HTML Basics

Frontend Junior Program

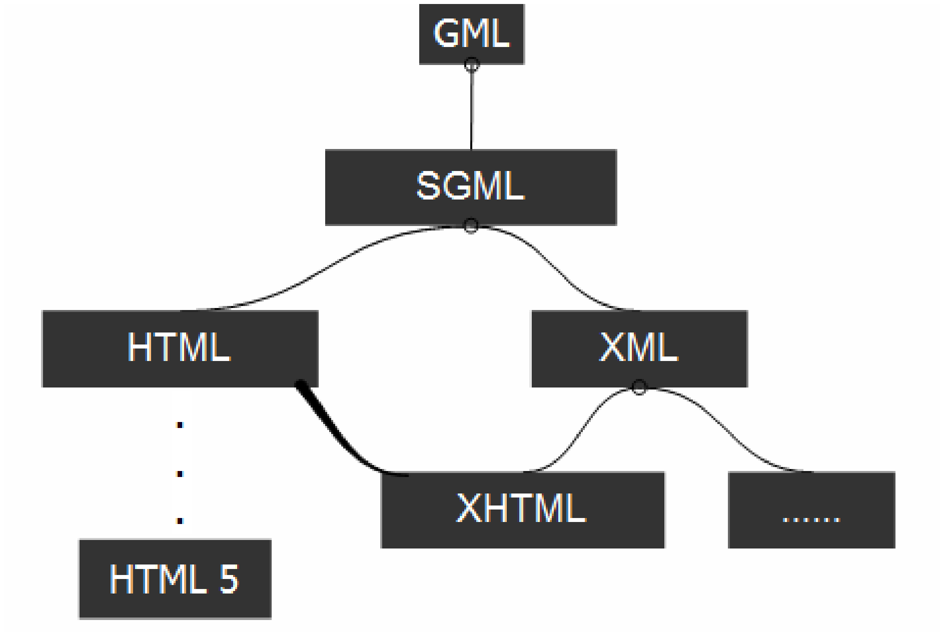
2022

Agenda

1. Markup Languages, Semantic HTML
2. Lists, Links, Media
3. Sectioning
4. Forms, inputs
5. Good to know

MARKUP LANGUAGES

Markup languages



Hypertext Markup Language - Features

- Web page is a document that is suitable for the World Wide Web and the web browser
- Hypertext Markup Language is the standard markup language for documents designed to be displayed in a web browser
- HTML is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>).
- HTML describes the structure of a website semantically along with cues for presentation, making it a markup language rather than a programming language

Main points

- Markup language != programming language
- Hypertext is a text which is not constrained to be linear.
- Hypertext is a text which contains links to other texts. The term was coined by Ted Nelson around 1965.
- Hypermedia is a term used for hypertext which is not constrained to be text: it can include graphics, video and sound, for example. Apparently, Ted Nelson was the first to use this term too.
- Hypertext and Hypermedia are concepts, not products.

Webpage example

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <link rel="stylesheet" href="css/style.css">
</head>
<body>
    ...
    <script src="scripts/script.js"></script>
</body>
</html>
```

<!DOCTYPE>

Element <!DOCTYPE> is intended to indicate the type of the document - DTD (Document Type Definition), for the browser to know how to parse the page.

There are several versions HTML / XHTML (eXtensible HyperText Markup Language), as well as HTML5, which differ in syntax.

The <!DOCTYPE> declaration is not an HTML tag; it is an instruction to the web browser about what version of HTML the page is written in.

Tags

- Opening and closing tags:

```
<tag>Some text</tag>
```

- Single (empty) tag:

```
<tag />
```

- Parent and child tags:

```
<first-tag>
    <second-tag>Some text</second-tag>
</first-tag>
```

- Comment:

```
<!-- Comment content -->
```

Attributes

Example:

```
<tag attribute1="value" attribute2="value"> ... </tag>
```

The order of the attributes doesn't matter.

Examples: `id, class, name, title, style, src, type`

Special Characters and Signs

- Many mathematical, technical, and currency symbols are not presented on a normal keyboard. To add these symbols to an HTML page, you can use an HTML entity name.
- If no entity name exists, you can use an entity number; a decimal (or hexadecimal) reference.

```
<p>Euro symbol: &euro; &#8364; &#x20AC;</p>
<p>&copy; &trade; &laquo; &raquo; &amp; &nbsp; &lt; &gt; &ndash; &mdash;</p>
```

Euro symbol: € € €

© ™ « » & < > – —

Web page structure

<html>
This is a container that includes the entire page content.
There is always two sections on the page: head and body.

<head>
Contains information about the HTML file:
the name of the page, style, meta tags and additional information.

<meta>
Meta tags that contain information for browsing and retrieval systems (charset, description, keywords) for SEO, OpenGraph, etc.

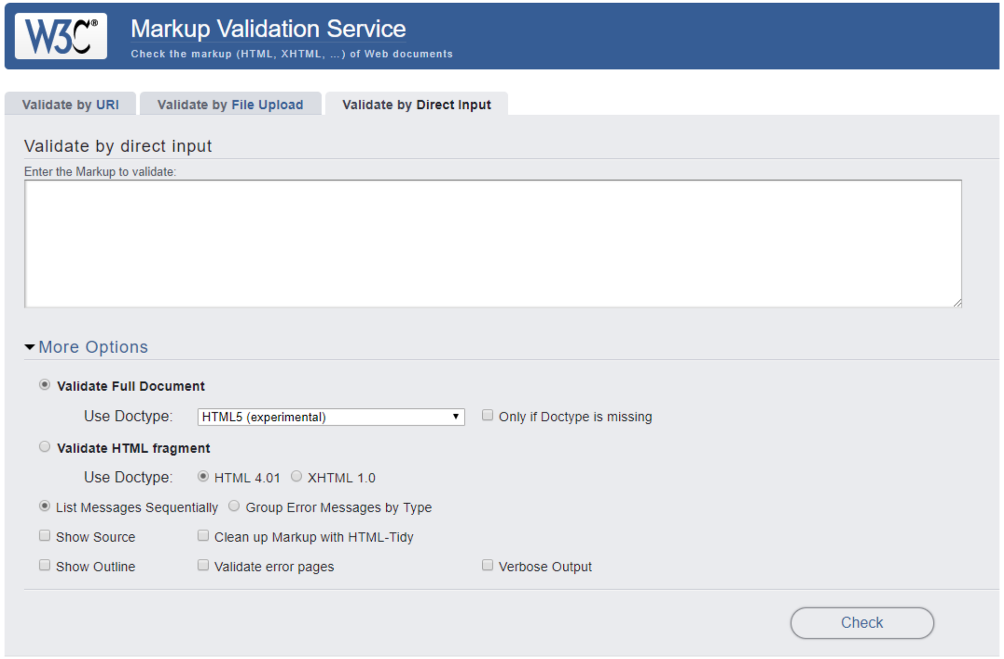
<body>
Contains all the text and tags that are displayed on the page.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

Validation

Validator checks the markup validity of Web documents in HTML validator.w3.org



Markup Validation Service
Check the markup (HTML, XHTML, ...) of Web documents

Validate by URI | Validate by File Upload | Validate by Direct Input

Validate by direct input

Enter the Markup to validate:

More Options

[● Validate Full Document]

Use Doctype: HTML5 (experimental) ☐ Only if Doctype is missing

○ Validate HTML fragment

Use Doctype: [● HTML 4.01] ○ XHTML 1.0

[● List Messages Sequentially] ○ Group Error Messages by Type

☐ Show Source ☐ Clean up Markup with HTML-Tidy

☐ Show Outline ☐ Validate error pages ☐ Verbose Output

Check

Elements of the page

Headings:
<h1>, <h2>, <h3>, <h4>, <h5>, <h6>

Some block elements:
<p>, <pre>, <address>, <blockquote>

Some inline elements:
<sub>, <sup>, <del>, <ins>, <code>, <q>, <cite>,
<big>, <small>, <b>, <strong>, <i>, <em>, <br>, <hr>

https://html.spec.whatwg.org/multipage/
https://developer.mozilla.org/en-US/docs/Web/HTML

Elements of the page - examples

Images

```
<img src="img/picture.png" alt="Alt text" />
```

Links

```
<a href="http://www.epam.com/">Go to Epam</a>
```

Demo:

Go to Epam

Image inside a link

```
<a href="http://www.epam.com/">
    <img src="img/picture.png" />
</a>
```

Demo:

SEMANTIC HTML

Semantic Tags

<article>	<mark>
<section>	<audio>
<aside>	<video>
<nav>	<source>
<figure>	<canvas>
<figcaption>	<svg>
<header>	<datalist>
<footer>	<progress>

Note: the list is not complete.
There are around a hundred semantic elements used in html5.
https://developer.mozilla.org/en-US/docs/Web/HTML/Element

Elements with no semantic meaning

<div> - block element
<span> - inline element

Div is a block-level element that creates a line break to make separate containers or boxes within a page or document, hence it is an abbreviation for 'division', whereas span is a generic container for inline elements and content that allow us to apply styles and other attributes to the content within the span element.

LISTS

Lists

Ordered list:

```
<ol>
    <li>Item first</li>
    <li>Item second</li>
</ol>
```

Demo:

1. Item first
2. Item second

Unordered list:

```
<ul>
    <li>Item 1</li>
    <li>Item 2
        <ul>
            <li>Item 2.1</li>
            <li>Item 2.2</li>
        </ul>
    </li>
</ul>
```

Demo:

Item 1

Item 2

Item 2.1

Item 2.2

Ordered list – attributes

reversed - this Boolean attribute specifies that the items of the list are specified in reversed order.
start - this integer attribute specifies the start value for numbering the individual list items.
type - indicates the numbering type.

```
<ol type="I" start="20" reversed>
    <li>first item</li>
    <li>second item</li>
    <li>third item</li>
</ol>
```

Demo:

XX. first item
XIX. second item
XVIII. third item

`<ol> or <ol type=“1”>`	indicates numbers (default). (1,2,3,...)
`<ol type=“A”>`	indicates uppercase letters (A,B,C,...)
`<ol type=“a”>`	indicates lowercase letters (a,b,c,...)
`<ol type=“I”>`	indicates uppercase Roman numerals (I,II,III,...)
`<ol type=“i”>`	indicates lowercase Roman numerals (i,ii,iii,...)

Lists – Description list

```
<dl>
    <dt>Title 1</dt>
    <dd>Definition</dd>
    <dt>Title 2</dt>
    <dd>Definition</dd>
    <dd>Definition</dd>
</dl>
```

Demo:

Title1
    Definition
Title2
    Definition
    Definition

LINKS

Uniform Resource Locator (URL)

Relative Paths

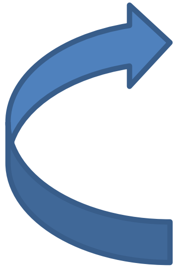
```
index.html
/graphics/image.png
/help/articles/how-do-i-set-up-a-webpage.html
```

Absolute Paths

```
https://www.mysite.com
https://www.mysite.com/graphics/image.png
https://www.mysite.com/help/articles/how-do-i-set-up-a-webpage.html
```

Anchor

A hash mark (**#**), specifies an internal target location (an ID of an HTML element) within the current document. Hyperlinks are not restricted to Web (HTTP)-based documents but can use any protocol supported by the browser.

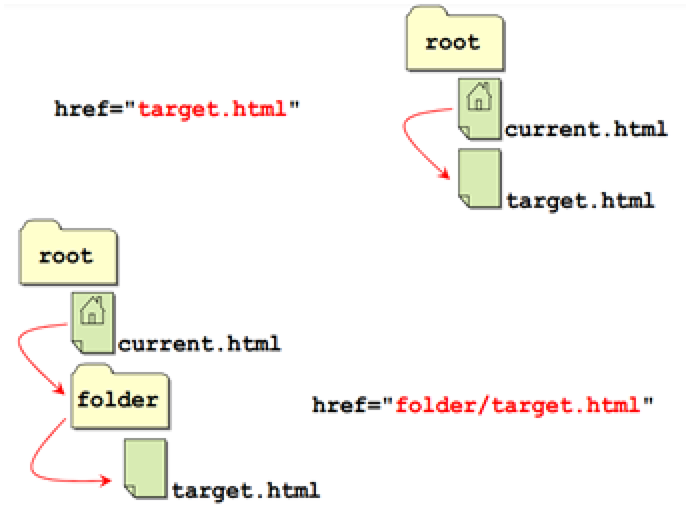


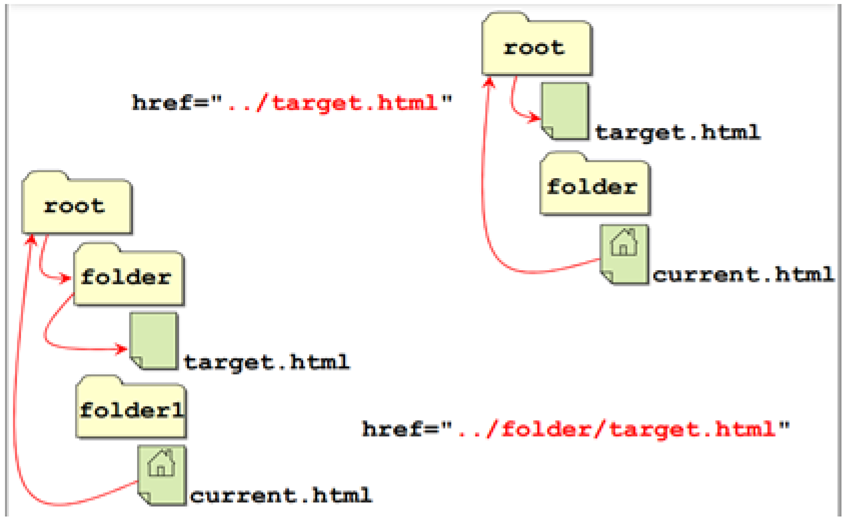
```
<div id="top"></div>



<a href=”#top”>To the top</a>
```

Relative links





Email link, Skype link, Phone link

```html
<a href="tel:+04951234567">+0 (495) 123-45-67</a>
<a href="mailto:example@mail.com">example@mail.com</a>
<a href="skype:someskype?call">someskype</a>
```

See more at Web based protocol handlers.

<a> attributes

href

is a mandatory attribute and it must be a valid URL.

target

if present, must be a valid browsing context name or keyword.
It gives the name of the browsing context that will be used.

type

if present, gives the MIME type of the linked resource. The value must be a valid mime type.

download

if present, indicates that the author intends the hyperlink to be used for downloading a resource.

Attribute ”target”

The target attribute specifies where to open the linked document

`_blank`	Opens the linked document in a new window or tab
`_self`	Opens the linked document in the same frame as it was clicked (this is default)
`_parent`	Opens the linked document in the parent frame
`_top`	Opens the linked document in the body of the window
framename	Opens the linked document in a named frame

```
<a href="https://www.example.com" target="_blank">example</a>
```

Attribute ”rel”

Specifies the relationship between the current document and the linked document

Value	Description
alternate	Provides a link to an alternate representation of the document (i.e. print page, translated or mirror)
author	Provides a link to the author of the document
bookmark	Permanent URL used for bookmarking
next	Provides a link to the next document in the series
nofollow	Links to an unendorsed document, like a paid link. ("nofollow" is used by Google, to specify that the Google search spider should not follow that link)
noreferrer	Requires that the browser should not send an HTTP referer header if the user follows the hyperlink
noopener	Requires that any browsing context created by following the hyperlink must not have an opener browsing context
prev	The previous document in a selection
search	Links to a search tool for the document
tag	A tag (keyword) for the current document

MEDIA

<img>

<img> represents an image in the document

src: specifies the path to the image
alt: specifies an alternate text for the image, if the image for some reason cannot be displayed

Mainstream image formats are WebP, JPEG, PNG, GIF

https://developer.mozilla.org/en-US/docs/Web/Media/Formats/Image_types

```
<img src="epam-logo.png" alt="EPAM logo" width="500" height="600">
```

<img>

- inserts a graphical image into the page (inline)
- the **src** attribute specifies the image URL
  - **relative** URL is a partial URL specified in relation to some existing URL:
    `src="images/gollum.jpg"`
  - **absolute** URL is a complete URL to a web source:
    `src="https://i.imdb.com/images/nb15/logo2.gif"`
- HTML5 also requires an **alt** attribute describing the image
  - If the browser is unable to fetch the image, it will show the **alt** text
- if placed in an anchor, the image becomes a **link**
- **title** attribute is an optional tooltip (on ANY element)
- Other attributes are **width** and **height**:
  their value can be given as pixels or percentage of window.
  - If not used, the image will be shown in its actual size

<figure>, <figcaption>

The figure element represents some flow content, optionally with a caption, that is self-contained and is typically referenced as a single unit from the main flow of the document.

The figure element can be used to annotate illustrations, diagrams, photos, code listings, etc., that are referenced in the main content of the document, but that could, without affecting the flow of the document, be moved away from that primary content — e.g., to the side of the page, to dedicated pages, or to an appendix.

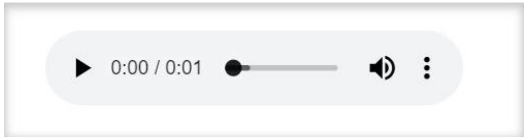
```
<figure>
    <img ...> (or video, table etc)
    <figcaption>A rabid unicorn goring a fairy.</figcaption>
</figure>
```

<audio>

Represents a sound or audio stream.

Content may be nested inside the audio element. User agents should not show this content to the user. Authors should use this content to force older browsers to use a legacy audio plugin or to inform the user of how to access the audio content.

```
<audio src="music.oga" controls>
    <a href="music.oga">Download song</a>
</audio>
```

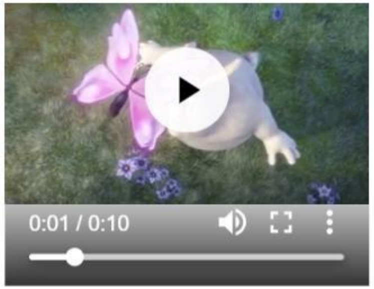


<video>

Represents a video or movie.

Content may be nested inside the video element. User agents should not show this content to the user. Authors should use this content to force older browsers to use a legacy video plugin or to inform the user of how to access the video content.

```
<video src="video.ogv" controls width="240" height="200" poster="poster.jpg">
    <a href="video.ogv">Download Video</a>
</video>
```



TABLE

<table>

<table>
It is used as a container for elements that define the contents of the table.

Each table row is defined with the **<tr>** tag. A table header is defined with the **<th>** tag.
By default, table headings are bold and centered. A table data/cell is defined with the **<td>** tag.

colspan attribute is used to combine neighboring columns in the table
rowspan - to combine rows

Inside **<table>** you can use only the following elements:
<caption>, <col>, <colgroup>, <tbody>, <td>, <tfoot>, <th>, <thead> and **<tr>**

For common column styles we can specify the information once, on a **<col>** element.
<col> elements are specified inside a **<colgroup>** container just below the opening **<table>** tag.

Tables - example

```
<table>
    <caption>Prices for buses</caption>
    <thead>
        <tr>
            <th>Bus number</th>
            <th>Price</th>
        </tr>
    </thead>
    <tbody>
        <tr>
            <td>3A</td>
            <td rowspan="2">3 EUR</td>
        </tr>
        <tr>
            <td>25</td>
        </tr>
        <tr>
            <td>133</td>
            <td>4 EUR</td>
        </tr>
    </tbody>
</table>
```

Demo:

Prices for buses

Bus number	Price
3A	3 EUR
25	
133	4 EUR

<thead>, <tfoot>, <tbody>

```
<p>Table with thead, tfoot and tbody</p>
<table>
 <thead>
  <tr>
   <th>Header content 1</th>
   <th>Header content 2</th>
  </tr>
 </thead>
 <tbody>
  <tr>
   <td>Body content 1</td>
   <td>Body content 2</td>
  </tr>
  </tbody>
  <tfoot>
   <tr>
    <td>Footer content 1</td>
    <td>Footer content 2</td>
   </tr>
  </tfoot>
 </table>
```

The <thead> tag is used to group header content in an HTML table.

The <tbody> tag is used to group the body content in an HTML table.

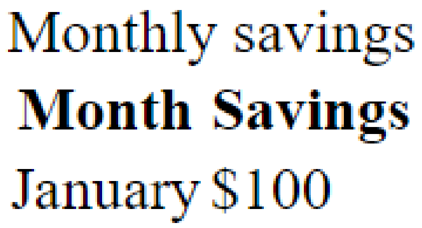
The <tbody> tag must be used in the following context:
As a child of a <table> element, after any <caption>, <colgroup>, and <thead> elements.

The <tfoot> tag is used to group footer content in an HTML table.

<caption>

The <caption> tag defines a table caption.
The <caption> tag must be inserted immediately after the <table> tag.

```
<table>
  <caption>Monthly savings</caption>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>January</td>
    <td>$100</td>
  </tr>
</table>
```



Monthly savings

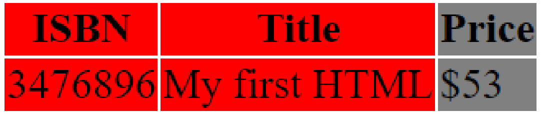
Month Savings

January $100

columns

The <colgroup> tag specifies a group of one or more columns in a table for formatting.
The <colgroup> tag must be a child of a <table> element, after any <caption> elements and before any <thead>, <tbody>, <tfoot>, and <tr> elements.

The <col> tag specifies column properties for each column within a <colgroup> element.
The <col> tag is useful for applying styles to entire columns, instead of repeating the styles for each cell, for each row.



ISBN	Title	Price
3476896	My first HTML	$53

```
<table>
  <colgroup>
    <col span="2">
    <col>
  </colgroup>
  <tr>
    <th>ISBN</th>
    <th>Title</th>
    <th>Price</th>
  </tr>
  <tr>
    <td>3476896</td>
    <td>My first HTML</td>
    <td>$53</td>
  </tr>
</table>
```

colspan, rowspan

The colspan attribute defines the number of columns a cell should span

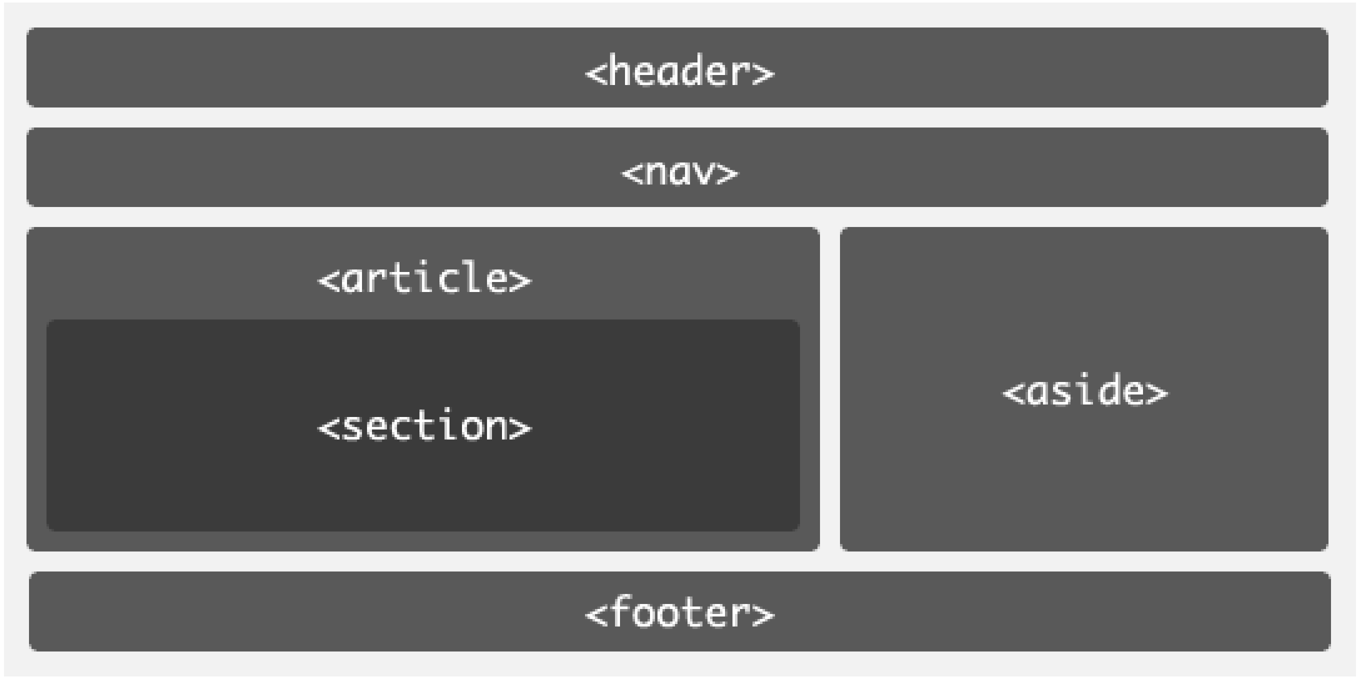
```
<table>
  <tr>
    <th>Month</th>
    <th>Savings</th>
  </tr>
  <tr>
    <td>February</td>
    <td>$80</td>
  </tr>
  <tr>
    <td colspan="2">Sum: $180</td>
  </tr>
</table>
```

The rowspan attribute specifies the number of rows a cell should span.

```
<table>
  <tr>
    <th>Month</th>
    <th>Savings</th>
    <th rowspan="2">Savings for holiday!</th>
  </tr>
  <tr>
    <td>February</td>
    <td>$80</td>
  </tr>
 </table>
```

SECTIONING

HTML5 sectioning elements



https://css-tricks.com/how-to-section-your-html

<section>

Represents a generic document or application section. In this context, a section is a thematic grouping of content, typically with a header, possibly with a footer. Examples include chapters in a book, the various tabbed pages in a tabbed dialog box, or the numbered sections of a thesis. A web site's home page could be split into sections for an introduction, news items, contact information.

```
<section>
    <h1>Level 1</h1>
    <section>
        <h1>Level 2</h1>
    </section>
</section>
```

<article>

- Represents a section of a page that consists of a composition that forms an independent part of a document, page, or site. This could be a forum post, a magazine or newspaper article, a Web log entry, a user-submitted comment, or any other independent item of content

- Each **<article>** should be identified, typically by including a heading (**<h1>** - **<h6>** element) as a child of the **<article>** element.

```
<article>
  <header>
    <h4>
       <a href="#comment" rel="bookmark">Comment #2</a> by
       <a href="http://example.com/">Jack Osborne</a>
    </h4>
  </header>
  <p>Pellentesque habitant morbi tristique senectus.</p>
</article>
```

<nav>

- Represents navigation for a document. The nav element is a section containing links to other documents or to parts within the current document.

- Not all groups of links on a page need to be in a nav element — only groups of primary navigation links. It is common for footers to have a list of links to various key parts of a site, but the footer element is more appropriate in such cases.

```
<nav>
  <ul>
    <li>Nav Link</li>
    <li>Nav Link</li>
    <li>Nav Link</li>
  </ul>
</nav>
```

<aside>

- Represents a section of a page consisting of content that is tangentially related to the content around the aside element, and which could be considered separate from that content.

- Such sections are often represented as sidebars in printed typography.

```
<aside>
  <h2>Blogroll</h2>
  <ul>
    <li><a href="#">My Friend</a></li>
    <li><a href="#">My Other Friend</a></li>
    <li><a href="#">My Best Friend</a></li>
  </ul>
</aside>
```

<main>

- Represents the dominant content of the **<body>** of a document, portion of a document or application.
- The main content area consists of content that is directly related to or expands upon the central topic of a document, or the central functionality of an application.
- The content of a **<main>** element should be unique to the document

```
<main>
  <h1>Level 1</h1>
  <section>
    <h2>Level 2</h2>
  </section>
</main>
```

<header>

- Represents the "header" of a document or section of a document. The header element is typically used to group a set of h1–h6 elements to mark up a page's title with its subtitle or tagline.

- header elements may, however, contain more than just the section's headings and subheadings — e.g., version history information or publication date.

```
<header>
    <h1>Header string</h1>
    <p>Other content</p>
</header>
```

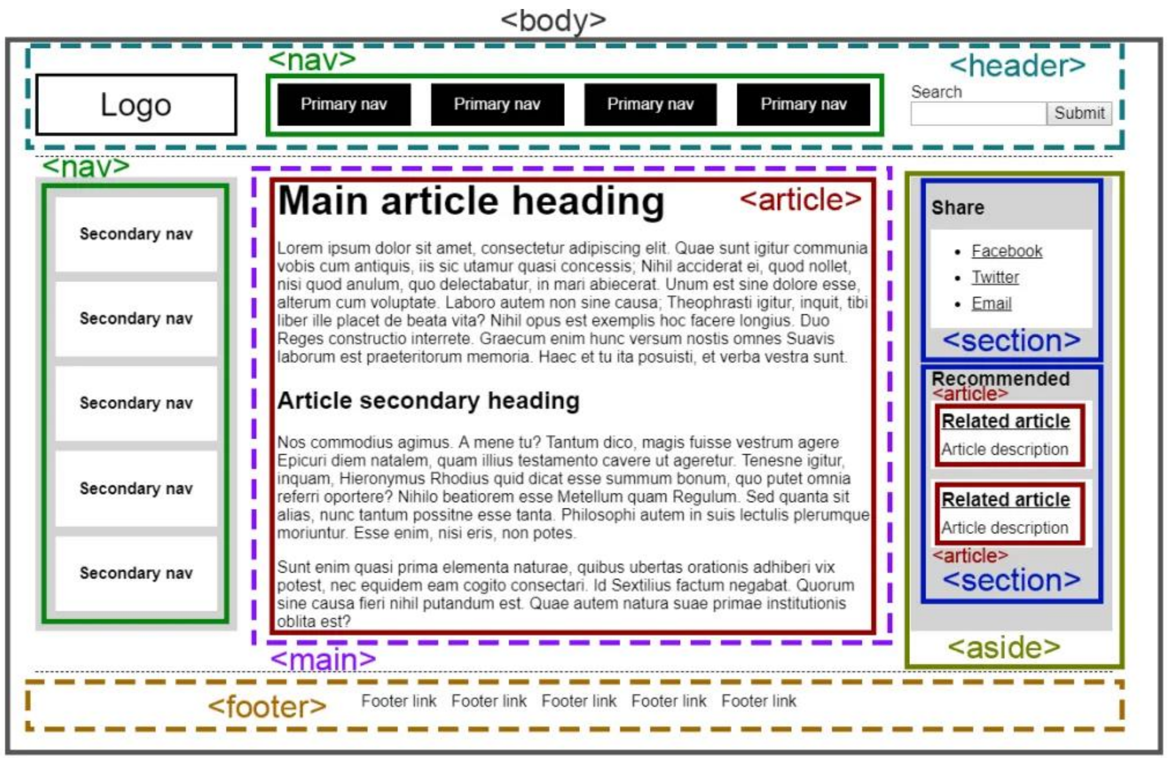
<footer>

- Represents the "footer" of a document or section of a document. The footer element typically contains metadata about its enclosing section, such as who wrote it, links to related documents, copyright data, etc.

- Contact information for the section given in a footer should be marked up using the address element.

```
<footer>
    <h3>Contacts</h3>
    <p>Other content</p>
</footer>
```

HTML sectioning in practice



Main article heading

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Quae sunt igitur communia vobis cum antiquis, iis sic utamur quasi concessis; Nihil acciderat ei, quod nollet, nisi quod anulum, quo delectabatur, in mari abiecerat. Unum est sine dolore esse, alterum cum voluptate. Laboro autem non sine causa; Theophrasti igitur, inquit, tibi liber ille placet de beata vita? Nihil opus est exemplis hoc facere longius. Duo Reges constructio interrete. Graecum enim hunc versum nostis omnes Suavis laborum est praeteritorum memoria. Haec et tu ita posuisti, et verba vestra sunt.

Article secondary heading

Nos commodius agimus. A mene tu? Tantum dico, magis fuisse vestrum agere Epicuri diem natalem, quam illius testamento cavere ut ageretur. Tenesne igitur, inquam, Hieronymus Rhodius quid dicat esse summum bonum, quo putet omnia referri oportere? Nihilo beatiorem esse Metellum quam Regulum. Sed quanta sit alias, nunc tantum possitne esse tanta. Philosophi autem in suis lectulis plerumque moriuntur. Esse enim, nisi eris, non potes.

Sunt enim quasi prima elementa naturae, quibus ubertas orationis adhiberi vix potest, nec equidem eam cogito consectari. Id Sextilius factum negabat. Quorum sine causa fieri nihil putandum est. Quae autem natura suae primae institutionis oblita est?

FORMS, INPUTS

Forms

A form on a web page allows a user to enter data that is sent to a server for processing.

- A form is an area that can contain form elements: buttons, checkboxes, text fields, radio buttons, drop-down menus, etc
- Form elements include: Other kinds of HTML tags can be mixed in with the form elements
- The syntax is: **<form>** … form elements … **</form>**
- A form usually contains a Submit button to send the information in the form elements to the server

Example

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```

datalist experiment: https://demo.agektmr.com/datalist/

<form>

The **<form arguments> ... </form>** tag encloses form elements

The arguments to form tell what to do with the user input:

- **action="url"** specifies where to send the data when the Submit button is clicked
- **method="get"**
  - Form data is sent as a URL with ?form_data info appended to the end;
  - Can be used only if data is all URL encoded;
- **method="post"**
  - Form data is sent in the body of the HTTP request;
  - Cannot be bookmarked by most browsers;
- **target="target" -** Tells where to open the page sent as a result of the request
  - target="_blank" means open in a new window;
  - target="_top" means use the same window.

Methods: GET and POST

GET - Requests data from a specified resource

/test/demo_form.php**?name1=value1&name2=value2**

POST - Submits data to be processed to a specified resource

```
POST /test/demo_form.php  HTTP/1.1
Host: example.com
name1=value1&name2=value2
```

Full list of HTTP methods (especially when dealing with HTTP APIs):
https://developer.mozilla.org/en-US/docs/Web/HTTP/Methods

Compare GET vs. POST

GET	POST
GET requests can be cached GET requests remain in the browser history GET requests can be bookmarked GET requests should never be used when dealing with sensitive data GET requests have length restrictions GET requests should be used only to retrieve data	POST requests are never cached POST requests do not remain in the browser history POST requests cannot be bookmarked POST requests have no restrictions on data length

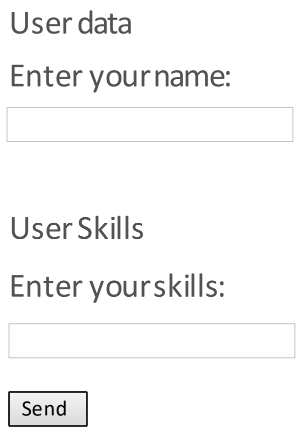
Form Elements

Tag	Description
<form>	Defines an HTML form for user input
<input>	Defines an input control
<textarea>	Defines a multiline input control (text area)
<label>	Defines a label for an <input> element
<fieldset>	Groups related elements in a form
<legend>	Defines a caption for a <fieldset> element
<select>	Defines a drop-down list
<optgroup>	Defines a group of related options in a drop-down list
<option>	Defines an option in a drop-down list
<button>	Defines a clickable button
<datalist>	Specifies a list of pre-defined options for input controls

<fieldset>, <legend>

The fieldset is a useful tool for organizing and grouping related items within a form, and has been used for a long time in desktop applications.

```
<form action="script.php">
    <fieldset>
        <legend>User data</legend>
        <label for="userName">Enter your name:</label>
        <input type="text" id="userName" />
    </fieldset>
    <fieldset>
        <legend>User Skills</legend>
        <label for="userSkills">Enter your skills:</label>
        <input type="text" id="userSkills" />
    </fieldset>
    <input type="submit" name="Submit" value="Send">
</form>
```


User data

Enter your name:

User Skills

Enter your skills:

The <input> tag

The input element represents a typed data field, usually with a form control to allow the user to edit the data.

Most, but not all form elements use the input tag, with a type="..." argument to tell which kind of element it is. The most used types text, checkbox, radio, password, hidden, submit, reset, button, file, or image.

Input tag arguments:

name: the name of the element
value: the "value" of the element; used in different ways for different values of type
readonly: the value cannot be changed
disabled: the user can't do anything with this element

Input types

This element can be displayed in several ways, depending on the type attribute:

- text;
- password;
- submit;
- radio;
- checkbox;
- button;
- color;
- date;
- datetime-local;
- email;
- month;
- number;
- range;
- search;
- tel;
- time;
- url;
- week;

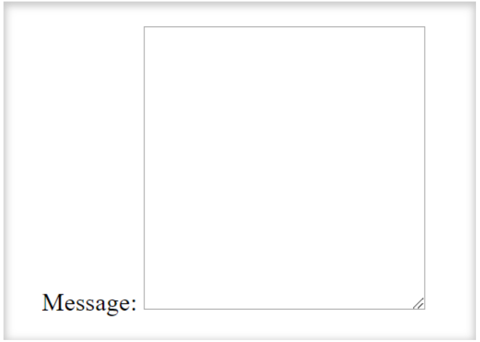
See full list: https://developer.mozilla.org/en-US/docs/Web/HTML/Element/input

<textarea>

The textarea element represents a multiline plain text editor control for the element's raw value.

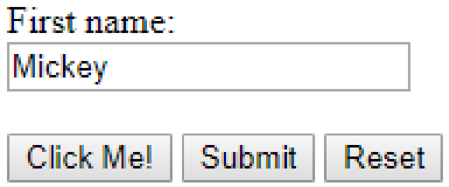
The contents of the control represent the control's default value.

```
<textarea cols="20" rows="10"></textarea>
```



Example [submit reset button]

```
<form action="/action_page.php">
  First name:<br>
  <input type="text" name="firstname" value="Mickey"><br>
  <input type="button" onclick="alert('Hello World!')" value="Click Me!">
  <input type="submit" value="Submit">
  <input type="reset">
</form>
```



<datalist>

The **<datalist>** element specifies a list of pre-defined options for an <input> element.

```
<input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
  ...
```

Hidden fields

- All input fields are sent back to the server, including hidden fields.
  This is a way to include information that the user doesn’t need to see.

- The value of a hidden field can be set programmatically (by JavaScript) before the form is submitted.

```
<input type="hidden" name="secretMessage" value="hello">
```

Examples

The **<input type="color">** is used for input fields that should contain a color.

```
<input type="color" name="favcolor" value="0000FF">
```

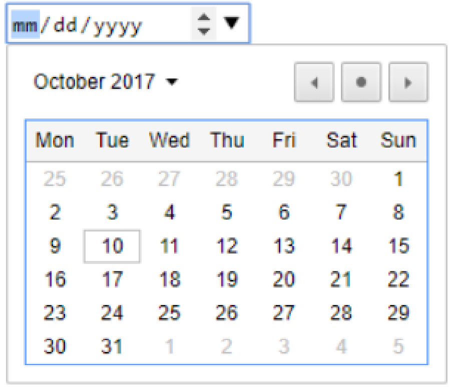
The **<input type="date">** is used for input fields that should contain a date.

```
<input type="date" name="bday">

<input type="date" name="bday" min="2000-01-02">

<input type="datetime-local" name="bdaytime">

<input type="month" name="bdaymonth">
```



Number and Range

The **<input type="number">** defines a **numeric** input field.

```
<form>
  Quantity (between 1 and 5):
  <input type="number" name="quantity" min="1" max="5">
</form>
```

The **<input type="range">** defines a control for entering a number whose exact value is not important (like a slider control). Default range is 0 to 100. However, you can set restrictions on what numbers are accepted with the min, max, and step attributes

```
<form>
  <input type="range" name="points" min="0" max="10">
</form>
```

Input - attributes

- disabled
- max
- maxlength
- placeholder
- min
- pattern
- readonly
- tabindex
- required
- size
- step
- value

```
<input type="number" min="10" max="100" value="50">
```

GOOD TO KNOW

<picture>

- Serves as a container for zero or more **<source>** elements and one **<img>** element to provide versions of an image for different display device scenarios.
- The browser will consider each of the child **<source>** elements and select one corresponding to the best match found; if no matches are found among the **<source>** elements, the file specified by the **<img>** element's **src** attribute is selected.

```html
<picture>
    <source srcset="/media/examples/picture-small.png"
            media="(max-width: 1000px)">
    <source srcset="/media/examples/picture-wide.png"
            media="(min-width: 1000px)">
    <img src="/media/examples/picture-narrow.png">
</picture>
```

srcset image attribute

- Allows to list multiple alternative image sources which vary in pixel density
- This allows the browser to pick an image of the appropriate quality for the user’s device

```
<img src=“images/low-res.jpg"
     srcset=“images/low-res.jpg 1x, images/high-res.jpg 2x, images/ultra-high-res.jpg 3x">
```

Canvas and SVG

- Provide native drawing functionality
- Completely integrated into HTML5 documents (part of DOM)
- Can be styled with CSS
- Can be controlled with JavaScript
- Use for animation, charts, images, pixel manipulation, and so on
- Canvas supports 2D and 3D (WebGL)

HTML5 Canvas

The HTML <canvas> element is used to draw graphics, on the fly, via JavaScript.

The <canvas> element is only a container for graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

Scalable Vector Graphics

SVG stands for Scalable Vector Graphics, and it is a language for describing 2D-graphics and graphical applications in XML and the XML is then rendered by an SVG viewer.

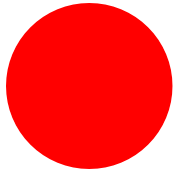
- Provide native drawing functionality
- Completely integrated into HTML5 documents (part of DOM)
- Can be styled with CSS
- Can be controlled with JavaScript

<svg>

```
<div>HTML5 SVG Circle</div>
<svg id="svgelement" height="150" xmlns="http://www.w3.org/2000/svg">
    <circle id="redcircle" cx="50" cy="50" r="50" fill="red" />
</svg>


<div>HTML5 SVG Rectangle</div>
<svg id="svgelement" height="150" xmlns="http://www.w3.org/2000/svg">
    <rect id="redrect" width="300" height="100" fill="red" />
</svg>
```

HTML5 SVG Circle



HTML5 SVG Rectangle

SVG Icons

Icons are a great place to start using SVG on the web. SVGs are flexible, resolution independent and lightweight, so icons naturally lend themselves to the vector format.

With SVG, the freedom exists to style individual elements within an icon, which opens entirely new possibilities.

Q&A

edu_hu@epam.com