JÖNKÖPING UNIVERSITY

School of Engineering

HTML

Web Development with JavaScript and DOM

TWJK14 Spring 2017

Peter Larsson-Green

HYPERTEXT MARKUP LANGUAGE

Is a markup language.

- No programming!

Syntax:

```
<tag>
```

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

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

↕

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

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

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

↕

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

VERSIONS AND BROWSERS...

1991: HTML "1.0" (a paper with list of tags).

1995: HTML 2.0.

```
<blink>Text</blink>
```

Netscape

```
<marquee>Text</marquee>
```

Internet Explorer

1997: HTML 3.2.

1997: HTML 4.0 (three versions: strict, transitional, frameset).

2000: XHTML 1.0.

2001: XHTML 1.1.

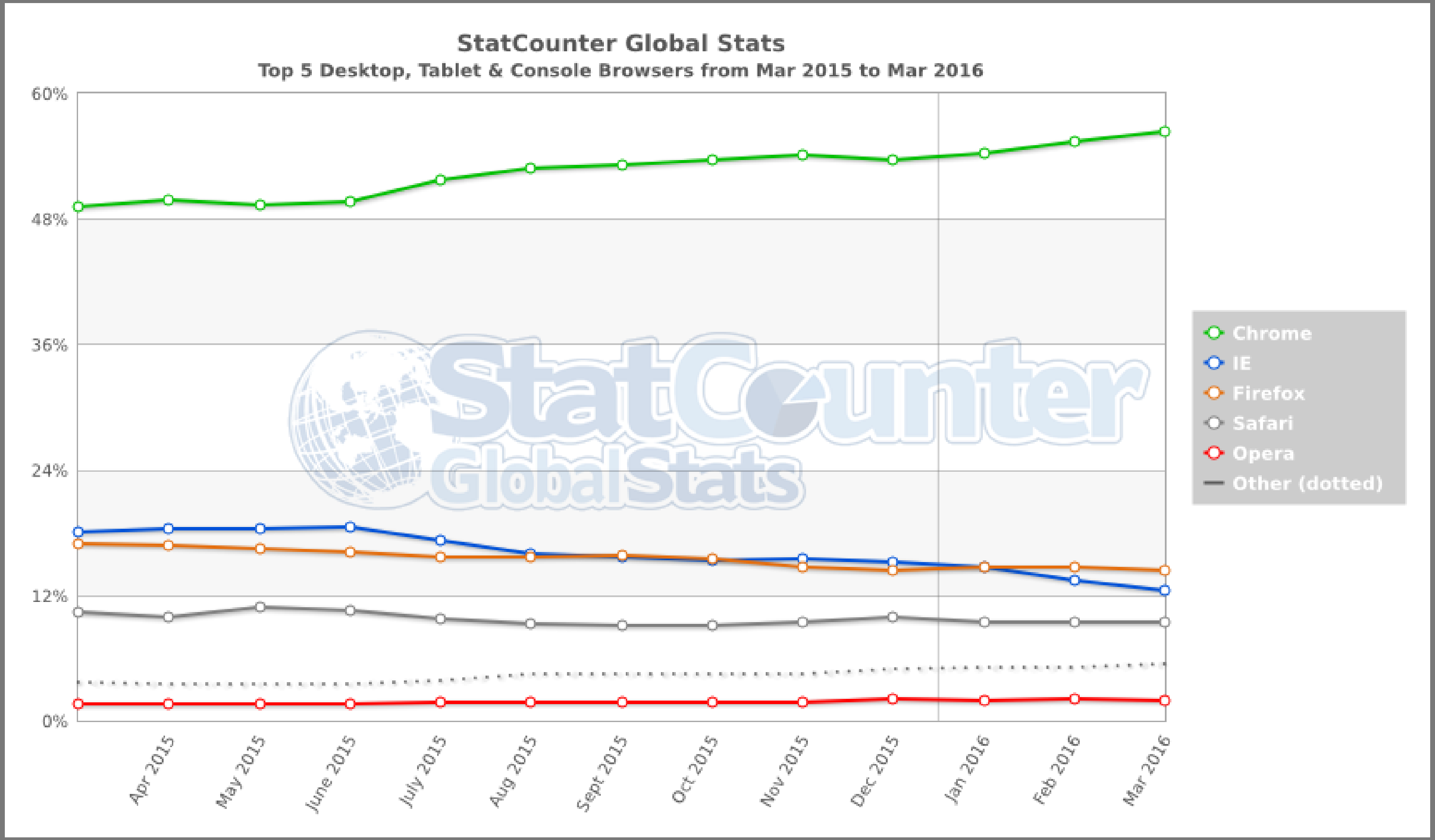
2014: HTML 5.0.

XHTML VS HTML

XHTML is stricter → less mistakes by programmers.

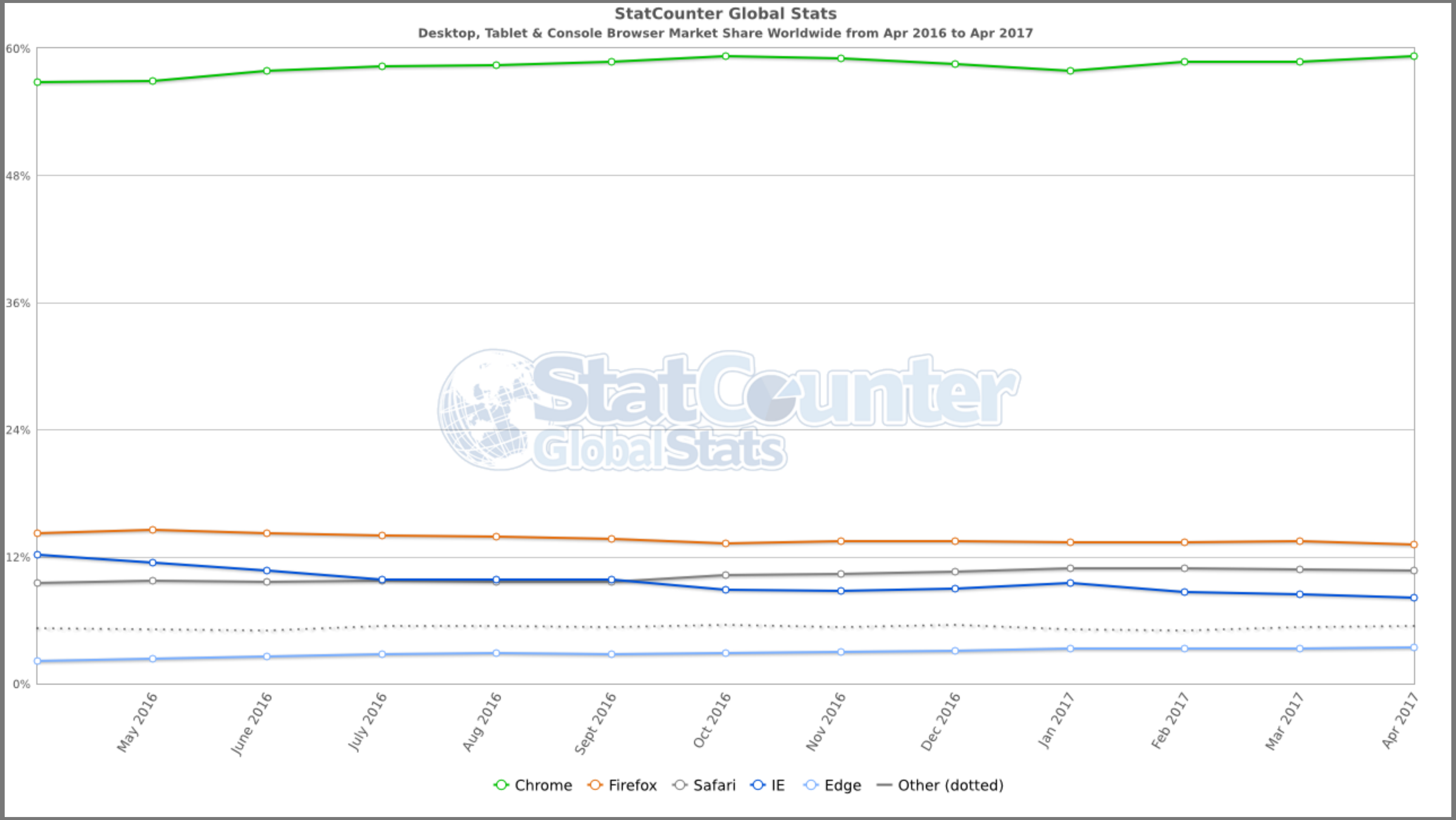
- XHTML needs to be valid XML.
  - `<tag>` is not allowed; must close tag, e.g.: `<tag/>`
  - `<tag attribute=value>` is not allowed; must use single/double quotes.
  - Tags must be closed in right order. Wrong: `<b><i>Word</b></i>`
- Tag & attribute names need to be lowercase in XHTML.

BROWSER USAGE (WORLD, 15-16)



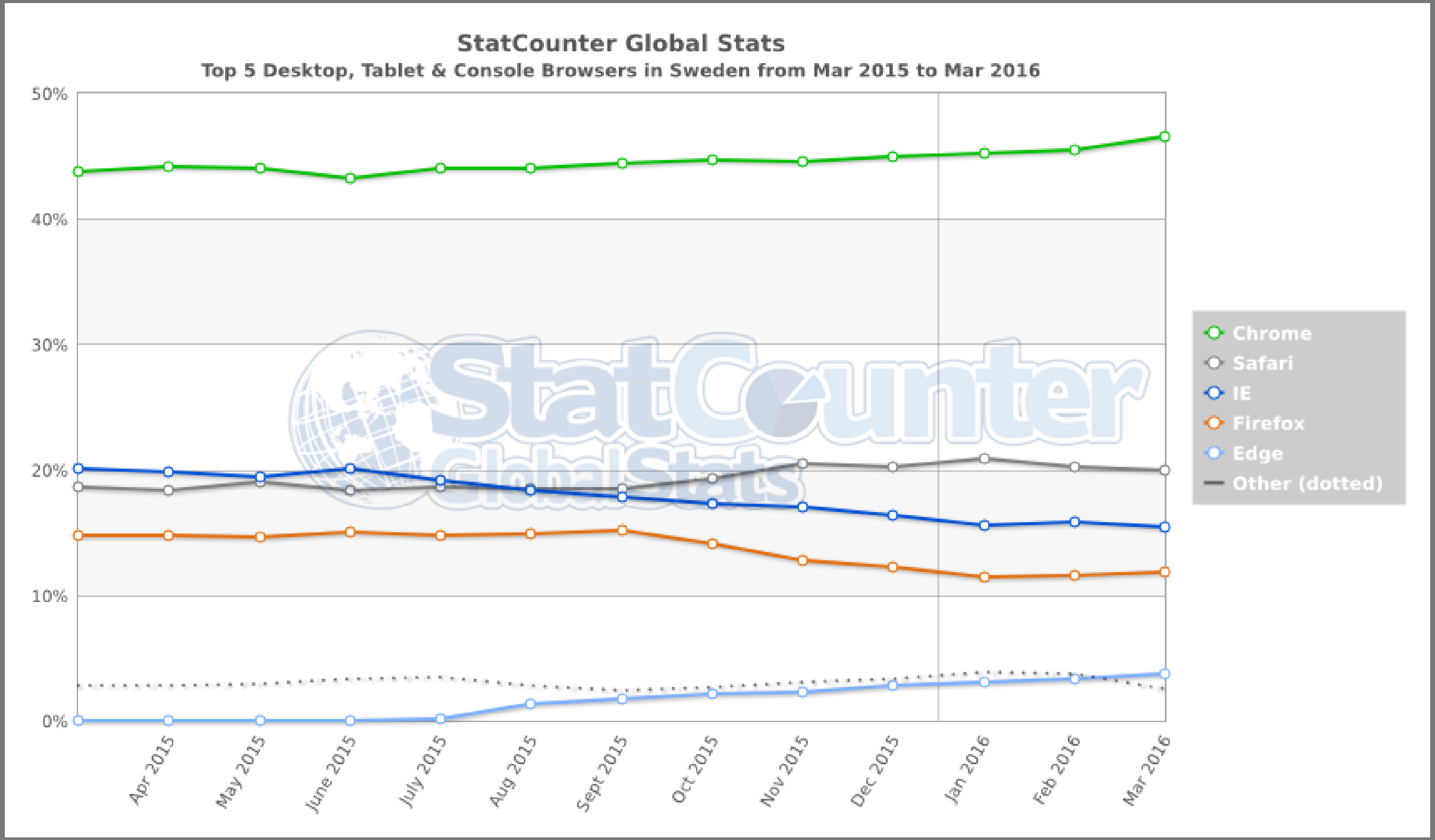
http://gs.statcounter.com/

BROWSER USAGE (WORLD, 16-17)



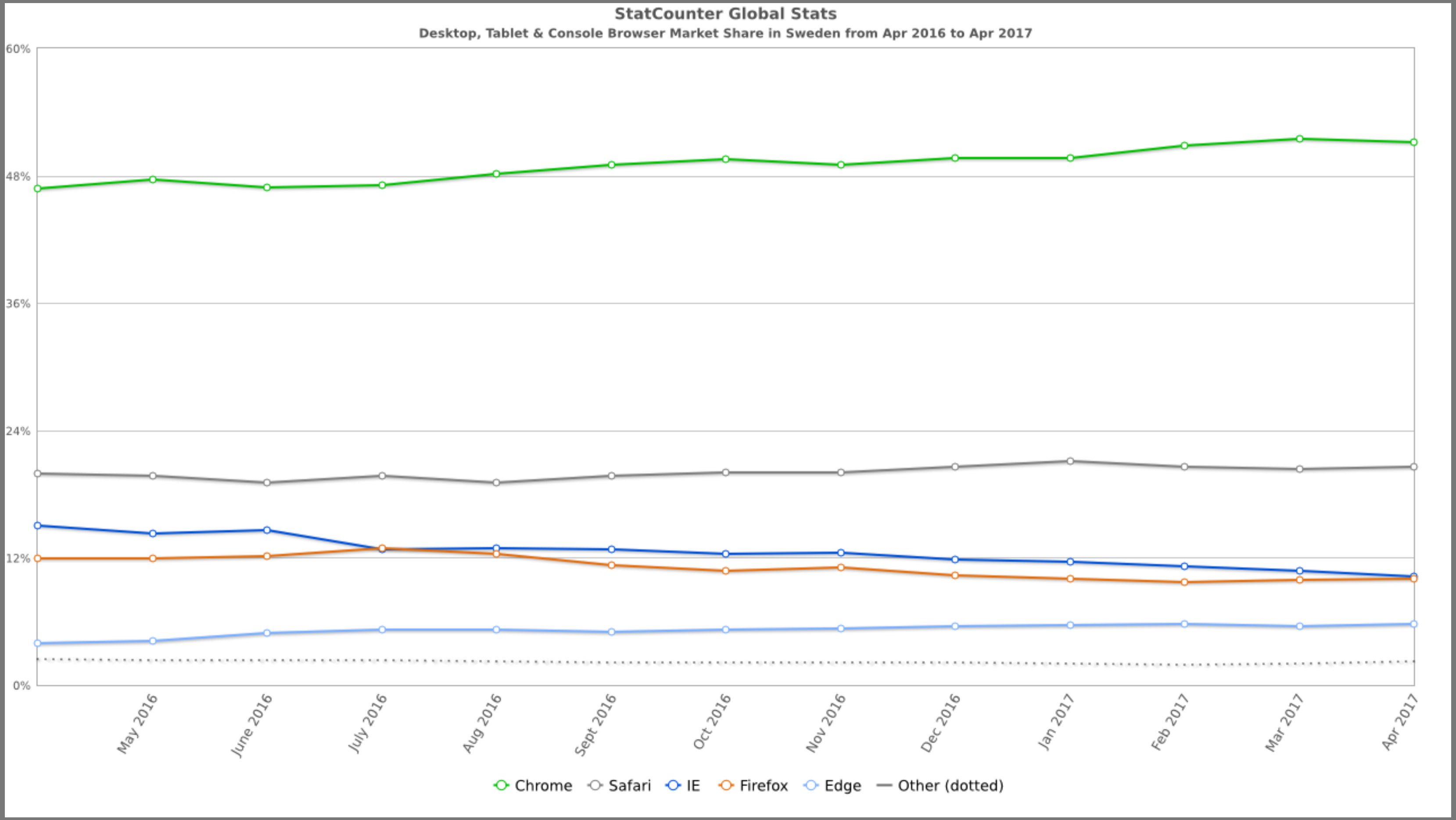
http://gs.statcounter.com/browser-market-share/desktop-tablet-console/worldwide/#monthly-201604-201704

BROWSER USAGE (SWEDEN, 15-16)



http://gs.statcounter.com/#browser-SE-monthly-201503-201603

BROWSER USAGE (SWEDEN, 16-17)



http://gs.statcounter.com/browser-market-share/desktop-tablet-console/sweden/#monthly-201604-201704

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,talk snake,quidditch">
```

Search engines!

Check it out: http://www.svt.se/svttext/web/pages/100.html

SOME <BODY> ELEMENTS

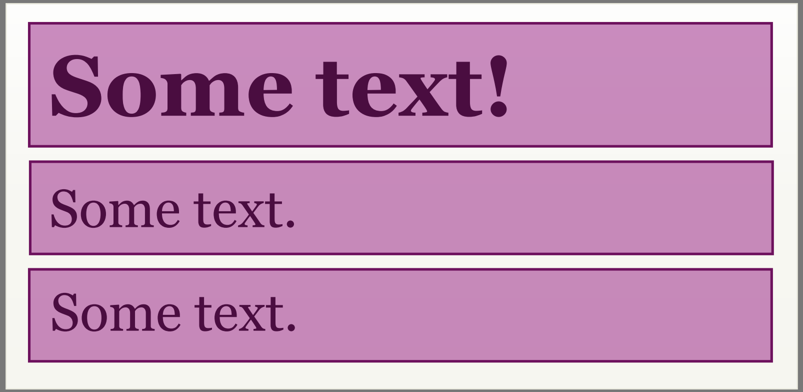
Most tags belong to one of the following groups:

- Block elements:
  - `<h1>`
  - `<p>`
  - `<div>`
  - …
- Inline elements:
  - `<a>`
  - `<abbr>`
  - `<span>`
  - …

BLOCK VS INLINE

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

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



Some text!

Some text.

Some text.

⬍

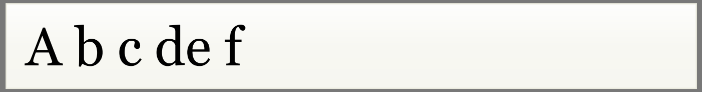
```
<h1>Some text</h1><p>Some text.</p><p>Some text.</p>
```

BLOCK VS INLINE

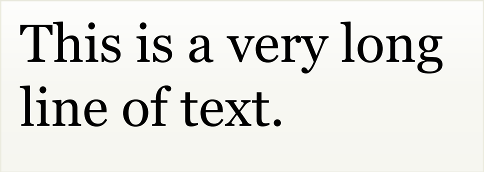
- Block elements spans the entire row.
- Inline elements spans as little as possible (surrounds the text).

```
<p>  A   b       c


de   f</p>
```


A b c de f

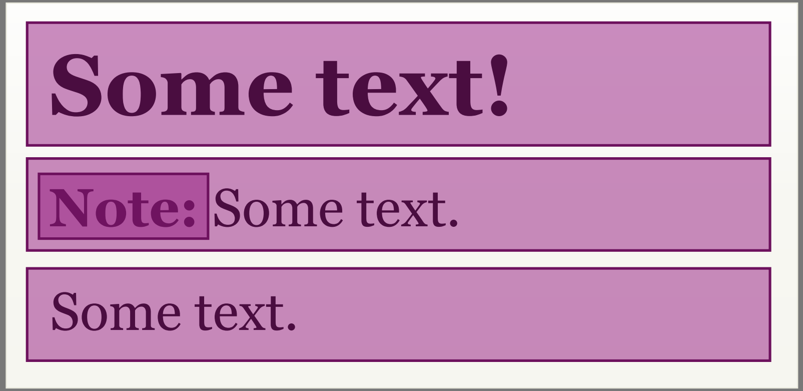
```
<p>This is a very long line of text.</p>
```


This is a very long line of text.

BLOCK VS INLINE

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

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



Some text!

Note: Some text.

Some text.

BLOCK VS INLINE

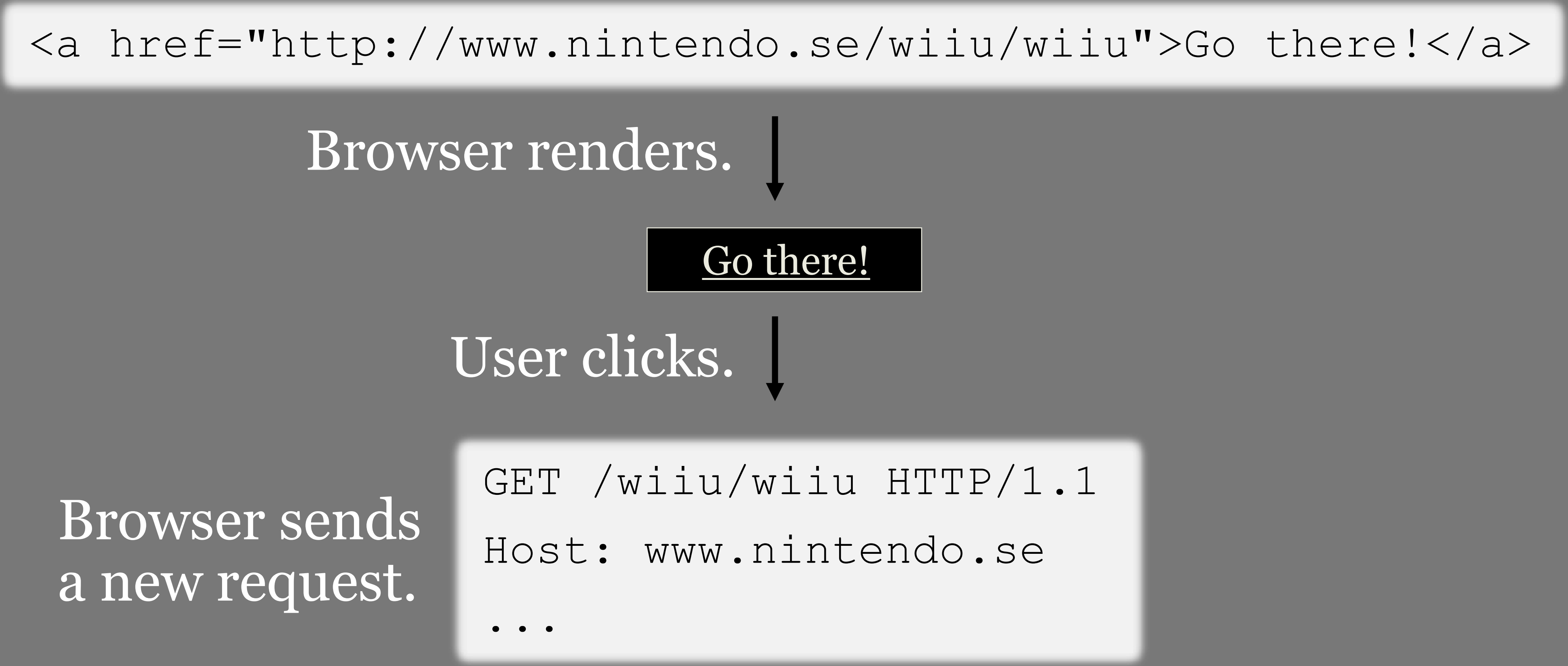
How can they be nested?

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

Additional restrictions apply.

- Do not put paragraphs in paragraphs.
- …

HYPER LINKS



```
<a href="http://www.nintendo.se/wiiu/wiiu">Go there!</a>
```

Browser renders.

Go there!

User clicks.

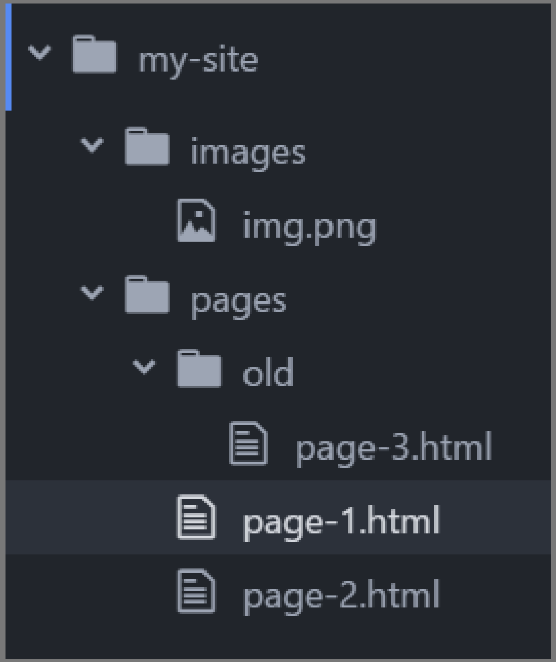
Browser sends a new request.

```
GET /wiiu/wiiu HTTP/1.1
Host: www.nintendo.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`



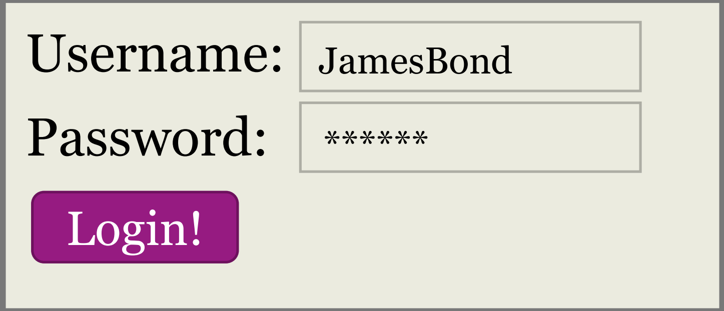
`../` can be written multiple times to go back multiple folders.

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: ******

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

User submits.

THE QUERY STRING

Request 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
```

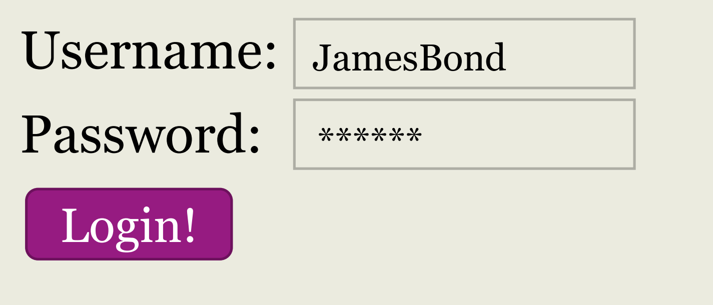
Reserved characters (`; / ? : @ = &`) and some special characters (`space " < > # % { } | \ ^ ~ [ ] ``) should be encoded.

- `=` → `%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: ******

User submits.

```
POST /login HTTP/1.1
Host: www.mi6.se
Content-Type: application/x-www-form-urlencoded
Content-Length: 22
...


un=JamesBond&pw=missMP
```

GLOBAL ATTRIBUTES

Attributes any element can have.

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

- `accesskey`
  - Access to element through keyboard combination.
- `hidden`
  - Hides elements (can later be shown with JavaScript).
- `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

GLOBAL ATTRIBUTES

Attributes any element can have.

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

- `title`
  - Advisory information about the element.
- `class`
  - Used to group similar elements together (used by CSS and JavaScript).
- `id`
  - Used by CSS and JavaScript to uniquely identify an element.

WEB CONTENT ACCESSIBILITY GUIDELINES

https://www.w3.org/WAI/intro/wcag

Quick reference:

- https://www.w3.org/WAI/WCAG20/quickref

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.

CHARACTER ENTITIES

< and > denote tags.

- What about smilies? `:-<` `>.<` `>.>`
    - `&NAME;` alternative representation.
    - `&lt;` → `<`
    - `&gt;` → `>`
- What about `&`?
    - `&amp;` → `&`

https://www.w3.org/TR/html5/syntax.html#named-character-references

USEFUL TOOLS

https://jsfiddle.net

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

http://browsershots.org

- Takes screenshots of your website in different browsers.

https://validator.w3.org

- Validates your HTML code.

RECOMMENDED READING

Dive into HTML5 by *Mark Pilgrim* (funny to know)

- Chapter 1, Introduction: Five Things You Should Know About HTML5
    - http://diveintohtml5.info/introduction.html
- Chapter 2, A Quite Biased History of HTML5
    - http://diveintohtml5.info/past.html
- Chapter 3, What does it all mean? (the section *THE DOCTYPE* only)
    - http://diveintohtml5.info/semantics.html

W3Sschools

- HTML tutorial
    - https://www.w3schools.com/html/default.asp

RECOMMENDED READING

Mozilla's resources

- HTML guide
  - https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Introduction
- HTML documentation
  - https://developer.mozilla.org/en-US/docs/Web/HTML/Reference

W3's HTML specification:

  - https://www.w3.org/TR/html5/