# A Brief Overview of HTML

Çağatay Sönmez

01.10.2021

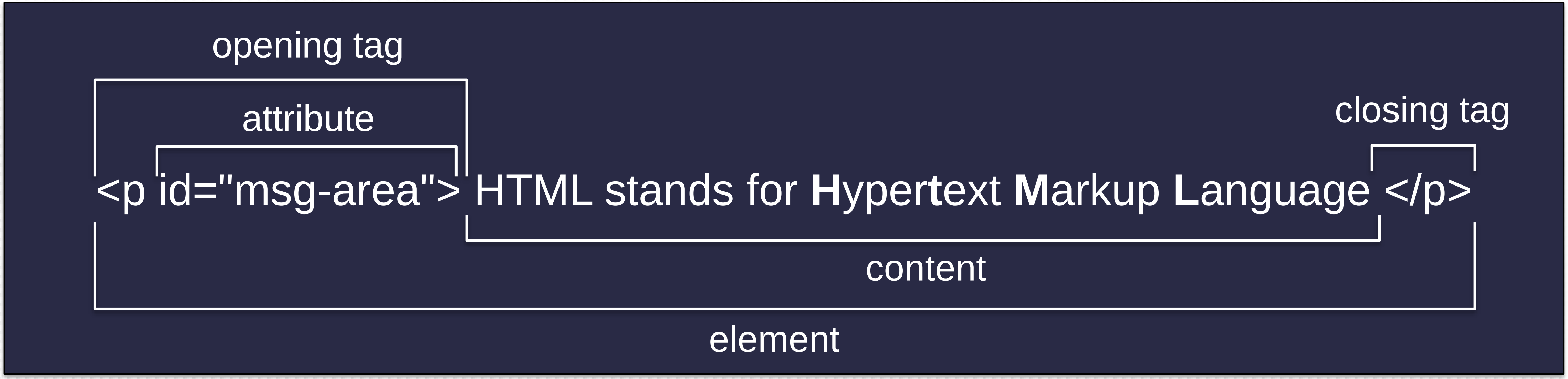
Arcelik Electronics Plant, R&D Team – Software Design

# Agenda

- What is HTML
- HTML5
- New Elements in HTML5
- New Attributes in HTML5
- New Events in HTML5

# What is HTML

- HTML (**H**yper**t**ext **M**arkup **L**anguage) is the standard markup language
- It describes the structure of a web page
- It consists of a series of elements



opening tag

attribute

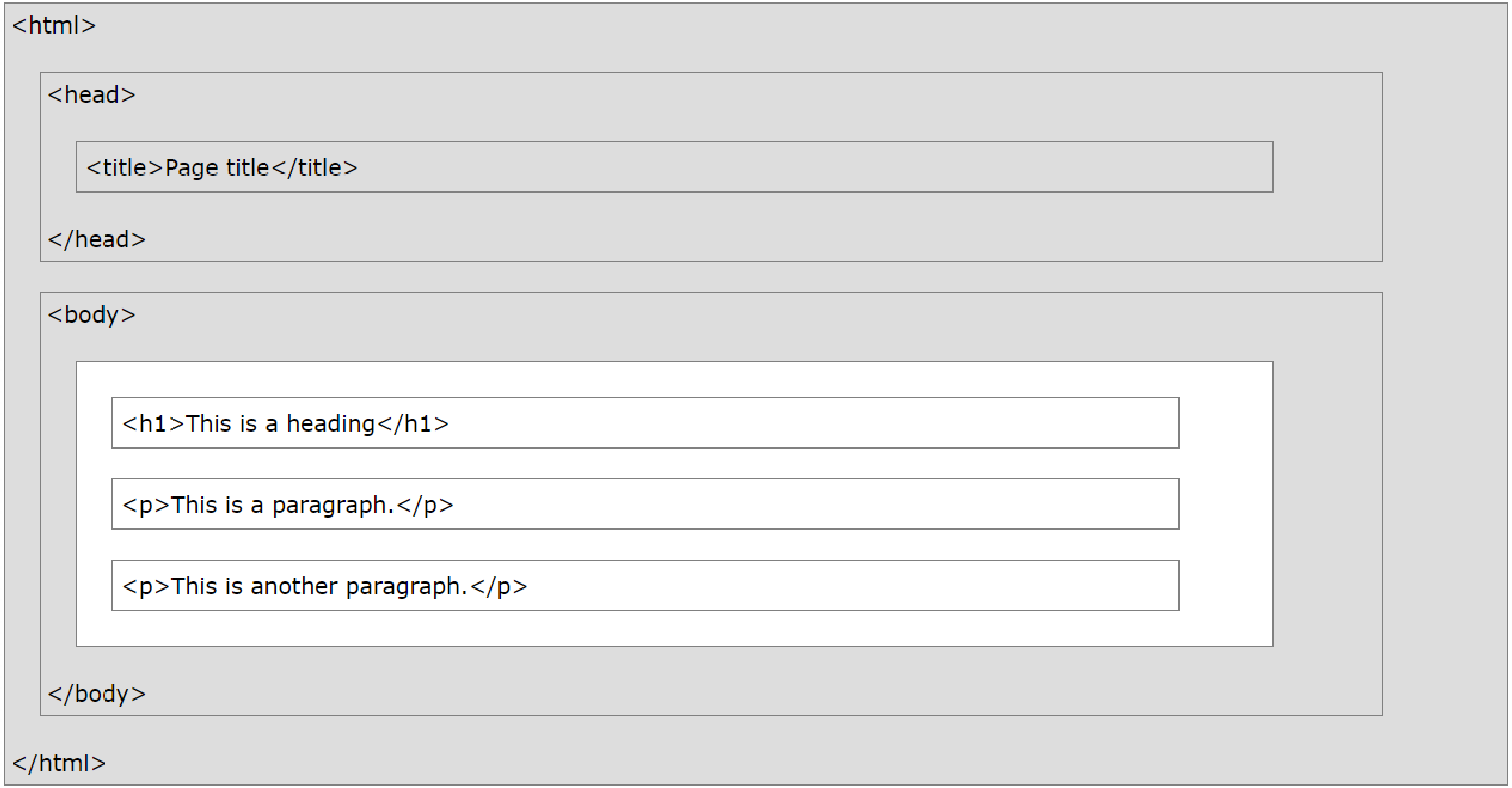
closing tag

<p id="msg-area"> HTML stands for **H**yper**t**ext **M**arkup **L**anguage </p>

content

element

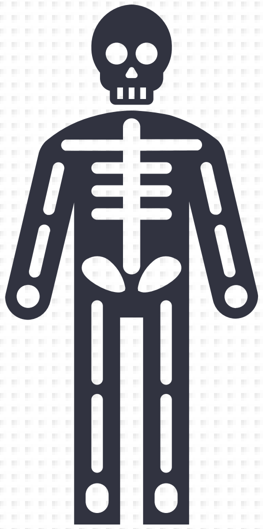
# Example HTML Page Structure



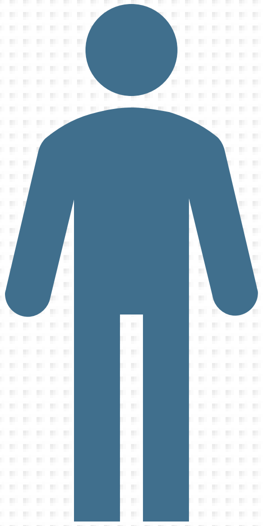
```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
      <h1>This is a heading</h1>
      <p>This is a paragraph.</p>
      <p>This is another paragraph.</p>
  </body>
</html>
```

# HTML & CSS & JavaScript

- HTML is a markup language for creating web pages, CSS beautifies it and JavaScript makes it dynamic



HTML



CSS

JavaScript

# History of Web Technologies

| Timeline | Web Technologies |
|---|---|
| 1991 | HTML 1.0 |
| 1994 | HTML 2.0 |
| 1996 | CSS1 + JavaScript |
| 1997 | HTML 4.01 |
| 1998 | CSS2 |
| 2000 | XHTML1 |
| 2002 | CSS3 |
| 2005 | Ajax |
| 2008 | First draft of HTML5 is released by WHATWG |
| 2014 | Official HTML5 recommendation is released by W3C |
| 2016 | HTML5.1 recommendation is released |
| 2017 | HTML5.2 recommendation is released |

# HTML5

- HTML5 is the next generation of HTML.
- New features are based on HTML, CSS, DOM, and JavaScript.
- In HTML5 there is only one <!doctype> declaration, and it is very simple:
    - <!DOCTYPE html>

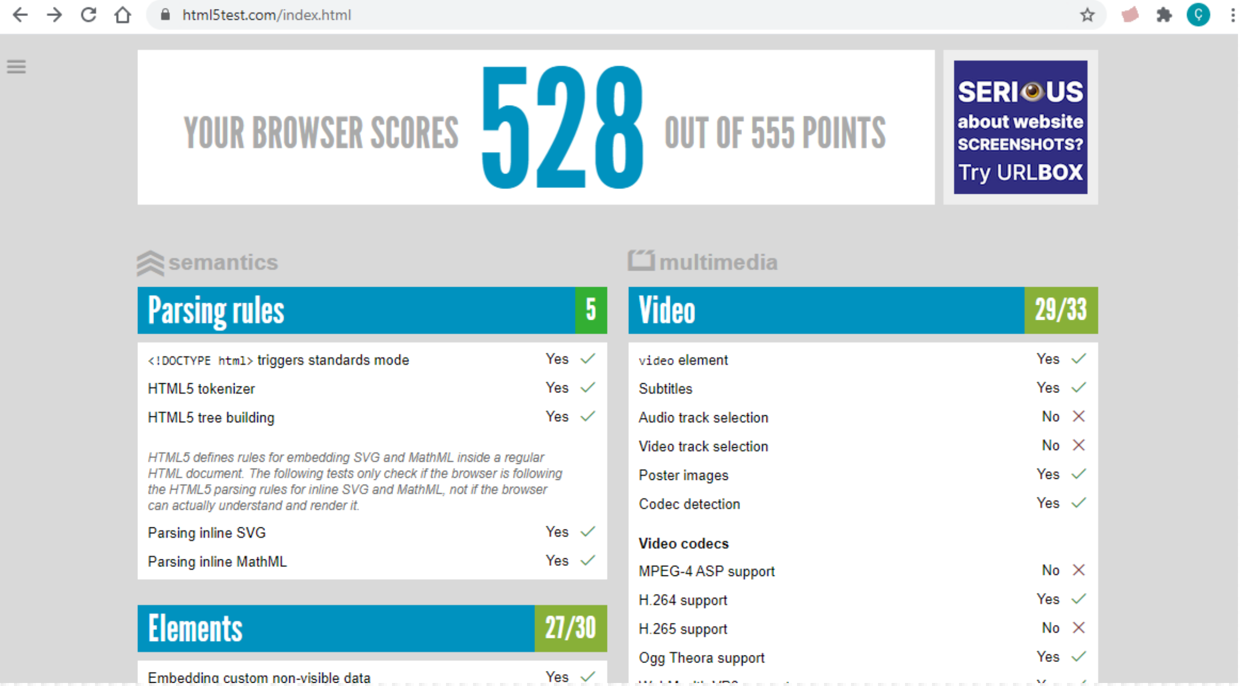
- HTML5 ~= HTML + JS + CSS

# HTML5 is The New HTML Standard

- New Elements
- New Attributes
- Full CSS3 Support
- Video and Audio
- 2D/3D Graphics
- Richer Web Applications with JavaScript API

# HTML5 Support of Browsers

- Every browser has different level of html5 support
- Consider your target browsers during the development phase
- You can visit html5test.com to check how well your browser supports html5
  - [https://html5test.com/index.html](https://html5test.com/index.html)



html5test.com/index.html

YOUR BROWSER SCORES 528 OUT OF 555 POINTS

SERIOUS about website SCREENSHOTS? Try URLBOX

semantics

**Parsing rules** 5

| | |
|---|---|
| `<!DOCTYPE html>` triggers standards mode | Yes ✓ |
| HTML5 tokenizer | Yes ✓ |
| HTML5 tree building | Yes ✓ |

*HTML5 defines rules for embedding SVG and MathML inside a regular HTML document. The following tests only check if the browser is following the HTML5 parsing rules for inline SVG and MathML, not if the browser can actually understand and render it.*

| | |
|---|---|
| Parsing inline SVG | Yes ✓ |
| Parsing inline MathML | Yes ✓ |

**Elements** 27/30

| | |
|---|---|
| Embedding custom non-visible data | Yes ✓ |

multimedia

**Video** 29/33

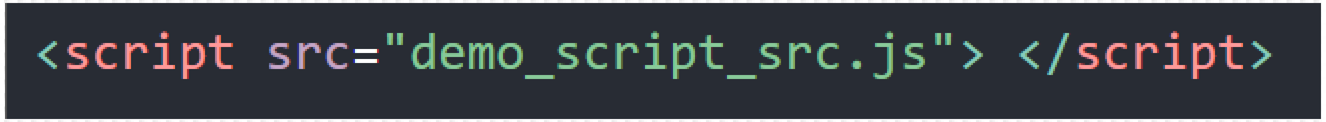
| | |
|---|---|
| `video` element | Yes ✓ |
| Subtitles | Yes ✓ |
| Audio track selection | No ✕ |
| Video track selection | No ✕ |
| Poster images | Yes ✓ |
| Codec detection | Yes ✓ |

**Video codecs**

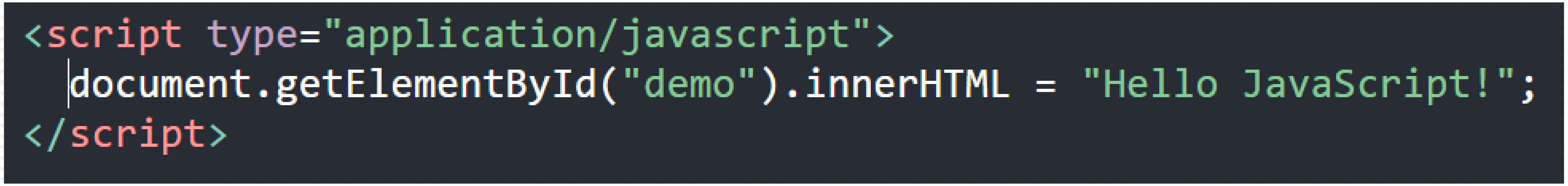
| | |
|---|---|
| MPEG-4 ASP support | No ✕ |
| H.264 support | Yes ✓ |
| H.265 support | No ✕ |
| Ogg Theora support | Yes ✓ |

# Embedding JavaScript to HTML

- The <script> tag is used to embed a client-side script (JavaScript).
- The <script> element either contains scripting statements, or it points to an external script file through the src attribute.
- Common uses for JavaScript are image manipulation, form validation, and dynamic changes of content.



```html
<script src="demo_script_src.js"> </script>
```

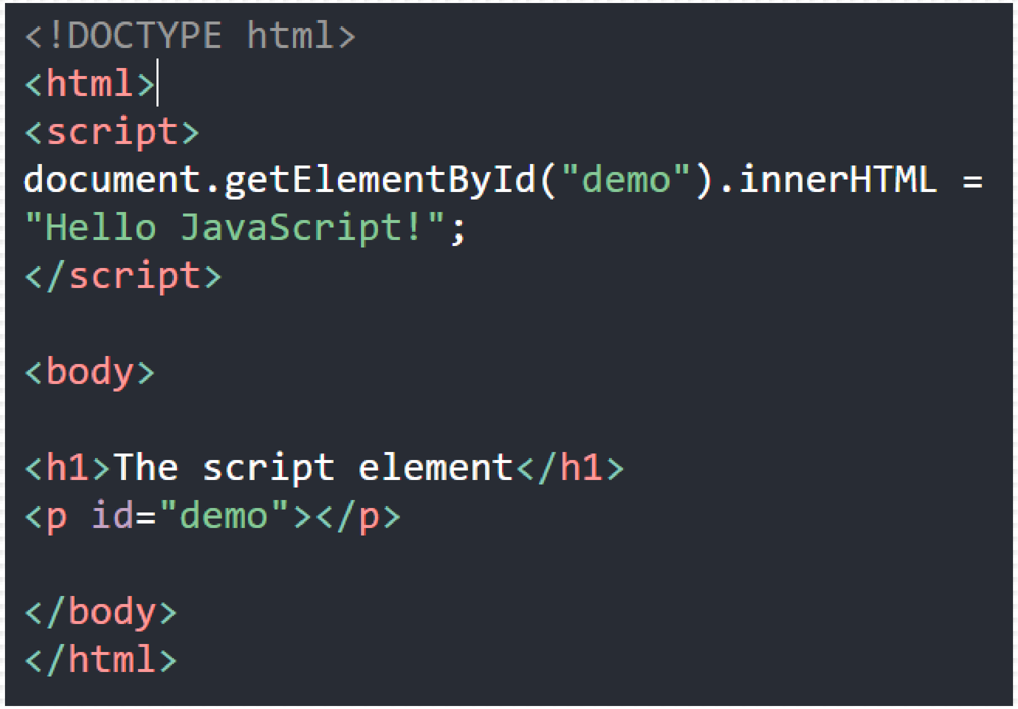


```html
<script type="application/javascript">
  document.getElementById("demo").innerHTML = "Hello JavaScript!";
</script>
```

# JavaScript Execution

- The JavaScript is interpreted when the HTML parser finds the script tag.
- Only DOM nodes found before the script tag are valid while JS code is running.

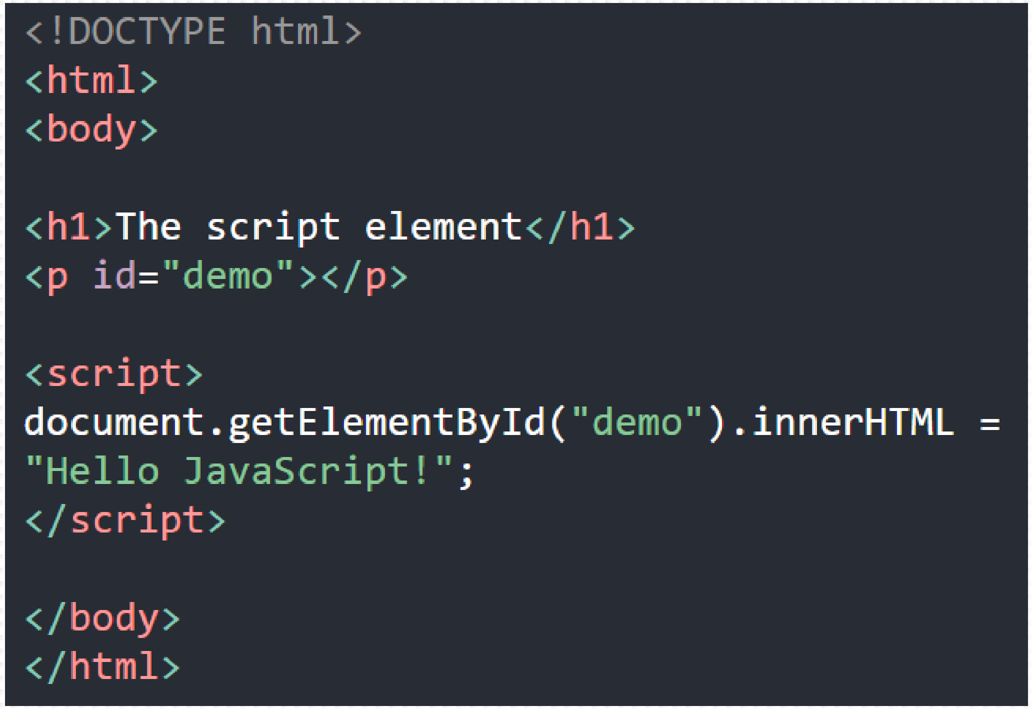
✗



```
<!DOCTYPE html>
<html>
<script>
document.getElementById("demo").innerHTML =
"Hello JavaScript!";
</script>

<body>

<h1>The script element</h1>
<p id="demo"></p>

</body>
</html>
```

VS

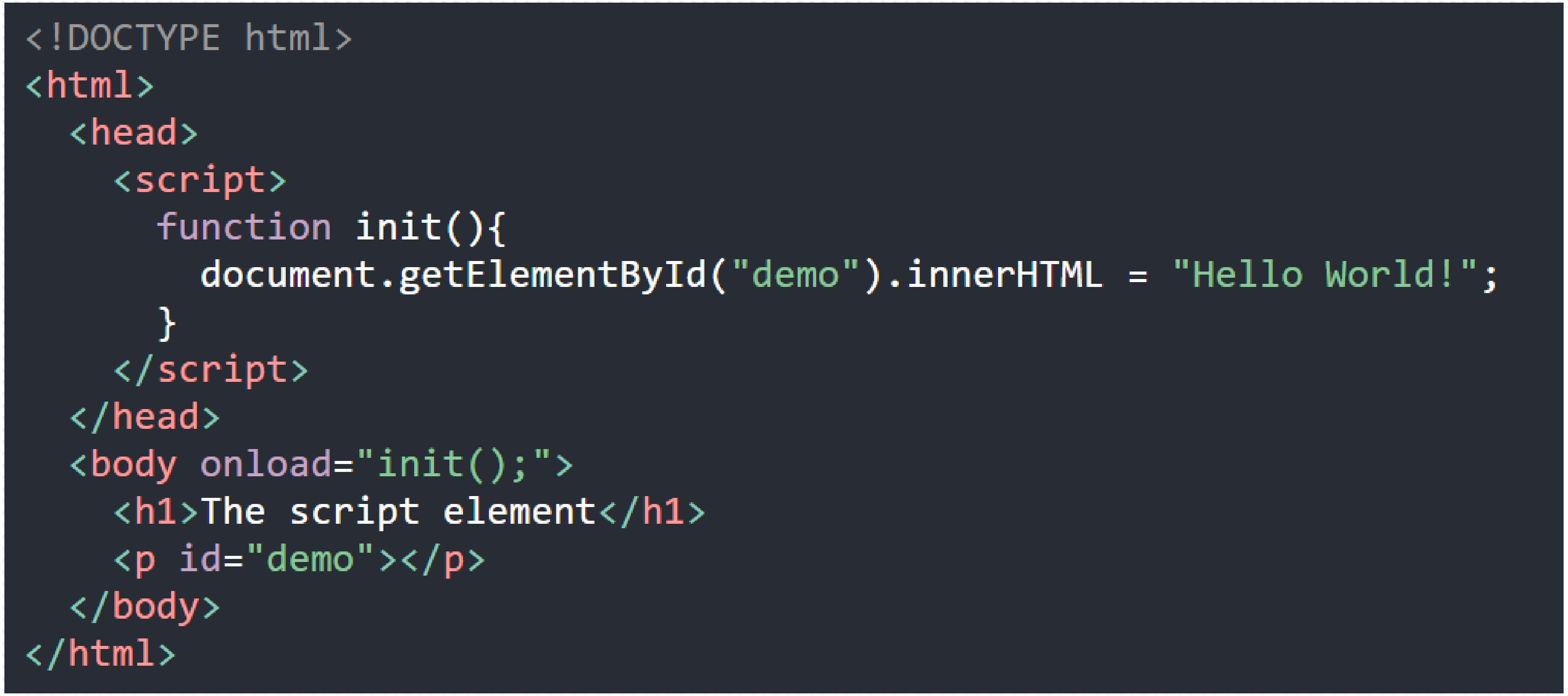
✓



```
<!DOCTYPE html>
<html>
<body>

<h1>The script element</h1>
<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML =
"Hello JavaScript!";
</script>

</body>
</html>
```

# JavaScript Execution cont.

- JavaScript codes are usually run after the DOM content is fully loaded.



```
<!DOCTYPE html>
<html>
  <head>
    <script>
      function init(){
        document.getElementById("demo").innerHTML = "Hello World!";
      }
    </script>
  </head>
  <body onload="init();">
    <h1>The script element</h1>
    <p id="demo"></p>
  </body>
</html>
```

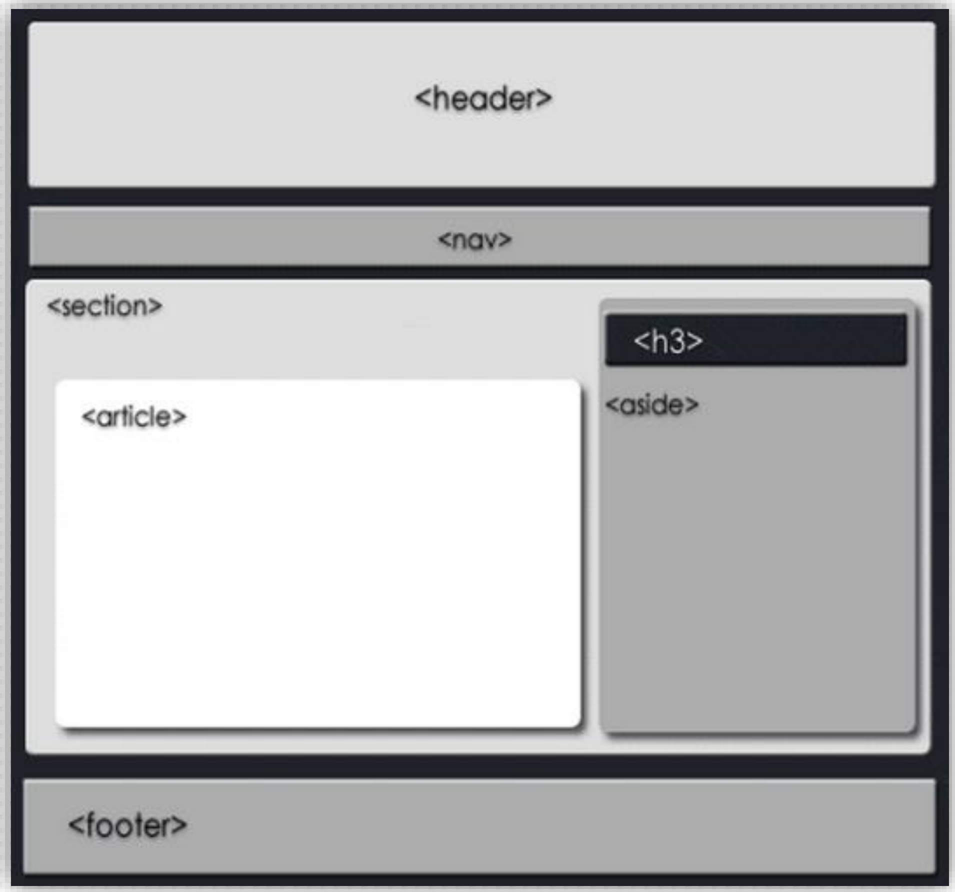
# New Elements & Attributes & Events

# New HTML5 Elements

| | | |
|---|---|---|
| <article> | <meter> | <audio> |
| <aside> | <nav> | <video> |
| <bdi> | <progress> | <source> |
| <command> | <ruby> | <embed> |
| <details> | <rt> | <track> |
| <summary> | <rp> | … |
| <figcaption> | <section> | <datalist> |
| <footer> | <time> | <keygen> |
| <header> | <wbr> | <output> |
| <hgroup> | … | … |
| <mark> | <svg> | <canvas> |

⬜ Semantic/Structural Elements 🟨 Media Elements 🟩 Form Elements

# HTML5 Semantic/Structural Elements



# HTML5 <video> Element

```
<!DOCTYPE html>
<html>
  <body>

     <video width="320" height="240" controls>
       <source src="movie.mp4" type="video/mp4" />
       <source src="movie.ogg" type="video/ogg" />
       Your browser does not support the video tag.
     </video>

  </body>
</html>
```

# HTML5 <audio> Element

```
<!DOCTYPE html>
<html>
    <body>

        <audio controls>
            <source src="sound.mp3" type="audio/mpeg" />
            <source src="sound.ogg" type="audio/ogg" />
            Your browser does not support the audio tag.
        </audio >

    </body>
</html>
```

# HTML5 <canvas> Element

- A canvas is a drawable region defined in HTML code with height and width attributes.
- The <canvas> element is only a container for graphics, you must use a script to draw the graphics.
- Canvas has several methods for drawing paths, boxes, circles, characters, and adding images.

# HTML5 <canvas> Element cont.

```
<!DOCTYPE html>
<html><body>
    <canvas id="myCanvas" width="200" height="100" ></canvas>
    <script type="text/javascript">
        var ctx=document.getElementById("myCanvas").getContext("2d");
        ctx.fillStyle="red";
        ctx.beginPath();
        ctx.arc(70,18,15,0,Math.PI*2,true);
        ctx.closePath();
        ctx.fill();
    </script>
</body></html>
```

See Example

# HTML5 Inline SVG

- SVG stands for Scalable Vector Graphics. SVG is used to define vector-based graphics for the Web.
- SVG defines the graphics in XML format. So SVG graphics do NOT lose any quality if they are zoomed or resized.
- Every element and every attribute in SVG files can be animated.
- SVG is a W3C recommendation.

# HTML5 Inline SVG cont.

```
<!DOCTYPE html>
<html>
 <body>

  <svg xmlns="http://www.w3.org/2000/svg" version="1.1">
   <circle cx="100" cy="50" r="40" stroke=“blue" fill="red"/>
 </svg>

 </body>
</html>
```

See Example

# New HTML5 Global Attributes

| Attribute | Description |
|---|---|
| contenteditable | Specifies whether the content of an element is editable or not. |
| contextmenu | Specifies a context menu for an element. The context menu appears when a user right-clicks on the element. |
| draggable | Specifies whether an element is draggable or not. |
| dropzone | Specifies whether the dragged data is copied, moved, or linked when dropped. |
| hidden | Specifies that an element is not yet, or is no longer, relevant. |
| spellcheck | Specifies whether the element is to have its spelling and grammar checked or not. |

# HTML5 contenteditable & spellcheck

```
<!DOCTYPE html>
<html>
 <body>

 <p contenteditable="true" spellcheck="true">This is editable.</p>
  First name: <input type="text" name="fname" spellcheck="true" />

 </body>
</html>
```

See Example

# HTML5 Drag and Drop

```
<!DOCTYPE HTML>
<html>
    <body>
        <div id="div1" ondrop="drop(event)" ondragover="drop(event)">
        </div>

        <img id="drag1" src="img_logo.gif" draggable="true"
          ondragstart="drag(event)" width="336" height="69" />
    </body>
</html>
```

See Example

# New HTML5 Event Attributes

| | | | |
|---|---|---|---|
| onerror | oncanplay | onvolumechange | ondragstart |
| onoffline | ondurationchange | onprogress | ondrop |
| ononline | onended | ontimeupdate | onmousewheel |
| onmessage | onerror | onwaiting | onscroll |
| onpageshow | onloadeddata | onemptied | … |
| onpagehide | onpause | … | oncontextmenu |
| onresize | onplay | ondrag | onformchange |
| onstorage | onplaying | ondragend | onforminput |
| onredo | onseeked | ondragenter | oninput |
| onunload | onseeking | ondragleave | oninvalid |
| … | onratechange | ondragover | … |

🟦 Window Events 🟩 Media Events 🟨 Mouse Events 🟥 Form Events

# Example: onplay/onpause

```
<!DOCTYPE html>
<html>
    <script>
       function onPlay(){ alert("onplay event is fired"); }
       function onPause(){ alert("onpause event is fired"); }
    </script>
    <body>
       <audio onplay="onPlay();" onpause="onPause();">
          <source src="sound.mp3" type="audio/mpeg" />
       </audio >
    </body>
</html>
```

See Example

# Example: oninput

```
<!DOCTYPE html>
<html>
<body>

    <form oninput="o.value = a.value + b.value">
        <input name="a" type="number" /> +
        <input name="b" type="number" /> =
        <output name="o"></output>
    </form>

</body>
</html>
```

See Example

# Example: online/offline

```
<!DOCTYPE html>
<html><body>
    <script>
        window.addEventListener('online', function(e) {
          alert("You are online...");
        }, false);
        window.addEventListener('offline', function(e) {
          alert("You are offline!!!");
        }, false);
    </script>
</body>
</html>
```

# Example: onerror

```
<!DOCTYPE html>
<html><body>
<script>
    window.onerror = function(msg, url, line) {
            alert("an error has occurred! No need to wrap everything in
            try/catch blocks.");
    };
    EvilFunc();
</script>
</body>
</html>
```

# HTML5 New Input Types

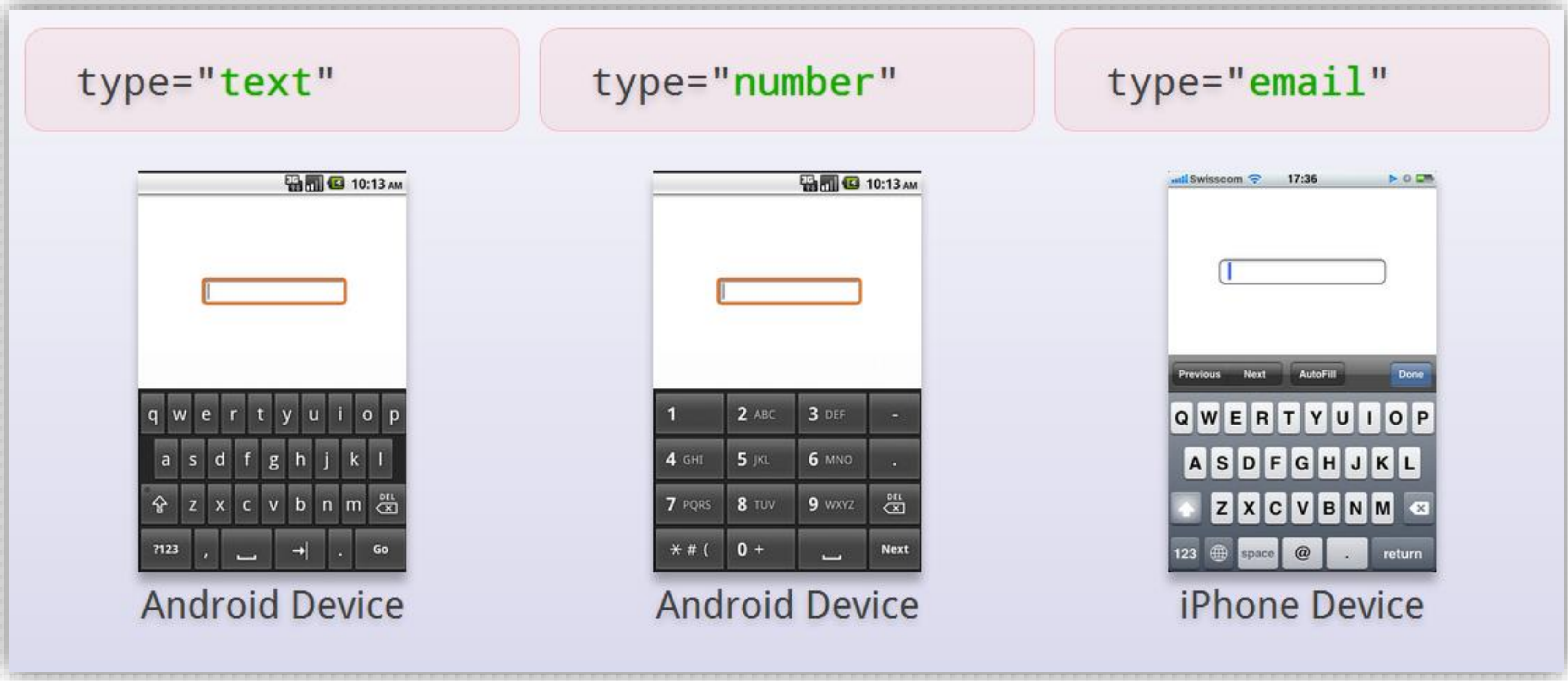
- HTML5 has several new input types for forms.
- These new types allow better input control and validation.

| | | | | |
|---|---|---|---|---|
| color | email | range | datetime-local | time |
| date | month | search | url | |
| datetime | number | Tel | week | |

**Example:**

```
<div>
Quantity : <input type="number" name="quantity" min="1" max="5" />
</div>
```

# HTML5 Input Type Example



type="text"

type="number"

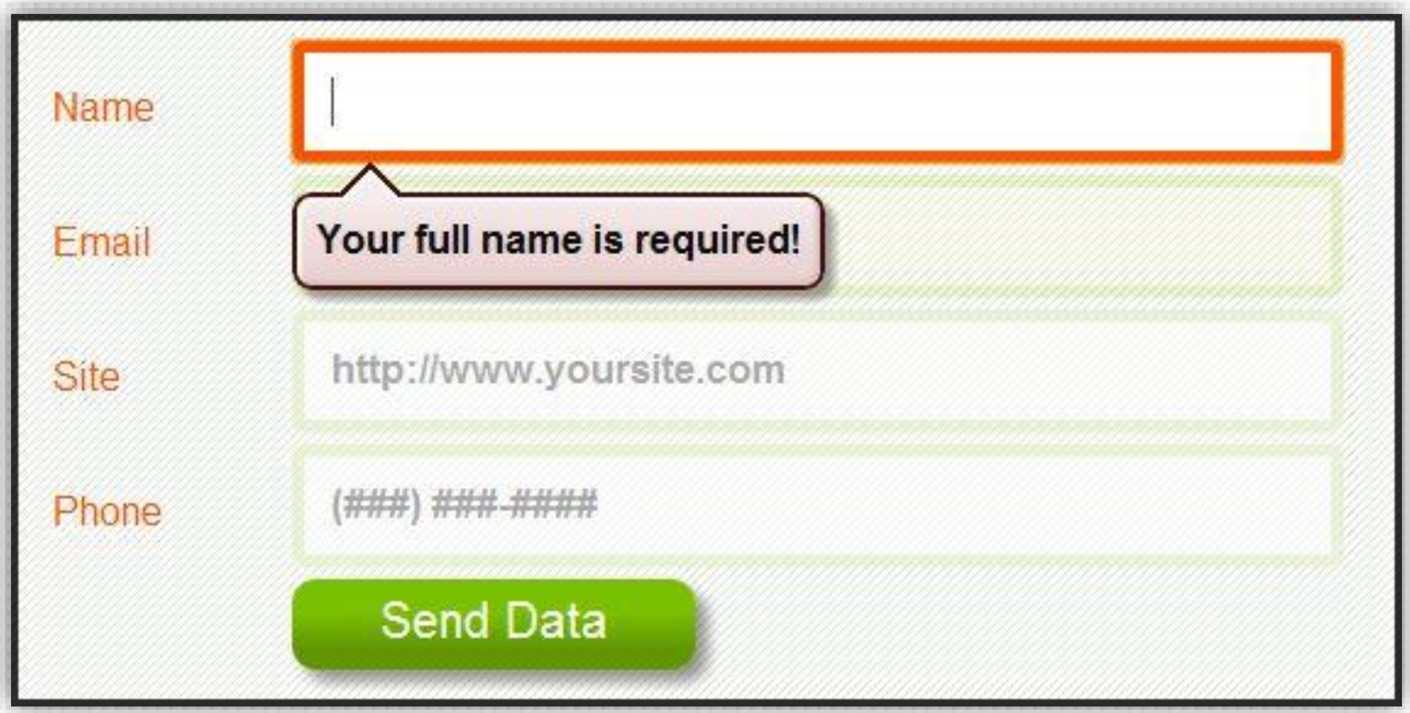
type="email"

Android Device

Android Device

iPhone Device

# HTML5 Form Validation



Your full name is required!

See Example

# QUESTIONS?