HTML5

Dive Into HTML5

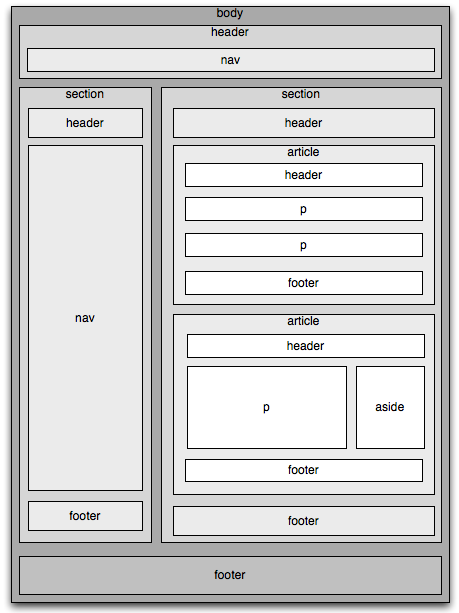
CheatSheet

* <http://diveintohtml5.info/peeks-pokes-and-pointers.html>

Detection

* Use [Modernizr](http://www.modernizr.com/) to detect HTML5.
* Include <script src=”modernizr.min.js” />
* If(Modernizr.xyzFeature) { … } else { … } //xyzFeature e.g. .canvas, .canvastext, .geolocation etc…
* Video: Modernizr.video, .video.webm(for firefox, chrome), .video.ogg(for firefox, chrome), .video.h264(for safari, ie10)
  + A no js solution to display HTML5 video but fallback to plugin automatically: [Video for everyone](http://camendesign.com/code/video_for_everybody)
* Modernizr.localstorage // p.s. modernizr use all lowercase for properties, while DOM property use window.localStorage
  + Security: Same-origin restriction, only the original site can r/w the value
* Modernizr.webworkers – DOM window.Worker
* Modernizr.applicationcache – DOM window.applicationCache
* Modernizr.inputtypes.date/email/url/search/color/… // all together 13 new input types
* Modernizr.input.placeholder – placeholder text is displayed inside the input field using a (placeholder attribute) as long as the field is empty and not focused. As soon the input field got focused, the placeholder text disappear.
* Modernizr.input.autofocus – put focus on the 1st input field using an autofocus attribute.
* Modernizr.history – deep linking in Ajax app
* PolyFill – to provide fallback when HTML5 features not supported by the browser. See [here](https://github.com/Modernizr/Modernizr/wiki/HTML5-Cross-browser-Polyfills).

Semantics

* HTML5 specific doctype: <!DOCTYPE html> // just these 15 characters ONLY
* HTML5 root element: <html lang=”en”> // no need of xmlns or xml:lang
* <meta charset=”utf-8” /> // no need for http-equiv= or content=
* <link rel=”stylesheet” href=”xyz.css” />
* [Have old browser handle unknown elements](http://diveintohtml5.info/semantics.html#unknown-elements)
* **<section> <nav> <article> <aside>(think sidebar) <hgroup> <header> <footer> <time> <mark>**
* <**hgroup**>: group related heading elements <h1>-<h6> together and use it as a single node. Excellent explanation [here](http://diveintohtml5.info/semantics.html#header-element).
* HTML5 <h1>-<h6> inside an <article>: <h1>-<h6> will be self-contained with the <article>, it won’t be affected by other <h1>-<h6> outside their <article>!
* <time datetime=”[machine-readable-datetime](http://www.whatwg.org/specs/web-apps/current-work/multipage/common-microsyntaxes.html#valid-global-date-and-time-string)” pubdate>human readable datetime</time>
  + pubdate: publication day of article if is within an <article> else the whole document
* 

Canvas

* Html
  + <canvas width=”” height=”” > </canvas>
* Context

var b\_canvas = document.getElementById(“myCanvas”);

var b\_context = b\_canvas.**getContext**(“2d”);

b\_context.**someDrawingMethod**();

* Rectangle: .**fillRect**(), **strokeRect**(), **clearRect**()
* **Path**
  + Line: context.**moveTo**(), .**lineTo**(), .**strokeStyle**, .**stroke**() // move/line simply trace it, stroke will “ink” it
  + context.**beginPath**() to start a new path, context.closePath() to end the path
  + .arc(…) to draw circle
* **Text**
  + **Default** font, style: inherit from **canvas**.
  + Override at **context** e.g.: .**font**=”bold 12px sans-serif”, .**texAlign**=”left”, .**textBaseline**=”top”, .**fillText**(“zombie”, 0, 0), .**fillStyle**
* **Gradient**
  + var myGradient = context.**createLinear/RadialGradient**(0,0,300,0);

myGradient.**addColorStop**(0, “black”);

myGradient.**addColorStop**(0, “white”);

context.**fillStyle** = myGradient;

context.fillRect(…);

* **Image**
  + From **<img>**: var image = document.getElementById(“myImgId”);
  + **Dynamic**: var image = **new Image()**; image.src = “xyz.png”;
  + Finally **draw** it on canvas: context.**drawImage**(**image**, values for position, clipping & scaling);
  + Pre-ie9, use excanvas.js to support the HTML5 canvas API<http://code.google.com/p/explorercanvas/>
* **Tutorial**
  + [Canvas tutorial](https://developer.mozilla.org/en/Canvas_tutorial) on Mozilla Developer Center
  + [HTML5 canvas](http://dev.opera.com/articles/view/html-5-canvas-the-basics/) — the basics, by Mihai Sucan

Video

* can containe 1+ urls, browser will choose the 1st one it can play
* <video src=”xyz.webm” width=”” height=”” ***controls* [preload autoplay]**/>
  + ***controls***: tell browser to show the built-in control buttons
  + custom control buttons: event handler calls .play(), .pause(), .currentTime, .volume, .muted
* 1+ video files: <video width=”” height=””> <source src=”x.mp4” type=”…” > <source src=”x.webm” type=”…” >
* A complete [example](http://diveintohtml5.info/video.html#example).
* Pre-built custom controls for HTML5 video: [VideoJS](http://videojs.com/), [MediaElement.js](http://mediaelementjs.com/), [Kaltura](http://www.kaltura.org/project/HTML5_Video_Media_JavaScript_Library).

Geolocation

* **navigator.geolocation.getCurrentPosition**(*myCallback, errorHandler, {enableHighAccuracy|maximumAge|timeout}*); // browser will then popup the opt-in UI
* function *myCallback*(position) { position.**coords.latitude/longtitue/accuracy/altitude /altitudeAccuracy/heading/speed** … } // only the 1st 3 properties are guarantee, the rest depends on hardware.
* function errHandler(err) { err.code == 0/1/2/3 … } // 0: unknown error, 1: denied, 2: unavailable, 3: timeout
* continuous location update: .watchPosition(*myCallback, errorHandler*)
* Shim for ALL Browsers: [geo.js](http://code.google.com/p/geo-location-javascript/)
* Live [example](http://diveintohtml5.info/geolocation.html#putting-it-all-together).

LocalStorage/ DOMStorage

* Key is a string, **data will always be serialized and stored as string**. Getting the data back requires parseInt/Float etc… , 5 MB
* var foo = window.**localStorage[**“myKey”**]**;
* window.**localStorage[**“myKey”**]** = foo;
* support item changed event

IndexedDB

* A [tutorial](http://hacks.mozilla.org/2010/06/comparing-indexeddb-and-webdatabase/).

AppCache offline app

* Read Urls from manifest file, auto download the Urls, cache them and auto update while the browser is online
* <html **manifest**=”/cache**.manifest**” >
* Must add text/cache-manifest MIME type to your web server
* The current page is implicitly included in the AppCache, any referenced .html files from the current page will need to be explicitly list in the manifest file if you want that to be cached
  + - **Html** file:

<html ***mainifest=”test.appcache”***>

<head></head> <body>

<img src=”***logo.png***” />

<**video** src=”***fish.mp4***” />

<img src=”***kid.png***” />

</body> </html>

* + - **Manifest** file (test.appcache):

Cache Manifest

#1/1/2011 v3

**Cache**: // download from the web in the background and cached it for me

***logo.png***

**Network**: // MUST from the web

***fish.mp4***

**Fallback**: // if kid.png not found then use the noImg.png

***kid.png noImg.png***

* The following will **“Lazy Add”** pages **into** your **AppCache** as you visit them
* NETWORK: **\***  : download ANY resources from ANY domain
* FALLBACK: / /offline.html : the 1st / match any pages on your site, if not found that page in AppCache then display offline.html
* Seems complicated!

Form Input Types (13 of them)

* Placeholder: <input name=”q” **placeholder**=”Your username” />
* Autofocus: <input name=”q” autofocus /> // a [fallback](http://diveintohtml5.info/examples/input-autofocus-with-fallback-document-ready.html) solution.

1. [<input type="**search**">](http://www.whatwg.org/specs/web-apps/current-work/multipage/states-of-the-type-attribute.html#text-state-and-search-state) for search boxes which could have a **“clear” button**
2. [<input type="**number**" **min**=”0” **max**=”10” **step**=”2” **value**=”6”>](http://www.whatwg.org/specs/web-apps/current-work/multipage/number-state.html#number-state) for **spinboxes**
   * programmatic: input.**stepUp/Down(n)**, input.**valueAsNumber** vs input.**value** -> **float vs string**
3. [<input type="**range**" **min**=”0” **max**=”10” **step**=”2” **value**=”6”>](http://www.whatwg.org/specs/web-apps/current-work/multipage/number-state.html#range-state) for sliders
4. [<input type="**color**">](http://www.whatwg.org/specs/web-apps/current-work/multipage/number-state.html#color-state) for color pickers
5. [<input type="**tel**">](http://www.whatwg.org/specs/web-apps/current-work/multipage/states-of-the-type-attribute.html#telephone-state) for telephone numbers
6. [<input type="**url**">](http://www.whatwg.org/specs/web-apps/current-work/multipage/states-of-the-type-attribute.html#url-state) for web addresses
7. [<input type="**email**">](http://www.whatwg.org/specs/web-apps/current-work/multipage/states-of-the-type-attribute.html#e-mail-state) for email addresses
8. [<input type="**date**">](http://www.whatwg.org/specs/web-apps/current-work/multipage/states-of-the-type-attribute.html#date-state) for calendar date pickers
9. [<input type="**month**">](http://www.whatwg.org/specs/web-apps/current-work/multipage/states-of-the-type-attribute.html#month-state) for months
10. [<input type="**week**">](http://www.whatwg.org/specs/web-apps/current-work/multipage/states-of-the-type-attribute.html#week-state) for weeks
11. [<input type="**time**">](http://www.whatwg.org/specs/web-apps/current-work/multipage/states-of-the-type-attribute.html#time-state) for timestamps
12. [<input type="**datetime**">](http://www.whatwg.org/specs/web-apps/current-work/multipage/states-of-the-type-attribute.html#date-and-time-state) for precise, absolute date+time stamps
13. [<input type="**datetime-local**">](http://www.whatwg.org/specs/web-apps/current-work/multipage/states-of-the-type-attribute.html#local-date-and-time-state) for local dates and times

* Validation:
  + the browser will AUTO validate the fields w/o requiring ANY JavaScript
  + if validation failed, the browser WILL NOT ALLOW SUBMIT.
  + Default: ON for ALL fields. Use <form novalidate> to turn OFF.
  + Required field: <input id=”q” required >

History API

* Programmatic Add entries (forward), subscribe to Remove event (backward) w/o reload the page
* history.pushState(JSON\_data, null, locationBarUrl);