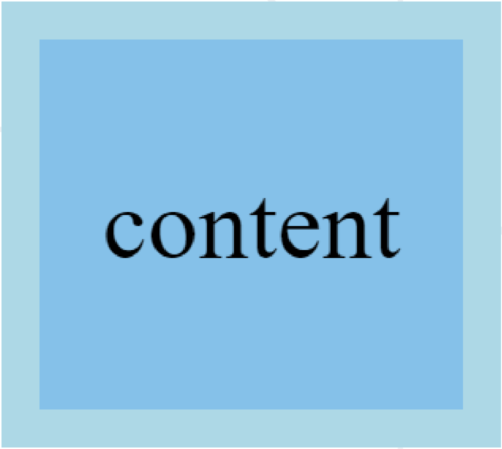
- **Border** – define the style of the borders:
  - **width** (e. g. **1px** / **2px** / **3px**)
  - **style** (e. g. **solid** / **dashed** / **dotted**)
  - **color** (e. g. **blue** / **#eee**)



```
border: 4px dashed navy;
```

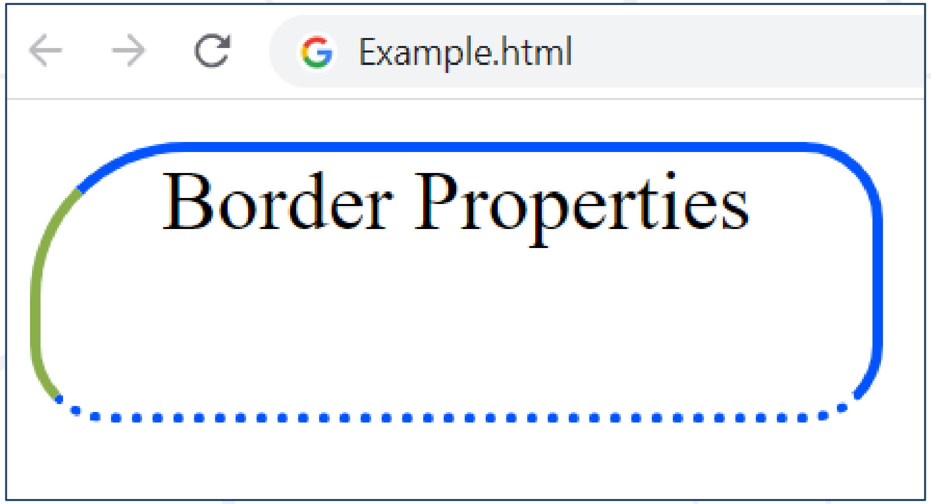


```
border: 6px solid lightblue;
```



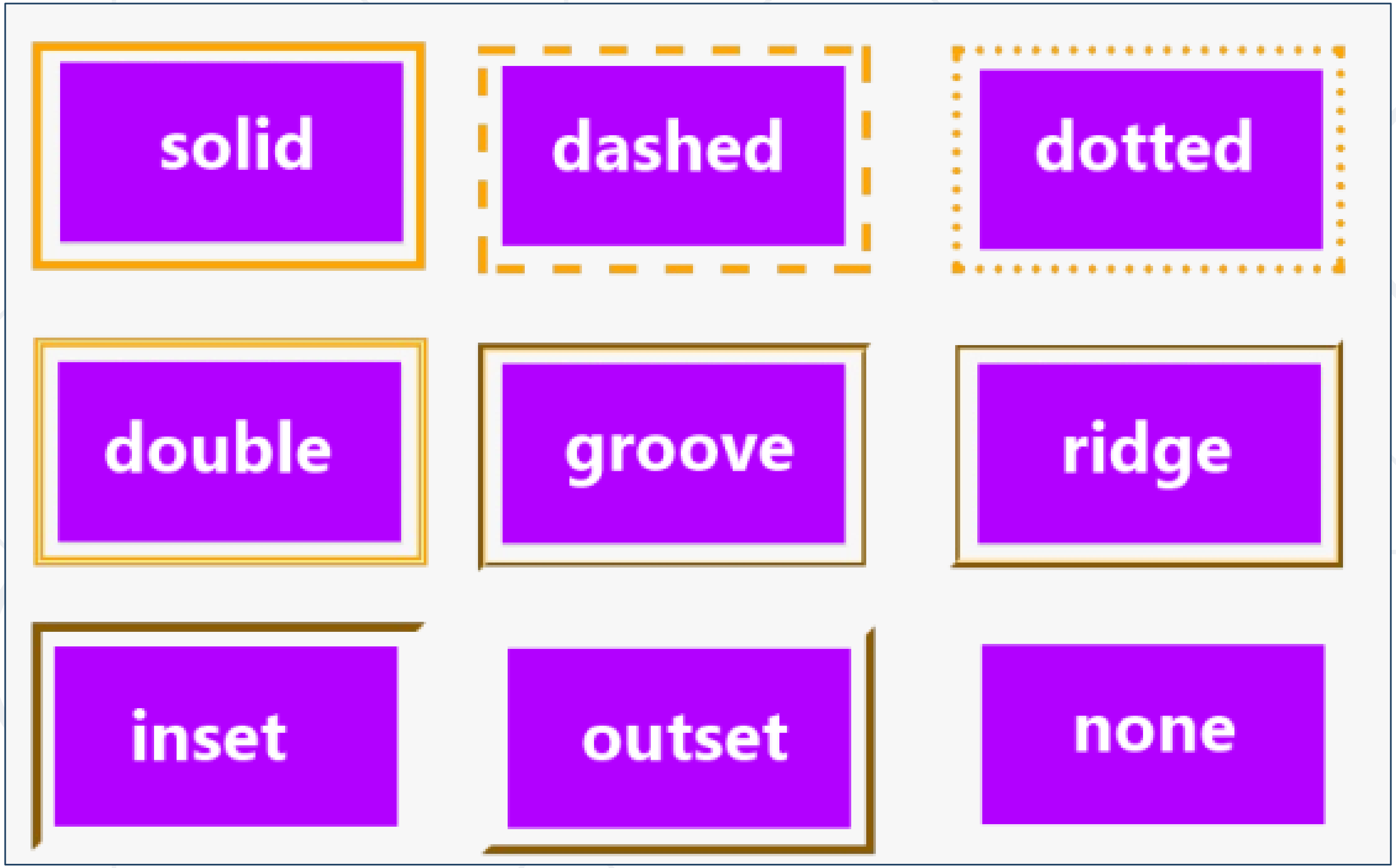
Border Properties

```
div {
  width: 160px;
  height: 50px;
  border-width: 2px;
  border-style: solid;
  border-color: #0053ff;
  border-radius: 15px;
  border-top-left-radius: 30px;
  border-bottom-style: dotted;
  border-left-color: #89af4c;
  text-align: center;
}
```



Border Properties

CSS Borders

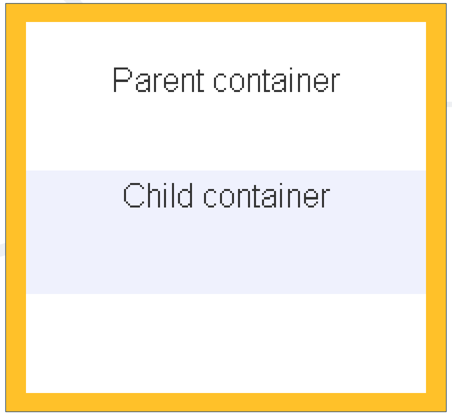


Include the Padding and Border in an Element's Total Width and Height

Box-sizing

- Sets how the total width and height of an element is calculated
  - **content-box** - initial and default value
  - The **width** and **height** properties include the content
  - They **do NOT include** the padding, border and margin

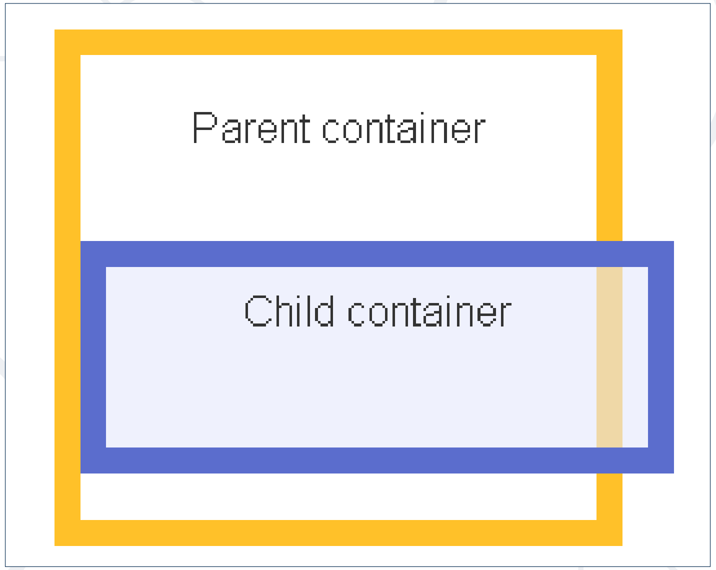
```
div {
  box-sizing: content-box;
  width: 200px;
}
```



Box-sizing

- The full width is: **200px + 2*10px + 2*5px = 230px**

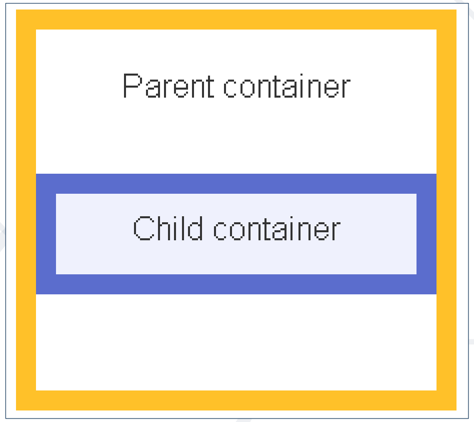
```
div {
  box-sizing: content-box;
  width: 200px;
  border: 10px solid #5b6dcd;
  padding: 5px;
}
```



Box-sizing

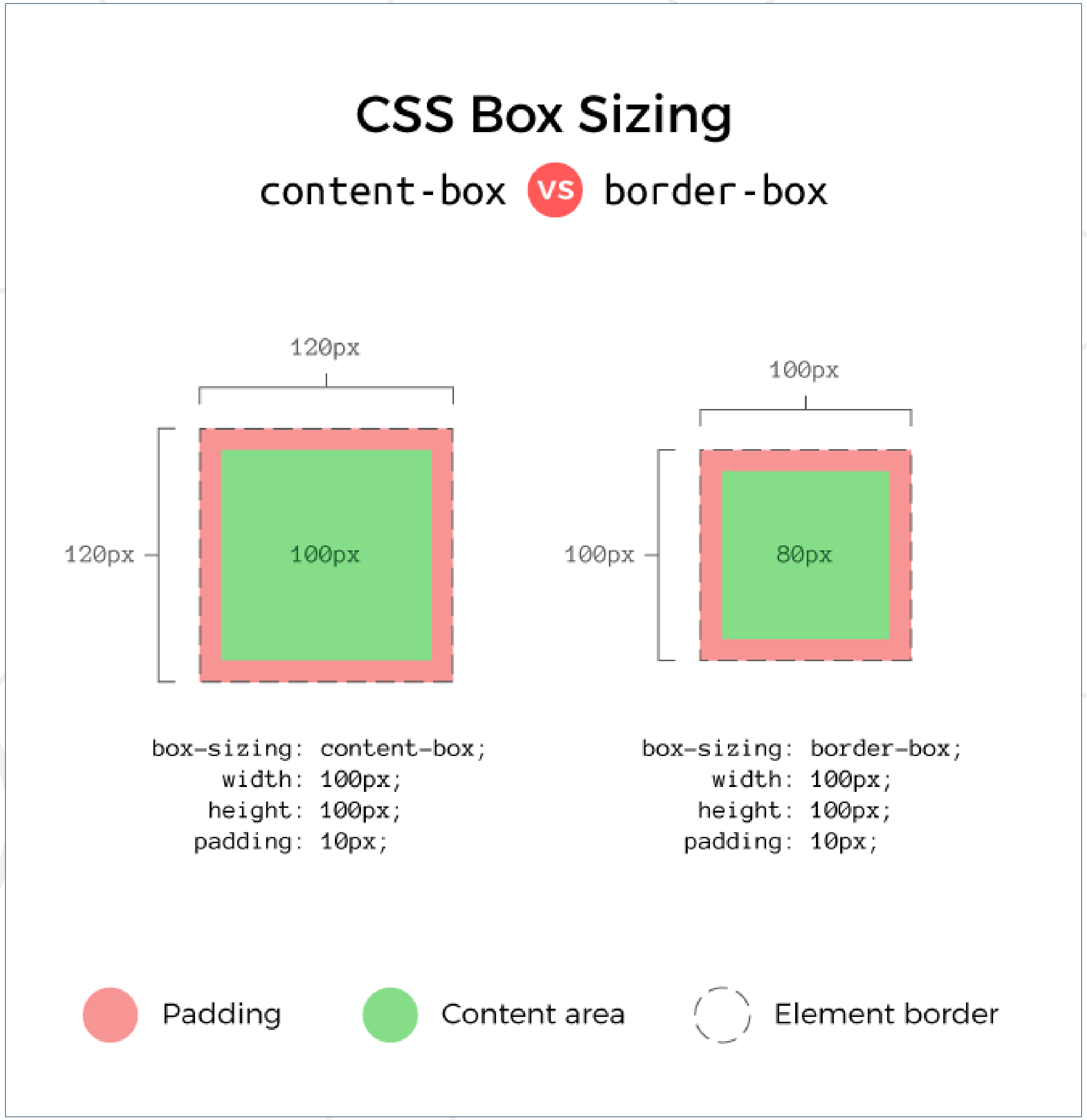
- **border-box** - the **width** and **height** of the element apply to all parts of the element: the **content**, the **padding** and the **borders**

```
box-sizing: border-box;
width: 200px;
border: 10px solid #5b6dcd;
padding: 5px;
```



- The full width is **200px**
- The content width is equal to: **200px - 2*10px - 2*5px = 170px**

Content-box vs Border-box



CSS Box Sizing

content-box vs border-box

```
box-sizing: content-box;
     width: 100px;
    height: 100px;
   padding: 10px;
```

```
box-sizing: border-box;
     width: 100px;
    height: 100px;
   padding: 10px;
```

Universal Box-sizing

- The box-sizing **Reset** takes care of the box-sizing of every element by setting it to border-box using universal CSS selector
- Save your **time** and don't write the same thing **again-and-again**
- Set the "**universal box-sizing**" with inheritance:

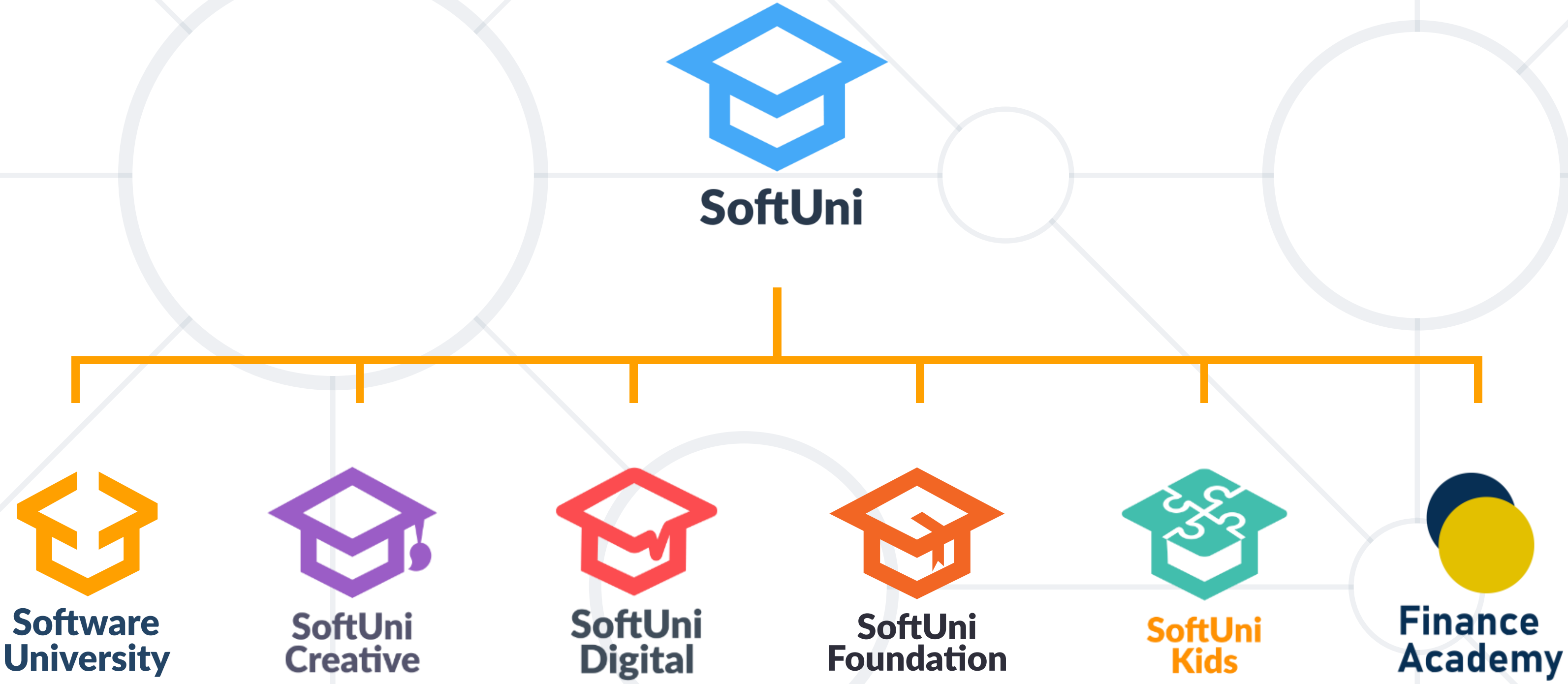
```css
html {
  box-sizing: border-box;
}

*,
*:before,
*:after {
  box-sizing: inherit;
}
```

Summary

- What is **Box Model**?
- **Width** and **Height** to the elements
- What are the **padding**, **border** and **margin**?
- What is **box-sizing**?
- How to **reset** box-sizing?

Questions?



SoftUni

Software University

SoftUni Creative

SoftUni Digital

SoftUni Foundation

SoftUni Kids

Finance Academy

SoftUni Diamond Partners

SUPER HOSTING .BG

Coca-Cola HBC Bulgaria

POKERSTARS
POKER | CASINO | SPORTS
a Flutter International brand

INDEAVR
Serving the high achievers

AMBITIONED

DRAFT KINGS

SOFTWARE GROUP

createX

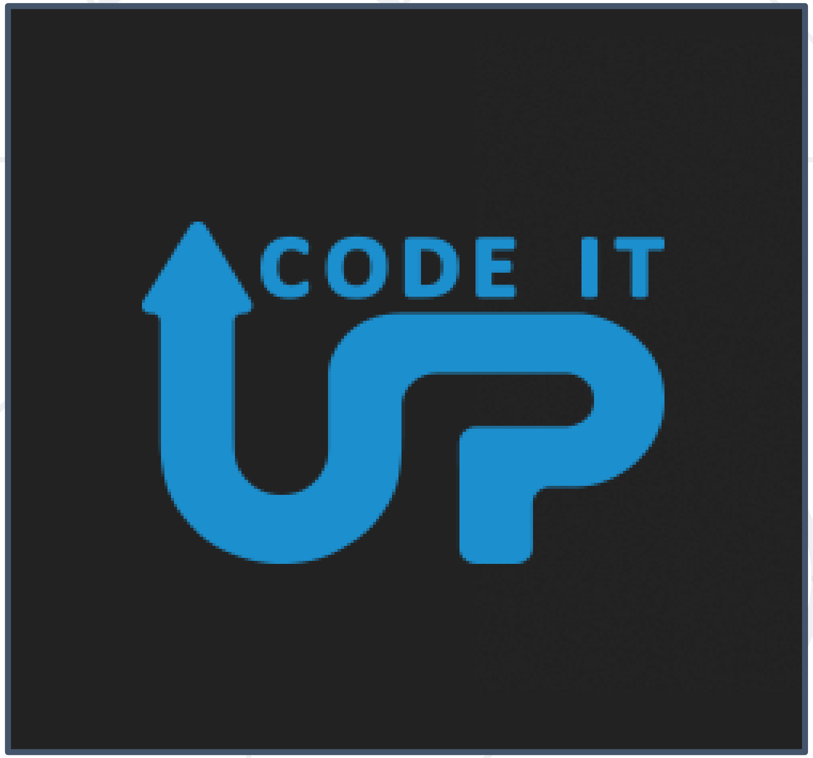
Postbank
Решения за твоето утре

BOSCH

DXC TECHNOLOGY

SmartIT

Educational Partners



License

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity
- Software University Forums
  - forum.softuni.bg