

ALERT: THIS DOCUMENT WILL BE UPDATED FROM TIME TO TIME!

HOW TO START THE CSS CODING:

1. Set the box-sizing to universal border-box for all elements

```
html {  
  -webkit-box-sizing: border-box;  
  -moz-box-sizing: border-box;  
  box-sizing: border-box;  
}  
*, *:before, *:after {  
  -webkit-box-sizing: inherit;  
  -moz-box-sizing: inherit;  
  box-sizing: inherit;  
}
```

2. Get the Erics Meyer's CSS reset or download normalize.css

<http://necolas.github.io/normalize.css/>

When in doubt about resetting CSS this might help you

<https://www.webfx.com/blog/web-design/should-you-reset-your-css/>

3. You can use bare-minimum Responsive CSS boilerplate such as:

<http://getskeleton.com/>

OR

You can use full blown bootstrap, foundation etc. or handcode everything.

4. Use auto prefixer for browsers compatibility.

<http://prefixr.com/>

<https://autoprefixer.github.io/>

5. Print a copy of CSS SELECTORS. Can come very handy.

<https://drafts.csswg.org/selectors-3/#selectors>

HOW TO START THE HTML5 CODING?

1. First you need a boilerplate

-Generate boilerplate from here: [<-----BASIC PREFERENCE]

<http://htmlshell.com/>

<http://www.initializr.com/>

<https://html5boilerplate.com/> [<-----THIS IS WHAT I PREFER. IT has NORMALIZE.CSS and everything else]

THIS IS EVERYTHING <http://htmlshell.com> has to offer as a HTML5 boilerplate except Include Autorun Javascript.

```
<!DOCTYPE html>
<!--[if lte IE 6]><html class="preIE7 preIE8 preIE9"><![endif]-->
<!--[if IE 7]><html class="preIE8 preIE9"><![endif]-->
<!--[if IE 8]><html class="preIE9"><![endif]-->
<!--[if gte IE 9]><!--><html><!--<![endif]-->
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
  <meta name="viewport" content="width=device-width,initial-scale=1">
  <title>title</title>
  <meta name="author" content="name">
  <meta name="description" content="description here">
  <meta name="keywords" content="keywords,here">
  <link rel="shortcut icon" href="favicon.ico" type="image/vnd.microsoft.icon">
  <link rel="stylesheet" href="//fonts.googleapis.com/css?family=font1|font2|etc" type="text/css">
  <link rel="stylesheet" href="stylesheet.css" type="text/css">
  <style type="text/css">

</style>
<script type="text/javascript">
  var _gaq = _gaq || [];
  _gaq.push(['_setAccount', 'UA-XXXXXXXX-Y']);
  _gaq.push(['_trackPageview']);
  (function()
  {
    var ga = document.createElement('script'); ga.type = 'text/javascript'; ga.async = true;
    ga.src = ('https:' == document.location.protocol ? 'https://ssl' : 'http://www') + '.google-analytics.com/
ga.js';
    var s = document.getElementsByTagName('script')[0]; s.parentNode.insertBefore(ga, s);
  })();
</script>
</head>
<body>

  <script src="//ajax.googleapis.com/ajax/libs/jquery/1.7.2/jquery.min.js"
type="text/javascript"></script>
  <script src="//ajax.googleapis.com/ajax/libs/jqueryui/1.8.21/jquery-ui.min.js"
type="text/javascript"></script>
  <script src="//ajax.googleapis.com/ajax/libs/angularjs/1.2.25/angular.min.js"
type="text/javascript"></script>
  <script src="//ajax.googleapis.com/ajax/libs/dojo/1.7.3/dojo/dojo.js" type="text/javascript"></script>
  <script src="//ajax.googleapis.com/ajax/libs/mootools/1.4.5/mootools-yui-compressed.js"
type="text/javascript"></script>
  <script src="//ajax.googleapis.com/ajax/libs/prototype/1.7.1.0/prototype.js"
type="text/javascript"></script>
  <script src="//cdnjs.cloudflare.com/ajax/libs/modernizr/2.5.3/modernizr.min.js"
type="text/javascript"></script>
```

```
</body>
</html>
```

Master these in CSS:

max-width and min-width:

- Use of max-width and min-width is great when you have to resize the browser.
- Use max-width with certain value, set margin to auto to center the element by default and resize the browser to see the effect.

box-sizing:

UNIVERSAL BOX SIZING.

Applies to

```
html {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  box-sizing: border-box;
}
*, *:before, *:after {
  -webkit-box-sizing: inherit;
  -moz-box-sizing: inherit;
  box-sizing: inherit;
}
```

By using box-sizing: border-box if an element is defined fixed width and height, regardless of defined border, margin and content, the actual size of the element will automatically be resized to its defined width and height.

content-box is default.

It's February 1st in CSStricks.com , which have decided to declare International box-sizing Awareness Day. :D

<https://www.paulirish.com/2012/box-sizing-border-box-ftw/>

white-space

text-decoration

outline:

Outline will give border extra layer but it will cover all four sides while border has top bottom left and right properties.

outline offset will create space with the border. You can go negative pixels so the outline will be inside bg color. You

don't need border to create outline. Usually can be used to create buttons that get zoomed out or set outline offset to make it look like animate. Also can be used transition property to slow down the animation

outline-offset

max-width

display

content

position

grid

overflow:

After setting fixed width and height, content can overflow the given dimension, it can be hidden, scrolled or made visible.

height: x width: y overflow: hidden

box shadow:

The first value is the horizontal offset and the second value is the vertical offset. The shadow color will be inherited from the text color

unless mentioned. If there is a border, then border is gonna make a shadow.

box-shadow: HO VO Color;

z-index:

Only works with positioned elements. Stacks the element using -1 0 1 2 3 etc. on top of each other

writing-mode

filter

cursor

align

stroke

top

bottom

margin:

Always watch out for margin collapse in between top and bottom with two elements

::before, ::after

insert content before or after an element.

don't try to put img as content on these pseudo elements.

Even if it's after or before, it's still gonna be inside that element.

You can use display: block to have its own space apart from the element.

pseudo element can't be selected

Three layouts:

Float:

Flex:

Main axis is horizontal x axis. Cross axis is vertical y axis.

Style using either main axis , row or cross axis , column.

On Main Axis:

justify-content property: flex-start is default. Change it to center to center all elements in that flex container

Use space-between to create space between items.

Use space-around to create space between and on side of items.

On Cross Axis:

align-items property: default is stretch to keep the same size. Change it to flex-start to start at the very top but no stretching.

Change it to center to center all the items.

align-content property: used on multi line flexbox container

flex

Grids:

Browsers:

-ms - Internet Explorer, Edge

-moz - Firefox

-o - Opera

-webkit - Google Chrome, Safari, Android Browser

-khtml- Konqueror

Auto prefixer

<http://prefixr.com/>

<https://github.com/sindresorhus/sublime-autoprefixer>

A list of CSS properties that requires vendor prefixes

<http://ireade.github.io/which-vendor-prefix/>

Boxes - The Great Division:

margin

border

padding

top

bottom

right

left

Positions:

Absolute:

Absolute will position absolutely to its parent element. In most cases its the main window. So, relative position can be used for parent, so absolute position can make reference to that parent element to position itself.

Relative

It takes the element out of document flow but still manages keep its original space down there. You can move the element by using top bottom left right etc. It will move according to the parent element if there is one, not the window.

Relative can overflow the element out of the parent element.

The use case for relative is when an absolute positioning has to be defined for child element respective to its parent. If relative positioning is not defined for parent, the child will take the main window as a point of reference to move itself using absolute value defined.

In short the parent is defined with relative position. Child element gets defined using absolute position.

Fixed

Fixed ignores everything but can be positioned using top bottom etc as well. it makes reference to main window. Its gonna be on the same position all the time even after scrolling.

Sticky

If parent has relative position it wont ignore it to refer to the main window but work as fixed when scrolled.

Static:

This is the default flow of the document if nothing is defined

CSS Units:

Animation:

transition-duration:

Give time in milliseconds just for javascript. It can smooth out hover effect on a button or something.

.box , .box:hover

transition-property:

What are the property you want to transition like width, background, all?

For PERFORMANCE ISSUES, define what things to consider for transition using comma seperated values such as: transition-property: background, transform;

transform:

How much you want to rotate the element?

transition-time-function:

Will take same time for animation but the speed fluctuates depending on what you defined. Create

Cubic-bezier using Web Inspector in

Firefox and chrome put it in as transition-time-function.

transition-delay:

How long to wait until the transition kicks in?

transition:

Use transition property to do all things in one line.

@keyframes

Master CSS selectors:

.class1.class2 <div class="name1 name2"> : selects elements with both classes set within class attributes.

class1 .class2<div class="name1"><div class="name2"></div></div> : selects elements with name2 that is descendant of name1 element.

Notice the space in between.

div, p : Selects all div and p elements. Any p elements outside the div is also selected.

div p : Selects all p elements inside the div element. The p elements can be inside other elements, can be grandsons. p element

has to be inside div elements to be selected

div>p:Selects all p elements inside the div elements but it has to be the CHILDren and div has to be its PARENT. No grandsons or sons of other elements.

div + p : Selects the first child or p elements that immediately follows the div element. If there is another element before p and div, its not selected.
