Internet Programming I

Chapter 2
Web Development Using HTML

Chere L. (M. Tech) and Biruk G. (MSc.)
Lecturer, Dept. of Software Eng.

Nov 2021, AASTU

Objectives

After success completion of the chapter you will be able to:

- Identify the core web development technologies
- Understand the HTML document structure and contents models.
- Identify and use HTML elements and attributes
- Build website using HTML

Lesson 1
Basic of HTML

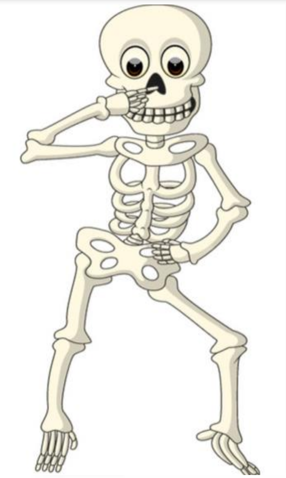
Outline

- The core Web development technologies
- What HTML is and is not.
- The evolution of HTML
- Structure of HTML Document
- Anatomy of HTML elements
- HTML Content Models
- Introduction to developer tools
- Tips
  - *HTML standards documents*
  - *Validation service*
  - *Learning resource*

1. The core Web development technologies

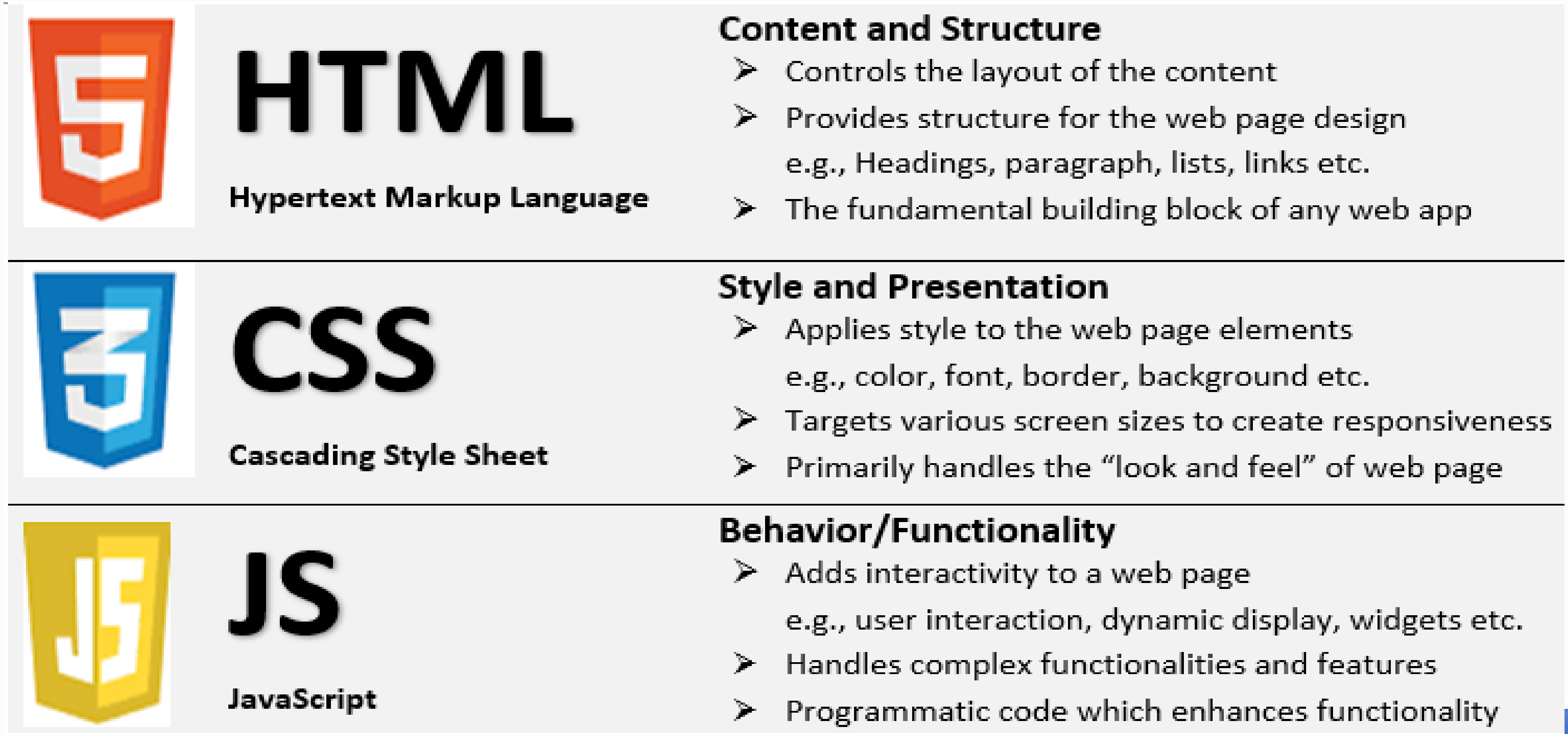
- HTML, CSS and JavaScript are the basic *building block* and core web development technologies.

HTML	CSS	JavaScript
Structure • Crate the structure of the website	**Presentation** • Stylize the layout of the website	**Behavior** • Add interactivity to the website



Cont’d

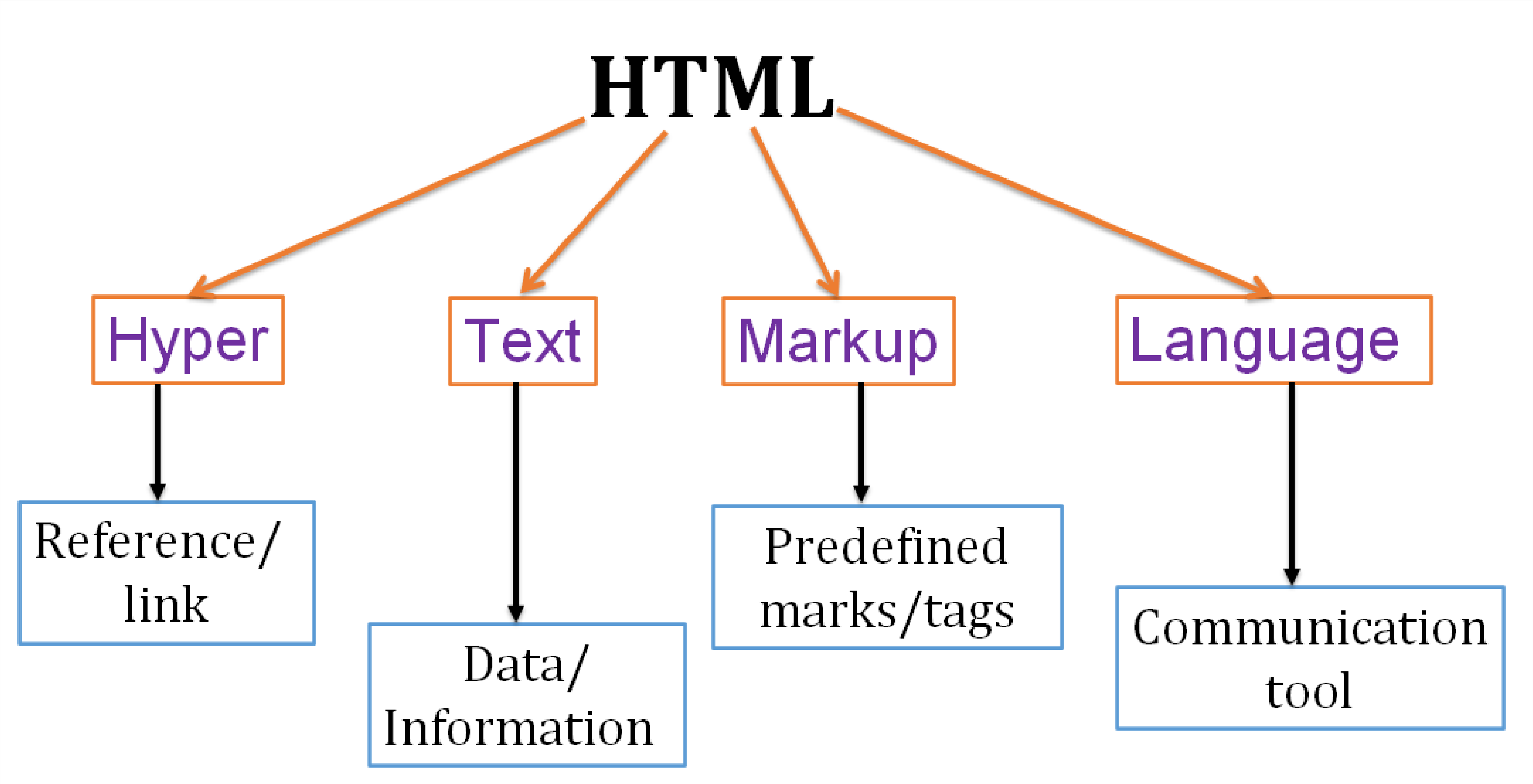
Summary of the difference between HTML, CSS and JS



HTML Hypertext Markup Language	**Content and Structure** ➢ Controls the layout of the content ➢ Provides structure for the web page design e.g., Headings, paragraph, lists, links etc. ➢ The fundamental building block of any web app
CSS Cascading Style Sheet	**Style and Presentation** ➢ Applies style to the web page elements e.g., color, font, border, background etc. ➢ Targets various screen sizes to create responsiveness ➢ Primarily handles the “look and feel” of web page
JS JavaScript	**Behavior/Functionality** ➢ Adds interactivity to a web page e.g., user interaction, dynamic display, widgets etc. ➢ Handles complex functionalities and features ➢ Programmatic code which enhances functionality

2. What HTML is and is not?

- **HTML** Stands for Hypertext Markup Language

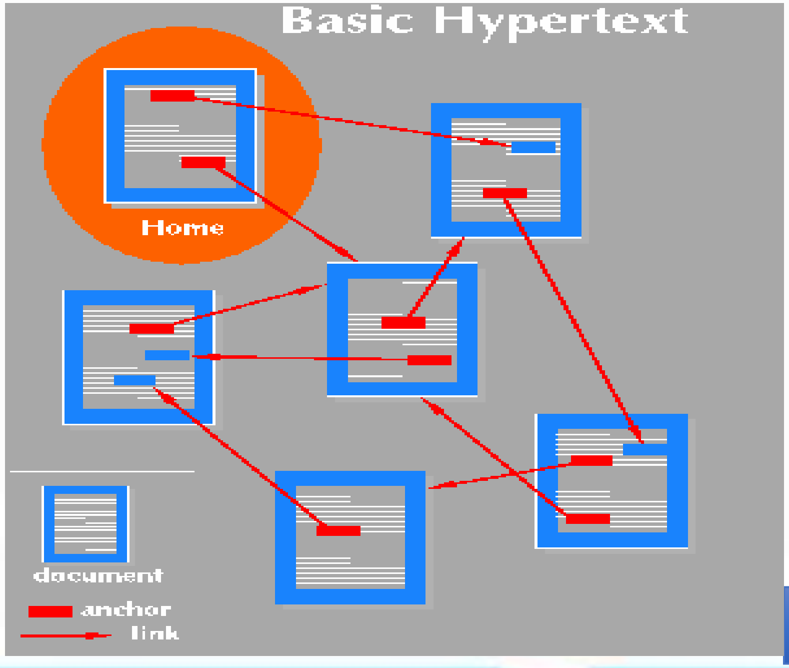


Cont’d

- **Hypertext**
  - Hyper is the opposite of linear
  - Hypertext basically refers to *Non-sequentially linked* pieces of text or other information which is also called “Nodes”.
  - More specifically it is the way in which Web pages (HTML documents) are linked together.

Goal

- Allow to access and read text and other information from multiple perspectives.



Cont’d

- **Markup Language**
  - Markup originally refers to the handwritten indicators on an author’s manuscripts.
  - The idea of markup Notify a typesetter the layout of a document and the typeface to use
  - A markup language is a set of characters or symbols that define a document’s logical structure.
  - In other words, it refers to the use of set of characters within a piece of information that can be used to process or identify that information in a particular way
  - It composed of set of symbols called markup tags.

Cont’d

- **HTML (Hypertext Markup Language)**
  - It is a markup language which based on Standard Generalized Markup Language (SGML).
  - SGML is a standard for specifying a markup language or tag set.
  - SGML Itself is not a document language, but a description of how to specify one and create a document type definition (DTD).
  - It define the structure of information on the Web page.
  - It doesn’t describe the actual presentation of a document
  - It tells the web browser what content to display.
  - Use a pre-defined set of tags to identify the webpage content types .
  - It is not a programming language.

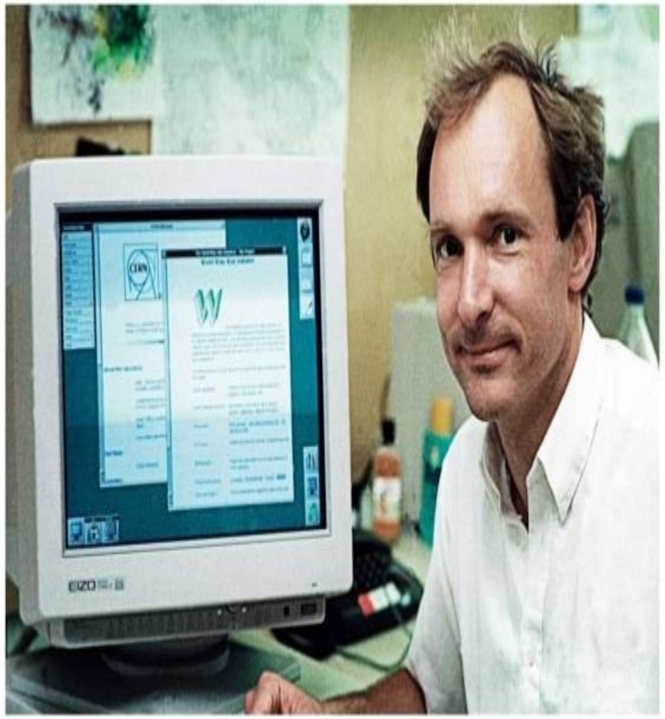
Cont'd

Applications of HTML

- **Web pages development**
  - ➢ HTML is used to create pages which are rendered over the web.
- **Internet Navigation**
  - ➢ HTML provides tags which are used to navigate from one page to another and is heavily used in internet navigation.
- **Responsive UI**
  - ➢ HTML pages now-a-days works well on all platform, mobile, tabs, desktop/laptops.
- **Offline support**
  - ➢ HTML pages once loaded can be made available offline on the machine without any need of internet.
- **Game development**
  - HTML5 has native support for rich experience and is now useful in gaming development arena as well.

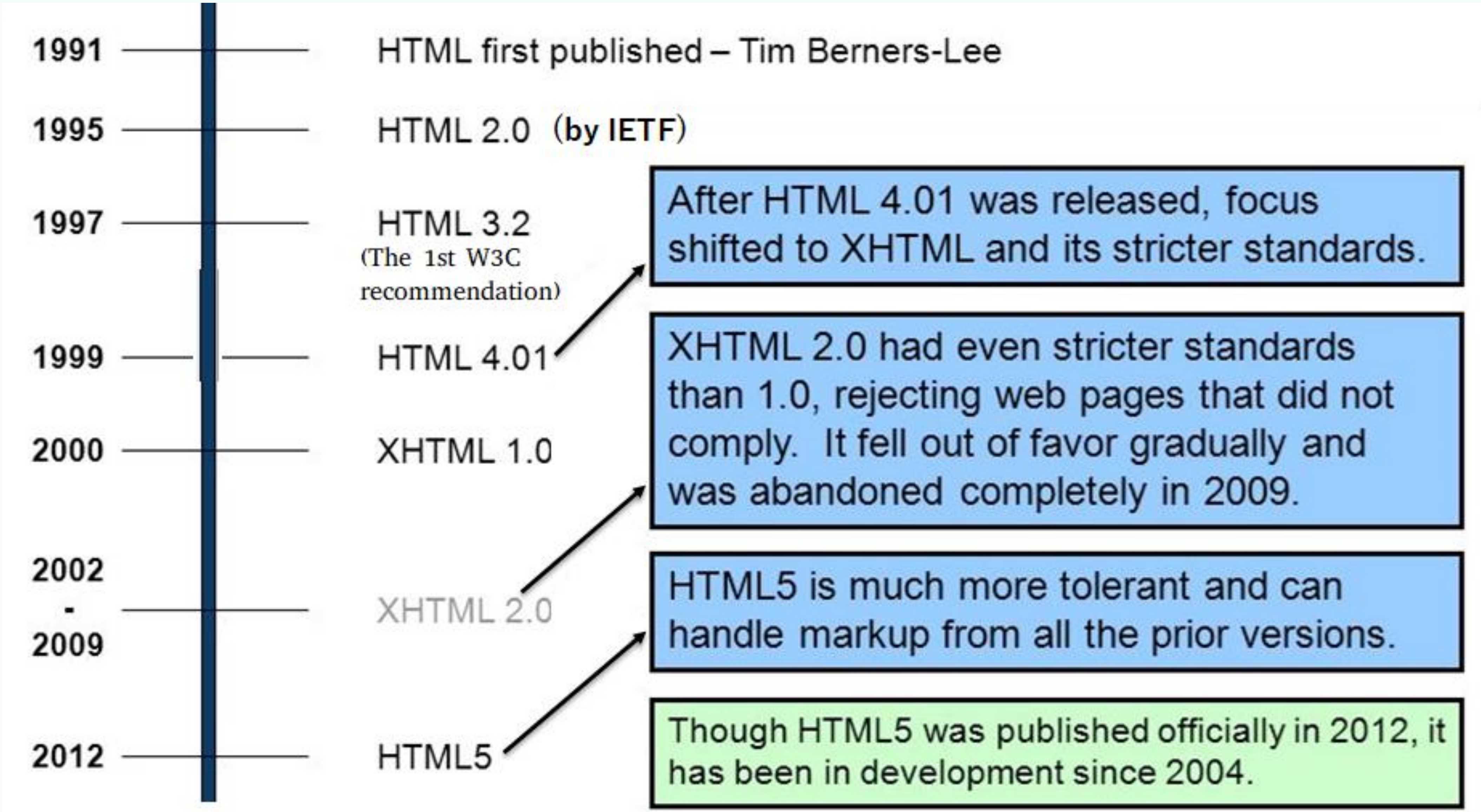
3. The evolution of HTML

- Before 1990 there was no easy way to find, download and view documents over the internet.
- HTML, initially invented by in 1991 by Tim Berners Lee.
- **Tim** invented a system – the web (WWW)
  - ➢ Store documents in one central place (i.e. a web server)
  - ➢ Make it possible to download and view a document with a single click (i.e. a web browser)
  - ➢ Allow to find new documents by clicking on "links" inside other documents.
  - ➢ Instead of **licensing and selling** his idea, he made free for every one



Cont’d

History of HTML



1991 HTML first published – Tim Berners-Lee

1995 HTML 2.0 (by IETF)

1997 HTML 3.2 (The 1st W3C recommendation)

1999 HTML 4.01

2000 XHTML 1.0

2002 - 2009 XHTML 2.0

2012 HTML5

After HTML 4.01 was released, focus shifted to XHTML and its stricter standards.

XHTML 2.0 had even stricter standards than 1.0, rejecting web pages that did not comply. It fell out of favor gradually and was abandoned completely in 2009.

HTML5 is much more tolerant and can handle markup from all the prior versions.

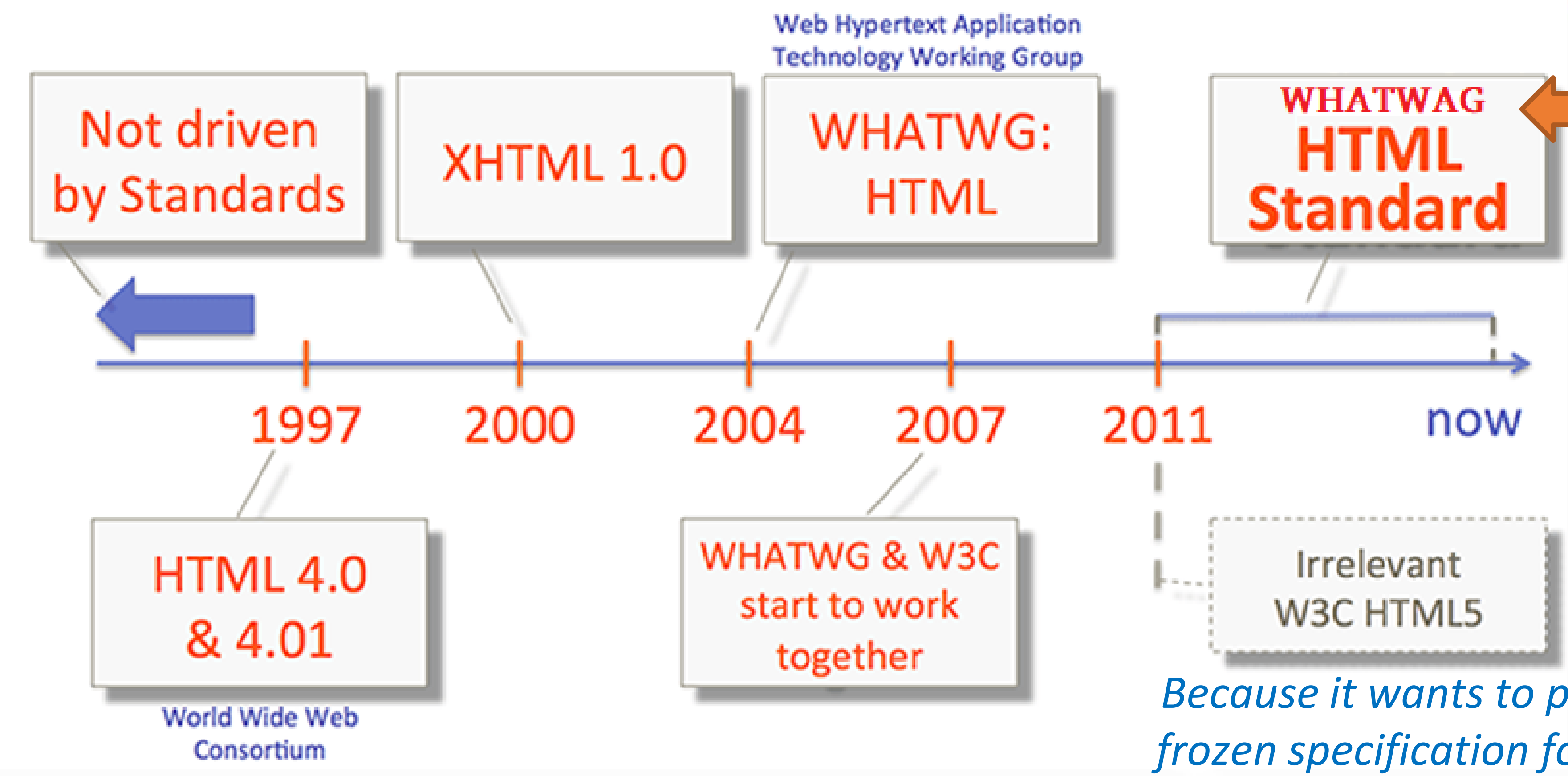
Though HTML5 was published officially in 2012, it has been in development since 2004.

HTML Standardization history

- Before 1997, there were no HTML standards, so browsers basically did whatever they wanted.
- Particularly in the "**War**" Years (1993 – 1997),
  - HTML was being defined by the elements that browser software developers chose to implement.
  - Developers invented new tags or implemented the same tags differently which was kind of the **wild west** of the web.
- Around **1997**, the W3C came up with the first standard that browsers started to pay attention to.
- The W3C defined HTML version 4 (or HTML4) which it shortly thereafter updated to HTML4.01.

Cont’d

HTML Standardization history



Not driven by Standards

XHTML 1.0

Web Hypertext Application Technology Working Group

WHATWG: HTML

WHATWAG HTML Standard

1997

2000

2004

2007

2011

now

HTML 4.0 & 4.01

World Wide Web Consortium

WHATWG & W3C start to work together

Irrelevant W3C HTML5

Continuously evolve the HTML standards and the only exclusively in charge of HTML

Because it wants to publish a frozen specification for HTM5

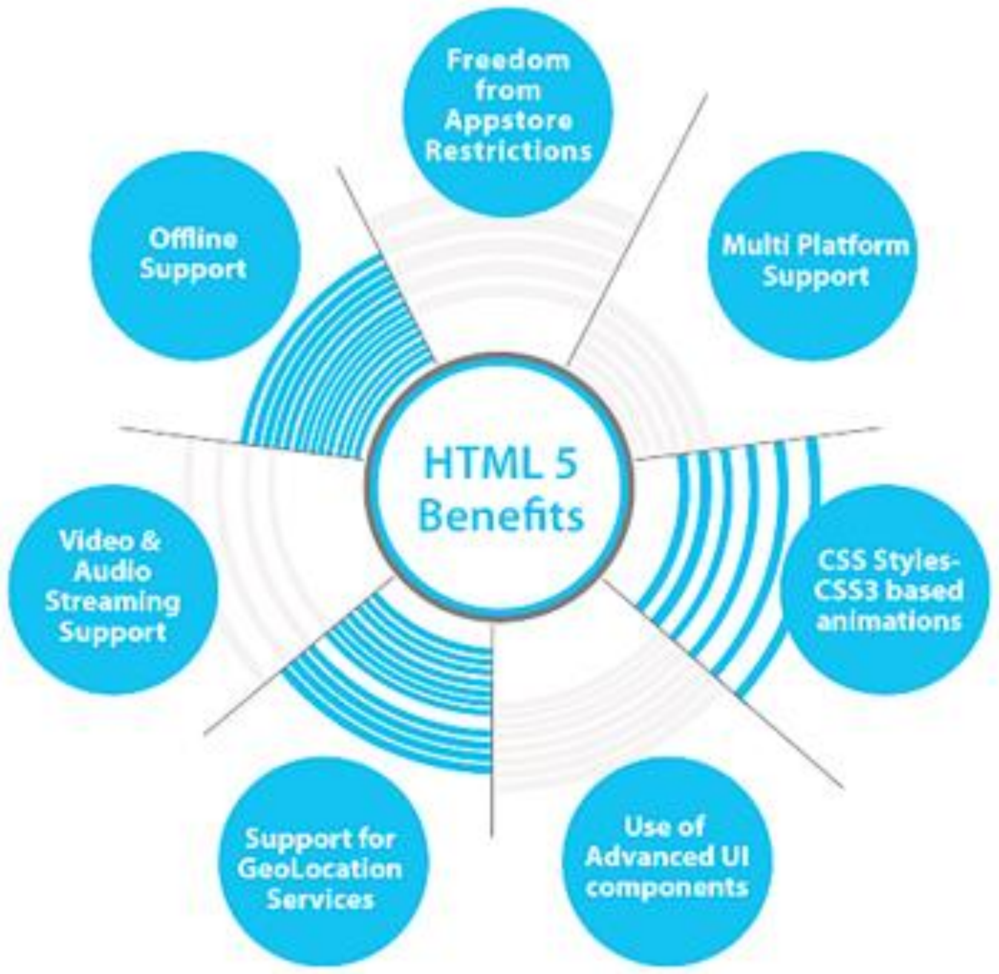
Note:

- **W3C** is the main international standards organization for the Web
- **WHATWG –** established in 2004 by the browser vendors

Cont’d

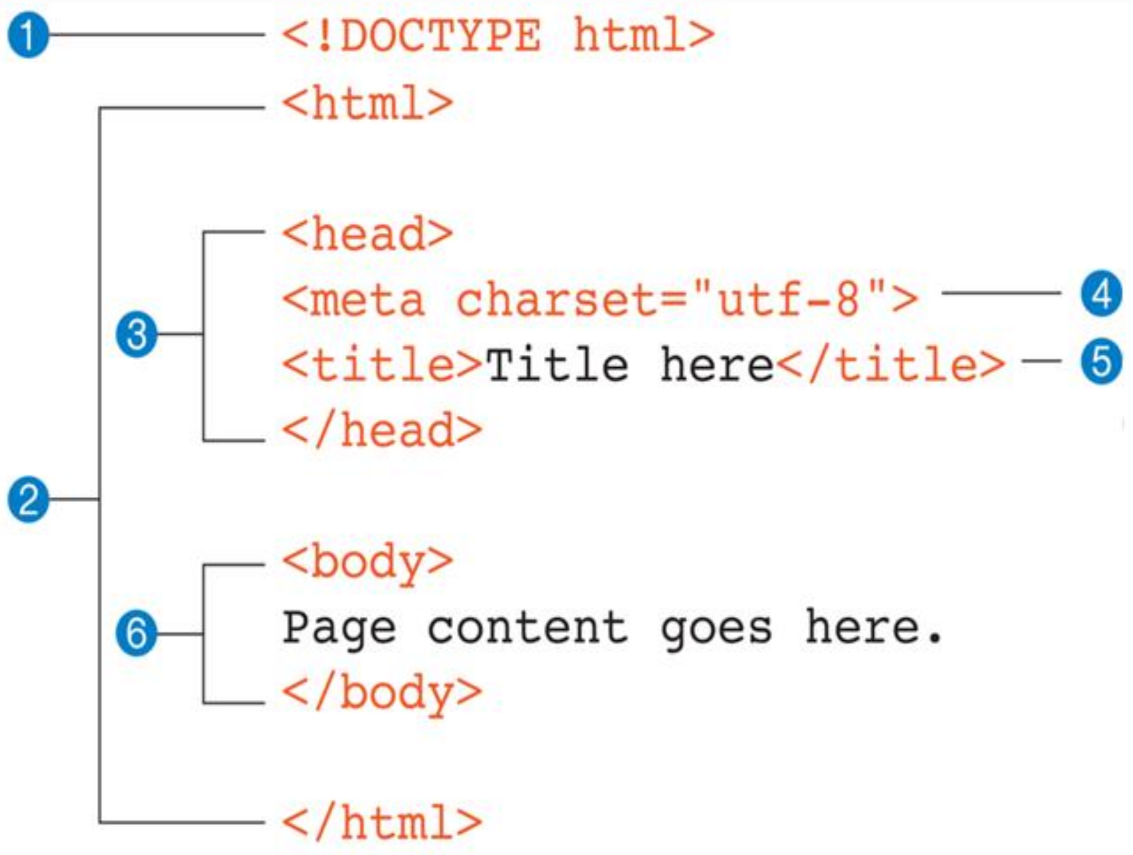
HTML5

- Current version of standard HTML which was defined in 2012 by WHATWAG.
- It is cross-platform.
  - *It will display content on a desktop computer, laptop, a tablet, smartphone, a notebook or a Smart TV.*
- All major browsers support most of the HTML5 elements.
- Backward compatible
- Provide content-specific tags, semantics markups, APIs for emerging idioms


HTML 5 Benefits
Freedom from Appstore Restrictions
Multi Platform Support
CSS Styles- CSS3 based animations
Use of Advanced UI components
Support for GeoLocation Services
Video & Audio Streaming Support
Offline Support

4. Structure of HTML Document

- Basically the HTML document is divided into head section and body Section



```
<!DOCTYPE html>
<html>

<head>
<meta charset="utf-8">
<title>Title here</title>
</head>

<body>
Page content goes here.
</body>

</html>
```

1. DOCTYPE declaration

- ➢ *Identifies the document and tell the browser the type of HTML standard*
- ➢ *At this point, it let the browsers to interpret the document that it written as per HTML5 standard.*
- ➢ *Must appear first*

2. Root element

- ➢ *Embed all the elements in the document*

3. Head element

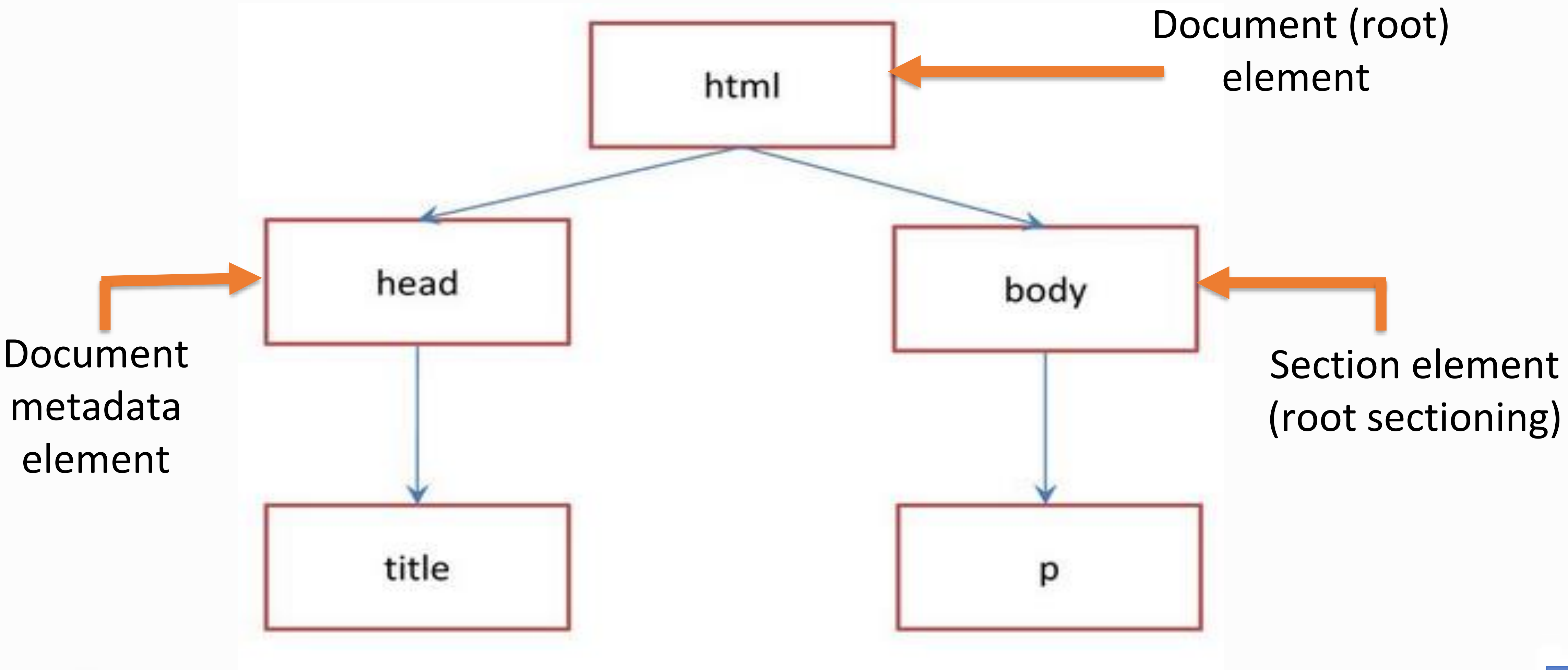
- ➢ *Contains descriptive information about the document itself.*
- ➢ *These are non web displayable info*

6. Body element

- ➢ *Contains everything that we want to show up (display) in the browser window.*

Cont'd

Tree structure of HTML Document

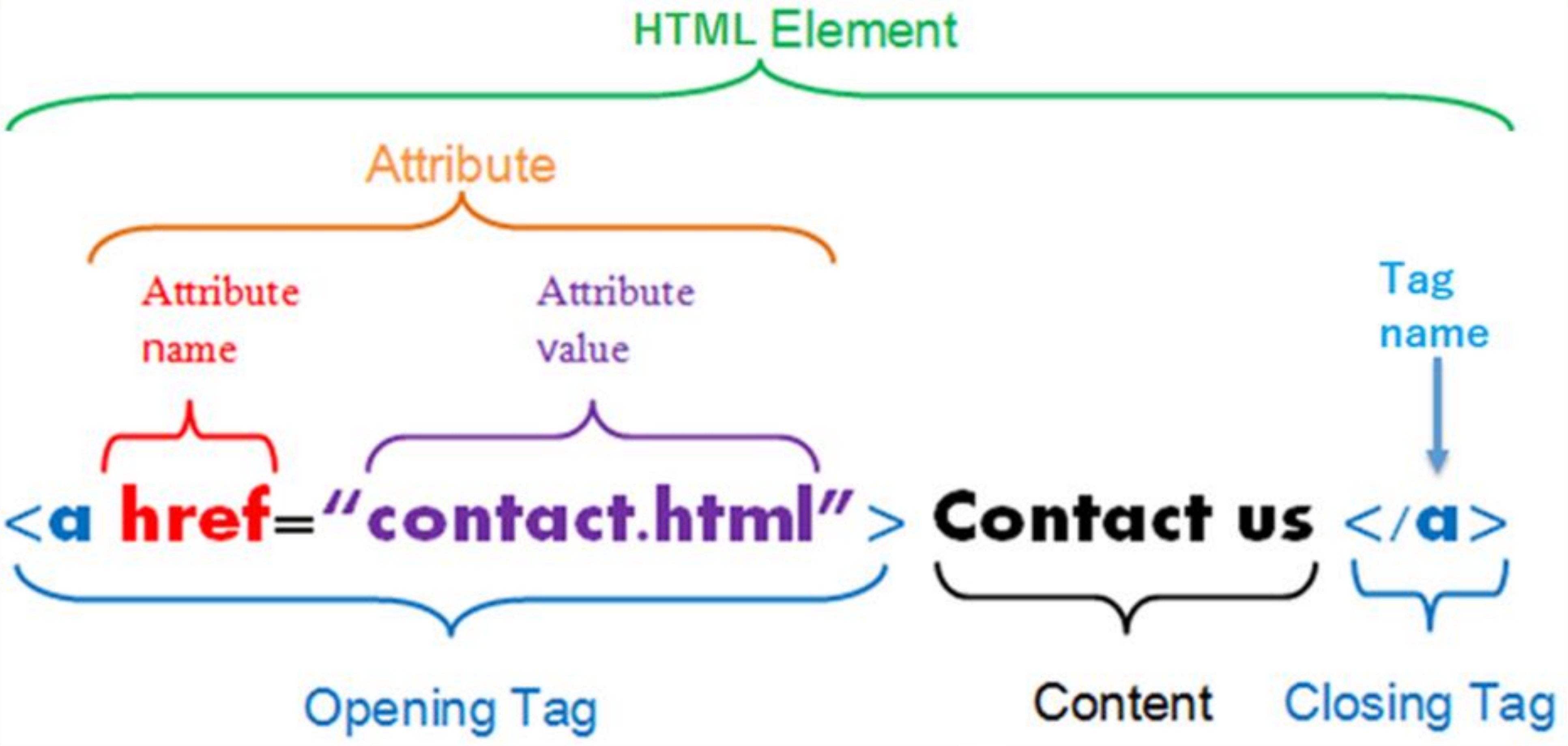


Document (root) element

Document metadata element

Section element (root sectioning)

5. Anatomy of HTML Elements



- **Basic Syntax of HTML element:** <tag attribute = “Value”> Content </tag>

Cont'd

- **HTML elements**
  - Represent some kind of structure
  - It is a combination of a tag and its character data (content)
  - HTML can be *Empty element* or ***Nested element*** also

(a) Nested Element

- An element that contain other HTML elements

(b) Empty Element

- An element with no character data (content)
- Also called *Non-container tags* or *self closing tags*
- Syntax: **<self-closing-tag-name/>**

Cont’d

HTML Tags

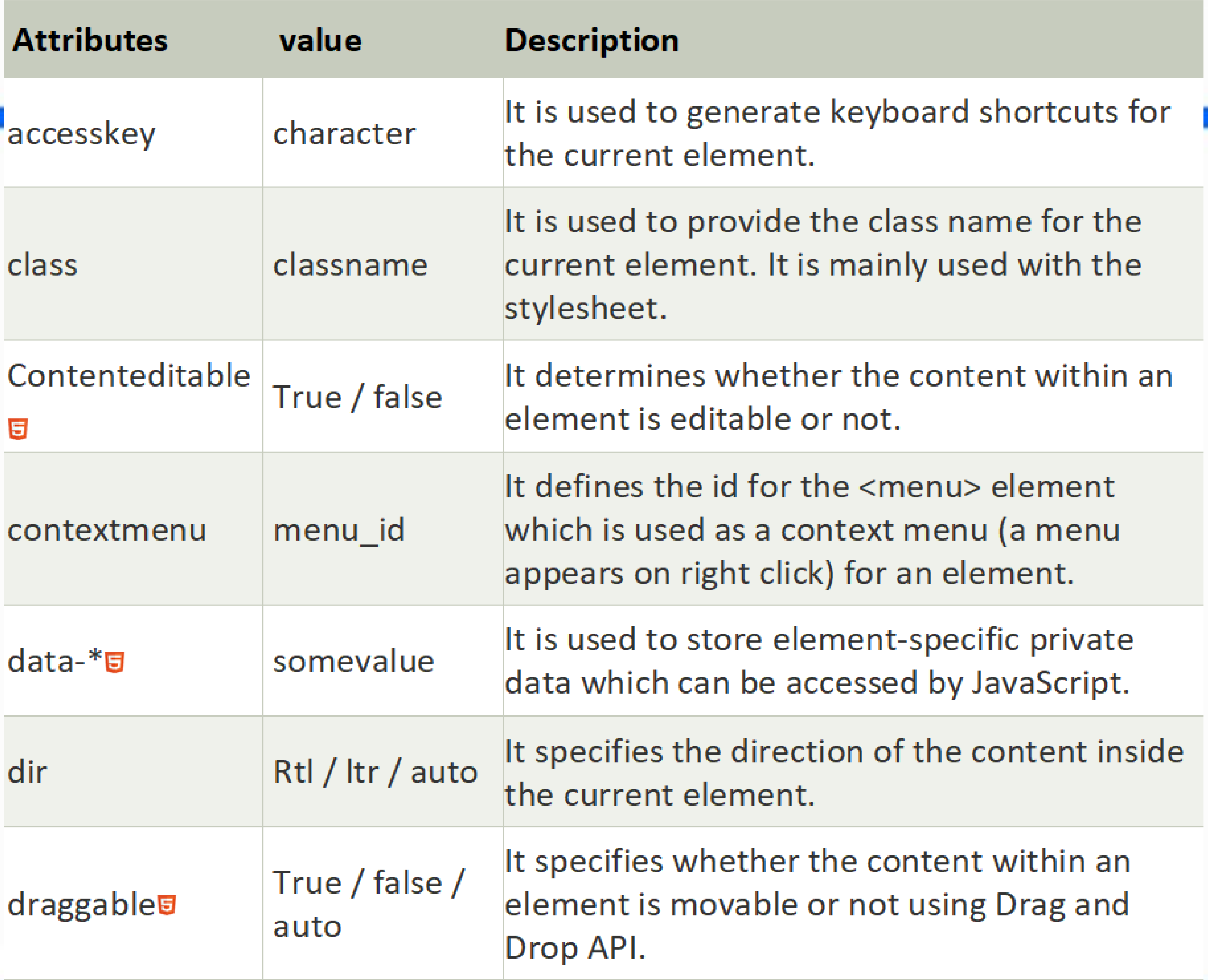
- Used to markup the text (content)
- Most of the tags come in as start and end tag pairs
- However, empty element tags doesn’t come in pairs
- The start tag may contain an attribute.

HTML attributes

- Defines a property for an elements
- It is a way to describe the tags
- It must enclosed within the start tag.
- It comes in a name/value pairs
- The value should enclosed within double quote
- An element can have one or more attributes
- Global attributes – an attributes that are common to all HTML elements

Cont’d

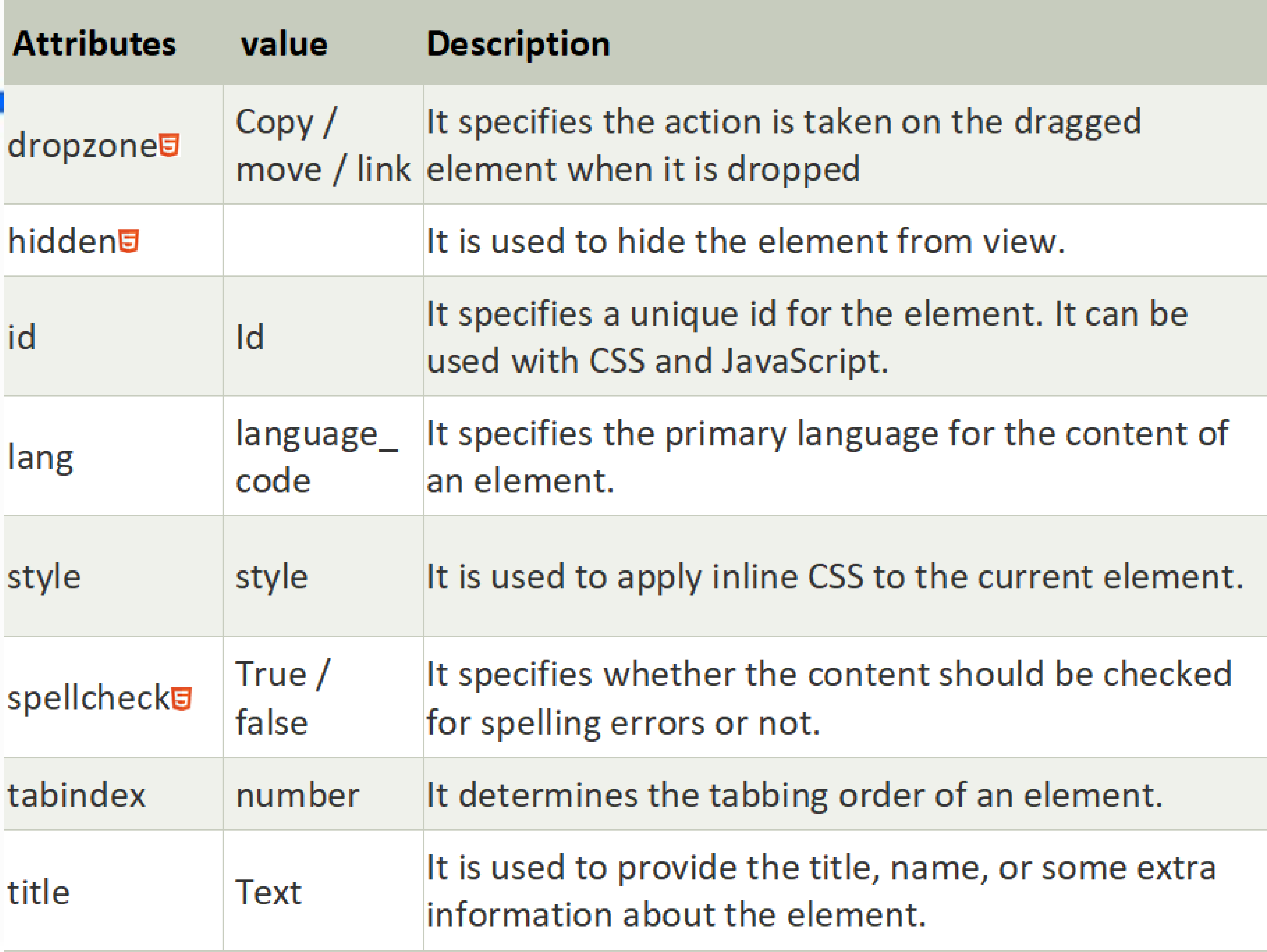
HTML Global Attributes



Attributes	value	Description
accesskey	character	It is used to generate keyboard shortcuts for the current element.
class	classname	It is used to provide the class name for the current element. It is mainly used with the stylesheet.
Contenteditable	True / false	It determines whether the content within an element is editable or not.
contextmenu	menu_id	It defines the id for the <menu> element which is used as a context menu (a menu appears on right click) for an element.
data-*	somevalue	It is used to store element-specific private data which can be accessed by JavaScript.
dir	Rtl / ltr / auto	It specifies the direction of the content inside the current element.
draggable	True / false / auto	It specifies whether the content within an element is movable or not using Drag and Drop API.

Cont’d

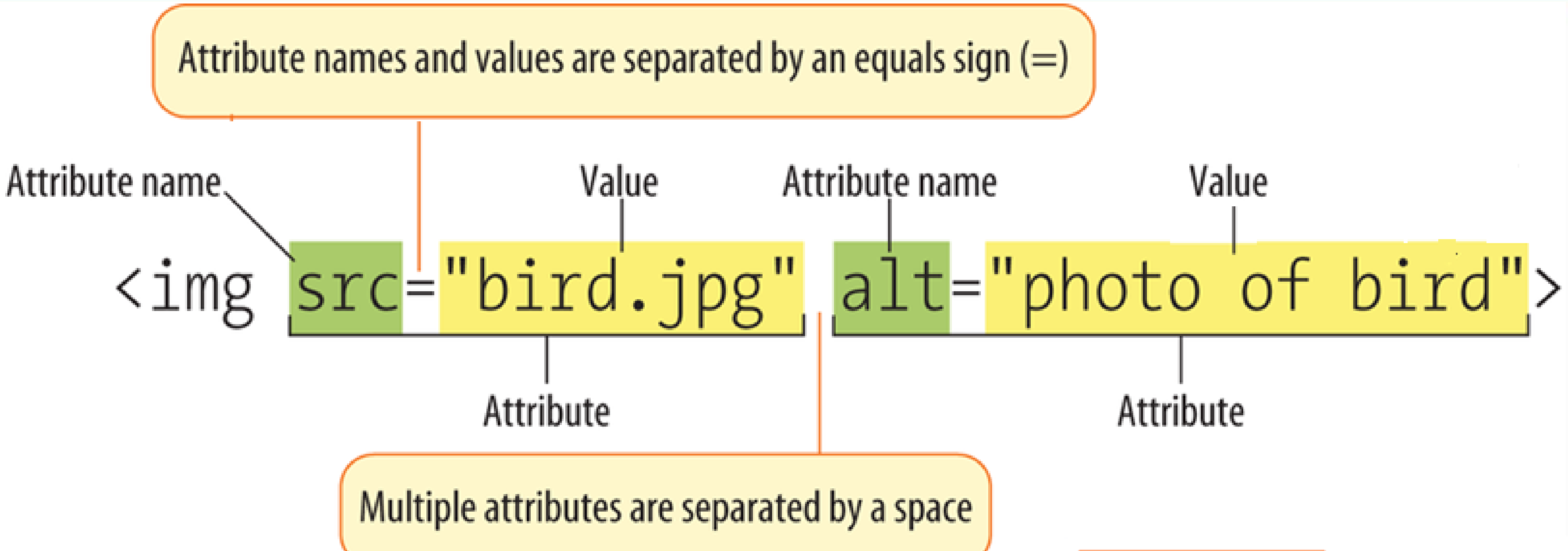
HTML Global Attributes



Attributes	value	Description
dropzone 5	Copy / move / link	It specifies the action is taken on the dragged element when it is dropped
hidden 5		It is used to hide the element from view.
id	Id	It specifies a unique id for the element. It can be used with CSS and JavaScript.
lang	language_ code	It specifies the primary language for the content of an element.
style	style	It is used to apply inline CSS to the current element.
spellcheck 5	True / false	It specifies whether the content should be checked for spelling errors or not.
tabindex	number	It determines the tabbing order of an element.
title	Text	It is used to provide the title, name, or some extra information about the element.

Cont’d

Simple rule



Attribute names and values are separated by an equals sign (=)

Attribute name

Value

Attribute name

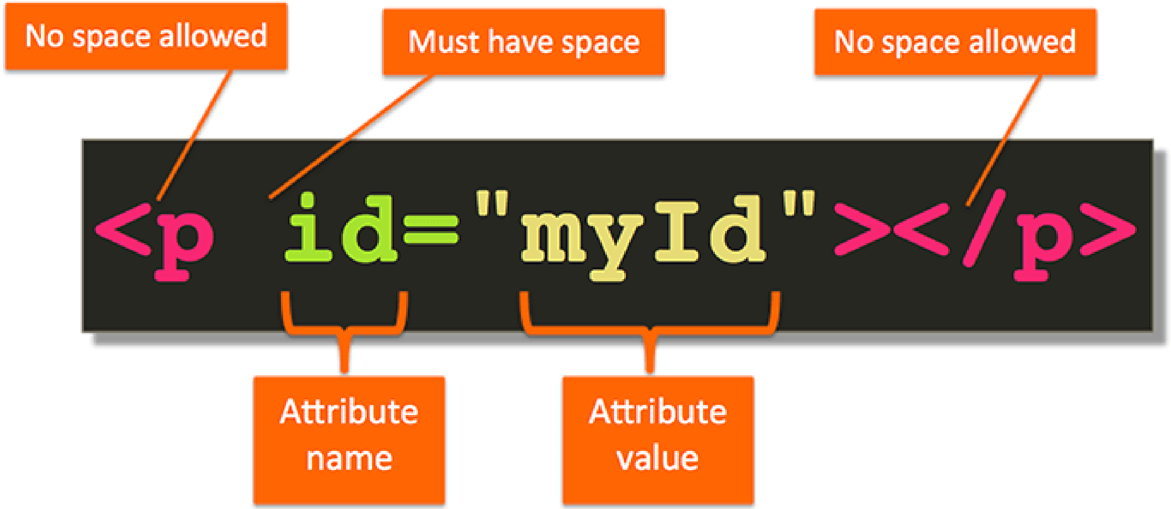
Value

```
<img src="bird.jpg" alt="photo of bird">
```

Attribute

Attribute

Multiple attributes are separated by a space



No space allowed

Must have space

No space allowed

```
<p id="myId"></p>
```

Attribute name

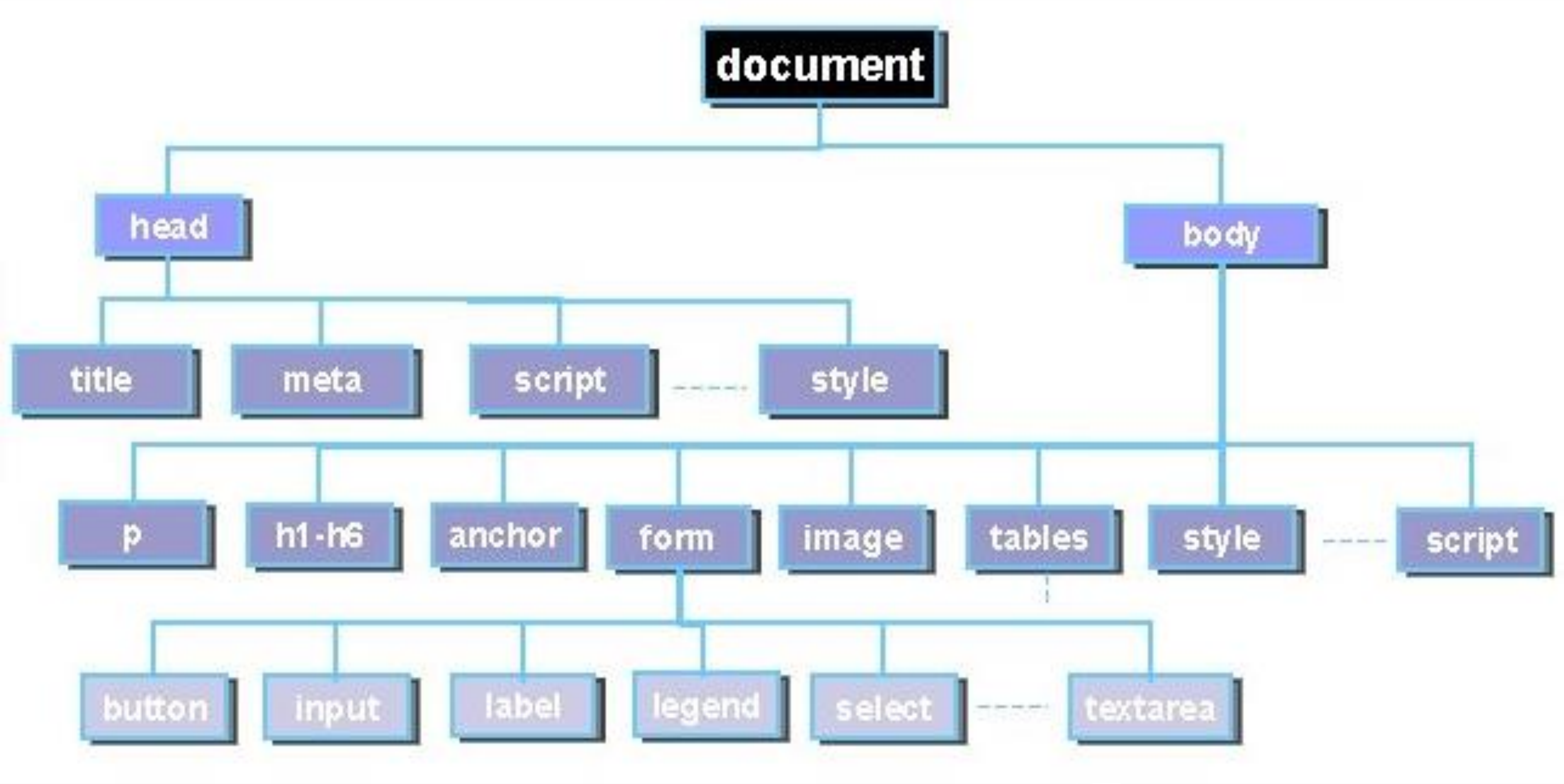
Attribute value

Note:

- *Don’t forget the End tag.*
- *Even though HTML is case insensitive use lowercase tags*

Cont’d

Basic HTML document elements hierarchy - a showcase



Cont’d

HTML elements reference

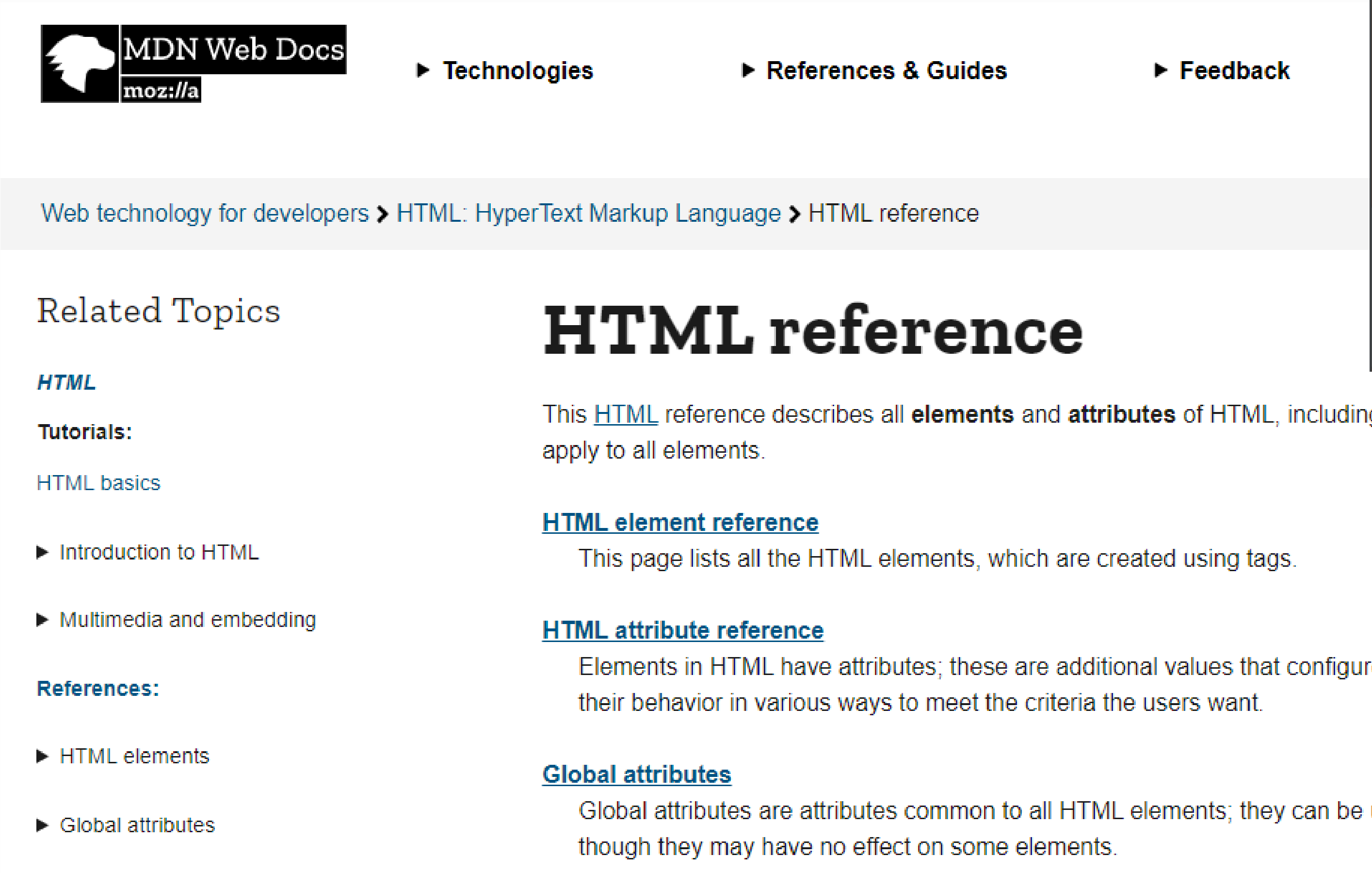
- The HTML elements are grouped by function to help you find what you have in mind easily.
- As per the HTML Living Standard 2021, the elements are grouped as follow

- Document metadata
- Sectioning root
- Content sectioning
- Text content
- Inline text semantics
- Image and multimedia
- Embedded content
- SVG and MathML
- Scripting
- Demarcating edits
- Table content
- Forms
- Interactive elements
- deprecated elements

Note: *For the details of elements under each category just* ***ctrl + click*** *on the element category.*

Cont’d

For detail and further refer check out



MDN Web Docs moz://a

▶ Technologies ▶ References & Guides ▶ Feedback

Web technology for developers › HTML: HyperText Markup Language › HTML reference

Related Topics

HTML

Tutorials:

HTML basics

▶ Introduction to HTML

▶ Multimedia and embedding

References:

▶ HTML elements

▶ Global attributes

HTML reference

This HTML reference describes all **elements** and **attributes** of HTML, including **global attributes** that apply to all elements.

HTML element reference
This page lists all the HTML elements, which are created using tags.

HTML attribute reference
Elements in HTML have attributes; these are additional values that configure the elements or adjust their behavior in various ways to meet the criteria the users want.

Global attributes
Global attributes are attributes common to all HTML elements; they can be used on all elements, though they may have no effect on some elements.

6. Content Model in HTML

- Content Model
  - Refers to a description of the element's expected contents
  - **Set of rules** that define what type of content each element is allowed to have.
  - Also describe what other elements are allowed to be nested inside which other elements.
- Prior to the **modern HTML** specification, HTML elements were either **block-level or inline elements**.
- Modern HTML specification split these two content models into ***seven basic models.***
- However, in practical terms, you can still think of those seven models as falling into the same traditional categories: *block-level and inline elements*.

Block-level elements Vs. Inline elements

Block-level Elements

- By default, Block-level elements render (displayed) to begin on a new line (i.e. the browser will automatically place the contents of that element on a new line in the flow of the document.

Inline Elements

- Render on the same line by default.
- Having new line characters in the content or between the tags in your code won’t make any difference.
- All white space characters get translated into a single space anyway and the whole bunch of inline elements still be displayed on the same line.
- Inline elements are restricted to only contain other inline elements (i.e. doesn’t contain block-level element)

Cont’d

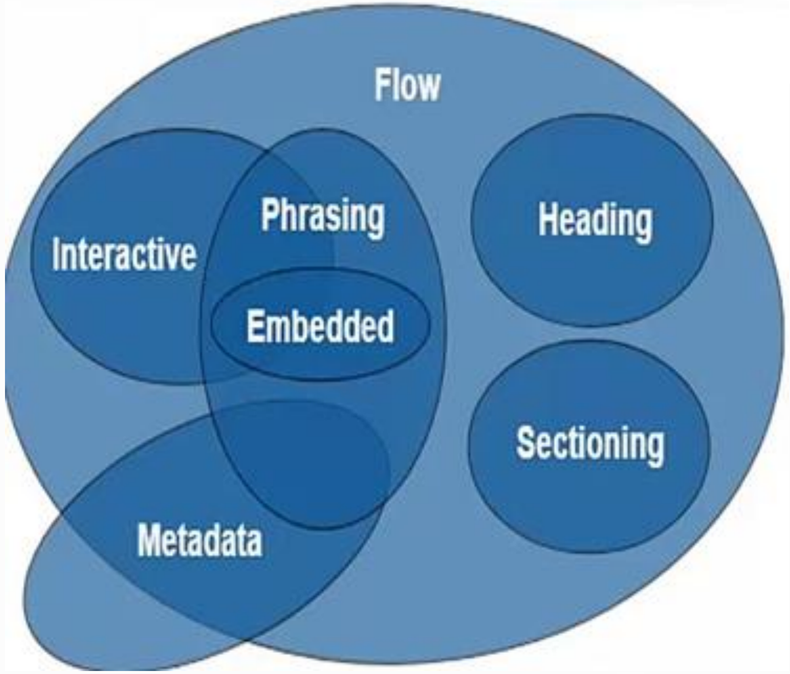
Modern HTML Content Models

- Current HTML5 speciation categories and group elements that share common characteristics.
- This is a loose grouping (it doesn't actually create a relationship among elements of these categories), but they help define and describe the categories' shared behavior and their associated rules.
- It's also possible for elements to not be a member of any of these categories.
- The main types of content categories are:
  - Main content categories – also called *kind of content models and describe common rules shared by many elements*
  - Form-related content categories
  - Specific content categories – also called *Transparent content* model

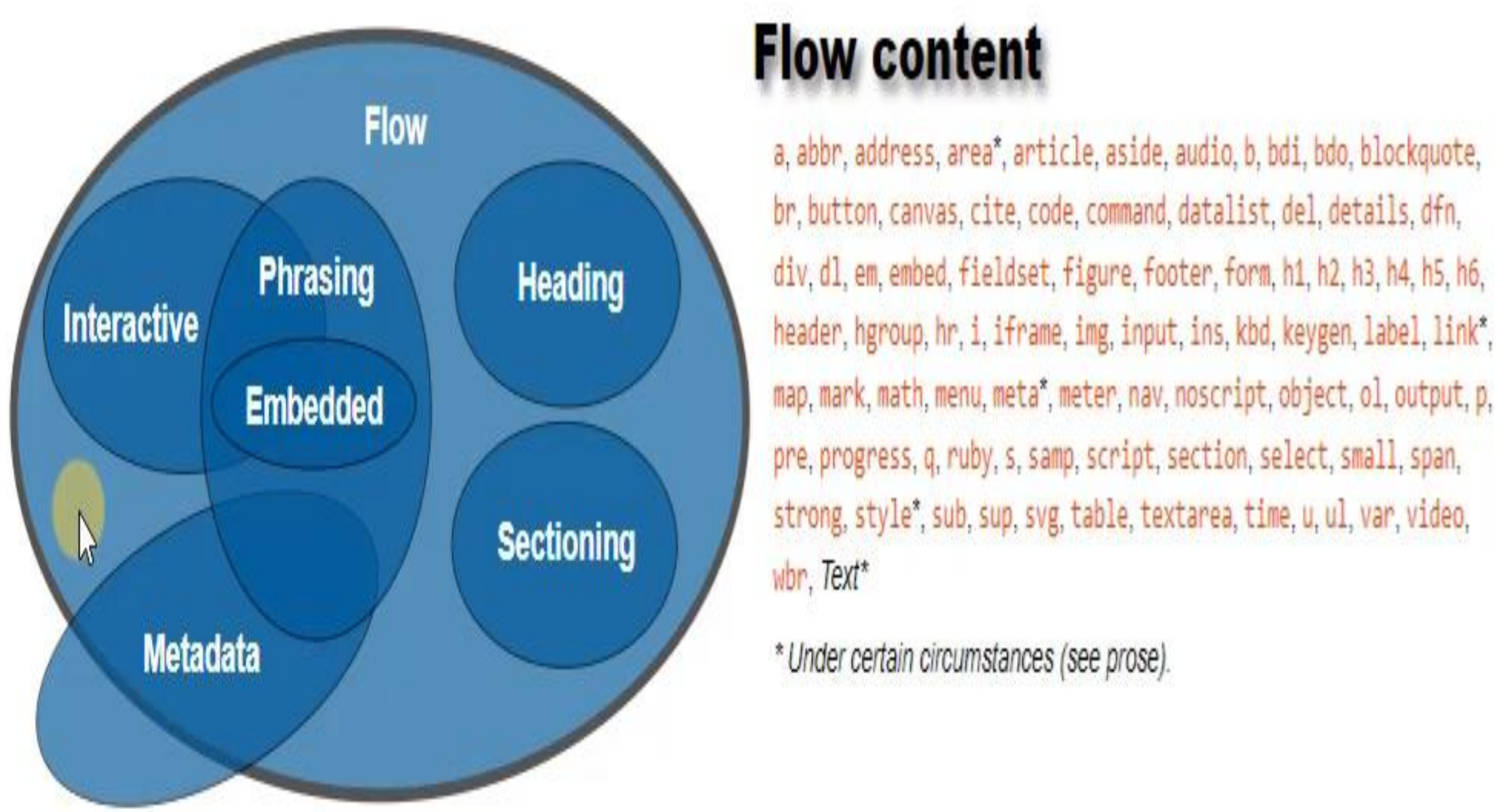
Cont'd

Main Content Models

- Each elements in HTML5 falls into zero or more categories that group elements with similar characteristics.
- The following content models are broad categories that are used in the current HTML specification:
  - Some elements are **overlaps** and the subset content models can be used everywhere flow content is expected.
  - Certain elements are categorized as form-associated elements and further subcategorized to define their role in various form-related processing models.
  - Some elements have unique requirements and do not fit into any particular category.



Kind of Content



Flow content

a, abbr, address, area*, article, aside, audio, b, bdi, bdo, blockquote, br, button, canvas, cite, code, command, datalist, del, details, dfn, div, dl, em, embed, fieldset, figure, footer, form, h1, h2, h3, h4, h5, h6, header, hgroup, hr, i, iframe, img, input, ins, kbd, keygen, label, link*, map, mark, math, menu, meta*, meter, nav, noscript, object, ol, output, p, pre, progress, q, ruby, s, samp, script, section, select, small, span, strong, style*, sub, sup, svg, table, textarea, time, u, ul, var, video, wbr, *Text**

* *Under certain circumstances (see prose).*

Cont’d

Summary of Content Models

No	Content Model	Description
1	Flow content	➢ A broad category that encompasses most elements that can go inside the <body> element
2	Headlining content	➢ Defines the title of a section, whether marked by an explicit sectioning content element, or implicitly defined by the heading content itself
3	Section content	➢ Create a section in the current outline that defines the scope of <header> elements, <footer> elements, and heading content
4	Meta content	➢ Modify the presentation or the behavior of the rest of the document, set up links to other documents etc.
5	Interactive content	➢ Includes elements that are specifically designed for user interaction
6	Phrasing content	➢ Defines the text and the markup it contains
7	Embedded content	➢ Imports another resource or inserts content from another mark-up language or namespace into the document,
8	Form-related content	➢ Comprising elements that have a form owner, exposed by a form attribute, and can be used everywhere flow content is expected

6. Browser Developer Tools

- Every modern web browser includes a powerful suite of developer tools.
- Also called **web development tools** or **devtools or inspect element**
- These tools do a range of things, from inspecting currently-loaded HTML, CSS and JavaScript to showing which assets the page has requested and how long they took to load
- With developer tools, we can directly interact with the source code that is fetched into the client side of our system.
  - Allow to understand the underlying structure of the code
  - Enables us to ***inspect*** the web content
  - Allow to test drive front-end changes,
  - Check site's responsive design, and even optimize its performance

Cont’d

- These are the tools that are browser dependent.
  - Come as browser add-ons or built-in features in web browsers.
  - Most popular web browsers, such as *Firefox, Google Chrome, Internet Explorer, Safari, Microsoft Edge and Opera* have built-in tools
  - Moreover, many additional **add-ons** can be found in their respective plugin download centers.
- Across all browsers, the basic functionality of the developer tools remains the same, only naming conventions and terminologies changes.
- For details how to use the developer tools refers to the link below under the Tools and setup section.
  - https://developer.mozilla.org/en-US/docs/Learn/Common_questions

Cont'd

How to find the developer tool?

Option 1:

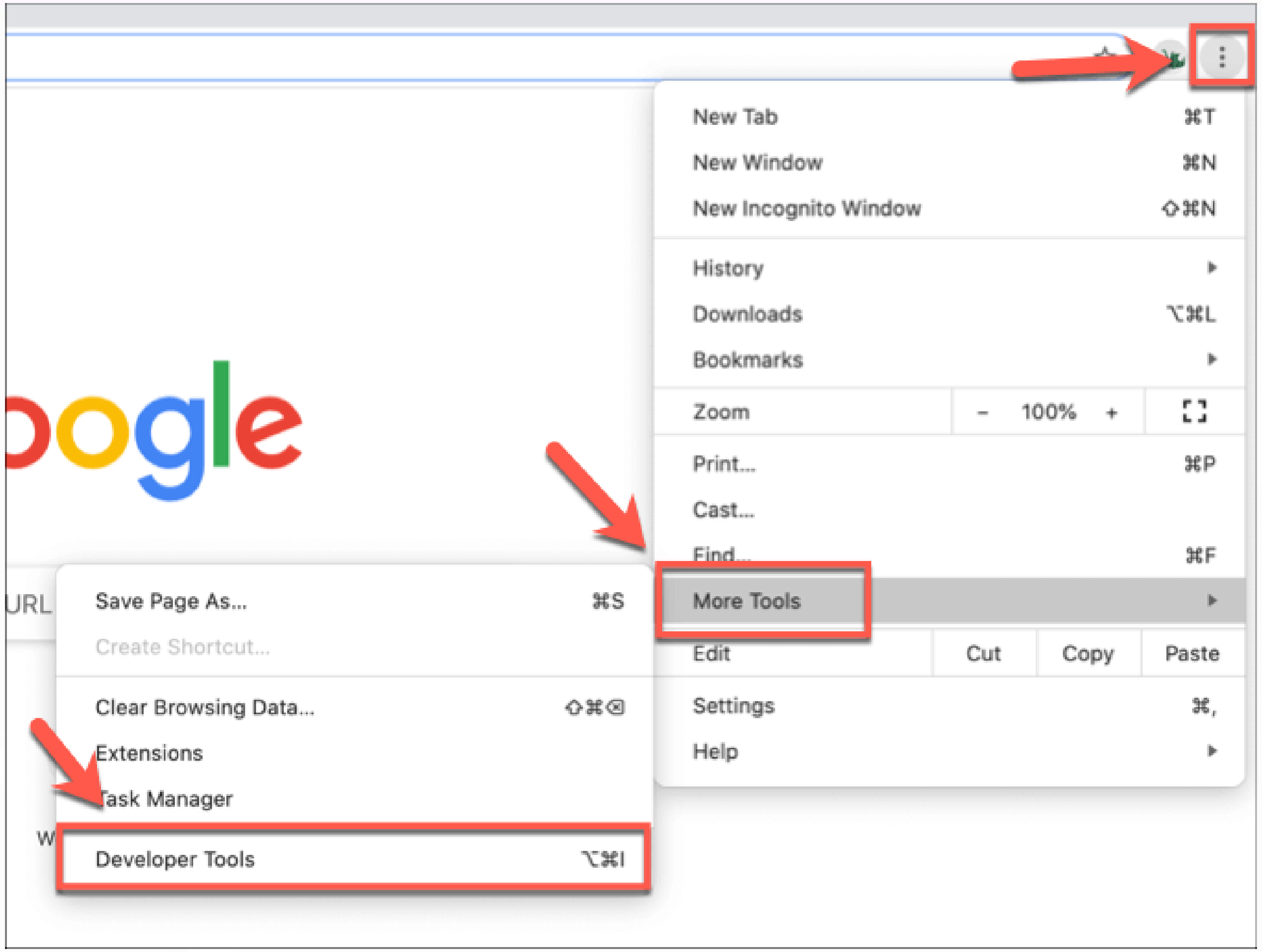
- Open the browser **Menu** in the upper-right-hand corner of the browser window
- Select *More Tools -> Developer Tools*

Option 2:

- Right click on the page and select *“inspect”*

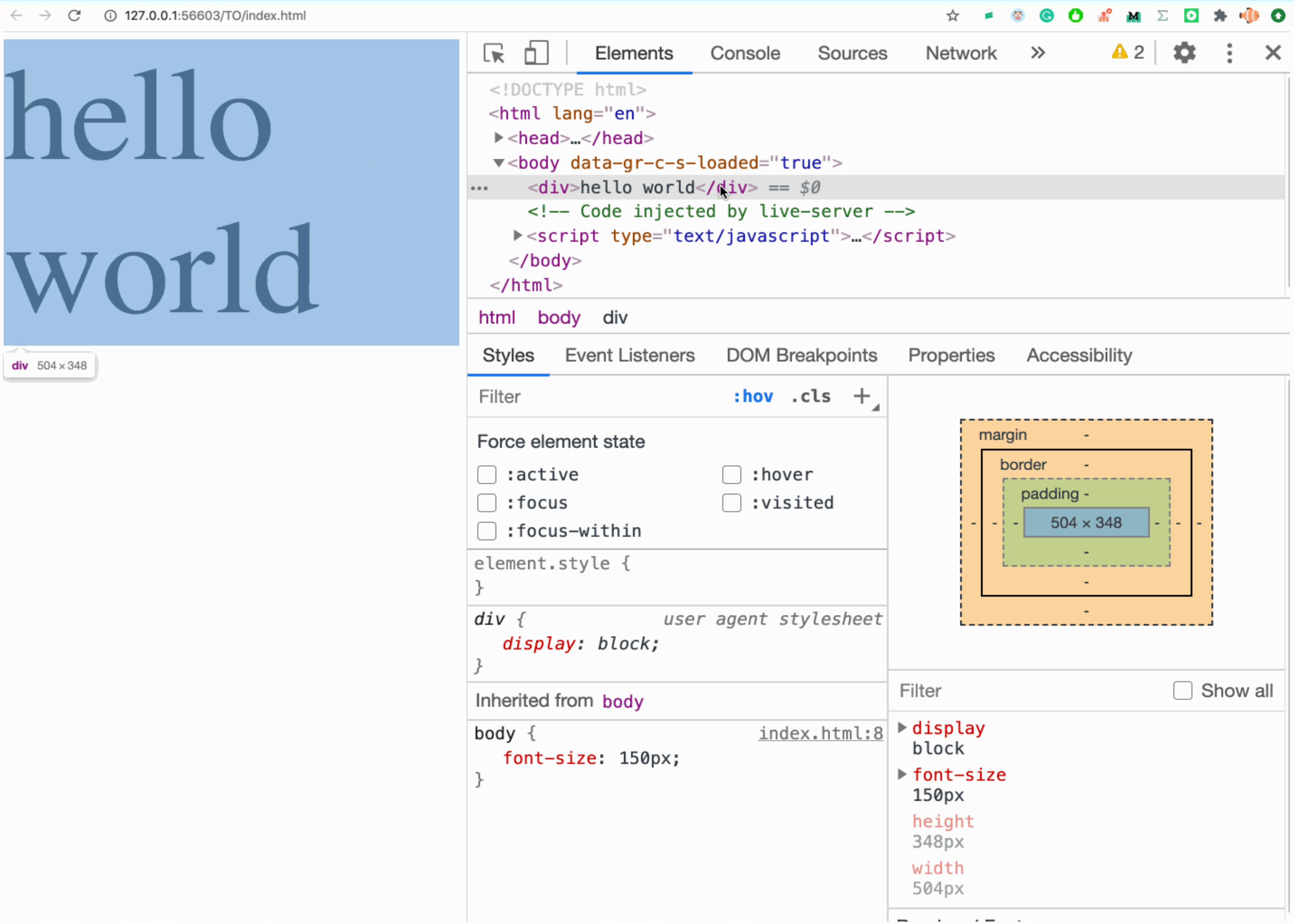
Option 3:

- Key shortcut



Option + ⌘ + J (on macOS), or Shift + CTRL + J (on Windows/Linux)

Cont’d The developer tool Components



```
<!DOCTYPE html>
<html lang="en">
  ▶<head>…</head>
  ▼<body data-gr-c-s-loaded="true">
...     <div>hello world</div> == $0
      <!-- Code injected by live-server -->
    ▶<script type="text/javascript">…</script>
    </body>
</html>
```

Force element state

- ☐ :active
- ☐ :hover
- ☐ :focus
- ☐ :visited
- ☐ :focus-within

```
element.style {
}
div {                    user agent stylesheet
   display: block;
}
Inherited from body
body {                   index.html:8
   font-size: 150px;
}
```

☐ Show all

```
▶ display
  block
▶ font-size
  150px
  height
  348px
  width
  504px
```

- Elements
- Resources
- Scripts
- Timeline
- Profiles
- Storage
- Audit
- Console

Tips

HTML standards documents

1. HTML Living Standard
   - https://html.spec.whatwg.org/multipage/#toc-semantics
2. A vocabulary and associated APIs for HTML and XHTML (W3C Working Draft, 2011)
   - http://www.w3.org/TR/2011/WD-html5-20110525/
3. MDN Web Docs - Resources for developers, by developers
   - https://developer.mozilla.org/en-US/

Validation service

- https://validator.w3.org/

Tips

Learning resources

1. W3Schools
   - https://www.w3schools.com/html/default.asp
2. Code Academy
   - https://www.codecademy.com/
3. FreeCodeCamp
   - https://www.freecodecamp.org/
4. Tutorials to Night
   - https://www.tutorialstonight.com/

5. MOOCs
   - HTML CSS JavaScript course for Web Developers by Coursera
   - Basics of Web Development by Coursera
   - Zero to Mastery course by Udemy

Summary

- HTML, CSS and JavaScript are the core web development technologies
- HTML is a markup language that used to define the structure of web document
- HTML uses tags (characters that sit inside angled brackets) to give the information they surround special meaning.
- HTML document mainly consist the “head” and “body” section.
- HTML elements are comprise tags and character data (content)
- Tags usually come in pairs. The opening tag denotes the start of a piece of content; the closing tag denotes the end.
- HTML attributes describe the content of the element and require a name and a value.

Summary

- HTML initially invented by in 1991 by Tim Berners Lee and standardized since 1997.
- HTML5 is the current web development standard
- Content model refers to **set of rules** that define what type of content each element is allowed to have.
- Prior to HTML5 the elements were either **block-level or inline elements**.
- Modern HTML specification (HTML5) split these two content models into ***seven basic models*** and each elements in HTML5 falls into zero or more categories that group elements with similar characteristics
- Browser developer tool are browser dependent inspection tools that come as browser add-ons or built-in features in web browsers.
- Developer tools allow to understand the underlying structure of the code and also to ***inspect*** the web content

Chapter Activity

1. Differentiate Hypertext, hyperlink and hypermedia.
2. What are other types of Markup Languages? List and discuss their differences.
3. What are the main elements that define the HTML document?
4. Why W3C and WHATWAG are not able continue to collaborate?
5. Why content models are important? Briefly discuss the common seven main content model categories.
6. Discuss the use of developer tool components.

Thank You
For Your Attention!!