



JÖNKÖPING UNIVERSITY

School of Engineering

HTML

Peter Larsson-Green

Jönköping University

Autumn 2018

HYPertext MARKUP LANGUAGE

Is not a programming language.

Syntax:

```
<tag>
```

```
<tag>Text</tag>
```

```
<tag attribute="value">Text</tag>
```



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

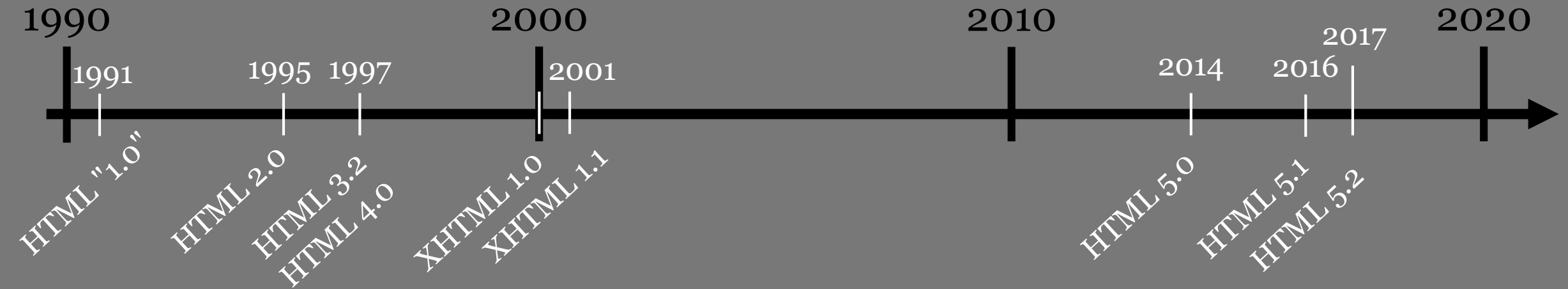
```
<tag attribute='value'>Text</tag>
```

```
<tag attributeA="value" attributeB="value">Text</tag>
```



```
<tag attributeB="value" attributeA="value">Text</tag>
```

VERSIONS



HTML 5.2 Specification: <https://www.w3.org/TR/html52/>

XHTML VS HTML

XHTML is stricter → less mistakes by programmers.

- XHTML needs to be valid XML.
 - `<tag>` is not allowed; must close tag, e.g.: `<tag />`
 - `<tag attribute=value>` is not allowed; must use single/double quotes.
 - Tags must be closed in right order. Wrong: `<i>Word</i>`
- Tag & attribute names must be lowercase in XHTML.

HELLO WORLD

HTML 5
doctype.

Meta data
(data about the data)

Actual data

```
• <!DOCTYPE html>
<html>
  <head>
    <title>Hello World Example</title>
  </head>
  <body>
    <h1>Hello World!</h1>
    <p>Hello World in HTML.</p>
  </body>
</html>
```

SOME <HEAD> ELEMENTS

```
<title>Hogwarts</title> • • •
```

Tabs,
bookmarks.

```
<meta charset="UTF-8">
```

```
<meta name="author" content="Helga Hufflepuff">
```

```
<meta name="description" content="Hogwarts School of Witchcraft and Wizardry">
```

```
<meta name="keywords" content="magic, quidditch, school">
```

Search engines!

Check it out: <https://ju.se/en.html>

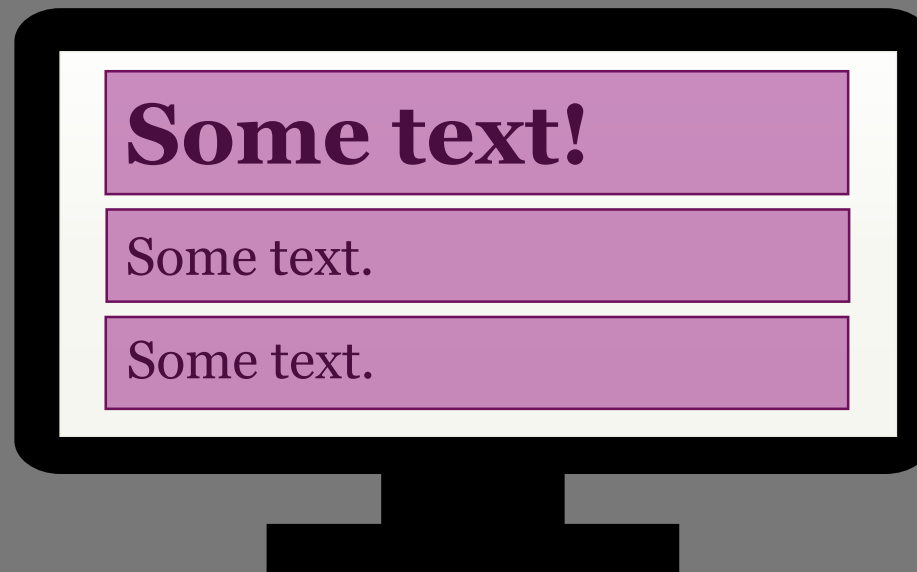
BLOCK VS INLINE

- Block elements span the entire row.
- Inline elements span as little as possible (surrounds the text).

```
<h1>Some text!</h1>  
<p>Some text.</p>  
<p>Some text.</p>
```



```
<h1>Some text</h1><p>Some text.</p><p>Some text.</p>
```



BLOCK VS INLINE

- Block elements span the entire row.
- Inline elements span as little as possible (surrounds the text).

```
<p>  A    b        c
```

```
de   f</p>
```

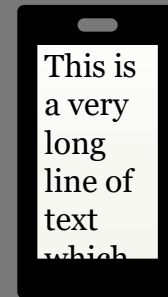
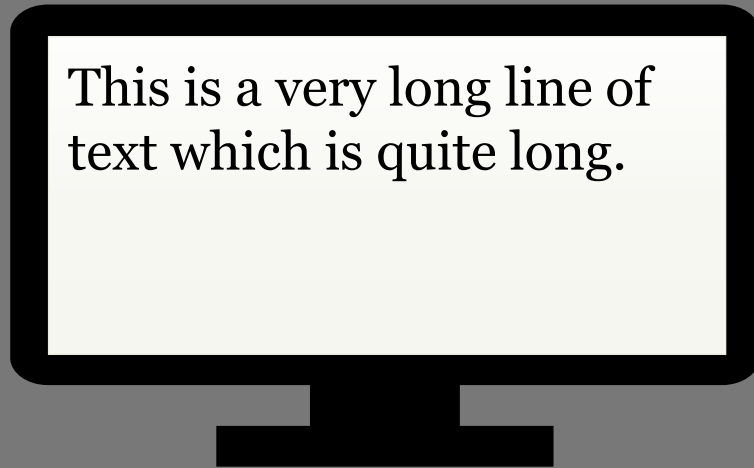
A computer monitor with a black frame and a white screen. The screen displays the text "A b c de f" in a monospaced font. The text is arranged in two lines: "A b c" on the first line and "de f" on the second line. The monitor is positioned to the right of the code blocks.

A b c de f

BLOCK VS INLINE

- Block elements span the entire row.
- Inline elements span as little as possible (surrounds the text).

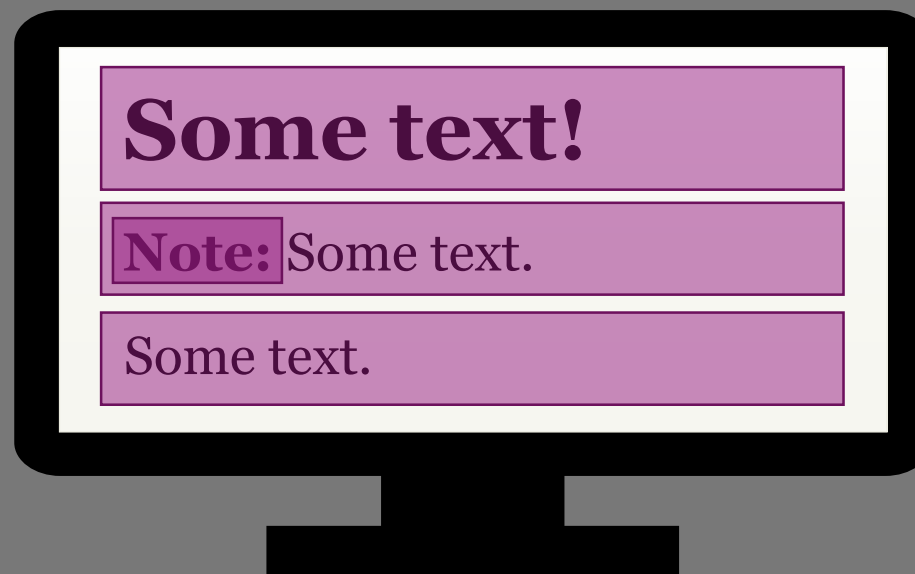
```
<p>This is a very long line of text which is quite long.</p>
```



BLOCK VS INLINE

- Block elements span the entire row.
- Inline elements span as little as possible (surrounds the text).

```
<h1>Some text</h1>
<p>
  <strong>Note:</strong>
  Some text.
</p>
<p>Some text.</p>
```



BLOCK VS INLINE

How can they be nested?

General rule 1:

- Block elements **can not** be used in inline elements.

General rule 2:

- Do not write code that does not make sense.
 - E.g. putting a paragraph inside another paragraph.

HYPER LINKS

```
<a href="https://ju.se/en.html">Go there!</a>
```

Browser renders.



User clicks.

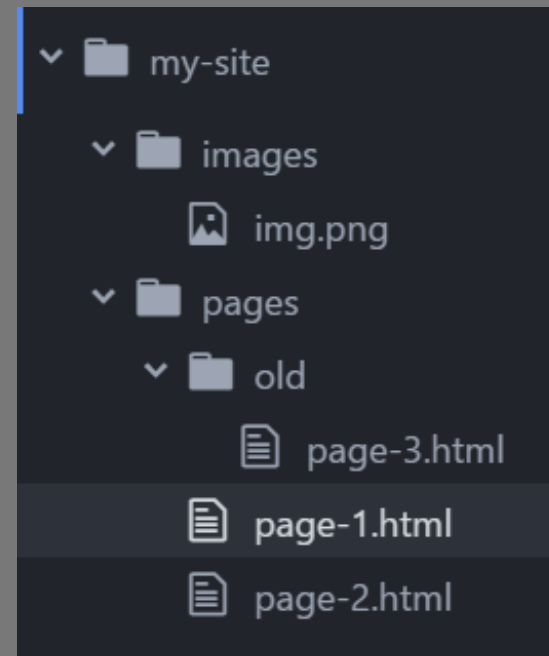
Browser sends
a new request.

```
GET /en.html HTTP/1.1  
Host: ju.se  
...
```

RELATIVE PATHS

In file `/my-site/pages/page-1.html`:

- `page-2.html`
→ `/my-site/pages/page-2.html`
- `old/page-3.html`
→ `/my-site/pages/old/page-3.html`
- `../images/img.png`
→ `/my-site/images/img.png`



`../` can be written multiple times to go back multiple folders.

EXAMPLE

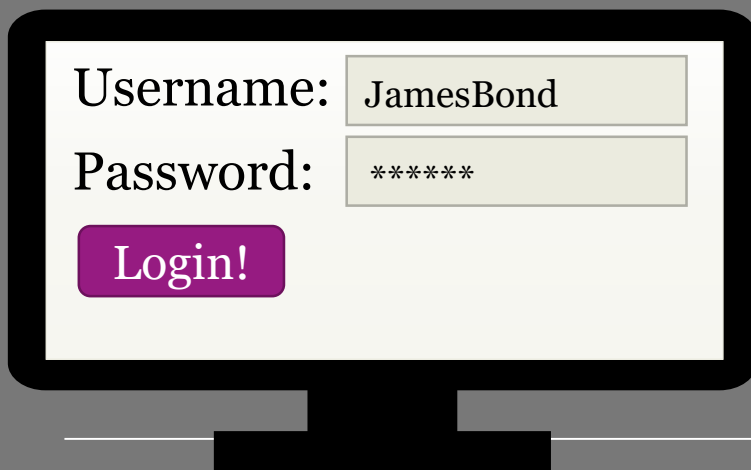
Example of a website with multiple webpages.

FORMS

```
<form method="GET" action="http://www.mi6.com/login">  
  Username: <input type="text"      name="un"><br>  
  Password: <input type="password"  name="pw"><br>  
            <input type="submit"    value="Login!">  
</form>
```

Query
string.

Browser renders.



Username: JamesBond

Password: *****

Login!

```
GET /login?un=JamesBond&pw=missMP HTTP/1.1  
Host: www.mi6.com  
...
```

User submits.

THE QUERY STRING

URIs can contain additional information in the query string.

```
/path/to/resource?THIS-IS-THE-QUERY-STRING
```

Query strings are written like this:

```
name1=value1
```

```
name1=value1&name2=value2
```

Reserved characters (`;` `/` `?` `:` `@` `=` `&`) and some special characters (`space` `"` `<` `>` `#` `%` `{` `}` `|` `\` `^` `~` `[` `]` ```) should be encoded.

- `=` → `%3D`

- `&` → `%26`

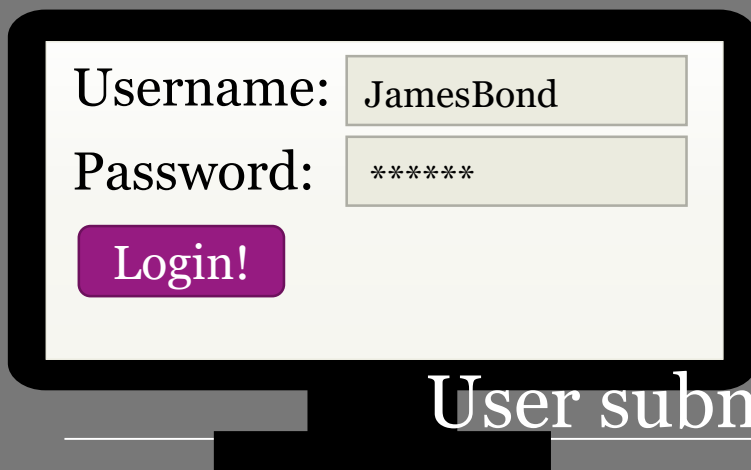
- `space` → `%20`

- `#` → `%23`

FORMS

```
<form method="POST" action="http://www.mi6.com/login">  
  Username: <input type="text"      name="un"><br>  
  Password: <input type="password" name="pw"><br>  
            <input type="submit"   value="Login!">  
</form>
```

Browser renders. ↓



Username: JamesBond
Password: *****
Login!

→ User submits.

```
POST /login HTTP/1.1  
Host: www.mi6.com  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 22  
...  
un=JamesBond&pw=missMP
```

MORE ELEMENTS...

Open a special webpage in web browser...

STRUCTURAL ELEMENTS

- `<header>` - the header part in the page.
- `<main>` - the main content in the page.
- `<footer>` - the footer in the page.
- `<aside>` - not the main content, but still relevant.
- `<nav>` - navigation links.
- `<article>` - to group the content of an article together.
- `<section>` - group relevant content together.
- `<div>` - to group elements together.
- `<div>` is an old element, the others are new in HTML5.

GLOBAL ATTRIBUTES

Attributes any element can have.

https://developer.mozilla.org/en-US/docs/Web/HTML/Global_attributes

- lang
 - Specify the language the text is written in inside the element.
 - ISO Language Codes: http://www.w3schools.com/tags/ref_language_codes.asp
 - ISO Country Codes: http://www.w3schools.com/tags/ref_country_codes.asp
- title
 - Advisory information about the element.
- class
 - Used to group similar elements together (used by CSS and JavaScript).
- id
 - Used to uniquely identify an element.

CHARACTER ENTITIES

< and > denote tags.

- What about smileys? :-< >.< >.>
- &NAME; alternative representation.
- < → <
- > → >
- What about &?
 - & → &

USEFUL TOOLS

<https://validator.w3.org>

- Validates your HTML code.

<https://jsfiddle.net>

- Write code online in a web browser.
- See the result.
- Share your code.