Website Design Notes – JS, HTML, CSS

**CSS:**

.classname{

}

#idname{

}

elementname{

}

## More HTML tags

* **Form tags**: There are quite a few tags that are used to create web forms, like [<button>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/button)s and [<input>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input)s. They are typically used to communicate data back to a server, something which we don't enable on Khan Academy. They can also be used with JS to create a game, which we do teach in the HTML/JS course. Learn more about form tags [with these slides](http://www.teaching-materials.org/htmlcss-1day/html-forms/slides.html#slide1). Learn server-side languages like PHP, Python, Ruby on [Codecademy](http://codecademy.com/).
* **Iframes**: The [<iframe>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/iframe) tag is a way to frame another webpage inside yours, and is handy for things like embedding Youtube videos, Google Maps, etc. That tag is disabled on Khan Academy currently for security and moderation purposes, but can be experimented with elsewhere. Learn more [with these slides](http://www.teaching-materials.org/htmlcss-1day/html-embeds/slides.html#slide1).
* **Multimedia tags**: The [<audio>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/audio) and [<video>](https://developer.mozilla.org/en-US/docs/Web/HTML/Element/video) tags are supported in modern browsers as a way to play audio and video files on webpages. Like the <iframe> tag, they are currently disabled on Khan Academy, but can be used elsewhere. Learn more [with these slides](http://www.teaching-materials.org/multimedia/).
* **New semantic tags**: In the past few years, as part of the "HTML5" specification, browsers have added new semantic tags for developers to use instead of <div>s, when appropriate, such as <article>, <aside>, <nav>, etc. There are many [articles about them here](http://html5doctor.com/article-archive/), including [this handy flowchart](http://html5doctor.com/downloads/h5d-sectioning-flowchart.png) for deciding when to use them.

## More CSS selectors and properties

You should be able to experiment with all of these on Khan Academy:

* **CSS3**: We covered CSS that's been supported in browsers for years, but there are many new CSS properties and selectors in what's known as the "CSS3" specifications, and browsers are adding support continuously as they release new versions. You can learn more in these [CSS3 selectors slides](https://www.teaching-materials.org/css3-selectors/#slide1), and [CSS3 properties slides](https://www.teaching-materials.org/css3-fx/). Make sure to check [caniuse.com](http://caniuse.com/) to find out what's supported in each browser.
* **Media queries**: This is a technique that lets you specify different CSS for different situations, like when your webpage is viewed at a smaller size or when it's printed. Learn more [from W3C](https://www.w3schools.com/css/css_rwd_mediaqueries.asp).
* **CSS frameworks**: A framework is a collection of CSS rules and classes, and many developers use CSS frameworks to enable them to write CSS faster. The most popular one is [Twitter Bootstrap](http://getbootstrap.com/), but there's also [ZURB foundation](http://foundation.zurb.com/), [Pure CSS](http://purecss.io/), [Topcoat](http://topcoat.io/), and more.

## Using JS in webpages

Web developers use JavaScript to make webpages interactive, to respond to user events and bring in data dynamically from servers. They do that by embedding <script> tags inside the HTML, and putting JavaScript inside those tags. Their JS code then uses the "DOM API", a set of functions that browsers make available on every webpage to let developers query and manipulate that page.

That means that if you want to learn to make your webpages interactive, you both need to learn the fundamentals of the JavaScript language (which you can do here on Khan Academy), plus you need to learn the DOM API (which you can learn in our [course here](https://www.khanacademy.org/html-css-js)).

**Text properties:**

Line-height: 1.5em; 🡨 changes space b/w lines in paragraph

Text-decoration: underline;

Text-align: center;

Margin/border/padding: top right bottom left (clockwise from top);

Margin: auto; /\*centers it\*/

Border: 1px (thickness) solid/double/ridge/groove/dotted/dashed color;

**Selectors**

* Grouping selectors: means that diff elements will just share styles (h1, h2{color: red;})
* Descendant selectors:
  + (depends on position in html) but you can change only members of a class that are *descendants* of another class using the **space.**
  + table .emperor { 🡨 only changes things with the emperor class that are also in the table.
  + background-color: red;
  + }
* Combining element and class selectors: all elements that have this class name will be changed, using the **.**
  + H2.emperor{
  + Font-family: cursive;
  + }
* Dynamic pseudo-classes with **colon**
  + a:hover{} changes things when the user is hovering
  + a:active{} selects elements that are currently being activated (like when a link is being clicked)
  + a:focus{} usually when someone is tabbing around the page
* **specificity**
  + id more specific than a class, class more specific than an element
  + if a tag has 2 rules that are equally specific, the *last one typed* is the one that wins.

Position: relative;

Top:

Bottom:

Right: left:

Position: absolute; 🡨 use this to make it always in the same place (static) so you can scroll away

Position: fixed; thing always stays at position so you can’t scroll away. Useful for the search tab, header boxes etc.

Float: left/right; //for text wrapping

*When floating a div (like for a sidebar) you have to give it a width so it doesn’t take up all the space*

Clear: both; //don’t’ wrap around any floating elements. Helpful for footer.

z-index: tells the browser what order to draw things in

so z-index: 1; means it’ll be drawn first (in the background)

A{

Text-decoration: none; //this takes away the line under links

}

**Fonts:**

**.song-lyrics {**

**background-color: yellow;**

**font-family: fantasy;**

**font-size: 13px;**

**font-style: italic;**

**font-weight: bold;**

**}**

Body{

Font-size: 12px;

}

H2{

Font-size: 2em

} /\*this makes the h2 font size double that of the body (which it inherits)\*/

HTML

***-DIFFERENCE BETWEEN IDS AND CLASS NAMES: CAN ONLY USE IDs FOR A SINGLE TAG (ONCE) WHILE CLASS NAMES CAN BE USED MULTIPLE TIMES (SO USE FOR GROUPS)***

***-use IDs to override the formatting of a class!***

***-IDs are a bit faster***

CODECADEMY WEB DESIGN NOTES

html:

* **HTML** stands for **H**yper**T**ext **M**arkup **L**anguage and is used to create the structure and content of a webpage.
* Most HTML elements contain opening and closing tags with raw text or other HTML tags between them.
* HTML elements can be nested inside other elements. The enclosed element is the child of the enclosing parent element.
* Any visible content should be placed within the opening and closing <body>tags .
* Headings and sub-headings, <h1> to <h6> tags, are used to enlarge text.
* <p>, <span> and <div> tags specify text or blocks.
* The <em> and <strong> tags are used to emphasize text.
* Line breaks are created with the <br> tag.
* Ordered lists (<ol>) are numbered and unordered lists (<ul>) are bulleted.
* Images (<img>) and videos (<video>) can be added by linking to an existing source

s review what you’ve learned this lesson:

* The <!DOCTYPE html> declaration should always be the first line of code in your HTML files. This lets the browser know what version of HTML to expect.
* The <html> element will contain all of your HTML code.
* Information about the web page, like the title, belongs within the <head> of the page.
* You can add a title to your web page by using the <title> element, inside of the head.
* A webpage's title appears in a browser's tab.
* Anchor tags (<a>) are used to link to internal pages, external pages or content on the same page.
* You can create sections on a webpage and jump to them using <a> tags and adding ids to the elements you wish to jump to.
* Whitespace between HTML elements helps make code easier to read while not changing how elements appear in the browser.
* Indentation also helps make code easier to read. It makes parent-child relationships visible.
* Comments are written in HTML using the following syntax: <!-- comment -->.
* The <table> element creates a table.
* The <tr> element adds rows to a table.
* To add data to a row, you can use the <td> element.
* Table headings clarify the meaning of data. Headings are added with the <th>element.
* Table data can span columns using the colspan attribute.
* Table data can span rows using the rowspan attribute.
* Tables can be split into three main sections: a head, a body, and a footer.
* A table's head is created with the <thead> element.
* A table's body is created with the <tbody> element.
* A table's footer is created with the <tfoot> element.
* All the CSS properties you learned about in this course can be applied to tables and their data.

<table>

<thead>

<tr>

<th>Pet name</th>

<th>Species</th>

<th>Color</th>

</tr>

</thead>

<tbody>

<tr>

<td>Black & white</td>

<td>rabbit</td>

<td>black and white</td>

</tr>

<tr>

<td>Daemon</td>

<td>cat</td>

<td>black</td>

</tr>

<tr>

<td>Angel</td>

<td>cat</td>

<td>orange</td>

</tr>

</tbody>

</table>

Congratulations on completing HTML Tables!

**CSS**

**Block-level Elements**

A block-level element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

The <div> element is a block-level element.

Examples of block-level elements:

* <div>
* <h1> - <h6>
* <p>
* <form>
* <header>
* <footer>
* <section>

**Inline Elements**

An inline element does not start on a new line and only takes up as much width as necessary.

This is an inline <span> element inside a paragraph.

Examples of inline elements:

* <span>
* <a>
* <img>

Display: none;

display: none; is commonly used with JavaScript to hide and show elements without deleting and recreating them. Take a look at our last example on this page if you want to know how this can be achieved.

The <script> element uses display: none; as default.

JavaScript, Khan Academy documentation:

**Shapes**

[rect(x, y, w, h)](https://www.khanacademy.org/computer-programming/rectx-y-w-h/839496660" \t "_blank)

[ellipse(x, y, w, h)](https://www.khanacademy.org/computer-programming/ellipsex-y-w-h/839435680" \t "_blank)

[triangle(x1, y1, x2, y2, x3, y3)](https://www.khanacademy.org/computer-programming/trianglex1-y1-x2-y2-x3-y3/839546599" \t "_blank)

[line(x1, y1, x2, y2)](https://www.khanacademy.org/computer-programming/linex1-y1-x2-y2/827916099" \t "_blank)

[point(x, y)](https://www.khanacademy.org/computer-programming/pointx-y/827809834" \t "_blank)

[arc(x, y, w, h, start, stop)](https://www.khanacademy.org/computer-programming/arcx-y-w-h-start-stop/1903619297" \t "_blank)

[bezier(x1, y1, cx1, cy1, cx2, cy2, x2, y2)](https://www.khanacademy.org/computer-programming/bezierx1-y1-cx1-cy1-cx2-cy2-x2-y2/956920482" \t "_blank)

[quad(x1, y1, x2, y2, x3, y3, x4, y4)](https://www.khanacademy.org/computer-programming/quadx1-y1-x2-y2-x3-y3-x4-y4/1907244018" \t "_blank)

[image(image, x, y, width\*, height\*)](https://www.khanacademy.org/computer-programming/imageimage-x-y/937672662" \t "_blank)

See also: [ellipseMode](https://www.khanacademy.org/computer-programming/ellipsemodemode/6709863212122112), [rectMode](https://www.khanacademy.org/computer-programming/rectmodemode/4556457341091840), [imageMode](https://www.khanacademy.org/computer-programming/imagemodemode/5295050787389440), [strokeCap](https://www.khanacademy.org/computer-programming/strokecapmode/5288182060941312), [bezierPoint](https://www.khanacademy.org/computer-programming/bezierpointa-b-c-d-t/4551007698681856), [bezierTangent](https://www.khanacademy.org/computer-programming/beziertangenta-b-c-d-t/4736929853603840), [curve](https://www.khanacademy.org/computer-programming/curve/5105742184972288), [curvePoint](https://www.khanacademy.org/computer-programming/curvepointa-b-c-d-t/5879387094253568), [curveTangent](https://www.khanacademy.org/computer-programming/curvetangenta-b-c-d-t/4708940860358656), [curveTightness](https://www.khanacademy.org/computer-programming/curvetightnesssquishy/4792873740402688)

**Complex Shapes**

[beginShape() / endShape() / vertex()](https://www.khanacademy.org/computer-programming/beginshapeendshape/5462945756610560" \t "_blank)

[curveVertex()](https://www.khanacademy.org/computer-programming/curvevertexx-y/6499542019080192" \t "_blank)

[bezierVertex()](https://www.khanacademy.org/computer-programming/beziervertexcx1-cy1-cx2-cy2-x-y/5085481683386368" \t "_blank)

See also: [strokeJoin](https://www.khanacademy.org/computer-programming/strokejoinmode/5662070842327040" \t "_blank), [curveTightness](https://www.khanacademy.org/computer-programming/curvetightnesssquishy/4792873740402688" \t "_blank)

**Colors**

* [background(r, g, b)](https://www.khanacademy.org/computer-programming/backgroundr-g-b/839653892)Set the background color
* [fill(r, g, b)](https://www.khanacademy.org/computer-programming/fillr-g-b/839774957)Set the fill color for shapes
* [noFill()](https://www.khanacademy.org/computer-programming/nofill/877946290)Turn off fill for shapes
* [stroke(r, g, b)](https://www.khanacademy.org/computer-programming/stroker-g-b/839545910)Set the outline color for shapes
* [strokeWeight(thickness)](https://www.khanacademy.org/computer-programming/strokeweightthickness/877859744)Change the thickness of lines and outlines
* [noStroke()](https://www.khanacademy.org/computer-programming/nostroke/839859412)Turn off outlines for shapes
* [color(r, g, b)](https://www.khanacademy.org/computer-programming/colorr-g-b/957020020)Store a color in a variable
* [blendColor(c1, c2, MODE)](https://www.khanacademy.org/computer-programming/blendcolorc1-c2-mode/4530750216994816)Blend two colors together
* [lerpColor(c1, c2, amount)](https://www.khanacademy.org/computer-programming/lerpcolorc1-c2-amount/4759935778816000)Find color between 2 colors

See also: [colorMode](https://www.khanacademy.org/computer-programming/colormode/5833774306689024), [red](https://www.khanacademy.org/computer-programming/redcolor/5102159326609408), [green](https://www.khanacademy.org/computer-programming/greencolor/5877638103040000), [blue](https://www.khanacademy.org/computer-programming/bluecolor/5177743654256640), [alpha](https://www.khanacademy.org/computer-programming/alphacolor/6687311345483776), [hue](https://www.khanacademy.org/computer-programming/huecolor/6620387366404096), [saturation](https://www.khanacademy.org/computer-programming/saturationcolor/6358678768713728), [brightness](https://www.khanacademy.org/computer-programming/brightnesscolor/5888575639912448)

**Text**

* [text(text, x, y)](https://www.khanacademy.org/computer-programming/texttext-x-y/937624625)Draw some text
* [textFont(font, size\*)](https://www.khanacademy.org/computer-programming/textfontfont-size/940030209)Changes the font of text
* [textSize(size)](https://www.khanacademy.org/computer-programming/textsizesize/937728198)Change the size of text

See also: [textWidth](https://www.khanacademy.org/computer-programming/textwidthstr/4799257177489408" \t "_blank), [textAscent](https://www.khanacademy.org/computer-programming/textascent/5975406490419200" \t "_blank), [textDescent](https://www.khanacademy.org/computer-programming/textdescent/5638769772331008" \t "_blank), [textLeading](https://www.khanacademy.org/computer-programming/textleadingdist/6369013500215296" \t "_blank), [textAlign](https://www.khanacademy.org/computer-programming/textalignalign-yalign/4508437190803456" \t "_blank)

**Transform**

* [rotate(angle)](https://www.khanacademy.org/computer-programming/rotateangle/6386091934351360)Rotate shapes by an angle
* [scale(amount)](https://www.khanacademy.org/computer-programming/scalex-y/6712922034143232)Scale shapes in both dimensions
* [translate(x, y)](https://www.khanacademy.org/computer-programming/translatex-y/6505693083336704)Translate shapes by an offset

See also: [pushMatrix/popMatrix](https://www.khanacademy.org/computer-programming/pushmatrixpopmatrix/5505194477486080" \t "_blank), [resetMatrix](https://www.khanacademy.org/computer-programming/resetmatrix/4597705468805120" \t "_blank), [printMatrix](https://www.khanacademy.org/computer-programming/printmatrix/5934612152844288" \t "_blank)

**Environment**

* [width](https://www.khanacademy.org/computer-programming/width/5933816543707136) / [height](https://www.khanacademy.org/computer-programming/height/4544657253990400" \t "_blank)The size of the canvas
* [draw = function(){ }](https://www.khanacademy.org/computer-programming/draw/5192527846309888)Called repeatedly during program execution
* [playSound(sound)](https://www.khanacademy.org/computer-programming/playsoundsound/6655307787534336)Plays one of the allowed sounds

See also: [Program.assertEqual](https://www.khanacademy.org/computer-programming/programassertequal/6737630444388352), [Program.restart](https://www.khanacademy.org/computer-programming/programrestart/5772421353439232), [frameRate(fps)](https://www.khanacademy.org/computer-programming/frameratefps/6427359154536448), [frameCount](https://www.khanacademy.org/computer-programming/framecount/5893935759097856), [loop](https://www.khanacademy.org/computer-programming/loop/5519218351013888), [noLoop](https://www.khanacademy.org/computer-programming/noloop/6342789906300928)

**Mouse**

* [mouseX, mouseY](https://www.khanacademy.org/computer-programming/mousex-mousey/5538427537719296)Current coordinates of the mouse
* [pmouseX, pmouseY](https://www.khanacademy.org/computer-programming/pmousex-pmousey/5082026180870144)Past coordinates of the mouse
* [mouseButton](https://www.khanacademy.org/computer-programming/mousebutton/6304348237725696)Which button is pressed
* [mouseIsPressed](https://www.khanacademy.org/computer-programming/mouseispressed/939933053)Whether mouse is being pressed
* [mouseClicked = function(){ }](https://www.khanacademy.org/computer-programming/var-mouseclicked-function/1897113673)Called when mouse is clicked
* [mousePressed = function(){ }](https://www.khanacademy.org/computer-programming/var-mousepressed-function/1907626123)Called when mouse is pressed
* [mouseReleased = function(){ }](https://www.khanacademy.org/computer-programming/var-mousereleased-function/1907670118)Called when mouse is released
* [mouseMoved = function(){ }](https://www.khanacademy.org/computer-programming/var-mousemoved-function/5689134450475008)Called when mouse is moved
* [mouseDragged = function(){ }](https://www.khanacademy.org/computer-programming/var-mousedragged-function/6273229589053440)Called when mouse is dragged
* [mouseOver = function(){ }](https://www.khanacademy.org/computer-programming/var-mouseover-function/4681234999410688)Called when mouse moves over canvas
* [mouseOut = function(){ }](https://www.khanacademy.org/computer-programming/var-mouseout-function/6643226391871488)Called when mouse moves out of canvas

**Keyboard**

* [key](https://www.khanacademy.org/computer-programming/key/5790940799172608)Number representing which key is pressed
* [keyCode](https://www.khanacademy.org/computer-programming/keycode/939888407)Represents when a special key is pressed
* [keyIsPressed](https://www.khanacademy.org/computer-programming/keyispressed/939855509)True if a key is being pressed, false otherwise
* [keyPressed = function(){ }](https://www.khanacademy.org/computer-programming/var-keypressed-function/6270179101114368)Called when a key is pressed
* [keyReleased = function(){ }](https://www.khanacademy.org/computer-programming/var-keyreleased-function/6349252120805376)Called when a key is released
* [keyTyped = function(){ }](https://www.khanacademy.org/computer-programming/var-keytyped-function/5762464763346944)Called when a key is typed

**Math**

* [random(low, high)](https://www.khanacademy.org/computer-programming/randomlow-high/827911487)Generate a random number
* [dist(x1, y1, x2, y2)](https://www.khanacademy.org/computer-programming/distx1-y1-x2-y2/1917352082)Calculates the distance between two points
* [constrain(value, min, max)](https://www.khanacademy.org/computer-programming/constrainvalue-min-max/5870136103796736)Constrain value between min and max
* [min(num1, num2)](https://www.khanacademy.org/computer-programming/minnum1-num2/4693347713155072)Return the minimum of two numbers
* [max(num1, num2)](https://www.khanacademy.org/computer-programming/maxnum1-num2/4755409722146816)Return the maximum of two numbers
* [abs(num)](https://www.khanacademy.org/computer-programming/absnum/877930637)Take the absolute value of a number
* [log(num)](https://www.khanacademy.org/computer-programming/lognum/877921884)Take the logarithm of a number
* [pow(num, exponent)](https://www.khanacademy.org/computer-programming/pownum-exponent/877858853)Raise a number to an exponent
* [sq(num)](https://www.khanacademy.org/computer-programming/sqnum/6588187426160640)Square a number
* [sqrt(num)](https://www.khanacademy.org/computer-programming/sqrtnum/6473360267542528)Take the square root of a number
* [round(num)](https://www.khanacademy.org/computer-programming/roundnum/5907281296228352)Return nearest integer
* [ceil(num)](https://www.khanacademy.org/computer-programming/ceil/5491781646942208)Return nearest integer of greater/equal value
* [floor(num)](https://www.khanacademy.org/computer-programming/floornum/5703004061696000)Return nearest integer of lesser/equal value
* [PVector(x, y)](https://www.khanacademy.org/computer-programming/pvectorx-y/5218818305556480)An object that describes a 2-dimensional vector

See also: [mag](https://www.khanacademy.org/computer-programming/magx1-y1-not-working-yet/5983219002376192), [exp](https://www.khanacademy.org/computer-programming/expvalue/5228990398726144), [map](https://www.khanacademy.org/computer-programming/mapvalue-low1-high1-low2-high2/4587974079545344), [norm](https://www.khanacademy.org/computer-programming/normvalue-low-high/6581050767572992), [lerp](https://www.khanacademy.org/computer-programming/lerpvalue1-value2-amount/6456916012171264), [noise](https://www.khanacademy.org/computer-programming/noise/5618485581316096), [noiseDetail](https://www.khanacademy.org/computer-programming/noisedetailoctaves-falloff/6549875814563840), [Random.nextGaussian](https://www.khanacademy.org/computer-programming/randomseed/5697038959247360)

**Trigonometry**

* [cos(degrees)](https://www.khanacademy.org/computer-programming/cosdeg/948226821)Take the cosine of an angle
* [sin(degrees)](https://www.khanacademy.org/computer-programming/sindeg/948255306)Take the sine of an angle
* [tan(degrees)](https://www.khanacademy.org/computer-programming/tandeg/948018680)Take the tangent of an angle

See also: [acos](https://www.khanacademy.org/computer-programming/acosval/4542953527705600" \t "_blank), [asin](https://www.khanacademy.org/computer-programming/asinval/5061655520083968" \t "_blank), [atan](https://www.khanacademy.org/computer-programming/atanval/4869834059808768" \t "_blank), [atan2](https://www.khanacademy.org/computer-programming/atan2x-y/6206505994420224), [radians](https://www.khanacademy.org/computer-programming/radiansangle/6628151023108096), [degrees](https://www.khanacademy.org/computer-programming/degreesangle/6674991668002816), [angleMode](https://www.khanacademy.org/computer-programming/anglemode-degrees-vs-radians/2350518561" \t "_blank)

**Date & Time**

* [day()](https://www.khanacademy.org/computer-programming/day/4526347808407552) / [month()](https://www.khanacademy.org/computer-programming/month/5388987023753216) / [year()](https://www.khanacademy.org/computer-programming/year/6216887939629056)Current date
* [hour()](https://www.khanacademy.org/computer-programming/hour/5806957302644736) / [minute()](https://www.khanacademy.org/computer-programming/minute/6638408210317312) / [second()](https://www.khanacademy.org/computer-programming/second/5743886110556160)Current time
* [millis()](https://www.khanacademy.org/computer-programming/millis/5970545493409792)Milliseconds elapsed since program start

**Debugging**

* [debug(arg1, arg2, ...)](https://www.khanacademy.org/computer-programming/debugarg1-arg2/939146973)Print to your browser's developer console
* [println(data)](https://www.khanacademy.org/computer-programming/printlndata/6120466259378176)Print a new line to the canvas console
* [print(data)](https://www.khanacademy.org/computer-programming/printdata/5110798099677184)Print to the canvas console

**JavaScript**

* [var drawWinston = function(){ }](https://www.khanacademy.org/computer-programming/var-drawwinston-function/877982168)Define a new function
* [var array = [0, 1, 2, 3, 4]](https://www.khanacademy.org/computer-programming/var-array-0-1-2-3-4/957074792)Make an array of 5 numbers
* [if (x < 20){ }](https://www.khanacademy.org/computer-programming/if-x-20/957023758)Only run code if a certain condition is true
* [for (var i = 0; i < 8; i++){ }](https://www.khanacademy.org/computer-programming/for-var-i-0-i-8-i-1/877960284)Repeat code a fixed number of times
* [while (x < 250){ }](https://www.khanacademy.org/computer-programming/while-x-250/1907383465)Only run code while a certain condition is true