Cascading Style Sheets

# Position

The position property specifies the type of positioning method used for an element. The top, right, bottom, and left properties determine the final location of positioned elements.

position:

1. static = Default value. Elements render in order, as they appear in the document flow
2. absolute = The element is positioned relative to its first positioned (not static) ancestor element
3. fixed = The element is positioned relative to the browser window
4. relative= The element is positioned relative to its normal position, so "left:20px" adds 20 pixels to the element's LEFT position
5. sticky = The element is positioned based on the user's scroll position
6. Initial = Sets this property to its default value.
7. Inherit= Inherits this property from its parent element.

Code:

position: absolute;  
border: 1px solid red;  
top: 70px;  
right: 15px;

# Box

**box-sizing:**

1. content-box;
2. border-box

**code:** box-sizing: content-box

# flex

**justify-content:**

1. flex-start -- Default value. Items are positioned at the beginning of the container.
2. flex-end -- Items are positioned at the end of the container .
3. center -- Items are positioned in the center of the container.
4. space-between -- Items will have space between them.
5. space-around -- Items will have space before, between, and after them.
6. space-evenly -- Items will have equal space around them.
7. Initial -- Sets this property to its default value.
8. Inherit -- Inherits this property from its parent element.

**code:** justify-content: flex-start

# display

1. **Inline:** Displays an element as an inline element (like <span>). Any height and width properties will have no effect
2. **block:** Displays an element as a block element (like <p>). It starts on a new line, and takes up the whole width
3. **contents:** Makes the container disappear, making the child elements children of the element the next level up in the DOM
4. **flex:** Displays an element as a block-level flex container
5. **Grid:** Displays an element as a block-level grid container
6. **inline-block:** Displays an element as an inline-level block container. The element itself is formatted as an inline element, but you can apply height and width values
7. **inline-flex:** Displays an element as an inline-level flex container
8. **inline-grid:** Displays an element as an inline-level grid container
9. **inline-table:** The element is displayed as an inline-level table
10. **list-item:** Let the element behave like a <li> element
11. **run-in:** Displays an element as either block or inline, depending on context
12. **table:** Let the element behave like a <table> element
13. **table-caption:** Let the element behave like a <caption> element
14. **table-column-group:** Let the element behave like a <colgroup> element
15. **table-header-group:** Let the element behave like a <thead> element
16. **table-footer-group:** Let the element behave like a <tfoot> element
17. **table-row-group**: Let the element behave like a <tbody> element
18. **table-cell:** Let the element behave like a <td> element
19. **table-column:** Let the element behave like a <col> element
20. **table-row:** Let the element behave like a <tr> element
21. **none:** The element is completely removed
22. **initial:** Sets this property to its default value.
23. **Inherit:** Inherits this property from its parent element.

**Code:** display : inline