Advanced CSS Continued

Objectives

01	Web Fonts in CSS	Understanding and Analyzing Fonts and Web Fonts in CSS.
02	Tables & Lists	Understanding and Analyzing Tables and Lists.
03	CSS Rules	Understanding and Analyzing !important Rules of CSS.
04	Display Property	Understanding and Analyzing Display Property in CSS.
05	CSS Flexbox	Understanding the Concept of CSS Flexbox
06	Position Property	Understanding and Analyzing Position Property in CSS.

Objectives

07	Float Property & Clear Both	Understanding and Analyzing Float Property and Clear Both.
08	Transform Property	Understanding the Concept of Transform Property in CSS
09	Transition Property	Understanding the Concept of Transition Property in CSS

CSS Fonts

Selecting and Decorating Fonts

- Generic Font Family:
  - Serif fonts: have a small stroke at the edges of each letter.
  - Sans-serif fonts: have clean lines (no small strokes attached).
  - Monospace fonts: all the letters have the same fixed width.
  - Cursive fonts: imitate human handwriting.
  - Fantasy fonts: decorative/playful fonts.

- Other important concepts and properties of CSS Font:
  - Fallback fonts, Web safe fonts
  - Google fonts, Multiple Google fonts
  - Font-style, Font-weight, Font-size, Font-family
  - Font shorthand property, Font pairings

```
<head>
<link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Audiowid
e|Sofia|Trirong&effect=neon|outline|emboss|shadow-mult
iple">
<style>
h1.a {font-family: "Audiowide", sans-serif;}
h1.b {font-family: "Sofia", sans-serif;}
h1.c {font-family: "Trirong", serif;}
.p1 {font-family: "Times New Roman", Times, serif;}
.p2 {font-family: Arial, Helvetica, sans-serif;}
.p3 {font-family: "Lucida Console", "Courier New",
monospace;}
.p4 {font: italic small-caps bold 12px/30px Georgia,
serif;}
</style>
</head>
<body><h1 class="font-effect-neon">See</h1></body>
```

CSS Web Fonts

CSS @font-face Rule

- Different Font Formats:
  - TrueType Fonts (TTF)
  - OpenType Fonts (OTF)
  - Web Open Font Format (WOFF / WOFF 2.0)
  - SVG Fonts/Shapes
  - Embedded OpenType fonts (EOT)

- Advantages of Web Fonts:
  - Allow to use fonts that are not installed on the user's computer
  - Automatically downloaded to the user when needed
  - May contain descriptors

```
<head>
<style>
@font-face {
 font-family: myWebFont;
 src: url(sansation_bold.woff);
 font-weight: bold;
}

div {
 font-family: myWebFont;
}
</style>
</head>
```

CSS List

Listing with Number, Letter, Bullet or Image Markers

- Unordered Lists <ul>:
  - list-style-type: none, circle, square, etc.
  - list-style-image: url('sqpurple.gif')
- Ordered Lists <ol>:
  - list-style-type: upper-roman,lower-alpha,etc.
  - list-style-image: url('sqpurple.gif')
- For both the types:
  - list-style-position: outside, inside
  - list-style: square inside url("some.gif")

```
<style>
ol {
   list-style-type: upper-roman;
   list-style-position: outside;
   /*list-style-image: url('some.png');*/
}

ol li {
 background: #ffe5e5;
 padding: 5px;
 margin-left: 35px;
}

ul {list-style: square inside url("some.gif");}
</style>
```

CSS Table

Tabulating Data in a Row-Column Format

- Commonly used HTML Tags for a Table:
  - <table>, <tr>, <td>, <th>
  - <thead>, <tbody>, <tfoot>
  - <caption>, <col>, <colgroup>

- Important CSS Properties:
  - border-spacing, border-collapse,
  - th:text-align, td:vertical-align
  - tr:nth-child(), tr:hover

- Responsive:
  - Means automatic resize
  - For a table to be responsive:
    overflow-x:auto could be set to the container element

```
<style>
table, td, th {
 border: 3px solid red;
 border-collapse: collapse;
 /*border-collapse: separate;*/
 /*border-spacing: 10px;*/
}

tr:hover {background-color: yellow;}
tr:nth-child(even) {background-color: #f2f2f2;}

/*
   for a responsive table
   the container element style could be:
*/
div {overflow-x:auto;}
</style>
```

CSS !important

Overriding Style Rule

- The !important rule is used to override all previous style property rules for that specific element
- The !important rule in CSS is used to add more importance to a property-value than normal.
- To override an !important rule, another !important rule with the same or higher specificity has to be declared. (Though it is not a good practice)
- Syntax:
  - selector {

```
            property : value !important;
    }
```

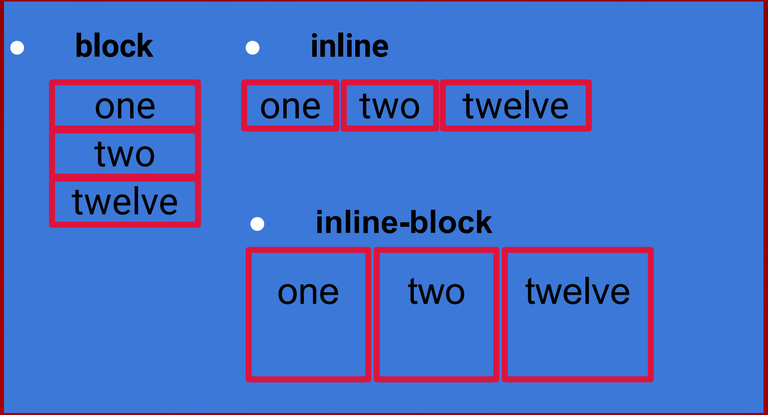
```
<head>
  <style>
    #p_id {color: blue;}
    .p_class {color: gray;}
    p {color: red !important;}
  </style>
  <style>
    #p_id {color: blue !important;}
    .p_class {color: gray !important;}
    p {color: red !important;}
  </style>
</head>
<body>
  <p>paragraph</p>
  <p class="p_class">paragraph</p>
  <p id="p_id">paragraph</p>
</body>
```

CSS Display Property

Display Behavior of an Element

- Default display properties of elements are:
  - display : block: starts on a new line and takes up the full width available (<div>, <h1> etc.)
  - display : inline: doesn't start on a new line and takes up only necessary width (<span>, <a> etc.)

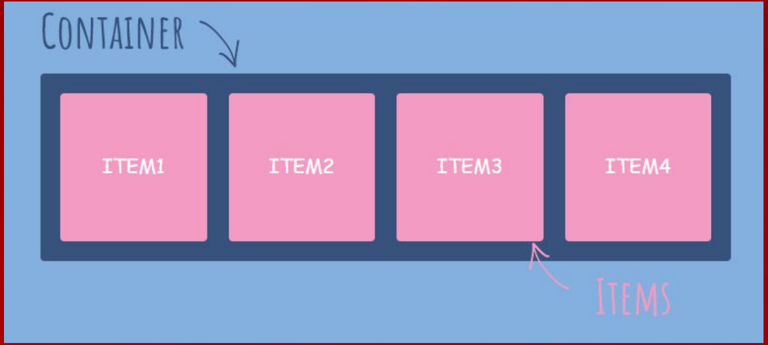
- Other common display properties:
  - display : inline-block: allows to set a width and height on the element without starting on a new line.
  - display : none: hides element without holding space. visibility : hidden: hides element but holds space.



CSS Flexbox

Organizing Display Behaviour as a Row or as a Column

- Another useful display property is flex:
  - The CSS flexbox provides efficient way to layout, align and distribute space among items
  - Automatically scale elements (alter height or width) so that they fill the available space
  - Automatically shrink or grow elements to make them fit into the container and prevent overflow
  - display : flex: setting display property of the container element to flex
- Flex container properties:
  - align-content, align-items, flex-direction, flex-flow, flex-wrap, justify-content
- Flex item properties:
  - align-self, flex, flex-basis, flex-grow, order, flex-shrink

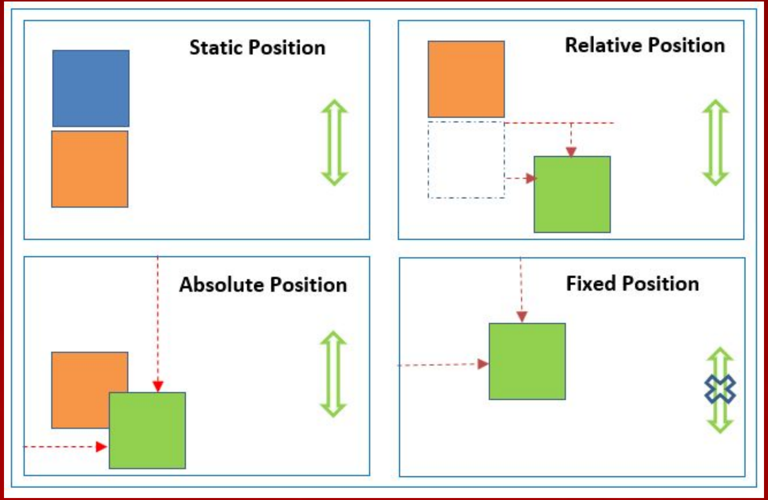


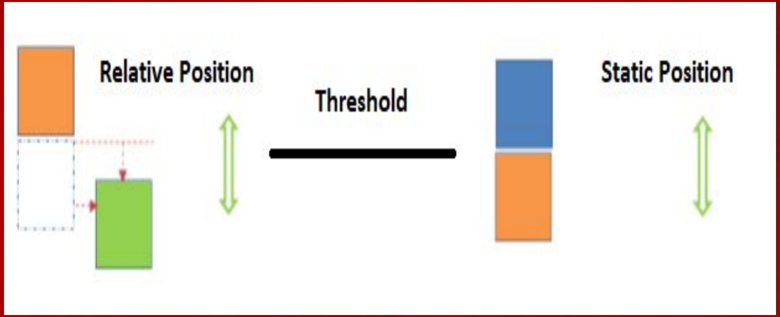
CSS Position Property

Positioning Elements

- Position is set using TRBL (top, right, bottom & left)

- There are five different position values:
  - static: default position (no effect of TRBL)
  - relative: positioned relative to its normal position
  - absolute: positioned relative to its relative or absolutely positioned parent, else document body
  - fixed: positioned at a fixed place relative to the document body and no effect of page scrolling
  - sticky: relative until a specified threshold, after that point it holds a static position
- Syntax:
  - position:static|relative|absolute|fixed|sticky
- z-index: order of overlapping elements (z-index:value)

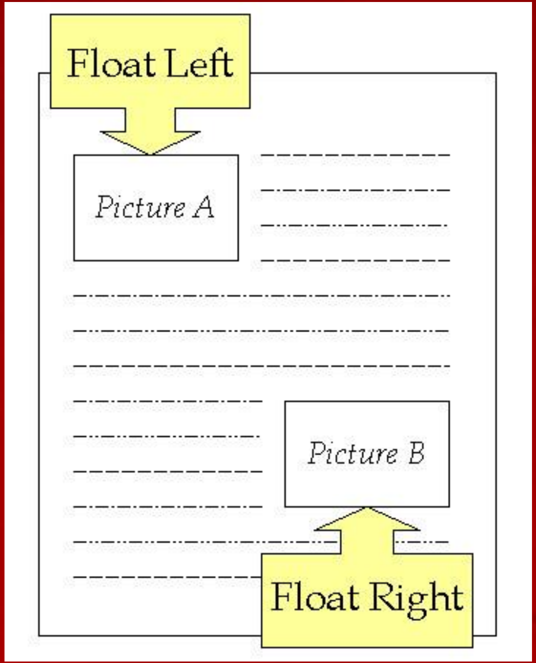




CSS Float Property

Specifies how an Element should Float

- Float property makes an element float to the left or right side inside its container, break its normal flow but still be placed inside its parent only
- Clear property is used to remove the float effect
- Syntax:
  - float:left|right|none
  - clear:left|right|both



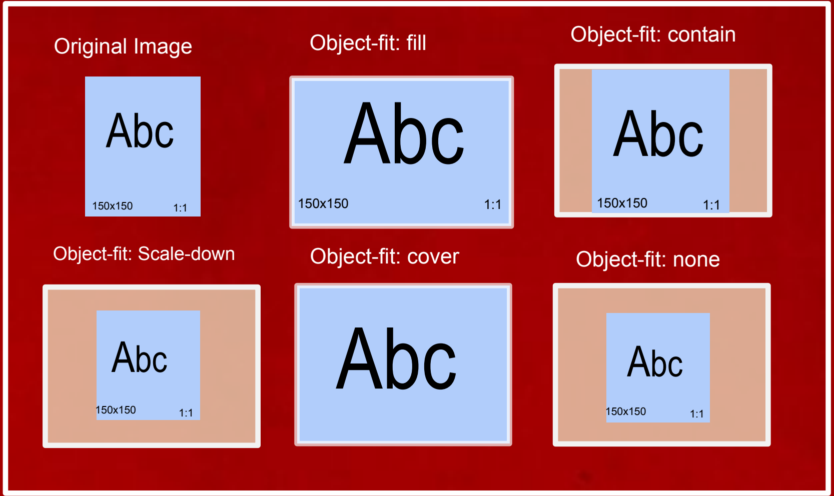
CSS Object-fit Property

Ways to Resize Content to Fit its Container

- Object-fit property makes the content fills its container in various ways:

- Five different values of object-fit:
  - fill: default, resized to fill the container dimension
    - (could be stretched or squeezed to fit)
  - contain: resized to fit the container dimension
    - (aspect ratio preserved)
  - cover: resized to fill the container dimension
    - (aspect ratio preserved, could be clipped to fit)
  - none: not resized
  - scale-down: scaled down to the smallest version of none or contain
- Syntax:

  object-fit:fill|contain|cover|none|scale-down



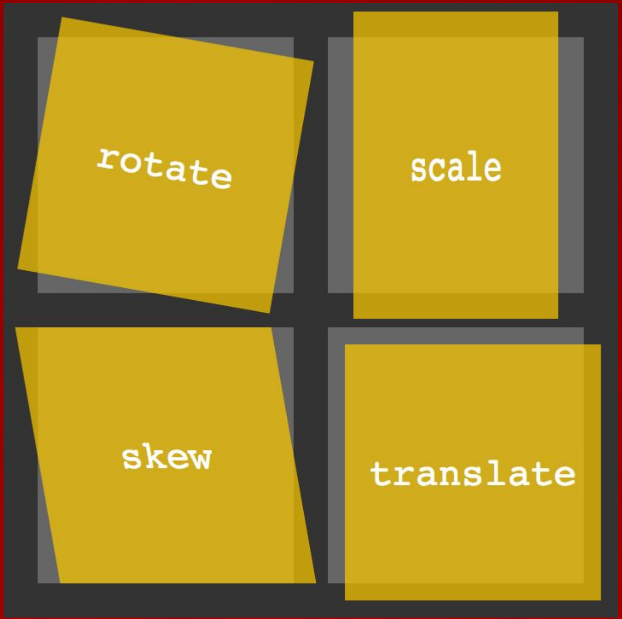
CSS Transform Property (2D/3D)

How to move, rotate, scale, and skew elements

- The transform property allows to move, rotate, scale, and skew elements.

- Different transformation methods (2D):
  - rotate(): transform:rotate(20deg)
  - [rotateX(), rotateY(), rotateZ() (for 3D)]:
  - translate(): transform:translate(50px,100px)
  - scaleX(): transform:scaleX(2)
  - scaleY(): transform:scaleY(3)
  - scale(): transform:scale(2,3)
  - skewX(): transform:skewX(20deg)
  - skewY(): transform:skewY(40deg)
  - skew(): transform:skew(20deg,40deg)
  - matrix(): transform:matrix(1,-0.3,0,1,0,0)

matrix(scaleX,skewY,skewX,scaleY,translateX,translateY)



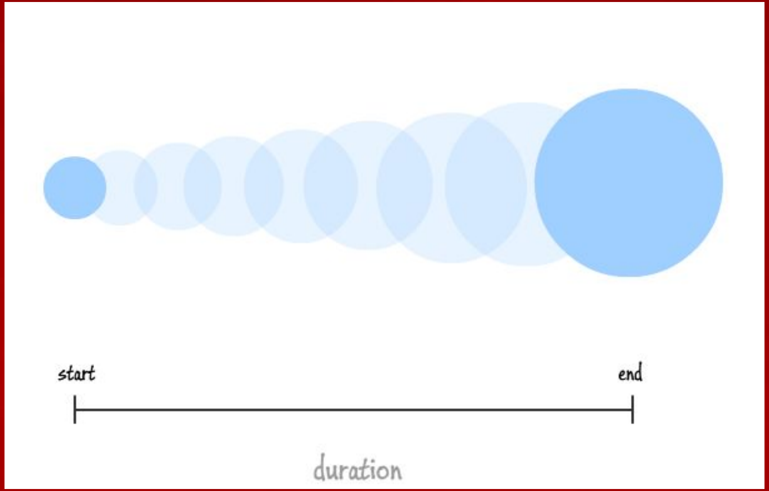
CSS Transition Property

Smooth Changes in Property Values

- transitions allows to change property values smoothly, over a given duration.

- Different transition properties:
  - transition-property:width|height|transform
  - transition-duration:2s
  - transition-timing-function: ease(default)|linear|ease-in|ease-out| ease-in-out|cubic-bezier(n,n,n,n)
  - transition-delay:1s

  - transition:width 2s linear 1s
    transition: transition-property transition-duration
    transition-timing-function transition-delay



Thank You