Web Designer's Font Card

Here are fonts supported by 50% of all browsers, or 75% of Macs, or 75% of Linux Systems

January 2004 Edition — Web browser font support for PC, Mac, Linux

99.5% of all browsers support Arial. 85% of Linux systems support Arial. Nearly 100% of Macs. Most of the text samples are 12px. The alphabet samples are in the universal default font-size: 16px (16 pixels high). The top 14 font names are in 32px.

Font	Code	Sample text	Sizes
Arial (sans-serif: meaning this is the default sans-serif font on PCs)	<span style="font-family:Arial,sans-serif">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." –H. Phelps Gates	9px 10px 11px 12px / 18px 16px 14px
Times New Roman (serif)	<span style="font-family:Times New Roman,serif">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." –H. Phelps Gates	9px 10px 11px 12px / 18px 16px 14px
Courier New (monospace)	<span style="font-family:Courier New,monospace">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." –H. Phelps Gates	9px 10px 11px 12px / 18px 16px 14px
Verdana	<span style="font-family:Verdana,sans-serif">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." –H. Phelps Gates (45% of Linux systems support Verdana)	9px 10px 11px 12px / 18px 16px 14px
Courier (monospace, monospace) — 100% of Linux systems support Courier	<span style="font-family:Courier,monospace">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." –H. Phelps Gates	11px 22px
Arial Black	<span style="font-family:Arial Black,sans-serif">	16px 20px 24px 30px 36px	
Comic Sans MS	<span style="font-family:Comic Sans MS,sans-serif">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." –H. Phelps Gates	9px 10px 11px 12px / 18px 16px 14px
Trebuchet MS	<span style="font-family:Trebuchet MS,sans-serif">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." -H. Phelps Gates	9px 10px 11px 12px / 18px 16px 14px
Impact	<span style="font-family:Impact,sans-serif">	20px 24px 28px 32px 36px 40px	
Georgia	<span style="font-family:Georgia,serif">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." –H. Phelps Gates	9px 10px 11px 12px / 18px 16px 14px
Tahoma	<span style="font-family:Tahoma,sans-serif">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." –H. Phelps Gates	9px 10px 11px 12px / 18px 16px 14px
Arial Narrow	<span style="font-family:Arial Narrow,sans-serif">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." –H. Phelps Gates	9px 10px 11px 12px / 18px 16px 14px
Helvetica (sans-serif, sans-serif)	<span style="font-family:Helvetica,sans-serif">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." –H. Phelps Gates	9px 10px 11px 12px / 18px 16px 14px
Times (serif, serif) — 85%	<span style="font-family:Times,serif">	"Oft as not, in talk or script, we'll force the use of some long hard word, and with it, blunt the keen edge and dull the sharp point of what we want to say." –H. Phelps Gates	9px 10px 11px 12px / 18px 16px 14px

Alphabet samples (each font):

ABCDEFGHIJKLMNOPQRSTUVWXYZ!@#$%^&*()
abcdefghijklmnopqrstuvwxyz1234567890
_+-={|[:";'<>?,./¶²°•£¥€©¿áèîõü“”«—...

Linux Courier, Mac Courier, PC Courier

These font names are in 24px.

Font	Sample
MS Sans Serif	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Century Gothic	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Lucida Console	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Garamond	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Lucida Sans Unicode	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Bookman Old Style	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Book Antiqua	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Palatino Linotype	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Terminal	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Microsoft Sans Serif	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Haettenschweiler	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin (20px)
Monotype Corsiva	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin (16px)
Script	"Each friend represents a world in us, a world possibly not born until they arrive..." —Anaïs Nin (28px)
Modern	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin (16px)
Franklin Gothic Medium	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Sylfaen (50%)	"Each friend represents a world in us, a world possibly not born until they arrive..." –Anaïs Nin
Papyrus (80%)	"That which hinders your task is your task." –Sanford Meisner
Arial Rounded MT Bold	"That which hinders your task is your task." –Sanford Meisner (20px)
Brush Script MT	"That which hinders your task is your task." –Sanford Meisner
Andale Mono	"That which hinders your task is your task." –Sanford Meisner
Palatino	"That which hinders your task is your task." –Sanford Meisner
Geneva	"That which hinders your task is your task." –Sanford Meisner
Gill Sans	"That which hinders your task is your task." –Sanford Meisner
Monaco	"That which hinders your task is your task." –Sanford Meisner
Optima	"That which hinders your task is your task." –Sanford Meisner
Chicago	"That which hinders your task is your task." –Sanford Meisner
Skia	"That which hinders your task is your task." –Sanford Meisner
Futura	"That which hinders your task is your task." –Sanford Meisner
Baskerville	"That which hinders your task is your task." –Sanford Meisner
Lucida Grande	"That which hinders your task is your task." –Sanford Meisner
Copperplate	"That which hinders your task is your task." –Sanford Meisner
Hoefler Text (15% of all browsers; 90% of Macs support Hoefler Text)	"That which hinders your task is your task." –Sanford Meisner
Herculanum	"That which hinders your task is your task." –Sanford Meisner
Apple Chancery	"That which hinders your task is your task." –Sanford Meisner
Helvetica Neue	"That which hinders your task is your task." –Sanford Meisner
New York	"That which hinders your task is your task." –Sanford Meisner (16px)
Techno	"That which hinders your task is your task." –Sanford Meisner
Zapfino	"That which hinders your task is your task." –Sanford Meisner (16px)
Marker Felt	"That which hinders your task is your task." –Sanford Meisner
Didot	"That which hinders your task is your task." –Sanford Meisner
Charcoal	"That which hinders your task is your task." –Sanford Meisner
American Typewriter	"That which hinders your task is your task." –Sanford Meisner
Big Caslon	"That which hinders your task is your task." –Sanford Meisner
Utopia	"That which hinders your task is your task." –Sanford Meisner
New Century Schoolbook	"That which hinders your task is your task." –Sanford Meisner
Lucida	"That which hinders your task is your task." –Sanford Meisner
Lucidabright	"That which hinders your task is your task." –Sanford Meisner
Lucidatypewriter	"That which hinders your task is your task." –Sanford Meisner
Fixed	"That which hinders your task is your task." –Sanford Meisner
Charter	"That which hinders your task is your task." –Sanford Meisner
ClearlyU (3%; 80%)	"That which hinders your task is your task." –Sanford Meisner (16px)

Characters

There are hexadecimal character codes as well as decimal. For example **>** is the same as **>** for a **>**. Decimal codes are never longer than hexadecimal.

See also the special characters for URL's on page 10, such as **%3E** for a **>**

Name	Char	Numeric codes	PC keystroke
†	†	† †	
‡	‡	‡ ‡	
¶	¶	¶	Alt-20
¦	¦	¦	
§	§	§	
¼	¼	¼	Alt-172
½	½	½	Alt-171
¾	¾	¾	
⁄	⁄	⁄	
¹	¹	¹	
²	²	²	Alt-253
³	³	³	
ª	ª	ª	Alt-166
º	º	º	Alt-167
•	•	• •	
·	·	·	Alt-250
°	°	°	Alt-248
‰	‰	‰ ‰	
″	″	″	
′	′	′	
¢	¢	¢	Alt-155
£	£	£	Alt-156
¤	¤	¤	
¥	¥	¥	Alt-157
€	€	€ €	
ƒ	ƒ	ƒ ƒ	
™	™	™ ™	
®	®	®	
©	©	©	

Name	Char	Numeric codes	PC keystroke
'	'	'	
"	"	"	
&	&	&	
<	<	<	
>	>	>	
“	“	“ “	
”	”	” ”	
„	„	„ „	
‘	‘	‘ ‘	
’	’	’ ’	
‚	‚	‚ ‚	
«	«	«	Alt-174
»	»	»	Alt-175
‹	‹	‹ ‹	
›	›	› ›	
(tilde key)	~		
˜	˜	˜ ˜	
¨	¨	¨	
ˆ	ˆ	ˆ ˆ	
´	´	´	
¸	¸	¸	
¯	¯	¯	
–	–	– –	
—	—	— —	
‾	‾	‾	
…	…	… …	

These four characters must be encoded for HTML (E.g. **>** instead of > by itself). Implementation diversity.

Name	Char	Numeric codes	PC keystroke
À	À	À	
Á	Á	Á	
Â	Â	Â	
Ã	Ã	Ã	
Ä	Ä	Ä	Alt-142
Å	Å	Å	Alt-143
Æ	Æ	Æ	Alt-146
Ç	Ç	Ç	Alt-128
Ð	Ð	Ð	
È	È	È	
É	É	É	Alt-144
Ê	Ê	Ê	
Ë	Ë	Ë	
Ì	Ì	Ì	
Í	Í	Í	
Î	Î	Î	
Ï	Ï	Ï	
Ñ	Ñ	Ñ	Alt-165
Ò	Ò	Ò	
Ó	Ó	Ó	
Ô	Ô	Ô	
Õ	Õ	Õ	
Ö	Ö	Ö	Alt-153
Ø	Ø	Ø	
Œ	Œ	Œ Œ	
Š	Š	Š Š	
Ù	Ù	Ù	
Ú	Ú	Ú	
Û	Û	Û	
Ü	Ü	Ü	Alt-154
Ý	Ý	Ý	
Ÿ	Ÿ	Ÿ Ÿ	
	Ž	Ž	
Þ	Þ	Þ	
þ	þ	þ	
ß	ß	ß	Alt-225
¿	¿	¿	Alt-168
¡	¡	¡	Alt-173

Name	Char	Numeric codes	PC keystroke
à	à	à	Alt-133
á	á	á	Alt-160
â	â	â	Alt-131
ã	ã	ã	
ä	ä	ä	Alt-132
å	å	å	Alt-134
æ	æ	æ	Alt-145
ç	ç	ç	Alt-135
ð	ð	ð	
è	è	è	Alt-138
é	é	é	Alt-130
ê	ê	ê	Alt-136
ë	ë	ë	Alt-137
ì	ì	ì	Alt-141
í	í	í	Alt-161
î	î	î	Alt-140
ï	ï	ï	Alt-139
ñ	ñ	ñ	Alt-164
ò	ò	ò	Alt-149
ó	ó	ó	Alt-162
ô	ô	ô	Alt-147
õ	õ	õ	
ö	ö	ö	Alt-148
ø	ø	ø	
œ	œ	œ œ	
š	š	š š	
ù	ù	ù	Alt-151
ú	ú	ú	Alt-163
û	û	û	Alt-150
ü	ü	ü	Alt-129
ý	ý	ý	
ÿ	ÿ	ÿ	Alt-152
	ž	ž	

Name	Char	Numeric codes	PC keystroke
∼	∼	∼	
≅	≅	≅	
≈	≈	≈	
≠	≠	≠	
≡	≡	≡	
≤	≤	≤	
≥	≥	≥	
−	−	−	
±	±	±	Alt-241
×	×	×	
÷	÷	÷	Alt-246
√	√	√	
∫	∫	∫	
∂	∂	∂	
∝	∝	∝	
∞	∞	∞	
⊗	⊗	⊗	
⋅	⋅	⋅	
∅	∅	∅	
∈	∈	∈	
∋	∋	∋	
∉	∉	∉	
⊂	⊂	⊂	
⊃	⊃	⊃	
⊄	⊄	⊄	
⊆	⊆	⊆	
⊇	⊇	⊇	
∧	∧	∧	
∨	∨	∨	
∩	∩	∩	
∪	∪	∪	
♠	♠	♠	
♣	♣	♣	
♥	♥	♥	
♦	♦	♦	
◊	◊	◊	

Name	Char	Numeric codes	PC keystroke
⌈	⌈	⌈	
⌉	⌉	⌉	
⌊	⌊	⌊	
⌋	⌋	⌋	
⟨	〈	〈	
⟩	〉	〉	
∠	∠	∠	
⊥	⊥	⊥	
¬	¬	¬	Alt-170
∴	∴	∴	
∀	∀	∀	
∃	∃	∃	
∇	∇	∇	
∗	∗	∗	
⊕	⊕	⊕	
∏	∏	∏	
∑	∑	∑	
ℑ	ℑ	ℑ	
℘	℘	℘	
ℜ	ℜ	ℜ	
ℵ	ℵ	ℵ	
µ	µ	µ	Alt-230
ϖ	ϖ	ϖ	
ϒ	ϒ	ϒ	
ϑ	ϑ	ϑ	
←	←	←	
↔	↔	↔	
→	→	→	
↑	↑	↑	
↓	↓	↓	
↵	↵	↵	
⇐	⇐	⇐	
⇔	⇔	⇔	
⇒	⇒	⇒	
⇑	⇑	⇑	
⇓	⇓	⇓	

Name	Char	Code	Name	Char	Code
Α	Α	Α	α	α	α
Β	Β	Β	β	β	β
Γ	Γ	Γ	γ	γ	γ
Δ	Δ	Δ	δ	δ	δ
Ε	Ε	Ε	ε	ε	ε
Ζ	Ζ	Ζ	ζ	ζ	ζ
Η	Η	Η	η	η	η
Θ	Θ	Θ	θ	θ	θ
Ι	Ι	Ι	ι	ι	ι
Κ	Κ	Κ	κ	κ	κ
Λ	Λ	Λ	λ	λ	λ
Μ	Μ	Μ	μ	μ	μ
Ν	Ν	Ν	ν	ν	ν
Ξ	Ξ	Ξ	ξ	ξ	ξ
Ο	Ο	Ο	ο	ο	ο
Π	Π	Π	π	π	π
Ρ	Ρ	Ρ	ρ	ρ	ρ
			ς	ς	ς
Σ	Σ	Σ	σ	σ	σ
Τ	Τ	Τ	τ	τ	τ
Υ	Υ	Υ	υ	υ	υ
Φ	Φ	Φ	φ	φ	φ
Χ	Χ	Χ	χ	χ	χ
Ψ	Ψ	Ψ	ψ	ψ	ψ
Ω	Ω	Ω	ω	ω	ω

IE4 and N4 do not support the greek alphabet. All other browsers do (including Konqueror).

To make a symbol appear on your web page, enter its &____; code into your HTML. Codes with no colors or marks are universal.

W3.org alone — Most users won't see the yellow-shaded characters. Though HTML 4.01 defines them, IE4-6 Windows and N4 don't support them, at least not consistently on all operating systems. Netscape 6-7 and Mozilla support them all. IE5 Mac, Opera Mac, Opera Windows, and Safari support most (those not marked -IE, -OpMac, -OpWin, -Sf). Konqueror supports only the ones marked Kq.

Not Netscape — Netscape 4 doesn't support these name-codes. Konqueror supports most (exceptions marked -Kq). All other browsers support them.

Not Version 4 — IE4, N4, and Konqueror 3 don't support these characters. Except Konqueror does support the Greek alphabet. IE5 supports some of them, IE5.5-6, N6-7, Mozilla and Safari support all.

Not W3.org — Numeric codes ‚ to Ÿ mutated from Microsoft character set windows-1252. They are still universally supported by browsers but w3.org standards and Unicode have entirely different plans for these codes. Most browsers support the 4-digit alternatives.

Not W3.org, but almost universal — W3.org deprecates; all browsers but IE5Mac and N4 support.

PC keystrokes — Many PC programs support these ancient keystrokes. Most also support the modern codes by adding a 0-prefix: **Alt-0181** is the same as **Alt-230** for directly entering µ. (May need NumLock.)

<!-- comments -->
<!DOCTYPE html vers dtd>

<a hyperlink
href=url (hyperlink)
target=
frame-name
_top (whole browser)
_blank (popup)
_self (same frame)
_parent (outer frame)
_search (IE5) >
title=hover-hint (tool tip)

<a name=aname> id
(aname is URL's #-suffix)

<abbr> (italics Kq, Sf)

<acronym> (italics Kq, Sf)

<address> italics <i>

<applet Java
contains <param>'s
code=url.class <object>
align=left|right|... (as img)
archive=java-archive-url
codebase=folder-url
width=n pix|n% of frame
height=n pix|n% of frame
hspace=n pixels } outside
vspace=n pixels } margin
name=applet-name
mayscript>

<area clickable img
inside <map>
shape=rect (rectangle)
coords=lft, top, rgt, btm
shape=circle
coords=xctr, yctr, radius
shape=poly (polygon)
coords=x1,y1, x2,y2, ...
shape=default (backgrnd)
href=url (hyperlink)
target=frame-name
alt=description, nongraphic browsers (tool tip, title)
nohref />

<b> boldface
{font-weight:bold}

<base inside <head>
href=folder-url (basis for all hyperlinks)
target=frame-name />

<basefont inside <body>
before any text {font}
size=same as <font size>
face=font-typeface
color=#rrggbb />

<bdo reading direction
dir=ltr|rtl>(N6,-Op,-Kq,-Sf)

<bgsound in <head> (Op)
src=url.mid|url.wav|...
loop=n cycles (-1=forever)
volume=
-10000 (mute) | 0 (full)
balance=
-10000 | 0 | +10000 />

HTML Parts

start-tag <h3> sunlit uplands </h3> end-tag
content
element

attribute its value

some elements never have an end-tag
XHTML requires this "/" when there's no end-tag.

⌘ XHTML requires attribute values be quoted (' or "). The HTML Card leaves out these quote marks for brevity. HTML 4.01, and all browsers today, require quotes only for a value with spaces or exotic punctuation (other than hyphen, underscore or colon).

<big> bigger font ×4
{font-size:larger}

<blink> (N7, Op)

<blockquote indent
{margin-left} to indent
cite=url>

<body inside <html>
bgcolor=#rrggbb
{background-color}
link=#rrggbb a:link{}
vlink=#rrggbb a:visited{}
alink=#rrggbb a:active{}
text=#rrggbb {color}
background=image-url
{background-image}
bgproperties=fixed
{background-attachment}
topmargin=n pixels
leftmargin=n pixels
rightmargin=n pixels
bottommargin=n pixels
marginwidth=n pixels
marginheight=n pixels
scroll=yes|no|auto (N7,Op)
nowrap>

<br hard line-break
clear=none|left|right|all
(end of img flow-around)
{clear} />

<button> (N6)
type=button|submit|reset
name=submitted-name>
visible-label</button>

<caption inside <table>
align=top|bottom|left|center|right
valign=top|bottom (above or below table) />

{text-align:center}

<cite> italics <i>

<code> monospace (-Kq)

<col|colgroup
span=n columns
width=n|n%|n*|*|0*
align } as <td>
valign }
bgcolor=#rrggbb />

<comment> <!-- -->

<dd indented definition
nowrap> in <dl>

<del strike-thru (N6)
datetime=YYYY-MM-DD...
cite=url>

<dfn italics (-Kq,-Sf) (N6)

<dir> (-Op)

<div attribute
 mouthpiece
align=left|center|right|justify {text-align}
style=CSS-expression...
class=style-name
id=id
nowrap>

<dl term-definition list
contains <dt><dd>
compact> <blockquote>

<dt outdented term
nowrap> in <dl>

 italics = <i>

<embed plugin (e.g. RealAudio®, Flash)
contains <param>
src=data-url <object>
pluginspage=program-url
width=n pix|n% of frame
height=n pix|n% of frame
align=left|right|top|bottom
type=MIME-type />

<fieldset box around
with <legend> {border}
align=left|right|top> (N6)

<font
size= {font-size}
1 (smallest)
3
7 (largest)
−n (smaller)
+n (larger)
face= {font-family}
comma-separated list:
serif
sans-serif
monospace (-Kq)
Verdana
Arial Black
Trebuchet MS,sans-serif
Georgia,Times,serif
Font Card
color=#rrggbb
{color}
point-size=n
weight=n (100-900)>

<form contains
<input><textarea>
<select><button> ...
action=submit-url
target=submit-frame-name
method=get|post
enctype= (MIME-type)
application/x-www-form-urlencoded
multipart/form-data
name=form-name id
accept=MIME-type
accept-charset=charset
autocomplete (IE5)>

<frame sub-window
inside <frameset>
src=page-url
name=frame-name
scrolling=no|auto|yes
noresize (IE5.5)
width=n pix|n% of frame
height=n pix|n% of frame
allowtransparency (IE5.5)
marginwidth=n pixels
marginheight=n pixels
frameborder=0|1
bordercolor=#rrggbb />

<frameset
in place of <body>
or in another <frameset>
cols|rows= comma-list of:
n pixels
n% of frame
n* (proportional)
* (remainder)
border=n } all 0 for
framespacing=n } seamless
frameborder=0|1 frames
bordercolor=#rrggbb>
set target attribute for all <a href>'s in the frame to _top if nothing else

section headings
<h1|h2|h3|h4|h5|h6>
align=left|center|right>

<head inside <html>
contains <title> <meta>
<style> <script> <link> ...
profile=url >

<hr horizontal line
align=left|center|right
size=n pixels thick =2
width=n pix|n% of frame
noshade (flat) CSS...
color=#rrggbb />

<html outermost tag
xml:lang=human-language
xmlns=XML-namespace>

<i> italics
{font-style:italic}

<iframe include pg
src=url.html (N6)
width=n pix|n% of frame
height=n pix|n% of frame
hspace=n pixels } outside
vspace=n pixels } margin
allowtransparency (IE5.5)
align=left|right|... (as img)
name=frame-name
scrolling=no|auto|yes
marginwidth=n pixels
marginheight=n pixels
frameborder=0|1
bgcolor=#rrggbb>

<ilayer> see <layer>

Flow-around image alignment

An image with align=left or align=right hugs the margin. Subsequent text flows around that image. But be careful using both too near each other; on some browers the images can overlap. <br clear=all> or the {clear} style stops this flow-around. Text after that point goes below the image, margin-to-margin.

align=left
vspace
hspace
align=right
vspace

Font-like image alignment

Époopée!

align=absmiddle align=absbottom align=bottom align=top

<img graphics
src=url.gif|url.jpg|...
width=n pix|n% of frame
height=n pix|n% of frame
hspace=n pixels } outside
vspace=n pixels } margin
border=n pixels 0
alt=description (tool tip)
for non-graphic browser
align=left (flow around)
right {float}
align=top
texttop } same
absmiddle (place image within the text
middle
bottom } same like a
baseline } font)
absbottom
{vertical-align}
usemap=url#map-name
ismap (submits x,y click pt)
lowsrc=image-url
dynsrc=video-url
loop=n cycles (-1=forever)
longdesc=page-url />

<input N4 requires in a <form>
name=invisible to user
value=usu. visible to user ("name=value" pairs are submitted with the form)
type=text (1-line) abc
<textarea>
size=n (visible chars)
maxlength=n (typable)
autocomplete (IE5)
value is default for user
type=password
size=n (visible)
maxlength=n (typable)
type=checkbox
checked default for user
value is submitted to server if user leaves checked
type=radio
(one of multiple choices)
checked user default
value is choice's name
(within mutually exclusive groups use same names, different values)
type=file Browse...
(upload)
size=n (visible chars)
accept=MIME-type
type=hidden (from user)
type=button PUSH
value is button's label
type=reset Reset
(clear all fields)
type=submit Submit
(send form to server)
type=image submit button with custom appearance
src=url.gif|url.jpg|...
align=as <img align>
width=n pixels|n%
height=n pixels|n%
border=n pixels
name=hover hint
alt=descr. (hover hint)
usemap=url#map-name
ismap (send x,y click pt)
id=input-id (for label) />

<ins underline (N6)
datetime=YYYY-MM-DD...
cite=url.html>

<isindex /> † inside <head>
<input type=text>

<kbd> monospace (-Kq)

<keygen> † cryptography

<label (N6) expanded
for=input-id> click zone

<layer(abs)|ilayer(rel) (-N6)
src=url.html
left=n pixels } absolute or
top=n pixels } relative
pagex=n pixels
pagey=n pixels
z-index=n (high on top)
above=layer-id } alterna-
below=layer-id } tive to
id=layer-id (this) } z-index
width=n pix|n% of frame
height=n pix|n% of frame
clip=left,top,right,bottom
visibility=show|hidden
bgcolor=#rrggbb
background=url.gif|... />

<legend
inside <fieldset> CSS
align=top|center|bottom>

<li numbered or bulleted paragraph
inside <ol> or
type=1|a|i|A|I
type=disc|circle|square
(as in <ol> and <ul>)
value=n (alter sequence)>

<link file relation
inside <head>
rel=stylesheet } by far the
type=text/css } most com-
href=url.css } mon link
rel=Shortcut Icon (N7, IE5)
href=favicon.ico
rel=(other relationships)
contents|next|prev|...
rev=(reverse relation)
type=MIME-type
media=screen|print|all
target=frame-name
charset=character-set />

<listing> (-Kq) dense
<pre> monospace

<map image's clicks
contains <area>'s
name=map-name>

<marquee text in motion
width=n pix|n% (N7,Kq,Sf)
height=n pix|n%
hspace=n pixels } outside
vspace=n pixels } margin
scrollAmount=n pix } speed
scrollDelay=n msec } tink-
truespeed } ering
loop=n cycles (-1=forever)
behavior=
scroll|alternate|slide
direction=left|right|up|down
bgcolor=#rrggbb>

<menu> list

<meta inside <head>
name=description
content=description
name=keywords
content=kwd,kwd,kwd...
name=robots (aka spiders)
content=
NOINDEX (not for search engines)
NOFOLLOW (ignore links)
NOARCHIVE (don't cache)
NOINDEX, NOFOLLOW } same
NONE
ALL
name=generator
content=softw-version
http-equiv=content-type
content="text/html; charset=charset"
http-equiv=pragma
content=nocache
http-equiv=expires (cache)
content=0|date
http-equiv=refresh (jump to new page)
content="nsec; url=url"
http-equiv=content-language
content=en|en-gb|fr|en-us|jp|da|...
http-equiv=pics-label
content=offense-rating
(see www.icra.org)
scheme=scheme />

<multicol text into columns
cols=n columns
gutter=n pixel gap, =10
width=n pix per column>

<nobr> no word wrap
contains <wbr>

<noembed> } for browsers
<noframes> } that don't
<nolayer> } support the
<noscript> } feature

<object (e.g. Macromedia® Flash™ animation)
contains <param>
classid=128-bit-ActiveX-id
codebase=url.cab#n.n.n.n
code=url.class (Java)
type=MIME-type (obj)
classid=url.jar (java archive)
type=MIME-type (obj)
classid=url
codebase=default-folder
codetype=MIME-type (obj)
type=MIME-type (data)
data=url
width=n pix|n% of frame
height=n pix|n% of frame
hspace=n pixels } outside
vspace=n pixels } margin
align=left|right|... as <img>
id=object-id> this part is active only if object tag is not supported</object>

<ol numbered list
contains 's
{list-style-type}
type=1 (1, 2, 3, 4, ...)
a (a, b, c, d, ...)
i (i, ii, iii, iv, ...
A (A, B, C, D, ...)
I (I, II, III, IV, ...)
start=n =1 >

<optgroup> (IE6,N6,-Op) indentation of <option>'s

<option in <select>
value=as-submitted
selected (default)
disabled (grayed-out) >
visible-choice</option>

<p paragraph
align=left|center|right|justify> {text-align}

<param inside <applet> <embed> <object>
valuetype=data|ref|object
name=name
value=value />

<plaintext> † mono-space
<pre> (-Kq)

<pre> (-Kq) monospace
<tt> like a .txt file

<q> "quote marks"

<rt> annotation (IE5)

<ruby> contains <rt> (IE5)

<s> strike-through
{text-decoration: line-through} =<strike>

<samp> monospace (-Kq)

<script JavaScript...
language= type
javascript|jscript
javascript 1.2
vbscript|vbs
xml (IE5)
type=(MIME type)
text/javascript
text/vbscript
text/xml (IE5)
src=url.js (script in a file as opposed to the content)
for=object-id
event=eventname
defer (faster load)
charset=charset>

<select multiple-choice drop-down
inside <form>
contains <option>'s
name=field-name
size=n (rows before open)
multiple simult. answers
type= (IE4)
select-one
select-multiple>

<small> smaller font ×2
{font-size:smaller}

<spacer (-N6)
1x1 transparent GIF img
type=block
width=n pixels|n%
height=n pixels|n%
align=left|right as img
type=vertical
size=n pixels
type=horizontal
size=n pixels>

<span attribute
 mouthpiece
style=CSS-expression...
class=style-name
id=id
xml:lang=language-code>

<strike> strike-thru
{text-decoration: line-through} =<s>

 bold <b>

<style CSS
inside <head>
media=screen|print|all
type= (MIME-type)
text/css
text/javascript>
cssprop:value;...</style>

<sub subscript 2

<sup superscript 2

Table parts and colors

<table
contains <tr>
<colgroup><col><caption>
<thead> <tfoot> <tbody>
{float} for layout
border=n pixels =0 (around outside of table)
{border-width}
cellspacing=n pix =2 to 4 (around & between cells)
cellpadding=n pixels =1 (extra room inside cells)
{padding}
bgcolor=#rrggbb
{background-color}
background=url.gif|url.jpg
{background-image}
bordercolor=#rrggbb
bordercolorlight=#rrggbb
bordercolordark=#rrggbb
width=n pix|n% of frame
height=n pix|n% of frame
hspace=n pixels } outside
vspace=n pixels } margin
align=left|center|right
{float}
cols=n virtual columns
frame=(around table)
void|box|border
above|below|lhs
rhs|hsides|vsides
rules=(between cells)
none|all|groups|
cols|rows
summary=summary>

<tbody> see <thead>

border
cellspacing
cellpadding
td bgcolor
table / body bgcolor
table bordercolor

<td table cell
inside <tr>
align=left|center|right|justify {text-align}
align=char } e.g. align
char=.|,|... } by decimal
charoff=n } points
valign=top|middle|bottom|baseline
{vertical-align}
bgcolor=#rrggbb
{background-color}
background=url.gif|url.jpg
{background-image}
bordercolor=#rrggbb
colspan|rowspan=
1
2...n (wide col, tall row)
0 (remainder of group)
nowrap (all on one line)
{white-space:nowrap}
width=n pix|n% of frame
{width}
height=n pix|n% of frame
{height}
headers=id, id, ...
scope=row|col|...
abbr=brief <img alt>
axis=categories>

<textarea multi-line text entry
inside <form>
<input type=text>
cols=visual-width (no limit)
rows=visual-lines to data)
wrap= wrap=soft
soft visually wraps, submits data unwrapped
hard visually wraps, submits with hard returns
off window scrolls rather than wrap text
readonly (N6)
name=field-name>
default value</textarea>

<th header cell same as <td> with bold contents

<thead tfoot tbody
(define in this order)
row grouping, improves pagination and scrolling
align } same values
valign } as <td>
bgcolor=#rrggbb>

<title> (inside <head>)
heading in browser window also becomes bookmark or favorite name </title>
avoid: & : / | \ ? * " < >

<tr table row
inside <table>
contains <td> <th>
align } as <td>
valign }
width=n pix|n% of frame
height=n pix|n% of frame
bgcolor=#rrggbb
bordercolor=#rrggbb>

<tt> monospace
<pre> (-Kq)
{font-family:monospace}

<u> underline
{text-decoration: underline}

<ul bulleted list
contains 's
{list-style-type}
type=disc 1st level
circle 2nd level
square 3rd, 4th...
compact>

<var> italics <i>

<wbr /> soft line-break
inside <nobr>

<xml> data island (IE5)

<xmp> monospace (-Kq)

Style Attributes
style=prop:val; prop:val; ...
CSS property values
class=class class ... as in: .class {prop:val}
id=id as used in CSS: #id {prop:val}
⌘ XHTML prefers id over name, suggesting both where compatibility is improved, such as in <a> <form> <img> <map>

Universal Attributes
you can use with any tag
lang=human-language
xml:lang
xml:lang=human-language
title=hover-hint (tool tip)
tabindex=n =0
accesskey=key shortcut
N7: Alt-(accesskey)
IE4: Alt-(accesskey), Enter
Op7: Shift-Esc-(accesskey)
language=script-language
atomicselection (IE5.5)
(get focus as a whole)
unselectable=on|off
hidefocus (IE5.5)
contenteditable= (IE5.5)
inherit|false|true
begin=begin-time (IE5.5)
syncmaster (IE5.5)

Event Attributes
DOM Pages

onload
onunload
onresize
onscroll
onclick
ondblclick
onkeydown
onkeyup
onkeypress
onmouseover
onmousemove
onmousedown
onmouseup
onmouseout
onchange
onerror
onfocus
onblur
onsubmit
onreset
ondragstart
ondrag
ondragend
ondragenter
ondragover
ondragleave
ondrop
oncopy
oncut
onpaste

{azimuth: *sound direction*
0deg | center | 90deg | right-side | 180deg | behind | -90deg | left-side | ... ;}

{background: *color image repeat attachment position*;}

{background-color: *color* | transparent;} *only on <body>, <td>, *

{background-image: *(-Op)*
none | url(*url.gif* | *url.jpg*);
background-repeat:
repeat-x | repeat-y | repeat | no-repeat;
background-attachment:
scroll | fixed; *(background motion when scrolling)*
background-position:
npx | ... | *nn%* | top | center | bottom ; *npx* | ... | *nn%* | left | center | right
background-position-x:
n (pixels);
background-position-y:
n (pixels);}

{behavior: url(*url.htc*) | url(#*object-id*);}

{border: *line(s) around* *extra margin or padding* *color style width*;} *(IE5.5)*

{border-color: *as {color}*
color color color color
transparent;}

{border-top-color | *(N6)*
border-right-color | **border-bottom-color** | **border-left-color:**
color;} *same as {color}*

{border-style:
style style style style ;}

{border-top-style | *(N6)*
border-right-style | **border-bottom-style** | **border-left-style:**
none
solid
hidden
dotted
dashed
double
groove
ridge
inset
outset;}

{border-width:
width width width width ;}

{border-top-width | *(N6)*
border-right-width | **border-bottom-width** | **border-left-width:**
npx | *npt* | *nem* | *nn%* |...;}

{border-collapse: *(IE5,N7)*
between cells *(-Sf,-Kq)*
collapse *(thin flat lines)*
separate *(ridged lines)*;}

{border-spacing: *(N6)*
between cell borders
npx | *npt* | *nem* | *nn%* |...;}

{bottom} *see {position}*

{caption-side:
top | bottom | left | right;}

{clear: =<*br clear*>
(end of image flow-around)
left | right | both | none;}

{clip: *visual limits*
auto | rect(*top,right,bottom,left*);}

{color: *text foreground*
#*rrggbb* *(00-FF)*
#*rgb* *(0-F)*
rgb(*rrr,ggg,bbb*) *(0-255)*
rgb(*rr%,gg%,bb%*) *(0-100)*
black | white | red | blue | ...
activeborder | activecaption | buttonface | buttonhighlight | buttonshadow | buttontext | captiontext | graytext | highlight | highlighttext | inactiveborder | inactivecaption | inactivecaptiontext | infobackground | infotext | menu | menutext | scrollbar | threeddarkshadow | threedface | threedhighlight | threedlightshadow | threedshadow | window | windowframe | windowtext ;}

{content} *(used with :before and :after pseudo-elements)*

{counter-increment:
name number;}

{counter-reset:
name number;}

{cue | **cue-after** | **cue-before:**
url(*url.wav* | *url.au* |...);}

{cursor: *when hovering*
auto *(N6)*
crosshair
default *(arrow)* *(N6)*
hand *(indicating clickability)*
pointer *(N6)*
move
n|ne|e|se|s|sw|w|nw-resize
text *(I-bar)* *(N6)*
wait *(hourglass)* *(N6)*
help *(arrow with '?')*
url(*url*) ;}

{direction: ltr | rtl}

{display: *appears-as*
inline *as *
block *as *
list-item *as *
none *vanishes, takes up no space on screen*
{*visibility*:hidden}
run-in *anti-* *(stuffs a heading in 1st line of ¶)*
compact
marker
table *as <table>*
inline-table
inline-block *(IE5.5)*
table-header-group
table-footer-group
table-row-group
table-row *as <tr>*
table-column-group
table-column
table-cell *as <td>*
table-caption ;}

{elevation: *sound angle*
n (degrees) | above *(90)* | level *(0)* | below *(-90)* |... ;}

{empty-cells: show | hide;}

{filter: progid:Image Transform.Microsoft.
Alpha(opacity=*percent*)
BasicImage(
grayScale=0 | 1,
invert=0 | 1, *(color neg)*
mirror=0 | 1,
opacity=*x.xx*, *(1.0 to 0.0)*
rotation=0 | 1 | 2 | 3,
XRay=0 | 1)
Blur *MotionBlur()*
Chroma(color=#*rrggbb*)
Compositor(function=*0-25*)
DropShadow(color,offX,offY)
Emboss(bias=*plusminus1*)
Engrave(bias=*plusminus1*)
FlipH *BasicImage()*
FlipV *BasicImage()*
Glow(color,strength=*npix*)
Gray *BasicImage()*
ICMFilter(colorSpace,intent)
Invert *BasicImage()*
Light() *(spotlight)*
mask *MaskFilter()*
MaskFilter(color=#*rrggbb*)
MotionBlur(strength=*npix*, direction=*angle360*)
Shadow(color,direction, strength=*npix*),
Wave(freq,phase,strength, lightStrength)
Xray *BasicImage()* ;}
(Search msdn.microsoft.com for "css filter")

{float: *flow text around*
=<*img align*>
left | right | none ;}

{font:
optional: bold | italic | oblique | small-caps
required: *size* | *size/hgt* ; *req:* *family*

tiny, readable system fonts:
caption *(-Sf,Op7)*
icon *(-Kq,-Sf,Op7)*
menu *(-Kq,-Sf,Op7)*
message-box | small-caption | status-bar ;} *(-Kq,-Sf, Op7,IE5.5)*

example font settings:
{*font*:bold italic 18px serif}
{*font*:10px/20px Arial}
{*font*:small monospace}
{*font*:caption}

{font-family: =<*font face*>
prioritized comma-separated list of font names:
Arial | Helvetica | Times New Roman | Courier New | Verdana | Trebuchet MS | Georgia | Comic Sans MS *designed for CRT screens by Microsoft*
serif | sans-serif | monospace *(-Kq)* *include one at end of the list*
fantasy *(N6,IE5.2,-Kq)*
cursive ;} *(IE5.2,-Kq)*
Font Card

{font-size:
=<*font size*>
{*line-height*}
npx | *npt* |... *px*
larger =<*big*>
smaller =<*small*>

typical sizes:

	N4	*N7*	*IE6*
xx-small	*9px*	*11px*	*10px*
x-small	*11px*	*13px*	*13px*
small	*14px*	*16px*	*16px*
medium	*16px*	*18px*	*18px*
large	*24px*	*22px*	*24px*
x-large	*36px*	*27px*	*32px*
xx-large;}	*54px*	*36px*	*48px*

{font-size-adjust:
(helps when mixing fonts)
n (aspect value, e.g. 0.5) ;}

{font-stretch: *(horizontally)*
ultra-condensed
extra-condensed
condensed
semi-condensed
normal
semi-expanded
expanded
extra-expanded
ultra-expanded
wider
narrower ;}

{font-style:
normal
italic =<*i*>
oblique *(N6)* ;}

{font-variant:
normal
small-caps *(N6,-Sf)* ;}

{font-weight:
normal
bold =<*b*>
lighter
bolder
100-400 *(normal)*
500-600
700-800 *(bold)*
900 ;}

{height: =<*td height*>
auto | *npx* |... | *nn%* ;}

{ime-mode: *(Asian text)*
input method editor
disabled | inactive | active | auto ;}

{layout-grid: *(Asian text)*
mode type char line ;}

{layout-grid-mode:
none | line | char | both ;}

{layout-grid-type:
loose | strict | fixed ;}

{layout-grid-char:
none | auto | *npx* | *nn%* |... ;}

{layout-grid-line:
none | auto | *npx* | *nn%* |... ;}

{left} *see {position}*

{letter-spacing *kerning*
{*word-spacing*} *(N6)*
+*npx*|... *wide)* *(Op6.03)*
−*npx*|... *tight)*
normal *(free to justify)* ;}

{line-break: *(Japanese text)*
normal | strict ;}

{line-height: *leading*
normal | *npx* | *nn%* |... ;}

{list-style:
type position image position;}

{list-style-image: *(N6)*
custom bullet
url(*url.gif*)|url(*url.jpg*)|... ;}

{list-style-position:
inside *(bullets indented)*
outside ;} *(N6,IE5)*

{list-style-type:
=<*ol type*><*ul type*>
disc
circle
square *(Opera:* □ *)*
decimal *1, 2, 3, 4, ...*
lower-alpha *a, b, c, d, ...*
upper-alpha *A, B, C, D, ...*
lower-roman *i, ii, iii, iv, ...*
upper-roman *I, II, III, IV, ...*
lower-latin *a,b,...* *(N6,-Op)*
upper-latin *A,B,...* *(N6,-Op)*
hebrew *(N6,-Op)*
lower-greek *(N6)*
decimal-leading-zero
armenian *(Op)*
georgian *(Op)*
cjk-ideographic
hiragana *(N6)*
katakana *(N6)*
hiragana-iroha *(N6)*
katakana-iroha *(N6)*
none ;} *(invisible bullets)*

{margin *beyond border*
(same background as whatever block it's in)
{*padding*}
width width width width;}

{margin-top | **margin-right** | **margin-bottom** | **margin-left:**
auto | *npx* | *nn%* |... ;}

{marker-offset}
see :before and :after

{marks} *see @page*

{max-height | **max-width:**
npx |... | *n%* | none ;}

{min-height | **min-width:**
npx |... | *n%* ;}

{orphans:*n lines* ;}

{outline: *(Kq,Sf)*
same as {border} in Kq,Sf
color style width ;}

{overflow: *when block isn't big enough...*
hidden *(clip at block edges)*
scroll *(scroll bars, always)*
visible *(expand block)* auto *(scroll bars, if nec.)* ;}

{overflow-x | **overflow-y:**
hidden | scroll | visible | auto;}

{padding: *around content*
(for <td> only)
{*margin*}
width width width width ;}

{padding-top | **padding-right** | **padding-bottom** | **padding-left:**
npx | *nn%* |... ;}

{page:*page-name*;}

{page-break-before:
always
auto *(for printing only)*
empty-string *nix new page*
avoid
left
right ;}

{page-break-inside:
auto | avoid ;}

{pause: *silence*
before after ;}

{pause-before | **pause-after:**
nms | *ns* | *n%* *(of word)* ;}

{pitch:
n Hz (120=male 210=female)
x-low | low | medium | high | x-high ;}

{pitch-range: *deviation*
n Hz ;}

{play-during:
url(*url.wav*)
url(*url.wav*) mix
url(*url.wav*) repeat
auto
none ;}

{position:static;}
(embed within text flow)

{position:relative;
(offset from static position)
top: *npx* | *n%* |... ;
right: *npx* | *n%* |... ; *(IE5)*
bottom: *npx* | *n%* |...; *(N6)*
left: *npx* | *n%* |... ;
z-index: *n* | auto ;} *+front*

{position:absolute;
(offset in container box)
top: *npx* | *n%* |... ;
right: *npx* | *n%* |... ; *(N6)*
bottom: *npx* | *n%* |...;
left: *npx* | *n%* |... ;
z-index: *n* | auto ;} *+front*

{position:fixed; *(no scroll, or appears on printed pages)*
top: *npx* | *n%* |... ; *(IEMac)*
bottom: *npx* | *n%* |... ;
z-index: *n* | auto ;}

{quotes: *symbols to use*
open close open close ...;}

{richness: *of voice*
n (0=soft, 100=carries) ;}

{right} *see {position}*

{ruby-align: *for <ruby>*
auto | left | center | right | distribute-letter | distribute-edge | line-center ;}

{ruby-overhang:
auto | whitespace | none ;}

{ruby-position:
above | inline ;}

{scrollbar-3dlight-color | **scrollbar-arrow-color** | **scrollbar-base-color** | **scrollbar-darkshadow-color** | **scrollbar-face-color** | **scrollbar-highlight-color** | **scrollbar-shadow-color** | **scrollbar-track-color:***color*;}

{size} *see @page*

{speak:
normal | none | spell-out ;}

{speak-header:
once | always ;}

{speak-numeral:
digits | continuous ;}

{speak-punctuation:
code | none *(pause)* ;}

{speech-rate:
n (words per minute)
x-slow | slow *(120)* | medium | fast *(300)* | x-fast ;}

{src} *see @font-face*

{stress: *syllable accent*
n (100=max, 0=none) ;}

{table-layout: *(IE5)*
auto *(best fit)*
fixed *(fastest render)* ;}

{text-align: =<*td align*><*p align*><*center*>
left | center | right | justify ;}

{text-autospace: *(mixed Asian and non-Asian text)*
none
ideograph-alpha
ideograph-numeric
ideograph-parenthesis
ideograph-space ;}

{text-decoration:
none
overline *(N6)*
line-through =<*s*>
underline =<*u*>
blink =<*blink*> ;}

{text-indent: *npx* | *nn%* |...;}
line 1 only {*margin-left*}
=<*blockquote*>

{text-justify: *details for* {*text-align:justify*}
auto | distribute | distribute-all-lines | inter-cluster | inter-ideograph | inter-word | kashida | newspaper ;}

{text-kashida-space:
n ;} *(for Arabic characters)*

{text-shadow: *drop shadow*
color dx dy rblur ;}

{text-transform:
capitalize | uppercase | lowercase | none ;}

{text-underline-position:
below | above ;}

{top} *see {position}*

{unicode-bidi: normal | embed | bidi-override ;}

{vertical-align:
top | middle | baseline | bottom *applies to table cell contents* =<*td valign*>
sub | super | text-bottom | text-top ;} *(N6)* *applies in-line*

{visibility:
visible
hidden *(takes up space)* {*display*:none} *(N6)*
collapse ;} *(N6, IEMac)*

{voice-family:
male | female | child | ... ;}

{volume:
n% | silent | x-soft *(0%)* | soft | medium *(50%)* | loud | x-loud *(100%)* ;}

{white-space:
normal
nowrap *(all on one line)* =<*td nowrap*> *(N6)*
pre *(monospace, like a .txt file)* =<*pre*> ;} *(-Kq)*

{widows:*n lines* ;}

{width: =<*td width*>
auto | *npx* |... | *n%* ;}

{word-break: *(for mixed Asian and non-Asian text)*
break-all | keep-all ;} *(IE5)*

{word-spacing:
(IE Mac only) *(N6)*
{*letter-spacing*}
+*npx* ... *wide)* *(justification*
−*npx* ... *tight*) *won't change)*
normal *(free to justify)* ;}

{word-wrap: *(IE5.5)*
break-word *(breaks words longer than a whole line)*
normal ;}

{writing-mode: *(IE5.5)*
lr-tb *(horizontal text)*
tb-rl *(vertical text, 90°cw)* ;}

{z-index} *see {position}*

{zoom: *(IE5.5)*
n (1.0=normal 2=out .5=in)
n% *(100%=normal)*
normal ;}

/* comments */

Style Sheet
@-Rules

@charset ISO-8859-1 |...

@import url(*url.css*); *(N6)*
styles from a file
=<*link rel=stylesheet*>

@font-face *downloadable*
{font-family: *font-family*;
src:url(*url.pfb* | *url.ttf* |...);
many more descriptors ;}

(N6) **@media** { all | screen | print | ... } {*media-specific rules*}

@page { **:first** | **:left** | **:right** | *page-name* } *printed page layout*
{**size:** *width height* | auto | landscape | portrait ;
margins:...; *as {margin}*
marks: crop | cross | none;}

Style Sheet
Distance Units

px pixels
pt points (1/72 in)
pc picas (1/6 in)
em current {*font-size*}
ex height of a small x
in inches
mm millimeters
cm centimeters

Style Sheet
Pseudo-Classes
Pseudo-Elements

a:link{*prop:val;...*} =<*body link*>
a:visited{*prop:val;...*} =<*body vlink*>
a:active{*prop:val;...*} =<*body alink*> *(N6)*
element:**active**{*prop:val;...*}
a:hover{*prop:val;...*} *(N6)* *(mouse over hyperlink)*
element:**hover**{*prop:val;...*}
element:**focus**{*prop:val;...*} *(seen when tabbing)* *(N6)*
element:**before** *(prefix)*
element:**after** *(suffix)* *(N6)*
{**content:**
string
url(*url*)
counter() *(for markers not hits)*
counters()
open-quote *(defined in*
close-quote *{quotes}*
no-open-quote *prop-*
no-close-quote *erty)*
attr(*attribute-name*);
display:marker; *e.g. bullet*
marker-offset:
auto | *npx* |... ;}
element:**first-letter** *of a ¶* {*property:value;...*} *(N6)*
element:**first-line** *of a ¶* {*property:value;...*} *(N6)*
element:**lang**(*human-lang*)
element:**first-child**
element:**first** *see @page*
element:**left** *see @page*
element:**right** *see @page*

Three examples of CSS in a web page

Apple tree.

Lime zest.

by tag — *http://www.visibone.com/css/x.css*

```
body {color:black; background-color:white;}
```

http://www.visibone.com/css/x.html

```
<html><head><title>CSS Examples</title>
<link rel="stylesheet" type="text/css" href="x.css" />
<style type="text/css">
 .fruity {color:red;}                    by class
</style></head><body>
 <p class="fruity">Apple tree.</p>       ad hoc span
 <p><span style="color:green;">Lime</span> zest.</p>
</body></html>
```

HTML → XHTML ***Migration Checklist*** *(getting compatible with* ***XML*** *- Extensible Markup Language)*

- ☐ Lower-case elements and attributes: <TD VALIGN="top"> → <td valign="top">
- ☐ Nasty Nesting → Nice Nesting
- ☐
 →
 Slashify empty elements (always put a space before the slash)
- ☐ paragraph → paragraph Endify nonempty elements, even if there's no content
- ☐ Quotify all attribute values: <ol type=a> → <ol type="a"> or <ol type='a'>, even empty values <img alt="">
- ☐ Don't <!-- comment --> style and script elements
- ☐ Don't use < or & or]]> or -- in style or script elements (but ok in <link>'d files) Doesn't this crimp your JavaScript!
- ☐ Encode & in URL's or other attribute values: <form action="go.cgi&level=1">
- ☐ No line breaks or multiple spaces in attribute values (inside the quotes)
- ☐ Use name and id together <a name="part" id="part"> (except radio buttons)
- ☐ Use lang and xml:lang together <html lang="en" xml:lang="en">
- ☐ Character set, specify it once: <?xml encoding="ISO-8859-1"?>
- ☐ Character set, specify it twice: <meta http-equiv="Content-type" content='text/html; charset="ISO-8859-1"' />

Number 2 1.5 2.5e3 0xFF 010

```
assert(2+2==4);          // numbers are 64-bit floating point
assert(1.5==3/2);        // (no separate integer type)
assert(2.5e3==2500);     // 2.5 x 10^3 exponential notation
assert(0xFF==255);       // hexadecimal
assert(010==8);          // octal
```

simple math

```
assert(  2+2==4);   // addition
assert( 10-3==7);   // subtraction
assert(  3*8==24);  // multiplication
assert(123/10==12.3); // real (not integer) division
assert(1234%100==34); // modulo (remainder)

var n=3;   n+=30; assert(n==33); // compute & store
var n=33;  n-=30; assert(n==3);  // x*=y is the same
var n=3;   n*=20; assert(n==60); // as x=x*y
var n=38;  n/=10; assert(n==3.8);
var n=38;  n%=10; assert(n==8);

assert(-3+3==0);  // negative number (unary minus)
var n=3;  n++; assert(n==4);  // increment
var n=3;  n--; assert(n==2);  // decrement
```

comparisons

```
assert( 99<100);  // less than
assert( 99<=100); // less than or equal
assert(100>99);   // greater than
assert(100>=99);  // greater than or equal
assert(100==100); // equal
assert( 99!=100); // not equal
```

32-bit math

```
assert(1000<<3==8000); // shift left
assert(1000>>3==125);  // shift right, signed ±=31-bit
assert(0xFFFF0000>>>8==0x00FFFF00); // unsigned
// Always use parentheses around terms with: & | ^
assert((0x55555555 & 0xFF00FFFF)==0x55005555); // and
assert((0x55555555 | 0x00FF0000)==0x55FF5555); // or
assert((0x55555555 ^ 0x00FF0000)==0x55AA5555); // xor
// >>>0 converts to unsigned, avoiding sign extension
assert((~0x55555555)>>>0==0xAAAAAAAA); // 1's compl.
assert((~0x55555555)      !=0xAAAAAAAA); // is signed!

var n=0x555;  n&=0x0F0;  assert(n==0x505);
var n=0x555;  n|=0x0F0;  assert(n==0x5F5);
var n=0x555;  n^=0x0F0;  assert(n==0x5A5);
var n=-10;    n<<=1;  assert(n==-20); // shift left
var n=-10;    n>>=1;  assert(n==-5);  // signed right
var n=0x8;    n>>>=1; assert(n==0x4); // unsigned
```

```
assert(Number.MIN_VALUE < 1e-307); // special
assert(Number.MAX_VALUE > 1e308);  // numbers
assert(Number.NEGATIVE_INFINITY==1/0);
assert(Number.POSITIVE_INFINITY==-1/0);
assert(isNaN(0/0)); // NaN stands for Not a Number
assert(0/0!=0/0);   // NaN is not equal to itself!
assert(!isFinite(1/0));  assert(isFinite(1));
```

Math Math.PI Math.max() Math.round()

```
assert(Math.abs(-3.2)==3.2);

assert(Math.max(1,2)==2 && Math.max(1,2,3,4)==4);
assert(Math.min(1,2)==1 && Math.min(1,2,3,0)==0);

assert(0<=Math.random() && Math.random()<1);

assert(Math.ceil(1.5)==2);  // round up, to the nearest
assert(Math.ceil(-1.5)==-1); // integer higher or equal
assert(Math.round(1.7)==2); // round to the nearest
assert(Math.round(1.2)==1); // integer, up or down
assert(Math.floor(1.5)==1); // round down to the nearest
assert(Math.floor(-1.5)==-2); // integer lower or equal

var n;
n=Math.E;       assertApprox(Math.log(n),1);
n=Math.LN10;    assertApprox(Math.pow(Math.E,n),10);
n=Math.LN2;     assertApprox(Math.pow(Math.E,n),2);
n=Math.LOG10E;  assertApprox(Math.pow(10,n),Math.E);
n=Math.LOG2E;   assertApprox(Math.pow(2,n),Math.E);
n=Math.PI;      assertApprox(Math.sin(n/2),1);
n=Math.SQRT1_2; assertApprox(n*n,0.5);
n=Math.SQRT2;   assertApprox(n*n,2);

assertApprox(Math.acos(1/2),Math.PI/3); // trig functions
assertApprox(Math.asin(1/2),Math.PI/6); // are in radians
assertApprox(Math.atan(1),Math.PI/4);
assertApprox(Math.atan2(1,1),Math.PI/4);
assertApprox(Math.cos(Math.PI/3),1/2);
assertApprox(Math.exp(1),Math.E);
assertApprox(Math.log(Math.E),1); // (base e, not 10)
assertApprox(Math.pow(10,3),1000);
assertApprox(Math.sin(Math.PI/6),1/2);
assertApprox(Math.sqrt(25),5);
assertApprox(Math.tan(Math.PI/4),1);
```

```
// Math functions are accurate to 15 digits:
function assertApprox(a,b) {
    assert((b*0.999999999999999 < a) &&
           (a < b*1.0000000000000001));
}
```

Number ⇔ String conversions

```
// First, a subtle distinction in JavaScript comparisons:
assert(3 == '3');  // == Equals  flexible about type
assert(3 != '4');
assert(3=== 3);    // === Identical  must be the
assert(3 !== '3');                   same type

assert(256 == '256');  // Strings in a numeric context are
assert(256.0 == "256");// converted to a number. This is
assert(256 == "256.0");// usually reasonable and useful.
assert("256" != "256.0"); // (String contexts, no convert!)
assert(256 == "0x100");// Hexadecimal 0x prefix works,
assert(256 == "0256"); // but no octal 0 prefix this way.
assert(256 != "256 xyz"); // No extraneous characters.

// Number ⇨ String
assert(256 === "256" - 0); // – converts string to number
assert("2560" === "256" + 0); // + concatenates strings
assert(256 === parseInt("256"));
assert(256 === parseInt("256 xyz"));  // extras forgiven
assert(256 === parseInt("0x100"));    // hexadecimal
assert(256 === parseInt("0400"));     // 0 for octal
assert(256 === parseInt("0256",10));  // certain decimal
assert(256 === parseInt("100",16));   // hexadecimal
assert(256 === parseInt("400",8));    // octal
assert(25.6 === parseFloat("2.56e1"));
assert("256" === "256".valueOf()); // (no conversion help)
assert(isNaN(parseInt("xyz")));    // gibberish handling
assert(isNaN(parseFloat("xyz")));

// Number ⇨ String, explicit conversions
assert(256 + ""          === "256");
assert((256).toString()  === "256");
assert((2.56).toString() === "2.56");
assert((256).toString(16)=== "100");  // 0 real
assert((2.56).toFixed()  === "3");
assert((2.56).toFixed(3) === "2.560");
assert((2.56).toPrecision(2) === "2.6");
assert((256).toExponential(4) === "2.5600e+2");
assert((1024).toLocaleString() === "1,024.00"); //!

// Exotic numbers convert to strings in precise ways:
assert((-1/0).toString()  === "-Infinity");
assert((0/0).toString()   === "NaN");
assert((1/0).toString()   === "Infinity");
```

Boolean true false

```
var t=true;   assert(t);
var f=false;  assert(!f);      // ! is boolean not
assert((true && false)==false); // && is boolean and
assert((true || false)==true);  // || is boolean or
assert((true ? 'a' : 'b')=='a'); // miniature if-else chooser
assert((false ? 'a' : 'b')=='b'); // (parentheses outside)
```

String 'abc' "abc" "line\u000D\u000A"

```
var s="string"; // double or single quotes, your choice
var s='string'; // but 'can't always use apostrophe's'
assert("str"+"ing"=="string"); // + concatenates
assert(s.length==6); // all strings have a length property
assert(s.charAt(0)=="s"); // and are indexed from zero
assert(s.charAt(5)=="g"); // no character type
assert(s[5]=="g");  // only Netscape indexes strings
assert(s.charCodeAt(5)==0x67); // ASCII character value
assert(String.fromCharCode(65,66,67)=="ABC");

assert(s.substring(2)=="ring");   // istart
assert(s.substring(2,4)=="ri");   // istart, iend+1
assert(s.substring(4,2)=="ri");   // iend+1, istart  } same
assert(s.substring(-2)=="string"); // (negative values
assert(s.substring(2,-2)=="st");  // are just like zero)
assert(s.slice(2)=="ring");   // istart
assert(s.slice(2,4)=="ri");   // istart, iend+1
assert(s.slice(-2)=="ng");  // – same as 0 before IE5.5
assert(s.slice(1,-1)=="trin");
assert(s.substr(2)=="ring");  // istart
assert(s.substr(2,2)=="ri");  // istart, inum
assert(s.substr(-2,2)=="ng");  // – same as 0 in IE

assert('abc'.toUpperCase()=='ABC');
assert('ABC'.toLowerCase()=='abc');
assert('abc'.toLocaleUpperCase()=='ABC');
assert('ABC'.toLocaleLowerCase()=='abc');

assert('str'.concat('ing')=='str'+'ing'); // two kindsa glue
assert(s.indexOf('ing')==3); // find substring, -1=can't
assert('strings'.lastIndexOf('s')==6); // find rightmost

// These involve ❖ Regular Expressions and/or ❖ Arrays
assert(/ing/.test(s));
assert(s.search(/ing/)==3);
assert('nature'.replace(/a/,'ur')=='nurture');
assert('a:b:c'.split(':').join('..')=='a..b..c');
assert('1-37-54'.match(/\d+/g).join()=='1,37,54');
RegExp.lastIndex=0;
assert(/o(.)r/.exec('courage').join()=='our,u');

// search expects a regular expression (where dot=any):
assert('imdb.com'.search(".")==0);     // so you must
assert('imdb.com'.search(/./)==0);     // not forget to
assert('imdb.com'.search(/\./)==4);    // double-escape
assert('imdb.com'.search("\\.")==4);   // your punctuation

s="\uFFFF"; // 16-bit hex Unicode      Slash
s="\xFF";   // hexadecimal ASCII       Characters
s="\377"; s="\77"; s="\7";  // 8-bit octal  16-bit
assert('\0'=="\u0000"); // NUL
assert('\b'=="\u0008"); // backspace (BS)
assert('\t'=="\u0009"); // tab (TAB)
assert('\f'=="\u000C"); // formfeed (FF)
assert('\r'=="\u0000"); // return (CR)
assert('\n'=="\u000A"); // newline (LF)
assert('\v'=="\u0008"); // vertical tab (VT)
assert('\"'=='"');      // double-quote
assert('\''=="'");      // single-quote
assert("\\"=="\u005C"); // a single backslash

// Multi-line strings (backslash works, plus is better)
s="this is a \
test"; // (comments not allowed on the line above)
assert(s=="this is a test"); // N4 inserts LF, Opera CR
s="this is a "+ // concatenate (it's a plus to have plus)
"better test";  // comments allowed on both of these lines
assert(s=="this is a better test");

// NUL isn't special, it's a character like any other
assert('abc\0def'.length==7); // Opera ignores \0, try
assert('abc\0def'!='abc\0xyz'); // String.fromCharCode(0)

// User-entered cookies or URLs must encode punctuation
assert(escape("that's all.")=="that%27s%20all.");
assert(unescape("that%27s%20all.")=="that's all.");
// These are escaped: *%<>[\]^`{|}#$&,:;=?!'()~
// plus space. Alphanumerics and these are not: *-._+/@
// encodeURI() translates  %<>[\]^`{|}
// encodeURIComponent() %<>[\]^`{|}#$&+,/:;=?
// decodeURI() and decodeURIComponent() the inverse
```

Duplicate definitions are harmless, the latter prevails.

```
var a=1  // Lines don't have to end with ; semicolons,
var b=2; // but using them consistently shows character.
// Multiple statements on the same line require semicolons
var c=3; c+=a; c+=b;  assert(c==6); // between them.
```

Array [1,'abc',new Date(),['x','y'],true]

```
var a=new Array;  // container of numbered things
assert(a.length==0);  // they begin with zero elements
a=new Array(8);       // unless you give them dimension
assert(a.length==8);
assert(a[0]==null);  // indexes are from 0 to length-1
assert(a[7]==null);  // uninitialized elements are null
assert(a[20]==null); // out-of-range elements equal null
a[20]='21st element'; // but writing out of range
assert(a.length==21); // just makes an array bigger

a[0]='a'; a[1]='cat'; a[2]=44;  // three identical
a=new Array('a','cat',44);      // ways to fill
a=['a','cat',44];               // up an array
assert(a.length==3);
assert(a[0]=='a' && a[1]=='cat' && a[2]==44);

assert([1,2,3]!=[1,2,3]); // Array compare is tricky.
// (Tech reason: objects compare by reference, not value.)
assert([1,2,3].join()=="1,2,3"); // So we'll use join() a lot
// to work with arrays because strings compare by value.

assert(a.join()=="a,cat,44"); // join turns array into string
assert(a.join("/")=="a/cat/44"); // default comma delimited

a="a,cat,44".split();  // split parses string into array
assert(a.join()=="a,cat,44");  // (we use join() to prove it)
a="a-cat--44".split("-");      // split() also defaults
assert(a.join("+")=="a+cat+44"); // to comma delimited

a="pro@sup.net".split(/[.\@]/); // split can also use a
assert(a.join()=="pro,sup,net"); // ❖ regular expression

// split("") turns a string into an array of characters
assert("the end".split("").join()=="t,h,e, ,e,n,d");

a=[2,36,111]; a.sort(); // case-sensitive string sort
assert(a.join()=="111,2,36");
a.sort(function(a,b) { return a-b; });  // numeric order
assert(a.join()=="2,36,111");
// Sort function should return –,0,+ signifying <,==,>
assert("a".localeCompare("z")<0); // sort function

a=[1,2,3]; a.reverse();  assert(a.join()=="3,2,1");
a=[1,2,3]; assert(a.pop()==3); assert(a.join()=="1,2");
a=[1,2,3]; a.push(4);  assert(a.join()=="1,2,3,4");
a=[1,2,3]; assert(a.shift()==1); assert(a.join()=="2,3");
a=[1,2,3]; a.unshift(0);  assert(a.join()=="0,1,2,3");
a=[1,2,3]; // splice(iStart, nDelete, xInsert1, xInsert2,...)
a.splice(2,0,'a','b'); assert(a.join()=="1,2,a,b,3");  // insert
a.splice(1,2);  assert(a.join()=="1,b,3");  // delete
a.splice(1,2,'z'); assert(a.join()=="1,z"); // insert & delete
var aleft=[1,2,3], aright=[4,5,6], aboth=aleft.concat(aright);
assert(aboth.join()=="1,2,3,4,5,6");

// slice(istart,iend+1) creates a new subarray
assert([6,7,8,9].slice(0,2).join()=="6,7");  // istart,iend+1
assert([6,7,8,9].slice(1).join()=="7,8,9");  // istart

assert([6,7,8,9].slice(1,-1).join()=="7,8");  //length added
assert([6,7,8,9].slice(-3).join()=="7,8,9");  // to – values
```

Function function zed() { return 0; }

```
function sum(x,y) {          // definition
    return x+y;              // return value
}
var n=sum(5,5);  assert(n==10);  // call

function sum1(x,y) { return x+y; };  // 3 ways to
var sum2=function(x,y) { return x+y; }  // define a
var sum3=new Function("x","y","return x+y;"); //function
assert(sum1.toString()==   // reveals definition code, but
  "function sum1(x,y) { return x+y; }"); // format varies

function sumx() {          // Dynamic arguments
    var retval=0;
    for (var i=0 ; i < arguments.length ; i++) {
        retval+=arguments[i];
    }
    return retval;
}
assert(sumx(1,2)==3);
assert(sumx(1,2,3,4,5)==15);
```

Date Date() new Date(1999,12-1,31,23,59)

```
var dNow=new Date();  // seize the present moment
var dPast=new Date(2002,5-1,20,23,59,59,999);
// (year,month-1,day,hours,minutes,seconds,milliseconds)

assert(dNow.getTime() > dPast.getTime());
// Compare dates only by their getTime() or valueOf()
assert(dPast.getTime()==1021953599999);
assert(dPast.getTime()==dPast.valueOf());
// Compute elapsed milliseconds by subtracting getTime()'s
var nHours=(dNow.getTime()-dPast.getTime())/(3600000);

// Example date and time formats:  all vary widely)
assert(dPast.toString()=='Mon May 20 23:59:59 EDT 2002');
assert(dPast.toGMTString()==
                        'Tue, 21 May 2002 03:59:59 UTC');
assert(dPast.toUTCString()==
                        'Tue, 21 May 2002 03:59:59 UTC');
assert(dPast.toDateString()==          'Mon May 20 2002');
assert(dPast.toTimeString()==          '23:59:59 EDT');
assert(dPast.toLocaleDateString()==
                        'Monday, 20 May, 2002');
assert(dPast.toLocaleTimeString()==    '23:59:59 PM');
assert(dPast.toLocaleString()==
                        'Monday, 20 May, 2002 23:59:59 PM');

var d=new Date(0);   // Dates count milliseconds
assert(d.getTime()==0);  // after midnight 1/1/1970 UTC
assert(d.toUTCString()=='Thu, 1 Jan 1970 00:00:00 UTC');
assert(d.getTimezoneOffset()==5*60); // minutes west

// ⌘ terminology: getTime() is millisec after 1/1/1970
// getDate() is day of month, getDay() is day of week
// Same for setTime() and setDate(). There is no setDay().

d.setFullYear(2002);   assert(d.getFullYear()==2002);
d.setMonth(5-1);       assert(d.getMonth()==5-1);
d.setDate(31);         assert(d.getDate()==31);
d.setHours(23);        assert(d.getHours()==23);
d.setMinutes(59);      assert(d.getMinutes()==59);
d.setSeconds(59);      assert(d.getSeconds()==59);
d.setMilliseconds(999);
                       assert(d.getMilliseconds()==999);
assert(d.getDay()==5); // 0=Sunday, 6=Saturday
d.setYear(99);  assert(d.getYear()==99); // Y2K bugs
d.setYear(2001); assert(d.getYear()==2001);

d.setUTCFullYear(2002);
                       assert(d.getUTCFullYear()==2002);
d.setUTCMonth(5-1);    assert(d.getUTCMonth()==5-1);
d.setUTCDate(31);      assert(d.getUTCDate()==31);
d.setUTCHours(23);     assert(d.getUTCHours()==23);
d.setUTCMinutes(59);   assert(d.getUTCMinutes()==59);
d.setUTCSeconds(59);   assert(d.getUTCSeconds()==59);
d.setUTCMilliseconds(999);
                       assert(d.getUTCMilliseconds()==999);
assert(d.getUTCDay()==5); // 0=Sunday, 6=Saturday

// Most set-functions can take multiple parameters:
d.setFullYear(2002,5-1,31);   d.setUTCFullYear(2002,5-1,31);
d.setMonth(5-1,31);           d.setUTCMonth(5-1,31);
d.setHours(23,59,59,999);     d.setUTCHours(23,59,59,999);
d.setMinutes(59,59,999);      d.setUTCMinutes(59,59,999);
d.setSeconds(59,999);         d.setUTCSeconds(59,999);

// If you must call more than one set function, it's
// probably better to call the longer-period function first.
d.setMilliseconds(0); // (following point too coarse for msec)
// Date.parse() works on the output of either toString()
var msec=Date.parse(d.toString()); // or toUTCString().
assert(msec==d.getTime());  // The formats of
msec=Date.parse(d.toUTCString()); // those strings vary
assert(msec==d.getTime());  // one computer to another.
```

Client-side JavaScript can be many places

1. *Header:* `<head> <script> ▪▪▪ </script> </head>`
   Runs first before body is loaded.
2. *Include:* `<script src='http://url/filename.js'></script>`
   Text file with JavaScript code in it. (Better for XHTML.)
3. *Body:* `<body> <script> ▪▪▪ </script> </body>`
   Generate HTML with document.writeln(raw_html_string);
4. *Event:* `<element onevent=" ▪▪▪ ">`
   All HTML attributes that begin with "on" take event code.
5. *URL:* `<a href="javascript: ▪▪▪ ; void 0;">`
   Executes on click. All one line. (void 0 avoids page fetch.)
6. *Bookmarklet aka Favelet:* `javascript: ▪▪▪ ; void 0;`
   Savable browser utility, more at www.bookmarklets.com
7. *String:* `eval(" ▪▪▪ ");`
   Executes expression or code, returns the final result

Type Symbols

a	array
b	boolean
f	function
i	*integer* number
n	number
o	object
p	*property*
r	regular expression
s	string
T	*Type (constructor)*
x	*variable, any type*
...	*parameters*

JavaScript Literals

```
n=2;
n=2.1;
n=2.1e3;
n=0xFF;
n=010;
s="string";
s='string';
b=false;
b=true;
a=[x, x, x];
f=function(p) { statements; };
o={p:x, p:x, p:x};
r=/regular expression/;
```

```
x=o.p;
o=a[i];
x=o["p"];
x=f(...);
o=new T(...);
n=n++;
n=n--;
n+=n;
n-=n;
n*=n;
n/=n;
n=n%n;
n=n+n;
n=n-n;
s=s+s;
s=s+n;
n=n<<i;
n=n>>i;
n=n>>>i;
b=n < n;
b=n <= n;
b=n >= n;
b=n > n;
b=s < s;
b=s <= s;
b=s >= s;
b=s > s;
b=o instanceof T;
b=s in o;
b=n == n;
b=n != n;
b=n === n;
b=n !== n;
b=s == s;
b=s != s;
b=s === s;
b=s !== s;
b=x === x;
b=x !== x;
n=n&n;
n=n^n;
n=n|n;
b=b&&b;
b=b||b;
x=b ? x : x;
x=x;     n<<=n;
n*=n;    n>>=n;
n/=n;    n>>>=n;
n%=n;    n&=n;
n+=n;    n^=n;
n-=n;    n|=n;
s+=s;
x=x,x;
```

tighter ↑ operator binding strength ↓ looser

assertiveness

```
// Unit testing is the best advance in programming since
// subroutines were invented. See www.xprogramming.com.
// At the core is the assert() function (also called check()).
// The goals are faster fixes and focused, fearless progress.

// Here assert()'s do more: describe JavaScript in JavaScript.
// Why would you learn and lookup JavaScript using a book
// that's 95% English? As Berlitz knew, to live is to immerse.

// All the code in this reference not only runs but tests itself.
// Try it:  www.visibone.com/javascript/unittest.html
function assert(fact) {       // assert() can be very simple
  if (!fact) alert("Assert failure!");
}
function assert(fact,details) { // this helps to tell them apart
  if (!fact) alert("Assert failure! "+details);
}
function assert(fact,details) { // this assert shows the name
  if (!fact) {                  // of the function it's called from
    var msg="Assert failure! "+details;
    if (arguments.callee.caller != null) { // but not in Opera
      msg=msg+" in function "+
        arguments.callee.caller.toString().match(
          /function\s+(\w+)/)[1];
    }
    alert(msg);
  }
}
// JsUnit is an open source JavaScript unit test framework.
// It has very elaborate and useful assert()'s.  www.jsunit.net
```

decision if else switch case

```
function choose1(b) {          // if demo
    var retval="skip";
    if (b) {
        retval="if-clause";
    }
    return retval;
}
assert(choose1(true) == "if-clause");
assert(choose1(false) == "skip");

function choose2(b) {          // else
    var retval="doesn't matter";
    if (b) {
        retval="if-clause";
    }
    else {
        retval="else-clause";
    }
    return retval;
}
assert(choose2(true) == "if-clause");
assert(choose2(false) == "else-clause");

function choose3(n) {          // else-if
    var retval="doesn't matter";
    if (n == 0) {
        retval="if-clause";
    }
    else if (n == 1) {
        retval="else-if-clause";
    }
    else {
        retval="else-clause";
    }
    return retval;
}
assert(choose3(0) == "if-clause");
assert(choose3(1) == "else-if-clause");
assert(choose3(9) == "else-clause");

function choose4(s) {          // switch
    var retval="doesn't matter";
    switch (s) {   // switch on a
    case "A":      // string
        retval="A-clause";
        break;
    case "B":
        retval="B-clause";
        break;
    case "Czikszentmilaly":
        retval="Czikwhatever-clause";
        break;
    default:
        retval="default-clause";
        break;
    }
    return retval;
}
assert(choose4("A") == "A-clause");
assert(choose4("B") == "B-clause");
assert(choose4("Czikszentmilaly") == "Czikwhatever-clause");
assert(choose4("Z") == "default-clause");
```

```
// Logic structures aren't rigid about curly braces.
if (document.cookie != "") alert(document.cookie);
if (document.cookie != "") {     // But including them can
    alert(document.cookie);      // improve readability.
}
```

```
assert(eval("2+2") == 4);
if (typeof(Error) == "function") { // Safe exception catching
    eval("try { dangerous(); } catch (e) { oops(e); }");
}
```

Code outside of any function runs when the page loads.

loop for while do-while for-in

```
function dotsfor(a) {                  // for demo
    var s=""; // concatenate array into dot-interleaved string
    for (var i=0; i < a.length; i++) {
        s+=a[i]+".";
    }
    return s;
}
assert(dotsfor(["a","b","c"]) == "a.b.c.");

// A for-loop behaves exactly like a while-loop in this way:
// for (init; test; next) { doit; }   identical (assuming
// init; while (test) { doit; next; }  no continue's)

function dotswhile(a) {                // while demo
    var s="";                  // functionally identical the for-demo
    var i=0;
    while (i < a.length) {     // while is conservative:
        s+=a[i]+".";           // ask before act. Repeat as long
        i++;                   // as what's in parentheses holds true.
    }
    return s;
}
assert(dotswhile(["a","b","c"]) == "a.b.c.");

function uline(s,columnwidth) {        // do-while demo
    do {                               // surround with underbars
        s="_"+s+"_";
    } while (s.length < columnwidth);  // do-while is
    return s;   // impulsive: act first, then ask question.
}   // Repeat as long as what's in parentheses holds true.
assert(uline("Qty",7) == "__Qty__");
assert(uline("Description",7) == "_Description_");

// for, while, and do-while can cause infinite loops
function forever1() { for (;true;) { } }     // most browsers
function forever2() { while(true) { } }      // will timeout
function forever3() { do { } while(true); }  // eventually

// break escapes from the innermost for, while, do-while
// or switch clause, ignoring if and else clauses.
// continue skips to the test in for, while, do-while clauses.

var a=["x","y","z"], s="";             // for-in demo for arrays
for (var i in a) {                     //  i goes thru indexes, not elements
    s+=a[i];
}
assert(s == "xyz");
```

Error (exceptions) try catch finally throw

```
try {                          // catch an exception
    var v=nodef;
}
catch (e) {
    assert(e.message == "'nodef' is undefined");
    assert(e.name == "ReferenceError");
    assert(e.description == "'nodef' is undefined");
    assert(e.number > 0);
}

function process () {          // throw an exception
    if (somethingGoesVeryWrong()) {
        throw new Error("msg","msg");
    }   // To work with IE4 you could double the message,
    catch (e) { // then message or description should get it.
        assert(e.message == "msg" || e.description == "msg");
    }
}

function ReliableHandler() {   // finally is for sure
    try {
        initialize();
        process();
    }
    finally {
        shutdown();
    }
}
// If the try-clause starts, the finally-clause must also,
// even if an exception is thrown or the function returns.

// Almost all browsers except N4 & IE4 support exceptions.
// A trick to avoid JavaScript errors is to put your exception-
// handling functions in a .js file and include it twice:
<script language="JavaScript1.4" src="excfile.js">
<script language="EcmaScript" src="excfile.js">
// The first works for N6, the second for IE5 and IE6.
// Both work for Opera but that's probably harmless.
// (Above these, include a .js file with non-exception versions
// of the same functions, even if empty. Then calling is safe.)
```

SSI Server Side Includes (Apache servers with mod_include option "Options +Includes")

```
<!--#config timefmt="%a, %m/%d/%Y %H:%M:%S%p %Z" -->
<!--#echo var="DATE_LOCAL" -->
<!--#printenv -->  (gives a list of all available var's, on some servers)
<!--#exec cgi="url" cmd=command -->
<!--#exec cmd="program.cgi" -->
<!--#include virtual="relative-file-path" -->
<!--#set var="var-name" value="the-value" -->
<!--#if expr="string1 = string2" -->
<!--#elif expr="string1 != string2" -->
<!--#else -->
<!--#endif -->
```

Server tip: Give your page an .shtml extension and in your .htaccess add:

```
AddType text/html .shtml
AddHandler server-parsed .shtml
```

```
<!--#if expr="${HTTP_USER_AGENT} = /MSIE/" -->
    <!--#include virtual="ie.html" -->
<!--#else -->
    <!--#include virtual="non-ie.html" -->
<!--#endif -->
```

```
<!--#if expr="${DOCUMENT_URI} = \"/top.shtml\"" -->
    <!--#include virtual="../navbar.html" -->
<!--#endif -->
```

Always insert a space before the "-->"

expr operators: = != < <= > >= && || !

All characters that could possibly need to be escaped inside a URL – which's all but:

```
* - . _ @
```

%20	(space)
%21	!
%22	"
%23	#
%24	$
%25	%
%26	&
%27	'
%28	(
%29	)
%2B	+
%2C	,
%2F	/
%3A	:
%3B	;
%3C	<
%3D	=
%3E	>
%3F	?
%5B	[
%5C	\
%5D	]
%5E	^
%60	`
%7B	{
%7C	|
%7D	}
%7E	~

Online Names: URLs (Universal Resource Locators)

XHTML requires & instead of &

Web: http://www.on24.com /newsline/channels/top/ 1.htm ? caller=on24 & speed=fast

host domain name — *directory or path prefix* — *path* — *file* — *name value* — *name value*

http:// . . . . . # aname

Scroll target, scrolls the web page to the element <a name=aname> (case-sensitive in some browsers)

Parameters, here in MIME type format application/x-www-form-urlencoded, as results from <form method=get>, but this string could have other formats.

SSL: https:// . . . . *Encrypted server access*

Email: mailto:atc@npr.org ?Subject= Music%20Buttons

FTP: ftp://ftp.netscape.com /pub/communicator/complete_install/ cc32e473.exe

host — *directory or path prefix* — *path* — *file*

Local: file://localhost /c|/webstaging/index.html

file:///c|/Program%20Files/Real/RealPlayer/videotest.rm

There is a place for a host name after the second slash

Operating-system-specific path, shown here c:\Program Files\Real\RealPlayer\videotest.rm
replacing: :⟶| \⟶/ space⟶%20 #⟶%23 %⟶%25

Object: clsid:D27CDB6E-AE6D-11cf-96B8-444553540000

Microsoft's 128-bit identifier for everything, such as <object classid> for Macromedia Flash.

JavaScript: javascript:var agent=navigator.userAgent;alert('your browser is '+agent);

Object

```
var o=new Object();  // Objects are created with new

o.property = "value"; // Properties are created by assigning
assert(o.property == "value");
assert(o.nonproperty == null); // check if property exists
assert(!("nonproperty" in o));      // another way to check
assert("property' in o);

o.toString=function() { return this.property; } // Giving an
assert(o.toString() == "value"); // object a toString() method
assert(o == "value");  // allows direct string comparisons!

var o2=new Object(); o2.property="value";
assert(o != o2);

delete o.property; // remove a property from an object
assert(o.property == null);
// delete is for properties, not objects.  Objects are
// destroyed automagically (via garbage collection).

var B=new Boolean(true); assert(B);     // object aliases
var N=new Number(8);     assert(N == 8);  // for simple
var S=new String("stg");  assert(S == "stg");  // types

// An Object is a named array of properties and methods
o=new Object;  o.name="bolt";  o.cost=1.99;
o.costx2=function() { return this.cost*2; }
assert(o["name"]     == o.name);
assert(o["cost"]     == o.cost);
assert(o["costx2"]() == o.costx2());

// Object literals in curly braces with name:value pairs
o={name:"bolt", cost:1.99, sold:{qty:5, who:"Jim" } };
assert(o.name == "bolt" && o.cost == 1.99);
assert(o.sold.qty == 5 && o.sold.who == "Jim");

var s="";                  // for-in loop demo for objects
for (var property in o) {  //  there's wide variation
    s+=property+" ";       // in what an object exposes
}
assert(s == "name cost sold ");
```

nothingness null "undefined"

undeclared uninitialized **"unknown"** *unsafe* **void**

JavaScript has more kinds of nothing than any other language.

"Undefined" means either uninitialized or undeclared.

```
var initvar=0; // declared and initialized
var uninit; // not initialized
// var undecl; not declared

// You can compare, assign or pass an uninitialized
variable but not an undeclared variable.
var u=uninit;
assert(u == uninit);
assert(u == undefined);  // Ok everywhere but IE Mac

// typeof is safe on either, as is checking if the var is a
member of the global window object, which all globals are.
assert(typeof(initvar) != "undefined");
assert(typeof(uninit) == "undefined");
assert(typeof(undecl) == "undefined");
assert(window.initvar != null); assert(window.initvar == 0);
assert(window.uninit == null);
assert(window.undecl == null);
```

You'll get an error trying to use an uninitialized or undeclared variable or property as a function, object or array.

```
// uninit()   undecl()   window.uninit()   window.undecl()
// uninit.p   undecl.p   window.uninit.p   window.undecl.p
// uninit[i]  undecl[i]  window.uninit[i]  window.undecl[i]
```

Most JavaScript errors on live sites are one of these.

```
// Undefined and null are ==equal but not ===identical.
assert(uninit == null);  // An undefined var is equal to null,
assert(uninit !== null);  // though not identical to null.
assert(typeof(null) == "object");  // == and != are type-blind
assert(typeof(uninit) == "undefined");  // === and !== see

So the classic technique for using a nonuniversal feature
safely is to compare with null each step of the way.
if (window != null &&
    window.location != null &&
    window.location.host != null) {
    h=window.location.host;
} // This often works.  Sometimes though it's still not safe.
```

In IE5, a few objects are what typeof calls "unknown" (e.g. window.location.reload). Comparing these objects with null is an error, so typeof can be safer than comparing with null.

There are a few objects that can't even be typeof'd safely. (e.g. window.history.back in N6 or window.history.next in IE4)

The **void** *operator has one obscure purpose, keeping a hyperlink from loading a new web page:*

```
<a href="javascript:void window.open('http://html-tags.info')"
>html popup</a>
```

Similar thing in a bookmarklet:

```
javascript:window.open("http://html-tags.info"); void 0;
```

object-orientation

```
function Part(name,cost) { //constructor is the class
    this.name=name;  // define and initialize properties
    this.cost=cost;  // "this" is always explicit
};
var partBolt=new Part("bolt",1.99);  // instantiation
assert(partBolt.constructor == Part);  // type test
assert(partBolt instanceof Part);  // ancestry test
assert(Part.prototype.isPrototypeOf(partBolt)); //type test
assert(typeof(partBolt) == "object");  // not a type test
assert(partBolt.name == "bolt" && partBolt.cost == 1.99);
var partNut=new Part ("nut",0.10);
assert(partNut.name == "nut" && partNut.cost == 0.10);

Part.prototype.description=function() { // methods
    return this.name+" $"+this.toFixed(2);
}
assert(partBolt.description() == "bolt $1.99");
assert(partNut.description() == "nut $0.10");
// Whatever the prototype contains, all instances contain:
Part.prototype.toString=function() { return this.name; }
assert(partBolt.toString() == "bolt");
var a=[partBolt,partNut];  assert(a.join() == "bolt,nut");

Part.CostCompare=function(l,r) {  // class method
    return l.cost - r.cost;
}
a.sort(Part.CostCompare);  assert(a.join() == "nut,bolt");

function WoodPart(name,cost,tree) {  // inheritance
    Part.apply(this,[name,cost]);  // base constructor call
    this.tree=tree;
}
WoodPart.prototype=new Part(); // clone the prototype
WoodPart.prototype.constructor=WoodPart;
var tpick=new WoodPart("toothpick",0.01,"oak");
assert(tpick.name == "toothpick");
assert(tpick instanceof Part);  // proof of inheritance

var a=[partBolt,partNut,tpick]; // polymorphism sorta
assert(a.sort(Part.CostCompare).join() == "toothpick,nut,bolt");
assert(a[0].tree == "oak" && a[1].tree == null);
assert(a[0] instanceof WoodPart);
assert(!(a[1] instanceof WoodPart));
assert("tree" in tpick);  // membership test - in operator
assert(!("tree" in partBolt));

WoodPart.prototype.description=function() { // override
    // Calling base class version of description() method:
    var dsc=Part.prototype.description.apply(this,[]);
    return dsc+" ("+this.tree+")";  // and overriding it
}
assert(tpick.description() == "toothpick $0.01 (oak)");
assert(partBolt.description() == "bolt $1.99");
```

type typeof constructor instanceof

```
var a=[1,2,3];                  assert(typeof(a) == "object");
var A=new Array(1,2,3);         assert(typeof(A) == "object");
var b=true;                     assert(typeof(b) == "boolean");
var B=new Boolean(true);        assert(typeof(B) == "object");
var d=new Date();               assert(typeof(d) == "object");
var e=new Error("msg");         assert(typeof(e) == "object");
function f1() { };              assert(typeof(f1) == "function");
var f2=function() { };          assert(typeof(f2) == "function");
var f3=new Function(";");       assert(typeof(f3) == "function");
var n=3;                        assert(typeof(n) == "number");
var N=new Number(3);            assert(typeof(N) == "object");
var o=new Object();             assert(typeof(o) == "object");
var s="stg";                    assert(typeof(s) == "string");
var S=new String("stg");        assert(typeof(S) == "object");
var u; /* not assigned */      assert(typeof(u) == "undefined");
/* x not declared */            assert(typeof(x) == "undefined");

assert( a.constructor == Array      && a instanceof Array);
assert( A.constructor == Array      && a instanceof Array);
assert( b.constructor == Boolean);
assert( B.constructor == Boolean);
assert( d.constructor == Date       && d instanceof Date);
assert(e.constructor == Error       && e instanceof Error);
assert(f1.constructor == Function   && f1 instanceof Function);
assert(f2.constructor == Function   && f2 instanceof Function);
assert(f3.constructor == Function   && f3 instanceof Function);
assert( n.constructor == Number);
assert(N.constructor == Number);
assert( o.constructor == Object     && o instanceof Object);
assert( s.constructor == String);
assert(S.constructor == String);
```

```
<!-- Example HTML Form -->
<form id="idform" name="idform" action="http://...">
<input type="text" id="idtext" name="idtext" value="Text"
  size="3" maxlength="10" />
<input type="checkbox" id="idcheckbox" name="idcheckbox" />
<input type="radio" id="idradio" name="idradio" value="A" />
<input type="radio" id="idradio" name="idradio" value="B" />
<input type="button" id="idbutton" name="idbutton"
  value="Button" />
<select id="idselect" name="idselect" size="2" />
  <option name="idoptionA" value="A" />aoption</option>
  <option name="idoptionB" value="B" />boption</option>
</select>
<textarea id="idtextarea" name="idtextarea "
  cols="10" rows="2">Text Area</textarea></form>
```

Text | Button | aoption / boption | Text Area

form object

```
function element(id) { // universal way to get element by id
    if (document.getElementById != null) {  // 1st choice
        return document.getElementById(id); // no N4,IE4
    }
    if (document.all != null) {             // 2nd choice
        return document.all[id];            // IE only
    }
    if (document.layers != null) {          // 3rd choice
        return document.layers[id];         // N4 only
    }
    return null;                            // give up
}
var elemForm=document.idform;
assert(elemForm.name == "idform");

var elemText=elemForm.idtext;  // Three ways to get
var elemText2=elemForm["idtext"];  // the field elements
var elemText3=elemForm.elements["idtext"];
var elemCheckbox=elemForm.idcheckbox; // getting
var elemRadioA=elemForm.idradio[0];  -N6 // all of the
var elemRadioB=elemForm.idradio[1];  -N6 // form's
var elemButton=elemForm.idbutton;          // inputable
var elemSelect=elemForm.idselect;          // elements
var elemOptionA=elemSelect.options[0];
var elemOptionB=elemSelect.options[1];
var elemTextarea=elemForm.idtextarea;

assert(elemText.name == "idtext");                    Text
assert(elemText.defaultValue == "Text");
assert(elemText.value == "Text");  // what the user types
assert(elemText.size == 3);  // visible width      -N4
elemText.maxLength=12;  // same as HTML maxlength=12

assert(elemCheckbox.name == "idcheckbox");
assert(elemCheckbox.defaultChecked == false); //original
assert(elemCheckbox.checked == false);          // typed

assert(elemRadioA.name == "idradio");
assert(elemRadioA.value == "A");
assert(elemRadioA.checked == false);      -N6
assert(elemRadioA.defaultChecked == false);

assert(elemButton.name == "idbutton");          Button
assert(elemButton.value == "Button");  // value is caption

assert(elemSelect.name == "idselect");        aoption
assert(elemSelect.multiple == false);         boption
assert(elemSelect.selectedIndex == -1);  // user's choice
    // if "multiple" is false:  0 to N-1, or -1=none
assert(elemOptionA.name == "idoptionA");
assert(elemOptionA.defaultSelected == false); // original
assert(elemOptionA.selected == false);  // user choice(s)
assert(elemOptionA.value == "A");

assert(elemTextarea.name == "idtextarea");   Text Area
assert(elemTextarea.value == "Text Area");
assert(elemTextarea.defaultValue == "Text Area");

assert(elemText.form == elemForm);  // All field elements
    // have a form property, pointing to their form object.
assert(elemCheckbox.type == "checkbox");  // Most HTML
// attributes have JavaScript properties with the same name.
// So see the HTML Card or Foldout for additional properties.
```

window...
the browser window

A page with frames forms a hierarchy of window objects.

Access a window object via:
window (the current frame)
window.frames[] (inner)
window.parent (outer)
window.top (outermost)

All window members are global and vice versa: any global vars or functions become members of the "current" window.

.alert("message") *popup*
comment(), prompt()
.back()
.blur() *keyboard focus off*
.captureEvents(*mask*) *(+Op)*
.clearInterval(*setIntervalID*)
.clearTimeout(*setTimeoutID*)
.clientInformation...
.appCodeName
.appMinorVersion
.appName
.appVersion
.browserLanguage
.cookieEnabled
.cpuClass
.onLine
.platform
.systemLanguage
.userAgent
.userLanguage
.close()
.closed
.confirm("question") *yes/no*
.crypto
.defaultStatus
.document
.event
.external
.focus()
.forward()
.frameRate *(-N6)*
.frames[] *sub-windows*
.handleEvent(event) *(-N6)*
.history[] *URLs in history*
.history...
.back()
.forward()
.go(±npages)
IE4,5
.current
.next *forward URL*
.previous *back URL*
N6
.home()
.innerHeight
.innerWidth
.java
.length =*frames.length*
.location... *r:href, w:reload()*
.hash *(what's after the #)*
.host *e.g. www.visibone.com*
.hostname
.href *complete URL*
.pathname
.port *'81' in 'imdb.com:81'*
.protocol *e.g. "http:"*
.search *(after the ?)*
.reload(*bypasscache*)
.replace(*url*) *load a page*
.locationbar.visible
.menubar.visible
.moveBy(*dx,dy*)
.moveTo(*x,y*)
.name
.navigator...
.appCodeName *(Mozilla)*
.appName *(e.g. Netscape)*
.appVersion
.cookieEnabled
.javaEnabled()
.plugins[]
.mimeTypes[]
.language *(e.g. en, en-US)*
.platform *(e.g. Win32)*
.userAgent
.userLanguage *(e.g. en-us)*
.navigate(*url*) *to a new page*
.netscape
.offscreenBuffering *(-N6)*
.onblur *losing user focus*
.ondragdrop *(N4 only)*
.onerror *JavaScript error*
true skips browser popup
.onfocus *user click or tab*
.onhelp *F1=help key*

window.open(u,n,o,h)

```
var winfat=window.open(
  "http://html-tags.info");
  (with no options specified,
  almost all default to yes)

var winlean=window.open(
  "http://html-tags.info",
  "htmltags","width=400");
  (just specify 1 option, and
  most others default to no)

var wcustom=window.open(
  "http://html-tags.info",
  "htmltags",
  "alwaysLowered=no,"+
  "alwaysRaised=no,"+
  "channelmode=no,"+
  "dependent=yes,"+
  "directories=no,"+
  "fullscreen=no,"+
  "height=<npixels>,"+
  "hotkeys=yes,"+
  "innerHeight=<npixels>,"+
  "innerWidth=<npixels>,"+
  "left=<npixels>,"+
  "location=no,"+
  "menubar=no,"+
  "outerHeight=<npixels>,"+
  "resizable=no,"+
  "screenX=<npixels>,"+
  "screenY=<npixels>,"+
  "scrollbars=no,"+
  "status=no,"+
  "titlebar=yes,"+
  "toolbar=no,"+
  "top=<npixels>,"+
  "width=<npixels>,"+
  "z-lock=no");
```

.onload *<body onload>*
.onresize *desktop window*
.onscroll *browser scrollbar*
.onunload *new page coming*
.open(*url,name,options,ovrfl*)
url,name act just like href and target in a hyperlink
call window.open() so as not to confuse with document.open() in events.
.opener *(window that opened)*
.outerHeight
.outerWidth
.Packages[]
.pageXOffset
.pageYOffset
.parent *containing frame*
.personalbar.visible
.print()
.prompt("question","default")
.releaseEvents(*mask*) *(+Op)*
.routeEvent(*event*)
.resizeBy(*dwidth,dheight*)
.resizeTo(*width,height*)
.screen...
.availHeight
.availLeft
.availTop
.availWidth
.bufferDepth
.colorDepth *24=true color*
.height
.pixelDepth
.updateInterval
.width
.screenLeft
.screenTop
.screenX
.screenY
.scroll(*x,y*) *scrollTo()*
.scrollBy(*dx,dy*)
.scrollbars.visible
.scrollTo(*x,y*)
.screenLeft
.secure *(-N6)*
.self *(the window itself)*
.setInterval(*function,ms*)
.setTimeout(*function,ms*)
.setInterval("*expression*",*ms*)
.setTimeout("*expression*",*ms*)
.status
.statusbar.visible
.sun
.toolbar.visible
.stop()
.top *outermost frame*

document...
web page

Access via:
document *<html>*
(any frame).document

(Node members belong here) (form and image names too)

.activeElement
.alinkColor *<body alink>*
.all[] *all elements*
.anchors[] *<a name>*
.applets[] *Java <applet> properties are public fields*
.bgColor *<body bgcolor>*
.body... *<body>*
(see Element, Node also)
.aLink *(color)*
.background *(image)*
.bgColor
.bgProperties
.bottomMargin
.leftMargin
.link *(color)*
.noWrap
.onafterprint
.onbeforeprint
.onbeforeunload
.onload
.onselect
.onunload
.rightMargin
.scroll *(bars, yes or no)*
.text *(foreground color)*
.topMargin
.vLink *(color)*
.captureEvents(*mask*) *(+Op)*
.charset
.clear() *open() and close()*
.close() *after .write()'s*
.cookie *permanent store*
r: nam=val; nam=val; ...
w: nam=val;expires=date
.createElement(*tag*) *node*
.createStyleSheet(*url,index*) *IE5*
.designMode
.domain *part of .URL IE5*
.doctype *<!DOCTYPE>*
.documentElement *<html>*
.elementFromPoint(*x,y*)
.embeds[] *<embed>*
.expando *catch name errors*
.execCommand(*command*)
.fgColor *<body text>*
.fileCreatedDate
.fileModifiedDate
.fileSize
.fileUpdatedDate *NT*
.forms[] *<form>*
.frames[] *window.frames*
.getElementById(*id*)
.getElementsByName(*name*)
.getElementsByTagName(*tag*)
.getSelection() *user-hilited*
.handleEvent(*event*) *(-N6)*
.height
.ids[] *(-N6)*
.images[] *<img>*
.implementation
.hasFeature(*feature,ver*)
.lastModified *date & time*
.layers[] *(-N6)*
.linkColor *<body link>*
.links[] *<a href>*
.location *window.location*
.media
.mimeType
.namespaces
.namespaceURI
.onactivate
.onafterupdate
.onbeforeactivate
.onbeforedeactivate
.onbeforeeditfocus
.onbeforeupdate
.onblur *(N6)*
.oncellchange
.onclick
.oncontextmenu
.oncontrolselect
.ondataavailable
.ondatasetchanged
.ondatasetcomplete
.ondblclick *(-N6)*
.ondeactivate
.ondragstart
.onerrorupdate
.onfocus *(N6)*
.onfocusin
.onfocusout
.onhelp
.onkeydown
.onkeypress
.onkeyup
.onmousedown
.onmousemove *(N6)*
.onmouseout
.onmouseover
.onmouseup
.onmousewheel
.onpropertychange
.onreadystatechange
.onrowenter
.onrowexit
.onrowsdelete
.onrowsinserted
.onselectionchange
.onselectstart
.onunload
.open() *prepare for .write()'s*
.parentWindow *outer frame*
.plugins *.embeds[]*
.previousSibling
.protocol
.readyState
.referer *inbound link*
.releaseEvents(*mask*) *(+Op)*
.routeEvent(*event*)
.scripts[] *<script>*
.text *JavaScript source*
.security
.selection
.styleSheets[] *<style>*
.cssRules
.item(*index*).cssText
.cssText
.deleteRule(*index*)
.disabled
.href
.id
.imports
.insertRule(*cssrule,index*)
.media
.owningElement
.ownerNode
.ownerRule
.pages
.parentStyleSheet
.readOnly
.title
.type
.title *<title>*
.uniqueID
.URL *same as in address bar*
.URLUnencoded
.vlinkColor *<body vlink>*
.width
.write(*raw_html_string*)
.writeln(*raw_html_string*)

⌘ window.open *(u,n,o,h) pops up a new browser window with a web page in it.*

document.open()
document.write(string)
document.close()
rewrite from scratch new HTML content for an existing browser window.

style *style –N4*

Cascading Style Sheets with the names reworded slightly. Value syntax mostly the same.

❖ *CSS page of HTML Card*

Netscape 4 supports merely what's in document.layers[]

document.styleSheets[]

dimensions need units, e.g. elem.style.top="200 px";

defaults are not exposed, members are empty if not set.

.accelerator
.background
.backgroundAttachment
.backgroundColor
.backgroundImage
.backgroundPositionX
.backgroundPositionY
.backgroundPosition
.backgroundRepeat
.behavior
.border
.borderBottom
.borderBottomColor
.borderBottomStyle
.borderBottomWidth
.borderCollapse
.borderColor
.borderLeft
.borderLeftColor
.borderLeftStyle
.borderLeftWidth
.borderRight
.borderRightColor
.borderRightStyle
.borderRightWidth
.borderStyle
.borderTop
.borderTopColor
.borderTopStyle
.borderTopWidth
.borderWidth
.bottom
.clear
.clip
.color
.cssText
.cursor
.direction
.display *"none"=vanishes*
.filter
.font
.fontFamily
.fontSize
.fontStyle
.fontVariant
.fontWeight
.height
.imeMode
.layoutFlow *(IE5)*
.layoutGrid
.layoutGridChar
.layoutGridLine
.layoutGridMode
.layoutGridType
.left
.letterSpacing
.lineBreak
.lineHeight
.listStyle
.listStyleImage
.listStylePosition
.listStyleType
.margin
.marginBottom
.marginLeft
.marginRight
.marginTop
.minHeight *(IE5.5)*
.overflow
.overflowX
.overflowY
.padding
.paddingBottom
.paddingLeft
.paddingRight
.paddingTop
.pageBreakAfter
.pageBreakBefore
.posBottom
.posHeight
.position *animation*
.posLeft
.posRight
.posTop
.posWidth
.right
.rubyAlign
.rubyOverhang
.rubyPosition
.scrollbar3dLightColor
.scrollbarArrowColor
.scrollbarBaseColor
.scrollbarDarkShadowColor
.scrollbarFaceColor
.scrollbarHighlightColor
.scrollbarShadowColor
.scrollbarTrackColor
.styleFloat
.tableLayout
.textAlign
.textAlignLast *(IE5.5)*
.textAutospace
.textDecoration
.textDecorationBlink
.textDecorationLineThrough
.textDecorationNone
.textDecorationOverline
.textDecorationUnderline
.textIndent
.textJustify
.textJustifyTrim
.textOverflow
.textTransform
.textUnderlinePosition *(IE5.5)*
.top
.unicodeBidi
.verticalAlign
.visibility *transparent*
.whiteSpace
.width
.wordBreak
.wordSpacing
.wordWrap *(IE5.5)*
.writingMode *(IE5.5)*
.zIndex
.zoom

Element *All the visible parts of an HTML page, including document.body*

(in add'n to Node members)

.all[]
.behaviorUrns
.canHaveChildren
.canHaveHTML *(IE5.5)*
.children
.className *CSS class(es)*
.clientHeight
.clientWidth
.clientLeft *.offsetLeft*
.clientTop *.offsetTop*
.contentEditable *(IE5.5)*
.currentStyle
.dir
.disabled *greyed-out (+Op)*
Input .disabled
.document
.filters
.getAttributeNode(*name*)
.getAttribute(*name*)
.getElementsByTagName(*n*)
.id *XHTML says to use in tandem with .name*
.innerHTML *what's between start & end tags N6 body*
.innerText *subtags stripped*
.isContentEditable *(IE5.5)*
.isDisabled *(IE5.5)*
.isMultiLine *(IE5.5)*
.isTextEdit
.lang
.language
.name *.id*
.offsetWidth
.offsetHeight
.offsetTop *relative to*
.offsetLeft *parent element*
.offsetParent
.onactivate *(IE5.5)*
.onafterupdate
.onbeforeactivate *(IE5.5)*
.onbeforecopy
.onbeforecut
.onbeforedeactivate *(IE5.5)*
.onbeforeeditfocus
.onbeforepaste
.onbeforeupdate
.oncellchange
.onclick
.oncontextmenu
.oncontrolselect *(IE5.5)*
.oncopy
.oncut
.ondataavailable
.ondatasetchanged
.ondatasetcomplete
.ondblclick *(+N4)*
.ondeactivate *(IE5.5)*
.ondrag
.ondragend
.ondragenter
.ondragleave
.ondragover
.ondragstart
.ondrop
.onerrorupdate
.onfilterchange
.onlayoutcomplete *(IE5.5)*
.onlosecapture
.onmousedown
.onmouseenter *(IE5.5)*
.onmouseleave *(IE5.5)*
.onmousemove *(-N4)*
.onmouseout *roll-overs*
.onmouseover
.onmouseup
.onmousewheel *(IE5.5)*
.onmove *(IE5.5)*
.onmoveend *(IE5.5)*
.onmovestart *(IE5.5)*
.onpage *(IE5.5)*
.onpaste
.onpropertychange
.onreadystatechange
.onresizeend *(IE5.5)*
.onresizestart *(IE5.5)*
.onrowenter
.onrowexit
.onscroll
.onselectstart
.outerHTML
.outerText
.parentElement
.parentTextEdit
.readyState
.removeAttribute(*name*)
.removeAttributeNode(*attr*)
.runtimeStyle
.scopeName
.scrollWidth
.scrollHeight
.scrollLeft *(+Op for body)*
.scrollTop *(+Op for body)*
.setAttribute(*name,value*)
.setAttributeNode(*attr*)
.sourceIndex
.style
.tagUrn
.tagName
.title

inputable elements *<input> <select> <textarea> <a href>*

Things you can click, tab to, or type on.

Access via:
document.idform.idelement
document.forms[].elements[]
document.links[]
document.getElementById()
document.all[]

Loose association: members belong to at least 1 inputable. (in add'n to Element members)

.accessKey *shortcut*
.align *(-N4)*
.blur() *focus elsewhere*
.checked *(checkbox, radio)*
.click() *simulated mouse*
.cols *(textarea) (-N4)*
.dataFld
.dataFormatAs
.dataSrc
.defaultChecked *read-only,*
.defaultSelected *used at*
.defaultValue *reset time*
.disabled *grayed-out*
.focus()
.form
.length *(of select's options)*
.maxLength *(typable text)*
.multiple *(select) (-N4)*
.onblur
.onchange
.onclick
.ondblclick *(+N4)*
.onfocus
.onfocusin
.onfocusout
.onhelp
.onkeydown
.onkeypress
.onkeyup
.onselect *(-N4)*
.options[] *(select)*
.select() *swipe visible text*
.selected *<option>*
.selectedIndex *(select, 0...)*
.size *(visible text width)*
.readOnly
.recordNumber
.remove()
.rows *(–N4) (textarea)*
.tabIndex
.type
.value
.wrap *(textarea)*

Node *In IE5 & N6, an HTML page is a hierarchical tree of nodes, with the document object at the root. document.createElement or document.createTextNode can make new nodes.*

.nodeType *9=doc, 1=element, 2=attribute, 3=text, etc.*
.nodeName *id or name*
.nodeValue
.childNodes[]
.firstChild
.lastChild
.nextSibling
.previousSibling
.parentNode *Op*
.ownerDocument *root (-IE5)*
.attributes *element subnodes*
.cloneNode(*descendentstoo*)
.hasChildNodes()
.insertBefore(*nnew,nref*)
.appendChild(*nnew*)
.removeChild(*nnref*)

Image *<img>*

Access: document.imgname

(in addn to Element members)

.align
.alt
.border
.complete
.dynsrc
.fileCreatedDate
.fileModifiedDate
.fileSize
.fileUpdatedDate
.height
.hspace
.isMap
.longDesc
.loop
.lowsrc
.mimeType
.onabort()
.onerror()
.onload()
.protocol
.src *prefetch for rollovers:* (new Image).src='x.gif'; *then change on the fly*
.start
.useMap
.vrml
.vspace
.width
.x *Element .offsetLeft*
.y *Element .offsetTop*

form element

<form> Access via document.forms[index] or document.formname

(Element and Node members are form members also)

.action
.elements[] *inputables*
.encoding *enctype*
.method
.onreset
.onsubmit
.target

anchor element

<a name> Access via document.anchors[index]

(Element and Node members are anchor members also)

.text *betw. start & end tags*
.x *location on page (N4 only)*
.y

Event *object*
details of the occurrence

.altKey
.altLeft *(IE5.5)*
.behaviorCookie *(IE5.5)*
.behaviorPart *(IE5.5)*
.bookmarks
.boundElements
.bubbles *true=propagates*
.button *mouse button generally: 0,1=left 2=right*
.cancelBubble
.cancelable *w/preventDefault*
.charCode
.clientX *click point within*
.clientY *browser window*
.ctrlKey
.contentOverflow *(IE5.5)*
.ctrlLeft *(IE5.5)*
.currentTarget *handler (+Op)*
.data *dropped URLs (N4)*
.dataFld
.dataTransfer
.detail *e.g. click count*
.eventPhase
.fromElement
.getPreventDefault()
.height *(N4 only)*
.initEvent(*type,c,c*) *(+Op)*
.initKeyEvent(*type,c,c,...*)
.initMouseEvent(*type,c,c,...*)
.initUIEvent(*type,c,c,...*)
.isChar
.keyCode
.layerX *local click point*
.layerY *or dimensions*
.metaKey
.modifiers *(N4 only)*
.nextPage *(IE5.5)*
.offsetX
.offsetY
.originalTarget
.pageX *click point within*
.pageY *browser window*
.preventBubble()
.preventCapture()
.preventDefault() *(+Op)*
.propertyName
.qualifier
.rangeOffset
.rangeParent
.reason
.recordset
.relatedTarget *(+Op)*
.repeat
.returnValue
.screenX *screen click point*
.screenY
.shiftKey
.shiftLeft *(IE5.5)*
.srcElement
.srcFilter
.srcUrn
.stopPropagation() *(+Op)*
.target *generating element*
.timeStamp
.toElement
.type *e.g. "click" for onclick*
.view *(window)*
.wheelData *(IE5.5)*
.which *mouse button generally: 1=left 3=right*
.width *(N4 only)*
.x *local click point*
.y

Besides all the diversity within an Event object, there are 3 ways just to get the thing:
1. *"event" pseudo-parameter in the attribute code (N)*
2. *window.event (IE, Op)*
3. *parameter to an assigned handler function (N, Op)*

The following HTML example neatly mixes ways 1 and 2:

```
<form id="frm" name="frm">
  <input id="btn" name="btn"
    type="button"
    onmousedown=
    "return mdown(event);" />
</form>
<script>
function mdown(e) {
  if (e.button == 2 ||
      e.which == 3) {
    alert("right-click");
  }
  else {
    alert("left-click");
  }
  return false;
  // false=browser ignore
} // true=browser handle it
</script>
```

This script uses way 3 or 2:

```
function mdown2(e) {
  if (e != null) {
    return mdown(e);
  }
  if (window.event != null) {
    return mdown(
      window.event);
  }
}
document.frm.btn
  .onmousedown=mdown2;
```

Layer *Netscape 4 Omni/Web*

<layer> and some others: <a> if style="position:relative" or "position:absolute"

Access by name or number: document.layers.name document.layers[index]

.above
.background *(e.g. "bg.gif")*
.below
.bgColor *(e.g. "#FFFFFF")*
.captureEvents(*mask*)
.handleEvent(*event*)
.clip...
.height .left .top .right
.width .bottom
.document *(HTML within) .open() .write() .close() to rewrite <layer> contents.*
.hidden *(true or false)*
.layers[] *contained layers*
.left
.load(*url,width*)
.moveAbove(*layerobj*)
.moveBelow(*layerobj*)
.moveBy(*dx,dy*)
.moveTo(*x,y*) *(within layer)*
.moveToAbsolute(*x,y*) *(page)*
.name
.offset(*dx,dy*)
.pageX
.pageY
.parentLayer
.releaseEvents(*mask*)
.resizeBy(*dw,dh*) *changes*
.resizeTo(*w,h*) *clipping*
.routeEvent(*event*)
.siblingAbove *other layers*
.siblingBelow *in z order*
.src
.top
.visibility
.window *(containing window)*
.x
.y
.zIndex *z-index (high=top)*

Regular Expressions for JavaScript

```
assert(/hello/.test('hello world'));
```

/ regular expression / — *Regular expressions are **patterns** used to match portions of strings*

```
// The test method says whether there's a match anywhere.
assert(!/i/.test("courage"));  // there is no "i" in courage
assert(/our/.test("courage"));  // there is "our" in courage

// The search method says how many characters precede.
assert("courage".search(/our/) == 1); // 1 letter before "our"
assert("courage".search(/i/)  == -1); // -1 means no match

// The replace method changes a matched substring
assert("recieve".replace(/ie/,"ei") == "receive");
```

*Regular expressions can help **validate** a string's syntax, **search** for and **extract** patterns, or **modify** strings.*

Character Classes *(sets of matchables)*

\d digit / **\D non-digit**

```
// /\d/ matches any decimal digit
assert(/\d/.test('9'));
// same as /[0123456789]/ or /[0-9]/
assert(/[0123456789]/.test('9'));
assert(/[0-9]/.test('9'));
```

\s space / **\S non-space**

```
// /\s/ matches space, tab, terminators
assert(/\S\S\s\S\S/.test("to be"));
// any invisible "white space" character
assert(/\s/.test(" "));
assert(/[ \t\n\u000B\f\r]/.test(" "));
```

\w word char / **\W non-word**

```
// /\w/ is a letter, number, underscore
assert(/\w\w\W\w\w\w\W\w\w/
       .test("21-May/02"));
assert(/\w/.test('X'));
assert(/[0-9A-Za-z_]/.test('X'));
```

[ ] one of

```
var vowel=/[aeiouy]/;
assert('story'.search(vowel) == 2);
assert(!vowel.test('mfgr'));
```

[^] one not of

```
var nonvowel=/[^aeiouy]/;
assert(nonvowel.test('our'));
assert(!nonvowel.test('eye'));
```

- range

```
assert(/[a-z]/.test('Story'));
assert('$8ea'.replace(/[0-9]/,'X') == '$Xea');
assert(/[^a-zA-Z]/.test('Yes?'));
```

Repeaters *(find a repetitive pattern)*

? optional (0 or 1)

```
assert(/to?t/.test("tt"));
assert(/to?t/.test("tot"));
assert(!/to?t/.test("toot"));
assert(!/to?t/.test("tooot"));
```

*** any (0 or more)**

```
assert(/to*t/.test("tt"));
assert(/to*t/.test("tot"));
assert(/to*t/.test("toot"));
assert(/to*t/.test("tooot"));
```

+ etc. (1 or more)

```
assert(!/to+t/.test("tt"));
assert(/to+t/.test("tot"));
assert(/to+t/.test("toot"));
assert(/to+t/.test("tooot"));
```

{n} exactly (n)

```
assert(!/to{2}t/.test("tt"));
assert(!/to{2}t/.test("tot"));
assert(/to{2}t/.test("toot"));
assert(!/to{2}t/.test("tooot"));
```

{n,} min (n or more)

```
assert(!/to{2,}t/.test("tt"));
assert(!/to{2,}t/.test("tot"));
assert(/to{2,}t/.test("toot"));
assert(/to{2,}t/.test("tooot"));
```

{n,m} range (n to m)

```
assert(!/to{1,2}t/.test("tt"));
assert(/to{1,2}t/.test("tot"));
assert(/to{1,2}t/.test("toot"));
assert(!/to{1,2}t/.test("tooot"));
```

character matching *one-for-one:*

A…Z a…z 0…9 alphanumeric / **\ punc-tuation**

\u0000…\uFFFF Unicode hexadecimal

\x00…\xFF ASCII hexadecimal

\cA…\cZ control characters

\0…\7 \00…\77 \000…\377 ASCII Octal *N6

\0	[\b]	\t	\v	\f	\r	\n
NUL \x00	back space \x08	TAB \x09	VT \x0B	FF \x0C	CR \x0D	new line \x0A

\d digit / **\D non-digit** / **\s space** / **\S non-space** / **\w word char** / **\W non-word**

```
// Use digit, word or space in or outside class [ brackets ]
assert(/^[\d\s\(\)\-\+\/]*$/.test('(+20) 2 0900 0700'));
assert(/^[\d\s\(\)\-\+\/]*$/.test('714/921-5424'));
assert(!/^[\d\s\(\)\-\+\/]*$/.test('1-866-4MORTAL'));
```

```
// Most punctuation symbols have some special meaning in
// regular expressions. To search for those characters in
// your target, you should precede them with a slash.
// Here are those characters all slashified:
assert(/\(\)\*\+\.\/\:\=\?\[\\\]\^\{\|\}/
   .test("!$()*+./:=?[\\]^{|}"));
// The following characters don't (yet) need to be slashified:
assert(/#%&,-;<>@_~/
   .test("#%&,-;<>@_~"));
// But they work fine if they are:
assert(/\#\%\&\,\-\;\<\>\@\_\~/
   .test("#%&,-;<>@_~"));
// Similarly, quotes don't need to be slashified but can be:
assert(/'/.test(' '));    assert(/\"/.test('"'));
assert(/'/.test("'"));    assert(/\"/.test("'"));
```


Anchors *(matches between characters)*

^ string starts with — *use both ^ and $ when you want to cover the entire string*

```
assert(/^a/.test('apple'));
assert(/^a/.test('alpha'));
assert(!/^a/.test('baker'));
```

$ string ends with

```
assert(/e$/.test('apple'));
assert(!/e$/.test('baker'));
assert(/e$/.test('charlie'));
```

\b word boundary / **\B not a word boundary, in or outside a word**

```
// "words" are consecutive letters, numbers or underscores
assert(/I am\?/.test("I am?")); // same as what matches \w
assert(/\bI\b \ba\Bm\b\?\B/.test("I am?"));
```

a subexpression *groups or captures contents*

( ) // groups *characters for a repeater:*

```
assert(/friend(ship)?/.test('friend'));
assert(/friend(ship)?/.test('friendship'));
```

*// or **captures** submatched characters for later recall:*

```
assert('abc'.replace(/(a)(b)(c)/,'$1.$2.$3') == 'a.b.c');
assert('in 206BC is'.replace(/(\d+)BC/,'<i>$1</i>BC')
   == 'in <i>206</i>BC is');
// only plain ( ) parens capture, not (?: ) nor (?= ) nor (?! )
```

\1 \2 … submatch recur — *insist that what matched a subexpression appear again*

```
var s='the the first man to to see';
var r=/(\w+) \1/g;
assert(s.replace(r,"$1 [$1]") ==
       'the [the] first man to [to] see');
```

(?=) look-ahead equals — *specify what comes next (but leave it alone)* -N4 -IE5 -Op

```
assert("abc".replace(/ab(c)/,'-') == "-");
assert("abc".replace(/ab(?=c)/,'-') == "-c");
```

(?!) look-ahead not-equals — *specify what's not next*

```
assert("abc".replace(/ab[^d]/,'-') == "-");
assert("abc".replace(/ab(?!d)/,'-') == "-c");
```

(?:) a passive subexpression — *only groups its contents* -N4 -IE5 -Op

```
assert("abc".replace(/(a)(b)(c)/,'$1..$2') == "a.b");
assert("abc".replace(/(a)(?:b)(c)/,'$1..$2') == "a.c");
```

Flags: *global repeat, insensitive to case, multi-line*

/g

```
assert('aardvark'.replace(/a/,'o')  == 'oardvark');
assert('aardvark'.replace(/a/g,'o') == 'oordvork');
```

/i

```
assert('Aardvark'.replace(/a/,'e')  == 'Aerdvark');
assert('Aardvark'.replace(/a/i,'e') == 'eardvark');
assert('Aardvark'.replace(/a/gi,'e') == 'eerdverk');
```

/m -N4 -IE5 — *makes ^ and $ match **line** start and end*

```
assert('ten\nton\ntin'.replace(/n$/,'x')
   == 'ten\nton\ntix');
assert('ten\nton\ntin'.replace(/n$/m,'x')
   == 'tex\nton\ntin');
assert('ten\nton\ntin'.replace(/n$/gm,'x')
   == 'tex\ntox\ntix');
```

*Regular expressions can help **validate** a string's syntax, **search** for and **extract** patterns, or **modify** strings.*

characters inside class [brackets]

A…Z a…z 0…9 alphanumeric / **\ punc-tuation**

\x0000…\xFFFF Unicode hexadecimal

\x00…\xFF ASCII hexadecimal

\cA…\cZ control characters

\0…\7 \00…\77 \000…\377 ASCII Octal

```
// Six ways to specify a newline in a regular expression:
assert("abc\n".search(/\n/) == 3);
assert("abc\n".search(/\cJ/) == 3);  // Not in early IE4
assert("abc\n".search(/\x0A/) == 3);
assert("abc\n".search(/\u000A/) == 3);
assert("abc\n".search(/\12/) == 3);  // Not in Netscape 6
assert("abc\n".search(/\012/) == 3); // nor Opera
```

Think of regular expressions like Lego® building blocks. They make sense once you envision how all the pieces fit together.

simple methods

b=r.test(s); — *is there a match anywhere?*

```
assert(/a/.test("apple"));
assert(/e/.test("apple"));
assert(!/x/.test("apple"));
```

n=s.search(r); — *where is the match?* ❖String

```
assert("apple".search(/a/) == 0);
assert("apple".search(/pp/) == 1);
assert("apple".search(/l/) == 3);
assert("apple".search(/e/) == 4);
assert("apple".search(/x/) == -1);  // -1=not found
```

s=s.replace(r,s); — *replace pattern with string*

```
assert('alan'.replace(/a/,"A") == "Alan");
assert('alan'.replace(/a/,"e") == "elan");
assert('alan'.replace(/a/g,"e") == "elen");
```

replace()'s special symbols:

$` $& $' $_ $1 $2 … $+ $$

```
// symbols for any regular expression
assert('abcd'.replace(/bc/,'$`') == "aad");  // leftContext
assert('abcd'.replace(/bc/,'$&') == "abcd");  // lastMatch
assert('abcd'.replace(/bc/,"$'") == "add");  // rightContext
assert('abcd'.replace(/bc/,'$_') == "aabcdd");  // input
assert('abcd'.replace(/bc/,'$$') == "a$d");  // (literal $)
// symbols recalling subexpression-matched characters:
assert('abcd'.replace(/(b)(c)/,'$1') == "abd");  // $1
assert('abcd'.replace(/(b)(c)/,'$2') == "acd");  // $2
assert('abcd'.replace(/(b)(c)/,'$+') == "acd"); // lastParen
```

advanced methods *(each outputs an array)*

a=s.split(rd); ❖Array — *Delimiter breakdown*

```
var s="to be the *first* -- that is the idea"; // will parse by
var a=s.split(/\W+/); // any series of non-word characters
assert(a.join() == "to,be,the,first,that,is,the,idea");
var a=s.split(/\W+/,4); // split can also limit array length
assert(a.join() == "to,be,the,first");
var a="to be first".split(/(\W+)/);  // Netscape split() will
assert(a.join() != "to, ,be, ,first");  // include submatches
```

a=s.match(rg); ★ *Global breakdown*

```
var sBingo="Calling b7 i20 n33, anybody win yet?";
// extract bingo codes: one bingo-letter plus 1 or 2 digits
var ag=sBingo.match(/[bingo]\d{1,2}/g);
assert(ag.length == 3);
assert(ag[0] == 'b7');  // a match on a g-option expression
assert(ag[1] == 'i20');  // returns an array with each match
assert(ag[2] == 'n33');
assert(ag.input == sBingo);
```

a=s.match(rp); *Parenthetical breakdown*

```
// (subexpression 1: bingo-letter) (subexpression 2: digits)
var ap=ag[2].match(/([bingo])(\d{1,2})/);
assert(ap.length == 3);  // non-g match array very different:
assert(ap[0] == 'n33');  // 0: entire matched string
assert(ap[1] == 'n');    // 1: 1st subexpression's match
assert(ap[2] == '33');   // 2: 2nd subexpression's match
assert(ap.index == 0);
assert(ap.input == ag[2]);
```

a=r.exec(s); *Parenthetical & global breakdown*

```
// Without the g option, r.exec(s) is identical to s.match(rp).
// With the g option, something very unusual happens.
// Each call to exec() matches the entire expression once
// (as if there were no g option) and breaks it down
// parenthetically (as s.match(rp)) but it picks up where the
// last call left off.  So repeatedly calling exec() until null
// gets every match the g-option would normally have hit.
RegExp.lastIndex=0;  // (IE wise to do before first exec)
var r=/([bingo])(\d{1,2})/g;
var a=r.exec(sBingo); // first exec breaks down first match
assert(a.join() == "b7,b,7");
a=r.exec(sBingo);    // next exec breaks down next match
assert(a.join() == "i20,i,20");
a=r.exec(sBingo);
assert(a.join() == "n33,n,33");
a=r.exec(sBingo);    // ...there are no more matches
assert(a == null);

// Here's exec() flexing its muscle in a 2D loop:
var s="";
RegExp.lastIndex=0;  // (less grief: always start with this)
while ((a=r.exec(sBingo)) != null) {
   for (var i=1 ; i < a.length ; i++) {  // (skipping a[0])
      s+=a[i]+' ';
   }
   s+='/ ';
}                                              -IE5
assert(s == "b 7 / i 20 / n 33 / ");
```

Constructors

```
r=new RegExp(s);
r=new RegExp(s,sFlags);   // IE4-5 do not work without the "new"
s=r.source;
r=r.compile(s,sFlags);
```

```
var r1=/e/g;                    // Four ways
var r2=new RegExp("e","g");     // to make a
var r3=eval("/e/g");            // regular
var r4=new RegExp("e","g");     // expression
assert(r1 != r3 && r2 != r4);  // (IE Compare by reference)
assert("meme".replace(r1,"i") == "mimi");  // They all
assert("meme".replace(r2,"i") == "mimi");  // perform
assert("meme".replace(r3,"i") == "mimi");  // identically.
assert("meme".replace(r4,"i") == "mimi");
var r=new RegExp("\\d\\w");  // Double backslashes, since
assert(r.test('3D'));  // strings use them for escaping too
var r=new RegExp("\\\\");  // Quadruple literal backslashes
assert(r.test("\u005C"));  // string-doubling, regexp-doubling
assert(r1.source == 'e');
assert(r2.source == 'e');
```

s=RegExp.leftContext; / **s=RegExp.lastMatch;** / **s=RegExp.rightContext;** — *side effects of the most recent regular expression (that made a match)*

s=RegExp.input; / **n=RegExp.index;**

```
/and/.test('to-and-fro');
assert(RegExp.leftContext == 'to-');
assert(RegExp.lastMatch == 'and');
assert(RegExp.rightContext == '-fro');
assert(RegExp.input == 'to-and-fro');
assert(RegExp.index == 3);
```

s=RegExp.$1; / **s=RegExp.$2;** / **⋮** / **s=RegExp.lastParen;** — *side effects of **subexpressions** of the most recent regular expression (that made a match)*

```
/^(\w+)-(\w+)-(\w+)$/.test('to-and-fro');
assert(RegExp.$1 == 'to');
assert(RegExp.$2 == 'and');
assert(RegExp.$3 == 'fro');
assert(RegExp.lastParen == RegExp.$3);
```

Most regular expressions are somewhere between dirt simple and fiendishly complex.


```
// Example regular expressions to extract email addresses.  Hmm, what could one use this for...  Spam?  Spammer blacklist?
var reLoose=/\w[\-\~]*\@[\-\~]+\w/g;  // A loose look for things remotely resembling email addresses, blah@blah
var reTight=/^[a-z0-9][^\(\)\<\>\@\,\;\:\\\"\[\]]*\@[a-z0-9][a-z0-9\-\.]*\.[a-z]{2,4}$/i;  // For validation
// Alphanum, RFC822 characters, @-sign, alphanums dashes and dots, a 2-4 letter suffix, no more no less, case insensitive
var TextBlob="From: spammer4275@remail.com\nSubject: Whiter teeth\n\n (Removal: morespam@whiter.com.)\n";
var JustEmails=     "spammer4275@remail.com,"+                    "morespam@whiter.com";

function ExtractEmails(text) {
   var maybes=text.match(reLoose);
   var definites=new Array;
   for (var i=0, j=0 ; i < maybes.length ; i++) {
      if (reTight.test(maybes[i])) {
         definites[j++]=maybes[i];
      }
   }
   return definites;
}

assert(ExtractEmails(TextBlob).join() == JustEmails);
```

The VisiBone Browser Book for Web Designers

August 2004

A client-side technology memory jogger

Table of Contents

LEGEND

Common features:

- Internet Explorer (4-6) and AOL (5) do not support
- Netscape (4-6) and Mozilla (6) do not support
- Opera (6) does not support
- Not a W3C or ECMA standard, all browsers support
- Netscape 4 does not support
- IE4 no (IE5 maybe)
- N4 and IE4 do not support (IE5 maybe)
- Opera and IE4 do not support (IE5 maybe)
- Opera and Netscape 4 do not support (N6 maybe)
- Opera, Netscape 4, IE4 do not support (IE5 maybe)
- (universal features have no coloring)

Rare features:

- Only supported by Internet Explorer (and AOL)
- Only supported by Netscape 4
- Unimplemented W3C or ECMA features (except N6 implements most

Further Complications:

(IE5.5) Internet Explorer support begins with version 5.5
(–N6) Netscape 6 and 7 do not support (nor Mozilla 1)

Other browsers support white-background features, except:

(Op) (-Op) Opera, *(Sf) (-Sf)* Safari™, *(Kq) (-Kq)* Konqueror } supports: (yes) or (-no)

(a) → (b) W3C or ECMA deprecates (a) in favor of (b) i.e. the standards say don't use (a), use (b)
(x) Microsoft suggests (x)
(x) VisiBone suggests (x)

- Not supported by IE5 on the Mac
- Not supported by Netcape on the Mac
- IE bug
- Netscape bug
- Opera bug
- IE diversity
- Netscape diversity
- Opera diversity
- Diversity among all implementations
- Astonishing diversity (freak flags fly)
- unlike the C or C++ programming languages
- unlike the Java programming language
- Block element (rectangular area, as opposed to a stream of text)
- Nestable (inside itself)
- ×2 Nestable to two levels
- Nestable in Netscape
- Default value
- Default in IE
- Default in Netscape
- See section
- Contrast with
- Identical to
- Confusing issue, attempt to unravel
- Gotcha, potential bug, something to watch out for
- Null bites (comparison is an error, but typeof is safe)
- Poisonous, no safe handling (e.g. typeof is an error)
- Security issues (signed script, or HTML application)
- Unsettlingly advanced
- *or* ★ Neat Trick
- Mysterious
- † Obsolete
- Corresponds to HTML element (e.g. <body>)
- Corresponds to HTML attribute (e.g. <body link>)
- Corresponds to HTTP header field (e.g. User-Agent)

#rrggbb (HTML or CSS)
Hexadecimal color code or color name ❖Color Card (#000000 or black, #FFFFFF or white, #FF0000 or red, ...)

color (CSS only)
#rrggbb (00-FF)
#rgb (0-F)
rgb(*rrr*,*ggg*,*bbb*) (0-255)
rgb(*rr*%,*gg*%,*bb*%) (0-100)
or black or white or red ...

| *Tags use same attributes or properties same values*
— *Alternative sets*
| *Alternative values*
/> *Element that never has an end-tag, XHTML syntax*

<tag>	XHTML
{style}	CSS - Cascading Style Sheet property
&character;	Special character
javascript	language keyword or syntax
dom	browser document object model
regular expression	string handling

Functional Index for Tags and Styles pages (4,5,6,7)

Tables <table> <thead> <tr> <tfoot> <td> <th> <tbody> <caption> <colgroup> <col> {border-collapse} {empty-cells} {table-layout}

Lists   <dl> <dt> <dd> <dir> <menu> {list-style} {counter-xxx}

Forms <form> <textarea> <input type=text> <input type=password> <input type=checkbox> <input type=radio> <input type=file> <input type=hidden> <input type=button> <input type=reset> <input type=submit> <input type=image> <select> <option> <optgroup> <button> <isindex> <label> <fieldset> <legend> <datasrc> <datafld> <dataformatas>

Hyperlinks <a href> <map> <area href> a:link{} a:visited{} a:active{} a:hover{} <base> <a name>

Image maps <map> <area> <img ismap> <img usemap=map-name>

Programming <object> <applet> <param> <embed> <noembed> <script> <noscript> <xxx language> <xxx onxxx> {behavior}

Frames <frame> <frameset> <noframes> <a href=url target=frame-name>

Style sheets <style> <xxx style> <xxx class> <xxx id> <link rel=stylesheet>

Audio <bgsound> {azimuth} {cue-xxx} {elevation} {pause} {pitch} {play-during} {richness} {speak-xxx} {speech-rate} {stress} {voice-family} {volume}

Help for blind users: <area alt> <img alt> <xxx title> <td abbr> <table summary> {speak-header} *(and all of)*

Multilingual: <bdo> {unicode-bidi} <xxx dir> {direction} <q> {quotes} xxx:before{} xxx:after{} <xxx lang> xxx:lang{} <meta http-equiv=content-language> @charset <?xml encoding?> <meta http-equiv=content-type>

Font <font face> <font size> <big> <basefont> <small> {font-size} {font-family} {font-stretch} {font-variant:small-caps} {text-transform} @font-face{font-family}

Bold <b> {font-weight}

Shadow {text-shadow}

Italics <i> <cite> <var> <xmp> <address> <dfn> {font-style:italic} {font-style:oblique}

Subscript <sub> {vertical-align:sub}

Superscript <sup> {vertical-align:super}

Strike-through <s> <strike> <del> {text-decoration: line-through}

Monospace <tt> <code> <kbd> <xmp> <plaintext> <listing> <pre> <samp> {white-space:pre} {font-family:monospace}

Overline {text-decoration: overline}

Underline <u> <ins> {text-decoration: underline}

Lines: <hr> <u> <td bgcolor> {border} {border-xxx} {outline} {outline-xxx}

Graphics: <img> <body background> <td background> {background-image} {filter} {zoom}

Interactive: <xxx tabindex> <xxx accesskey> {cursor} a:hover{} xxx:hover{} xxx:focus{} <xxx disabled> *(also Forms)*

Promotion: <title> <meta name=description> <meta name=keywords> <meta name=robots>

Chimp-attract: <blink> <marquee> {text-decoration:blink} {writing-mode:tb-rl} xxx:first-letter{} xxx:first-line{} {filter} {scrollbar-xxx-color}

Annotation: <ruby> <rt> {ruby-xxx} <caption> <legend> <thead> <tfoot> <title> <img alt> <xxx alt> <xxx title>

Color: <body bgcolor> <body link> a:link{} <body vlink> a:visited{} <body alink> a:active{} <body text> {color} <td bgcolor> <td bordercolor> <xxx bgcolor> <xxx bordercolor> <font color> <hr color> {background-color} {border-color} {scrollbar-xxx-color-xxx}

Layout: <xxx align> <xxx valign> <layer> <nolayer> {clip} {overflow} {display} {float} {position} {visibility} {height} {width} {vertical-align}

Space: <spacer> {margin} {padding}

Text Flow: <blockquote>
 <nobr> <wbr> <p> <td nowrap> <img align> <div align> <multicol> {clear} {float} {letter-spacing} {line-height} {orphans} {widows} {page-break-xxx} {text-align} {text-indent} {text-justify} {white-space:nowrap} {word-wrap:break-word}

Print: @media{} @page {page-break-xxx}

System: <?xml?> <!DOCTYPE> <html> <head> <meta> <title> <link> <body> <h1> <h2> <h3>... <!-- comment --> <comment> <xml> @import <meta http-equiv=refresh content="n;'url=url'"> <meta http-equiv=pragma content=nocache>

www.visibone.com

VisiBone Browser Book for Web Designers — Color Card

2006 Edition
www.visibone.com/color

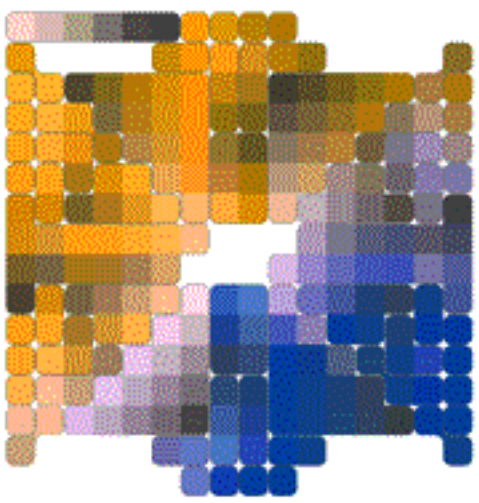
Here are the 216 colors most widely supported by web browsers.

Colors are grouped by hue.
Touching color chips have the same hue.

Swatch libraries with this color layout are available for several design tools.
www.visibone.com/swatches

FFFFFF 255,255,255 W	CCCCCC 204,204,204 PG	999999 153,153,153 LG	666666 102,102,102 DG	333333 51,51,51 OG	000000 0,0,0 K	FFCC00 255,204,0 YYO	FF9900 255,153,0 OOY	FF6600 255,102,0 OOR	FF3300 255,51,0 RRO														
99CC00 153,204,0 DYS						CC9900 204,153,0 DYO	FFCC33 255,204,51 LYO	FFCC66 255,204,102 LOY	FF9966 255,153,102 LOR	FF6633 255,102,51 LRO	CC3300 204,51,0 DRO												CC0033 204,0,51 DRP
CCFF00 204,255,0 YYS	CCFF33 204,255,51 LYS	333300 51,51,0 OWY	666600 102,102,0 ODY	999900 153,153,0 DFY	CCCC00 204,204,0 DHY	FFFF00 255,255,0 Y	CC9933 204,153,51 MOY	CC6633 204,102,51 MOR	330000 51,0,0 OWR	660000 102,0,0 ODR	990000 153,0,0 DFR	CC0000 204,0,0 DHR	FF0000 255,0,0 R	FF3366 255,51,102 LRP	FF0033 255,0,51 RRP							
99FF00 153,255,0 SSY	CCFF66 204,255,102 LSY	99CC33 153,204,51 MSY	666633 102,102,51 DWY	999933 153,153,51 DDY	CCCC33 204,204,51 MPY	FFFF33 255,255,51 LHY	996600 153,102,0 DOY	993300 153,51,0 DOR	663333 102,51,51 DWR	993333 153,51,51 DDR	CC3333 204,51,51 MFR	FF3333 255,51,51 LHR	CC3366 204,51,102 MPR	FF6699 255,102,153 LPR	FF0066 255,0,102 PPR							
66FF00 102,255,0 SSG	99FF66 153,255,102 LSG	66CC33 102,204,51 MSG	669900 102,153,0 DBY	999966 153,153,102 MWY	CCCC66 204,204,102 LDY	FFFF66 255,255,102 LFY	996633 153,102,51 DDO	663300 102,51,0 DDO	996666 153,102,102 MWR	CC6666 204,102,102 LOR	FF6666 255,102,102 LPR	990033 153,0,51 DPR	CC3399 204,51,153 MPM	FF66CC 255,102,204 LPM	FF0099 255,0,153 PPM							
33FF00 51,255,0 GGS	66FF33 102,255,51 LGS	339900 51,153,0 DBG	66CC00 102,204,0 DHS	99FF33 153,255,51 LHS	CCCC99 204,204,153 LWY	FFFF99 255,255,153 PDY	CC9966 204,153,102 LDO	CC6600 204,102,0 DHO	CC9999 204,153,153 LWR	FF9999 255,153,153 PDR	FF3399 255,51,153 LHP	CC0066 204,0,102 DHP	990066 153,0,102 DPM	FF33CC 255,51,204 LMP	FF00CC 255,0,204 MMP							
00CC00 0,204,0 DHG	33CC00 51,204,0 DGS	336600 51,102,0 DGS	669933 102,153,51 SDS	99CC66 153,204,102 LDS	CCFF99 204,255,153 PDS	FFFFCC 255,255,204 PWY	FFCC99 255,204,153 PDO	FF9933 255,153,51 LHO	FFCCCC 255,204,204 PWR	FF99CC 255,153,204 PDP	CC6699 204,102,153 LDP	993366 153,51,102 DDP	660033 102,0,51 DGP	CC0099 204,0,153 DMP	330033 51,0,51 DWM							
33CC33 51,204,51 MFG	66CC66 102,204,102 LDG	00FF00 0,255,0 G	33FF33 51,255,51 LHG	66FF66 102,255,102 LFG	99FF99 153,255,153 PDG	CCFFCC 204,255,204 PWG				CC99CC 204,153,204 LWM	996699 153,102,153 MWM	993399 153,51,153 DDM	990099 153,0,153 DFM	663366 102,51,102 DWM	660066 102,0,102 ODM							
006600 0,102,0 ODG	336633 51,102,51 DWG	009900 0,153,0 DFG	339933 51,153,51 DDG	669966 102,153,102 MWG	99CC99 153,204,153 LWG				FFCCFF 255,204,255 PWM	FF99FF 255,153,255 PDM	FF66FF 255,102,255 LFM	FF33FF 255,51,255 LHM	FF00FF 255,0,255 M	CC66CC 204,102,204 LDM	CC33CC 204,51,204 MFM							
003300 0,51,0 OWG	00CC33 0,204,51 DGT	006633 0,102,51 ODT	339966 51,153,102 DDT	66CC99 102,204,153 LDT	99FFCC 153,255,204 PDT	CCFFFF 204,255,255 PWC	3399FF 51,153,255 LHA	99CCFF 153,204,255 PDA	CCCCFF 204,204,255 PWB	CC99FF 204,153,255 PDV	9966CC 153,102,204 LDV	663399 102,51,153 DDV	330066 51,0,102 ODV	9900CC 153,0,204 DMV	CC00CC 204,0,204 DHM							
00FF33 0,255,51 GGT	33FF66 51,255,102 LGT	009933 0,153,51 DTG	00CC66 0,204,102 DHT	33FF99 51,255,153 LHT	99FFFF 153,255,255 PDC	99CCCC 153,204,204 LWC	0066CC 0,102,204 DHA	6699CC 102,153,204 LDA	9999FF 153,153,255 PDB	9999CC 153,153,204 LWB	9933FF 153,51,255 LHV	6600CC 102,0,204 DHV	660099 102,0,153 DVM	CC33FF 204,51,255 LMV	CC00FF 204,0,255 MMV							
00FF66 0,255,102 TTG	66FF99 102,255,153 LTG	33CC66 51,204,102 MTG	009966 0,153,102 DTC	66FFFF 102,255,255 LFC	66CCCC 102,204,204 LDC	669999 102,153,153 MWC	003366 0,51,102 ODA	336699 51,102,153 DDA	6666FF 102,102,255 LFB	6666CC 102,102,204 LDB	666699 102,102,153 MWB	330099 51,0,153 DVB	9933CC 153,51,204 MVM	CC66FF 204,102,255 LVM	9900FF 153,0,255 VVM							
00FF99 0,255,153 TTC	66FFCC 102,255,204 LTC	33CC99 51,204,153 MTC	33FFFF 51,255,255 LHC	33CCCC 51,204,204 MFC	339999 51,153,153 DDC	336666 51,102,102 DWC	006699 0,102,153 DAC	003399 0,51,153 DAB	3333FF 51,51,255 LHB	3333CC 51,51,204 MFB	333399 51,51,153 DDB	333366 51,51,102 DWB	6633CC 102,51,204 MVB	9966FF 153,102,255 LVB	6600FF 102,0,255 VVB							
00FFCC 0,255,204 CCT	33FFCC 51,255,204 LCT	00FFFF 0,255,255 C	00CCCC 0,204,204 DHC	009999 0,153,153 DFC	006666 0,102,102 ODC	003333 0,51,51 OWC	3399CC 51,153,204 MAC	3366CC 51,102,204 MAB	0000FF 0,0,255 B	0000CC 0,0,204 DHB	000099 0,0,153 DFB	000066 0,0,102 ODB	000033 0,0,51 OWB	6633FF 102,51,255 LBV	3300FF 51,0,255 BBV							
00CC99 0,204,153 DCT															3300CC 51,0,204 DBV							
							0099CC 0,153,204 DCA	33CCFF 51,204,255 LCA	66CCFF 102,204,255 LAC	6699FF 102,153,255 LAB	3366FF 51,102,255 LBA	0033CC 0,51,204 DBA											
							00CCFF 0,204,255 CCA	0099FF 0,153,255 AAC	0066FF 0,102,255 AAB	0033FF 0,51,255 BBA													

hexadecimal **HTML**
decimal **RGB**
www.visibone.com/vocccc



This figure simulates deuteranopia, one of many forms of color blindness. Color blindness affects about 1 in 12 men and 1 in 250 women.

Sources:
Christine Rigden, BT Labs,
Thomas Wolfmaier, HCIRN.
www.visibone.com/colorblind

The 216-Color Webmaster's Palette

Here are the colors most widely supported by browsers on the world-wide web. Using colors from this set for backgrounds, fonts and graphics will give your web site the best chance of a consistent appearance across operating systems, color monitors and browser versions. This is particularly true of older computers with limited color palettes that can display only 256 different colors at a time.

Each color chip here is stamped with a hexadecimal HTML color code. CCFF00 These codes can be used to specify the background color of a web page <BODY BGCOLOR="#CCFF00"> or table cell. <TD BGCOLOR= "#00FF00">

To color text, you could use these codes in style sheets H1 {color: #FFCC99} or the officially unfashionable FONT element, <FONT COLOR="#006600"> entrenched by older browser demand.

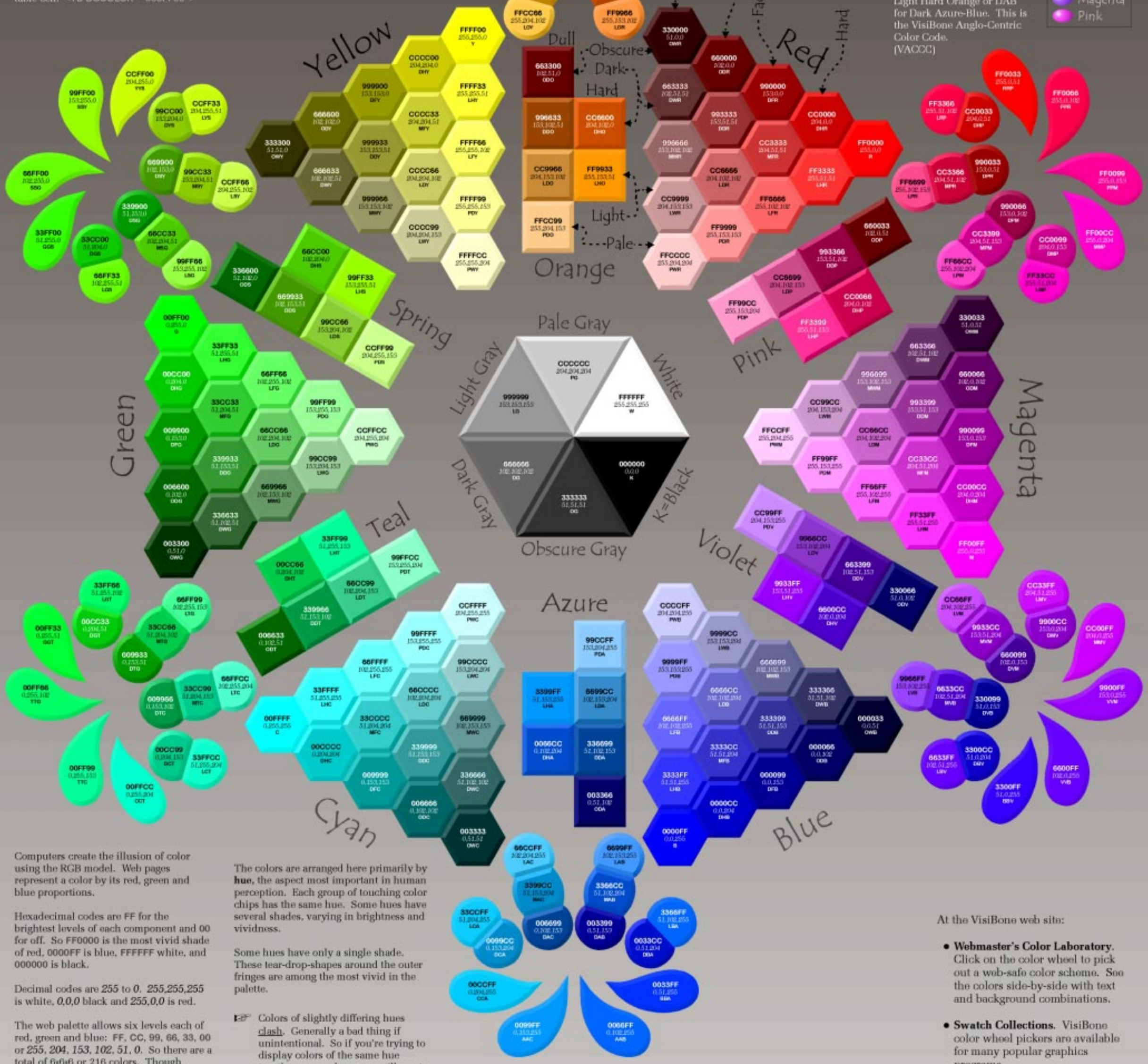
Hyperlink colors use these codes too.
<BODY LINK= "#FFFFCC"
VLINK="#CCCC99"
ALINK="#FFFFFF">

Each color chip is also marked with its decimal RGB values. *204,255,0* These are common in graphics tools and can be used in style sheets.
H1 {color: rgb(255,204,153)}

Each color has a name based on its brightness, vividness and hue.

Brightness	Vividness	Hue
White	Vivid	Red
Pale	Hard	Orange
Light	Faded	Yellow
Medium	Dull	Spring
Dark	Weak	Green
Obscure	Gray	Teal
Black		Cyan
		Azure
		Blue
		Violet
		Magenta
		Pink

The 1-3 letter codes on the chips are abbreviations for the name, such as LHO for Light Hard Orange or DAB for Dark Azure-Blue. This is the VisiBone Anglo-Centric Color Code. (VACCC)



Computers create the illusion of color using the RGB model. Web pages represent a color by its red, green and blue proportions.

Hexadecimal codes are FF for the brightest levels of each component and 00 for off. So FF0000 is the most vivid shade of red, 0000FF is blue, FFFFFF white, and 000000 is black.

Decimal codes are *255* to *0*. *255,255,255* is white, *0,0,0* black and *255,0,0* is red.

The web palette allows six levels each of red, green and blue: FF, CC, 99, 66, 33, 00 or *255, 204, 153, 102, 51, 0*. So there are a total of 6x6x6 or 216 colors. Though computer screen colors can be identified by these numbers, it's much more useful to deal with brightness, vividness and hue when comparing the way colors *look*.

The colors are arranged here primarily by **hue**, the aspect most important in human perception. Each group of touching color chips has the same hue. Some hues have several shades, varying in brightness and vividness.

Some hues have only a single shade. These tear-drop-shapes around the outer fringes are among the most vivid in the palette.

☞ Colors of slightly differing hues clash. Generally a bad thing if unintentional. So if you're trying to display colors of the same hue together on a web page, you'll want to pick them from the same group.

☞ On many PCs, color codes with **33** or *51* in them are indistinguishable from those with **00** or *0*.

At the VisiBone web site:

- **Webmaster's Color Laboratory.** Click on the color wheel to pick out a web-safe color scheme. See the colors side-by-side with text and background combinations.
- **Swatch Collections.** VisiBone color wheel pickers are available for many popular graphics programs.
- **Posters** and other reference products for sale.

VisiBone

www.visibone.com