



CSS & CSS3

w i t h m o d u l e s

HANDS ON WORKSHOP

By Vijay Shivakumar



Welcome to the workshop

Requirements for the training

VS Code : <https://code.visualstudio.com>

Browser : Chrome, Safari, Firefox

Notepad / Pen / Pencil

Phones on silent mode



Introduction

About me

Vijay Shivakumar
Designer | Developer | Trainer



Certified Expert

Training & Consultation of
Contemporary Web Technologies and Adobe products from past 20+ years

Vijay Shivakumar


DOTJS.in



Introduction

About You

Designer

Developer

Tester

Content Writer

Prior Knowledge of HTML CSS and JS



Introduction

About The Workshop

What is CSS ?

Why is it required ?

The past, present and future

Usages, limitations and its workaround



What is CSS ?

The Cascading Style Sheets is a language that's used to write formatting instructions

how webpage content should 'look'— in terms of : Layout & style



Why is it required ?

Data without format and indent

Data : 16tablet5600017laptop21000018mobile52500



Why is it required ?

Data with alignment and indents

Data, Alignment & Indents:

16	tablet	56000
17	laptop	210000
18	mobile	52500



Why is it required ?

Data with alignment, indents and formatting

Data, Alignment, Indents & Formatting :

<u>SI #</u>	<u>Product</u>	<u>Price in INR</u>
16	Tablet	₹ 56000
17	Laptop	₹ 210000
18	Mobile	₹ 52500



Why to use CSS ?

Advantages

Separation of content and presentation

CSS provided in a file or a block that is separate to the (content) HTML that can be more easily be updated.

Smaller webpage file sizes I personally have seen file size reduction up to 50%

Improved webpage download speed

Can be cached for re use

Save bandwidth

Improve rendering performance and speed



Why to use CSS ?

Advantages More...

Streamlined maintenance

Reduction in errors

Easy up gradation

Different devices different presentations For web For print For mobile etc...



Where can CSS be used ?

Where to declare the styles

Inline Style :

```
<h1 style="font-family: Arial">Welcome!</h1>
```

Embedded Style :

```
<style>
    h1 { font-family: Arial; }
</style>
```

```
<h1>Welcome!</h1>
```



Where can CSS be used ?

Where to declare the styles

Linked Style :

```
<link rel="stylesheet" href="style.css" />
```

```
<style type="text/css">
```

```
@import "style.css"; or @import url ("style.css")
```

```
</style>
```



CSS code anatomy

What do you call them ?

```
p {  
  color: #ff0000;  
  font-size : 24px;  
}
```

```
selector {
```

```
  property: value ; - declaration
```

```
  property: value ; - declaration
```

```
}
```

- declaration block

- style rule



Selectors and Combinators



Selectors & Combinators

Identify specific HTML elements as targets for CSS styles

Selectors Combinators

Element / Tag	Descendent	Lang Pseudo
Class	Adjacent Sibling	Child
Id	General Sibling	Selector Groups / Selector List
Attribute	Pseudo Class	
Universal	Dynamic Pseudo Class	
	Pseudo Element	



Selectors & Combinators

Identify specific HTML elements as targets for CSS styles

Element / Tag	<code>div{ prop : val }</code> all <code><div></code> elements
Class	<code>.box{ prop : val }</code> all elements with class attribute "box"
Id	<code>#play{ prop : val }</code> the element with "id" attribute "play"
Attribute	More on Attribute Selectors
Universal	Matches every element
Selector Groups	<code>div, .box, .play { prop : val }</code> all matching elements
Adjacent Sibling	<code>+</code> Matching element that appear immediately after
General Sibling	<code>~</code> Matching element that appear any where after



Selectors & Combinators

Identify specific HTML elements as targets for CSS styles

Descendent

`div span{ prop : val }`
all ``s that are descendants of a `<div>`

Pseudo Element

`div:before{ prop : val }` All elements that match pseudo-element

Pseudo Class

`div:hover{ prop : val }` All divs with pseudo-class hover

Dynamic Pseudo Class

Dynamic Input Elements that matches `:hover` `:focus` `:active`

Lang Pseudo Class

`p:lang(en){ prop : val; }` Element that match `:lang` declaration

Child

[More on Child Selectors](#)



Child Selectors

Logical selectors introduced in CSS3

:first-child	Selects an element that is the first child of its parent.
:last-child	Selects an element that is the last child of its parent.
:only-child	Selects an element that is the only child of its parent & no siblings.
:first-of-type	Selects an element that is the first descendent or child of its type.
:last-of-type	Selects an element that is the last descendent or child of its type.
:only-of-type	Selects an element that is the only descendent or child of its type.
:nth-child()	Selects the nth-child of its parent using the value between the parentheses number / odd / even / $2n$ / $2n + 3$ / $2n - 3$.



Child Selectors

Logical selectors introduced in CSS3

:nth-of-type

Selects the nth sibling of its type will select from its parent number / odd / even / $2n$

:nth-last-child

Selects the nth-child of its parent counting from the last one.

:nth-last-of-type

Selects the nth sibling of its type counting from the last one.



Pseudo Element Selectors

Logical selectors introduced in CSS3

::selection	User selected content
:not()	Negation selector
:is()	:is(simple selectors group) target_element { prop : value }
:where()	Similar to :is() but maintains the specificity to 0
:empty	Selects an element with no children
>	Child selector



Attribute Selectors

Select an element based on its attribute value

[attr]	<div attr> With attribute attr
[attr='val']	<div attr="val"> Where attribute attr has value val
[attr^='val']	<div attr="val1 val2"> Where attr's value begins with val
[attr\$='val']	<div attr="sth aval"> Where the attr's value ends with val
[attr*='val']	<div attr="somevalhere"> Where attr contains val anywhere
[attr~='val']	<div attr="val val2 val3"> Where val appears in whitespace-separated list of attr
[attr = 'val']	<div attr="val-sth etc"> Where attr's value is exactly val, or starts with val and immediately followed by -



CSS Features



Types of Elements

Basic classification of element

Inline

Block

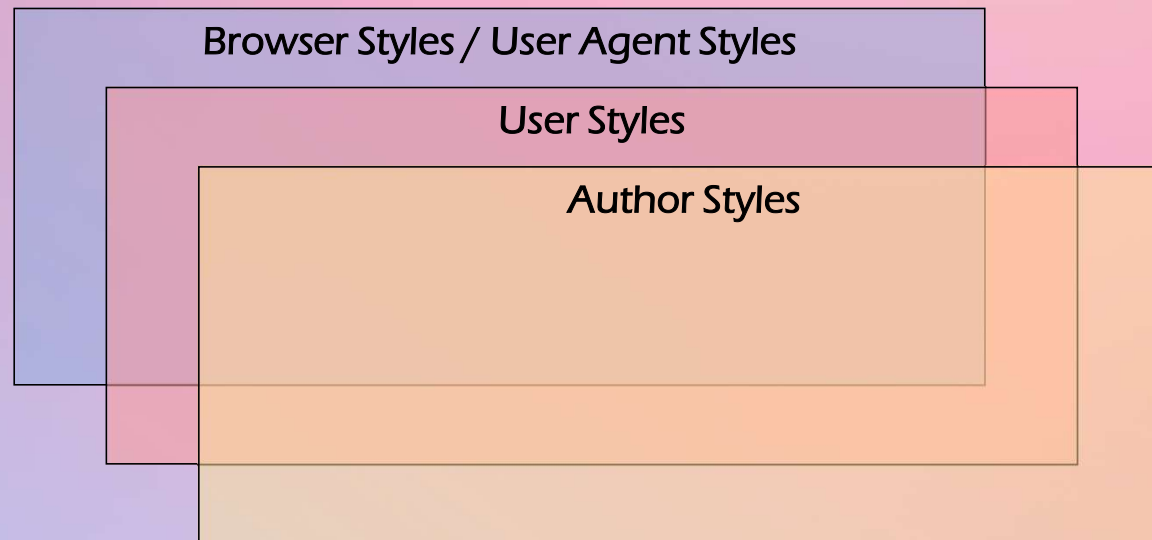
Inline-block

Table



Sources of Styles

Hierarchy of style sheets applied to a page





Specificity

Values for selectors

Inline	1000	 	1.0.0.0
Id	100	 	1.0.0
Class / Pseudo Class / Attribute	10	 	1.0
Element / Pseudo Element	1	 	1



!important

Override Specificity

The order of importance

User styles flagged as !important

Author styles flagged as !important

Author styles

User styles

Styles applied by the browser/user agent



Presentation properties and Values



CSS Units

Units of measurement

px | pixel value

% | percentage

em | width of em

rem | relative width of em

vw | view port width

vh | view port hieght

vmin | view port minimum

vmax | view port maximum



CSS Color

Color values

color name | “red”

hex | pixel value

hash | percentage

rgb

rgba

hsl

hsla



Border

Border to elements

border-top-width : 1px | border-right-width : 1px | border-bottom-width : 1px | border-left-width : 1px

border-width : thin, thick, medium sizes are allowed

border : 2px solid red - short form



Margin

Space between elements

margin-top: 5px;

margin-right: 5px;

margin-bottom: 5px;

margin-left: 5px;

margin:<top> <right> <bottom> <left>

margin:<left> <right> <top | bottom>

margin:<left | right> < top | bottom>

margin:<top | right | bottom | left>



Padding

Space between content and container

padding-top: 5px;

padding-right: 5px;

padding-bottom: 5px;

padding-left: 5px;

padding:<top> <right> <bottom> <left>

padding:<left> <right> <top | bottom>

padding:<left | right> < top | bottom>

padding :<top | right | bottom | left>

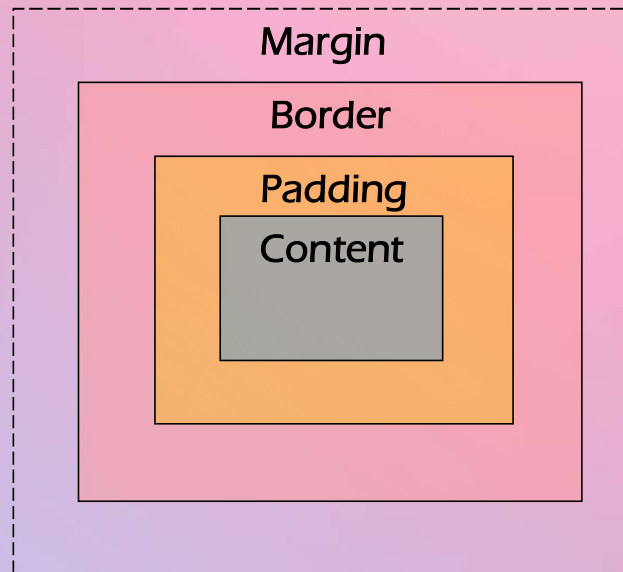


Layouts



CSS box model

How to calculate the computed width and height





Position Layout

Precise layout with position

- position: static** - default layout property
- position: fixed** - location is calculated from view port
- position: absolute** - location is calculated from relative parent / view port
- position: relative** - provides location to absolute children
- position: inherit** - uses the parent's (container's) position



Float Layout

Simple Layout with floats

float: left - floats to the left of siblings

float: right - floats to the right of siblings

clear : left - clears the siblings floating to left

clear : right - clears the sibling floating to right

clear : both - clears the sibling floating left or right



CSS3 Flex box module



Flex Box Module

Flexible layouts

Highly optimised for UI design, they are flexible.

They can be laid out in any direction both horizontally and vertically in any order and can be flexible.



Flex Box History

How we got here...

An implementation by Mozilla in XUL

July 2009 Working Draft (`display: box;`)

March 2011 Working Draft (`display: flexbox;`)

November 2011 Working Draft (`display: flexbox;`)

March 2012 Working Draft (`display: flexbox;`)

June 2012 Working Draft (`display: flex;`)

September 2012 Candidate Recommendation (`display: flex;`)



Flex Box Properties

Properties on parent

display : flex

flex-direction : row | row-reverse | column | column-reverse;

flex-wrap: nowrap | wrap | wrap-reverse;

flex-flow: shorthand for flex direction and flex wrap

justify-content: flex-start | flex-end | center | space-between |

space-around | space-evenly | start | end | left | right;

align-items: stretch | flex-start | flex-end | center | baseline | start | end | self-start | self-end ;

align-content: normal | flex-start | flex-end | center | space-between | space-around | space-evenly | stretch | start | end | baseline ;

gap : single value / gap : multiple values / row-gap : px value / column-gap : px value



Flex Box Properties

Properties on child element

flex-basis: **auto** (value in px);

flex-grow: **0**; (value in number)

flex-shrink: **1**; (value in number)

flex: **none** ; / flex-grow flex-shrink flex-basis

order: **0**;

align-self: **auto** | flex-start | flex-end | center | baseline | stretch;

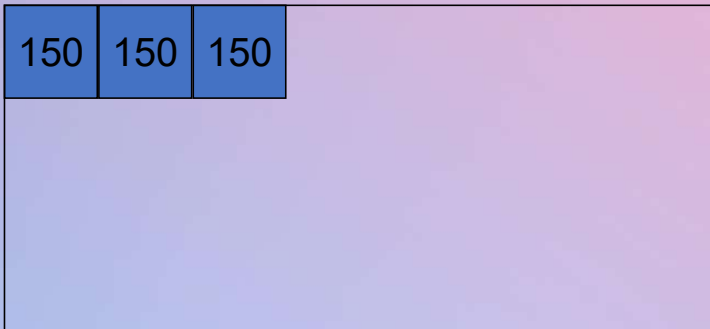


Flex box

flex-basis

flex-basis : auto(default) | width in px, percent, rem, em

With 150px it will make the element to have a basic size of 150px
You can think of this as the natural width of the element until other layout properties affect it
which is the max width they can grow.





CSS3 Grid module



Grid Module

Grids for layouts

CSS Grid (Grid Layout Module level 1)

supported by all major browser versions

CSS Sub Grid (Grid Layout Module level 2)

supported partially by modern browsers



CSS Grids Module

Grids for layout

Properties on parent element

display : grid;

Properties on nested elements



CSS3 Modules

New in CSS3

Flex box

Grid

Box Sizing

Border radius

Border Image

Box, text shadow

Transparency

Background Images

Multi column

Media queries

Font embedding

Transformations

Animation



CSS3 Box Sizing Module



Sizing Module

Box sizing

content-box - Makes the selected element to take the content width within the assigned width. (default)

border-box - Makes the selection to restrict the padding and the border within the width of the element, and wont grow outside.

```
selector{  
    box-sizing: border-box;  
}  
*, *:before, *:after{  
    box-sizing : border-box;  
}
```



Sizing Module

Box sizing

border-box: The painted content is clipped to the border box. (Default value)

content-box: The painted content is clipped to the content box.

fill-box: The painted content is clipped to the object bounding box.

margin-box: The painted content is clipped to the margin box.

padding-box: The painted content is clipped to the padding box.

stroke-box: The painted content is clipped to the stroke bounding box.

view-box: Uses the nearest SVG viewport as reference box. If a viewBox attribute is specified for the SVG viewport creating element:

- The reference box is positioned at the origin of the coordinate system established by the viewBox attribute.
- The dimension of the reference box is set to the width and height values of the viewBox attribute.

no-clip: The painted content is not clipped.

initial: Applies the property's default setting, which is border-box.

inherit: Adopts the mask-clip value of the parent.

unset: Removes the current mask-clip from the element.



CSS3 Nesting Selector Module



Nesting Selector Module

Nest your selectors

```
.x {
```

```
    /* styles for element with class="x" */
```

```
    .y {
```

```
        /* styles for element with class="y" which is a descendant of class="x" */
```

```
    }
```

```
    &.y {
```

```
        /* styles for element with class="x y" */
```

```
    }
```

```
    & + p {
```

```
        /* styles for element p which is adjacent to class="x" */
```

```
    }
```

```
}
```



CSS3 Border Radius Module



Details

Border Radius

No more boxes



CSS3 Border Image Module



Details

Border Image

The scale 9 images



CSS3 Box / Text Shadow Module



Details

Box – Text Shadow

Single and multiple



CSS3 Transparency Module



Details

Opacity

Transparency and Opacity



CSS3 Gradient Module



Details

Gradient

Lets add more colors



CSS3 Background Image Module



Details

Background Images

composition



CSS3 Multi Column Module



Details

Multi column Text Layout composition



CSS3 @mediaquery



CSS3 Media Query

@media-query : query device for its properties

Ethan Marcotte coined the term responsive web design (RWD) in a May 2010 article in A List Apart.



CSS3 Media Query

What can we query for ?

Media Queries allow you to style elements for specific devices as per their

Resolution

device-width

device-height

resolution

Aspect ratio

Orientation

orientation : landscape / portrait

Color index



CSS3 @FontFace for font embedding



@font-face

Support for custom fonts

@font-face was created by Microsoft

EOT : Embedded Open Type.

A proprietary file standard supported only by IE8 and older browsers.

OTF / TTF : OpenType Font and TrueType Font.

An initial standardized version for older browsers.

SVG : Scalable Vector Graphics

This format is the only allowed by Safari for iOS below version 4.1

WOFF / WOFF2 : Web Open Font Format

Developed by Mozilla, compressed ,loads faster, includes metadata & license info.

Future web standard followed by all browsers..



@font-face

Support for custom fonts

```
@font-face {  
  font-family: 'NewFontName';  
  src: url('webfont.eot'); /* for IE9 */  
  src: local('local avalialbe name'),  
    url('webfont.eot?#iefix') format('embedded-opentype'), /* IE6-IE8 */  
    url('webfont.woff2') format('woff2'), /* Browsers that support the latest features */  
    url('webfont.woff') format('woff'), /* Browsers that support the latest features */  
    url('webfont.ttf') format('truetype'), /* Most browsers and devices */  
    url('webfont.svg#svgFontName') format('svg'); /* Legacy iOS devices */  
}
```

```
.container {  
  font-family: 'NewFontName', FallbackFontName, sans-serif;  
}
```

Vijay Shivakumar



CSS3 Transformations



Transformations

CSS3 Transform property

skew | skewX | skewY

transform: skew(45deg) / skewX(45deg) / skewY(45deg)

scale | scaleX | scaleY | scaleZ

transform: scale(1) / scaleX(2) / scaleY(3)

rotate | rotateX | rotateY | rotateZ

transform: rotate(45deg) / rotateX(45deg) / rotateY(45deg) / rotate(45deg)

translate | translateX | translateY

transform: translate(50px) / translateX(45px) / translateY(45px)



Transformations

CSS3 Multiple Transform property

transform: skew(45deg) scale(2) rotate(45deg)



CSS3 Animation



Keyframes

Configurations for CSS3 animation with from and to

```
@keyframes keyFrameName{  
    from{  
        prop: val;  
    }  
    to{  
        prop: val;  
    }  
}
```

```
selector{  
    animation-name: keyFrameName;  
    animation-duration: 3s;  
}
```



Keyframes

Configurations for CSS3 animation with ratio

```
@keyframes keyFrameName{  
    10%{  
    }  
    50%{  
    }  
    100%{  
    }  
}
```

```
selector{  
    animation-name: keyFrameName;  
    animation-duration: 3s;  
}
```



Animation related properties

And their values

animation-name : myAnimationName

specifies the name of the @keyframes

animation-delay : 6s / 120ms

delay between the time the page is loaded & animation.

animation-direction : normal / reverse / alternate / alternate-reverse

alternate direction on each run through the sequence or reset to the start point and repeat itself.

animation-duration : 6s / 120ms

time that an animation should take to complete one cycle.

animation-iteration-count : infinite / number / initial (default = 1)

number of times the animation should repeat.

animation-play-state : paused / running / initial / inherit

lets you pause and resume the animation sequence.

animation-timing-function : ease / ease-in / ease-out / ease-in-out / linear / cubic-bezier / step-start / step-end / steps(i start / end) / initial / inherit

how the animation transitions through keyframes, by establishing acceleration curves.

animation-fill-mode : none / forwards / backwards / both / initial / inherit

values applied by the animation before & after it is executing.



Thank you

