

HTML and its origin

Web Engineering



<https://www.w3schools.com/html/default.asp>

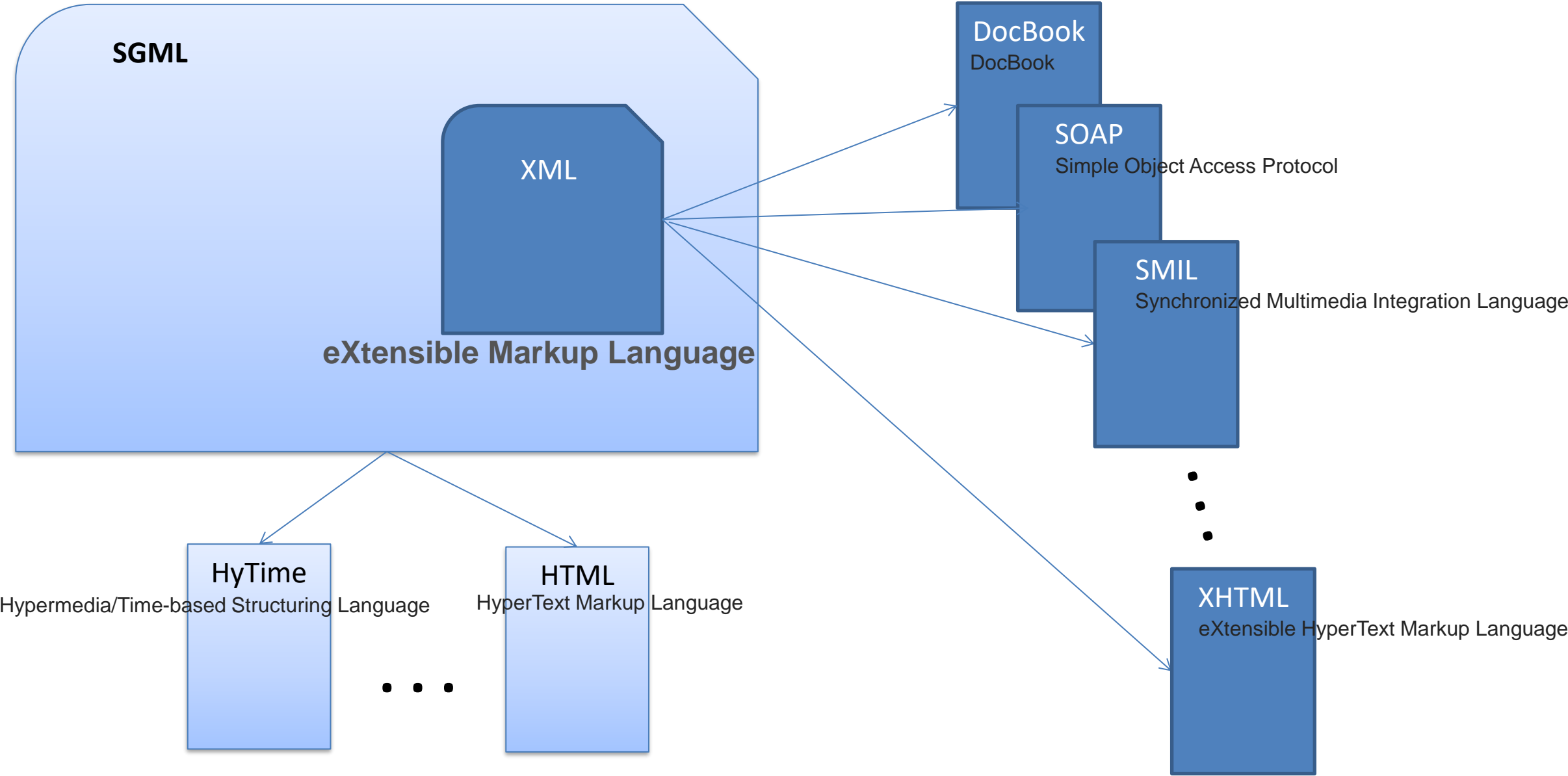
```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>

    <h1>This is a Heading</h1>
    <p>This is a paragraph.</p>

  </body>
</html>
```

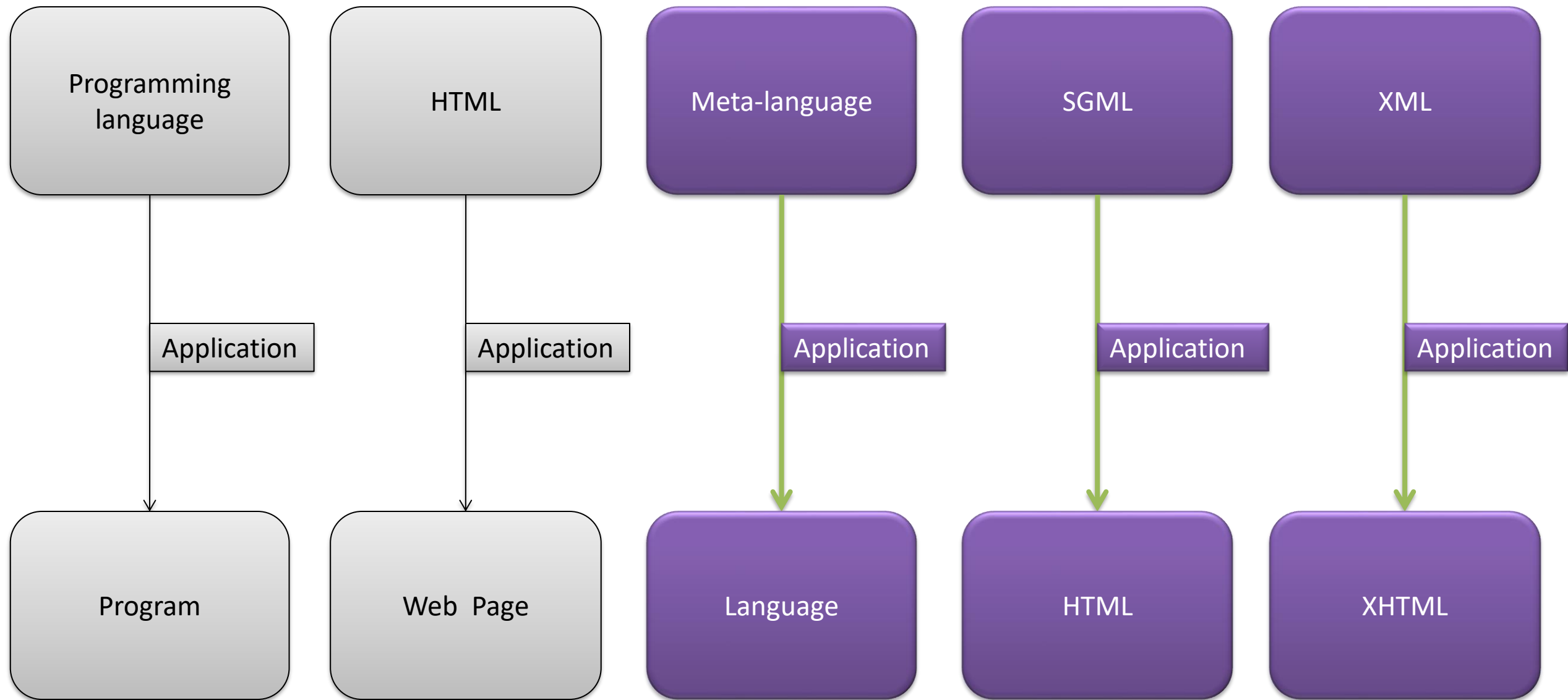
What are the origins of HTML?

Standard Generalized Markup Language



SGML and XML are meta-languages

What is a meta-language?



Each markup language defined in SGML is called the *SGML application*.

An *SGML application* is characterized by:

1. **An SGML statement**. SGML declaration specifies which characters and delimiters can be used in the application.
2. **A document type definition (DTD)**. The DTD defines the syntax of tag constructs. The DTD may include additional definitions such as numeric and named character entities (for example: " or ").
3. **The specification that describes the semantics** to assign to tags. This specification also imposes syntactic constraints that cannot be expressed in DTD.
4. **Document instances** that contain data (content) and tags. Each instance contains the reference to the DTD to be used to interpret it.

Which of these characters are valid in HTML 4.0?

Unicode
1.1

[ISO/IEC 10646-1:1993 UCS-4](#)

[HTML 4.0](#)

UCS = Universal Character Set

119575



32160

綠

綠

9

Tabulação horizontal

Tabulação horizontal

23

Fim de bloco de transmissão

INVALID

SGML Declaration

<https://www.w3.org/TR/html4/sgml/sgmldecl.html>

```
<!SGML "ISO 8879:1986 (WWW)"
--
  SGML Declaration for HyperText Markup Language version HTML 4
  With support for the first 17 planes of ISO 10646 and
  increased limits for tag and literal lengths etc.
--

CHARSET
BASESET "ISO Registration Number 177//CHARSET
        ISO/IEC 10646-1:1993 UCS-4 with
        implementation level 3//ESC 2/5 2/15 4/6"
DESCSET 0 9 UNUSED
        9 2 UNUSED
        11 2 UNUSED
        13 1 13
        14 18 UNUSED
        32 95 32
        127 1 UNUSED
        128 32 UNUSED
        160 55136 160
        55296 2048 UNUSED -- SURROGATES --
        57344 1056768 57344

CAPACITY
SGMLREF
TOTALCAP 150000
GRPCAP 150000
ENTCAP 150000

SCOPE DOCUMENT
SYNTAX SHUNCHAR CONTROLS 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
        17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
```

Tabulação horizontal
(9)

Fim de bloco de transmissão
(23)

DTD – Document Type Definition

A DTD defines the document structure with a list of legal **elements** and **attributes**

DTD Newspaper Example

```
<!DOCTYPE NEWSPAPER [  
  
  <!ELEMENT NEWSPAPER (ARTICLE+)>  
  <!ELEMENT ARTICLE (HEADLINE,BYLINE,LEAD,BODY,NOTES)>  
  <!ELEMENT HEADLINE (#PCDATA)>  
  <!ELEMENT BYLINE (#PCDATA)>  
  <!ELEMENT LEAD (#PCDATA)>  
  <!ELEMENT BODY (#PCDATA)>  
  <!ELEMENT NOTES (#PCDATA)>  
  
  <!ATTLIST ARTICLE AUTHOR CDATA #REQUIRED>  
  <!ATTLIST ARTICLE EDITOR CDATA #IMPLIED>  
  <!ATTLIST ARTICLE DATE CDATA #IMPLIED>  
  <!ATTLIST ARTICLE EDITION CDATA #IMPLIED>  
  

```



Document Instances

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
    "http://www.w3.org/TR/html4/strict.dtd">  
<HTML>  
  <HEAD>  
    <TITLE>My first HTML document</TITLE>  
  </HEAD>  
  <BODY>  
    <P>Hello world!  
  </BODY>  
</HTML>
```

DTD – Document Type Definition

- Entity definitions

```
<!ENTITY % head.misc "SCRIPT|STYLE|META|LINK">  
<!ENTITY % heading "H1|H2|H3|H4|H5|H6">  
<!ENTITY % attrs "%coreattrs %il8n %events">  
<!ENTITY attrs "substitution text">
```

- Element definitions

```
<!ELEMENT OL - - (LI)+>  
<!ELEMENT BR - O EMPTY>  
<!ELEMENT OPTION - O #PCDATA>  
<!ELEMENT TABLE - - (CAPTION?, (COL*|COLGROUP*), THEAD?, TFOOT?, TBODY+)>
```

- Attribute definitions

...

DTD – Document Type Definition

- Entity definitions
 - ...
- Element definitions
 - ...
- Attribute definitions

```
<!ENTITY % TAlign "(left|center|right)">
...
<!-- table element --
<!-- table width relative to window --
<!-- used for immediate display mode --
<!-- table position relative to window --
>
<!-- th element --
<!-- th element --
>
```

DTD – Document Type Definition

- PCDATA
parsed character data
- CDATA
character data

<https://www.w3.org/TR/html401/sgml/dtd.html>

What is the meaning, in SGML DTD for HTML 4.0:

```
<!ELEMENT BR - O EMPTY>
```

And why in the XHTML 1.0 DTD is:

```
<!ELEMENT br EMPTY>
```

?

HTML 4

<!ELEMENT BR – O EMPTY>

BR exists

Opens with

No closure
required

It has no
content

<!ELEMENT br EMPTY>

br exists

In HTML5 can be done in two ways, depending
on the type of syntax you choose for the page:
<https://html.spec.whatwg.org/multipage/introduction.html#html-vs-xhtml>

XHTML 1.0

What is the meaning, in SGML DTD for HTML 4.0:

```
<!ELEMENT TABLE - - (CAPTION?, (COL* | COLGROUP*),  
    THEAD?, TFOOT?, TBODY+)>
```

?

<!ELEMENT TABLE - - (CAPTION?, (COL* | COLGROUP*), THEAD?, TFOOT?, TBODY+)>

That the TABLE element:

- Needs <TABLE> and </TABLE>
- And that, in content:
 - * May have one CAPTION or not
 - * May have several COL or COLGROUP or not
 - * May have one THEAD or not
 - * May have one TFOOT or not
 - * Must have at least one TBODY
 - * Must follow the order given (except between COL and COLGROUP)

No, because the content does not specify #PCDATA or #CDATA



A TABLE element can have text inside?

```
<!ELEMENT TABLE - - (CAPTION?, (COL* | COLGROUP*),  
    THEAD?, TFOOT?, TBODY+)>
```

<!ELEMENT TABLE - - (CAPTION?, (COL* | COLGROUP*), THEAD?, TFOOT?, TBODY+)>

So how can we have these tables in HTML?

Station	Latitude	g
Quito, Ecuador	zero degrees N	nine point seven eight zero m slash s sup two base
Madras, India	one three degrees N	nine point seven eight three m slash s sup two base
Hong Kong	two two degree N	nine point seven eight eight m slash s sup two base
Cairo, Egypt	three zero degree N	nine point seven nine three m slash s sup two base
New York, USA	four one degree N	nine point eight zero three m slash s sup two base
London, England	five one degree N	nine point eight one one m slash s sup two base
Oslo, Norway	six zero degree N	nine point eight one nine m slash s sup two base
Murmansk, USSR	six nine degree N	nine point eight two five m slash s sup two base
Spitsbergen	eight zero degree N	nine point eight three one m slash s sup two base
North Pole	nine zero degree N	nine point eight three two

Nothing in this line prevents text inside
CAPTION, COL, COLGROUP, THEAD, TFOOT,
or TBODY ...

```
<!ELEMENT TABLE - - (CAPTION?, (COL* | COLGROUP*),  
    THEAD?, TFOOT?, TBODY+)>
```

following the definitions...

```
<!ELEMENT CAPTION    - - (%inline;)*    -- table caption -->
<!ELEMENT THEAD       - O (TR)+          -- table header -->
<!ELEMENT TFOOT       - O (TR)+          -- table footer -->
<!ELEMENT TBODY       O O (TR)+          -- table body -->
<!ELEMENT COLGROUP   - O (COL)*          -- table column group -->
<!ELEMENT COL         - O EMPTY          -- table column -->
<!ELEMENT TR         - O (TH|TD)+        -- table row -->
<!ELEMENT (TH|TD)    - O (%flow;)*       -- table header cell, table data cell-->
```

```
<!ENTITY % block
    "P | %heading; | %list; | %preformatted; | DL | DIV | NOSCRIPT | BLOCKQUOTE |
    FORM | HR | TABLE | FIELDSET | ADDRESS">
```

```
<!ENTITY % inline "#PCDATA | %fontstyle; | %phrase; | %special; | %formctrl;">
```

```
<!ENTITY % flow "%block; | %inline;">
```

<!ELEMENT TABLE - - (CAPTION?, (COL* | COLGROUP*), THEAD?, TFOOT?, TBODY+)>

Is this table valid?

Yes?! (Attention to THEAD)

<TABLE border=1>

<THEAD><TR>Tabela 2

<TBODY><TR><TD>1<TD>2<TD>3

<TR><TD>6<TD>8<TD>5

<TBODY><TR><TD>234</TD><TD>123</TD>

</TABLE>

Tabela 2

1	2	3
6	8	5
234	123	

<!ELEMENT TABLE - - (CAPTION?, (COL* | COLGROUP*), THEAD?, TFOOT?, TBODY+)>

Is this table valid?

<TABLE border=1>

<THEAD><TR>Tabela 2

<TBODY><TR><TD>1<TD>2<TD>3

<TR><TD>6<TD>8<TD>5

<TFOOT><TR>Dezembro de 2008</TR>

</TABLE>

No!

(TFOOT cannot appear after TBODY)

In the HTML 4.01 specification:

<http://www.w3.org/TR/REC-html40/struct/tables.html>

```
<!ELEMENT TABLE - -
    (CAPTION?, (COL*|COLGROUP*), THEAD?, TFOOT?, TBODY+)>

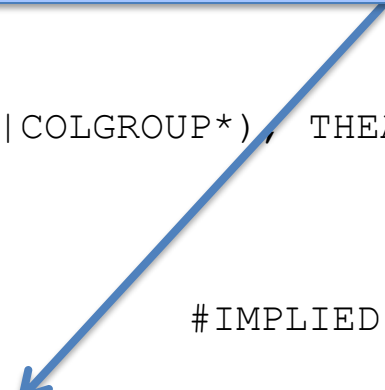
<!ATTLIST TABLE                                -- table element --
    %attrs;                                     -- %coreattrs, %i18n, %events --
    summary      %Text;                        #IMPLIED -- purpose/structure for speech
                                                    output--
    width         %Length;                     #IMPLIED -- table width --
    border        %Pixels;                    #IMPLIED -- controls frame width around
                                                    table --
    frame         %TFrame;                     #IMPLIED -- which parts of frame to
                                                    render --
    rules         %TRules;                    #IMPLIED -- rulings between rows and
                                                    cols --
    cellspacing   %Length;                    #IMPLIED -- spacing between cells --
    cellpadding  %Length;
>
```

**<!ENTITY % Length "CDATA" -- nn for pixels
or nn% for percentage length -->**

**<!ENTITY % Length "CDATA" -- nn for pixels
or nn% for percentage length -->**

```
<!ELEMENT TABLE - -
    (CAPTION?, (COL*|COLGROUP*), THEAD?, TFOOT?, TBODY+)>

<!ATTLIST TABLE -- table element --
    %attrs; -- %coreattrs, %i18n, %events --
    summary %Text; #IMPLIED -- purpose/structure for speech
    width %Length; #IMPLIED -- table width --
    border %Pixels; #IMPLIED -- controls frame width around
    frame %TFrame; #IMPLIED -- which parts of frame to
    rules %TRules; #IMPLIED -- rulings between rows and
    cellspacing %Length; #IMPLIED -- spacing between cells --
    cellpadding %Length; #IMPLIED -- spacing within cells --
>
```



<TABLE width="10%">

means 10% of the screen, 10% of the
browser window or 10% of something else?

Does this definition say TABLE is a table?

```
<!ELEMENT TABLE - - (CAPTION?, (COL* | COLGROUP*), THEAD?,  
                        TFOOT?, TBODY+)>
```

Not! Just say how you can type
(it's the **syntax**)

width %Length

just say how you spell it (it's the **syntax**)

Don't say what it means
(don't have **semantics**)

```
<!ELEMENT TABLE - - (CAPTION?, (COL* | COLGROUP*), THEAD?,  
                        TFOOT?, TBODY+)>
```

```
width %Length
```

Semantics is this ...

width = *length*

This attribute specifies the desired width of the entire table and is intended for visual user agents. When the value is a percentage value, the value is relative to the user agent's available horizontal space. In the absence of any width specification, table width is determined by the user agent.

-- HTML 4.01 specification

Semantics is this...

<https://www.w3.org/TR/REC-html40/struct/tables.html>

Tables in HTML documents - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://www.w3.org/TR/REC-html40/struct/tables.html

Most Visited Getting Started Latest Headlines

W3C Recommendation

11.1 Introduction to tables

The HTML table model allows authors to arrange data -- text, preformatted text, images, links, forms, form fields, other tables, etc. -- into rows and columns of cells.

Each table may have an associated caption (see the [CAPTION](#) element) that provides a short description of the table's purpose. A longer description may also be provided (via the [summary](#) attribute) for the benefit of people using speech or Braille-based user agents.

[Table rows](#) may be grouped into a head, foot, and body sections, (via the [THEAD](#), [TFOOT](#) and [TBODY](#) elements, respectively). Row groups convey additional structural information and may be rendered by user agents in ways that emphasize this structure. User agents may exploit the head/body/foot division to support scrolling of body sections independently of the head and foot sections. When long tables are printed, the head and foot information may be repeated on each page that contains table data.

Authors may also [group columns](#) to provide additional structural information that may be exploited by user agents. Furthermore, authors may declare column properties at the start of a table definition (via the [COLGROUP](#) and [COL](#) elements) in a way that enables user agents to render the table incrementally rather than having to wait for all the table data to arrive before rendering.

[Table cells](#) may either contain "header" information (see the [TH](#) element) or "data" (see the [TD](#) element). Cells may span multiple rows and columns. The HTML 4 table model allows authors to label each cell so that [non-visual user agents](#) may more easily communicate heading information about the cell to the user. Not only do these mechanisms greatly assist users with visual disabilities, they make it possible for multi-modal wireless browsers with limited display capabilities (e.g., Web-enabled pagers and phones) to handle tables.

Tables should not be used purely as a means to layout document content as this may present problems when rendering to non-visual media. Additionally, when used with graphics, these tables may force users to scroll horizontally to view a table designed on a system with a larger display. To minimize these problems, authors should use [style sheets](#) to control layout rather than tables.

Note. *This specification includes more detailed information about tables in sections on [table design rationale](#) and [implementation issues](#).*

Here's a simple table that illustrates some of the features of the HTML table model. The following table definition:

```
<TABLE border="1"
  summary="This table gives some statistics about fruit
    flies: average height and weight, and percentage
    with red eyes (for both males and females).">
  <CAPTION><EM>A test table with merged cells</EM></CAPTION>
  <TR><TH rowspan="2"><TH colspan="2">Average
    <TH rowspan="2">Red<BR>eyes
  <TR><TH>height<TH>weight
  <TR><TH>Males<TD>1.9<TD>0.003<TD>40%
  <TR><TH>Females<TD>1.7<TD>0.002<TD>43%
</TABLE>
```

might be rendered something like this on a tty device:

```

  A test table with merged cells
  /-----\
  |         | Average | Red   | |
  |         |-----| eyes   |
  |         | height | weight |
  |         |-----|
  | Males   | 1.9     | 0.003 | 40%  |
  |         |-----|
  | Females | 1.7     | 0.002 | 43%  |
  \-----/
```

Who uses semantics?

Browser engine

- This web browser component is a rendering mechanism responsible for showing the graphical or textual representation of HTML+CSS content.

Who uses semantics?

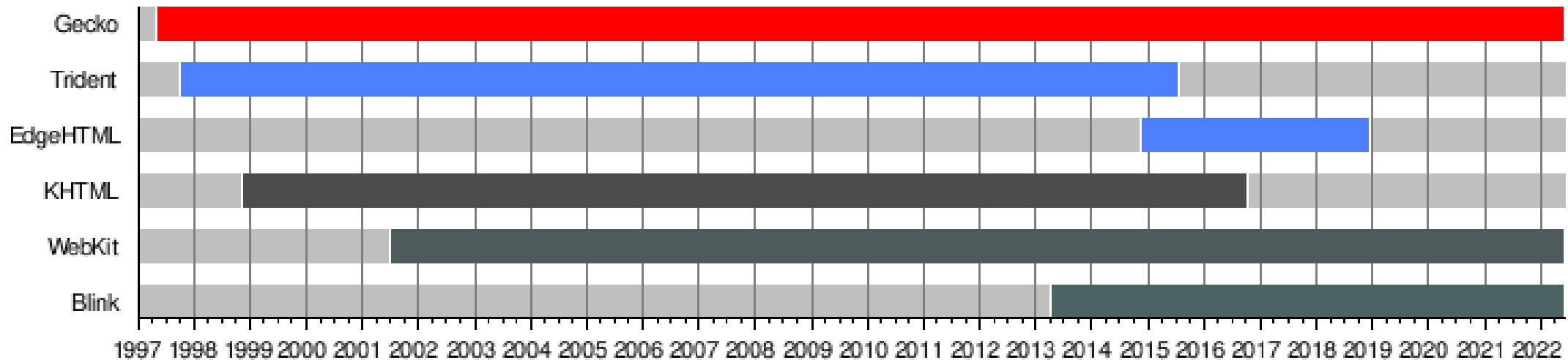
Browser engine

- In addition to layout and rendering, a browser engine enforces the security policy between documents, handles navigation through hyperlinks and data submitted through forms, and implements the Document Object Model (DOM) data structure exposed to page scripts.

Who uses semantics?

Active development of the browser engine

(is when relevant new Web standards continue to be added to the engine)



General information about browser engines

Engine ^	Status ^[a] ◆	Steward ◆	License ◆	Embedded in ◆
Blink	Active	Google	GNU LGPL, BSD-style	Google Chrome and all other Chromium-based browsers, notably Microsoft Edge, Brave, Vivaldi, Samsung Internet and Opera ^[4]
EdgeHTML	Maintained	Microsoft	Proprietary	some UWP apps, ^[8] formerly in the Edge browser ^[9]
Flow ^[10]	Maintained	Ekioh ^[11]	Proprietary	Flow browser ^[12]
Gecko	Active	Mozilla	Mozilla Public	Firefox browser and Thunderbird email client
Goanna ^[b]	Active	M. C. Straver ^[6]	Mozilla Public	Pale Moon, Basilisk and K-Meleon browsers
KHTML ^[21]	Discontinued	KDE	GNU LGPL	formerly in the Konqueror browser ^[22]
LibWeb ^[e]	Maintained	hobbyists ^[20]	2-clause BSD	Ladybird browser ^[19]
NetSurf ^[d]	Maintained	hobbyists ^[17]	GNU GPLv2	NetSurf browser ^[18]
Presto	Discontinued	Opera	Proprietary	formerly in the Opera browser
Servo	Maintained	Linux Foundation	Mozilla Public	experimental browsers ^{[13][14]}
Trident ^[c]	Maintained	Microsoft	Proprietary	Internet Explorer browser
WebKit	Active	Apple	GNU LGPL, BSD-style	Safari browser, plus all browsers for iOS, ^[3] GNOME Web

HTML – structure and syntax

Header

<HEAD>

<TITLE>

<META>

<BASE>

<LINK>

<STYLE>

<SCRIPT>

<https://www.w3schools.com/html/default.asp>

HTML – structure and syntax

Body

<BODY>

<TABLE>

<H1>

<H2>

<P>

<FORM>

...

<https://www.w3schools.com/html/default.asp>

HTML – structure and syntax

Body

<BODY>

...

<DIV>

<IFRAME>

...

HTML – structure and syntax

Inline Elements

An inline element does not start on a new line.

An inline element only takes up as much width as necessary.

Here are the inline elements in HTML:

| | | | | | | |
|----------|------------|-----------|---------|----------|----------|-------|
| <a> | <abbr> | <acronym> | | <bdo> | <big> |
 |
| <button> | <cite> | <code> | <dfn> | | <i> | |
| <input> | <kbd> | <label> | <map> | <object> | <output> | <q> |
| <samp> | <script> | <select> | <small> | | | <sub> |
| <sup> | <textarea> | <time> | <tt> | <var> | | |

HTML – structure and syntax

Block-level Elements

A block-level element always starts on a new line, and the browsers automatically add some space (a margin) before and after the element.

A block-level element always takes up the full width available (stretches out to the left and right as far as it can).

Here are the block-level elements in HTML:

| | | | | | | |
|------------------------------------|------------------------------|-------------------------------|---------------------------------|-----------------------------|-----------------------------|-------------------------------|
| <code><address></code> | <code><article></code> | <code><aside></code> | <code><blockquote></code> | <code><canvas></code> | <code><dd></code> | <code><div></code> |
| <code><dl></code> | <code><dt></code> | <code><fieldset></code> | <code><figcaption></code> | <code><figure></code> | <code><footer></code> | <code><form></code> |
| <code><h1>-<h6></code> | <code><header></code> | <code><hr></code> | <code></code> | <code><main></code> | <code><nav></code> | <code><noscript></code> |
| <code></code> | <code><p></code> | <code><pre></code> | <code><section></code> | <code><table></code> | <code><tfoot></code> | <code></code> |
| <code><video></code> | | | | | | |

HTML – structure and syntax

HTML elements that can generate HTTP requests

```
<a href=http://www.w3schools.com target="_self">Visit W3Schools.com!</a>
```

```
<form action="demo_form.asp" method="get">
```

```
  First name: <input type="text" name="fname"><br>
```

```
  Last name: <input type="text" name="lname"><br>
```

```
  <input type="submit" value="Submit">
```

```
  <input type="reset" value="Clear">
```

```
</form>
```


HTML – structure and syntax

HTML elements that generate HTTP requests

```
<FORM action="http://www.nobody.name/sendFile" method="post"
  enctype="multipart/form-data">
  <P>
    <LABEL for="localpath">Choose local file: </LABEL >
    <INPUT type="file" id="localpath"><BR>
    <LABEL for="filename">Target file name: </LABEL >
    <INPUT type="text" id="filename"><BR>
    <LABEL for="location">Target location: </LABEL >
    <INPUT type="text" id="location"><BR>
    <INPUT type="submit" value="Send"> <INPUT type="reset">
  </P>
</FORM>
```

- *application/x-www-form-urlencoded*
- *multipart/form-data*
- *text/plain*

HTML – structure and syntax

HTML elements that can generate HTTP requests

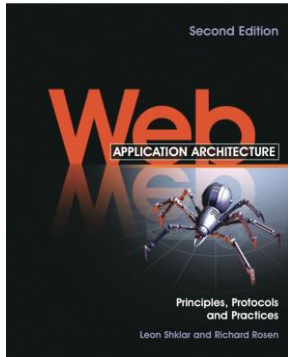


In addition to links `<a>` and forms `<form>`, are there more HTML elements responsible for generating HTTP requests?

Questions from the book

Would it be very difficult to implement an HTML interpreter? Why? How would you represent the semantics of HTML elements?

Bibliography

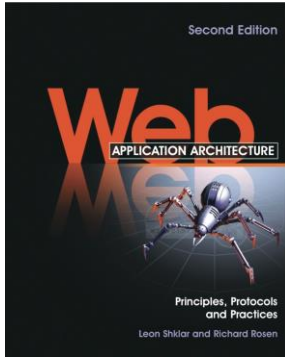


Shklar, Leon & Rosen, Rich (2009). *Web Application Architecture: Principles, Protocols and Practices*. Chichester, Reino Unido: John Wiley & Sons.

Pages: 63 to 83

Chapter 4

Next class



Shklar, Leon & Rosen, Rich (2009). *Web Application Architecture: Principles, Protocols and Practices*. Chichester, Reino Unido: John Wiley & Sons.

Pages: 85 to 96

Chapter 5

5.1 CoreXML

5.2 XHTML