HTML and CSS

Why do we need both HTML and CSS?

HTML ⇒ Structure

CSS ⇒ Style

History of HTML

- 1989 Tim Berners-Lee, CERN Lab (Switzerland)
  - How to share research papers
- 1991: WWW Talk presentation of HTML
- 1992: Dave Raggett HTML+
- Marc Andreessen National Center for Supercomputer Applications
  - Mosaic browser (1993)
- 1995 Release of **Java (by Sun Microsystem) & Javascript (by Netscape**

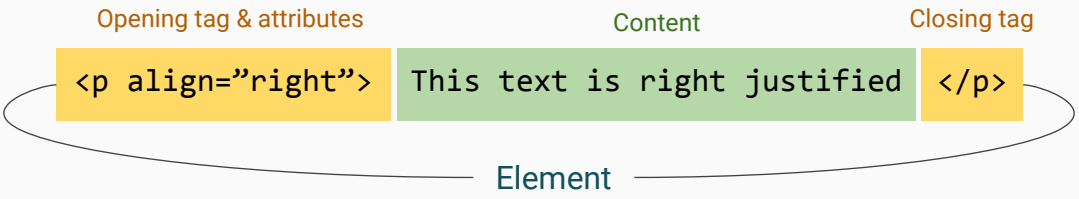
HTML Versions

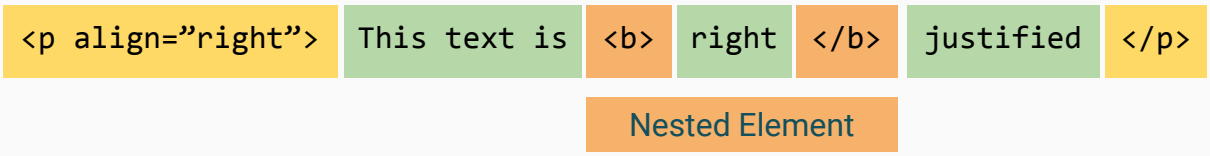
- 1991 HTML (20 elements)
- 1995 HTML 2.0
- 1997 HTML 3.2
- 1999 HTML 4.01 (First release of CSS)
- 2000 XHTML (HTML + XML)
- 2014 HTML5

HTML

- Hypertext Markup Language
- **Not** a programming language, but a *presentation* language
- Code for **structuring** and **displaying** a web document
- Vocabularies
  - Tags
  - Attributes
  - Elements

HTML Elements





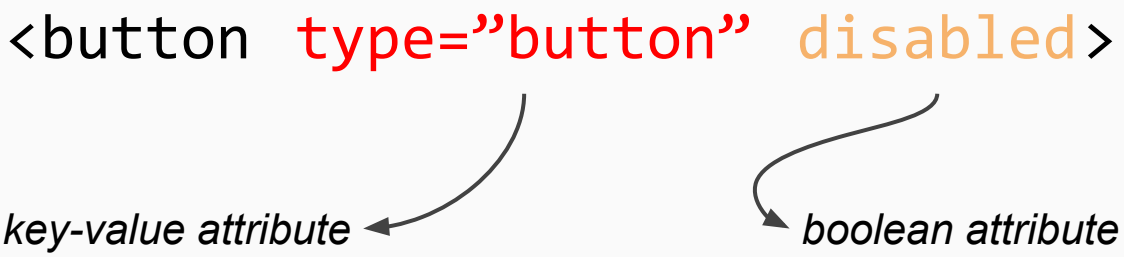
Empty Elements

- No closing tag
- No content

```
<link rel=”stylesheet” href=”.....”>
```

```
<img src=”http://my.host.org/path/to/your/image.jpg”>
```

HTML Attributes



Type of Elements

Two categories of HTML elements based on their effect on **new line**

- **Block**: induces a new line **before** and **after** its surrounding context
  - Examples: `p, h[1-6], ol, ul, pre, blockquote, dl, div, form, hr, table`
- **Inline**: contained within block-level elements without starting newlines
  - Examples: `b, i, tt, code, em, strong, a, br, img, span, sub, sup, button, input, select`

HTML Tags

- Alphabetical List
- Categorical List

Overall Structure of HTML Documents

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Sample Page</title>
  </head>

  <body>
    <h1>Welcome</h1>
    <p>Hello world!</p>
  </body>
</html>
```

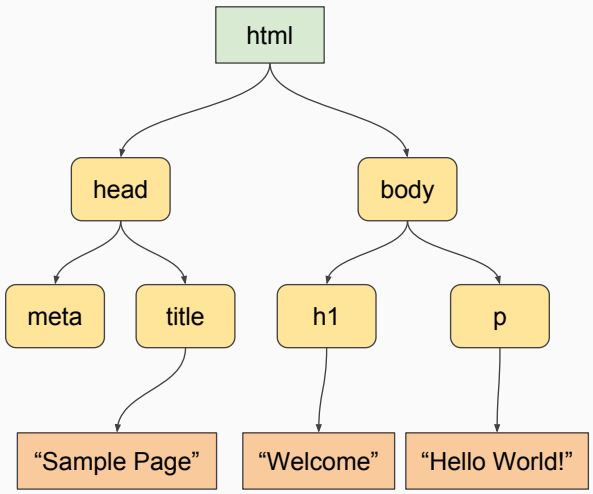
*parsed by web browsers, but **not rendered***

***parsed and rendered** by web browsers*

HTML Documents and DOM Tree

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Sample Page</title>
  </head>

  <body>
    <h1>Welcome</h1>
    <p>Hello world!</p>
  </body>
</html>
```



Document Object Model

<meta>

- Meta tags are used for “machine readable” information about the document
  - Document character encoding
  - Content Description for web bots
- App specific metadata that can be used to customize embedding of web content into another (app)
  - Facebook
  - Twitter
  - Pinterest
  - Slack, and many more...

Comments and Special Characters

```
<!-- this is a comment -->
```

Special Character	Encoding
<	<
>	>
&	&
“	"
‘	'

Element id & class

```
<table id=”taxbracket” >

<img id=”logo” class=”decor” >
```

Within a single page

- No two elements can have the same id
- Multiple elements may belong to the same class
- In CSS: `#taxbracket` `#logo` `.decor`

Styling in CSS

- Cascading Stylesheet
- Styles are defined using a set of rules
- Each rule
  - begins with a selector to select the element(s) onto which the rule is applied
  - Specify a group of properties to apply to the element(s)

```
selectorA {
    property1: value;
    property2: value;
}

selectorB {
    property1: value;
    property2: value;
}
```

```
p {
  margin: 4px;
  color: white;
  background-color: black;
}
```

- Selector: paragraphs
- Properties
  - 4-pixel margin
  - White text on black background

```
#sidebar {
  background-color: gray;
}
```

- Selector: element with id “sidebar”

```
.active {
  font-size: 120%;
}
```

- Selector: element with class “active”

Defining Styles

- Internal Stylesheet: **rules** are written in the same file as the HTML doc
  - Use `<style> </style>` in header
- External Stylesheet: **rules** are written in a file separate from the HTML doc
  - Use `<link rel="stylesheet" href="...." type="text/css">` in header
- ~~Inline Style: properties written as the `style` attribute of an element~~ (**not recommended**)
  - `<p style="color: red">.....</p>`

Class Exercise

HTML

- Hypertext Markup Language
- More lenient syntax validation
  - Some elements may omit end tags
  - Empty elements
- Content Type: text/html
- <html> must not include **xmlns** attribute

XHTML

- eXtensible HTML
- Must comply with XML syntax
  - Must include <!**DOCTYPE**>
  - Tags and attributes must be in **lowercase**
  - All elements must have an opening tag and a closing tag
  - Shortcut for **empty** elements
  - All values must be in **quotation marks**
- Content Type: text/html, text/xml, application/xhtml+xml, application/xml
- <html> must declare the **xmlns** attribute

HTML

```
<html>
  <head>
    <title>Sample Page</title>
  </head>
  <BODY>
    <p ALIGN=left>Hello
  </BODY>
</html>

<--  valid HTML but invalid XHTML -->
```

XHTML

```
<!DOCTYPE html>
<html xmlns=”http://www.w3.org/1999/xhtml”>
  <head>
    <title>Sample Page</title>
  </head>
  <body>
    <p align=”left”>Hello</p>
  </body>
</html>
```

Document Type Definitions (DTDs)

- An HTML document can be written in compliance with one of the HTML specifications (HTML 3.2, HTML 4.01, HTML 5, …)
  - Must inform the browser which version a document uses
- <!DOCTYPE…> declaration is the **first line** of an HTML document
  - It is not an HTML tag
- A DTD file defines the proper syntax of an XML document
  - Valid tags, valid attributes per tag
  - Hierarchical relationships among tags
  - Example of DTDs

DTDs (HTML 4 and HTML 5)

```
                                                    <!-- HTML 5 -->
<!DOCTYPE html>
<html> . . . </html>
```

```
                                                <!-- HTML 4.01 Strict -->
<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
<html> . . .  </html>
```

```
                        <!-- HTML 4.01 Transitional (allow deprecated elements) -->
<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html> . . . </html>
```

```
                <!-- HTML 4.01 Frameset (like Transitional but allow frameset) -->
<!DOCTYPE HTML PUBLIC
  "-//W3C//DTD HTML 4.01 Frameset//EN" "http://www.w3.org/TR/html4/frameset.dtd">
<html> . . . </html>
```

DTDs (XHTML 1.0 and XHTML 1.1)

```
                                                    <!-- XHTML 1.1 -->
<!DOCTYPE html PUBLIC
  “-//W3C//DTD XHTML 1.1//EN”  “http://www.w3.org/TR/xhtml1/DTD/xhtml11.dtd”>
```

```
                                                <!-- XHTML 1.0 Strict -->
<!DOCTYPE html PUBLIC
  “-//W3C//DTD XHTML 1.0 Strict//EN”
  “http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd”>
<html> . . . </html>
```

```
                                            <!-- XHTML 1.0 Transitional -->
<!DOCTYPE html PUBLIC
  “-//W3C//DTD XHTML 1.0 Transitional//EN”
  “http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd”>
```

```
                                                <!-- XHTML 1.0 Frameset -->
<!DOCTYPE html PUBLIC
  “-//W3C//DTD XHTML 1.0 Frameset//EN”
  “http://www.w3.org/TR/xhtml1/DTD/xhtml1-frameset.dtd”>
```

XHTML Empty Elements

Empty elements can be opened and closed using a single tag, by including **a slash at the end**

With start and end tags	Short Cut
` </br>`	` `
`<img src=”....” alt=”....”> </img>`	`<img src=”....” alt=”....” />`
`<link rel=”....” href=”....”> </link>`	`<link rel=”....” href=”....” />`

XML Namespace (xmlns)

- Every XML document is required to declare its namespace
- XML **namespaces** are like Java **packages**, they resolve ambiguity for conflicting classnames
  - **java.util**.Timer
  - **javax.management.timer**.Timer
  - **javax.swing**.Timer

Google AMP

- Accelerated Mobile Pages
  - **Goal**: make mobile webapps as speedy as native apps
- AMP = alternative version of a web page designed for mobile browsing
  - AMP HTML (subset of HTML)
  - AMP CSS
  - AMP JavaScript