

HTML5 & CSS3

A chance to Do things Differently

Eng. Niveen Nasr El-Den SD &

Gaming CoE

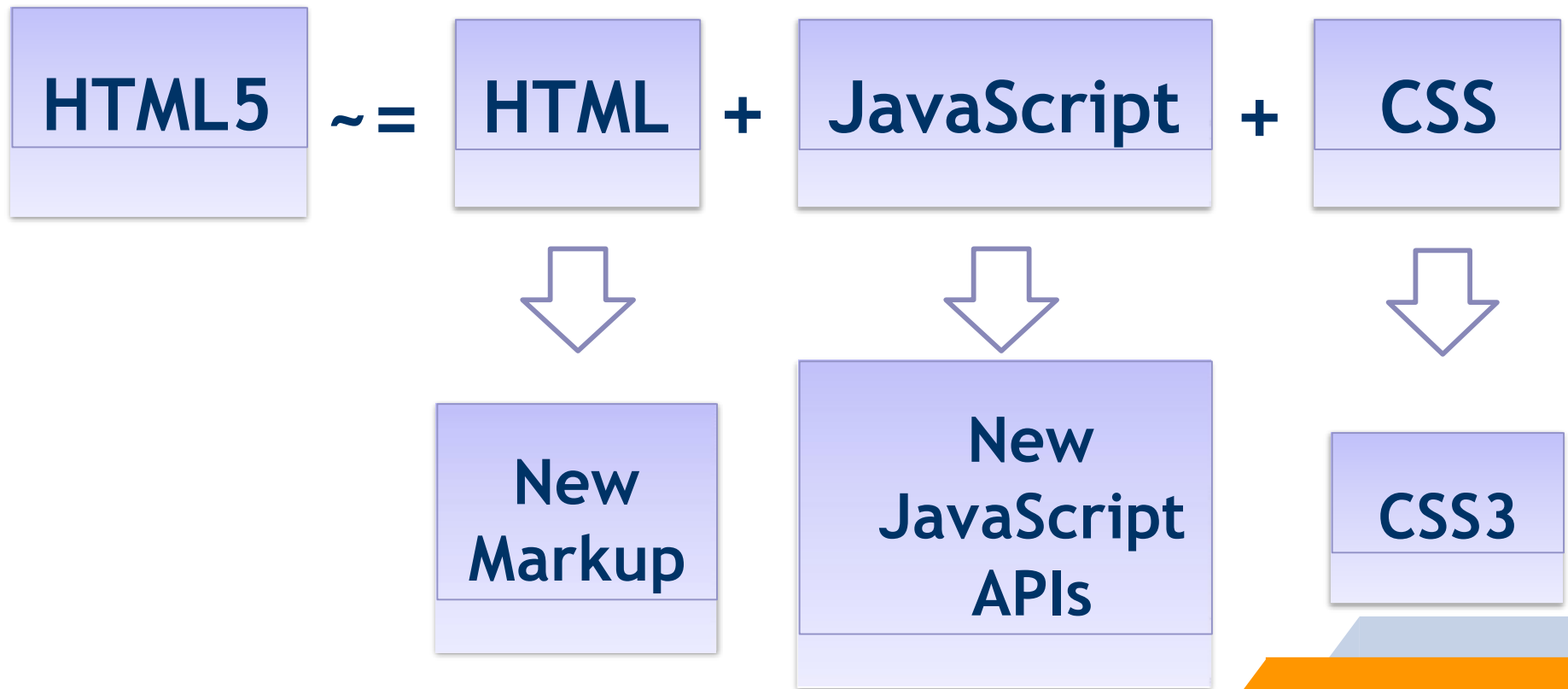
iSi

The background features a large, dark blue trapezoidal shape on the left side. To its right, there are several overlapping geometric shapes in light blue and white, creating a layered effect. At the bottom, there are two horizontal orange bars with 3D effects, one above the other, extending from the center towards the right edge.

Day 1


What is HTML5

HTML5 is a Constellation of technology




H5ML5

A Record of Tomorrow



HTML5 includes new elements for
better structure,
better form handling,
drawing, and
media content



Overview of Enhancements

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

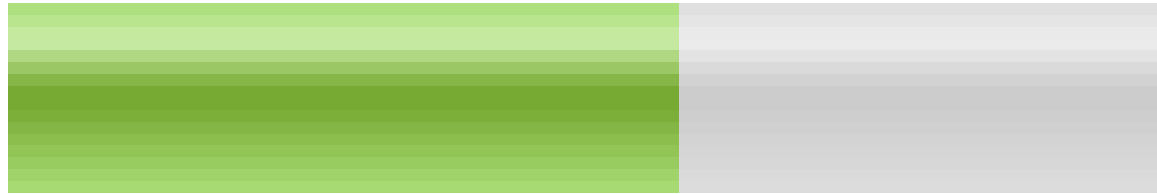
New Elements

The background features a large, dark blue trapezoidal shape on the left side, which tapers towards the right. Above this shape is a light blue area, and to the right is a white area. In the bottom right corner, there are several overlapping geometric shapes in light blue and orange, creating a layered effect.

<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>
```

<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">  
</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.
```

- The highlighted part below is where the error lies:

```
<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.



Forms

Other Form's Controls

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

▷	color		
▷	date	▷	email
▷	datetime	▷	number
▷	datetime-local	▷	range
▷	time	▷	search
▷	month	▷	tel
▷	week	▷	url

Other Form's Controls

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

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="">`

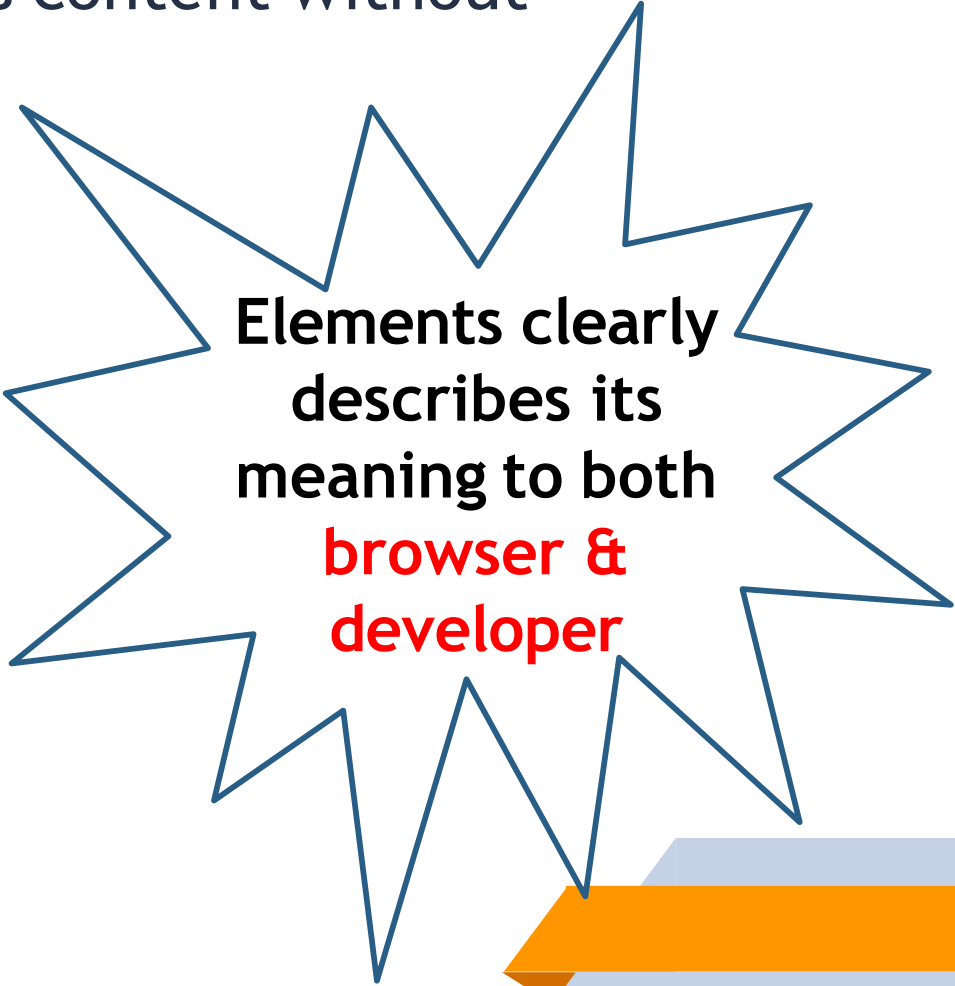
- ▷ Placeholder: add a hint inside the text-field, but where the automatically disappears when clicking inside it.

Semantics & Structure

The background features abstract geometric shapes. A large, light blue triangle points downwards from the top right. A dark blue shape, resembling a wide arrow pointing right, is positioned in the lower-left to center area. In the bottom right corner, there are several overlapping horizontal bars in light blue and orange.

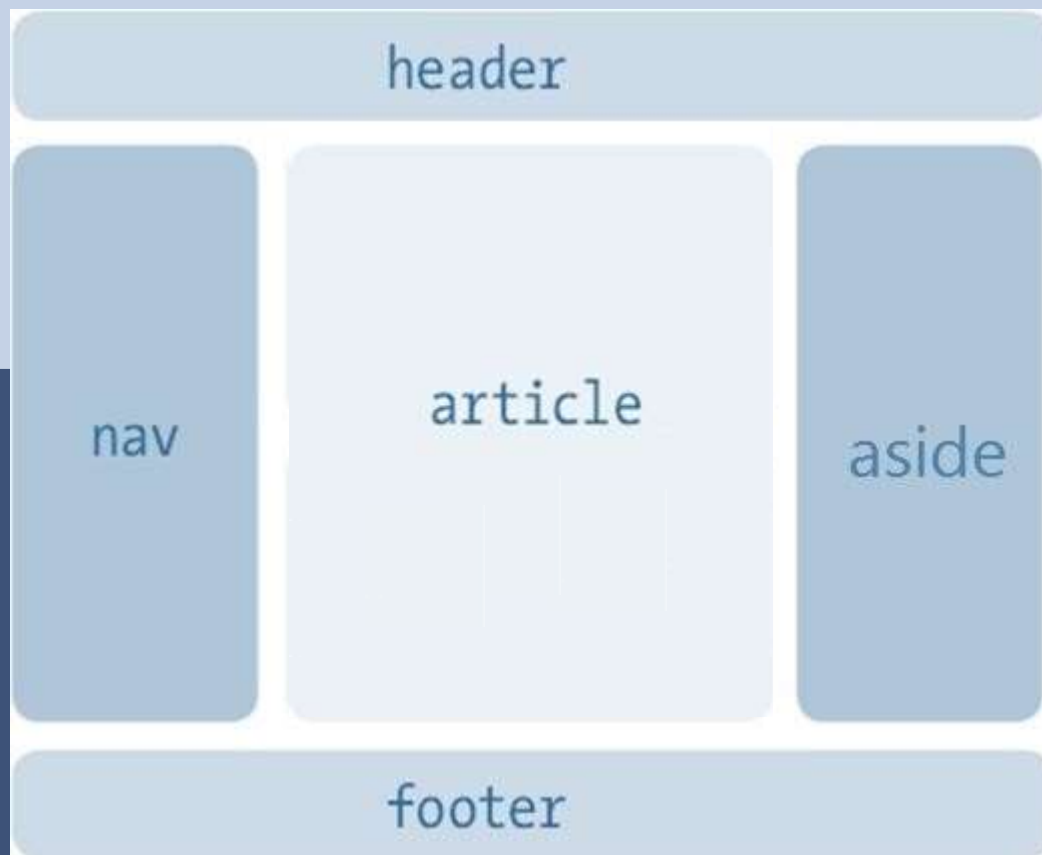
New Semantic Elements

- Markups that describe its content without presenting it e.g.
 - ▷ <header>
 - ▷ <hgroup>
 - ▷ <nav>
 - ▷ <section>
 - ▷ <article>
 - ▷ <aside>
 - ▷ <footer>
 - ▷ etc.



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

Structure



Structured Data

- 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.

■ Supported formats

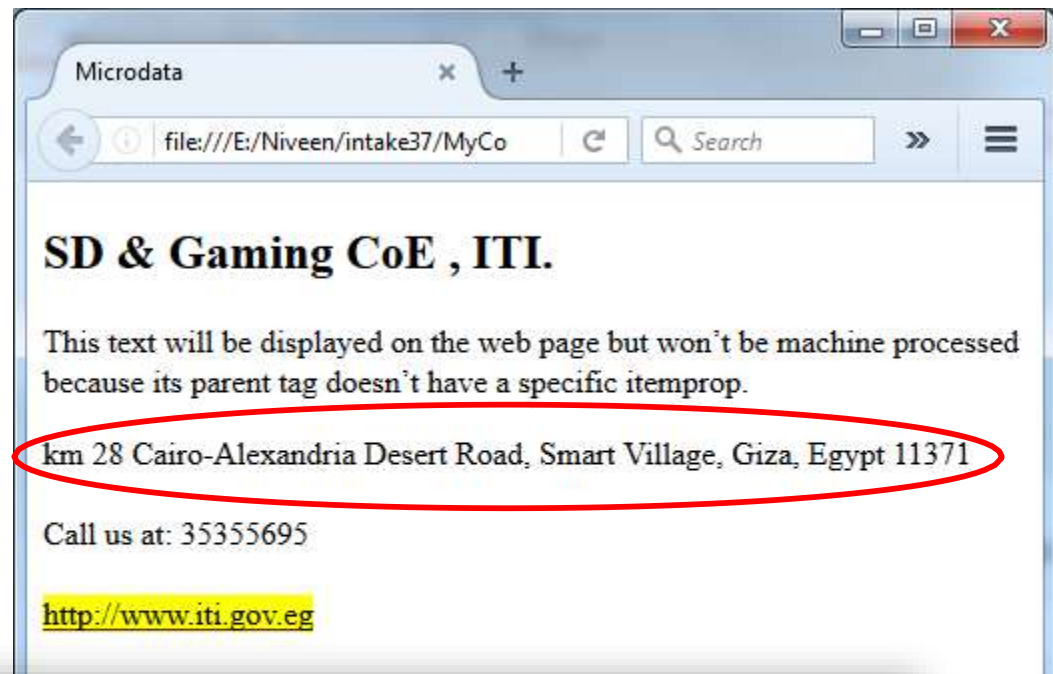
- ▷ JSON-LD
 - <https://json-ld.org/>
- ▷ Microdata

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

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/>



```
<div itemprop="address" itemscope
  itemType="http://data-vocabulary.org/Address">
  <span itemprop="street-address">
    km 28 Cairo-Alexandria Desert Road
  </span>,
  <span itemprop="locality">Smart Village</span>,
  <span itemprop="region">Giza</span>,
  <span itemprop="country-name">Egypt</span>
  <span itemprop="postal-code">11371</span>
</div>
```

Structured Data Testing

← → ↻

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

☆ ⓘ ⋮

Apps Mobile emulator and Getting Visual Studio 17 Great Things To Do Yoga for Weight-Loss » Other bookmarks

Google Structured Data Testing Tool

Sign in

NEW TEST ⚙

1 <div itemprop="address" itemscope

2 itemtype="http://data-vocabulary.org/Address">

3

4 km 28 Cairo-Alexandria Desert Road

5 ,

6 Smart Village,

7 Giza,

8 Egypt

9 11371

10 </div>

Address

All (1) ▾

Address 0 ERRORS 0 WARNINGS ⌵

@type	Address
street-address	km 28 Cairo-Alexandria Desert Road
locality	Smart Village
region	Giza
country-name	Egypt
postal-code	11371

Google Structured Data Testing Tool

NEW TEST

```
1 <div itemprop="address" itemscope
2   <itemtype="http://schema.org/PostalAddress">
3   <span itemprop="streetAddress">
4     km 28 Cairo-Alexandria Desert Road
5   </span>,
6   <span itemprop="addressLocality">Smart Village</span>,
7   <span itemprop="addressRegion">Giza</span>,
8   <span itemprop="addressCountry">Egypt</span>
9   <span itemprop="postalCode">11371</span>
10 </div>
```

PostalAddress

All (1) ▾

PostalAddress

0 ERRORS

0 WARNINGS



@type	PostalAddress
streetAddress	km 28 Cairo-Alexandria Desert Road
addressLocality	Smart Village
addressRegion	Giza
postalCode	11371
addressCountry	
@type	Country
name	Egypt

Demo

Developers can test pages containing Microdata using
Google's Rich Snippet Testing Tool

HTML 5 Microdata

<http://www.google.com/webmasters/tools/richsnippets>

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).

The background features a light blue gradient with a large, dark blue geometric shape on the left side. A diagonal line separates the light blue area from a white area on the right. In the bottom right corner, there are overlapping orange and light blue geometric shapes.

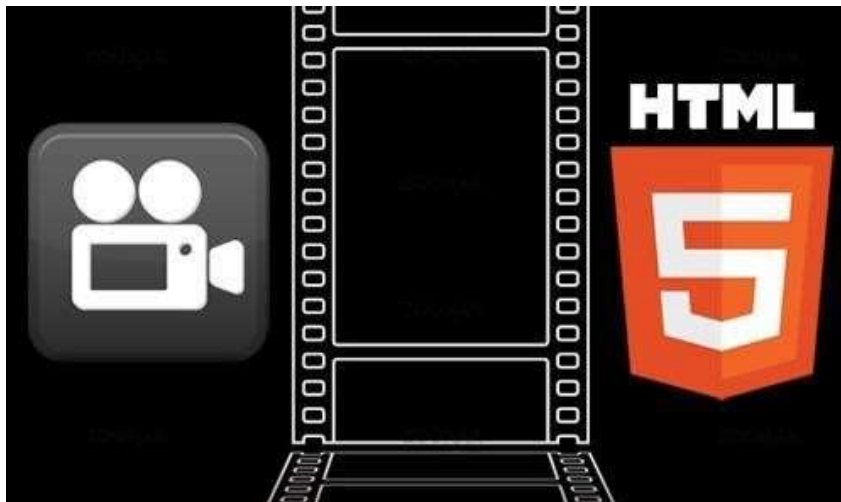
Embedded Content & Multimedia

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
- ▷ ooggVorbis

- 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)		6.0	4.0	5.1	10.5
H.264 (MP4)	9 beta	6.0	4.0	3.1	24 
Ogg Theora		3.0	3.5		10.5
					

Native Audio Browser Support

WAV



3.6

5.1

10.5

MP3

9 beta

6.0

3.6

5.1

10.5

Ogg Vorbis



6.0

3.6



10.5



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.
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 .

Media Methods & Properties

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
- currentTime
- duration
- src
- muted
- volume

The background features abstract geometric shapes. A large, dark blue trapezoidal shape on the left contains the text 'GeoLocation'. To its right is a light blue triangular shape pointing towards the top right. In the bottom right corner, there are two overlapping horizontal bars, one light blue and one orange, both with a 3D effect.

GeoLocation

Geolocation

- The Geolocation API is one of the most exciting features of the new web standard.
- Geolocation is the art of figuring out where you are in the world and (optionally) sharing that information with people you trust.
- The ability to get device's geographic location.
- It is set to request location once or continually.

Geolocation Facts

- HTML5 uses this API for working with maps.
- It is a new property that is added to the existing DOM browser object **navigator**
- The user must agree to share their location, and can tell the browser to remember his choice.

Geolocation Requesting Pattern

- To get user's current location (**once**)
 - ▷ **navigator.geolocation.getCurrentPosition(x[,y,z])**
 - **x**: is the onSuccess callback function where a **Position** object is passed in as the **only** invocation argument. This Position object contains a **coords** object which, in turn, contains our **latitude** and **longitude**, etc.. values.
 - **y**: is the errorHandler callback function where the object passed to this handler has **code** and **message** properties as follows:
 - ▷ 0: UNKNOWN_ERROR
 - ▷ 1: PERMISSION_DENIED
 - ▷ 2: POSITION_UNAVAILABLE
 - ▷ 3: TIMEOUT
 - **z**: is the options object

Location Option

- **enableHighAccuracy (Boolean)**
 - ▷ Attempt to gather more accurate location coordinates
 - ▷ May not do anything and cause request to take longer
 - ▷ Default **false**
- **timeout (msec)**
 - ▷ Determines max time allowed to calculate location
 - ▷ Default is **no limit**
- **maximumAge (msec)**
 - ▷ Determines how old location value may be before an attempt to refresh coordinates
 - ▷ Default is **0** (immediate recalc)

Example

```
var options = {
  enableHighAccuracy: true, //boolean (default: false)
  timeout: 10000, //00      // in ms (default: no limit)
  maximumAge: 1000        // in ms (default: 0)
};

navigator.geolocation.getCurrentPosition(showPosition, positionError, options);

function showPosition(position) {
  var coords = position.coords;
  console.log(coords.latitude);
  console.log(coords.longitude);
}
```

```
function positionError(e){//error has code and message properties
  switch (e.code) {
    case 0: // e.UNKNOWN_ERROR -->error.UNKNOWN_ERROR
      console.log("The application has encountered an unknown error while trying\
to determine your current location. Details: ")
      console.log(e.message);
      break;
    case 1: // e.PERMISSION_DENIED-->error.PERMISSION_DENIED
      //Permission denied - The user did not allow Geolocation
      console.log("You chose not to allow this application access to your location.");
      break;
    case 2: // e.POSITION_UNAVAILABLE--error.POSITION_UNAVAILABLE
      //Position unavailable - It is not possible to get the current location
      console.log("The application was unable to determine your location.");
      break;
    case 3: // e.TIMEOUT-->error.TIMEOUT
      //Timeout - The operation timed out
      console.log("The request to determine your location has timed out.");
      break;
  }
}
```

Geolocation Requesting Pattern

- To watch location change (**continual**)
 - ▷ navigator.geolocation.**watchPosition**(x[,y,z])
 - gets the user's current position and continually returns updated position.
 - ▷ navigator.geolocation.**clearWatch**()
 - used to stop “watchPosition()” running & execution.

Assignments