

# HTML overview

Web development I: Front-end engineering

GML → SGML → XML → HTML

Document type declaration

Markup vs Content

Tag vs Element

Attributes

Encoding

Character entities (escaping)



# HTML vs XML

## XML: eXtensible Markup Language

Focused on describing *data*

Dynamic: Transport purposes

Any tag is possible

Case-sensitive syntax

Must be well formed

Type safety and verbosity

## HTML: HyperText Markup Language

Focused on describing *documents*

Static: Display purposes

Predefined set of tags

Case-insensitive syntax

May have errors

# What browsers ignore

Multiple white spaces

Line breaks

Tabs

Unrecognized markup

Anything inside comments

No doctype implies “quirks mode”

Block vs inline elements

- Rendering implications: `width` and `margin`
- Any non-standard tags are considered inline

BLOCK-LEVEL ELEMENTS:



INLINE ELEMENTS:



JULIA EVANS  
@b0rk

## default stylesheets

every browser has a  
default stylesheet  
(aka "user agent stylesheet")

this is in Firefox's  
default stylesheet:

```
h1 {  
  font-size: 2em;  
  font-weight: bold;  
}
```

different browsers  
have different defaults



buttons & forms  
have some of the  
biggest differences

CSS that you write  
always overrides the  
browser's styles

```
h1 {  
  font-size: 2.5em;  
}
```

↑  
this overrides the  
default font size

you can read the  
default stylesheet

Firefox's default  
stylesheets are at:

`resource://gre-resources/`

every property also has  
a default "initial value"

the initial value (defined  
in the spec) is what's  
used if no stylesheet has  
set anything. for example,  
background-color's initial  
value is transparent

"CSS reset" stylesheets  
reduce browser  
inconsistencies

they'll often set default  
line heights, heading font  
sizes, margins, etc.

Provide **meaning** to documents

- Examples: `section`, `main`, `header`, `footer`, `aside`, `nav`

Microdata: <https://www.w3.org/TR/microdata/>

Microformats: <https://microformats.io/>

# Empty elements

Were introduced to HTML by mistake: presentational markup crept into the language

May or may not have attributes

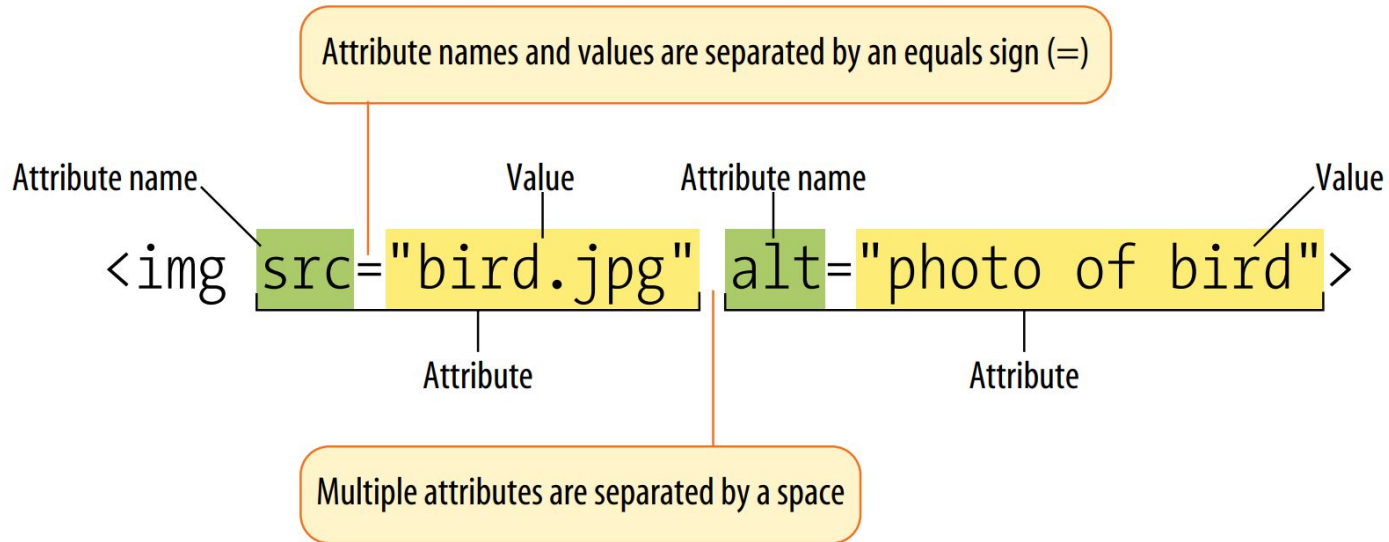
Examples: image, line break, rule, meta tags

Good practice: close them explicitly (e.g. `<br />` instead of `<br>`)

[https://www.w3.org/TR/xhtml1/#C\\_2](https://www.w3.org/TR/xhtml1/#C_2)



# Attributes



# Attributes

Order does not matter but repetition does

Some can be empty: `<input type="text" disabled />`

Some can omit quotes: `<input value=yes />`

Custom attributes are prefixed with `data-`: `<input data-hello="101" />`

Special semantics: `id` (unique identifier) and `class` (classifier)

# Common doctypes (and DTDs)

## HTML 4.01 (strict)

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

## XHTML 1.1

```
<?xml version="1.0" encoding="UTF-8"?>  
  
<!DOCTYPE html PUBLIC  
    "-//W3C//DTD XHTML 1.1//EN"  
    "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
```

## HTML5

```
<!DOCTYPE html>
```

# Character entities

Some characters are reserved, e.g. `<`, `>`, `&`, `"`

Escaping sequence is `& + word + ;` or `&# + number + ;` or `&#x + hex + ;`

`&lt;` is displayed as `<`

`&#169;` is displayed as ©

`&#x000A2;` is displayed as ¢

See <https://dev.w3.org/html5/html-author/charref>

# How many HTML tags are there?

“144 distinct tag names, out of which 28 are deprecated.”

— <https://developer.mozilla.org/en-US/docs/Web/HTML/Element>