COLLAGE OF COMPUTER SCIENCE & ENGINEERING

UNIVERSITY OF JEDDAH

كلية علوم و هندسة الحاسب

جامعة جدة

HTML: Hypertext Markup Language

CCSW 321 (Web Development)

What will be covered

- Introduction to HTML.
- Your first HTML web page.
- HTML elements such as headings, paragraphs, images, videos, links, and lists.
- Absolute vs relative path.
- HTML elements display types.
- The HTML Validation service.
- HTML5 Boilerplate.

Introduction

- Web pages are **text files**, written in a language called Hypertext Markup Language (**HTML**).
- A **markup language** is a language used to describe the document's structure and content.
- It is the job of the **web browser** to **interpret** HTML elements/tags and render the content accordingly.

Introduction

- Basics and more sophisticated HTML5 techniques as:
  - **Tables and Lists**: particularly useful for structuring information from databases (i.e., software that stores structured sets of data)
  - **Forms**: collect information from web-page visitors
  - **Internal linking**: easier page navigation.
  - **Meta elements**: specify information about a document
- **Web servers** store HTML5 documents.
- **Clients** (browsers on PC or smartphone) request specific resources such as HTML5 documents from web servers.

Basic HTML page

Content between <head> doesn't appear in the viewport of the browser

Contents between <body> renders in the viewport of the browser

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.1: main.html -->
4  <!-- First HTML5 example. -->
5  <html>
6     <head>
7        <meta charset = "utf-8">
8        <title>Welcome</title>
9     </head>
10
11    <body>
12       <p>Welcome to HTML5!</p>
13    </body>
14 </html>
```

Fig. 2.1 | First HTML5 example.

HTML Basics

- **Document Type Declaration (DOCTYPE)**:
  - Required in HTML5 documents so browsers render the page in standards mode (W3C and IETF recommendations).
- **HTML5** contains text, images, graphics, animations, audios and videos that represents the content of a document and elements that specify a document's structure and meaning.
- The html element **encloses** the **head** and the **body** elements.

HTML Basics

- The **head section** contains information about the HTML5 document (i.e. UTF-8 character set encoding scheme); helps browsers determine how to render the content and the title.
- The **head section** also can contain the CSS3 style sheets and client-side JavaScript for creating dynamic web pages.
- The **body section** contains the page's content, which the browser displays when the user visits the web page

HTML Basics

- HTML5 documents delimit elements with a **start and end tags**.
- A **start tag** consists of the element name in angle brackets. For example, *<html>*
- An **end tag** consists of the element name preceded by a forward slash (/) in angle brackets. For example, *</html>*
- There are several "**void elements**" that do **not** have end tags. For example, the <img/> tag.
- Many start tags have **attributes** that provide more information about an element. Browsers use it to decide how to process it.
- **Attributes** have name and value separated by an equals sign (=).

The Head Section

- The **head section** of the web page includes tags that do not appear in the viewport. The tags within the <head> section of a web page provide information to browsers and other software that processes web content.
- The **<title> tag** shows the title of your page at the top of your browser window tab
- The head section usually contain **links** to the the **CSS** and **JavaScript** files used in the page.

The Head Section

- One of the important tags included in the head section are the **meta tags.** They provide additional information to browsers and search engines.
- Here are a few examples:
  - <meta charset="UTF-8">
    - This tag specifies the character encoding used in the document. UTF-8 is a widely-used character encoding that supports a wide range of characters, including those used in many languages other than English.
  - <meta name="viewport" content="width=device-width, initial-scale=1.0">
    - This tag sets the viewport for the page, which determines how the page is displayed on different devices. The content attribute specifies the width of the viewport, and the initial-scale attribute sets the initial zoom level.

The Head Section

- <meta name="description" content="...">
  - It provides a short description of the page's content for search engines and social media platforms. The content attribute should contain a brief summary of the page's content.
- <meta name="keywords" content="...">
  - It provides a list of keywords or phrases that describe the page's content. These keywords can be used by search engines to help index the page.
- <meta name="robots" content="...">
  - It provides instructions to search engine crawlers about how to index the page. The content attribute can be set to "index", "noindex", "follow", or "nofollow", among others.

- There are many other meta tags that can be used in HTML, and new tags are often added as web standards evolve. Using meta tags can help improve the accessibility, usability, and visibility of your web pages, so it's worth taking the time to learn about them and use them effectively.

The Body Section

- The **<body> section** in HTML is used to define the main content area of a web page. It contains all the **visible content** of the page, including text, images, videos, and other media.

Top-level heading **h1, h2, … h6**	`<h1>Moby Dick</h1>`
Paragraph	`<p>Call me Ishmael.</p>`
Image	`<img src="cover.png" />`
Link	`<a href="google.com">click here!</a>`
Lists	`<ol><li>Item1</li></ol>` `<ul><li>Item1</li></ul>`

Headings

- HTML5 provides **six heading elements** (h1 through h6) for specifying the relative importance of information
- Heading element **h1** is considered most significant heading and is rendered in the largest font.
- Each successive heading element (i.e., h2, h3, etc.) is rendered in a progressively **smaller font**.

Headings

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.2: heading.html -->
4  <!-- Heading elements h1 through h6. -->
5  <html>
6     <head>
7        <meta charset = "utf-8">
8        <title>Headings</title>
9     </head>
10
11    <body>
12       <h1>Level 1 Heading</h1>
13       <h2>Level 2 heading</h2>
14       <h3>Level 3 heading</h3>
15       <h4>Level 4 heading</h4>
16       <h5>Level 5 heading</h5>
17       <h6>Level 6 heading</h6>
18    </body>
19 </html>
```

Fig. 2.2 | Heading elements h1 through h6. (Part 1 of 2.)

Headings

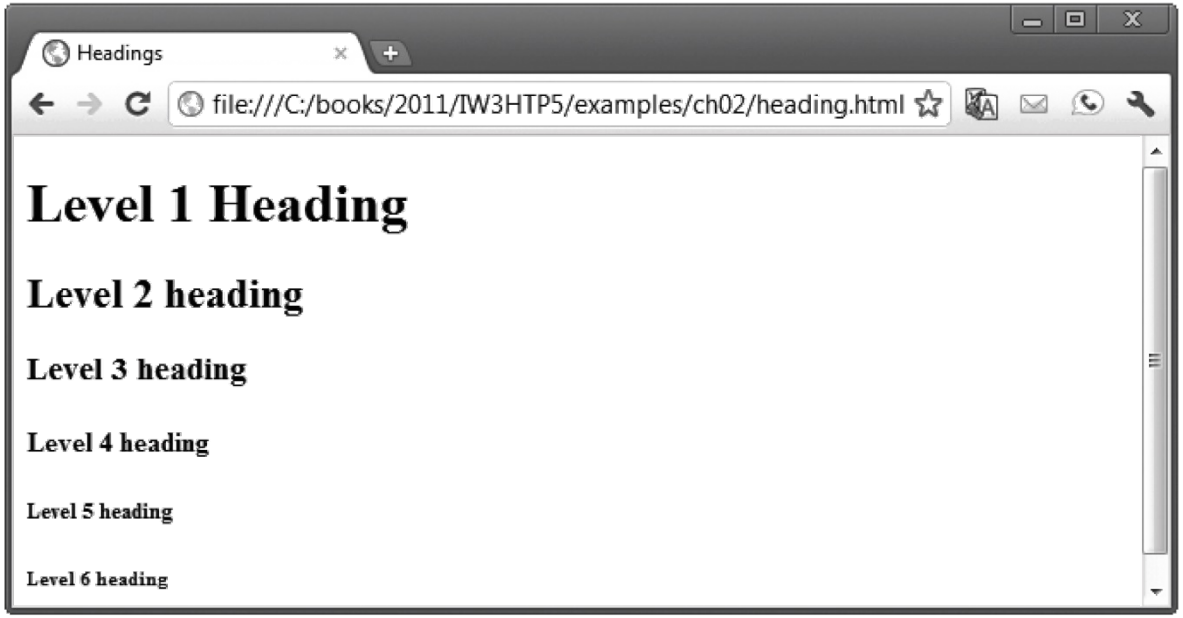


Fig. 2.2 | Heading elements h1 through h6. (Part 2 of 2.)

Images

- To add an image to a webpage, we use the **<img> element**.
- The img element's **src attribute** specifies image's location
- The img element must have an alt attribute, contains text displayed if the client cannot render the image
  - The alt attribute makes web pages more accessible to users with disabilities, mainly vision impairments
  - Width and height are optional attributes
    - If omitted, the browser uses the image's actual width and height
    - Images are measured in pixels

Images: Alt Attribute

Why an alt attribute is important?

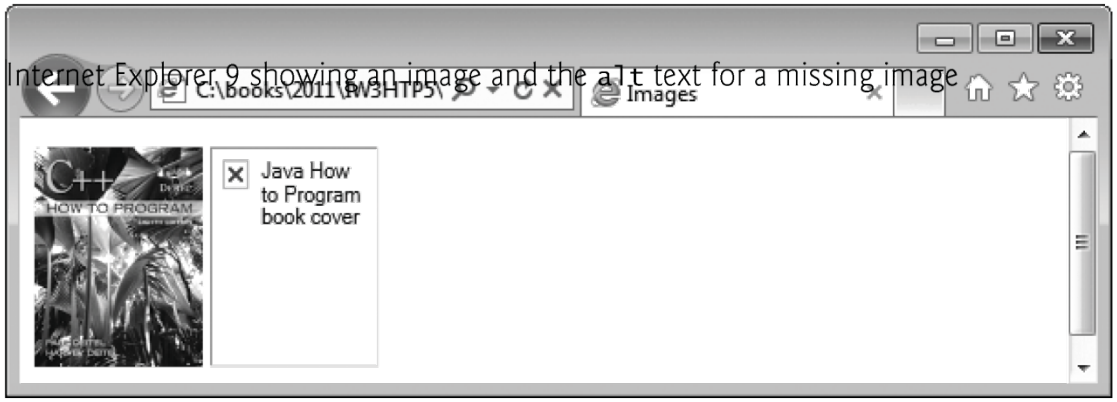
- A browser may not be able to render an image.
- If a browser cannot render an image, the browser displays the alt attribute's value.
- The alt attribute is also important for accessibility—speech synthesizer software can speak the alt attribute's value so that a visually impaired user can understand what the browser is displaying. For this reason, the alt attribute should describe the image's contents.

Images

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.6: picture.html -->
4  <!-- Including images in HTML5 files. -->
5  <html>
6     <head>
7        <meta charset = "utf-8">
8        <title>Images</title>
9     </head>
10
11    <body>
12       <p>
13          <img src = "cpphtp.png" width = "92" height = "120"
14             alt = "C++ How to Program book cover">
15          <img src = "jhtp.png" width = "92" height = "120"
16             alt = "Java How to Program book cover">
17       </p>
18    </body>
19 </html>
```

Fig. 2.6 | Including images in HTML5 files. (Part 1 of 2.)

Images



Internet Explorer 9 showing an image and the `alt` text for a missing image

Fig. 2.6 | Including images in HTML5 files. (Part 2 of 2.)

Videos

- To show a video in HTML, use the **<video> element**

```
<video width="320" height="240" autoplay muted controls>
  <source src="movie.mp4" type="video/mp4">
  <source src="movie.ogg" type="video/ogg">
  Your browser does not support the video tag.
</video>
```

- The **controls attribute** adds video controls, like play, pause, and volume.
- It is a good idea to always include **width and height attributes**. If height and width are not set, the page might flicker while the video loads.
- The **<source> element** allows you to specify alternative video files which the browser may choose from. The browser will use the first recognized format.
- Some useful attributes can be added. For example, '**autoplay**' to indicate whether the video should auto play at start. Also, the '**muted**' indicates whether a video should begin with a muted audio.

Links

- A hyperlink references or links to other resources, such as HTML5 documents and images.
- Web browsers typically underline text hyperlinks and color them blue by default.
- Links are created using the a (anchor) element. Attribute href (hypertext reference) specifies a resource's location, such as web page or location in a web page, a file or an e-mail address
- If the web server cannot locate a requested document, it returns an error indication to the web browser (known as a 404 error), and the browser displays a web page with an error message.

Links

Hyperlinking to an E-Mail Address

- Anchors can link to an e-mail address using mailto: URL
- When a user clicks this type of anchored link, most browsers launch the default e-mail program (e.g., Mozilla Thunderbird, Microsoft Outlook or Apple Mail) to enable the user to write an e-mail message to the linked address.

```
<a href = "mailto:deitel@deitel.com">
```

Links

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.3: links.html -->
4  <!-- Linking to other web pages. -->
5  <html>
6     <head>
7        <meta charset = "utf-8">
8        <title>Links</title>
9     </head>
10
11    <body>
12       <h1>Here are my favorite sites:</h1>
13       <p><strong>Click a name to visit that site.</strong></p>
14
15       <!-- create four text hyperlinks -->
16       <p><a href = "http://www.facebook.com">Facebook</a></p>
17       <p><a href = "http://www.twitter.com">Twitter</a></p>
18       <p><a href = "http://www.foursquare.com">Foursquare</a></p>
19       <p><a href = "http://www.google.com">Google</a></p>
20    </body>
21 </html>
```

Fig. 2.3 | Linking to other web pages. (Part 1 of 2.)

Links

Fig. 2.3 | Linking to other web pages. (Part 2 of 2.)

Paths

- *img src*, and *a href*, and *link href* can all take either relative or absolute paths to the resource:
  - About
  -
  - <link rel="stylesheet" href="css/style.css"/>
- An **absolute path** is a complete address that starts with the protocol (such as "http://" or "https://"), followed by the domain name or IP address of the server, and then the path to the resource.

Paths

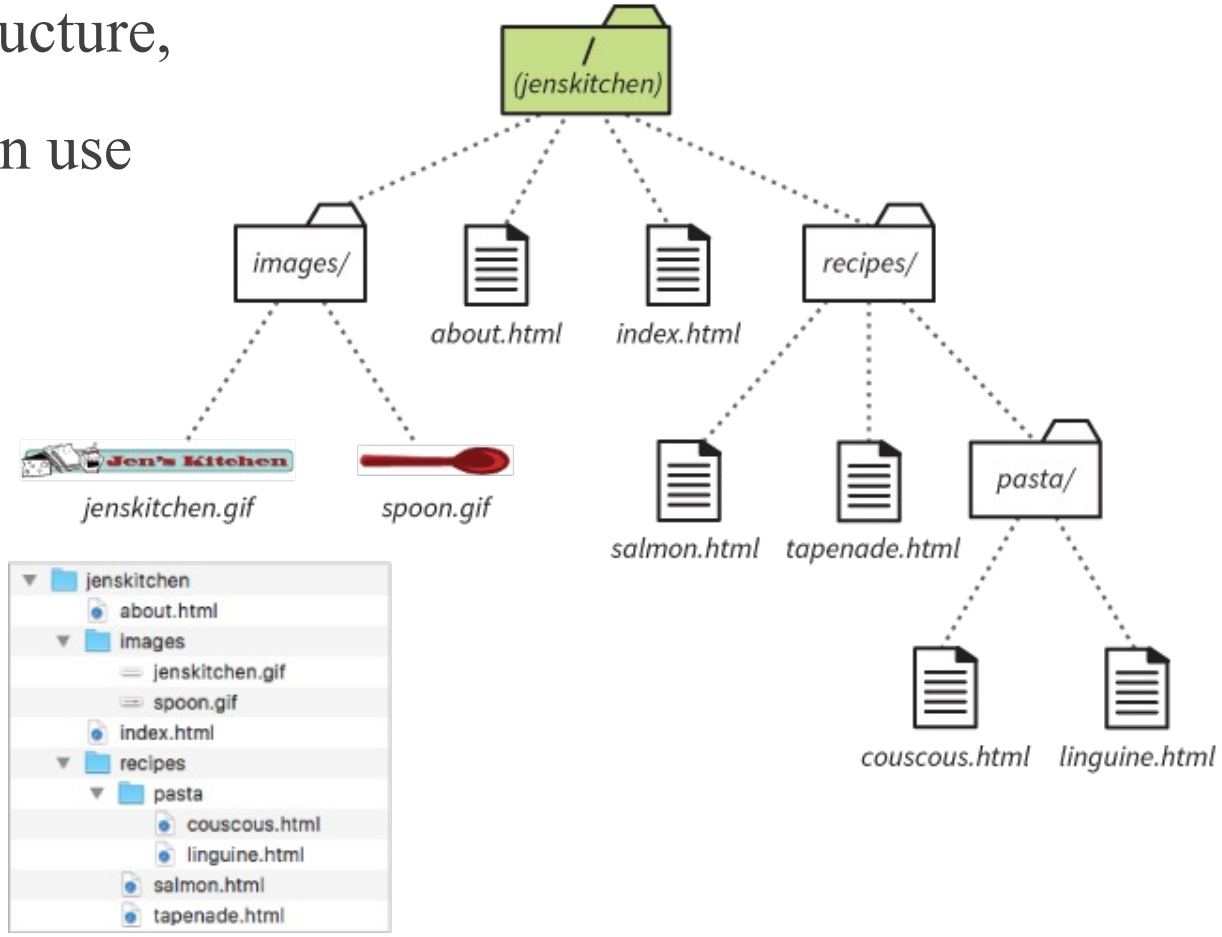
- A **relative path** specifies the location of a resource relative to the current page or file. Relative paths do not include the protocol or domain name, and instead start with one or more directory names or "../" to specify the parent directory.
- Relative paths are often used within a website to link to other pages or resources, since they can be shorter and more flexible than absolute paths. However, they can become problematic if the file structure of the website changes, since the relative path may no longer be correct.

Paths: Example

Given this folder structure,

lets see how we can use

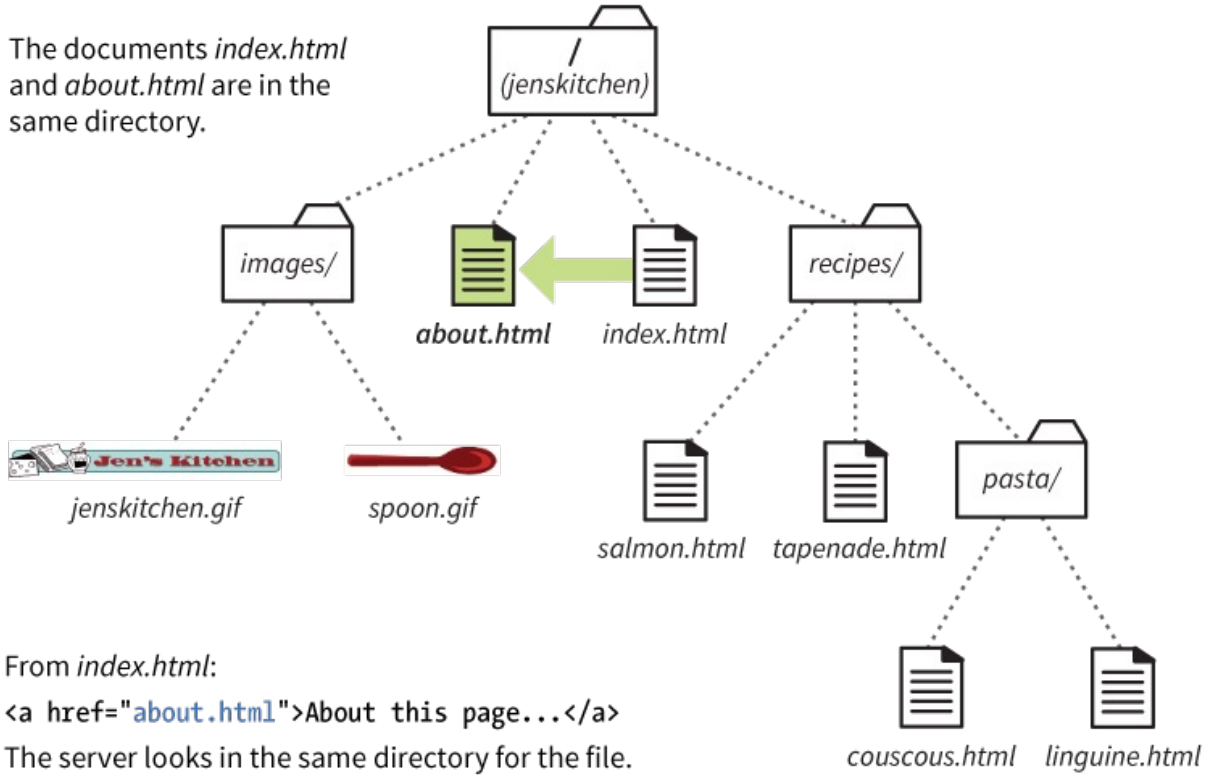
relative links



Linking in the Same Directory

When the linked document is in the same directory as the current document, just provide its filename:

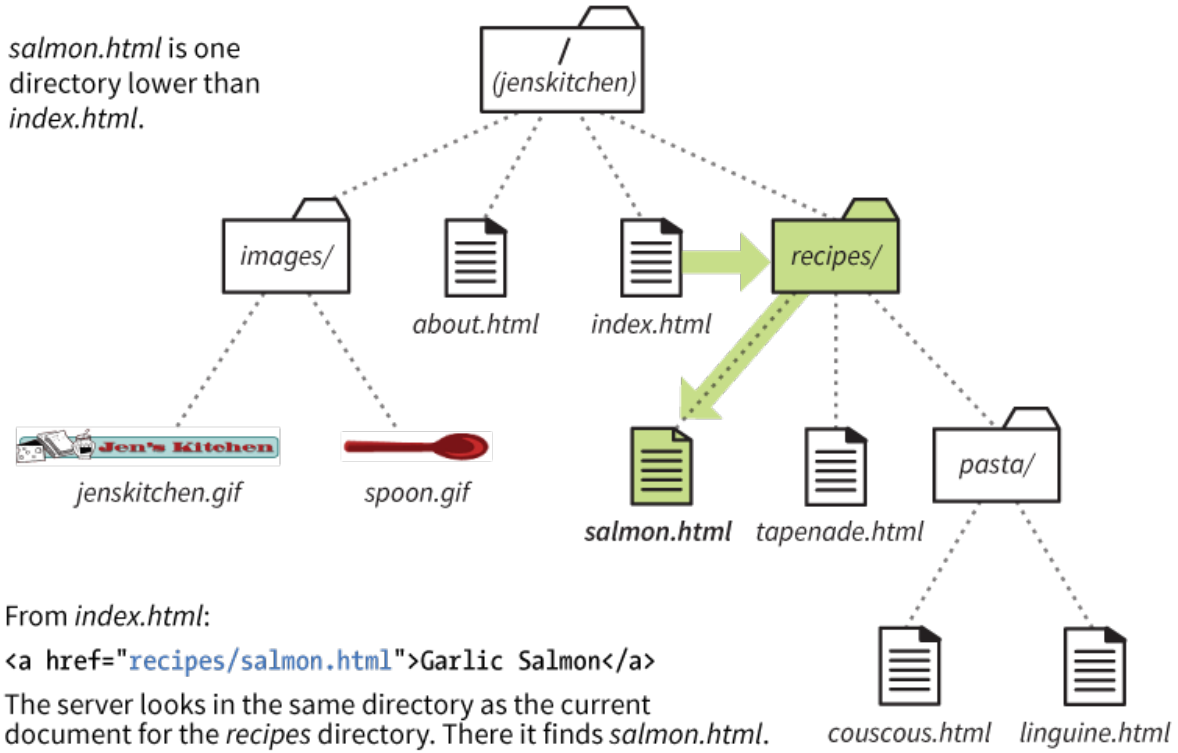
href="about.html"



Linking to a lower Directory

If the linked file is in a directory, include the directory name in the path.

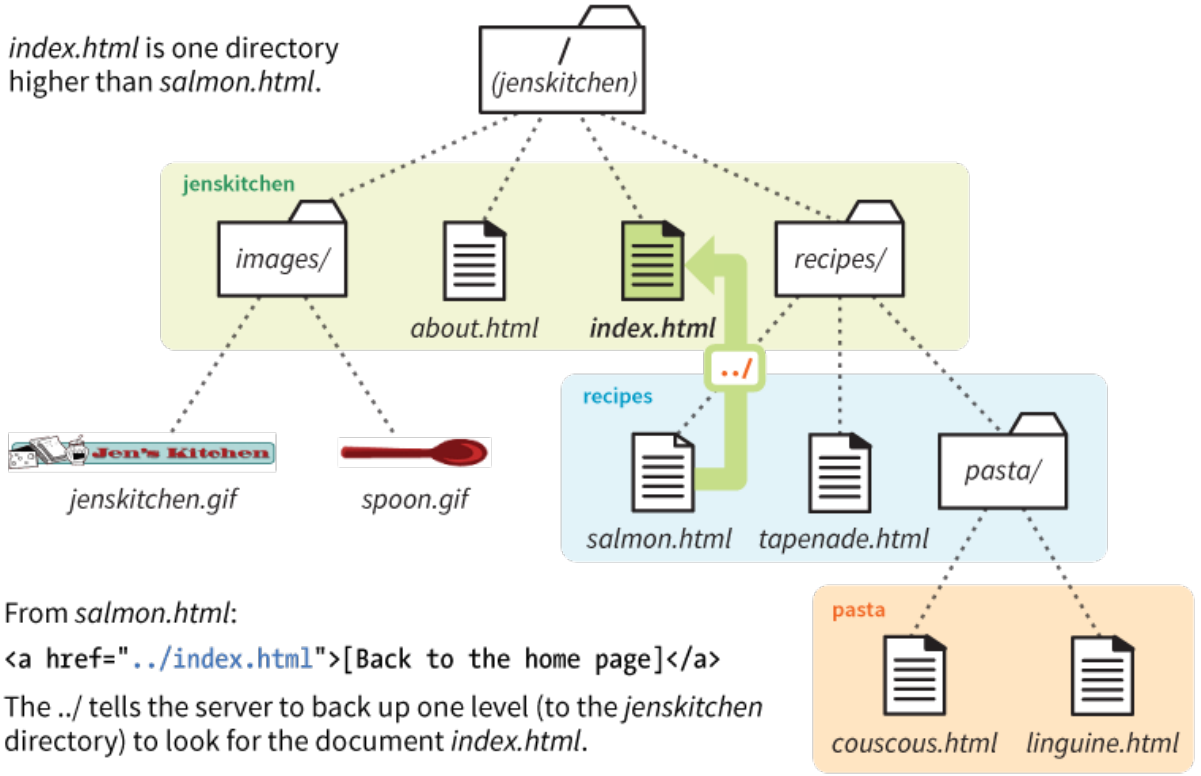
href="recipes/salmon.html"



Linking to a higher Directory

To back up a level, the ../ stands in for the name of the higher directory:

href="../index.html"



Lists

- Lists may be nested to represent hierarchical relationships, as in a multi-level outline.
- The **ordered-list element** ol creates a list in which each item begins with a number.
- The **unordered list element** ul creates a list in which each item in the list begins with a bullet symbol (typically a disc)
- Each **entry is an** li (list item) element. Most web browsers render these elements with a line break and a bullet symbol at the beginning of the line.

Lists

```
1  <!DOCTYPE html>
2
3  <!-- Fig. 2.10: links2.html -->
4  <!-- Unordered list containing hyperlinks. -->
5  <html>
6     <head>
7        <meta charset = "utf-8">
8        <title>Links</title>
9     </head>
10
11    <body>
12       <h1>Here are my favorite sites</h1>
13       <p><strong>Click on a name to go to that page</strong></p>
14
15       <!-- create an unordered list -->
16       <ul>
17          <!-- the list contains four list items -->
18          <li><a href = "http://www.youtube.com">YouTube</a></li>
19          <li><a href = "http://www.wikipedia.org">Wikipedia</a></li>
20          <li><a href = "http://www.amazon.com">Amazon</a></li>
21          <li><a href = "http://www.linkedin.com">LinkedIn</a></li>
22       </ul>
23    </body>
24 </html>
```

Fig. 2.10 | Unordered list containing hyperlinks. (Part 1 of 2.)

Lists

Fig. 2.10 | Unordered list containing hyperlinks. (Part 2 of 2.)

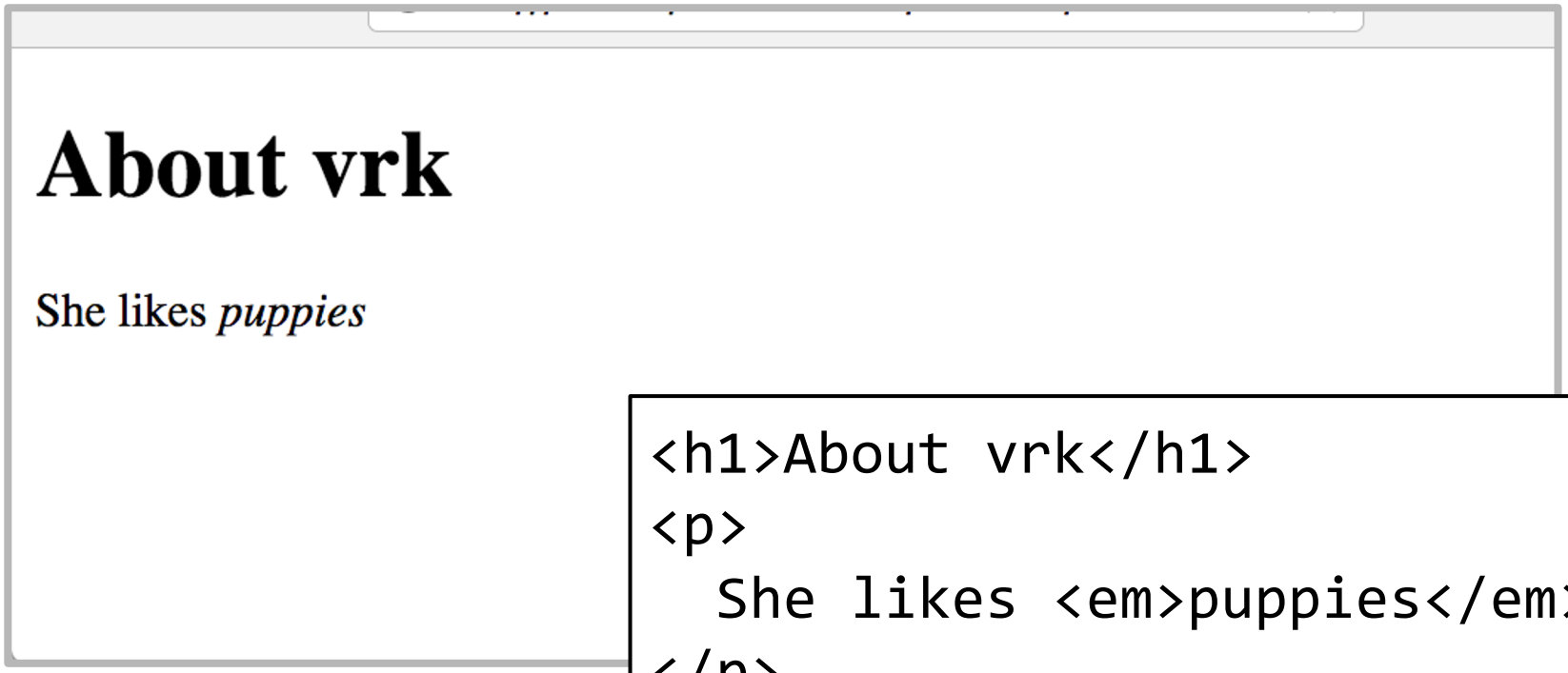
Types of HTML elements

- The way HTML elements are displayed in the browser can be viewed as one of three ways: :
  - **block**: large blocks of content, has height and width
    - <p>, <h1>, <ol>, <ul>, <table>
  - **inline**: small amount of content, no height or width
    - <a>, , ,
  - **inline block**: inline content with height and width
    - <img>

Block Elements

- Examples: <p>, <h1>, <ol>, <ul>, <table>
  - Take up the full width of the page (flows top to bottom)
  - Have a height and width
  - Can have block or inline elements as children

Example for block elements

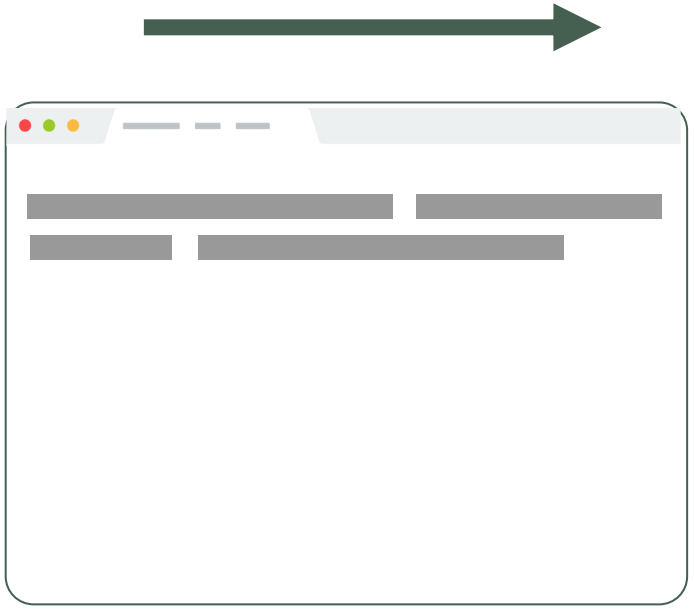



```
<h1>About vrk</h1>
<p>
  She likes <em>puppies</em>
</p>
```

Inline Elements

- Examples: <a>, ,
- Take up only as much width as needed

  (flows left to right)
- **Cannot** have height and width
- **Cannot** have a block element child



Inline-block Elements

- Examples: <img>
- Width is the size of the content, it takes only as much space as needed (flows left to right)
- Can have height and width
- Can have a block element as a child

W3C HTML5 Validation Service

- HTML5 documents that are syntactically correct are guaranteed to render properly
- HTML5 documents that contain syntax errors may not display properly
- Validation services (e.g., validator.w3.org/#validate-by-upload) ensure that an HTML5 document is syntactically correct
- Most browsers attempt to render HTML5 documents even if they are invalid. This can lead to unexpected and undesirable results. Thus, using a validation service to confirm the validity of an HTML5 document can help address this issue

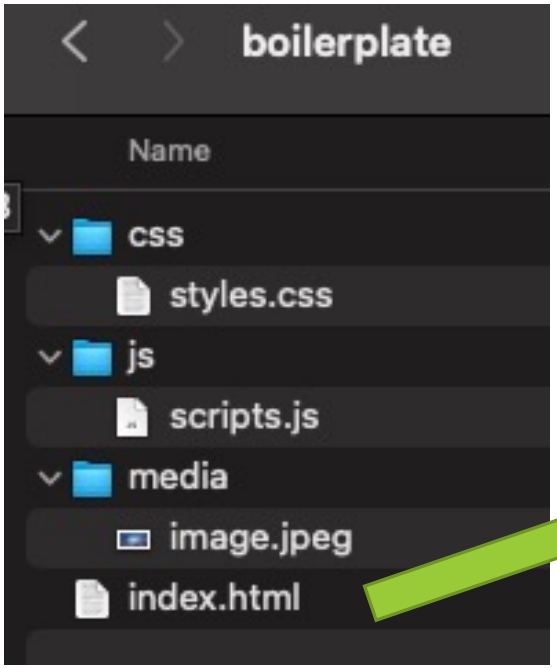
HTML5 Boilerplate

- The HTML5 Boilerplate is a set of files and directories that provide a starting point for web development projects.
- Using the HTML5 Boilerplate can save developers time and effort by providing a solid foundation for their projects, and can help ensure that their code follows best practices.
- A Boilerplate will describe the basic folder structure, along with the basic HTML code that is needed across the website.
- The provided HTML5 file should be the starting point for any new page.

HTML5 Boilerplate

- What does an HTML5 Boilerplate looks like?

1 Folder Structure



2 HTML5 Template

```
1  <!doctype html>
2
3  <html lang="en">
4
5  <head>
6    <meta charset="utf-8">
7    <meta name="viewport" content="
     width=device-width, initial-scale=1.0">
8    <title>A Basic HTML5 Template</title>
9
10   <!-- CSS here -->
11   <link rel="stylesheet" href="css/styles.css">
12
13   <!-- Scripts here -->
14   <script src="js/scripts.js"></script>
15
16 </head>
17
18 <body>
19   <h1>Hello World!</h1>
20 </body>
21 </html>
```

More about HTML

- There are still many aspects we did not cover, which you should explore such as tables, iframes, special chars, etc.
- You can find out more about this on w3school.com
- **W3Schools** is an online educational platform for web development that provides free tutorials, examples, and references on a variety of web technologies. It is a popular resource for beginners and intermediate web developers looking to learn or improve their skills.

Any questions?
Please feel free to raise your hands and ask.