**<link rel="stylesheet" href="…">**

**@import url("styles.css");**

**Selectors**

**Common Selectors**

* **By Tag**

**h1 { font-family: verdana, sans-serif; }**

* **By Class**

**.myClass {border: 1px solid red}**

* **By ID**

**#element\_id { color: #ff0000; }**

* **Combinated**

**h1, .link, #top-link {font-weight: bold;}**

* **Nested**
  + - * + **p a {text-decoration: underline}**
        + **p \* {color: black}**

**Това ще съответства на всички потомци на <P> елемент**

* + - * + **img + .link {floar: right}**

Това ще съответства на всички братя и сестри с името на класа връзка, които се появяват веднага след <IMG> таг

* + - * + p > .error {font-size: 8px}

Това ще съответства на всички елементи с клас error, директни деца на <P> таг

* + - * + .class1.class2 {font-weight: bold}

Съответства на елементите които съдържат всички вложени класове

**Attribute Selectors**

* **a[title] {color:black}**

**Element with a given attributeSelects <a> elements with title**

* **input[type=text] { font-family:Consolas}**

**Elements with a concrete attribute value**

* **a[title\*=logo] {border: none}**

**Elements whose attribute values contain a word**

**Pseudo Selectors**

* **Common Pseudo Selectors**
  + - * + **:hover**
        + **:visited**
        + **:active**
        + **:lang**
        + **:firs-line**
        + **:before**
        + **:after**
      * **Structural Pseudo-classes**
        + **:root**

**The root of the document**

* + - * + **E:nth-child(n)**

**An E element, the n-th child of its parent**

* + - * + **E:nth-last-child(n)**

**An E element, the n-th child of its parent, counting from the last on**

* + - * + **E:nth-of-type(n)**

**An E element, the n-th sibling of its type**

* + - * + **E:nth-last-of-type(n)**

**An E element, the n-th sibling of its type, counting from the last one**

* + - * + **E:last-child**

**An E element, last child of its parent**

* + - * + **E:first-of-type**

**An E element, first sibling of its type**

* + - * + **E:last-of-type**

**An E element, last sibling of its type**

* + - * + **E:only-child**

**An E element, only child of its parent**

* + - * + **E:only-of-type**

**An E element, only sibling of its type**

* + - * + **E:empty**

**An E element that has no children (including text nodes)**

* + - * + **More detailed descriptions:**

**https://www.w3.org/TR/css3-selectors/#structural-pseudos**

* + - * **UI Selectors**
        + **E:enabled**

**A user interface element E which is enabled**

* + - * + **E:disabled**

**A user interface element E which is disabled**

* + - * + **E:checked**

**A user interface element E which is checked (for instance a radio-button or checkbox)**

* + - * **Other Selectors**
        + **E:target**

**An E element being the target of the referring URI**

* + - * + **E:not(s)**

**An E element that does not match simple selector**

* + - * + **E~F**

**An F element preceded by an E element**

**Values**

**Size Values**

* + - * + **Pixels, ems, e.g. 12px , 1.4em**
        + **Points, inches, centimeters, millimeters**

**E.g. 10pt , 1in, 1cm, 1mm**

* + - * + **Percentages, e.g. 50%**
        + **Zero can be used with no unit: border: 0;**

**Color Values**

* + - * + **RGB Colors**

**#44fa36 – values are in hex**

**rgb(<red>, <green>, <blue>) – decimal values between integers 0 and 255**

* + - * + **RGBA Colors**

**(<red>, <green>, <blue>, <alpha>) – alpha value is between 0.0 and 1.0**

* + - * + **HLS Colors**

**Hue is a degree on the color wheel**

**0 (or 360) is red, 120 is green, 240 is blue**

**Saturation is a percentage value**

**100% is the full color**

**Lightness is also a percentage**

**0% is dark (black)**

**100% is light (white)**

**50% is the average**

**hsl(0, 100%, 50%)**

* + - * + **HLSA Colors**

**hsla(0, 100%, 50%, 0.5)**

**CSS Cascade (Precedence)**

**Browser styles (least priority)**

**Reset default browser styles:**

**\* {**

**margin: 0;**

**padding: 0;**

**}**

**Normal user styles**

**Normal author styles (external, in head, inline)**

**Important author styles**

* **Example**

**a {**

**color: red !important;**

**}**

**Important user styles (max priority)**

**Properties**

**Text-related Properties**

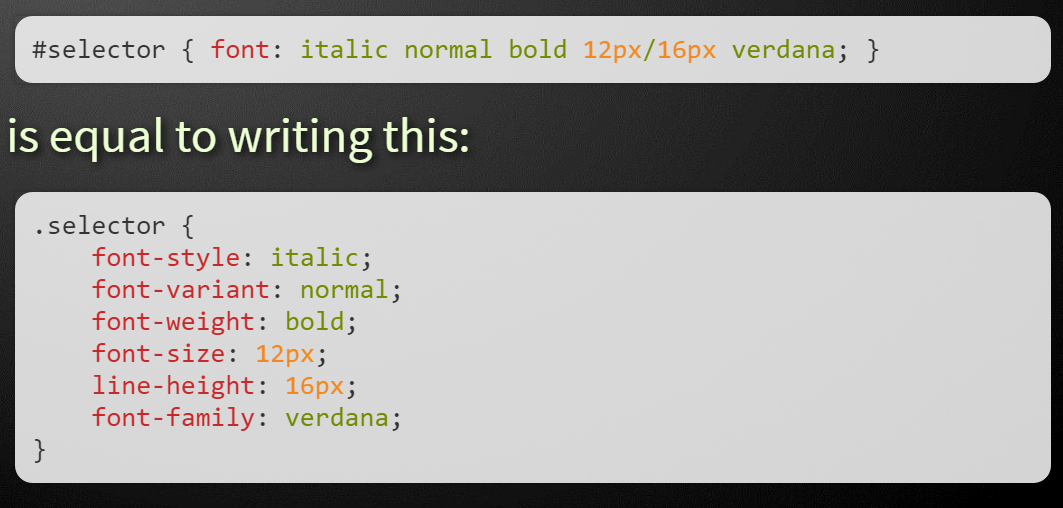
* **color – specifies the color of the text**
* **font-size – слагаме default в body, а другите ги променяма с em-ове**
* **font-family – comma separated font names**

**Example: verdana, sans-serif, etc.**

**The browser loads the first one that is available**

**The browser loads the first one that is available**

* + - * **font-weight can be normal, bold, bolder, lighter (or a number in range [100 … 900] – use only on special fonts!!!)**
      * **font-style – styles the font (italic)**
      * **text-decoration – decorates the text**
      * **text-align – defines the alignment of text or other content**
      * **letter-spacing / word sapcing**
      * **line height**
      * **font variant**
      * **font – shorthand rule for setting multiple font properties at the same time**

****

* + - * **@font-face – use to declarate font (google fonts)**

**@font-face {**

**font-family: SketchRockwell;**

**src: url('SketchRockwell-Bold.ttf');**

**}**

**.my\_CSS3\_class {**

**font-family: SketchRockwell;**

**font-size: 3.2em;**

**}**

* **Text Shadow - applies shadow to text**

**Syntax: text-shadow: <horizontal-distance> <vertical-distance> <blur-radius> <shadow-color>;**

**.selector {**

**text-shadow: 2px 2px 7px #000000;**

**}**

* **Text Overflow, overflow and with-space - specifies what should happen when text overflows the containing element**

**Syntax: text-overflow: <value>;**

**Possible values:**

* + - * + **ellipsis - Display ellipses to represent clipped text**

****

* + - * + **clip - Default value, clips text**

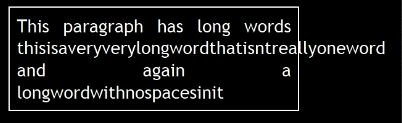
****

* **Word Wrapping**

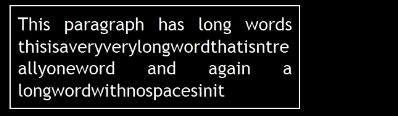
**Syntax: word-wrap: <value>;**

**Possible values:**

* + - * + **normal**

****

* + - * + **break-word**

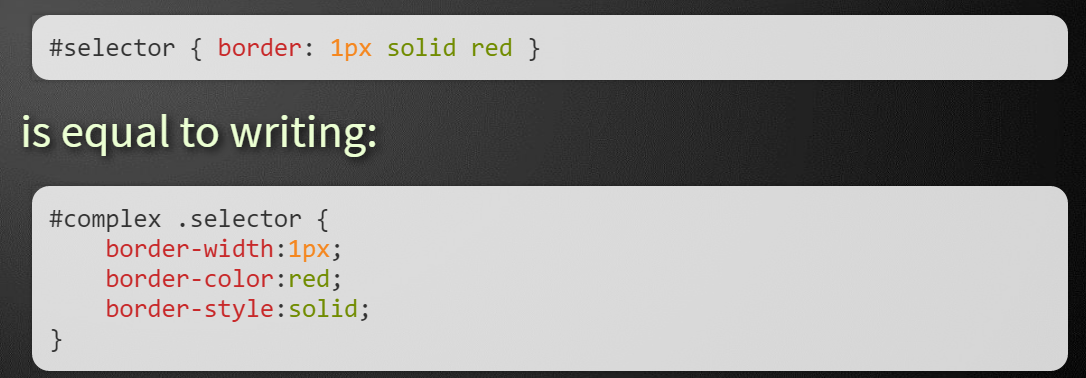
****

**Borders**

* + - * **border-width: thin, medium, thick or numerical value (e.g. 10px)**
      * **border-color: color alias or RGB value**
      * **border-style: none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset**
      * **Each property can be defined separately for left, top, bottom and right**

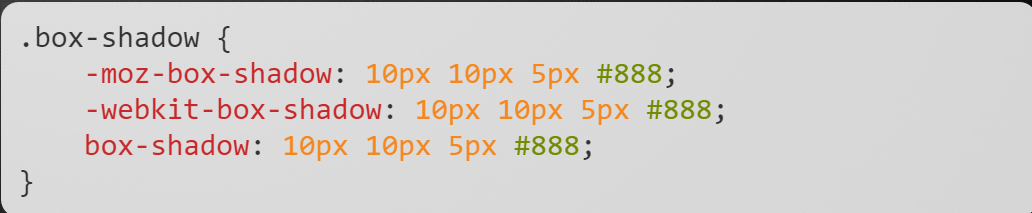
**border-top-style, border-left-color, …**

* + - * **border: shorthand rule for setting border properties at once:**

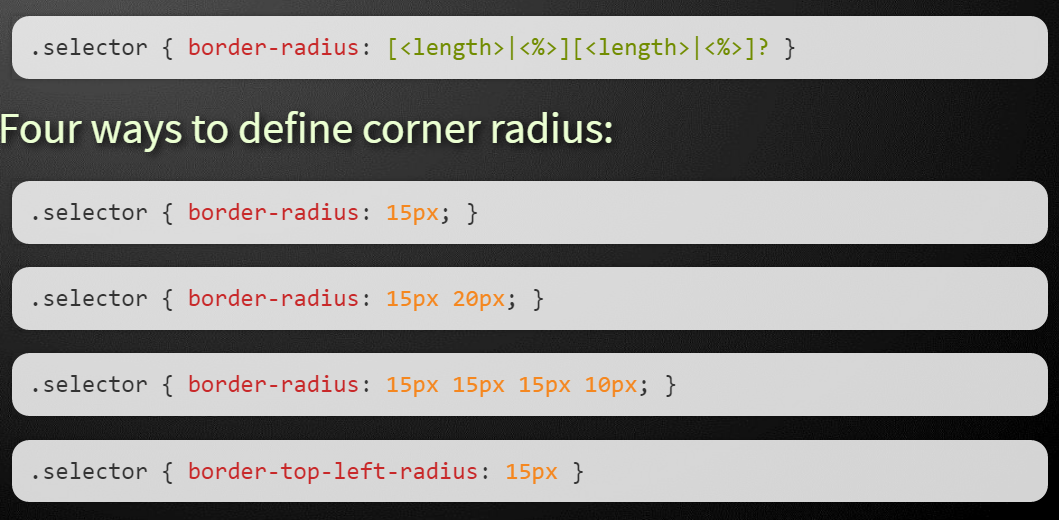
****

**This is valid and for: border-top, border-left, border-right, border-bottom**

* **Box shadow have specifying values for color, size, blur and offset**

****

* **Rounded Corners - done by the border-radius property**

****

**Background Properties**

* **background-image**

**background-image:url("back.gif");**

* **background-color**
* **background repeat**

**repeat-x, repeat-y, repeat, no-repeat**

* **background-attachment**

**fixed / scroll**

* **background-position - specifies vertical and horizontal position of the background image**

**Vertical position: top, center, bottom**

**Horizontal position: left, center, right**

**Both can be specified in percentage or other numerical values**

**.first-selector {**

**background-position: top left;**

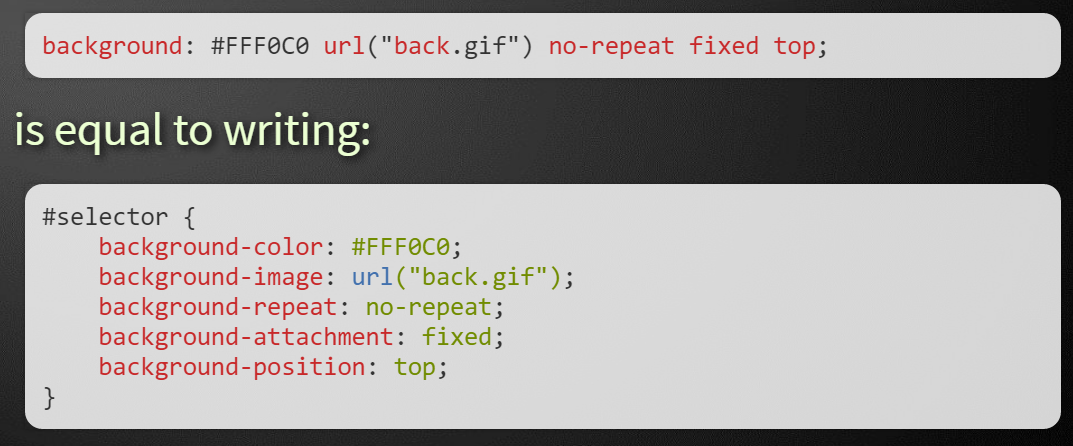
**}**

**.second-selector {**

**background-position: -5px 50%;**

**}**

* **background - shorthand rule for setting background properties at the same time:**



* **Gradient Backgrounds**

**.linear-gradient {**

**/\* Firefox 3.6+ \*/**

**background: -moz-linear-gradient(100% 100% 90deg,**

**#FFFF00, #0000FF);**

**/\* Safari 4-5, Chrome 1-9 \*/**

**background: -webkit-gradient(linear, 0% 0%, 0%**

**100%, from(#0000FF), to(#FFFF00));**

**/\* Safari 5.1+, Chrome 10+ \*/**

**background: -webkit-linear-gradient(#FFFF00,**

**#0000FF);**

**/\* Opera 11.10+ \*/**

**background: -o-linear-gradient(#2F2727, #0000FF);**

**}**

**(use** [**http://www.colorzilla.com/**](http://www.colorzilla.com/) **for generate gradients)**

* **Multiple Backgrounds - simple comma-separated list of images**

**background-image: url(sheep.png), url(grass.png);**

**Opacity**

* **opacity: specifies the opacity of the element**
  + - * + **Floating point number from 0 to 1**
        + **For old Mozilla browsers use –moz-opacity**
        + **For IE use filter:alpha(opacity=value) where value is from 0 to 100; also, "binary and script behaviors" must be enabled and hasLayout must be triggered, e.g. with zoom:1**

**Bonus:**

**Cursor:**

**Display: block, inline block**

**CSS Layout**

**Width and Height**

* **Width - the width property defines numerical value for the width of element, e.g. 200px**

**Applies only for block elements**

* **min-width**

**defines the minimal width**

**overrides width if width < min-width**

* **max-width**

**defines the maximal width**

**overrides width if width > max-width**

* **Height - the height property defines numerical value for the height of element, e.g. 100px**

**Applies only on block elements**

* **min-height**

**defines the minimal height**

**overrides height if height < min-height**

* **max-height**

**defines the maximal height**

**overrides height if height < min-height**

**best value of this properies is % (a percent of the available width)**

**Overflow**

* **overflow - defines the behavior of element when content needs more space than the available**
* **visible(default) – content spills out of the element**
* **auto – show scrollbars if needed**
* **scroll – always show scrollbars**
* **hidden – any content that cannot fit is clipped**

**Display**

* **display - controls the display of the element and the way it is rendered and if breaks should be placed before and after the element**
* **inline - no breaks are placed before or after**
* **block - breaks are placed before AND after the element**
* **inline - block - no breaks are placed before and after (like inline)**

**height and width can be applied (like block)**

* **none - element is hidden and its dimensions are not used to calculate the surrounding elements rendering**
* **table, table-row, table-cell - the elements are arranged in a table-like layout**

**Visibility**

* **Visibility - determines whether the element is visible**
* **hidden - element is not rendered, but still occupies place on the page**
* **visible - element is rendered normally**
* **collapse - removes a row or column, but it does not affect the table layout - only for table elements**

**Margins and Paddings**

* **margin and padding - define the spacing around the element**
* **They take numerical value, e.g. 10px or -5px**
* **Can be defined for each of the four sides separately – margin-top, padding-left, …**
* **margin is the spacing outside of the border**
* **padding is the spacing between the border and the content**
* **Short Rules**
* **Sets all four sides to have margin of 5px**

**.container {**

**margin: 5px;**

**}**

* **top and bottom to 10px, left and right to 20px**

**.container {**

**margin: 10px 20px;**

**}**

* **top 5px, left and right 3px, bottom 8px**

**.container {**

**margin: 5px 3px 8px;**

**}**

* **top, right, bottom, left (clockwise from top)**

**.container {**

**margin: 1px 3px 5px 7px;**

**}**

**CSS Box Model**

* **box-sizing - determine whether you want an element to render it's borders and padding within its specified width, or outside of it.**
* **content-box**

**.container {**

**width: 300px;**

**box-sizing: content-box;**

**}**

**box width: 288px + 10px padding + 1px border on each side = 300px**

* **border-box**

**.container {**

**width: 300px;**

**box-sizing: border-box;**

**}**

**box width: 300px, including padding and borders**

**Positioning**

* **position - defines the positioning of the element in the page content flow**
* **static is the default value**
* **relative –according to where the element would appear with static position**
* **absoulute - relative to the first parent element that has a position other than static**
* **fixed - relative to the browser window, but ignores page scrolling**
* **top, left, bottom, right - specifies offset of absolute/fixed/relative positioned element as numerical values**
* **z-index - specifies the stack level of positioned elements**
* **vertical-align - sets the vertical alignment of an inline element, according to the line height**
* **Values: baseline, sub, super, top, text-top, middle, bottom, text-bottom or numeric**
* **Also used for content of table cells (which apply middle alignment by default)**

**Floating**

* **float: left/right - makes the element “float” to one side**
* **clear - sets the sides of the element where other floating elements are NOT allowed**
* **Used to "drop" elements below floated ones or expand a container, which contains only floated children**
* **Values: left, right, both**
* **Clearing floats**

**.clearfix:after {**

**content: ".";**

**visibility: hidden;**

**display: block;**

**height: 0;**

**clear: both;**

**}**

**For more info: https://css-tricks.com/all-about-floats/**

**Flexbox**

**https://css-tricks.com/snippets/css/a-guide-to-flexbox/**