



**altexsoft**  
software r&d engineering

# Introduction

BOX MODEL, DISPLAY PROPERTIES, FLOAT



# CSS selectors priorities

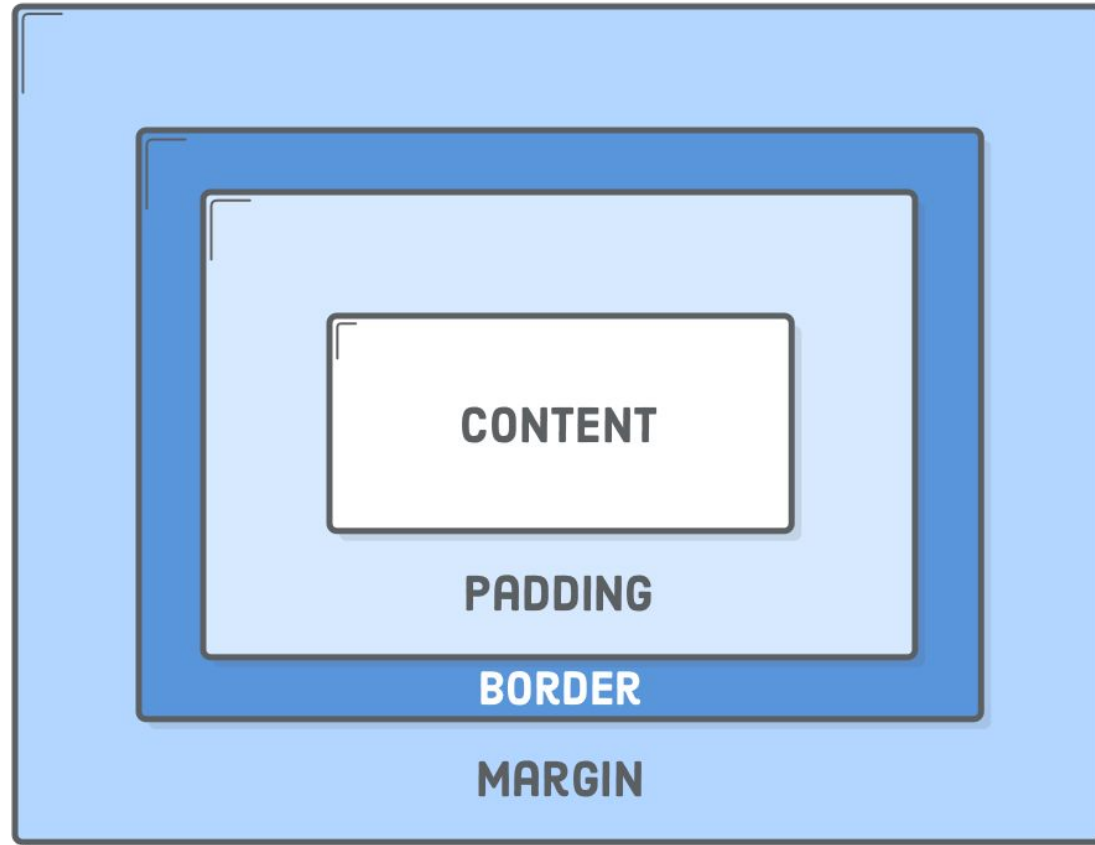
---

- *Tag selector has the minimal priority. Use to define generic styles*
- *Class selector has more priority then the tag one. Use to define styles for multiple elements and avoid repeating of the code.*
- *ID selector has more priority then the class. Actually is not very good idea to use it.*
- *Descendant selector has more priority then the class. Used to define condition from the parent and scope individual styles.*
- *If there are equals selectors, the most priority will have those which is bellow in the code.*



# Box model

---





# Padding

---

Individual values	Shorthand
padding-top: 10px;	padding: 10px 20px 30px 40px;
padding-right: 20px;	padding: 10px 20px 30px;
padding-bottom: 30px;	padding: 10px 20px;
padding-left: 40px;	padding: 10px;



# Margin

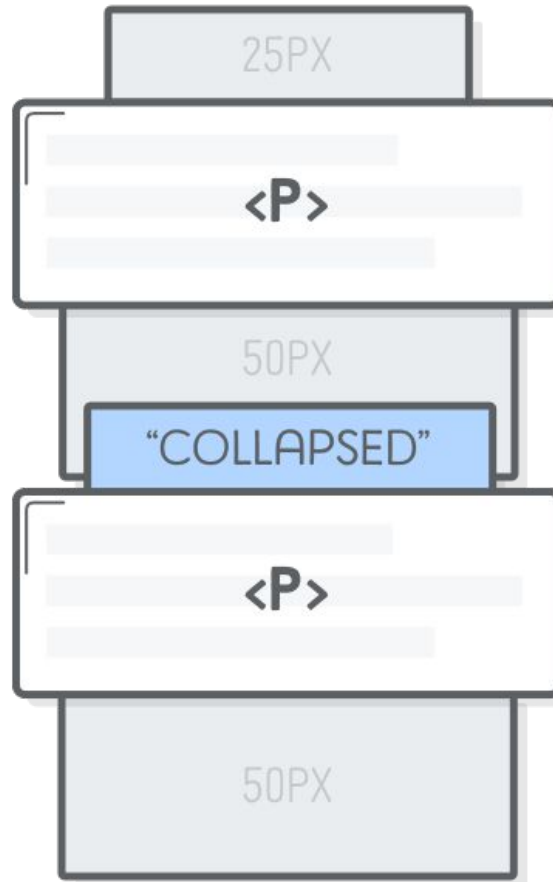
---

Individual values	Shorthand
margin-top: 10px;	margin: 10px 20px 30px 40px;
margin-right: 20px;	margin: 10px 20px 30px;
margin-bottom: 30px;	margin: 10px 20px;
margin-left: 40px;	margin: 10px;



# Vertical margin collapse

---





# Use margin for centering your layout

---

**Pay attention:** the centered block should have a defined width



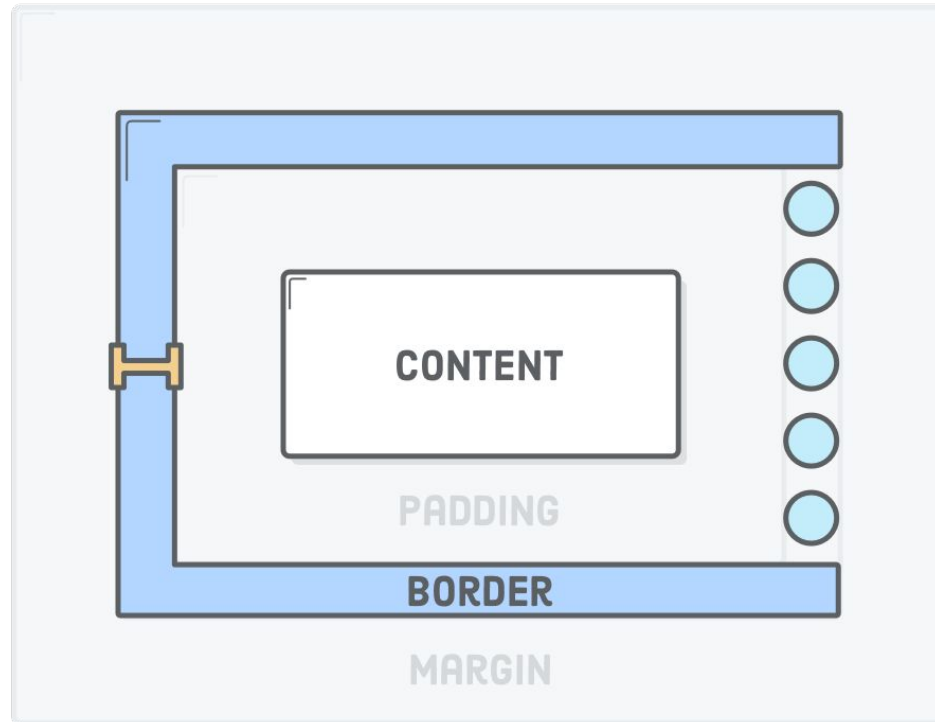
**CENTER ALIGN**

MARGIN: 0 AUTO;



# Border

---



```
BORDER : SIZE STYLE COLOR
```





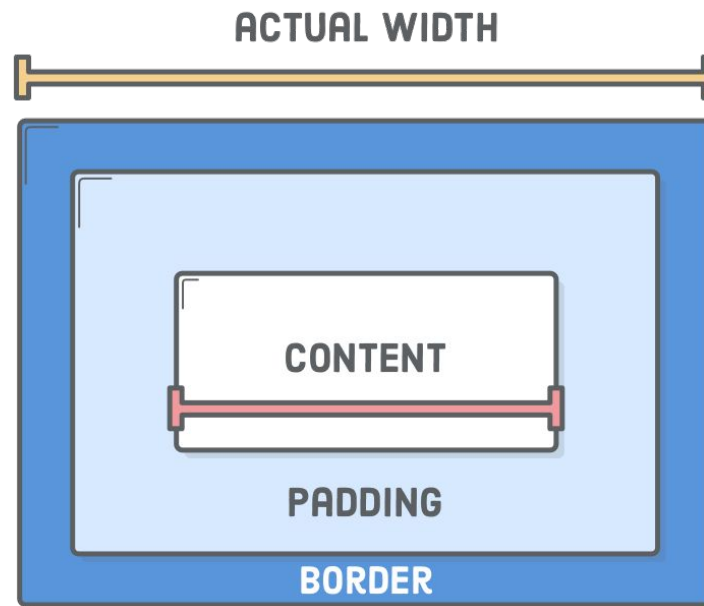
# Border properties

---

Property	Example
style	<code>border-style: dotted;</code>
width	<code>border-width: 5px;</code>
color	<code>border-color: red;</code>
side	<code>border-top-style: dotted;</code> <code>border-bottom-color: green;</code> <code>border-left-width: 10px;</code>
shorthand	<code>border: 5px solid red;</code>



# Box-sizing: content-box



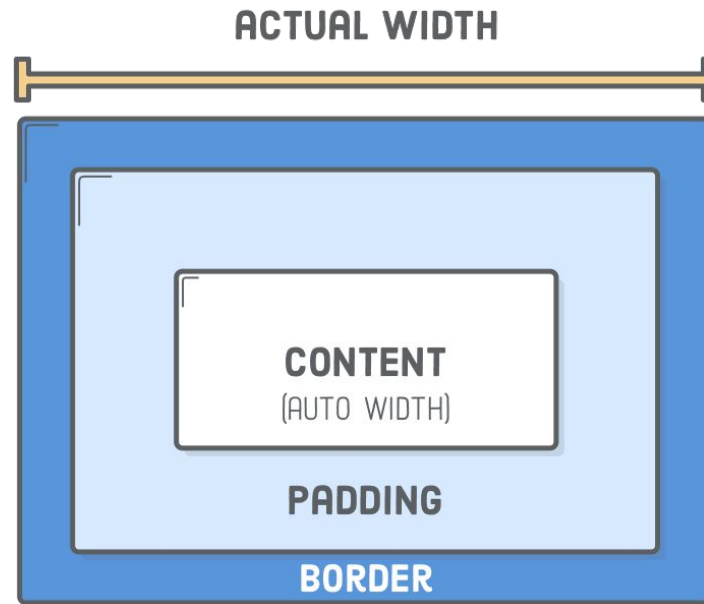
WIDTH : 200PX

BOX-SIZING : CONTENT-BOX



# Box-sizing: border-box

---



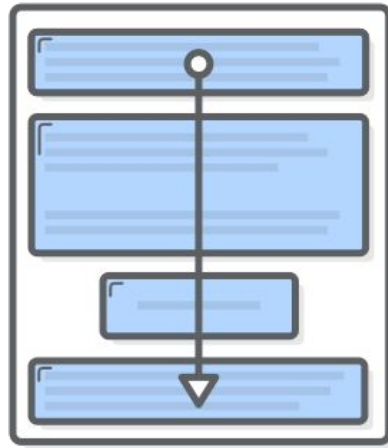
WIDTH : 200PX

BOX-SIZING : BORDER-BOX

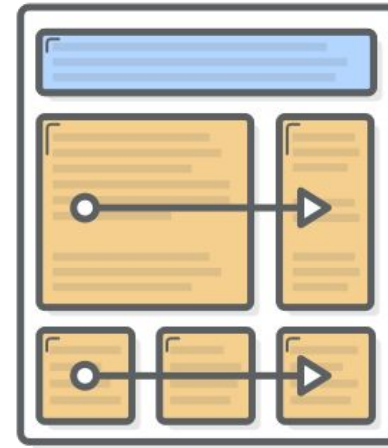


# Floats

---



**VERTICAL FLOW**

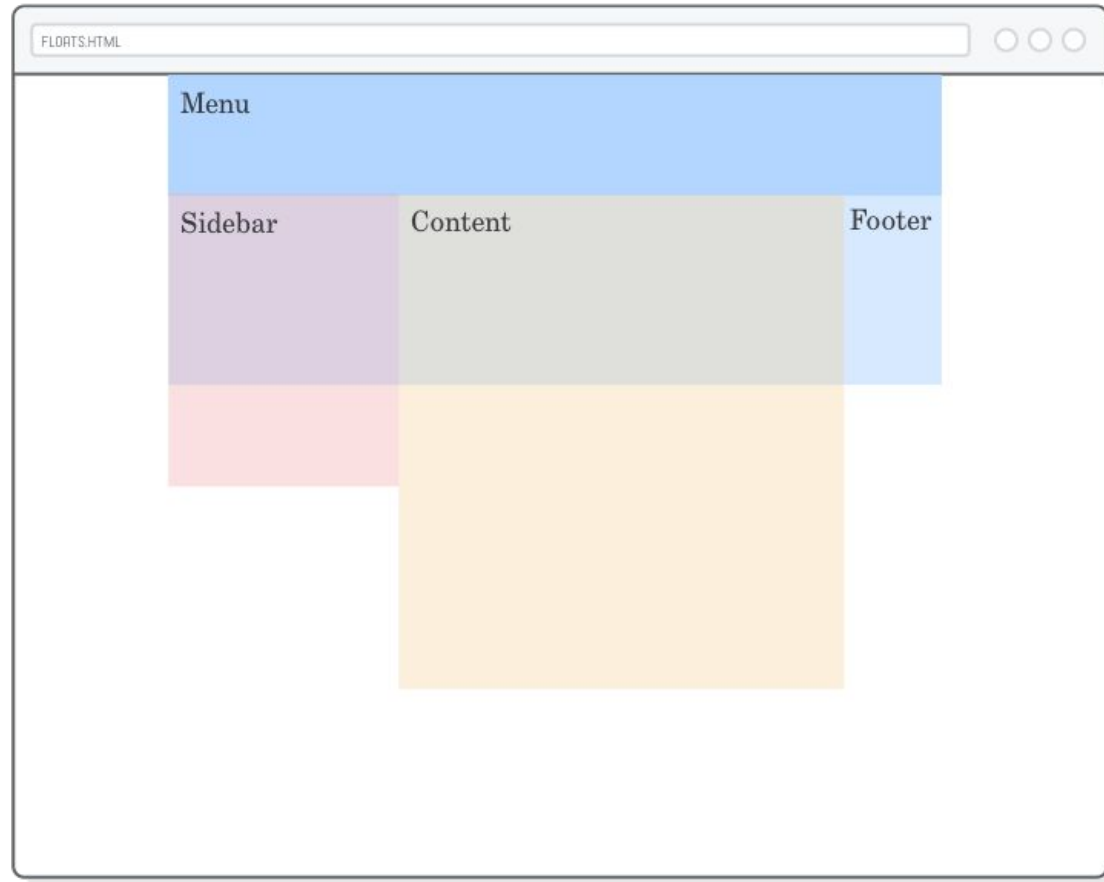


**HORIZONTAL FLOW**



# Float by itself

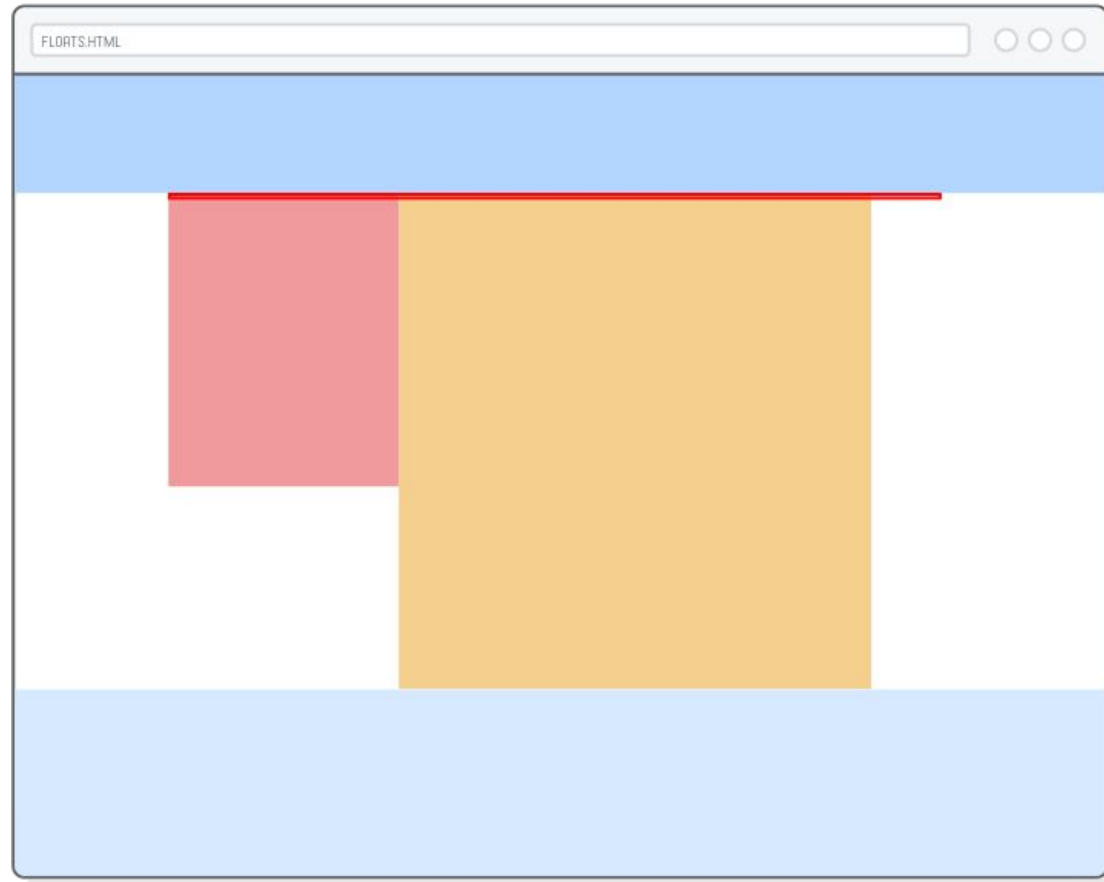
---





# How to test it?

---

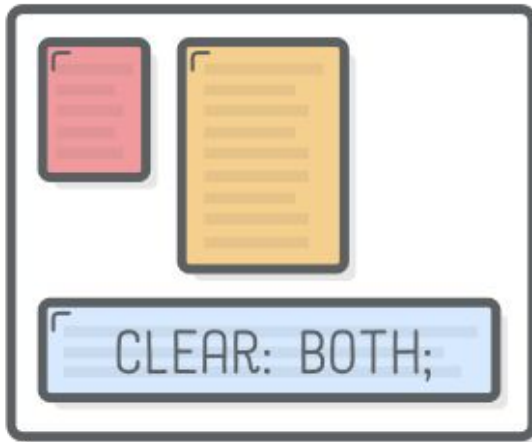




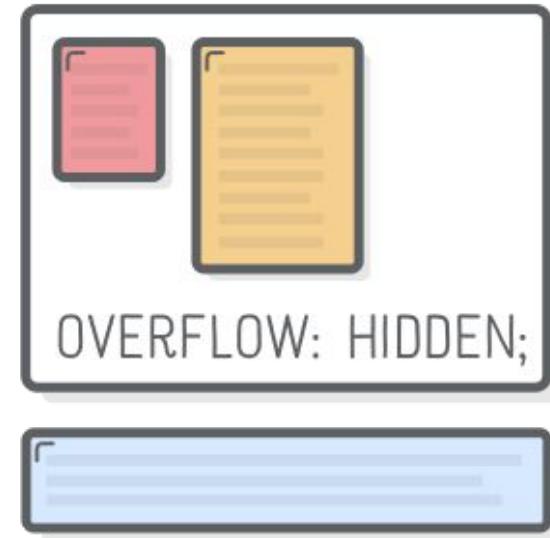
# How to fix it?

---

## CLEARING WITH CHILD ELEMENT

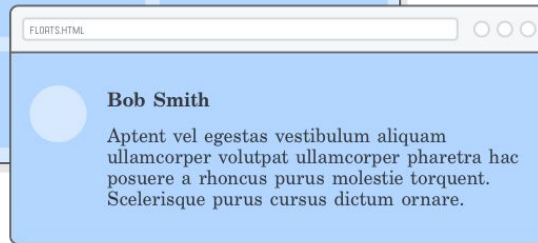
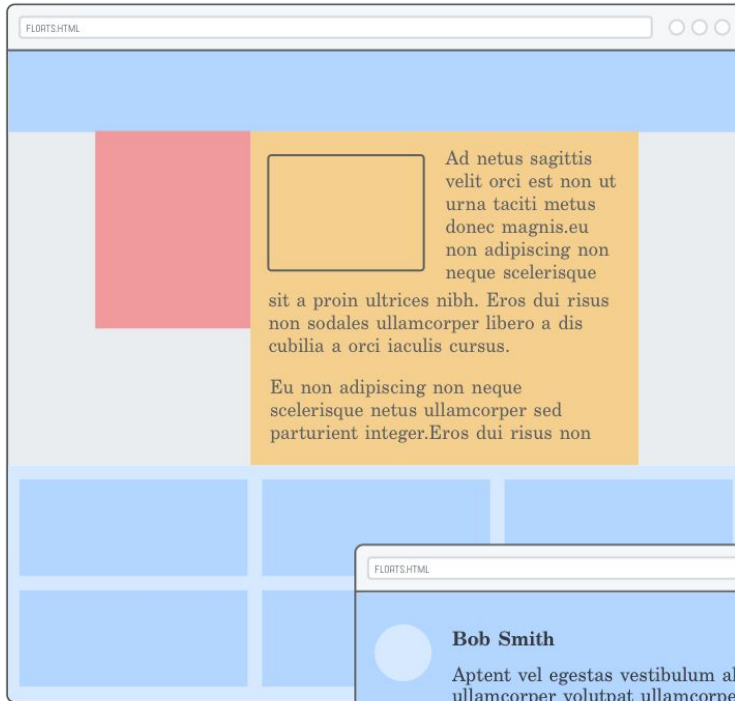


## CLEARING WITH PARENT ELEMENT

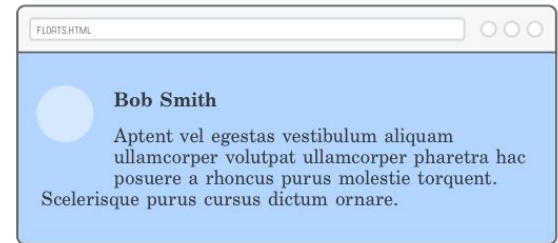




# Floats for content



**WITH HIDDEN OVERFLOW**



**WITHOUT HIDDEN OVERFLOW**





# “Display” property

---



“BLOCK BOX”



“INLINE BOXES”



# Default display property

---

BOXES.HTML

## Headings Are Block Elements

Paragraphs are blocks, too. *However*, `<em>` and `<strong>` elements are not. They are **inline** elements.

Block elements define the flow of the HTML document, while inline elements do not.



# Block and inline behavior

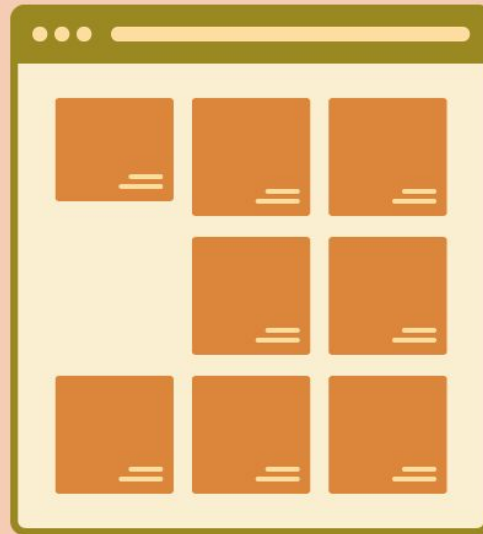
---

- **Block** boxes always appear **below** the previous block element. This is the “natural” or “static” flow of an HTML document when it gets rendered by a web browser.
- The **width of block** boxes is set automatically based on the width of its parent container.
- The default **height of block** boxes is based on the content it contains. When you narrow the browser window, the `<h1>` gets split over two lines, and its height adjusts accordingly.
- **Inline** boxes don’t affect **vertical spacing**. They’re not for determining layout—they’re for styling stuff inside of a block.
- The **width of inline** boxes is based on the content it contains, not the width of the parent element.

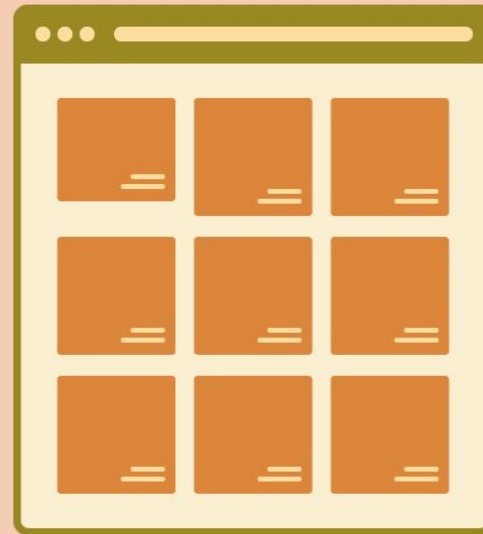


# inline-block for layout

---



float



inline-block





# Tips

---

- *Let your markup to be **super obvious**. Use **simple** and **understandable** class names and try to avoid abbreviations short forms of words*
- *For clickable elements use only **<button>** and for links tag **<a>***
- *Use **pixels** to define the size of the elements. Use percentage only when it's needed e.g. to define equal size blocks in a row*
- ***Don't** set a fixed **height** to the elements. **It's a hardcode**. Let your content to be flexible*
- *Use **equal** horizontal or vertical padding's values*
- *Choose **one side** of vertical margin and use it all the time*



# Bitbucket

---

- *Create a repository with the project which should be named by your name and surname with the following pattern: **name-surname***
- ***Create two branches 'empty' and 'master'***
- *In the empty branch always should be only **.gitignore***
- *Create new branches for new homeworks from **'empty'** branch*
- *Each homework should be in a separate branch and named with the following pattern: **homework-1***
- *Send merge request to the reviewer*



**Thank you for your  
attention**

**QUESTIONS?**