CSS 2.1

Vijay Shivakumar

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 ?

1apple1402mango803grapes78

Why ?

1	Apple	140
2	Mango	80
3	Grapes	78

Why ?

1	Apple	140
2	Mango	80
3	Grapes	78

Why ?

Sl No	Product	Price
1	Apple	140
2	Mango	80
3	Grapes	78

The benefits of CSS

- Separation of content and presentation
  - CSS provided in a file that is separate to the (content) HTML. can more easily be updated.

The benefits of CSS

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

The benefits of CSS

- Improved webpage download speed
  - Since it is small it will download faster
  - Can be cached for re use
  - Save bandwidth
  - Improve rendering speed

The benefits of CSS

- Streamlined maintenance
  - Reduction in errors
  - Easy up gradation

The benefits of CSS

- Different devices different presentations
  - For web
  - For print
  - For mobile etc…

The benefits of CSS

- Table-less layout
  - Tables were meant to display data, and not to make a layout
  - Longer download times, uses more bandwidth
  - Table causes accessibility issues

Where can a CSS be..?

Inline Style

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

Where can a CSS be..?

Embedded Style

```
<style type="text/css" >
h1 {
font-family: Arial;
}
</style>
<h1>Welcome!</h1>
```

Where can a CSS be..?

Linked Style

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

Where can a CSS be..?

Imported Styles

<style type="text/css">

@import "style.css"; or @import url (“style.css”)

</style>

The import needs to the be the first declaration if there is any embedded style declared in the file…

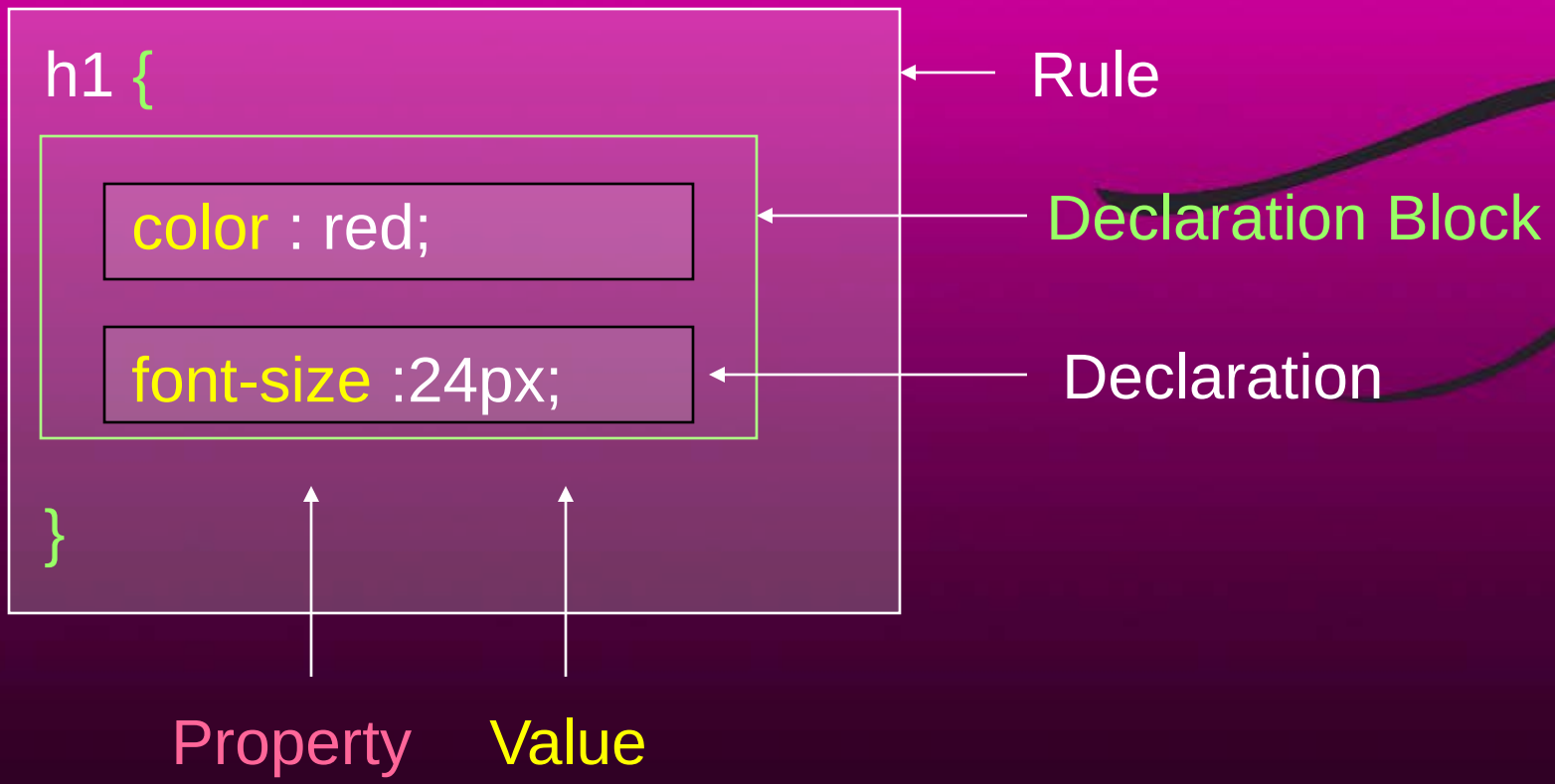
Few vocabulary as we move on ..

- Cascade
  - Multiple sheets and types of sheets can be used.
- Inheritance
  - Style Inheritance relies on the document tree.
- Specificity
  - The style starts applying from generic to specific

Code Anatomy

```
Selector {
Property: Value ;
}
e.g.
p {
color: #ff0000;
}
```

Terminology



Selectors

Element Selectors	Dynamic Pseudo Class Selectors
Class Selectors	Lang Pseudo Class Selectors
ID Selectors	Child Selectors
Descendant Selectors	Adjacent Sibling Selector
Pseudo Class Selectors	General Sibling Selector
Pseudo Element Selectors	Attribute Selectors
Selector Groups	Universal Selector

Element Selectors

```
body {
}
h1 {
 }
p {
}
ul {
}
```

Class Selector

```
.small {
}
```

```
<p class="small">Welcome to your life there's no
    turning back  </p>
```

ID Selector

```
#selection{
}
```

```
<div id="selection ">
This is a division on my page
</div>
```

Must be unique in the document

Descendant Selectors

```
p em {
}
```

```
<p> this is my
<em>paragraph </em>
</p>
```

Pseudo class selectors

a:link {

}

a:hover {

}

You can also make a combination of class and pseudo selection if an anchor tag has a class main then for the link property….

a.main:link

Pseudo element selector

:first-line - Applies style to the first line in a given element

:first-letter- Applies style to the first line in a given element

:before- Applies style before a given element

:after- Applies style after a given element

```
h1:after { content: "header note" }
p:after {content: url(paramark.gif);}
```

Selector Groups

```
h1, h2, h3 {
font-family: Arial;
}
```

Dynamic Pseudo Class Selectors

:focus -Applies the style when focused

:active –Applies style when element is clicked

:hover -Applies the style only when the mouse hovers over the selected element

Language Pseudo Class Selectors

```
p:lang(es) {
}
<p lang="es"> some Spanish text </p>
```

Note: The support for this feature is currently limited

Child Selectors

```
div:first-child{
}
<body>
    <div> Hello every one
        <em>Good Afternoon</em>
    </ div >
</ body >
```

First Child Selectors

```
p:first-child{
}
```

Any rule you write for this selector will be applied to the first paragraph child of every tag.

Note: The support for this feature is currently limited

Adjacent Sibling Selectors

```
th+td  {
}
```

As a table can contain both th and td in the same level they are siblings within the same table tag…

General Sibling Selectors

```
ul~p {
}
```

Any p that comes after the ul as a sibling

Attribute Selectors

```
input[type="submit"] {
}
```

```
<input type=“submit” />
```

Other flavors available

[att] = Apply the style to a given attribute, no matter the value

[att=val] = Apply the style to a given attribute with a specific value

[att~=val] = Apply style to any attributes with space-separated specified values

- e.g. (class=“val" and class="important val“ and class=“val high",
- but not class="my-val“ or class=“value")
- as they don’t match val (should not have any alphabets before or after it)

Universal Selector

```
* {
padding: 0;
margin: 0;
 }
```

Advanced Selectors

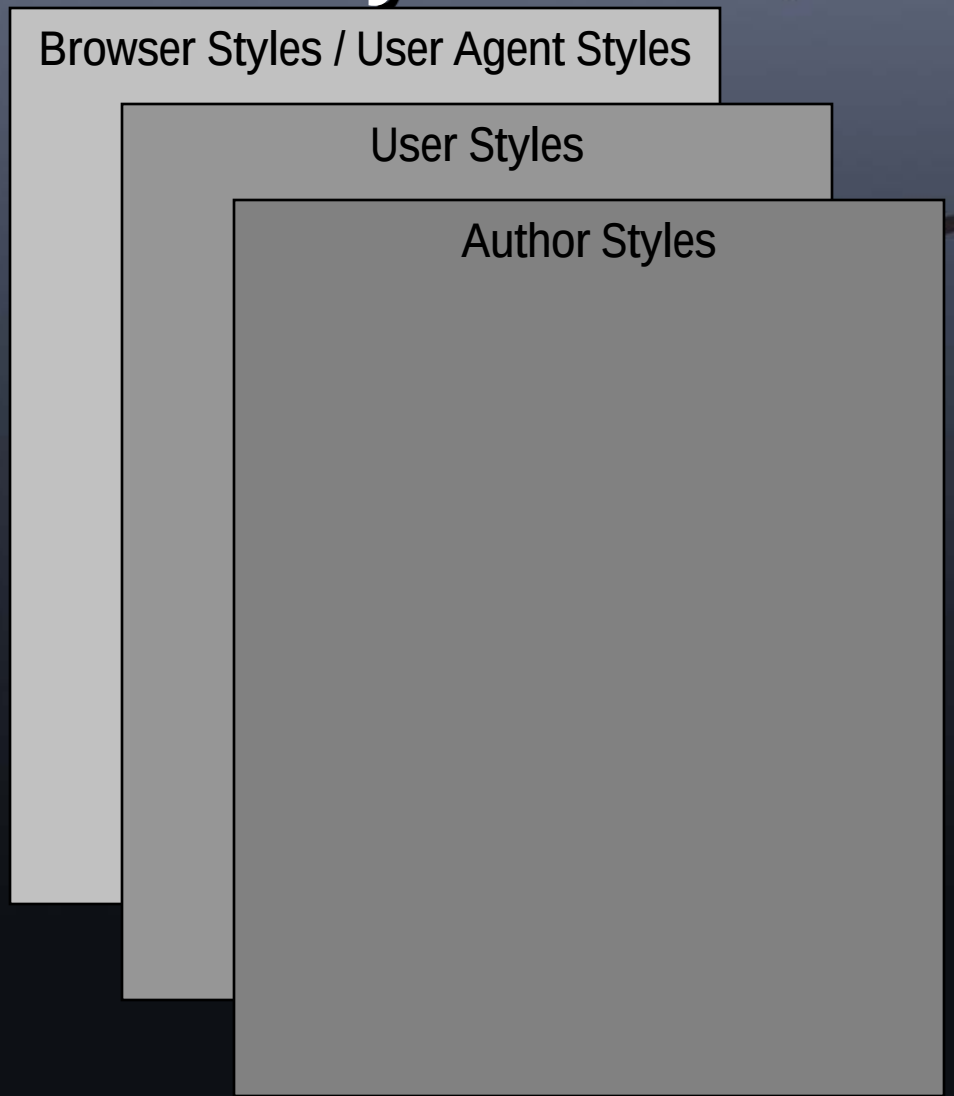
Child and adjacent sibling selectors

Child and adjacent sibling selectors

- #nav > li {font-weight: bold;}

```
<ul id="nav">
<li>Home</li>
<li>Services
<ul>
<li>Design</li>
<li>Development</li>
<li>Consultancy</li>
</ul>
</li>
<li>Contact Us </li>
</ul>
```

Sources of Style



Specificity

• Style=""	1,0,0,0	1000
• #id #id {}	0,2,0,0	200
• #id .class {}	0,1,1,0	110
• tag #id {}	0,1,0,1	101
• #id {}	0,1,0,0	100
• tag.class .class {}	0,0,2,1	21
• tag.class{}	0,0,1,1	11
• tag tag {}	0,0,0,2	2
• tag {}	0,0,0,1	1

Specificity

In 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

DIV tag

A Block level element

Used to create a block with nested elements

Idea is to group the nested elements and to be able to refer with a name or an id

Span tag

Just like div span creates an inline element

That can be addressed to

color property

color:red	color keyword
color:#rrggbb	color in hex notation
color:rgb(rrr,ggg,bbb)	values from 0-255
color:rgb(rrr%,ggg%,bbb%)	values in percentage
color:inherit	color of the parent tag
color:#f00	will also be treated as hex

border property

border-width: thin

border-width: medium

border-width: thick

Note : thin, thick, medium sizes are provided by browser

border-width: 5px

border-width: 1em

---

border-top-width: 1px

border-bottom-width: 1px

border-right-width: 1px

border-left-width: 1px

border-color property

Same as color property

border-style property

border-style: none

border-style: double

border-style: hidden

border-style: dashed

border-style: dotted

border-style: ridge

border-style: solid

border-style: groove

border-style: inset

border-style: outset

border-color property

none

solid

double

groove : A three-dimensional effect that gives the impression that the border is carved into the canvas.

hidden : same as none but works when applied to tables

inset

dashed

outset

dotted

Ridge : a 3D effect that has the opposite effect of groove

Border properties shorthand

border: <width> <style> <color>

Eg; border:thick groove #F00;

margins

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

margins can also accept negatives

padding

padding-top

padding-right

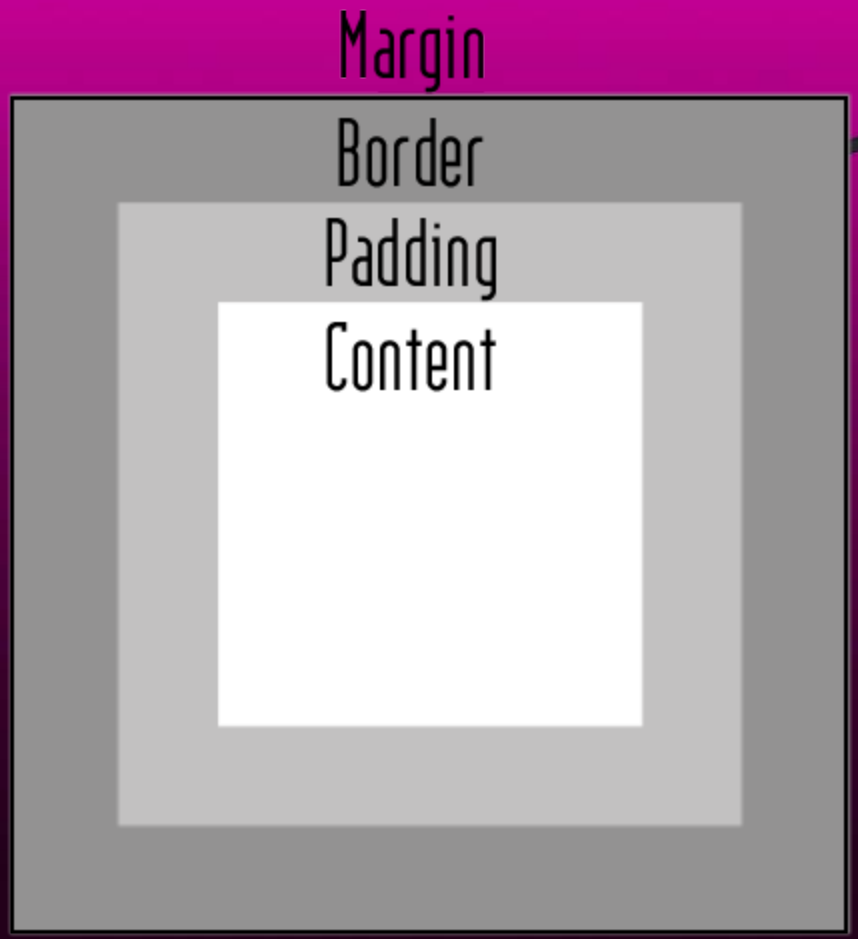
padding-bottom

padding-left

Shorthand

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

The Box Model



position

position: static	the default place
position: relative	takes a position relative to the container
position: absolute	takes user defined position from a relative parent
position: fixed	takes a position irrespective of container
position: inherit	takes position property from parent

Float

Arranges the element

float: left

float: right

float: none

float: inherit

Clear

Clears the block contents

clear: none

clear: left

clear: right

clear: both

clear: inherit

Background

background:url()

background:#rrggbb

Background image

background-image:url(“”)

background-image:none

background-image:inherit

Background repetition

background-repeat:repeat

background-repeat:repeat-x

background-repeat:repeat-y

background-repeat:no-repeat

background-repeat:inherit

Background Attachment

background-attachment:scroll

background-attachment:fixed

background-attachment:inherit

Background Position

background-position:<x> <y>

background-position:top

background-position:bottom

background-position:left

background-position:center

background-position:right

Font Families

Font Styles

Font Weight

Font Size

font-size:12pt

font-size:12pt

font-size:medium

font-size:small

font-size:normal

font-size:larger

font-size:50%

Text indenting

text-indent: <length>

text-indent: <percentage> by containing block

Aligning Text

text-align:left

text-align:right

text-align:center

text-align:justify

Text Decoration

- text-decoration:none
- text-decoration:underline
- text-decoration:overline
- text-decoration:line-through
- text-decoration:sub
- text-decoration:super
- text-decoration:blink

Types of media

```
<link rel="stylesheet" type="text/css" href="print.css" media="print" />
<link rel="stylesheet" type="text/css" href="screen.css" media="screen" />
 <link rel="stylesheet" type="text/css" href="projection.css" media="projection" />
```

all : Users for all devices
aural : Used for speech and sound synthesizers
braille : Used for Braille tactile feedback devices
embossed : Used for Braille printers
handheld : Used for handheld or small devices like PDAs and smartphones
print : Used for printers and print preview
projection : Used for projected presentations
screen : Used for color monitors
tty : Used for teletypes, terminals, and portable devices with limited characters
tv : Used for television and WebTV

Display common properties

- Block elements, such as paragraphs, headings, and lists, sit one above another when displayed in the browser.
- Inline elements such as a, span, and img, sit side by side when they are displayed in the browser and only appear on a new line if there is insufficient room on the previous one.