

HTML5 Peeks, Pokes and Pointers

Common abbreviations (you'll see these throughout this chart):

`$new=document.createElement`

`$bool=function(any){return!(any=="no"||!any)}`

Most new features can be detected in JavaScript. To test for HTML5 video support, create a `<video>` element and check for a property in its DOM: `if("canPlayType" in $new("video")){...}` See Chapter 2: Detecting HTML5 Features.

New elements

See Chapter 3: What Does It All Mean?

Test for support

<code><command></code>	<code>"type" in \$new("command")</code>
<code><datalist></code>	<code>"options" in \$new("datalist")</code>
<code><details></code>	<code>"open" in \$new("details")</code>
<code><output></code>	<code>"value" in \$new("output")</code>
<code><progress></code>	<code>"value" in \$new("progress")</code>
<code><meter></code>	<code>"value" in \$new("meter")</code>
<code><time></code>	<code>"valueAsDate" in \$new("time")</code>

Text annotations `<ruby>`, `<rt>`, `<rp>`

Semantics Usable in all browsers. IE < 9 requires a shim.

`<article>`, `<aside>`, `<header>`, `<hgroup>`, `<footer>`,
`<section>`, `<nav>`, `<figure>`, `<figcaption>`, `<mark>`,
`<summary>`

Newly documented `<embed>`, `<keygen>`, `<wbr>`

Obsolete Still supported, but won't validate.

`<basefont>`, `<big>`, `<center>`, ``, `<s>`, `<strike>`,
`<frame>`, `<frameset>`, `<noframes>`, `<applet>`, `<dir>`,
`<isindex>`, `<tt>`, `<u>`, `<acronym>` (use `<abbr>`)

Always quote your attribute values unless you're a rockstar.
Keep your trailing slashes if you like. Validation is still cool:
html5.validator.nu

Forms

5 > 2

Backward-compatible with HTML 4. See Chapter 9: A Form of Madness.

Test for support

Validation	<code>"noValidate" in \$new("form")</code>
Regex constraint	<code>"pattern" in \$new("input")</code>
Placeholder text	<code>"placeholder" in \$new("input")</code>
Autofocus	<code>"autofocus" in \$new("input")</code>
Required	<code>"required" in \$new("input")</code>

New input types Browsers may customize style or input methods.

<code>type="search"</code>	search box
<code>type="number"</code>	spinbox
<code>type="range"</code>	slider
<code>type="color"</code>	color picker
<code>type="tel"</code>	telephone number
<code>type="url"</code>	web address
<code>type="email"</code>	email address
<code>type="date"/"time"/"month"/"week"/"datetime"</code>	

Test for new input types All tests follow the same pattern.

```
function() {  
    var i = $new("input");  
    i.setAttribute("type", "search");  
    return i.type !== "text";  
}
```

Some browsers claim to "support" an input type but offer no default user interface. Modernizr can detect this.

Multimedia

look ma, no plugins

Encode video with Firefogg (Theora), HandBrake (H.264), or Miro Video Encoder (multiple). See Chapter 5: Video on the Web.

<audio> support	"canPlayType" in \$new("audio")		
Vorbis	\$bool(\$new("audio").canPlayType('audio/ogg; codecs="vorbis"'))		
MP3	\$bool(\$new("audio").canPlayType('audio/mpeg;'))		
AAC	\$bool(\$new("audio").canPlayType('audio/mp4; codecs="mp4a.40.2"'))		
<video> support	"canPlayType" in \$new("video")		
WebM	\$bool(\$new("video").canPlayType('video/webm; codecs="vp8, vorbis"'))		
Theora	\$bool(\$new("video").canPlayType('video/ogg; codecs="theora"'))		
H.264	\$bool(\$new("video").canPlayType('video/mp4; codecs="avc1.42E01E, mp4a.40.2"'))		
Properties	These apply to both <audio> and <video> elements.		
src	string	currentSrc	string, read-only
preload	string	tracks	array of TimedTrack objects
currentTime	in seconds	buffered	TimeRanges object, read-only
initialTime	in seconds, read-only	played	TimeRanges object, read-only
duration	in seconds, read-only	seekable	TimeRanges object, read-only
startOffsetTime	datetime, read-only	networkState	enumerated, read-only
paused	boolean	readyState	enumerated, read-only
ended	boolean, read-only	error.code	enumerated, read-only
autoplay	boolean		
loop	boolean		
controls	boolean		
volume	0.0 to 1.0, default = 1.0		
muted	boolean		
playbackRate	default = 1.0		

List multiple <source> elements in an <audio> or <video>. HTML5-supporting browsers don't render children of <video>, so put your Flash fallback there. Audio and video must be served with the proper MIME type, so check your Content-Type headers!

Offline

See Chapter 8: Let's Take This Offline.

Test for support	window.applicationCache
<html manifest>	Links to cache manifest.
Cache manifest sections	
CACHE:	Always cached. No wildcards.
NETWORK:	Never cached. "*" wildcards.
FALLBACK:	Pairs; second is used offline.
Events	First four are most common.
checking	always first
downloading	found manifest, fetching stuff still fetching stuff
progress	
cached	all resources cached
noupdate	manifest hasn't changed
updateready	call swapCache() to activate
obsolete	manifest is 404 (or 410)
error	it all went wrong somewhere

HTTP semantics still apply to resources listed in the cache manifest, so check your Expires and Cache-Control headers. Manifest must be served as text/cache-manifest, so check your Content-Type headers too. If any required resource fails to load, application will not work offline.

Geolocation

IsGeolocationPartOfHTML5.com

Location sharing is always opt-in. See Chapter 6: You Are Here.

Test for support navigator.geolocation

Functions

```
Position getCurrentPosition(callback, err, opt);
long watchPosition(callback, err, opt);
void clearWatch(watchId);
void callback(position); Called on success
void err(positionError); Called on error
```

PositionOptions object

timeout	in milliseconds also milliseconds
maximumAge	
enableHighAccuracy	true or false

Position object (in callback) has timestamp and coords.

Coordinates object Unsupported properties will be null.

latitude	in decimal degrees
longitude	also decimal degrees
altitude	meters above the reference ellipsoid
accuracy	in meters
altitudeAccuracy	also in meters
heading	degrees CCW from true north
speed	in meters/second

PositionError object (in err callback) has message and code:

TIMEOUT, POSITION_UNAVAILABLE,
PERMISSION_DENIED, or UNKNOWN_ERROR

Paths are like tracing in pencil; nothing is drawn until `fill()` or `stroke()`! See Chapter 4: Let's Call It A Draw(ing) Surface.

Basic support	"getContext" in \$new("canvas")		
Text support	typeof \$new("canvas").fillText=="function"		
Functions	Root path is implicit; subpaths must be explicit. <code>drawImage()</code> can also draw video or canvas.		
<code>beginPath();</code>	<code>drawImage(image,dx,dy,dw,dh);</code>	<code>quadraticCurveTo(cpx,cpy,x,y);</code>	
<code>closePath();</code>	<code>rotate(angle);</code>	<code>bezierCurveTo(cp1x,cp1y,cp2x,cp2y,x,y);</code>	
<code>moveTo(x,y);</code>	<code>translate(x,y);</code>	<code>arc(x,y,radius,startAngle,endAngle,anticlockwise);</code>	
<code>lineTo(x,y);</code>	<code>arcTo(x1,y1,x2,y2,radius);</code>	<code>getImageData(sx,sy,sw,sh);</code>	
<code>rect(x,y,w,h);</code>	<code>isPointInPath(x,y);</code>	<code>putImageData(imagedata,dx,dy,x,y,w,h);</code>	
<code>fill();</code>	<code>fillRect(x,y,w,h);</code>	<code>fillText(text,x,y,maxWidth);</code>	
<code>stroke();</code>	<code>strokeRect(x,y,w,h);</code>	<code>strokeText(text,x,y,maxWidth);</code>	
<code>clip();</code>	<code>clearRect(x,y,w,h);</code>	<code>measureText(text);</code>	
<code>save();</code>	<code>setTransform(a,b,c,d,e,f);</code>	<code>createLinearGradient(x0,y0,x1,y1);</code>	
<code>restore();</code>	<code>transform(a,b,c,d,e,f);</code>	<code>createRadialGradient(x0,y0,r0,x1,y1,r1);</code>	
<code>scale(x,y);</code>	<code>createImageData(sw,sh);</code>	<code>createPattern(image,repetition);</code>	
Properties	All properties are read/write. <code>fillStyle</code> and <code>strokeStyle</code> can also be a gradient or pattern.		
<code>fillStyle</code>	CSS color, default = "black"	<code>shadowColor</code>	CSS color, default = "black"
<code>strokeStyle</code>	CSS color, default = "black"	<code>shadowOffsetX</code>	in pixels, default = 0
<code>font</code>	CSS font, default = "10px sans-serif"	<code>shadowOffsetY</code>	in pixels, default = 0
<code>textAlign</code>	enumerated, default = "start"	<code>shadowBlur</code>	in pixels, default = 0
<code>textBaseline</code>	enumerated, default = "alphabetic"		
<code>globalAlpha</code>	0.0 (transparent) to 1.0 (opaque)		
<code>lineWidth</code>	in pixels, default = 1		
<code>lineCap</code>	enumerated, default = "butt"		
<code>lineJoin</code>	enumerated, default = "miter"		
<code>miterLimit</code>	float, default = 10		

Learn about states! A canvas state includes the clipping path, all properties, and all transformations. `save()` pushes a state onto the stack and `restore()` pops it off.

Bits & Bytes

Doctype	<!DOCTYPE html> Triggers standards-based rendering in all browsers. It's 2011; stop relying on "quirks mode"!		
Text encoding	<meta charset="utf-8"> Always declare a charset, even if you're a rockstar. UTF-8 is always a safe choice.		
Optional end tags	<html>, <head>, <body>, , <p>, <dt>, <dd>, <colgroup>, <option>, <optgroup>, <rt>, <rp>, <thead>, <tbody>, <tfoot>, <tr>, <td>. Exception: always close <p> before <table> to avoid IE weirdness.		
Optional start tags	<html>, <head>, <body>, <tbody>, <colgroup>. Amaze your friends! Skip the <html> tags and still validate!		
New attributes	<a media>, <a ping>, <base target>, <style scoped>, <script async>, <ol reversed>		
Miscellaneous tests	See The All-In-One Almost-Alphabetical Guide to Detecting Everything.		
IndexedDB	<code>window.indexedDB</code>	<code>contentEditable</code>	"isContentEditable" in \$new("a")
Web Workers	<code>window.Worker</code>	Drag-and-drop	"draggable" in \$new("span")
Web Sockets	<code>window.WebSocket</code>	File API	<code>typeof FileReader!="undefined"</code>
X-doc messaging	<code>window.postMessage</code>	Undo history	<code>typeof UndoManager!="undefined"</code>
Web SQL	<code>window.openDatabase</code>	<iframe sandbox>	"sandbox" in \$new("iframe")
Web Storage	"localStorage" in window && window["localStorage"] !== null See Chapter 7: Local Storage.		
History API	<code>window.history</code> && <code>window.history.pushState</code> See Chapter 11: History API.		
Inline SVG	<code>function() { var e=\$new("div"); e.innerHTML="<svg></svg>"; return window.SVGSVGELEMENT && e.firstChild instanceof window.SVGSVGELEMENT}</code>		

Credits

Writing	Mark Pilgrim	References	HTML5, HTML-diff, HTML vs. XHTML
Code	Modernizr (Paul Irish et. al.)	Inspiration	Beagle Bros
Typography	Chunk, Latin Modern	License	CC-BY-3.0

