HTML 5 TAG REFERENCE

	<!--...-->	Define a comment
	<!DOCTYPE>	Defines the document type
	<a>	Defines a hyperlink href, hreflang, media, ping , rel, target, type
	<abbr>	Defines an abbreviation
✻	<acronym>	Used to define an embedded acronyms
	<address>	Defines an address element
✻	<applet>	Used to define an embedded applet
	<area>	Defines an area inside an image map alt, coords, href, hreflang, media, ping, rel, shape, target, type
★	<article>	Defines an article cite, pubdate
★	<aside>	Defines content aside from the page content
★	<audio>	Defines sound content autobuffer, autoplay, controls, src
	<b>	Defines bold text
	<base>	Defines a base URL for all the links in a page href, target
✻	<basefont>	Used to define a default font-color, font-size, or font-family for all the document
	<bdo>	Defines the direction of text display dir
✻	<big>	Used to make text bigger
	<blockquote>	Defines a long quotation cite
	<body>	Defines the body element
	 	Inserts a single line break
	<button>	Defines a push button autofocus, disabled, form, formaction, formenctype, formmethod, formnovalidate, formtarget, name, type, value
★	<canvas>	Defines graphics height, width
	<caption>	Defines a table caption
✻		Used to center align text and content
	<cite>	Defines a citation
	<code>	Defines computer code text autobuffer, autoplay, controls, src
	<col>	Defines attributes for table columns
	<colgroup>	Defines groups of table columns span
★	<command>	Defines a command button checked, disabled, icon, label, radiogroup, type
★	<datalist>	Defines a dropdown list
	<dd>	Defines a definition description
	<del>	Defines deleted text cite, datetime
★	<details>	Defines details of an element open
★	<dialog>	Defines a dialog (conversation)
	<dfn>	Defines a definition term
✻	<dir>	Used to define a directory list
		Defines a section in a document
	<dl>	Defines a definition list
	<dt>	Defines a definition term
		Defines emphasized text
★	<embed>	Defines external interactive content or plugin height, src, type, width
	<fieldset>	Defines a fieldset disabled, form, name
★	<figure>	Defines a group of media content, and their caption
✻	<font>	Used to define font face, font size, and font color of text
★	<footer>	Defines a footer for a section or page
	<form>	Defines a form accept-charset, action, autocomplete, enctype, method, name, novalidate, target
	<frame>	Used to define one particular window (frame) within a frameset
✻	<frameset>	Used to define a frameset, which organized multiple windows (frames)
	<h1> to <h6>	Defines header 1 to header 6
	<head>	Defines information about the document
★	<header>	Defines a header for a section or page
★	<hgroup>	Defines information about a section in a document
	<hr>	Defines a horizontal rule
	<html>	Defines an html document manifest, xmlns
	<i>	Defines italic text
	<iframe>	Defines an inline sub window height, name, sandbox, seamless, src, width
	<img>	Defines an image alt, src, height, ismap, usemap, width
	<input>	Defines an input field accept, alt, autocomplete, autofocus, checked, disabled, form, formaction, formenctype, formmethod, formnovalidate, formtarget, height, list, max, maxlength, min, multiple, name, pattern, placeholder, readonly, required, size, src, step, type, value, width
	<ins>	Defines inserted text cite, datetime
★	<keygen>	Defines a generated key in a form autofocus, challenge, disabled, form, keytype, name
	<kbd>	Defines keyboard text
	<label>	Defines an inline sub window for, form
	<legend>	Defines a title in a fieldset
		Defines a list item value
	<link>	Defines a resource reference href, hreflang, media, rel, sizes, type
	<map>	Defines an image map name
★	<mark>	Defines marked text
	<menu>	Defines a menu list label, type
	<meta>	Defines meta information charset, content, http-equiv, name
★	<meter>	Defines measurement within a predefined range high, low, max, min, optimum, value
★	<nav>	Defines navigation links
✻	<noframes>	Used to display text for browsers that do not handle frames
	<noscript>	Defines a noscript section
	<object>	Defines an embedded object data, form, height, name, type, usemap, width
		Defines an ordered list reversed, start
	<optgroup>	Defines an option group label, disabled
	<option>	Defines an option in a drop-down list disabled, label, selected, value
★	<output>	Defines some types of output for, form, name
	<p>	Defines a paragraph
	<param>	Defines a parameter for an object name, value
	<pre>	Defines preformatted text
	<progress>	Defines progress of a task of any kind max, value
	<q>	Defines a short quotation cite
★	<rp>	Used in ruby annotations to define what to show browsers that to not support the ruby element
★	<rt>	Defines explanation to ruby annotations
★	<ruby>	Defines ruby annotations
✻	<s>, <strike>	Used to define strikethrough text.
	<samp>	Defines sample computer code
	<script>	Defines a definition list async, type charset defer, src
★	<section>	Defines a section cite
	<select>	Defines a selectable list autofocus, disabled, form, multiple, name, size
	<small>	Defines small text
★	<source>	Defines media resources media, src, type
		Defines a section in a document
		Defines strong text
	<style>	Defines a style definition type, media, scoped
	<sub>, <sup>	Defines sub/super-scripted text
	<table>	Defines a table summary
	<tbody>	Defines a table body summary
	<td>	Defines a table cell colspan, headers, rowspan
	<textarea>	Defines a text area autofocus, cols, disabled, form, maxlength, name, placeholder, readonly, readonly, required, rows, wrap
	<tfoot>, <thead>	Defines a table footer / head
	<th>	Defines a table header colspan, headers, rowspan, scope
	<time>	Defines a date/tim datetime
	<title>	Defines the document title
	<tr>	Defines a table row datetime
✻	<tt>	Used to define teletype text
✻	<u>	Used to define underlined text
		Defines an unordered list
	<var>	Defines a variable
★	<video>	Defines a video autobuffer, autoplay, controls, height, loop, src, width

★ HTML 5 new tag

✻ Tag not supported in HTML 5

Designed by Antonio Lupetti
(http://woorkup.com)
http://facebook.com/antoniolupetti
http://www.twitter.com/woork

THE ✻ WORKING BRAIN

Boxes
margin *
margin-top
margin-right
margin-bottom
margin-left

padding *
padding-top
padding-right
padding-bottom
padding-left

border *
border-top *
border-bottom *
border-right *
border-left *

border-color *
border-top-color
border-right-color
border-bottom-color
border-left-color

border-style *
border-top-style
border-right-style
border-bottom-style
border-left-style

border-width *
border-top-width
border-right-width
border-bottom-width
border-left-width

Positioning
display
position
top
right
bottom
left
float
clear
z-index
direction
unicode-bidi
overflow
clip
visibility

Dimensions
width
min-width
max-width
height
min-height
max-height
line-height
vertical-align

Miscellaneous
content
quotes
counter-reset
counter-increment
marker-offset
list-style *
list-style-type
list-style-image
list-style-position

Shorthand properties are marked with *

CSS

SYNTAX

```
/* Comment */
@media type {
    selector {
        property: values;
    }
}
```
(Media type optional)

SELECTORS

	Styles apply to:
*	All elements
div	
div *	Elements within
div span	within
div, span	and
div > span	with as parent
div + span	preceded by ...
.class	Elements of class "class"
div.class	of class "class"
#itemid	Element with id "itemid"
div#itemid	with id "itemid"
a[class]	<a> with class attribute
a[class='x']	<a> when class is "x"
a[class~='x']	<a> when class is a list of space-seperated values and one of those is 'x'
a[lang|='en']	<a> when lang begins with "en"

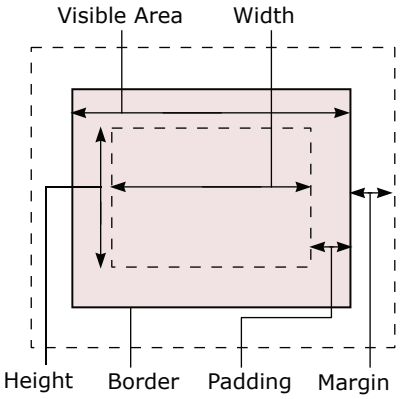
PSEUDO-SELECTORS

	Styles apply to:
:first-child	First child of element
:first-line	First line of element
:first-letter	First letter of element
:hover	Element when mouse over
:active	Active element
:focus	Element with focus
:link	Non-active, unvisited links without mouse over.
:visited	Visited links
:lang(lang)	Element with text of language "lang"

MEDIA TYPES (@media)

all	projection
braille	screen
embossed	speech
handheld	tty
print	tv

BOX MODEL



UNITS

px	Pixels
em	1em equal to font size of parent (same as 100%)
ex	Height of lower case "x"
%	Percentage
in	Inches
cm	Centimeters
mm	Millimeters
pt	1pt = 1/72in
pc	1pc = 12pt
#789abc	RGB Hex Notation
#acf	Equates to "#aaccff"
rgb(0,25,50)	Value (0 to 255) of each of red, green, and blue. May also be percentages
0	0 requires no unit

PROPERTIES THAT INHERIT

azimuth	list-style
border-collapse	list-style-image
border-spacing	list-style-position
caption-side	list-style-type
color	orphans
cursor	page
direction	page-break-inside
empty-cells	quotes
font	speak
font-family	speak-header
font-stretch	text-align
font-size	text-indent
font-size-adjust	text-transform
font-style	volume
font-variant	white-space
font-weight	widows
letter-spacing	word-spacing
line-height	

Paging
size
marks
page-break-before
page-break-after
page-break-inside
page
orphans
widows

Color / Background
color
* background
background-color
background-image
background-repeat
background-attachment
background-position

Fonts
* font
font-family
font-style
font-variant
font-weight
font-stretch
font-size
font-size-adjust

Text
text-indent
text-align
text-decoration
text-shadow
letter-spacing
word-spacing
text-transform
white-space

Tables
caption-side
table-layout
border-collapse
border-spacing
empty-cells
speak-header

Interface
cursor
* outline
outline-width
outline-style
outline-color

Aural
volume
speak
* pause
pause-before
pause-after
* cue
cue-before
cue-after
play-during
azimuth
elevation
speech-rate
voice-family
pitch
pitch-range
stress
richness
speak-punctuation
speak-numeral

JavaScript

Methods

Object
toString
toLocaleString
valueOf
hasOwnProperty
isPrototypeOf
propertyIsEnumerable

String
charAt
charCodeAt
fromCharCode
concat
indexOf
lastIndexOf
localeCompare
match
replace
search
slice
split
substring
substr
toLowerCase
toUpperCase
toLocaleLowerCase
toLocaleUpperCase

RegEx
test
match
exec

Array
concat
join
push
pop
reverse
shift
slice
sort
splice
unshift

Number
toFixed
toExponential
toPrecision

Date
parse
toDateString
toTimeString
getDate
getDay
getFullYear
getHours
getMilliseconds
getMinutes
getMonth
getSeconds
getTime
getTimezoneOffset
getYear
setDate
setHours
setMilliseconds
setMinutes
setMonth
setSeconds
setYear
toLocaleTimeString

XMLHttpRequest

Safari, Mozilla, Opera:

```
var req = new XMLHttpRequest();
```

Internet Explorer:

```
var req = new
ActiveXObject("Microsoft.XMLHTTP");
```

XMLHttpRequest Object Methods

abort()
getAllResponseHeaders()
getResponseHeader(header)
open(method, URL)
send(body)
setRequestHeader(header, value)

XMLHttpRequest Object Properties

onreadystatechange
readyState
responseText
responseXML
status
statusText

XMLHttpRequest readyState Values

0	Uninitiated
1	Loading
2	Loaded
3	Interactive
4	Complete

JAVASCRIPT IN HTML

External JavaScript File

```
<script type="text/javascript"
src="javascript.js"></script>
```

Inline JavaScript

```
<script type="text/javascript">
  <!--
    // JavaScript Here
  //-->
</script>
```

Functions

Window
alert
blur
clearTimeout
close
focus
open
print
setTimeout

Built In
eval
parseInt
parseFloat
isNaN
isFinite
decodeURI
decodeURIComponent
encodeURI
encodeURIComponent
escape
unescape

REGULAR EXPRESSIONS - FORMAT

Regular expressions in JavaScript take the form:

var RegEx = **/pattern/modifiers**;

REGULAR EXPRESSIONS - MODIFIERS

/g	Global matching
/i	Case insensitive
/s	Single line mode
/m	Multi line mode

REGULAR EXPRESSIONS - PATTERNS

^	Start of string
$	End of string
.	Any single character
(a|b)	a or b
(...)	Group section
[abc]	Item in range (a or b or c)
[^abc]	Not in range (not a or b or c)
a?	Zero or one of a
a*	Zero or more of a
a+	One or more of a
a{3}	Exactly 3 of a
a{3,}	3 or more of a
a{3,6}	Between 3 and 6 of a
!(pattern)	"Not" prefix. Apply rule when URL does not match pattern.

EVENT HANDLERS

onAbort	onMouseDown
onBlur	onMouseMove
onChange	onMouseOut
onClick	onMouseOver
onDblClick	onMouseUp
onDragDrop	onMove
onError	onReset
onFocus	onResize
onKeyDown	onSelect
onKeyPress	onSubmit
onKeyUp	onUnload
onLoad	

FUNCTIONS AND METHODS

A method is a type of function, associated with an object. A normal function is not associated with an object.

DOM Methods

Document
clear
createDocument
createDocumentFragment
createElement
createEvent
createEventObject
createRange
createTextNode
getElementsByTagName
getElementById
write

Node
addEventListener
appendChild
attachEvent
cloneNode
createTextRange
detachEvent
dispatchEvent
fireEvent
getAttributeNS
getAttributeNode
hasChildNodes
hasAttribute
hasAttributes
insertBefore
removeChild
removeEventListener
replaceChild
scrollIntoView

Form
submit

DOM Collections
item

Range
collapse
createContextualFragment
moveEnd
moveStart
parentElement
select
setStartBefore

Style
getPropertyValue
setProperty

Event
initEvent
preventDefault
stopPropagation

XMLSerializer
serializeToString

XMLHTTP
open
send

XMLDOM
loadXML

DOMParser
parseFromString

JSON CHEAT SHEET

WHAT'S JSON?

JSON (or JavaScript Object Notation) is an attempt to model reality in a way that can be understood by humans and parsed by many programming languages. In doing so, it brings disparate technologies together such as your database, middleware, and frontend, who can all work with JSON in one form or another.

ARRAYS

Below is an array written in JSON notation. Again, this should be familiar.

```
var my_dog = [
  "Greg",
  "Westie",
  ["Mr. Greg","Gregs"],
  8,
  10,
  23,
  "i.imgur.com/j0QxJqy.jpg"
]
```

Accessing Array Elements...

my_dog[0]	returns "Greg"
my_dog[2][0]	returns "Mr. Greg"
my_dog[3]	returns 8

OBJECTS

Below is an object written in JSON notation. If you're familiar with JavaScript, this shouldn't look too unusual. The primary syntactical difference is that the key must be stored as a string using quotes.

```
var my_dog = {
  "name":"Greg",
  "breed":"Westie",
  "nicknames":["Mr. Greg", "Greg
s"],
  "age":8,
  "height":10,
  "weight":23,
  "photo":"i.imgur.com/j0QxJqy.j
pg"
}
```

Accessing Object Properties...

my_dog.name	returns "Greg"
my_dog["breed"]	returns "Westie"
my_dog[3]	returns 8

by Elizabeth Blackburn and Jan Dennison

for Dev Bootcamp Localhost : Sea Lion Cohort, Feb 2015

JSON.STRINGIFY()

The stringify method serializes a JavaScript value, converting it to a JSON string. For a detailed explaination of the optional arguments, see stringify()

Here's a simple my_dog object written in JavaScript.

```
var my_dog = {
  name: "Greg",
        breed: "Westie",
        age: 8
}

dog_string = JSON.stringify(my_d
og);

console.log(dog_string);

//The above will have the follow
ing console output:
{"name":"Greg","breed":"Westie",
"age":8}
```

For cleaner output, you can pass a number of spaces as the third argument.

```
dog_string = JSON.stringify(my_d
og, null, 2);

console.log(dog_string);

//The above will have the follow
ing console output:

{ "name": "Greg",
  "breed": "Westie",
  "age": 8
}
```

JSON.PARSE()

The parse method deserializes a JavaScript value, returning the object corresponding to the given JSON string. For a detailed explaination of the optional arguments, see parse()

Let's return to our original example, the my_dog object written in JSON notation. This time, we'll leave it formatted as it might be when passed as a string.

```
var my_dog = '{
  "name": "Greg",
  "breed": "Westie",
  "nicknames": ["Mr. Greg", "Gre
gs"],
  "age": 8,
  "height": 10,
  "weight": 23,
  "photo": "i.imgur.com/j0QxJqy.
jpg"
}'
```

As you can see, the value of my_dog is stored as a string and all of the keys are strings.

```
dog_parsed = JSON.parse(my_dog);

console.log(dog_parsed);

//The above will have the follow
ing console output:

{ name: 'Greg',
  breed: 'Westie',
  nicknames: ['Mr. Greg', 'Gregs
'],
  age: 8,
  height: 10,
  weight: 23,
  photo: 'i.imgur.com/j0QxJqy.jp
g' }
```