

floating boxes

the viewport

- The **viewport** is the window or viewing area that displays web pages. When the viewport is smaller than the web page, scroll bars should be available ([link](#))

the overflow

- The overflow property specifies what happens if content overflows an element's containing box.
- The following values could be applied to the overflow property:
 - visible
 - hidden
 - scroll
 - auto

[link](#)

the initial containing box

- The **initial containing box** has the entire width and height of your web page - including parts of the page that are outside the viewport ([link](#))

containing boxes

- A **containing box** is a box or block that contains other elements (descendant boxes). An element's containing box means "the containing box in which the element lives" ([link](#))

block elements and block boxes

- **Block elements** (or block boxes) are elements that occupy the entire specified width of their containing box, oblivious to the width of the enclosed content. For example, a paragraph of text ([link](#))

inline elements and inline boxes

- **Inline elements** are elements that do not form new blocks of content; the content is distributed in lines. For example, an emphasized piece of text within a paragraph ([link](#))

normal flow

- Normal flow is the way a document will display if you had no positioning or floating applied to elements. The content will flow in a text direction down the page, starting with the first element in your document and finishing with the last element in your document ([link](#))

static positioning

- A statically positioned box is one that is in normal flow

reminder: float positioning

- When you float an element, its box is then shifted to the left or right on the current line. The markup options are "float: [left](#)", "float: [right](#)" or "float: none"
- A floated box is laid out according to the normal flow, then taken out of the flow and shifted to the [left](#) or [right](#) as far as possible. Content can wrap down the right side of a left-floated box and down the left side of a right-floated box
- You can also put several floats beside each other.

[Example: three elements without any float applied](#)

[Example: three elements with "float: left" applied](#)

[Example: three elements with "float: right" applied](#)

where will a floated element move to?

- Floated boxes will move to the left or right **until their outer edge touches the containing block edge or the outer edge of another float.**

Example: "float: left" element against initial containing block

Example: "float: right" element against initial containing block

Example: "float: left" element against containing block

Example: "float: right" element against containing block

Example: "float: left" element against another floated box

- If there isn't enough horizontal room on the current line for the floated box, it will move downward, line by line, until a line has room for it.

Example: "float: left" element shifted down

do floated items need a width?

- You should **always set a width** on floated items (except if applied directly to an image - which has implicit width). W3C's Cascading Style Sheets, level 2, CSS2 Specifications states:

"A floated box must have an explicit width..."

- If no width is set, the results can be unpredictable. Theoretically, a floated element with an undefined width should shrink to the widest element within it. This could be a word, a sentence or even a single character - and results can vary from browser to browser.

Example: "float: left" with no width set

elements above and below floated elements

- Block level elements above a floated element will not be affected by it. However, elements below will wrap around the floated element:

[Example: "float: left" element with block level element above](#)

[Example: "float: left" element with element below](#)

borders, background images and background color

- While content will wrap around a floated element, border, background image and background color will extend underneath.

Example: "float: left" element with bordered block level element below

Example: "float: left" element with background-colored block level element below

- If you do not want elements below a floated element to wrap around it, you can apply the clear property to the following element using "clear: left", "clear: right" or "clear: both".

positioning

- The following properties can be used to position element box relative to its containing box:
 - top
 - left
 - right
 - bottom

relative positioning

- Relatively positioned elements are positioned within the normal flow and then moved. Elements that come after a relatively-positioned element behave as if the relatively-positioned element was still in its 'normal flow' position - leaving a gap for it.

[Example: relatively positioned content](#)

absolute positioning

- An absolute positioned box is moved out of the normal flow entirely.
[Example: absolutely positioned content](#)

fixed positioning

- Fixed positioned elements are moved out of the normal flow entirely - relative to the viewport. This means that they don't move if the page is scrolled.

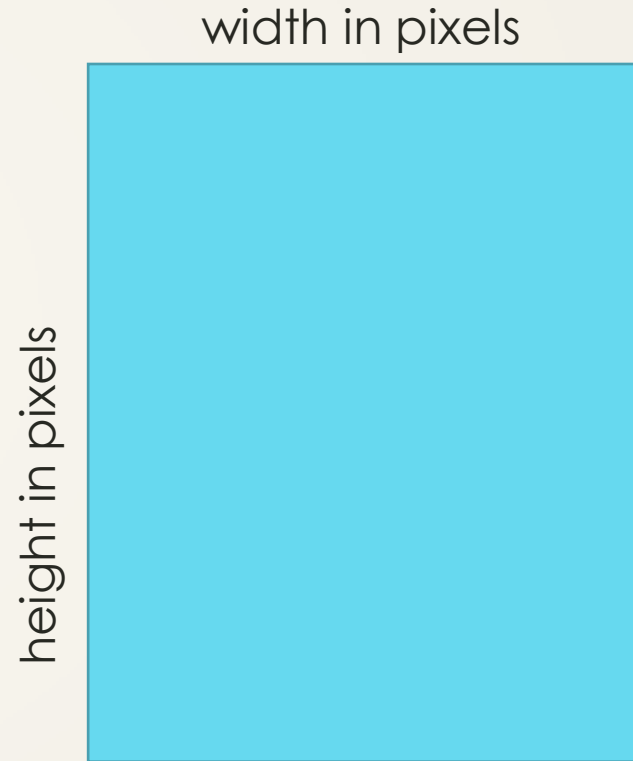
[Example: fixed positioned content](#)

why not absolute or fixed positioning

- **Floating** is used for most CSS layouts, not `<div>`s placed to the specific exact place on the viewport using absolute positioning
- The problem with absolute positioning that it relies on a rigid set of circumstances to lay out correctly: namely, that everybody uses the same browser and a screen with the same properties
- Web is not a rigid media as print is: you have no way to know what the size of a browser window and the user-agent default font size will be
- Absolutely positioned elements are removed from document flow, and they are positioned without regard to the surrounding elements

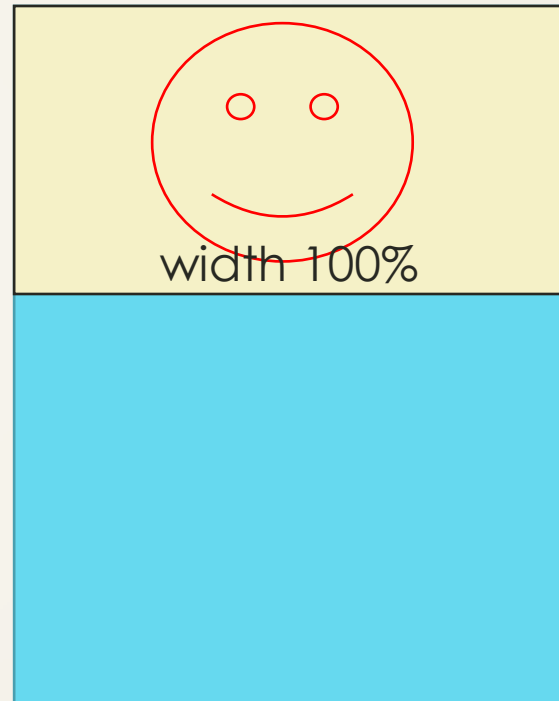
creating a list of floating boxes

- First, create `<div>` element
- Give it a class
- And style this class with a fixed width and height



creating a list of floating boxes

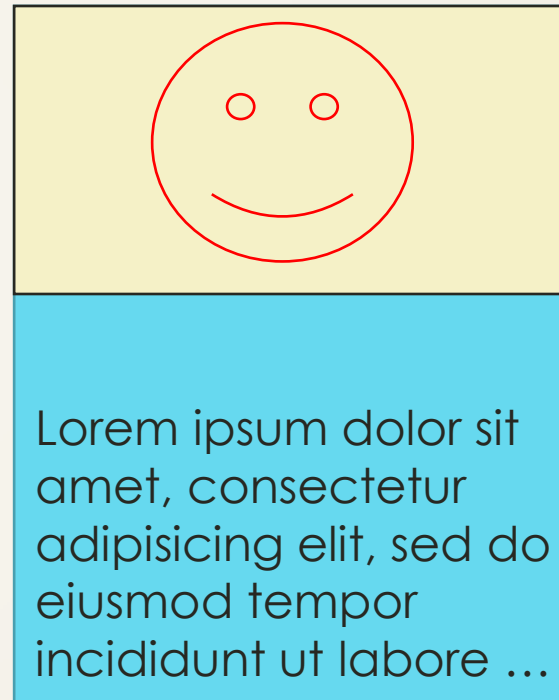
- Place an image inside the div, give the image width 100% of the containing box



creating a list of floating boxes

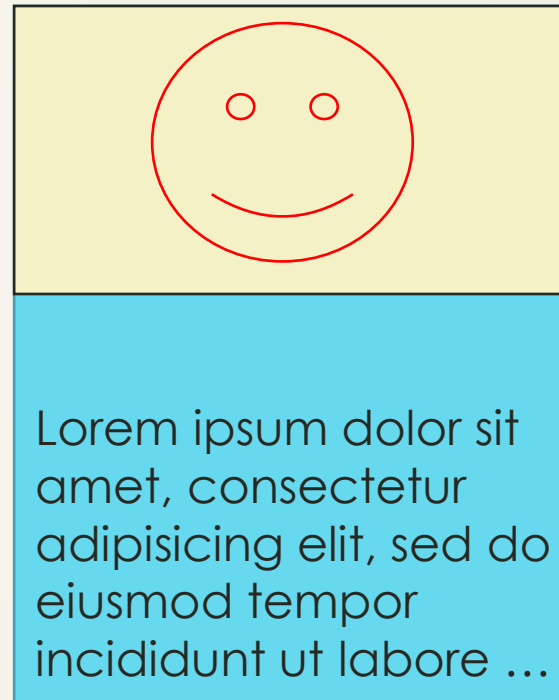
- Place a paragraph of text next to image inside the div, give the paragraph width 100% and give to the containing div:

overflow: hidden;



creating a list of floating boxes

- Now we can create a list of floating divs, by setting the property of each div to **float:left;**
- Use padding and margins to visually separate the divs



Kitten



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

Kitten



Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.