Developing Web Pages

Dr Jon Bird

Overview

This lecture is about how to develop web pages.

The focus is on the client side.

It covers:

- HTML
- best practice for structuring web pages with HTML5

Markup Ignguages

Historically, ‘marking up’ a paper manuscript was done by editors to show authors how to revise their manuscripts.

The markup was done in blue pen to make it distinguishable from the manuscript text.

Markup languages are used to annotate electronic documents and tags are often used to make the markup distinguishable from the content.

HTML

HTML stands for Hypertext Markup Language

It consists of a fixed set of tags that describe how information should be displayed by browsers e.g. <h1>, <p>,

Browsers do not display the HTML tags, but use them to render the contents of the page.

Short history of HTML

From 1991 to 1999, HTML developed from version 1 to version 4.

In 2000, the World Wide Web Consortium (W3C) recommended XHTML 1.0. The XHTML syntax was strict, and developers were forced to write valid and "well-formed" code.

In 2004, W3C decided to close down the development of HTML, in favor of XHTML.

Also in 2004, WHATWG (Web Hypertext Application Technology Working Group) formed to develop HTML that was consistent with how the web was used, as well as being backward compatible with older versions of HTML.

2014 HTML5 was released by WC3

2017 HTML5.1 and HTML5.2

HTML5

HTML5

HTML5 is different from HTML:

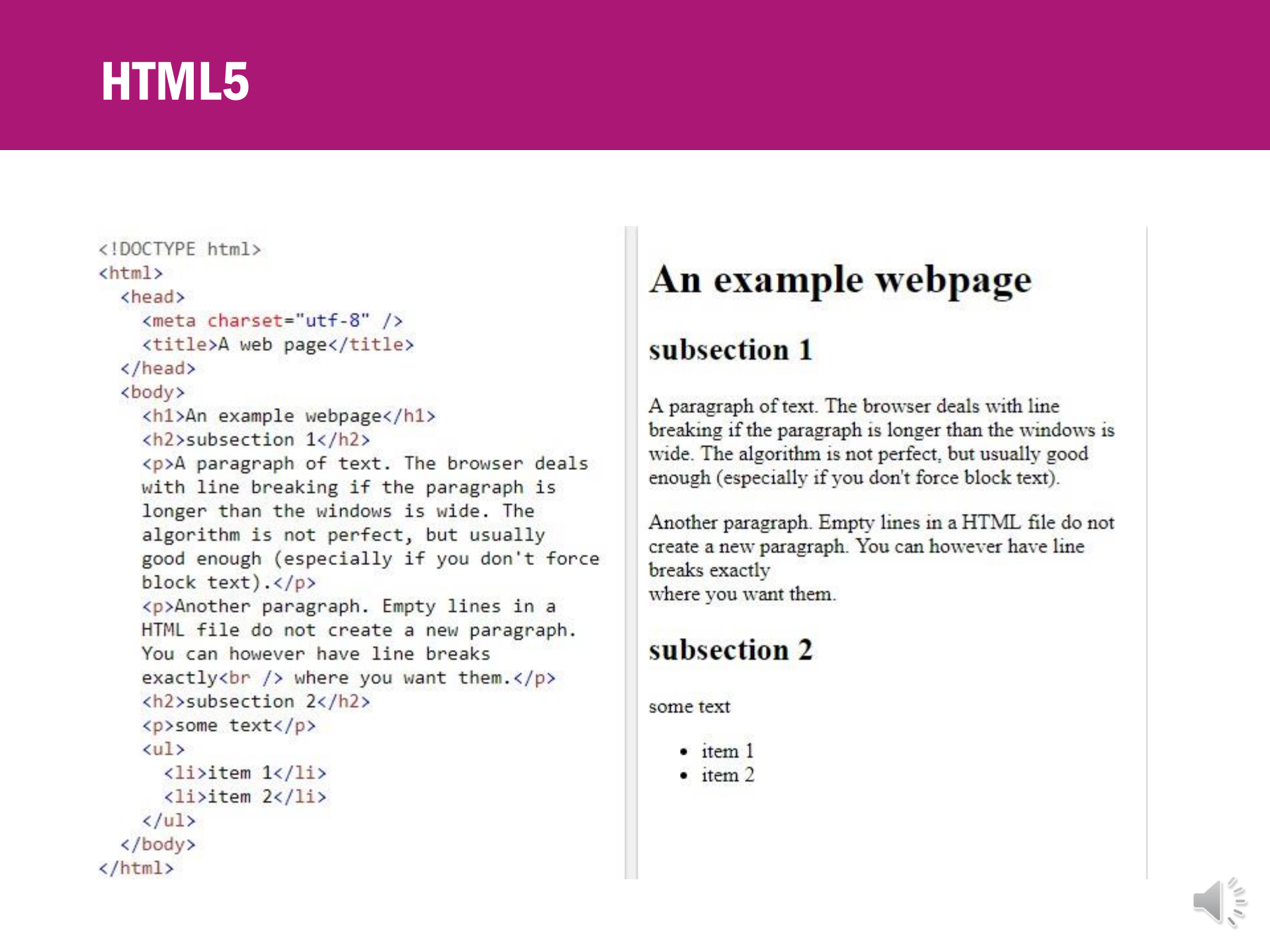
- simpler
- semantic (some of the tags describe what the data means as well as how it should be displayed)
- more features

HTML5

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title>A web page</title>
  </head>
  <body>
    your content here
  </body>
</html>
```

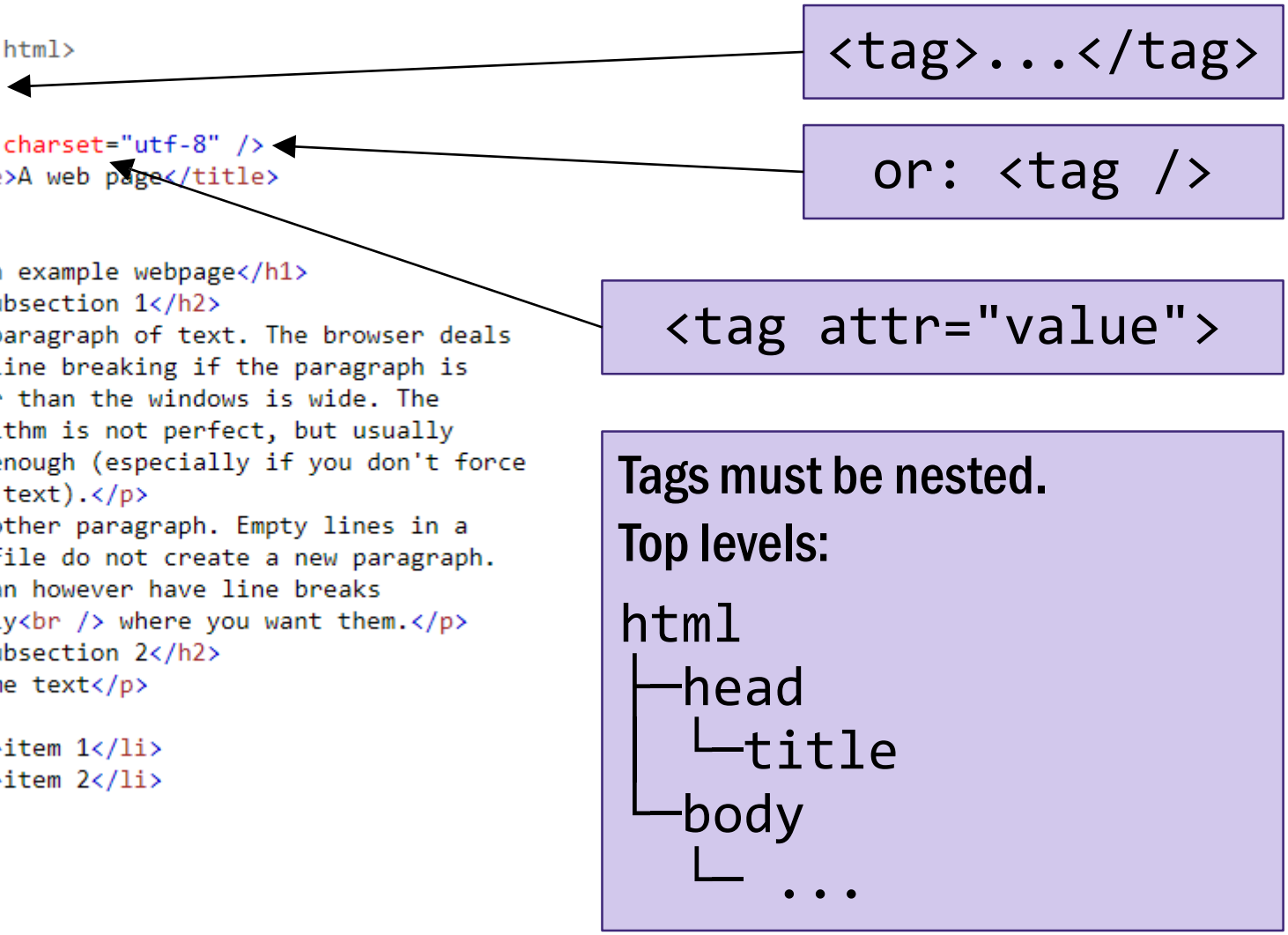
HTML5

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>A web page</title>
  </head>
  <body>
    <h1>An example webpage</h1>
    <h2>subsection 1</h2>
    <p>A paragraph of text. The browser deals
    with line breaking if the paragraph is
    longer than the windows is wide. The
    algorithm is not perfect, but usually
    good enough (especially if you don't force
    block text).</p>
    <p>Another paragraph. Empty lines in a
    HTML file do not create a new paragraph.
    You can however have line breaks
    exactly<br /> where you want them.</p>
    <h2>subsection 2</h2>
    <p>some text</p>
    <ul>
      <li>item 1</li>
      <li>item 2</li>
    </ul>
  </body>
</html>
```



HTML5

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>A web page</title>
  </head>
  <body>
    <h1>An example webpage</h1>
    <h2>subsection 1</h2>
    <p>A paragraph of text. The browser deals
    with line breaking if the paragraph is
    longer than the windows is wide. The
    algorithm is not perfect, but usually
    good enough (especially if you don't force
    block text).</p>
    <p>Another paragraph. Empty lines in a
    HTML file do not create a new paragraph.
    You can however have line breaks
    exactly<br /> where you want them.</p>
    <h2>subsection 2</h2>
    <p>some text</p>
    <ul>
      <li>item 1</li>
      <li>item 2</li>
    </ul>
  </body>
</html>
```



HTML5

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>A web page</title>
  </head>
  <body>
    <h1>An example webpage</h1>
    <h2>subsection 1</h2>
    <p>A paragraph of text. The browser deals
    with line breaking if the paragraph is
    longer than the windows is wide. The
    algorithm is not perfect, but usually
    good enough (especially if you don't force
    block text).</p>
    <p>Another paragraph. Empty lines in a
    HTML file do not create a new paragraph.
    You can however have line breaks
    exactly<br /> where you want them.</p>
    <h2>subsection 2</h2>
    <p>some text</p>
    <ul>
      <li>item 1</li>
      <li>item 2</li>
    </ul>
  </body>
</html>
```

`h1 ... h6` headings

`<p>` paragraphs
`<br />` new line

`<ul>` unordered list
`<ol>` ordered list
`<li>` list item

New tags

new		old	
`<em>`	emphasis	`<b>`	**bold**
`<strong>`	important	`<i>`	*italics*
`<q>`	quotation	`<u>`	underline
`<cite>`	citation	`<s>`	~~strike out~~
`<var>`	variable	`<tt>`	`monospace`
`<code>`	source code	`<small>`	small
		`<span>`	custom

https://developer.mozilla.org/en/docs/Web/HTML/Element

Block and inline elements

A *block-level* element always starts on a new line and takes up the full width available (stretches out to the left and right as far as it can).

```
<p>hello world</p>
```

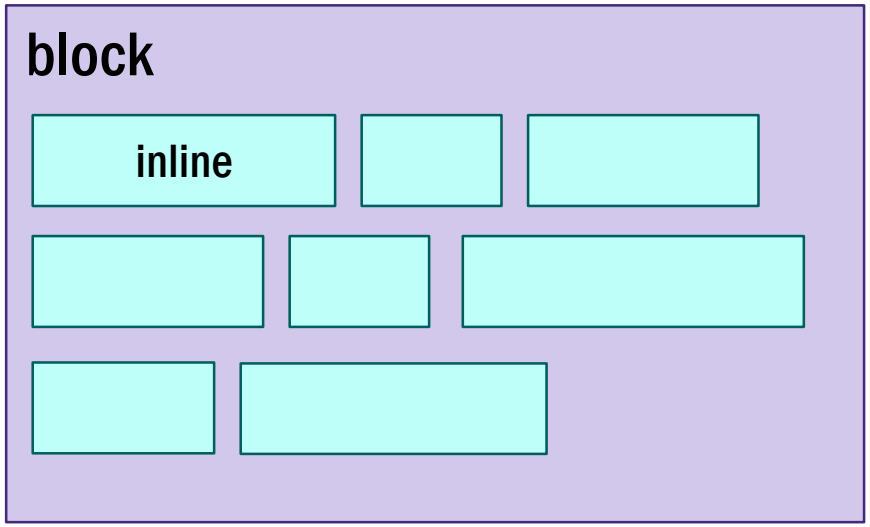
```
<form>
```

```
<table>
```

An *inline* element does not start on a new line and only takes up as much width as necessary.

```
<b>make this text bold</b>
```

Block and inline

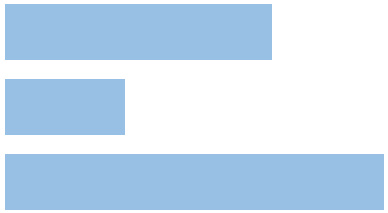


Blocks stack vertically, inline elements horizontally within blocks.

Block and inline

Block elements

Block elements appear on a new line.
Examples: <h1><p>
<table><form>



```
<h1>Hiroshi Sugimoto</h1>
<p>The dates for the ORIGIN OF ART exhbibition are as
    follows:</p>
<ul>
    <li>Science: 21 Nov- 20 Feb 2010/2011</li>
    <li>Architecture: 6 Mar - 15 May 2011</li>
</ul>
```

Hiroshi Sugimoto

The dates for the ORIGIN OF ART exhbibition are as follows:

- Science: 21 Nov- 20 Feb 2010/2011
- Architecture: 6 Mar - 15 May 2011

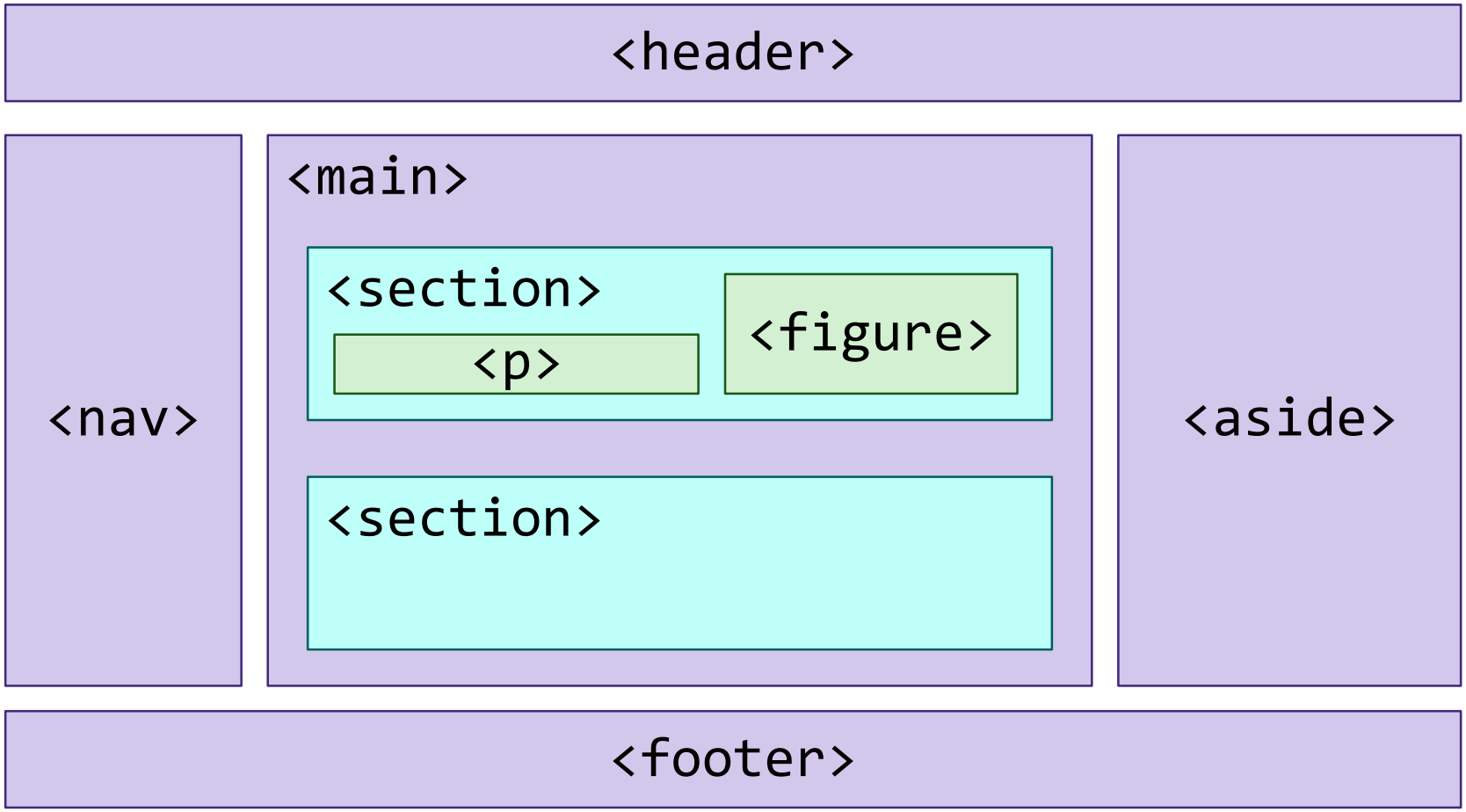
Inline elements

Inline elements appear to continue on the same line.
Examples: <a><b><input><img>

```
Timed to a single revolution of the planet around the sun
at a 23.4 degrees tilt that plays out the rhythm of the
seasons, this <em>Origins of Art</em> cycle is organized
around four themes: <b>science, architecture, history</b>,
and <b>relgion</b>.
```

Timed to a single revolution of the planet around the sun at a 23.4 degrees tilt that plays out the rhythm of the seasons, this *Origins of Art* cycle is organized around four themes: **science, architecture, history**, and **relgion**.

Block tags



div

`<div>` = block level tag with no specific meaning

This is ok to use for layout purposes but not as a replacement for something that should be a semantic tag, e.g. a navigation section <nav>

Semantic tags are mostly new to HTML5 – older frameworks used all over the place to structure pages.

Attributes, id, class

```
<p id="today">
  28 September
</p>
<p class="info">
  Lecture 2
</p>
<p class="info">
  QB 0.18
</p>
```

id: should be unique on the page – for scripts, labels etc.

class: marks any number of elements that you might want to operate on as a group, for example for styling.

Links

```
<a href="/courses/courses.html">Our Courses</a>
```

Almost anything can go inside an <a> tag: text, images, other HTML elements.

The href can be a full URL or relative to the current page.

HTML5

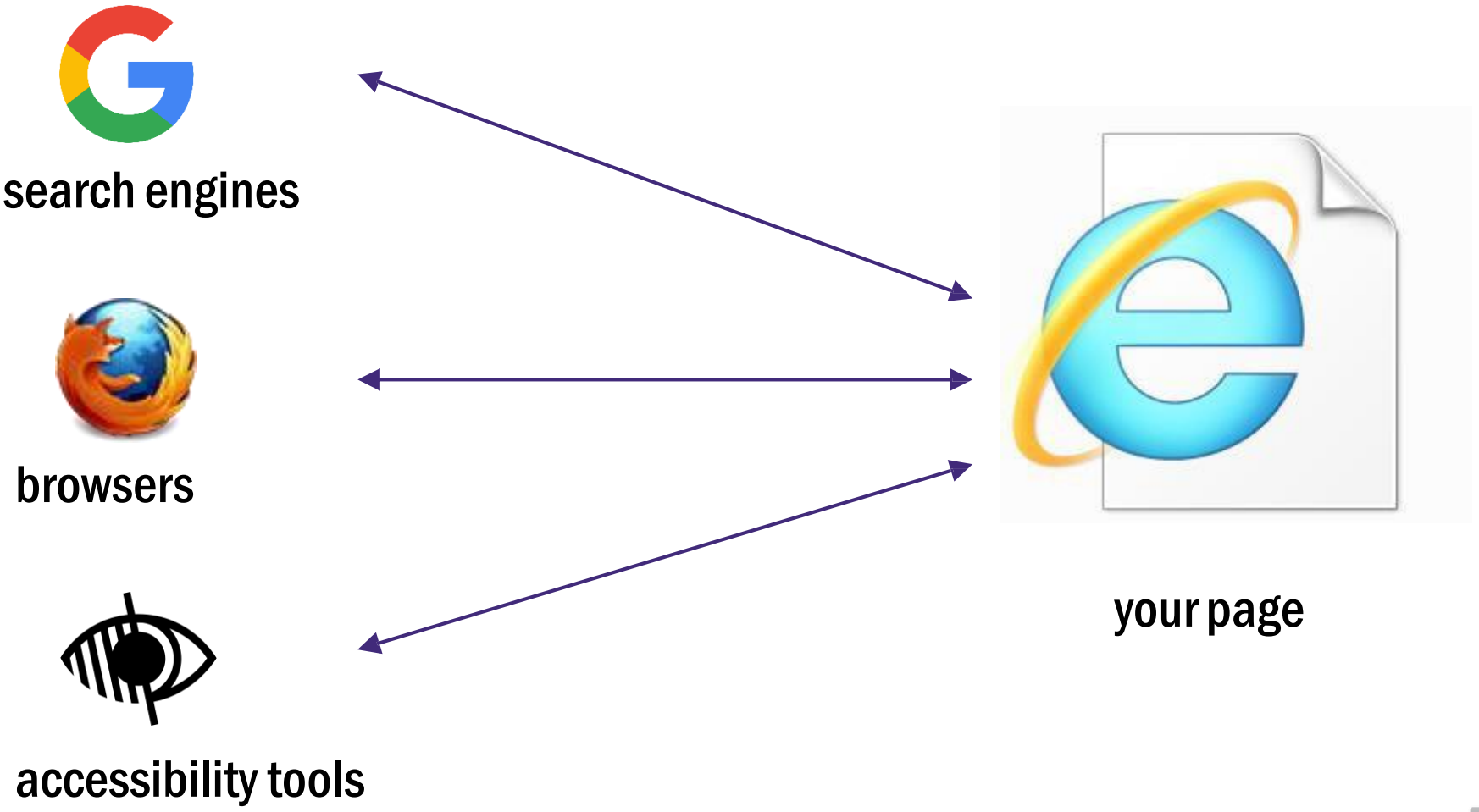
HTML5 is a bit more sloppy when it comes to code validation in comparison to XHTML

A consistent use of style makes it easier for others to understand your HTML.

In the future, programs like XML readers may want to read your HTML.

Using a well-formed "close to XHTML" syntax is smart.

Why structure web pages carefully?



How to structure your HTML 1

Use lower case element names

```
<SECTION>
  <p>This is a paragraph.</p>
</SECTION>
```

BAD

```
<Section>
  <p>This is a paragraph.</p>
</SECTION>
```

BAD

```
<section>
  <p>This is a paragraph.</p>
</section>
```

GOOD

How to structure your HTML 2

In HTML5, you don't have to close all elements (for example the <p> element) but it is best practice to close all HTML elements.

```
<section>
  <p>This is a paragraph.
  <p>This is a paragraph.
</section>
```

BAD

```
<section>
  <p>This is a paragraph.</p>
  <p>This is a paragraph.</p>
</section>
```

GOOD

How to structure your HTML 3

In HTML5, it is optional to close empty elements. However, the closing slash (/) is REQUIRED in XHTML and XML.

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

BAD

```
<meta charset="utf-8" />
```

GOOD

How to structure your HTML 4

HTML5 allows the mixing of uppercase and lowercase letters in attribute names.

Use lowercase attribute names because:

- mixing uppercase and lowercase names is confusing;
- developers normally use lowercase names (as in XHTML);
- lowercase look cleaner;
- lowercase are easier to write.

```
<div CLASS="menu">
```

BAD

```
<div class="menu">
```

GOOD

How to structure your HTML 5

HTML5 allows attribute values without quotes.

It is best to quote attribute values because:

- developers normally quote attribute values (as in XHTML);
- quoted values are easier to read;
- you MUST use quotes if the value contains spaces.

```
<table class=striped>
```
BAD

```
<table class="striped">
```
GOOD

How to structure your HTML 6

Always add the alt attribute to images.

This attribute is important when the image for some reason cannot be displayed.

It is also needed by screen readers used by the visually impaired.

Also, always define image width and height. It reduces flickering because the browser can reserve space for the image before loading.

```
<img src="html5.gif">
```

BAD

```
<img src="html5.gif" alt="HTML5" style="width:128px;
height:128px">
```

GOOD

How to structure your HTML 7a

In HTML5, the <html> tag and the <body> tag can be omitted.
The following code will validate as HTML5 but it is **BAD**

```
<!DOCTYPE html>
<head>
 <title>Page Title</title>
</head>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

How to structure your HTML 7b

Omitting <html> or <body> can crash XML software.

Omitting <body> can produce errors in older browsers (IE9).

The <html> element is the document root. It is the recommended place for specifying the page language.

Declaring a language is important for accessibility applications, such as screen readers, and search engines.

```
<!DOCTYPE html>
<html lang="en-US">
```

GOOD

How to structure your HTML 8

To ensure proper interpretation and correct search engine indexing, both the language and the character encoding should be defined as early as possible in a document:

```
<!DOCTYPE html>
<html lang="en-US">
<head>
 <meta charset="UTF-8" />
 <title>HTML5 Syntax and Coding Style</title>
</head>
```

GOOD

How to structure your HTML 9a

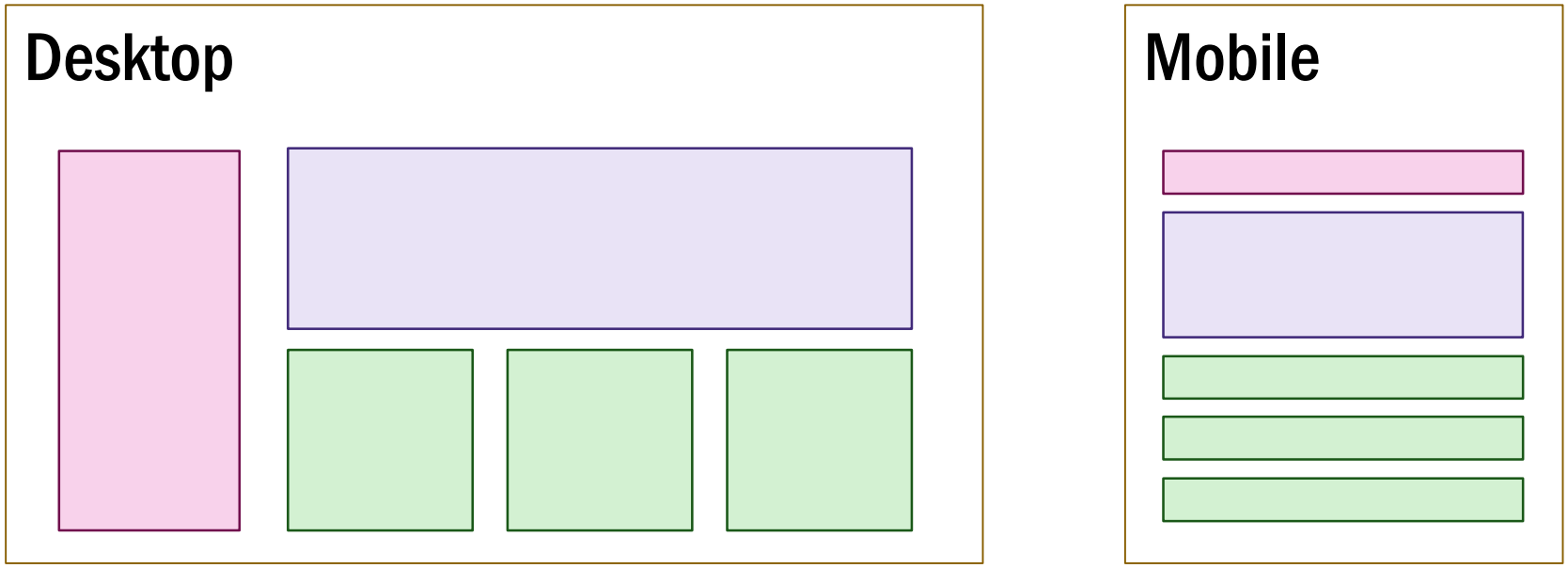
Don’t specify a fixed page width e.g.

```
body { width: 800px; }
```

- empty space on bigger screens
- scrollbars on smaller screens
- people will hate you

Layouts need to adapt to different screen sizes e.g. from a smartphone to a widescreen TV.

Responsive layout



How to structure your HTML 9b

HTML5 introduced a method to let web designers take control over the viewport, through the <meta> tag.
A <meta> viewport element gives the browser instructions on how to control the page's dimensions and scaling.
The width=device-width part sets the width of the page to follow the screen-width of the device (which will vary depending on the device).
The initial-scale=1.0 part sets the initial zoom level when the page is first loaded by the browser.

```
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
```

GOOD

Reading

www.w3schools.com

This is a great resource for information on a wide range of web technologies, including HTML5 and CSS