

css cheat sheet page 1

Comments in CSS

```
/* This is a Comment for CSS. Comments are not displayed in the browser */
```

Adding link to your CSS (in HTML)

```
<link type="text/css" rel="stylesheet" href="stylesheet">
```

Remove bullits on

```
ul {  
    list-style-type: none;  
}
```

Universal selector

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

Syntax for multiple columns and gap)

```
-webkit-column-count: 3; /* Chrome, Safari, Opera */  
-moz-column-count: 3; /* Firefox */  
column-count: 3;  
  
-webkit-column-gap: 40px; /* Chrome, Safari, Opera */  
-moz-column-gap: 40px; /* Firefox */  
column-gap: 40px;
```

css cheat sheet page 2

Targeting Classes and ID's

```
<!--IN HTML-->

<div class="classname"> </div>
<div id="idname"> </div>

/* IN CSS */

.classname {
/*Some CSS!*/
}

#idname {
/*Some CSS!*/
}
```

Targeting element that is child of another element

```
div div p { /* Some CSS */ }
```

Targeting direct children

```
div > p { /* Some CSS */ }
```

Background images

```
body {
    background-image: url("paper.gif");
    background-color: #cccccc;
}
```

css cheat sheet page 3

 and for navigation

```
<!--NAVIGATION SECTION IN HTML-->

<ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="About">About</a></li>
  <li><a href="Movies">Movies</a></li>
  <li><a href="Contact">Contact</a></li>
</ul>
```

 and for navigation

```
/* NAVIGATION IN CSS */

ul {
  list-style-type: none;
  padding: 0px;
}

li a:hover {
  background-color: black;
  color: gray;
}

li a {
  color: white;
  line-height: 35px;
  display: block;
  width: 90px;
  background-color: gray;
  text-align: center;
  text-decoration: none;
  padding: 2px;
  margin-bottom: 5px;
}
```

css cheat sheet page 4

** and
for navigation**

```
<!--NAVIGATION SECTION IN HTML-->

<ul>
  <li><a href="default.asp">home</a></li>
  <li><a href="news.asp">news</a></li>
  <li><a href="contact.asp">contact</a></li>
  <li><a href="https://www.google.com/fonts/specimen/Lato">about</a>
</li>
</ul>
```

** and
for navigation
w/ pseduo-class
"states"**

```
/* NAVIGATION IN CSS */

ul li {
  list-style: none;
  margin: 0px;
  padding: 0px;
}

li a {
  display:block;
  padding: 5px 10px 5px 10px;
  text-decoration: none;
  font-variant: small-caps;
  letter-spacing: .2em;
  float: left;
  color: white;
  background-color:#cbdd7a;
  margin-right: 5px;
  margin-top:5px;
  width:100px;
  text-align: center;
}

li a:hover {
  color:grey;
  background-color:#cbd3d6;
  font-weight: bold;
}

li a:active {
  color:white;
  background-color:#b0b8bc;
  font-weight: bold;
}
```

css cheat sheet page 5

Pseudo-classes for link states

```
/* unvisited link */
a:link {
  color: gray;
}

/* visited link */
a:visited {
  color: gray;
}

/* mouse over link */
a:hover {
  color: black;
  background-color: yellow;
}

/* selected link */
a:active {
  color: red;
}
```

Positioning properties: display summary

The first positioning property: the **display** property. There are four possible values:

block: This makes the element a block box. It won't let anything sit next to it on the page! It takes up the full width.

inline-block: This makes the element a block box, but will allow other elements to sit next to it on the same line.

inline: This makes the element sit on the same line as another element, but without formatting it like a block. It only takes up as much width as it needs (not the whole line).

none: This makes the element and its content disappear from the page entirely!

css cheat sheet page 6

Opacity property

```
div {  
  opacity: 0.5;  
}
```

Overflow

visible: The overflow is not clipped. It renders outside the element's box. (This is default.)

hidden: The overflow is clipped, and the rest of the content will be invisible.

scroll: The overflow is clipped, but a scroll-bar is added to see the rest of the content.

auto: If overflow is clipped, a scroll-bar should be added to see the rest of the content.

initial: Sets this property to its default value.

inherit: Inherits this property from its parent element.

The z-index property

The z-index specifies the stack order of an element (which element will be placed in front of, or behind, others). Example:

```
img {  
  position: absolute;  
  left: 0px;  
  top: 0px;  
  z-index: -1;  
}
```

css cheat sheet page 7

CSS positioning properties

left: Sets the left margin edge for a positioned box

overflow: Specifies what happens if content overflows an element's box

position: Specifies the type of positioning for an element

z-index: Sets the stack order of an element

right: Sets the right margin edge for a positioned box

top: Sets the top margin edge for a positioned box

Positioning

position: static

- if you don't specify an element's positioning type, it defaults to "static"
- "static" specifies that the element will fall in the "normal document flow"

position: absolute

- when an element is set to "position: absolute," it's then positioned in relation to the first parent element it has that doesn't have "position: static."
- Absolutely positioned elements are removed from the normal flow. The document and other elements behave like the absolutely positioned element does not exist. Absolutely positioned elements can overlap other elements.

- if there's no such element, the element with "position: absolute" it gets positioned relative to <html>

position: relative

- "relative positioning" tells the element to move relative to where it would have landed if it just had the default static positioning
- if you give an element relative positioning, it moves from where it otherwise would have been (not from a particular thing or other element)

position: fixed

- fixed positioning anchors an element to the browser window
- if you scroll up and down, the fixed element stays put even as other elements scroll past

css cheat sheet page 8

**CSS3
transform
property**

```
-ms-transform: rotate(30deg); /* IE 9 */  
-webkit-transform: rotate(30deg); /* Chrome, Safari, Opera */  
transform: rotate(30deg);
```
