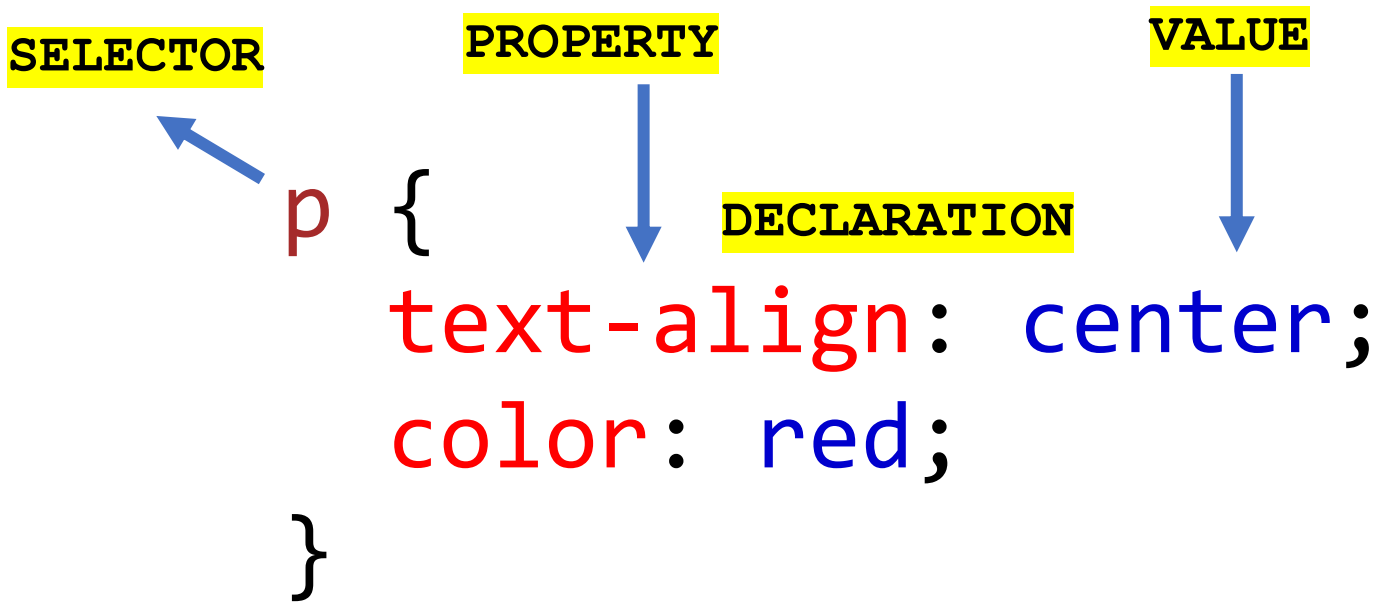
CMPU 1031 - Web Development 1

Lecturer: Dr Fatmaelzahraa Eltaher

Week 4: More on CSS

Revision - syntax



- Each CSS style declaration is made up of a property and a value.
- The property and the value are separated by a colon (:).
- Each declaration must be followed with a semi-colon (;).
- This follows the format of a **name: value;** pair.
- The property (name) identifies the feature of the element that requires styling, the value describes how this will display on the page.

ID Selector

- When an element has been given an id attribute, a CSS rule can be written for this id.
- In the stylesheet, a hash symbol (#) must be added before the ID when you use it as a selector.

HTML

```
<p id="uniqueParagraph">This paragraph has an ID
associated to it.</p>
```

CSS

```
#uniqueParagraph {
  font-family: "Helvetica";
  color: magenta;
  font-size: 20px;
}
```

This paragraph has an ID associated to it.

Click on the code to see the example in CodePen

Class Selector

- When an element has been given a class attribute, a CSS rule can be written for this.
- However, differently from ID, the name given to the class does not have to be unique to the page.
- There can be multiple elements with the same class attribute on the same page. This can be used to style multiple occurrences of the same element on the page
- In the stylesheet, a period symbol (.) must be added before the ID when you use it as a selector.

Class Selector

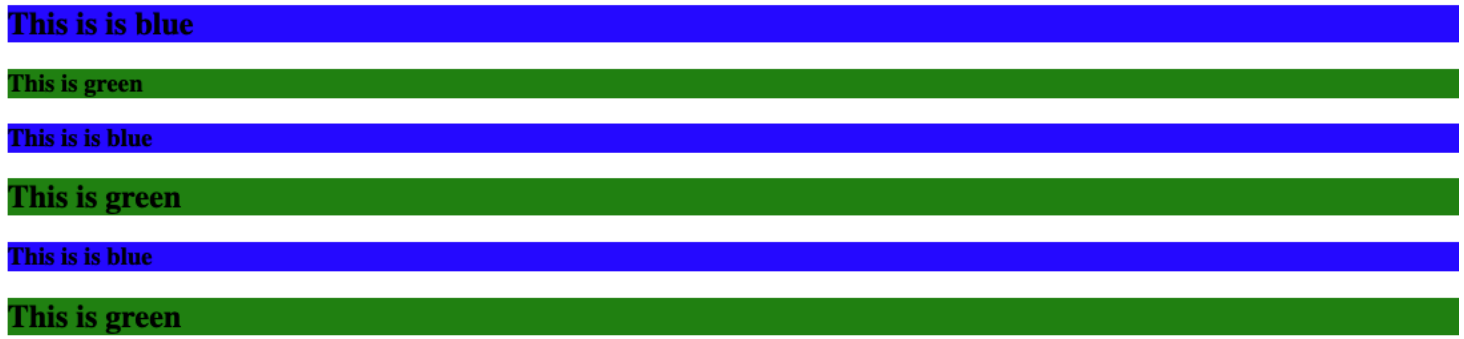
HTML

```
<h2 class="blueHeading">This is is blue</h2>
<h3 class="greenHeading">This is green</h3>
<h3 class="blueHeading">This is is blue</h3>
<h2 class="greenHeading">This is green</h2>
<h3 class="blueHeading">This is is blue</h3>
<h2 class="greenHeading">This is green</h2>
```

CSS

```
.greenHeading{
  background-color: green;
}

.blueHeading{
  background-color: blue;
}
```



Click on the code to see the example in CodePen

CSS combinators

If we want to create a **graphic representation** of a website, we need to distribute the tags according to their parent-child relationship. In this example, the body tag is the parent of p, div, p, and p, and div is the parent of p, p, and section, and section is the parent of p.

```
<body>
  <p>We will style all elements that are descendants of the div.</p>

  <div>
    <p>Paragraph 1 in the div.</p>
    <p>Paragraph 2 in the div.</p>
    <section><p>Paragraph 3 in the div.</p></section>
  </div>

  <p>Paragraph 4. Not in a div.</p>
  <p>Paragraph 5. Not in a div.</p>
</body>
```

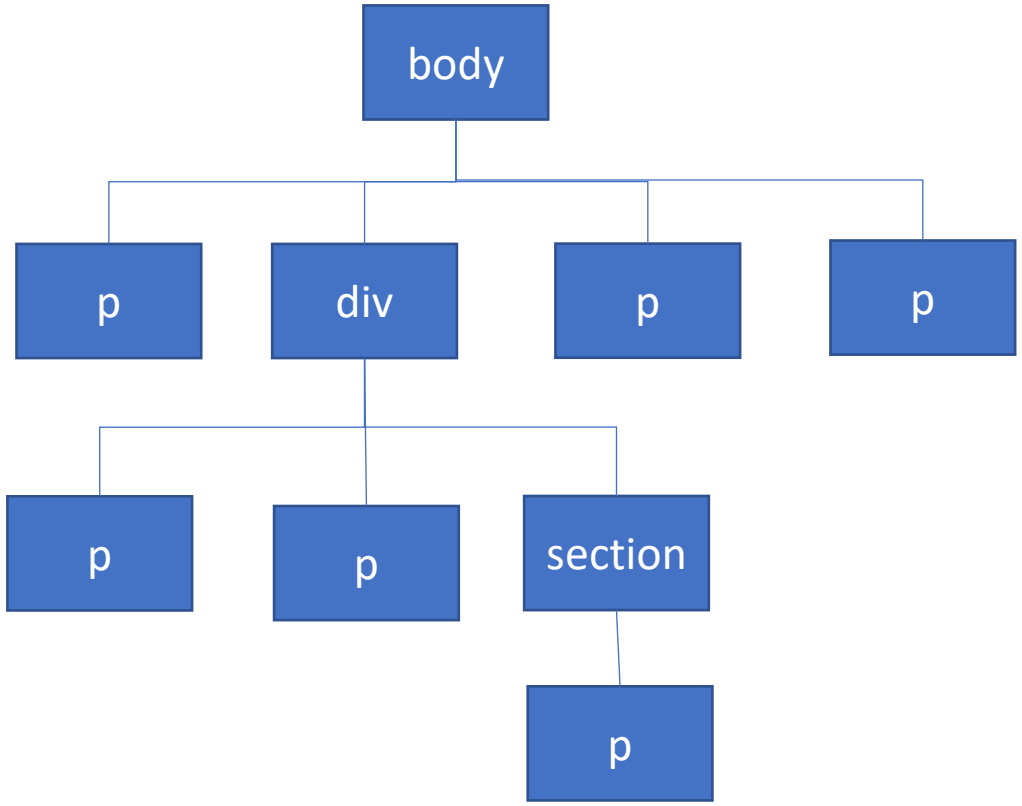
CSS combinators

If we want to create a **graphic representation** of a website, we need to distribute the tags according to their parent-child relationship. In this example, the body tag is the parent of p, div, p, and p, and div is the parent of p, p, and section, and section is the parent of p.

```
<body>
  <p>We will style all elements that are descendants of the div.</p>

  <div>
    <p>Paragraph 1 in the div.</p>
    <p>Paragraph 2 in the div.</p>
    <section><p>Paragraph 3 in the div.</p></section>
  </div>

  <p>Paragraph 4. Not in a div.</p>
  <p>Paragraph 5. Not in a div.</p>
</body>
```



Descendant selector

- We can use the relationship between HTML elements to work on our selectors.
- One example is the descendant selector, which will style all children of a certain element.

- For example: we can select every li element from the ul element and set up a background colour:

```
ul li {
  background-color: #23B433;
}
```

- Pay attention on the space between the ul and li selectors

Descendant selector

```html
<body>
  <p>We will style all elements that are descendants of the div.</p>

  <div>
    <p>Paragraph 1 in the div.</p>
    <p>Paragraph 2 in the div.</p>
    <section><p>Paragraph 3 in the div.</p></section>
  </div>

  <p>Paragraph 4. Not in a div.</p>
  <p>Paragraph 5. Not in a div.</p>
</body>
```

```css
div p {
  background-color: yellow;
}
```

We will style all elements that are descendants of the div.

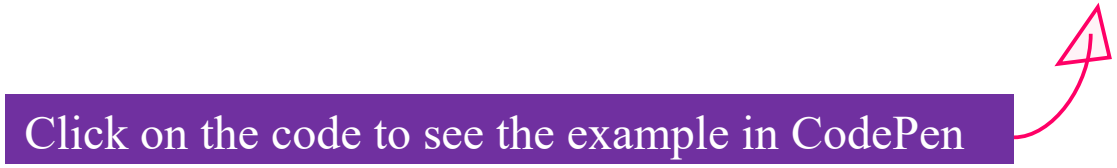
Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.



Child selector

- The child selector will style all **direct children** of a certain element.

- For example: we can select the direct **P** children of the **div** element and set up a background colour:

```
div > p {
  background-color: #23B433;
}
```

- Pay attention on the angle bracket between the ul and li selectors

Child selector

```html
<p>We will style only elements that are direct children of the div.</p>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p></section>
  <h4>This won't be coloured.</h4>
</div>

<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>
```

```css
div > p {
  background-color: yellow;
}
```

We will style only elements that are direct children of the div.

Paragraph 1 in the div.

Paragraph 2 in the div.

Paragraph 3 in the div.

This won't be coloured.

Paragraph 4. Not in a div.

Paragraph 5. Not in a div.

Click on the code to see the example in CodePen

Adjacent selector

- The adjacent selector will select an element that is a sibling of another element.
- Sibling elements must have the same parent element, and "adjacent" means "immediately following".

```
div+p{
  background-color: #23B433;
}
```

- Pay attention on the plus sign between the ul and li selectors

Adjacent selector

- Which element do you think will be coloured?

HTML

```
direct children of the div.</p>

<div>
  <p>Paragraph 1 in the div.</p>
  <p>Paragraph 2 in the div.</p>
  <section><p>Paragraph 3 in the div.</p>
</section>
  <h4>This won't be coloured.</h4>
</div>

<p>Paragraph 4. Not in a div.</p>
<p>Paragraph 5. Not in a div.</p>
```

CSS

```
div + p {
  background-color: yellow;
}
```

Attribute selectors

- HTML tags attributes can also be used to select and style an HTML element.
- It is possible to select only elements with a certain attribute, or elements with a certain attribute and a specific value.

HTML

```
<!-- It is possible to select only
elements with an especific attribute-->

<a href="www.google.com" alt="Google
website">Website</a><br>
<a href="www.google.com" >Website</a><br>
<a href="www.google.com" alt="Google
website">Website</a>

<br>
<br>
```

CSS

```
a[alt]{
  background-color: RGBA(0,255,0, 0.5);
  font-size: 20px;
}

a[alt="This is Google"]{
  background-color: RGBA(255,0,0, 0.5);
  font-size: 20px;
}
```

Website
Website
Website

Website
Website
Website

Click on the code to see the example in CodePen

Pseudo-class selectors

- A pseudo-class is used to **define a special state** of an element. These classes are not necessarily added to the element but are associated with their function
- For example, elements like <button> can have a pseudoclass **hover** so it changes the style when someone hovers the mouse
- Anchor <a> tags can have pseudoclasses like **visited** so the style changes when the user clicks on the link

HTML

```
<button href="www.tudublin.ie">Click me!</button>
<br>
<br>
<br>
<a href="www.tudublin.ie">TU Dublin</a>
```

CSS

```
button{
  background-color: RGBA(200, 50, 50, 0.6);
  width: 150px;
  height: 50px;
  font-size: 30px;
  color: white;
}

button:hover{
```

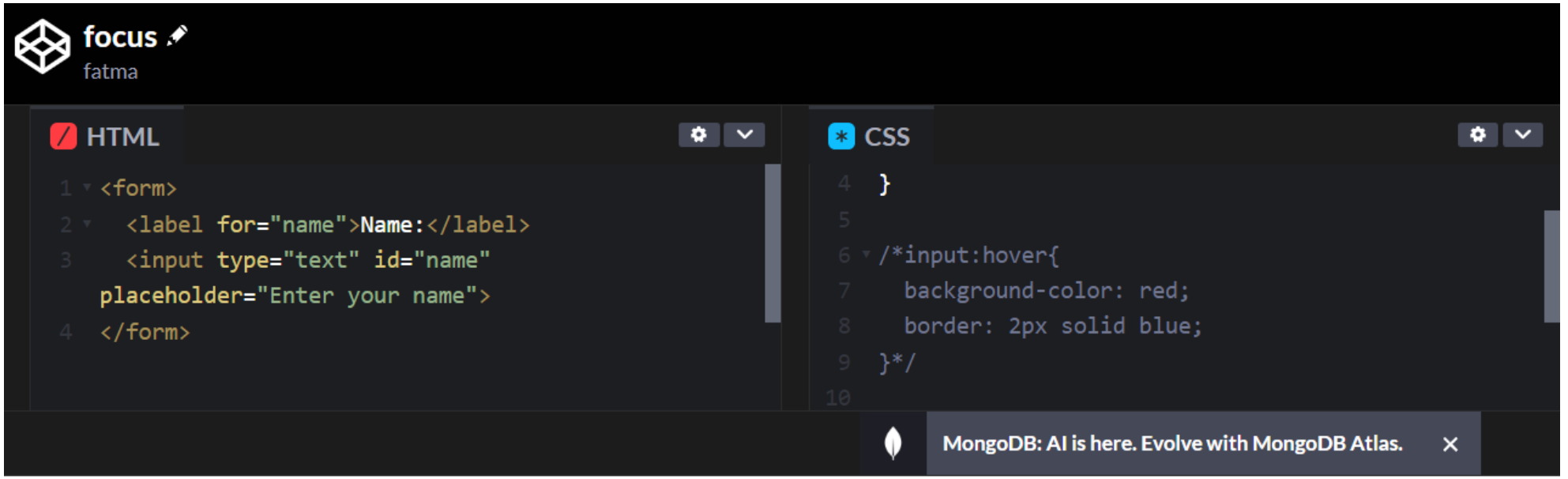
Click me!

TU Dublin

Click on the code to see the example in CodePen

Pseudo-class selectors

- When you click inside a form field (like a text box), your browser visually highlights it — this means the element is “in **focus**.”
- The :**focus** selector allows you to style elements that are currently active and ready for user input.



Click on the code to see the example in CodePen

Practice

External CSS

- When adding a CSS style sheet from an **external file**, we need to add the <link> tag to the had and provide it an **href attribute**, which will tell the browser about where the CSS file can be found:

```
<head>
        <link rel="stylesheet" href="mystyle.css">
</head>
```

- This attribute is what we call the path to the file. There are different types of path.

Absolute path

- An absolute path gives the full file path, including the root directory (the first folder accessed in order to find the file). For example:

/Users/mrocha/Desktop/WebDevWeek2/images/dinosaurBanner.jpg

- For a website, it includes the domain name. For example:

https://www.tudublin.ie/research/innovation-and-enterprise/tu-dublin-innovation/

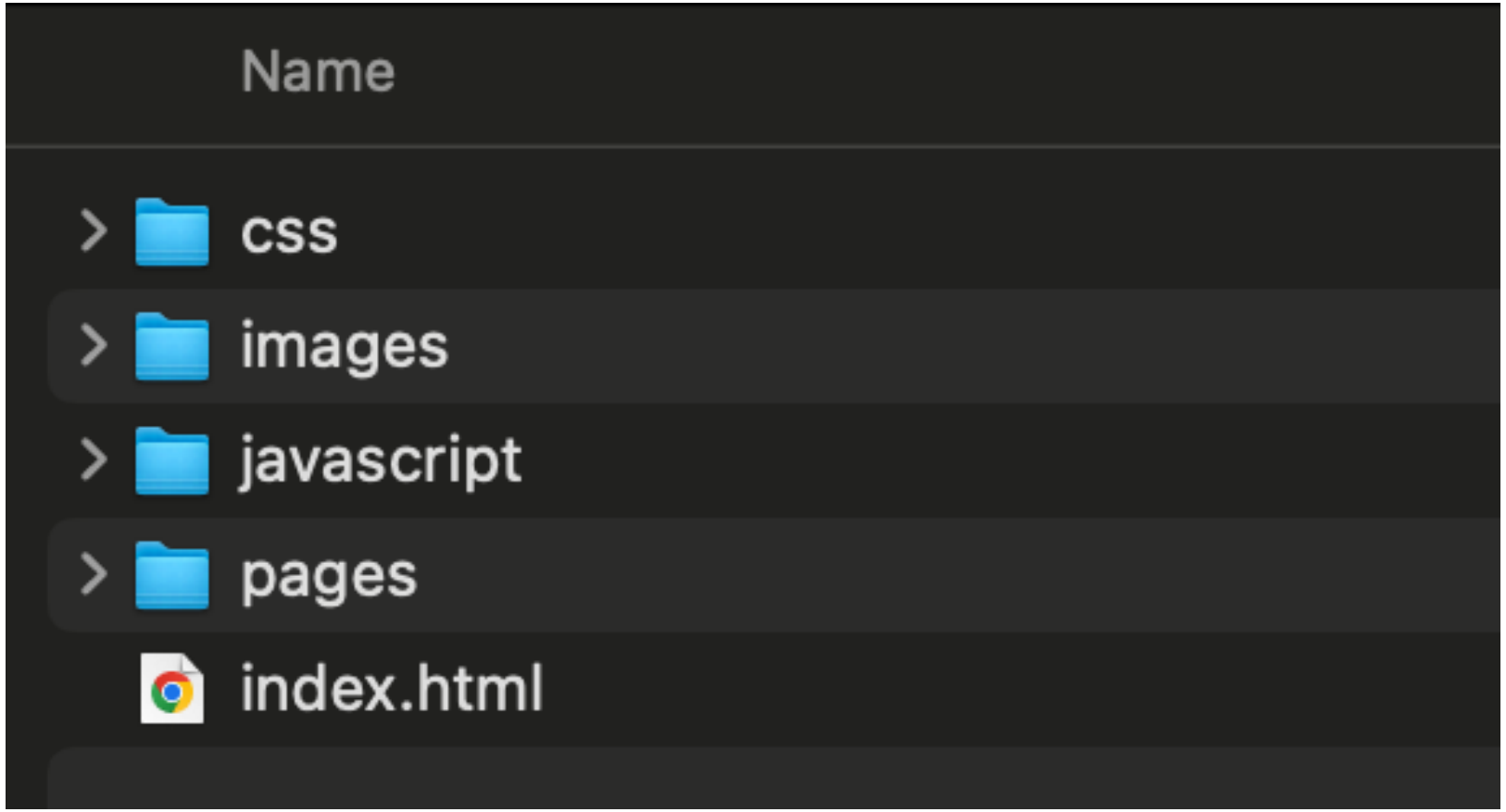
Files, folders and path

- When creating your website, it is a good practice to have a default name for the first page. Usually, we name it “index.html”

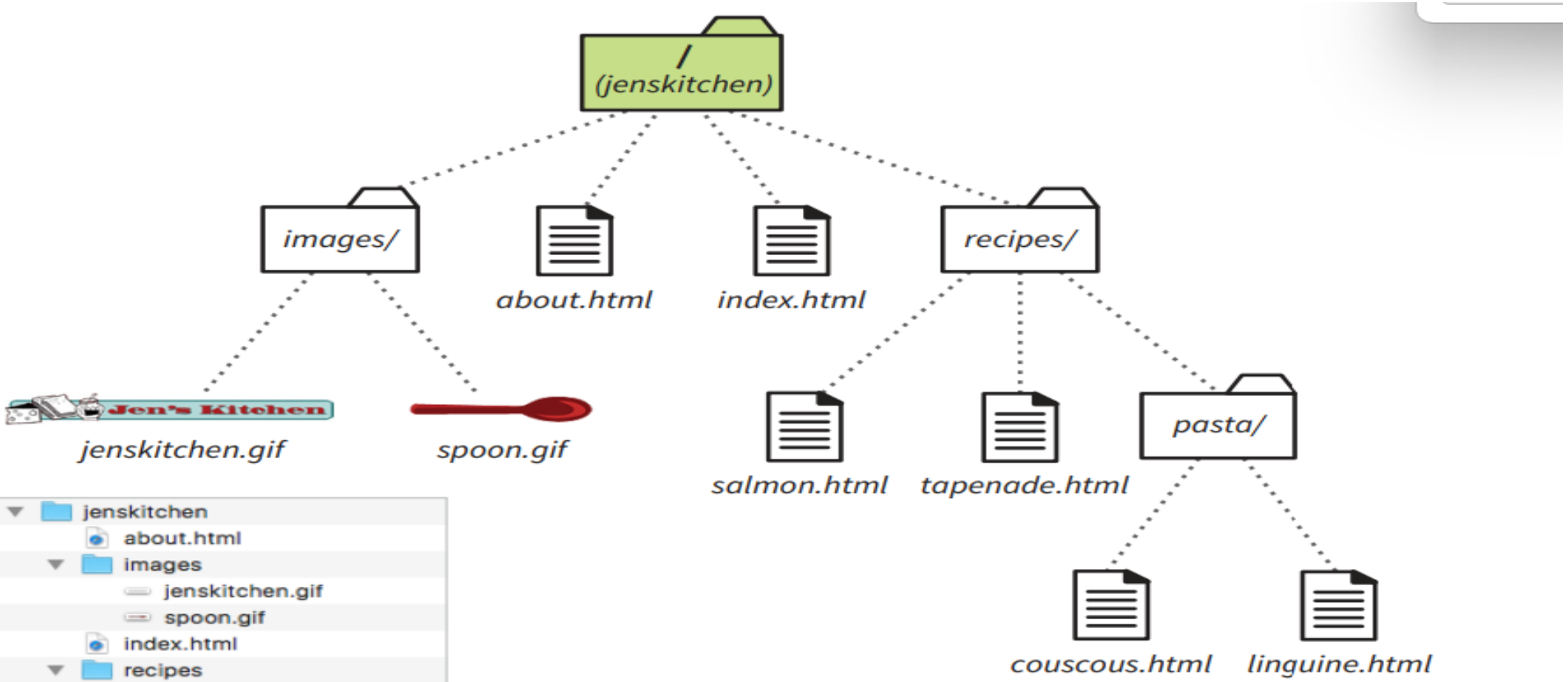
- When creating a web application it is important to structure the project folder.
- Similar content items should be grouped together in folders.
- Your html is usually placed in the root directory

- For a website, you should have subdirectories such as: css, scripts, images, vídeos.

Sample folder structure



Sample folder structure



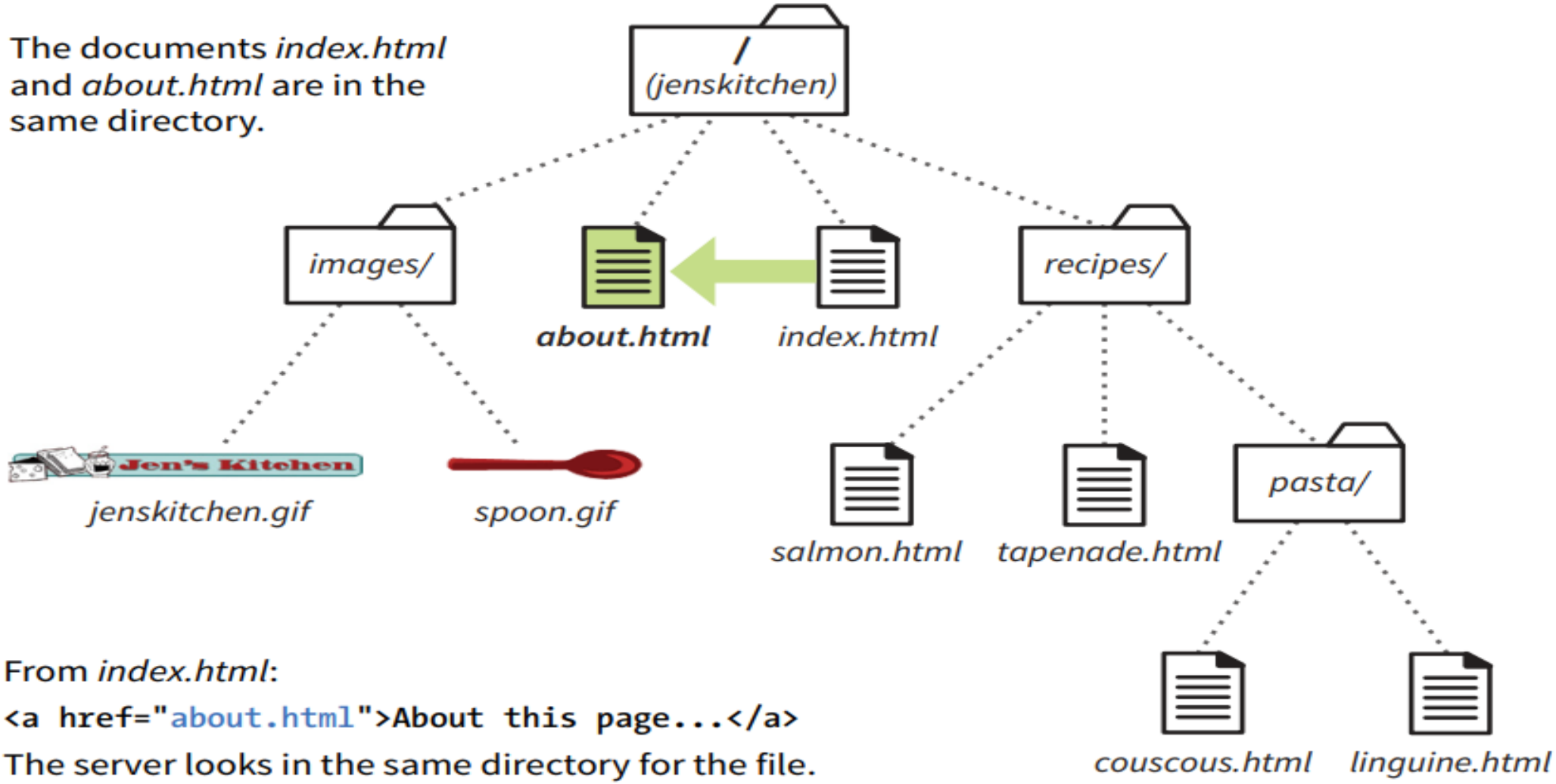
Robbins, J. N. (2018). *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics* (5th ed.). O'Reilly Media.

Relative path

- Relative paths do not use the full path or the domain name. Some basic rules:

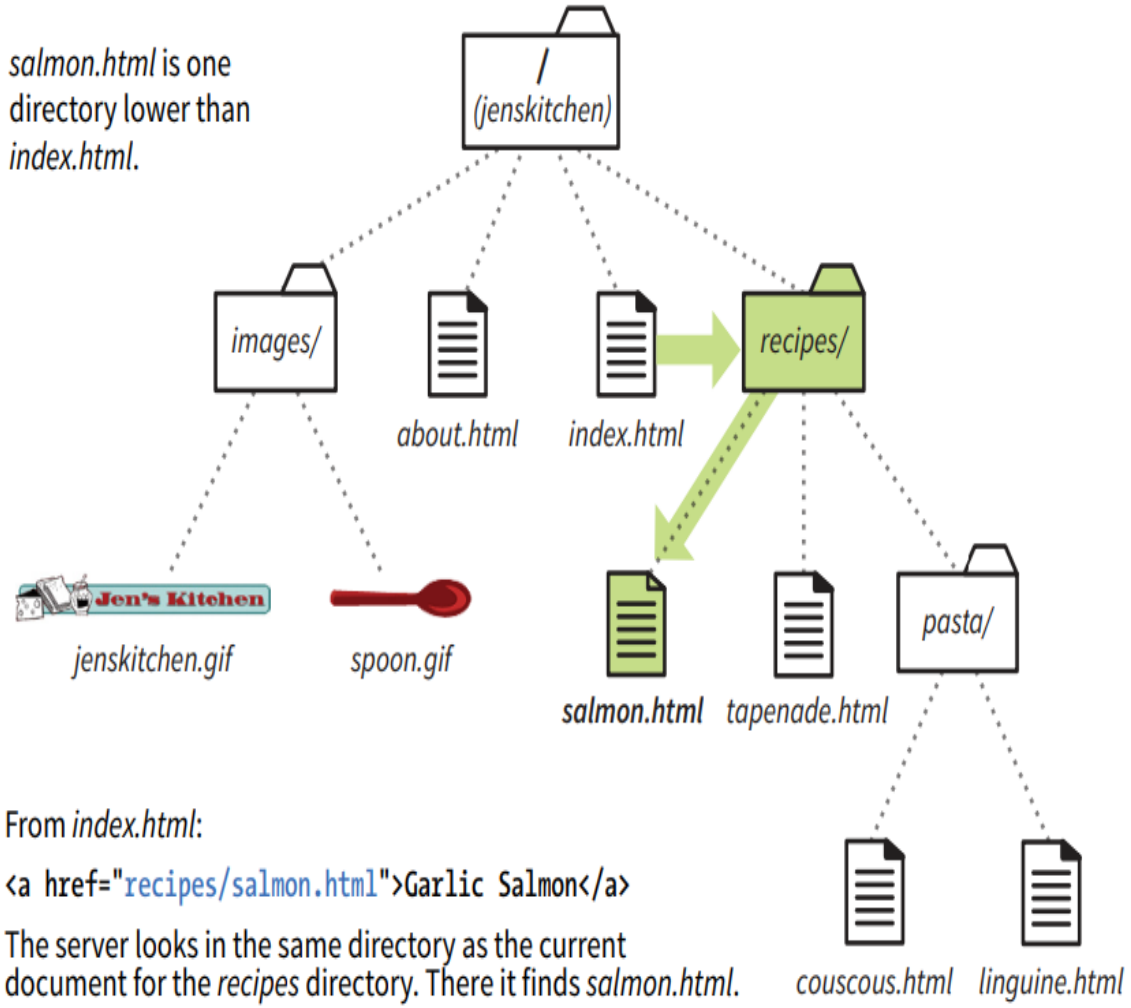
1. To link to a target file **in the same directory** as the HTML file, just use the filename, e.g. **my-image.jpg**.
2. To reference a **file in a subdirectory**, write the directory name in front of the path, plus a forward slash, e.g. **subdirectory/my-image.jpg**.
3. To link to a target file in the **directory above** the HTML file, write two dots. For example, if index.html was inside a **subfolder** and my-image.jpg was inside the main folder, you **need to go back one folder**, so you could reference my-image.jpg from index.html using **../my-image.jpg**.
4. You can combine these as much as you like, for example ../subdirectory/another-subdirectory/my-image.jpg.

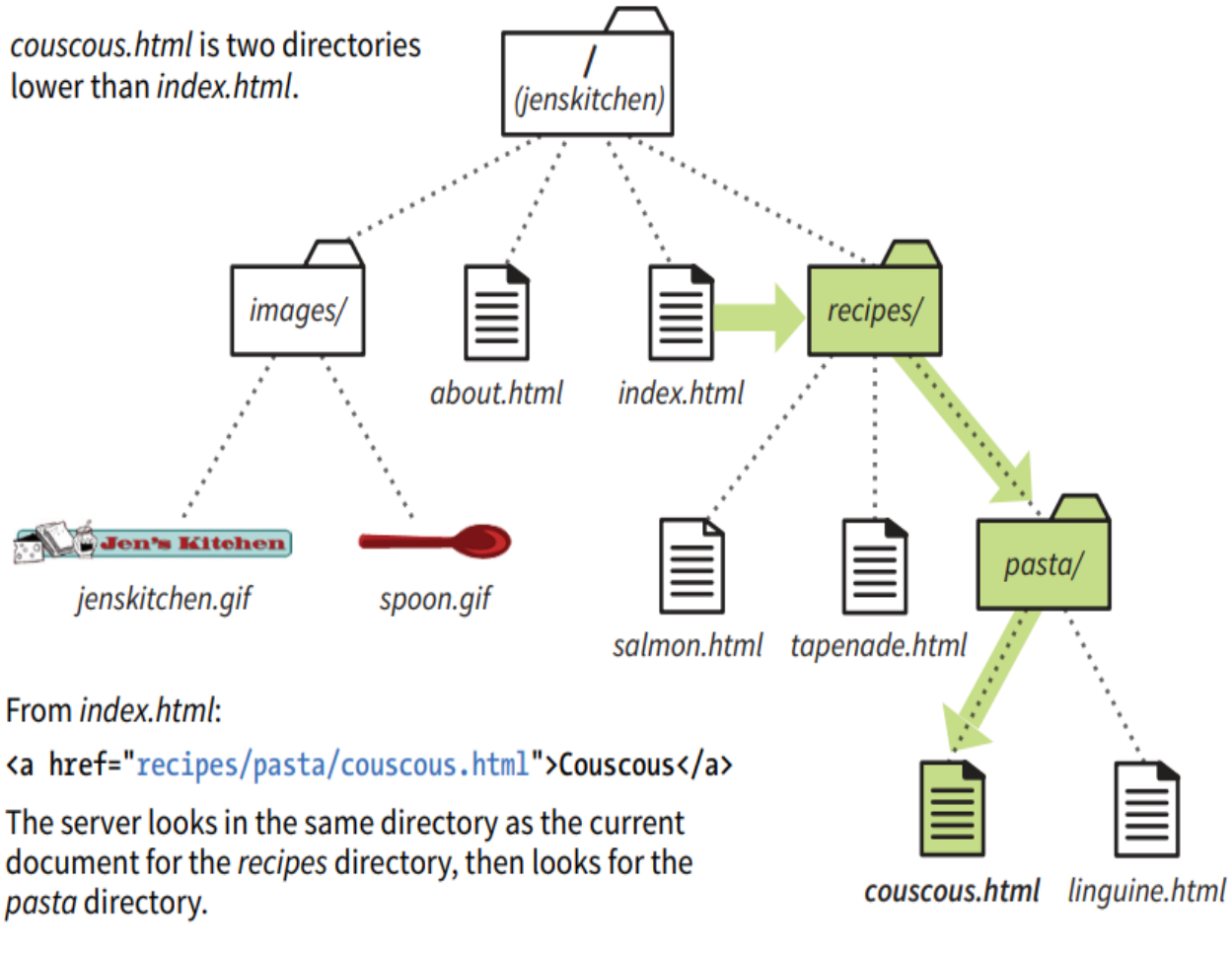
Relative path



Robbins, J. N. (2018). *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics* (5th ed.). O'Reilly Media.

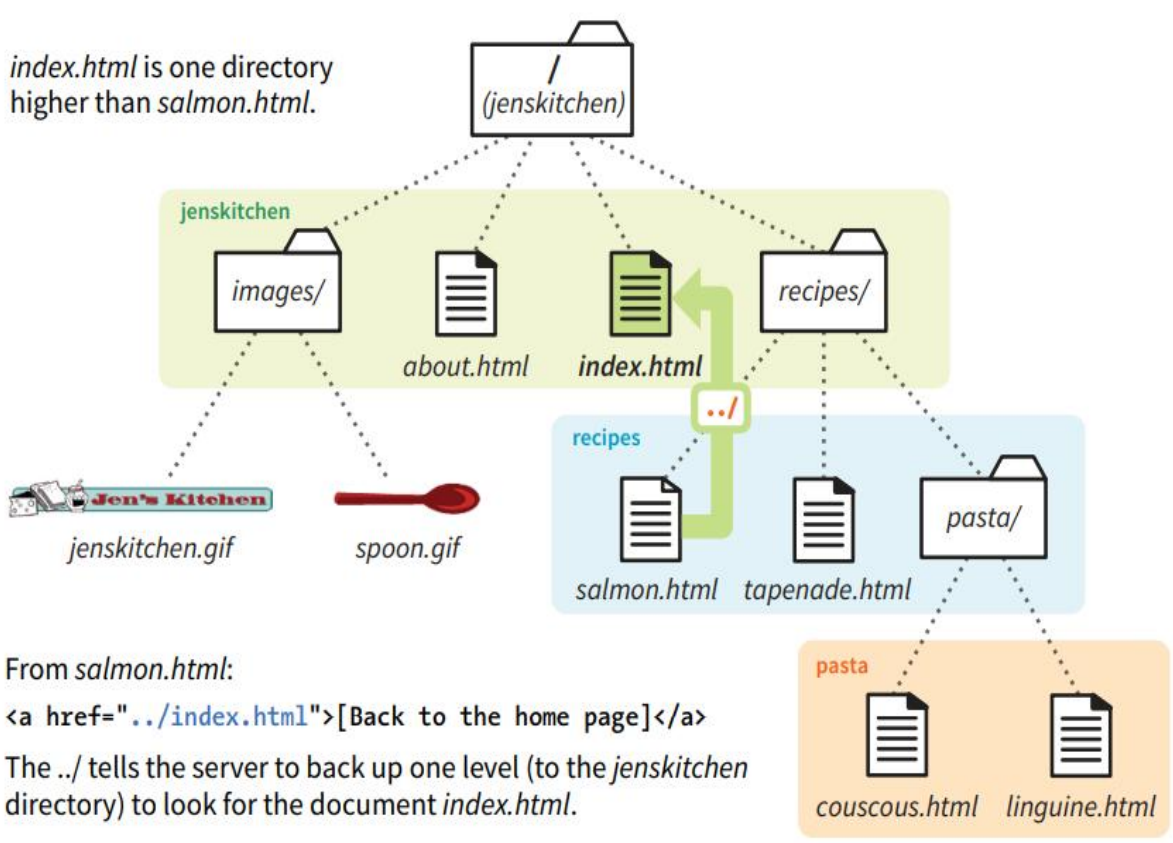
Relative path





Robbins, J. N. (2018). *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics* (5th ed.). O'Reilly Media.

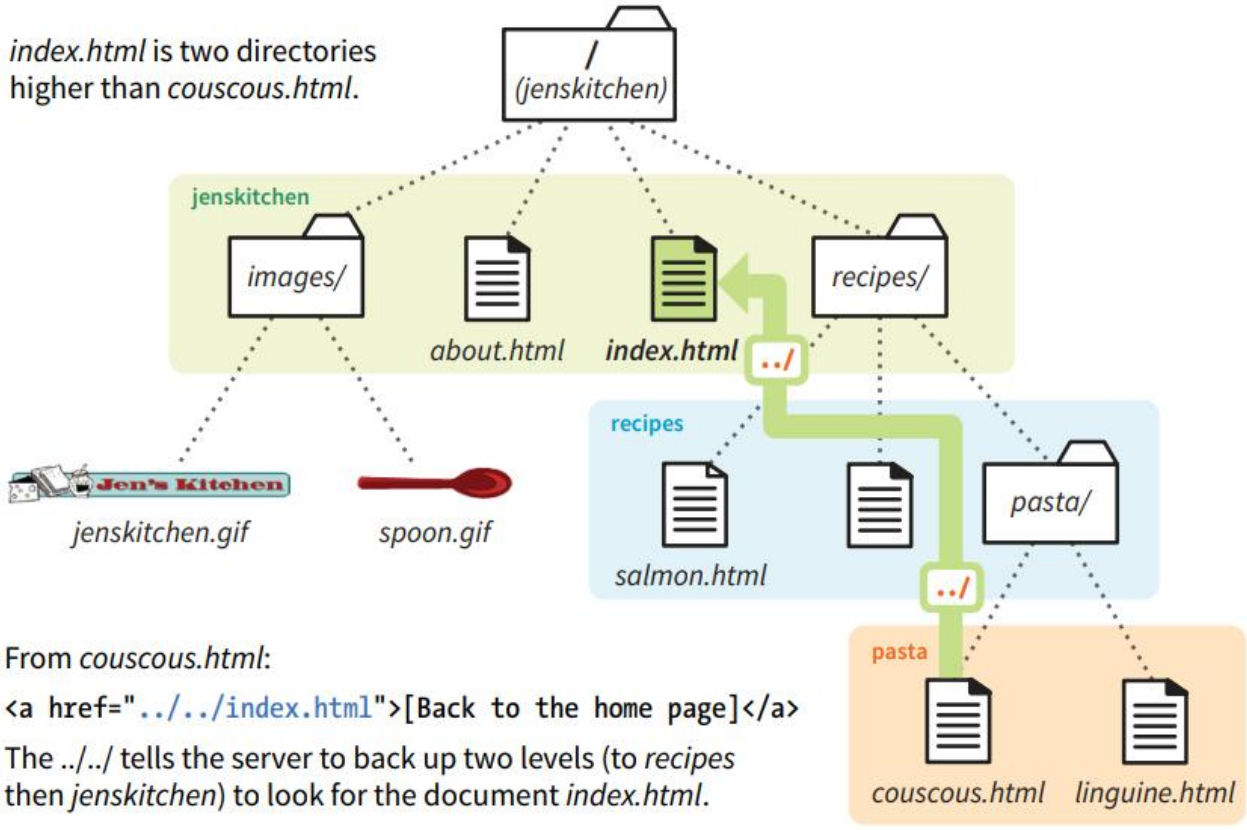
Relative path



From *salmon.html*:

```
<a href="../index.html">[Back to the home page]</a>
```

The ../ tells the server to back up one level (to the *jenskitchen* directory) to look for the document *index.html*.



From *couscous.html*:

```
<a href="../../index.html">[Back to the home page]</a>
```

The ../../ tells the server to back up two levels (to *recipes* then *jenskitchen*) to look for the document *index.html*.

Robbins, J. N. (2018). *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics* (5th ed.). O'Reilly Media.

Practice

💬 You have the following HTML file saved in:
project/pages/about.html
and your image file is saved in:
project/images/logo.png

✅ **Which is the correct relative path to link the image inside about.html?**

<img src="_____" alt="Logo">

Choices:

a) /images/logo.png
b) ../images/logo.png
c) pages/images/logo.png
d) ./images/logo.png