Positioning

Positioning allows you to take elements out of the normal document layout flow, and make them behave differently, for example sitting on top of one another, or always remaining in the same place inside the browser viewport

CSS allows us to position the HTML elements on the page creating required layouts

Different types of positioning within CSS

- floats
- inline blocks
- using position property

Document Flow

- By default, a block level element's content is 100% of the width of its parent element, and as tall as its content
- Inline elements are all tall as their content, and as wide as their content. Width or height cannot be set on inline elements
- The normal layout flow:
  - By default, block level elements are laid out vertically
  - Inline elements sit on the same line, as long as there is space

Exercise

- Create a page with 5 block elements (div, p, etc) and 5 inline elements (span, strong, em)
- Give the borders,margins and different background

Floats

the float property allows us to take an element, remove it from the normal flow of a page, and position it to the left or right of its parent element

the most used values are left and right

other values are:

- none (default)
- initial
- inherit (a property inherits its value from its parent element)

Floats

- when an element is floated, it will float all the way to the edge of its parent element. If there isn’t a parent element, the floated element will then float all the way to the edge of the page

- when an element is floated it is taken out of the flow and causes the width of the element to default to the width of the content

- float-based layouts have mostly been replaced with Flexbox in modern websites.

Clearing & Containing Floats

- Originally float property were design to allow content to wrap around images
- There are few things to be aware of:

  occasionally the proper styles will not render on an element that it is sitting next to or is a parent element of a floated element

  Often margin and padding property values aren’t interpreted correctly, causing them to blend into the floated element; other properties can be affected, too

  sometimes unwanted content begins to wrap around a floated element

  Floats May Change an Element’s Display Value !

Clearing Floats

the clear property, which accepts a few different values: the most commonly used values being left, right, and both

```
clear: left;

clear:right;

clear:both;
```

Positioning with Inline-Block

- the display property with inline-block value is mostly useful for laying out pages or for placing elements next to one another within a line
- accepts all box model properties, including height, width, padding, border, and margin

Positioning

- The position property identifies how an element is positioned on a page and whether or not it will appear within the normal flow of a document. This is used in conjunction with the box offset properties—top, right, bottom, and left—which identify exactly where an element will be positioned by moving elements in a number of different directions.

Position

- Static default, used to return the element to the default state
- Relative :

  the element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position
- Absolute

  The element is positioned relative to its first positioned (not static) ancestor element
- Fixed

  The element is positioned relative to the browser window, will not move when scrolling
- Sticky

  The element is positioned based on the user's scroll position

  Not supported in IE/Edge 15 or earlier. Supported in Safari from version 6.1 with a -webkit- prefix.

Introducing top, bottom, left, and right

- top, bottom, left, and right are used alongside position to specify exactly where to move the positioned element to (can take any units pixels, em, rems, %)

    top: 30px;

    left: 30px;

Relative position

- An element is moved in relative way to its parent element
- The space which it would occupy is preserved in the document flow
- Value of top: 30px; means the element is pushed down by 30px.

Absolute position

- the gap where the positioned element should be in the document no longer exists
- An absolutely positioned element no longer exists in the normal document layout flow
- it sits on its own layer separate
- Top, bottom, right, left properties specify the distance the element should be from each containing element's sides
- By default, containing element is <html>

Fixed and sticky

- fixed positioning fixes an element in place relative to the browser viewport itself
- There is a new positioning value available called position: sticky, support for which is not very widespread yet. This is basically a hybrid between relative and fixed position, which allows a positioned element to act like it is relatively positioned until it is scrolled to a certain threshold point (e.g. 10px from the top of the viewport), after which it becomes fixed.

Sources

- Feed the cat with the positioning:
- https://codepen.io/xumium/pen/EQOxmQ
- For examples of how positioning works:
- http://www.barelyfitz.com/screencast/html-training/css/positioning/
- https://learn.shayhowe.com/html-css/positioning-content/