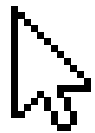
# Computer Science :
## HTML & CSS

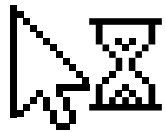


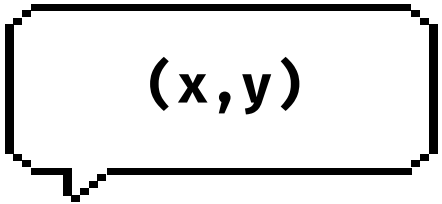
Week 4: Layout Techniques

# Layout Techniques

HTML without CSS will display information from left to right and top to bottom.

CSS provides properties that allow you to set up the lay out of your page





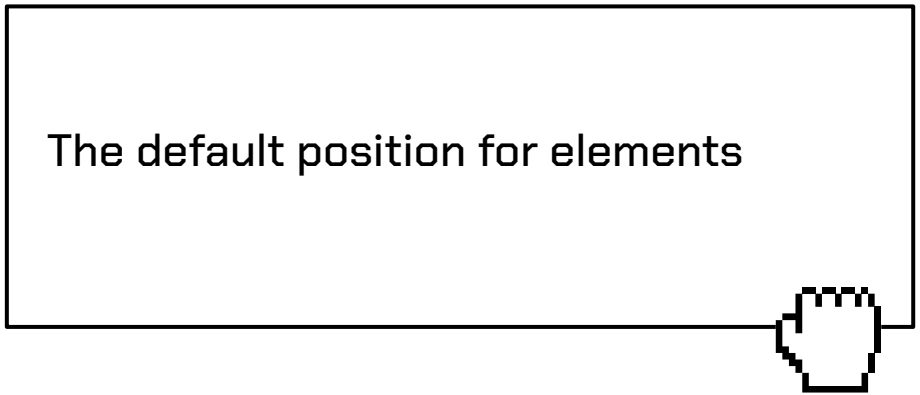

# 01

# Positions

Location and spacing for elements on a page

# position: static

Example: {
position: static;
}

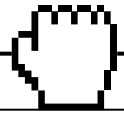


# position: relative

```
Example: {
        position: relative;
        top: 120px;
        }
```

Can position elements relative to its default page

Paired with offset properties
top - moves element down from top
bottom - moves element up from bottom
left - moves element to the right
right - moves element to the left



# z-index

Example: {

position: relative;

z-index: 1;

}

Controls how forward or back (depth) an element appears when it overlaps relative to other elements.

Can be used with relative but not with

# position: absolute

```
Example: {
        position: absolute;
        width: 100%;
        }
```

The other elements in the page will function as if the element is not there.

The absolute element will position itself relative to its closest element. Position can be modified with offset properties.

# position: fixed

Example: {
position: fixed;
}

Fixed an elements to a specific position on the page and will remain on the page when scrolling

Position defined by offset property
top - moves element down from top
bottom - moves element up from bottom
left - moves element to the right
right - moves element to the left

# position: sticky

```
Example: {
        position: sticky;
        top: 100px;
        }
```

Similar to fixed, when scrolling it will remain on the page but the defined offset properties make it stick to that position on the page.

Ex: the image will stick to the the position until you reach 100px from the top then it will be free to scroll

# 02

## Display

# display: block

Example: <h1><h2>

Block elements always start a new line. They modify the full width. Ex: <h1>, 

Some examples of block elements:
https://www.w3schools.com/html/html_blocks.asp

# display: inline

Example:

HTML <inline element>Welcome</inline element>

```
CSS {
        display: inline
         }
```

Modifies their particular content and only takes up as much space as its contents take up. It does not start a new line

Some examples of inline elements:
https://www.w3schools.com/html/html_blocks.asp

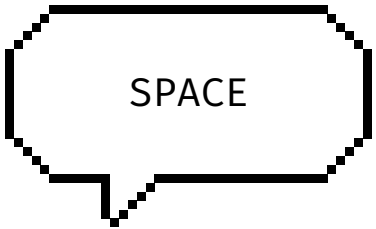
# display: inline-block

Example: {

display: inline-block;

}

Combining both elements inline-block features can appear next to each other and can be defined by width and height .




# 03

# Others

Float and Clear

# float


Example: {

float: right;

}

Can move elements as far left or right as possible in its content container. Ex: wrapping text around an image. Must have a specified width otherwise there will be no visible results.

Float values:

right - moves elements as far right as possible

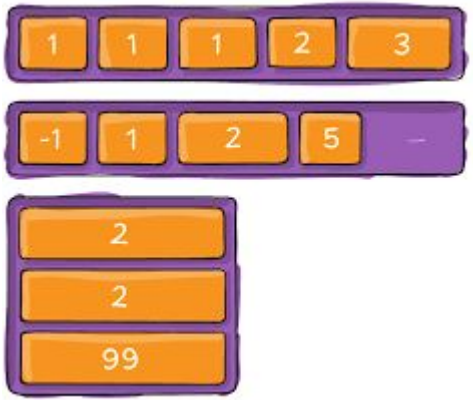
left - moves elements as far left as possible

# clear

```
Example: {
            clear: right;
            }
```

Controls the flow next to floated elements to avoid overcrowding between elements in the same container and specifies what should happen next to the floated elements

Values:
left - left side will not touch any other elements
right - right side will not touch any other elements
both - neither side of elements can touch other elements
none - elements can touch either side




# 04

# Flexbox

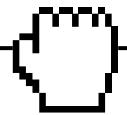
Layout that distributes space between items and allows for several types of alignments

# 1. Flexbox container

Example:

```
<div class= “flexbox_container”>
        <div>1</div>
        <div>2</div>
        <div>3</div>
</div>
```

Create a flexbox container with items inside . Set a overall class



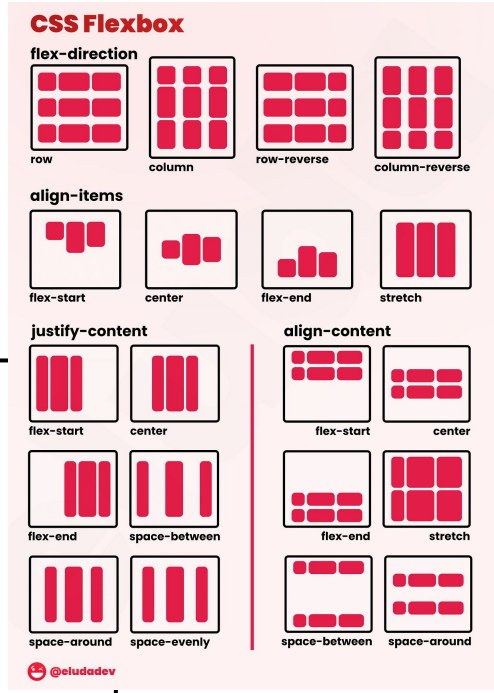
# 2. Flex display properties

Example: {

```
display: flex;
flex-direction: row;
}
```

Set display to flex so the container can be modified.

Then you define specific flex properties:
Flex Display Properties



# 3. Flex Item properties

Example: <div style= “[item property]”> 1

Now you can modify the individual elements with flex item properties in HTML code using style= “”

Flex Item Properties