

HTML5 & CSS3

A chance to Do things Differently

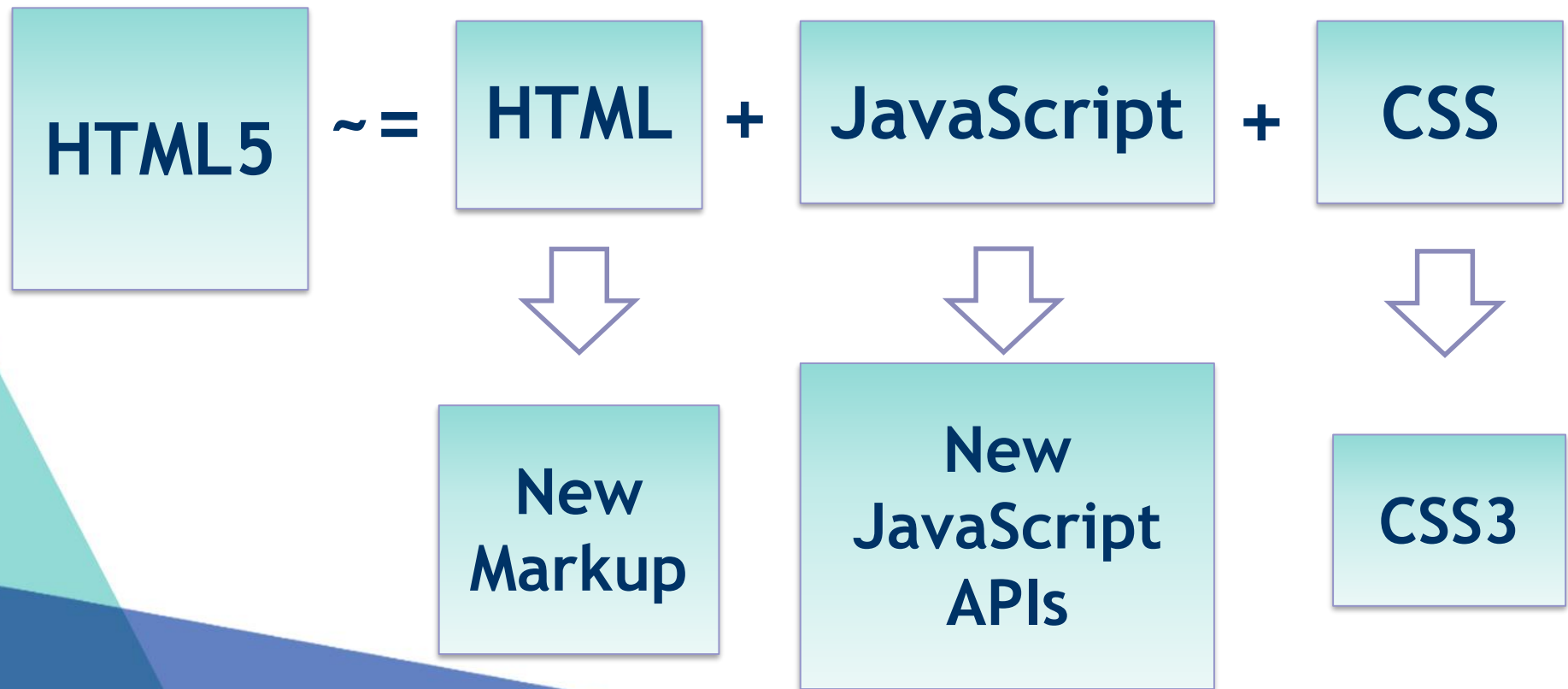
Eng. Niveen Nasr El-Den
iTi



Day 1

What is HTML5

HTML5 is a Constellation of technology



HJML5

A Record of Tomorrow

Overview of Enhancements

- Structure and Semantics
- Forms
- Microdata
- CSS
- Embedded Content and Multimedia
- DOM APIs drag and drop
- Web Storage
- Web worker
- ...



HTML5

includes new elements for
better structure,
better form handling,
drawing, and
media content



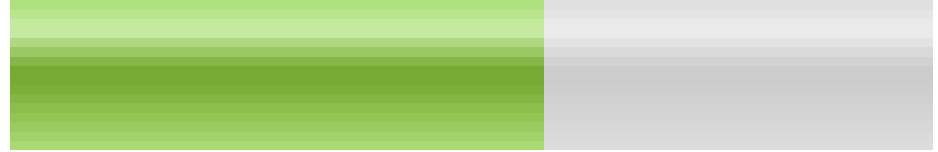
New Elements

<meter>

- Representing scalar measurements or fractional values
- Meter is also known as a **gauge**
- It should not be used to indicate progress
- Attributes:
 - ▷ **value**
 - ▷ **min**
 - ▷ **max**
 - ▷ **high**
 - ▷ **low**
 - ▷ **optimum**

Using <meter>

```
<meter value="0.6"  
        min=""  
        max=""  
        optimum=""  
        low=""  
        high="0.6">  
    Medium  
</meter>
```



<https://css-tricks.com/html5-meter-element/>

<progress>

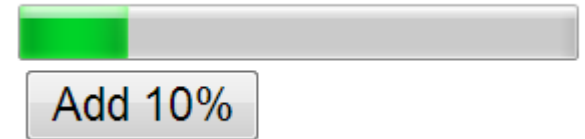
- Show completion progress of a task
- Progress bars are widely used in other applications
- Works with scripted applications
- Attributes
 - value : Specifies how much of the task has been completed
 - max : Specifies how much work the task requires in total
- Useful for:
 - Indicate loading progress of an AJAX application
 - Show user progress through a series of forms
 - Making impatient users wait

Using <progress>

<progress max="20">

Step 3 of 6

</progress>



<progress value="0.5">

Half way!

</progress>

<progress id="pBar" max="100" value="0">

Nothing...

</progress>

<mark>

- Marked or Highlighted text
- Indicates point of interest or relevance
- Useful for:
 - ▷ Highlighting relevant code in a code sample
 - ▷ Highlighting search keywords in a document

Using <mark>

The highlighted part below is where the error lies:

```
var i: Integer;  
begin  
    i := 1.1;  
end.
```

<p>The highlighted part below is
where the error lies:</p>

```
<pre><code>var<var> i</var>: Integer;  
begin  
    i := <mark>1.1</mark>;  
end.</code></pre>
```

is used for
computer code

is used to indicate a
variable within
code.

Using <details> & <summary>

is used for additional details which can be hidden or revealed.

▼ HTML

HTML stands for HyperText Markup Language.

- ▶ CSS
- ▶ JS

Note:
details has "**toggle**" event and "**open**" property set to **true** when it is opened

<details>

<summary> HTML </summary>

<p>HyperText Markup Language.</p>

</details>

is used in conjunction with the <details> tag for specifying a summary.



Forms

Other Form's Controls

- HTML 4 controls are too limited
- Several new types added
- New Input type:

<http://www.coreservlets.com/html5-tutorial/input-types.html>

- ▷ color
- ▷ date
- ▷ datetime
- ▷ datetime-local
- ▷ time
- ▷ month
- ▷ week

- ▷ email
- ▷ number
- ▷ range
- ▷ search
- ▷ tel
- ▷ url

<https://www.wufoo.com/html5/>

Other Form's Controls

- `<input type="tel">`
- `<input type="time">`
- `<input type="color">`
- `<input type="month">`
- `<input type="search">`
- `<input type="number">`
- `<input type="email">`
- `<input type="range">`
- `<input type="date">`
- `<input type="time">`
- `<input type="url">`
- ...

Other Form's Controls

```
<form oninput="res.value =  
    parseInt(p1.value)+parseInt(p2.value)" >  
  <input type="" id="p1">  
  <input type="" id="p2">  
  <output name="res" for="p1 p2">  
</form>
```

is a container used to inject
the results of an output
calculation due to user
action

Note:

The `<output>` "value", "name" attributes,
and "content" are not sent during form
submission

Other Form's Controls

```
<input type="" list="d1">
```

```
<datalist id="d1">
```

```
  <option label="item#1" value="n1"></option>
```

```
  <option value="v1">item</option>
```

```
  <option value="n2"></option>
```

```
  <option value="n3"></option>
```

```
</datalist>
```

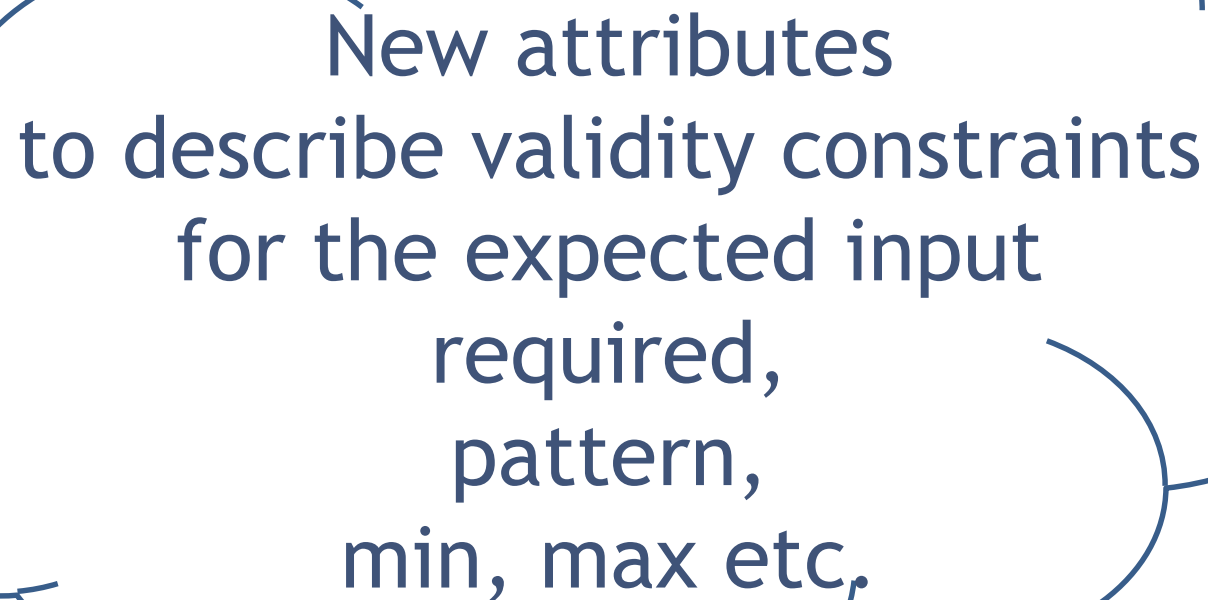
<https://developer.mozilla.org/en-US/docs/Web/HTML/Element/datalist>

<datalist> vs <select>

- <datalist> is used for **suggesting** the possible values from the large array of values relevant to that text filed.
- <select> **won't allow** users to have privilege to input their own value.



Form Validation



New attributes
to describe validity constraints
for the expected input
required,
pattern,
min, max etc.

Form's new Attributes

- `<input type="" required>`
 - ▷ Required: We also have required attribute to mark this field as mandatory.
- `<input type="" autocomplete="off">`
 - ▷ Autocomplete: tells the browser whether or not the value of this input should be saved for future, should be used to protect sensitive user data
- `<input type="" pattern="[0-9][A-Z]{3}">`
 - ▷ Pattern: custom validate, A part number is a digit followed by three uppercase letters.
- `input type="" placeholder="">`
 - ▷ Placeholder: add a hint inside the text-field, but where the hint automatically disappears when clicking inside it.

Form Validation

- We can present Form Validation using
 - ▷ JavaScript Custom Validation
 - ▷ HTML Built-in Form Validation
 - ▷ CSS rule Validation



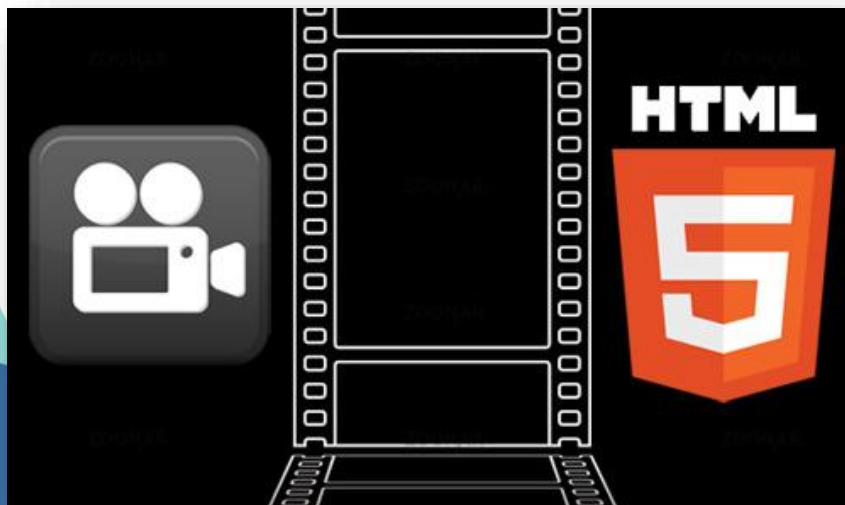
Embedded Content & Multimedia

Multimedia

No need for plugin to play video and audio
HTML 5 will do it for you

Video & Audio

- HTML5 offers the ability to easily embed **media** into HTML documents.
- Media playback can be controlled via JavaScript and media events.



- Nothing to install.
- Works in all browsers and phones (adding native support to browsers).

Native Media format for HTML5

- Video

- ▷ webM
- ▷ H.264(mp4)
- ▷ oggTheora





















- Audio

- ▷ wav
- ▷ mp3
- ▷ oogVorbis

- No common format to use.
- We have to encode in different multiple formats.
- Need of converter to convert into different format.

Native Video Browser Support

https://en.wikipedia.org/wiki/HTML5_video

VP8 (WebM)					
H.264 (MP4)					
Ogg Theora					
					

Native Audio Browser Support

WAV					
MP3					
Ogg Vorbis					
					

Media Attributes

Attribute	Description
src	Specifies the URL of the media source file
controls	Specifies whether or not to display media controls (such as a play/pause button etc).
autoplay	Specifies whether or not to start playing the media as soon as it has been loaded. Depending on browser policy
loop	Specifies whether to keep re-playing the media once it has finished.
poster=""	display a frame of the video (as a .jpg, .png..)
width=""	Specifies the width, in pixels, to display the video .
height=""	Specifies the height, in pixels, to display the video .

■ NOTE:

□ HTML5 isn't extension for XHTML

- There is no need to have a value for each attribute, otherwise set its value either to true or its name

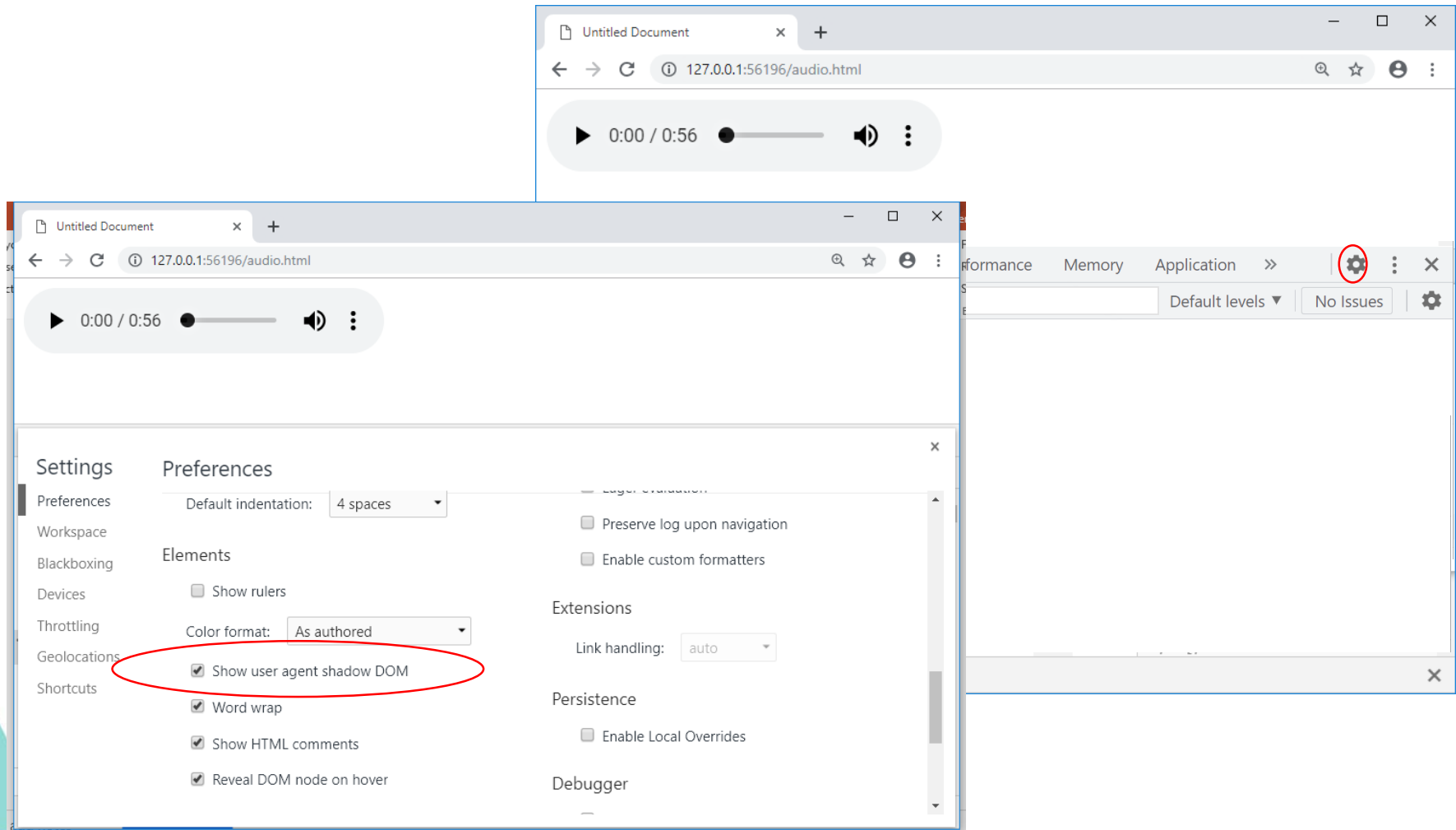
Media Methods & Properties

https://developer.mozilla.org/en/docs/Web/Guide/Events/Media_events

Method	Description
load()	Re-loads the audio/video element
play()	Starts playing the audio/video
pause()	Pauses the currently playing audio/video

- controls
- loop
- autoplay
- played
- paused
- ended
- playbackRate → range [0, 16]
- currentTime
- duration
- src
- muted
- volume → range [0, 1]

Shadow DOM Setting



Shadow DOM Example

The screenshot shows a web browser window with an audio player. The browser's developer tools are open, displaying the DOM tree and the CSS styles for the shadow root.

The DOM tree shows the following structure:

```
<audio data-brackets-id="6" src="audio/Muppet%20Show.mp3" controls>
  #shadow-root (user-agent)
    <div pseudo="-webkit-media-controls" class="phase-ready state-stopped">
      <div pseudo="-webkit-media-controls-overlay-enclosure">...</div>
      <div pseudo="-webkit-media-controls-enclosure">
        <div pseudo="-webkit-media-controls-panel">
          <input type="button" pseudo="-webkit-media-controls-play-button" aria-label="play" class="pause" style>
            #shadow-root (user-agent)
              <div pseudo="-internal-media-controls-button-hover-background"></div> == $0
          </input>
          <div aria-label="elapsed time: 0:00" pseudo="-webkit-media-controls-current-time-display" style>0:00</div>
          <div aria-label="remaining time: / 0:56" pseudo="-webkit-media-controls-time-remaining-display" style>/ 0:56</div>
        </div>
      <input type="range" step="any" pseudo="-webkit-media-controls-timeline" max="56.751188" aria-valuetext="0:00">
        #shadow-root (user-agent)
          <div>
            <div pseudo="-internal-media-controls-segmented-track" id="track">
              <div id="thumb"></div>
            </div>
          </div>
        /* Copyright (c) 2017 The Chromium Authors. All rights reserved.
           Use of this source code is governed by a BSD-style license that can be
           found at https://chromium.googlesource.com/external/webkit/+/master/src/media/controls/
```

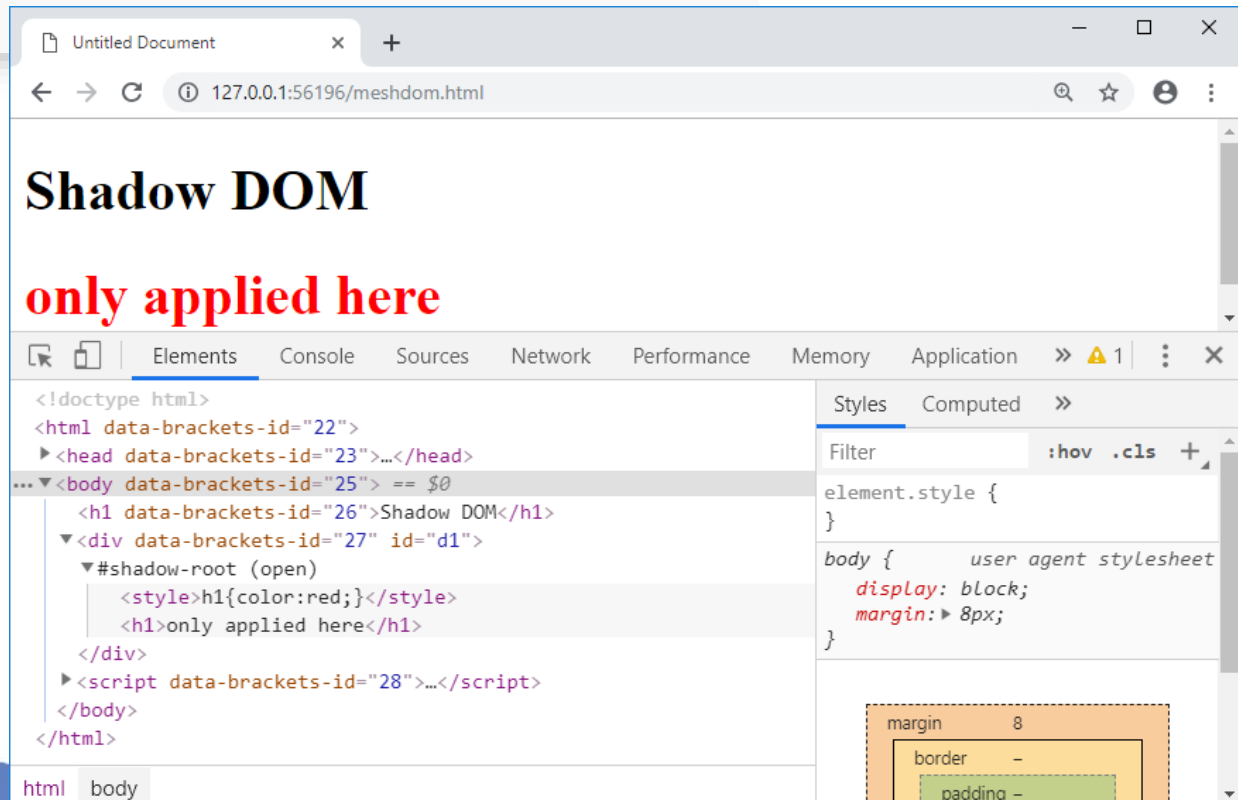
The browser's address bar shows the URL: 127.0.0.1:56196/audio.html. The browser's status bar shows the text: Copyright (c) 2017 The Chromium Authors. All rights reserved. Use of this source code is governed by a BSD-style license that can be found at https://chromium.googlesource.com/external/webkit/+/master/src/media/controls/

The browser's developer tools show the following CSS styles for the shadow root:

```
element.style {
  video::-webkit-media-controls.audio-only
  input[pseudo="-webkit-media-controls-play-button" i]::-internal-media-controls-button-hover-background, video::-webkit-media-controls.audio-only
  input[pseudo="-webkit-media-controls-fullscreen-button" i]::-internal-media-controls-button-hover-background, video::-webkit-media-controls.audio-only
  input[pseudo="-internal-media-controls-overflow-button" i]::-internal-media-controls-button-hover-background, audio::-webkit-media-controls input[pseudo="-webkit-media-controls-play-
```

Shadow DOM Example

```
<h1>Shadow DOM</h1>
<div id="d1"></div>
<script>
  document.getElementById("d1").attachShadow({
    mode: 'open' // 'closed'
  }).innerHTML = "\
    <style>h1{color:red;}</style>\
    <h1 id='aaa'>only applied here</h1>"
</script>
```



HTML5 data Attributes

- Store some extra information that doesn't have any visual representation.
- The name of a custom data attribute begins with **data-***, and must be at least one character long after this prefix.
- The attribute value can be any string that contains only [a-z], [0-9], (-), (.), (:), (_).
- It should not contain ASCII capital letters (A to Z).



Semantics & Structured Data

Semantic HTML

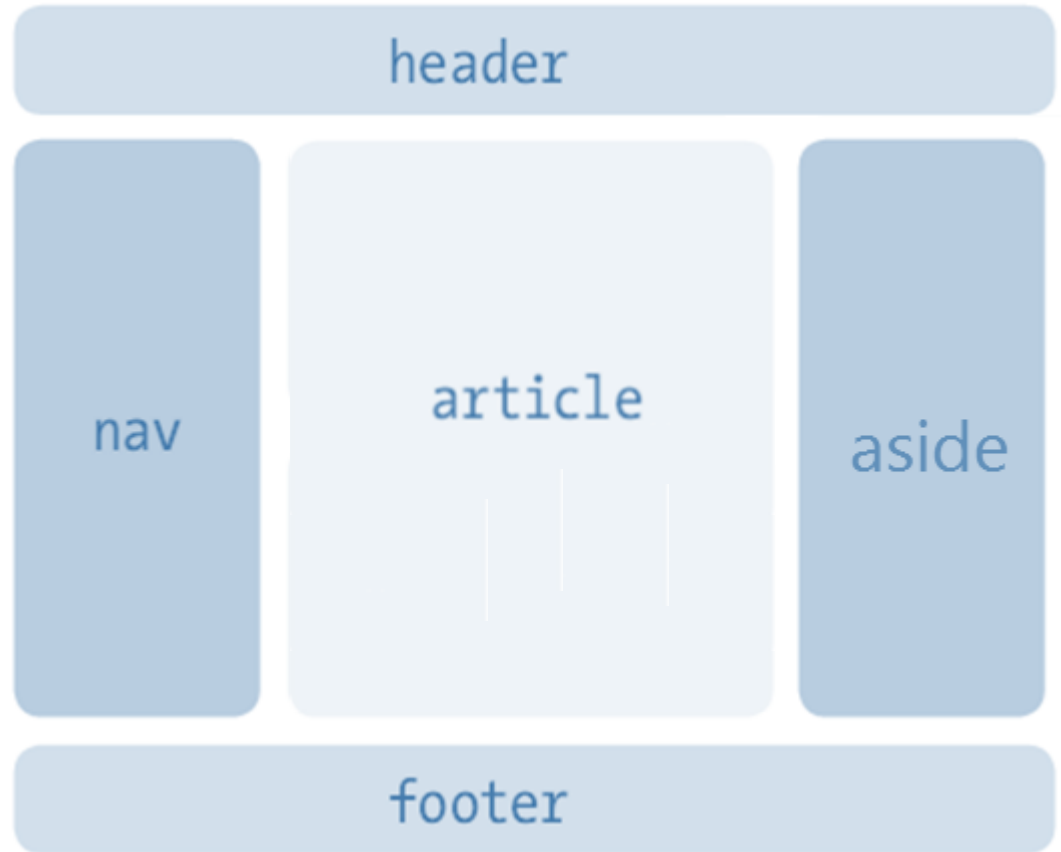
- Semantic HTML is the idea of using HTML elements for what they are rather than how they may appear in the browser by default.
- It is HTML that introduces meaning to the web page rather than just presentation.
- Semantic tags make it clear to the browser what the meaning of a page and its content is.

**Elements clearly describes its meaning to both
browser & developer**

New Semantic Elements

- Markups that describe its content without presenting it e.g.

- ▷ `<header>`
- ▷ `<hgroup>`
- ▷ `<nav>`
- ▷ `<section>`
- ▷ `<article>`
- ▷ `<aside>`
- ▷ `<footer>`
- ▷ etc.



Structured Data

- Structured data is a standardized format for providing information about a page and classifying the page content
- It helps understanding more about the web page and show better headline text, images etc..
- Search engines use **structured data** to
 - understand the content of the page
 - gather information about the web and the world in general.
 - enable special search result features and enhancements like appear in a graphical search result.

https://developers.google.com/search/docs/guides/intro-structured-data?visit_id=636758461413700902-4066872858&rd=1

Structured Data Supported formats

- JSON-LD (<https://json-ld.org/>)

<https://developers.google.com/search/docs/guides/sd-policies>

- ▷ JavaScript Object Notation for Linked Data
- ▷ JSON-LD uses a JavaScript object in your HTML page to define data.
- ▷ JavaScript notation embedded in a <script> tag in the page head or body.

- Microdata

- ▷ Microdata uses HTML tags and **attributes** to define data like RDFa.
- ▷ It nests the structured data within HTML content
- ▷ It is typically used in the page body, but can be used in the head.

- RDFa (<http://rdfa.info/>)

<https://developer.mozilla.org/en-US/docs/Web/HTML/Microdata>

- ▷ Resource Description Framework in **Attributes**
- ▷ Commonly used in both the head and body sections of the HTML page.

Microdata

- Microdata a new lightweight semantic meta-syntax.
<https://www.w3.org/TR/microdata/>
- Microdata defines five HTML attributes that can be applied to any HTML5 tag.
- It helps search engines and other applications better understand our content and display it in a useful, relevant way.
- It gives us a whole new way to *add extra semantic information and extend HTML5*.
- It provides a meaning of an Item.

Microdata

- Instead of elements, these name-value pairs are defined via attributes:

- ▷ **itemscope**

- Indicates the element is a microdata element and its child elements are part of its microdata format.

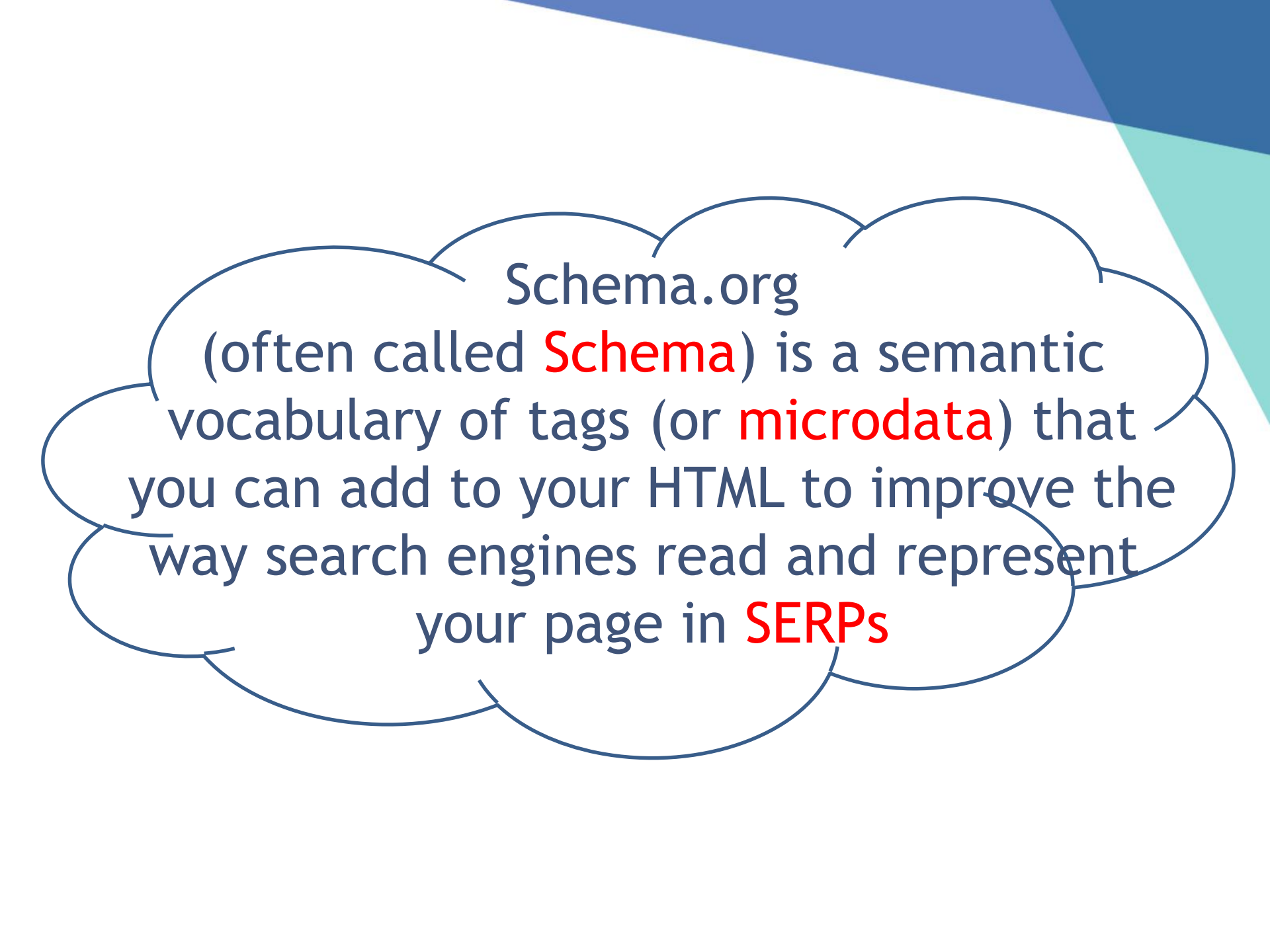
- ▷ **itemprop="property-name"**

- An individual data element that adds a *property* to a microdata item

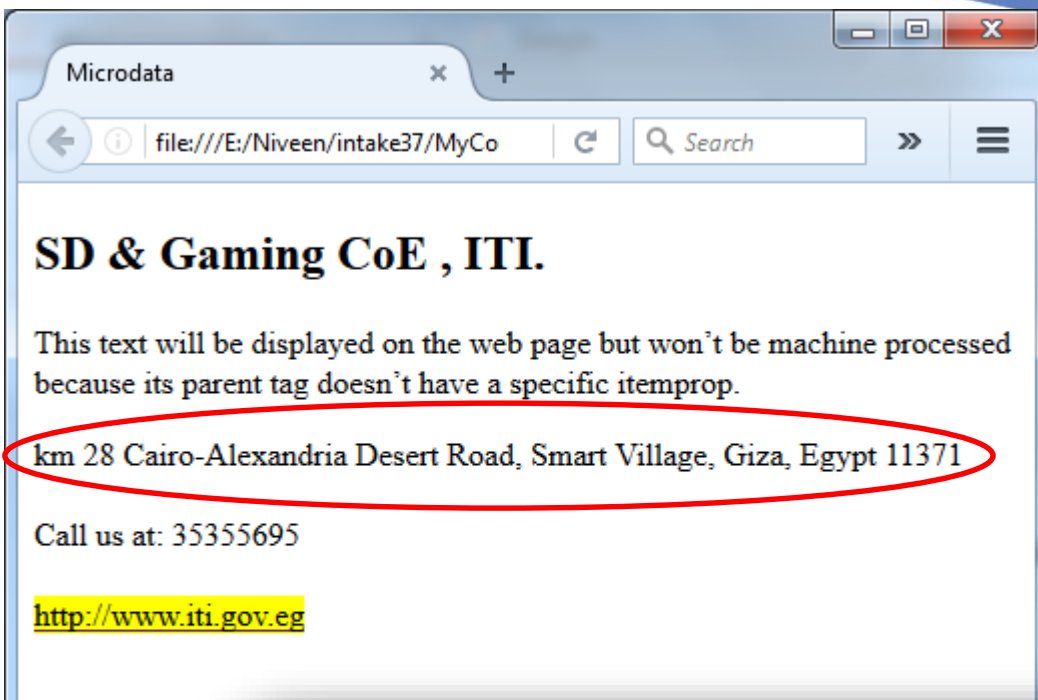
- ▷ **itemtype="URL"**

- Defines the vocabulary to be used by the microdata format.

<http://schema.org/docs/gs.html>
~~<http://data-vocabulary.org/>~~



Schema.org
(often called **Schema**) is a semantic
vocabulary of tags (or **microdata**) that
you can add to your HTML to improve the
way search engines read and represent
your page in **SERPs**



```
<div itemprop="address" itemscope
  itemtype="http://schema.org/PostalAddress">
  <span itemprop="streetAddress">
    km 28 Cairo-Alexandria Desert Road
  </span>,
  <span itemprop="addressLocality">Smart Village</span>,
  <span itemprop="addressRegion">Giza</span>,
  <span itemprop="addressCountry">Egypt</span>
  <span itemprop="postalCode">11371</span>
</div>
```



Developers can test pages
containing Microdata using any
Structured Data Testing Tool

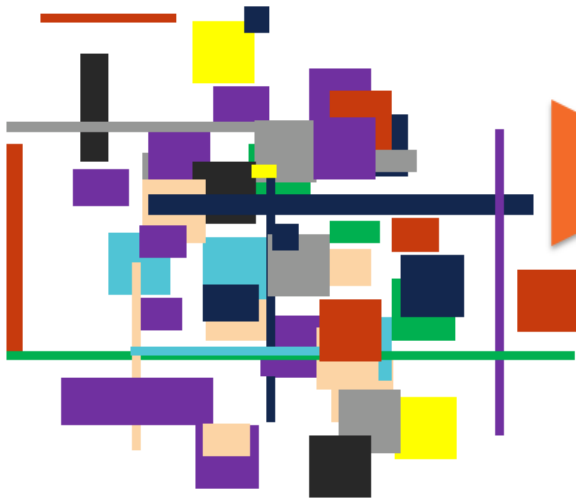
<http://linter.structured-data.org/>

Chrome Structured Data Tool Extension

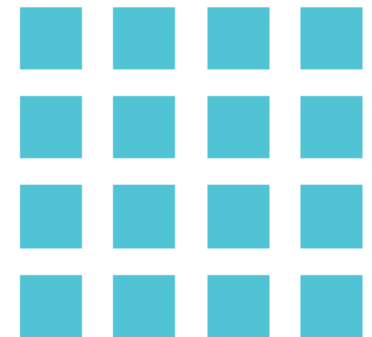
<https://search.google.com/structured-data/testing-tool>

JSON-LD

Unstructured Data



Structured Data



“Data is messy and disconnected.
JSON-LD organizes and connects it,
creating a better Web.”

<https://moz.com/blog/json-ld-for-beginners>

Facebook – log in or sign up

https://www.facebook.com

facebook

Email or Phone Password Log In

Forgotten account?

Facebook helps you connect and share with the people in your life.

Create an account

It's free and always will be.

First name Surname

Mobile number or email address

Elements Console Sources Network Performance Memory Application Security

```
<meta property="og:locale" content="en GB">
<script type="application/ld+json">
  {"@context":"http://schema.org","@type":"WebSite","name":"Facebook","
  url":"https://www.facebook.com/"}
</script>
<link rel="search" type="application/opensearchdescription+xml" href="/osd.xml"
title="Facebook">
<link rel="canonical" href="https://www.facebook.com/">
<link rel="alternate" media="only screen and (max-width: 640px)" href="https://
m.facebook.com/">
<link rel="alternate" media="handheld" href="https://m.facebook.com/">
<link rel="alternate" hreflang="x-default" href="https://www.facebook.com/">
<link rel="alternate" hreflang="en" href="https://www.facebook.com/">
```

html#facebook.tinyViewport.tinyHeight head meta

Styles Computed

Filter :hov .cls +

element.style {

me user agent stylesheet

ta

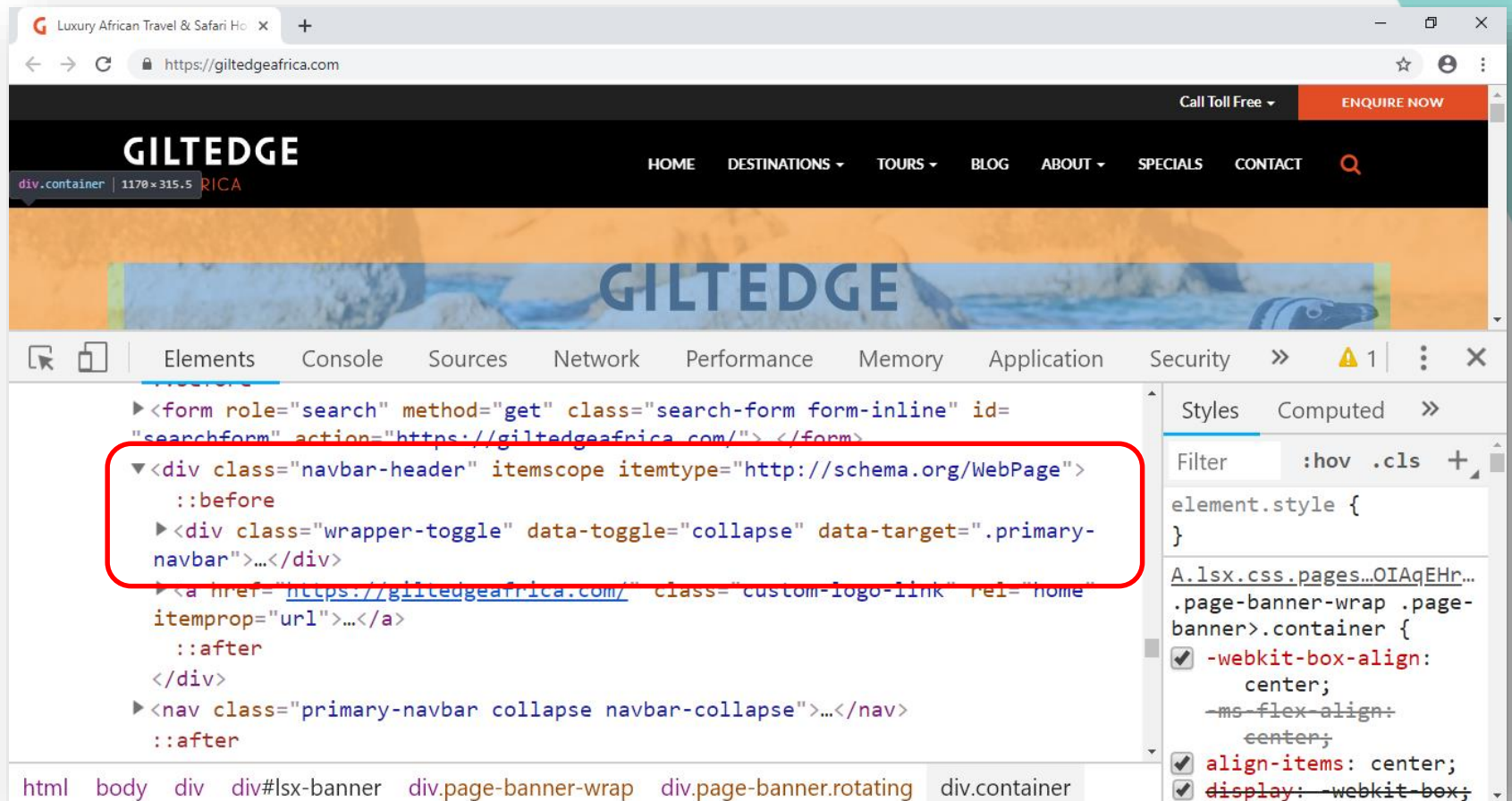
{

display: none;

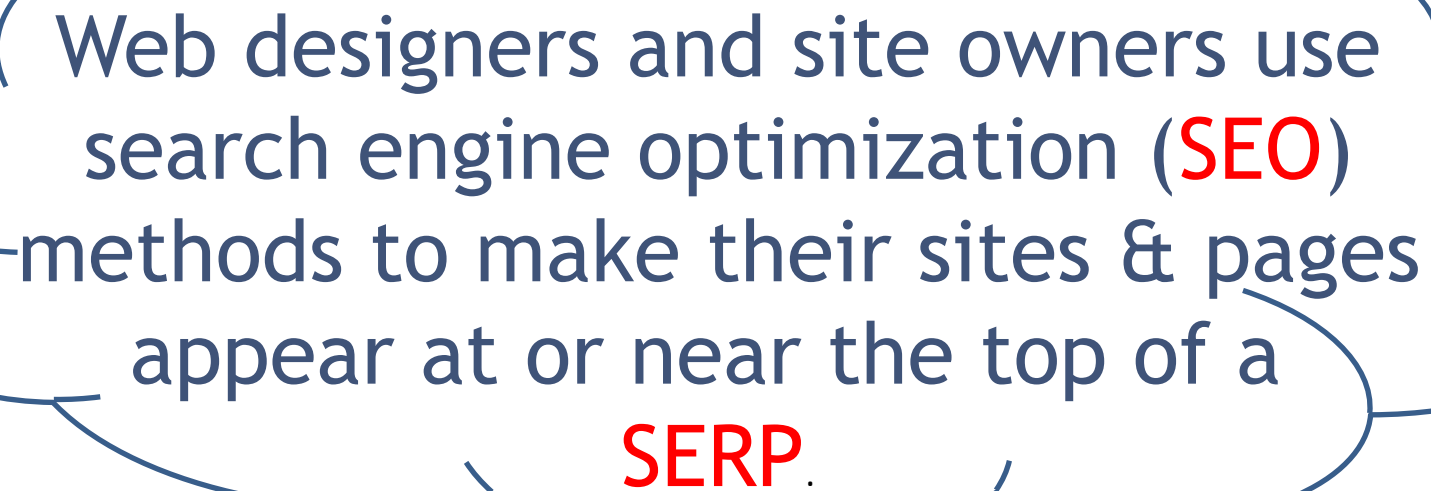
}

margin -

<https://w3techs.com/technologies/details/da-jsonld/all/all>



<https://github.com/lawrencewoodman/mida/wiki/Sites-Using-Microdata>



Web designers and site owners use search engine optimization (**SEO**) methods to make their sites & pages appear at or near the top of a **SERP**.

Assignment