

New Media Production

- HTML5

Modernizr, an HTML5 Detection Library

```
<head>  
<script src="modernizr.min.js"></script>  
</head>
```

Modernizr is an open source, MIT-licensed JavaScript library that detects support for many HTML5 & CSS3 features

Modernizr runs automatically. There is no `modernizr_init()` function to call.

Canvas

HTML5 defines the `<canvas>` element as "a resolution-dependent bitmap canvas that can be used for rendering graphs, game graphics, or other visual images on the fly."

A canvas is a rectangle in your page where you can use JavaScript to draw anything you want.

Video

HTML5 defines a new element called `<video>` for embedding video in your web pages. Embedding video used to be impossible without third-party plugins such as Apple QuickTime or Adobe Flash.

You can specify multiple video files, and browsers that support HTML5 video will choose one based on what video formats they support.

Local Storage

HTML5 storage provides a way for web sites to store information on your computer and retrieve it later.

Similar to cookies, but designed for larger quantities of information.

HTML5 storage stays on your computer, and web sites can access it with JavaScript after the page is loaded.

Offline Web Applications

The first time you visit an offline-enabled web site, the web server tells your browser which files it needs in order to work offline. These files can be anything — HTML, JavaScript, images, even videos.

Once your browser downloads all the necessary files, you can revisit the web site even if you're not connected to the Internet.

When you get back online, any changes you've made can be uploaded to the remote web server.

Geolocation

Geolocation is the art of figuring out where you are in the world and (optionally) sharing that information with people you trust.

There is more than one way to figure out where you are — your IP address, your wireless network connection, which cell tower your phone is talking to, or dedicated GPS hardware that calculates latitude and longitude from information sent by satellites in the sky.

HTML5 form input types

`<input type="search">` for search boxes

`<input type="number">` for spinboxes

`<input type="range">` for sliders

`<input type="color">` for color pickers

`<input type="tel">` for telephone numbers

`<input type="url">` for web addresses

`<input type="email">` for email addresses

`<input type="date">` for calendar date pickers

`<input type="month">` for months

`<input type="week">` for weeks

`<input type="time">` for timestamps

`<input type="datetime">` for precise, absolute
date+time stamps

`<input type="datetime-local">` for local dates and
times

DTD

```
<!DOCTYPE html  
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

HTML5:

```
<!DOCTYPE html>
```

Root Element

```
<html xmlns="http://www.w3.org/1999/xhtml"  
  lang="en"  
  xml:lang="en">
```

HTML5:

```
<html lang="en" xml:lang="en">
```

Character Encoding

```
<meta http-equiv="Content-Type"  
content="text/html; charset=utf-8">
```

HTML5:

```
<meta charset="utf-8" />
```

Link Relations

Regular links (`<a href>`) simply point to another page.

Link relations are a way to explain why you're pointing to another page.

- ...it's a stylesheet containing CSS rules
- ...it's a feed that contains the same content as this page
- ..it's a translation of this page into another language.
- ...it's the same content as this page, but in PDF format
- ...it's the next chapter of an online book

Link Relations: Two Categories

HTML5 breaks link relations into two categories:

Links to external resources

are links to resources that are to be used to augment the current document

hyperlink links

are links to other documents.

rel = stylesheet

```
<link rel="stylesheet" href="style.css"  
type="text/css" />
```

HTML5:

```
<link rel="stylesheet" href="style.css" />
```

rel = alternate

```
<link rel="alternate"  
      type="application/rss+xml"  
      title="My Weblog feed"  
      href="/feed/" />
```

HTML5:

(Unlike with rel="stylesheet", the type attribute matters here. Don't drop it!)

`<section> ... </section>`

The section element represents a generic document or application section. A section, in this context, is a thematic grouping of content, typically with a heading.

Ex: an introduction, news items, contact information.

`<nav> ... </nav>`

The `nav` element represents a section of a page that links to other pages or to parts within the page: a section with navigation links.

Not all groups of links on a page need to be in a `nav` element

Ex: major navigation blocks are appropriate for the `nav` element

`<article> ... </article>`

The article element represents a component of a page that consists of a self-contained composition in a document, page, application, or site.

Intended to be independently distributable or reusable, e.g. in syndication.

Ex: forum post, blog entry, an interactive widget

`<aside> ... </aside>`

The `aside` element represents a section of a page that consists of content that is tangentially related to the content around the `aside` element.

Can be considered separate from that content.

Ex: pull quotes, sidebars, advertising

`<hgroup> ... </hgroup>`

The `hgroup` element represents the heading of a section. The element is used to group a set of `h1`–`h6` elements when the heading has multiple levels

Ex: subheadings, alternative titles, or taglines

`<header> ... </header>`

The header element represents a group of introductory or navigational aids.

A header element is intended (but not required) to usually contain the section's heading (an h1-h6 element or an hgroup element)

Ex: table of contents, a search form, or any relevant logos

`<footer> ... </footer>`

The footer element represents a footer for its nearest ancestor sectioning content or sectioning root element.

A footer typically contains information about its section such as who wrote it, links to related documents, copyright data, etc...

When the footer element contains entire sections, they represent appendices, indexes, long colophons, etc...

<time> ... </time>

The time element represents either a time on a 24 hour clock, or a precise date in the proleptic Gregorian calendar, optionally with a time and a time-zone offset.

```
<time datetime="2009-10-22T13:59:47-04:00" pubdate>
```

October 22, 2009 1:59pm EDT

```
</time>
```

`<mark> ... </mark>`

The mark element represents a run of text in one document marked or highlighted for reference purposes.

Browsers: unknown elements

Every browser has a master list of HTML elements that it supports.

All browsers render unknown elements inline, i.e. as if they had a `display:inline` CSS rule.

```
article,aside,footer,header,hgroup,  
nav,section {  
    display:block;  
}
```