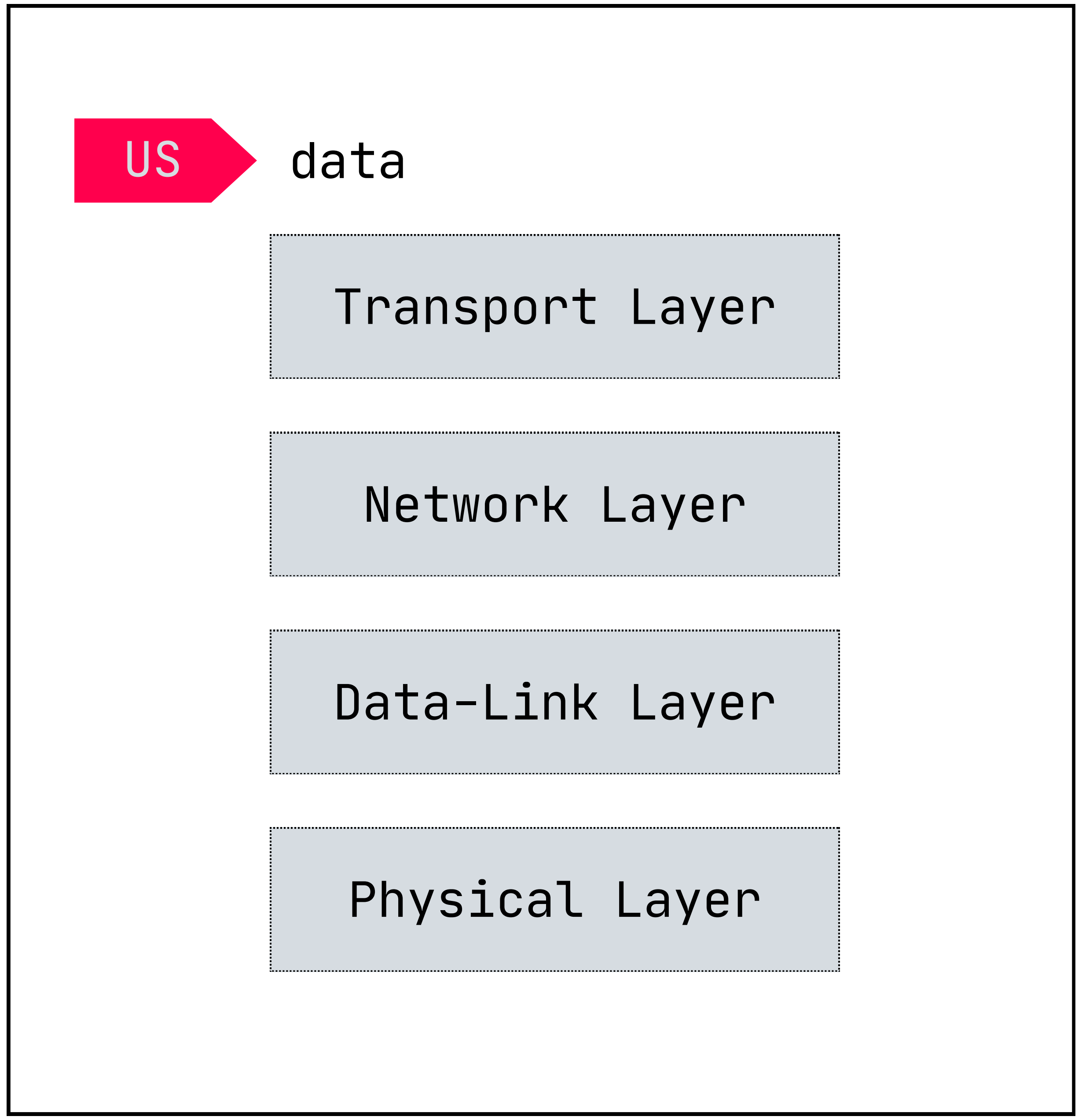
THE WEB



US

data

Transport Layer

Network Layer

Data-Link Layer

Physical Layer

Using This Stack Applications
can communicate with each others

Let’s build upon it to build a system for sharing information

The system unlike the chat server we thought of won’t build the programs it will just define a protocol for any program want to join our system to follow.

The protocol is the **HTTP** protocol

Hyper text transfer Protocol

HTTP protocol defines two main two entities

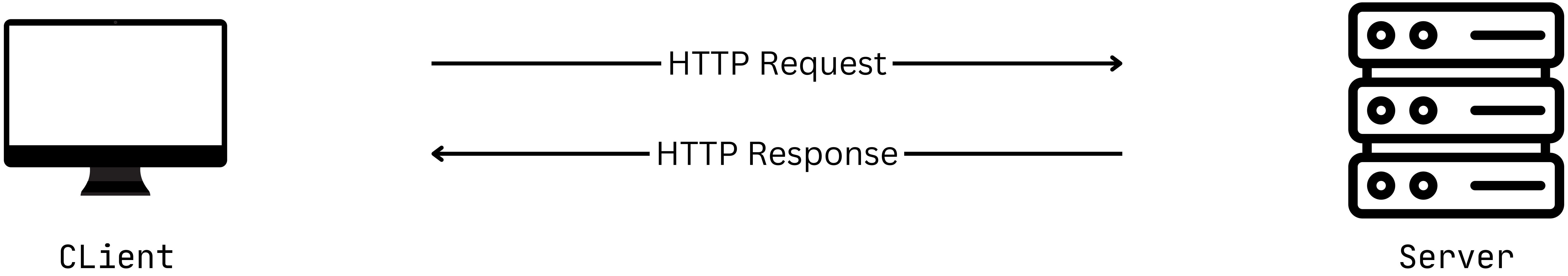
CLient

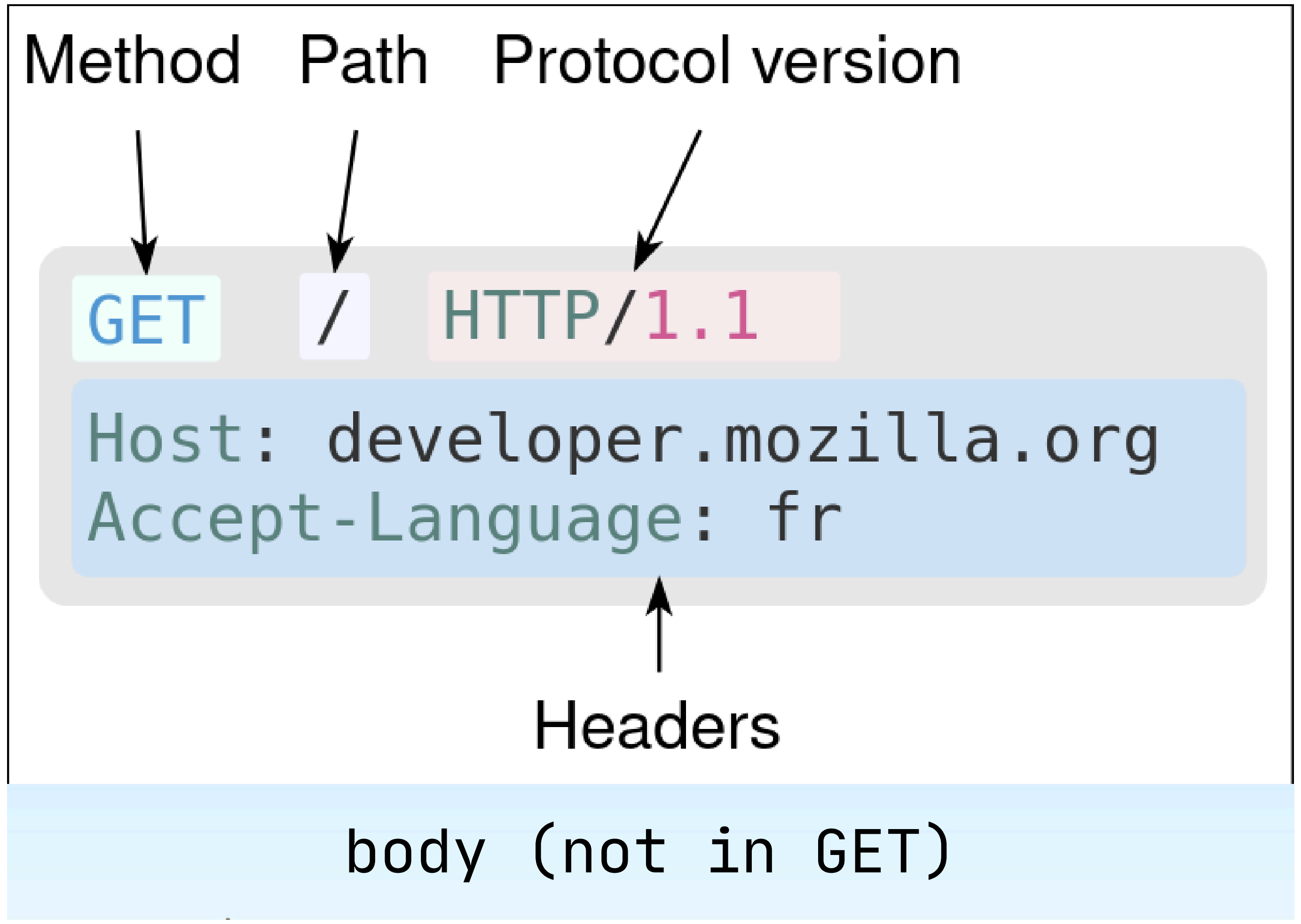
want information

Server

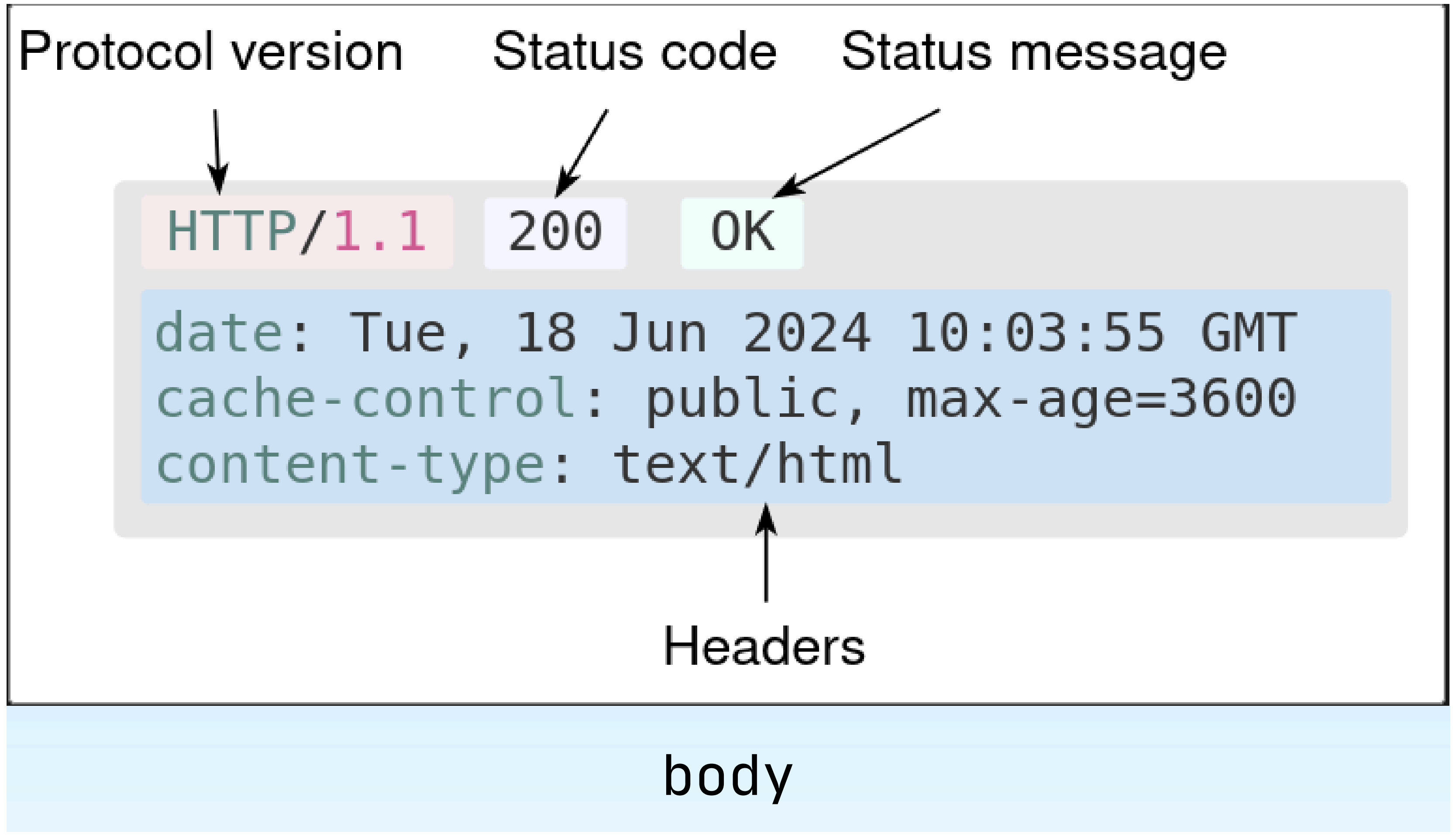
has information

Communication Between Client and Server





```
GET  /  HTTP/1.1
Host: developer.mozilla.org
Accept-Language: fr
```



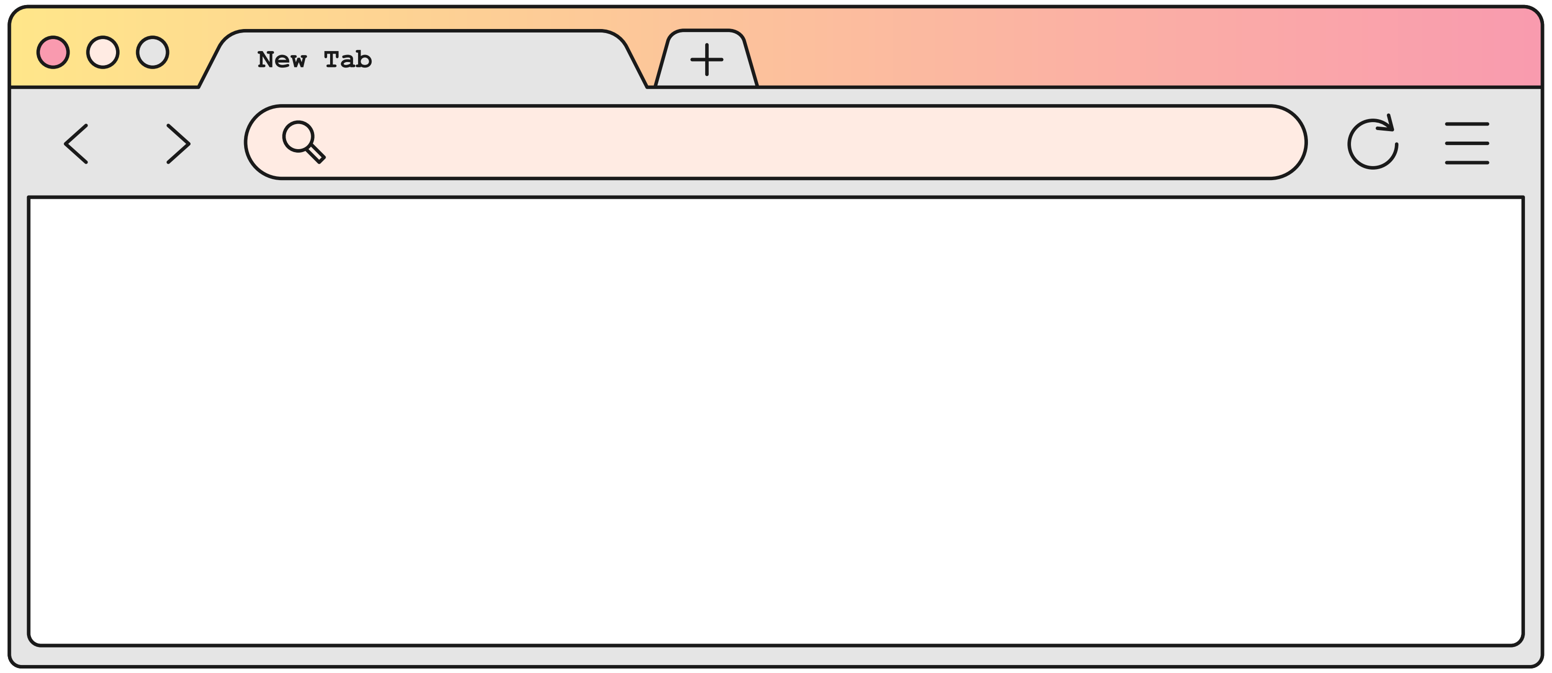
```
HTTP/1.1  200  OK
date: Tue, 18 Jun 2024 10:03:55 GMT
cache-control: public, max-age=3600
content-type: text/html
```

We want to be part of this system?

- we just have to speak their language. We will be mostly requesting information so we need a program that can send http requests for us in an easy without having to format messages in that hard way.

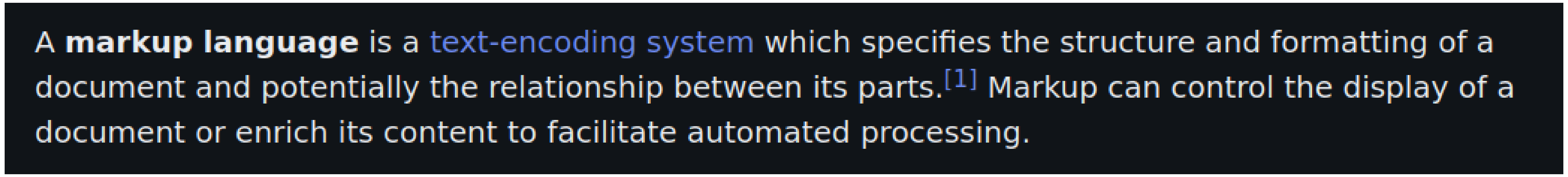
- Browsers are actually perfect for this job. Browsers are perfect http client but they are not the only ones.

- Let’s demonstrate an example



HTML

HTML is a markup language.



A **markup language** is a text-encoding system which specifies the structure and formatting of a document and potentially the relationship between its parts.[1] Markup can control the display of a document or enrich its content to facilitate automated processing.

How browsers make sense of of HTML?

- Similar as how compilers make sense of programming languages

Making Sense Of an expression

```
2 + (3 * 4)
```

Rules of HTML

If you looked at an web page you will see a lot of elements
each of them correspond to an HTML element

HOW we define an HTML element in HTML?

- Elements are pieces of content wrapped in an opening and closing HTML tags
- HTML opening tag tells the browser that is the start of the HTML element
- HTML closing tag tell the browser where the an element ends
- Attributes added to the opening tag provide additional information to the HTML element

```
<tagname attribute=value> element_content </tagname>
```

not all attributes require values (boolean ones)

Examples:

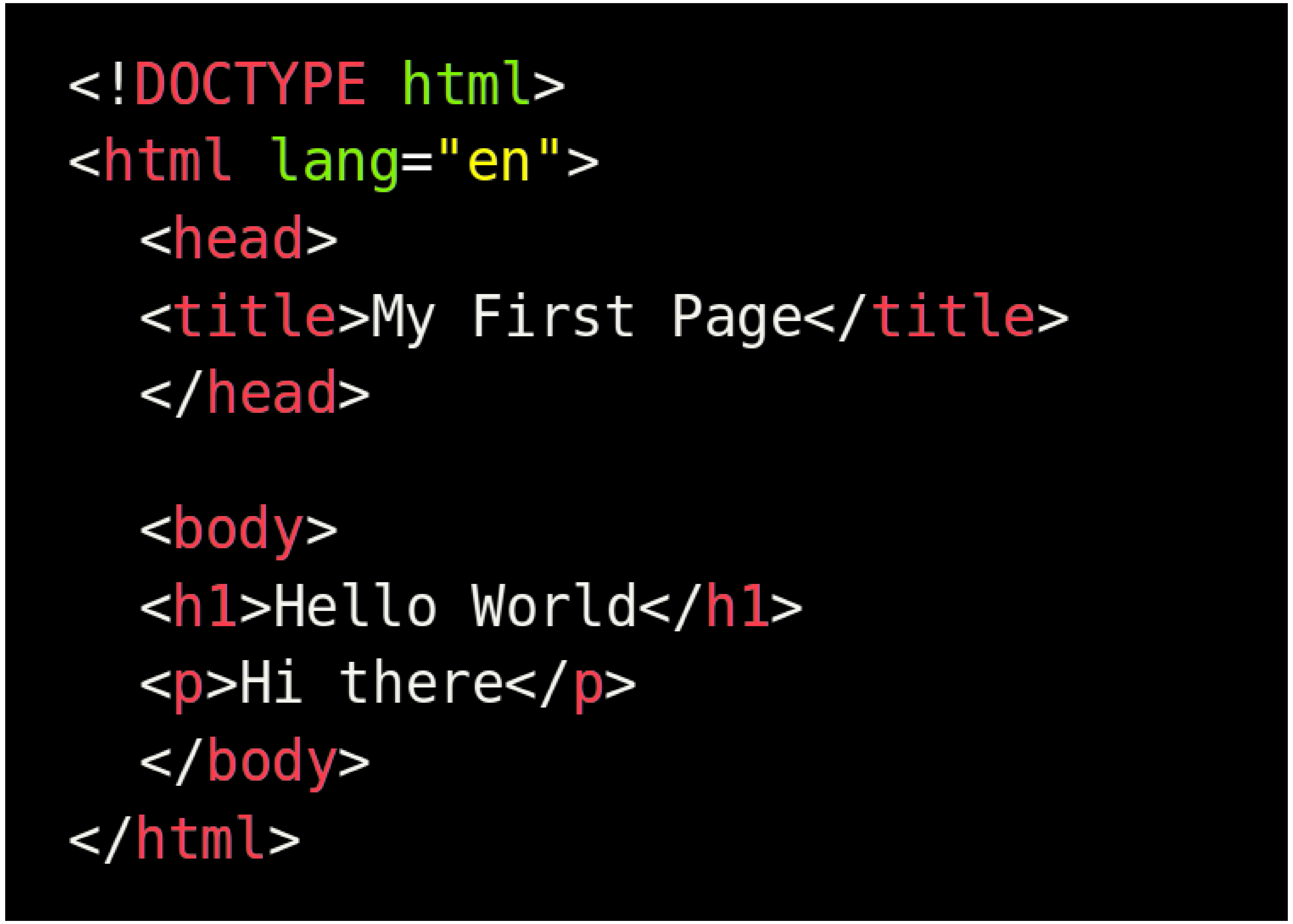
```
<h1 align="center">Hello World</h1>
```

- Some elements are “void” (do not take any content) and take the form of self-closing tag

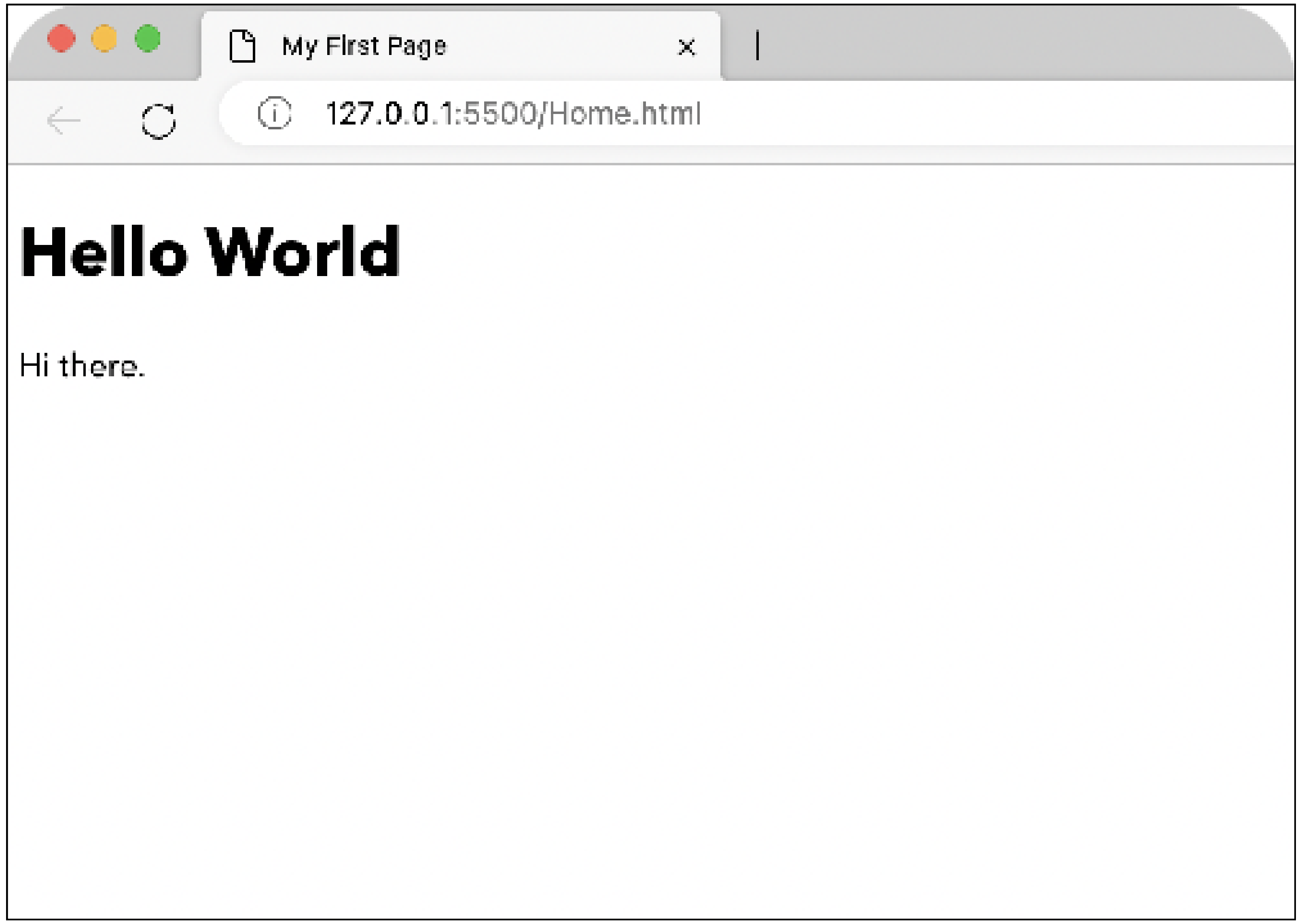
```
<img src=”./cat.jpg" alt="Cute Cat” />
```

- Note: HTML tags are not case-sensitive

HTML DOCUMENT BOILER PLATE



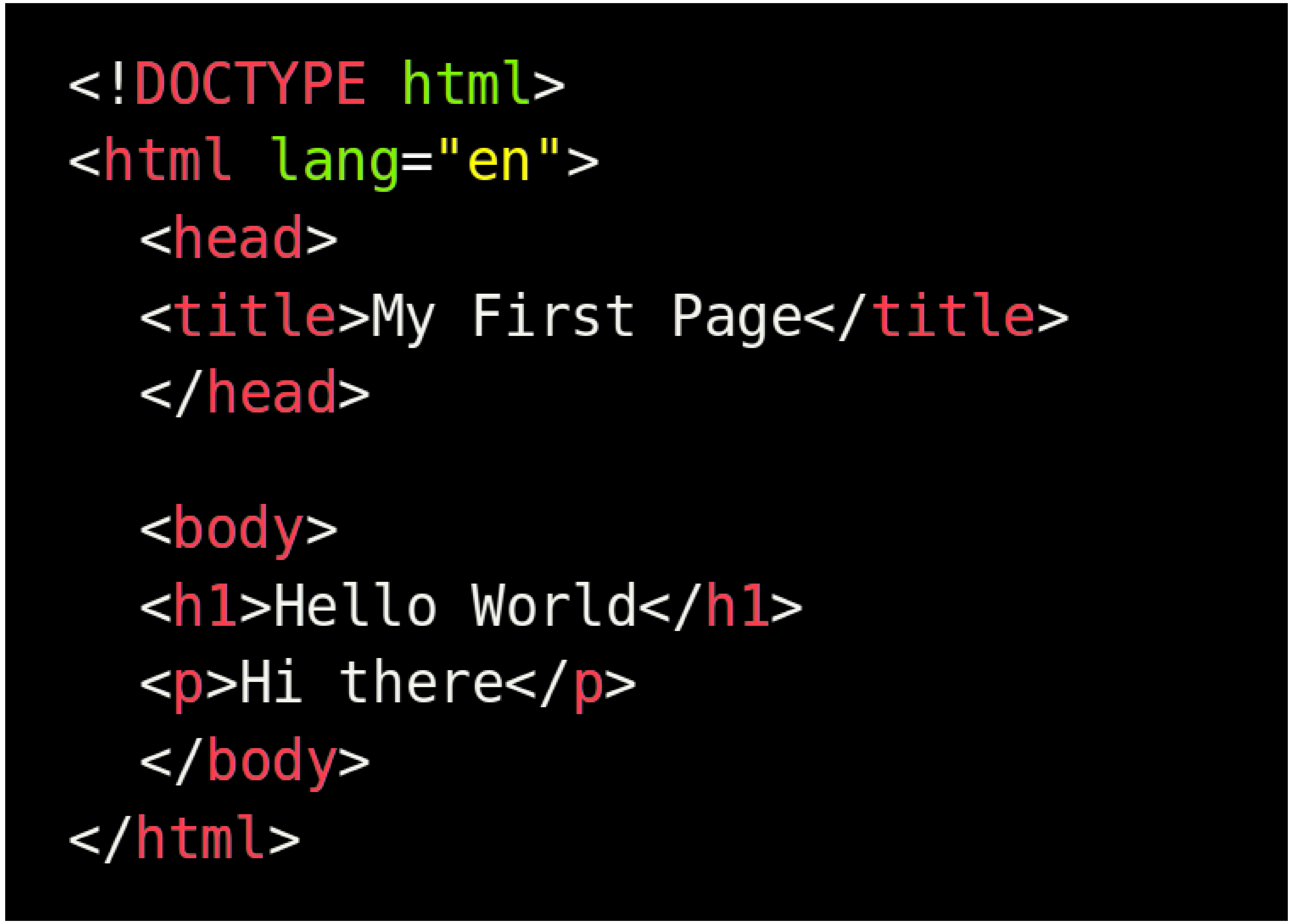
```
<!DOCTYPE html>
<html lang="en">
  <head>
  <title>My First Page</title>
  </head>

  <body>
  <h1>Hello World</h1>
  <p>Hi there</p>
  </body>
</html>
```



Hello World

Hi there.

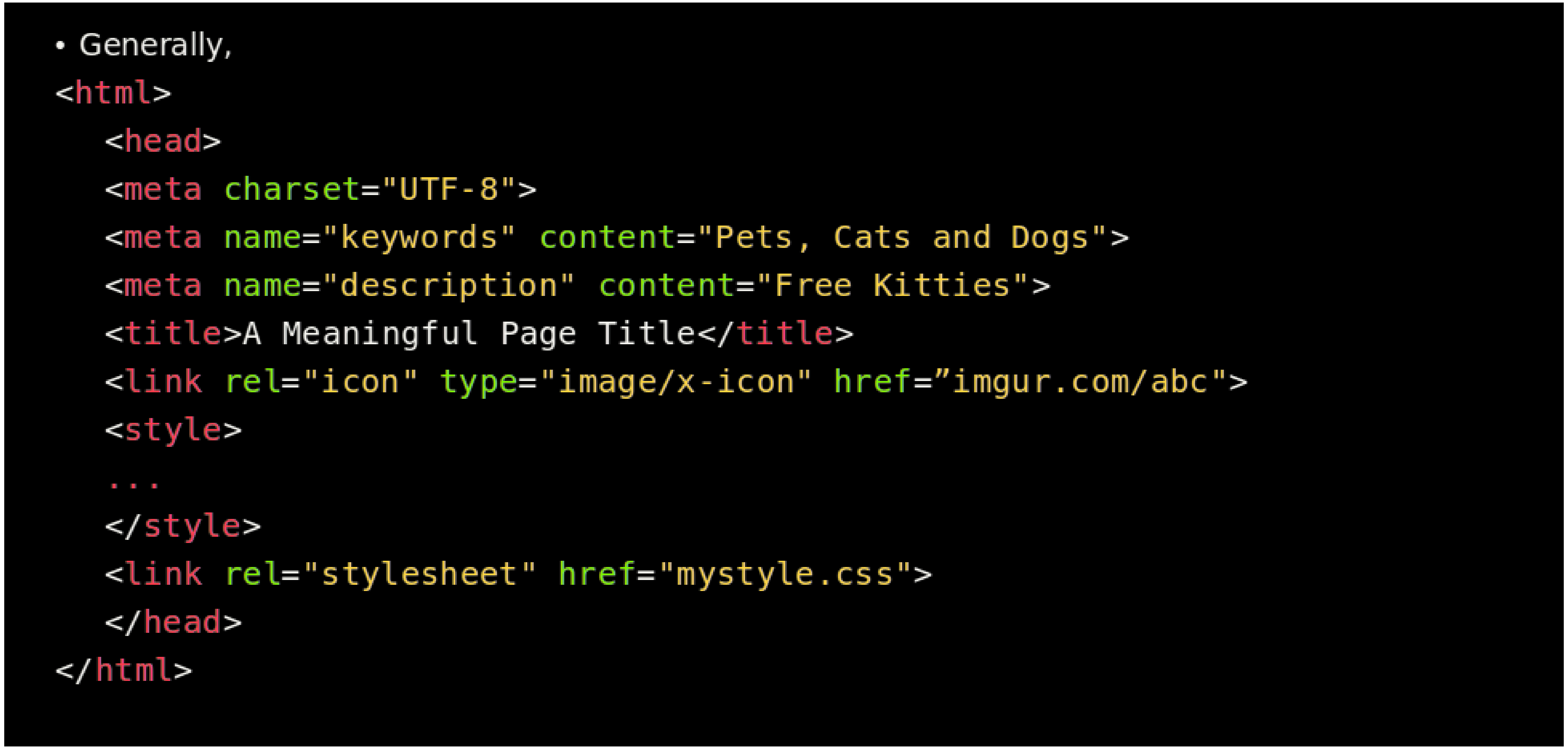
MAIN COMPONENTS



```
<!DOCTYPE html>
<html lang="en">
  <head>
  <title>My First Page</title>
  </head>

  <body>
  <h1>Hello World</h1>
  <p>Hi there</p>
  </body>
</html>
```

- doctype --> specify the html version
- html --> root element
- head --> contain metadata elements
- body --> contain the actual content

HEAD Tag

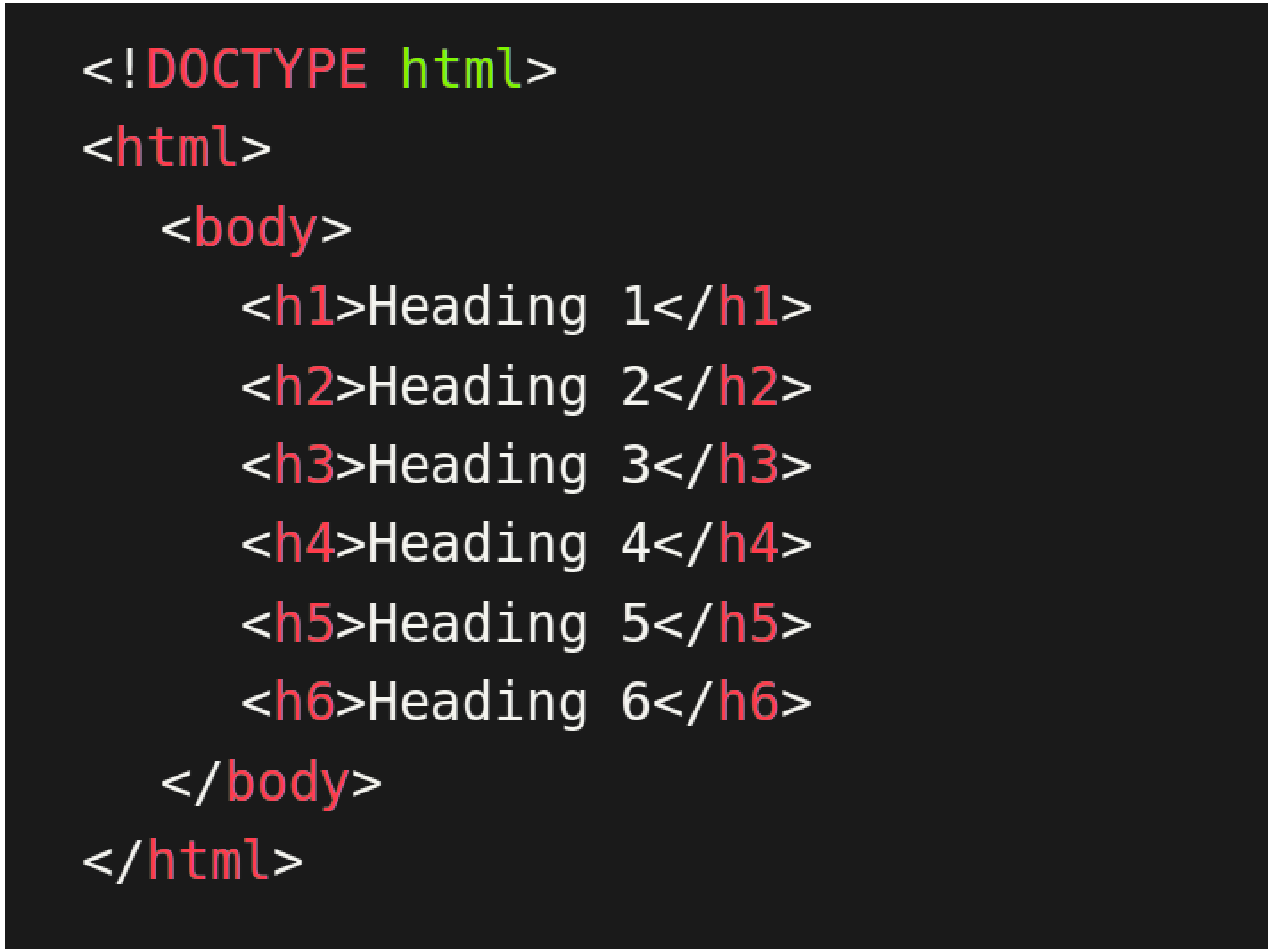


- Generally,

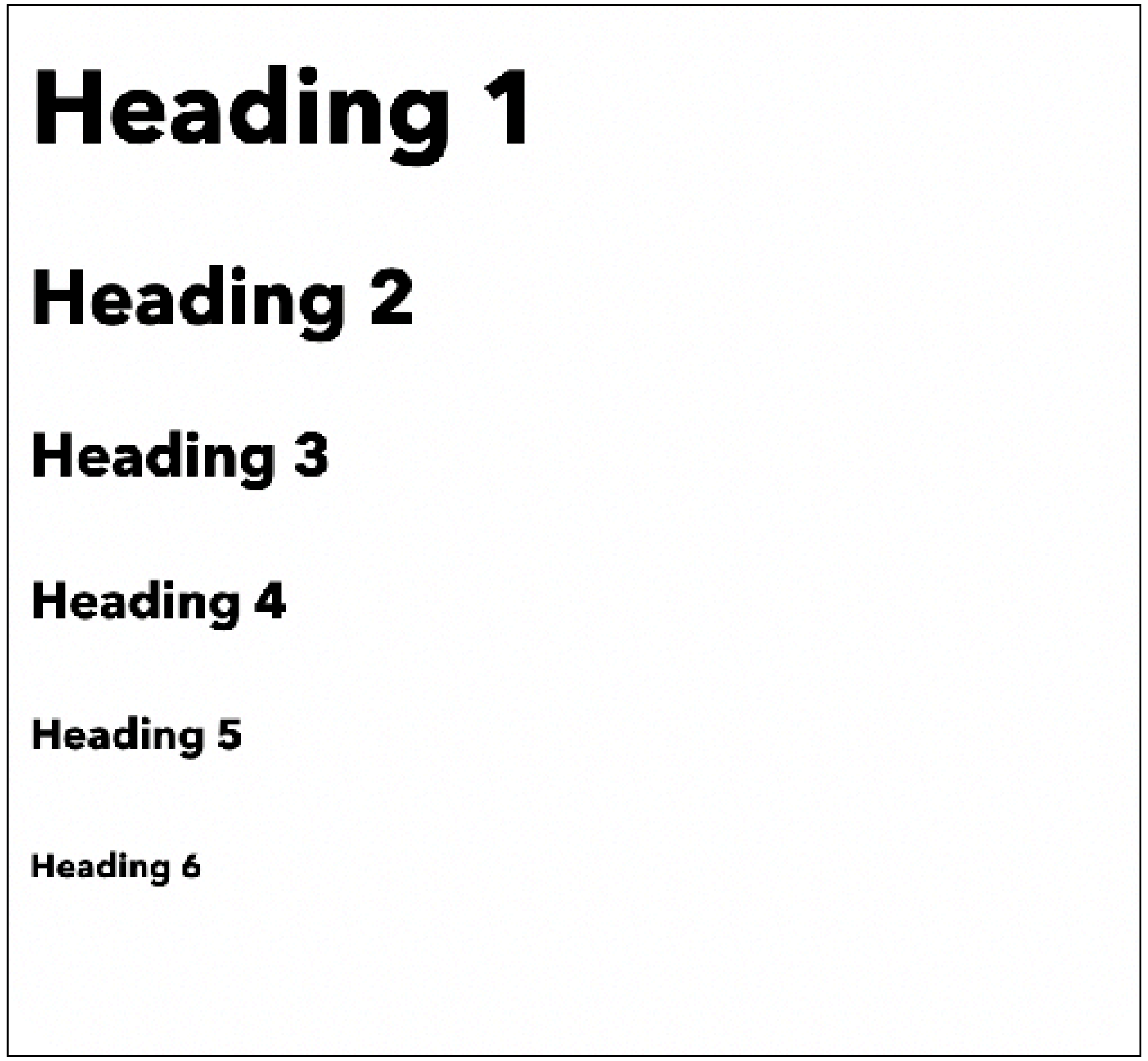
```
<html>
  <head>
  <meta charset="UTF-8">
  <meta name="keywords" content="Pets, Cats and Dogs">
  <meta name="description" content="Free Kitties">
  <title>A Meaningful Page Title</title>
  <link rel="icon" type="image/x-icon" href=”imgur.com/abc">
  <style>
  ...
  </style>
  <link rel="stylesheet" href="mystyle.css">
  </head>
</html>
```

HTML Elements - HEADINGS

- When you write a page, there is a main heading, subheadings, and subheadings of subheadings. HTML provides elements for this hierarchy.



```html
<!DOCTYPE html>
<html>
  <body>
    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
    <h3>Heading 3</h3>
    <h4>Heading 4</h4>
    <h5>Heading 5</h5>
    <h6>Heading 6</h6>
  </body>
</html>
```



Heading 1

Heading 2

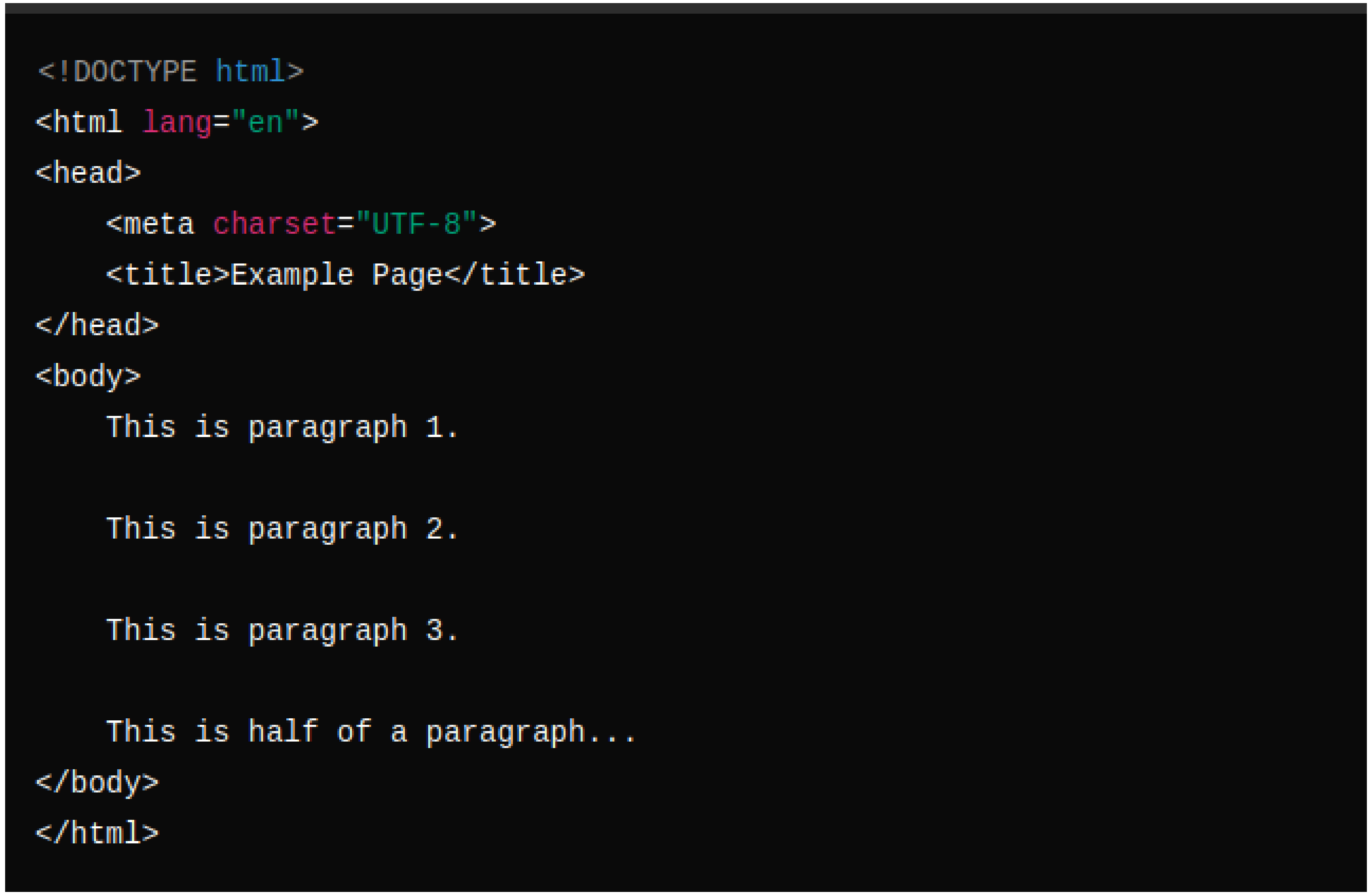
Heading 3

Heading 4

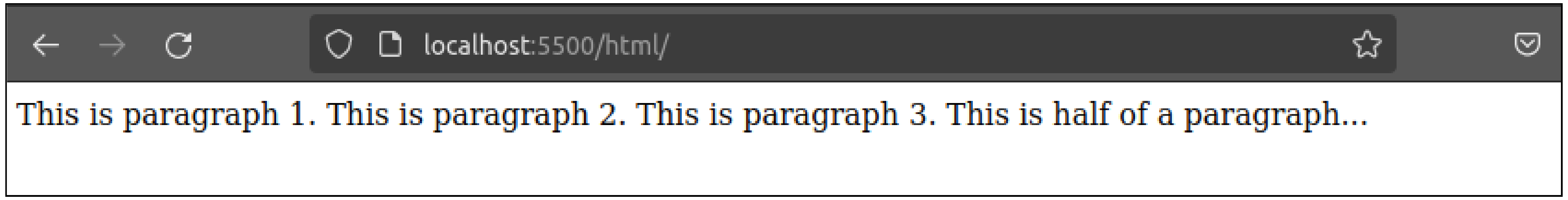
Heading 5

Heading 6

HTML Elements - Paragraphs

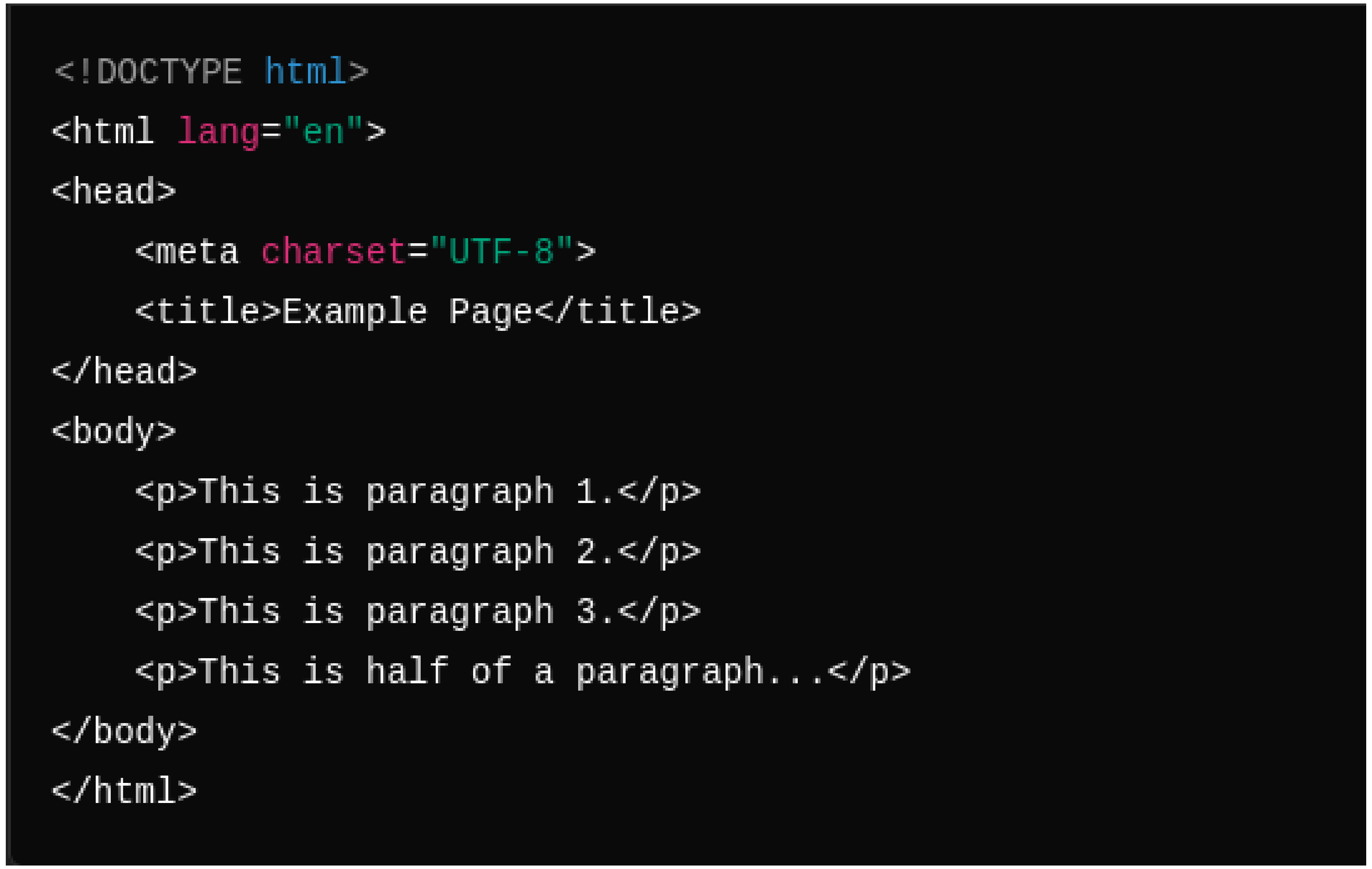


```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Example Page</title>
</head>
<body>
    This is paragraph 1.

    This is paragraph 2.

    This is paragraph 3.

    This is half of a paragraph...
</body>
</html>
```

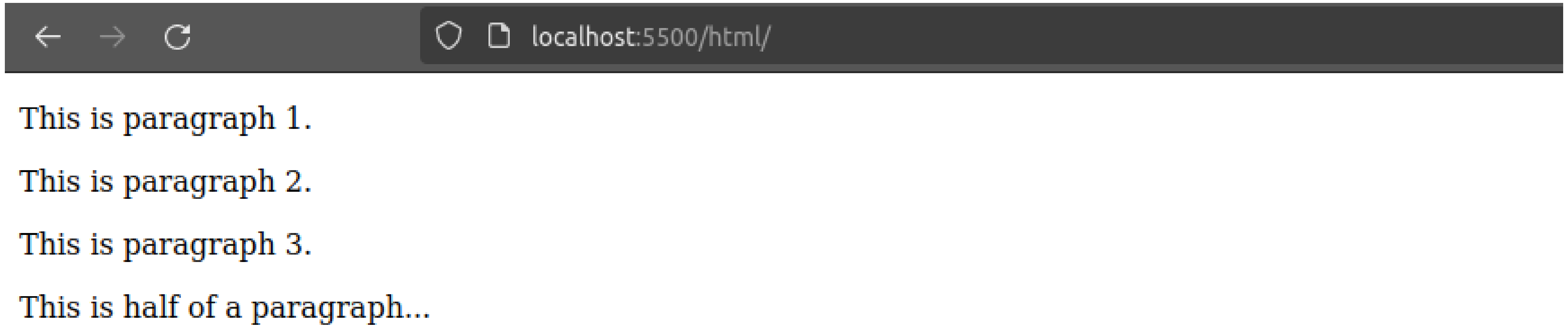


This is paragraph 1. This is paragraph 2. This is paragraph 3. This is half of a paragraph...

HTML Elements - Paragraphs



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Example Page</title>
</head>
<body>
    <p>This is paragraph 1.</p>
    <p>This is paragraph 2.</p>
    <p>This is paragraph 3.</p>
    <p>This is half of a paragraph...</p>
</body>
</html>
```



This is paragraph 1.

This is paragraph 2.

This is paragraph 3.

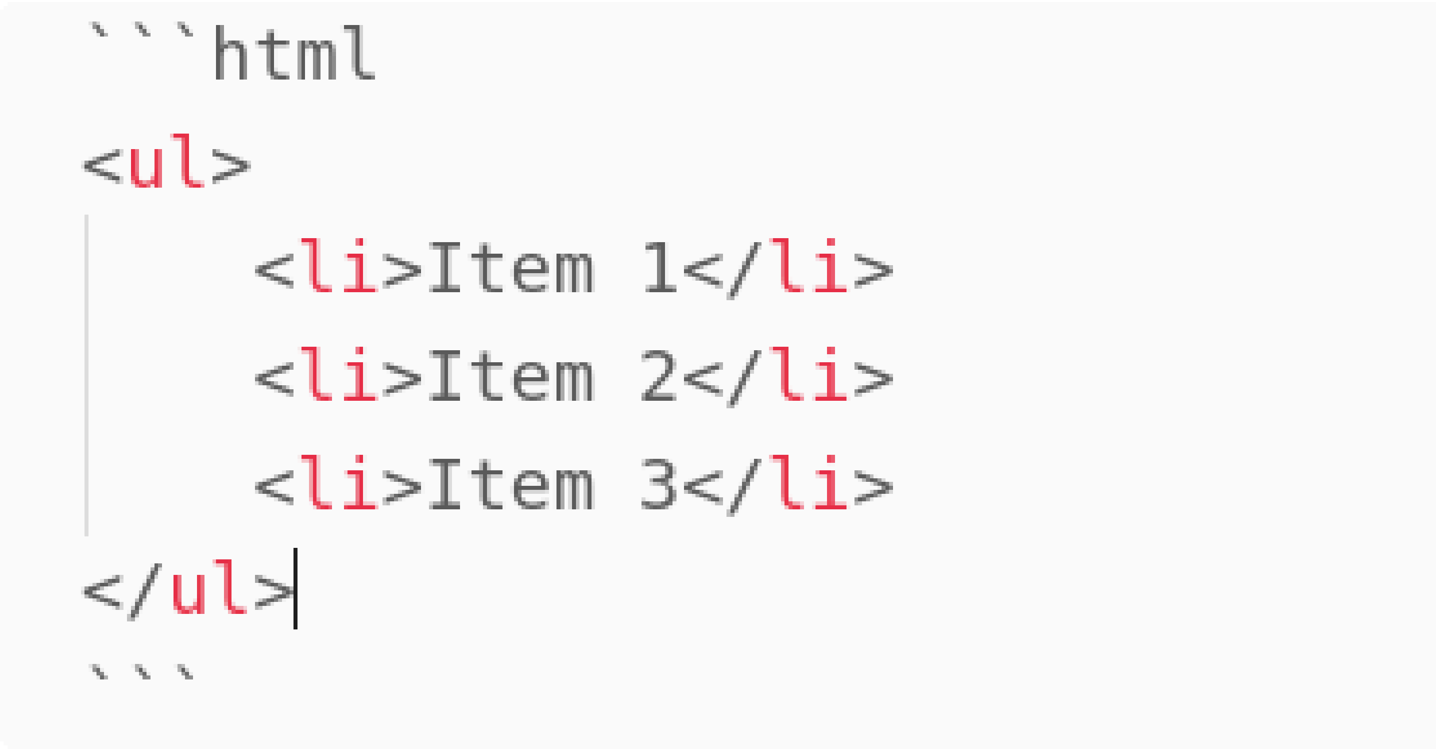
This is half of a paragraph...

HTML Elements - Lists

You might want to represent lists on your website, such as home needs. HTML provides two types of lists:

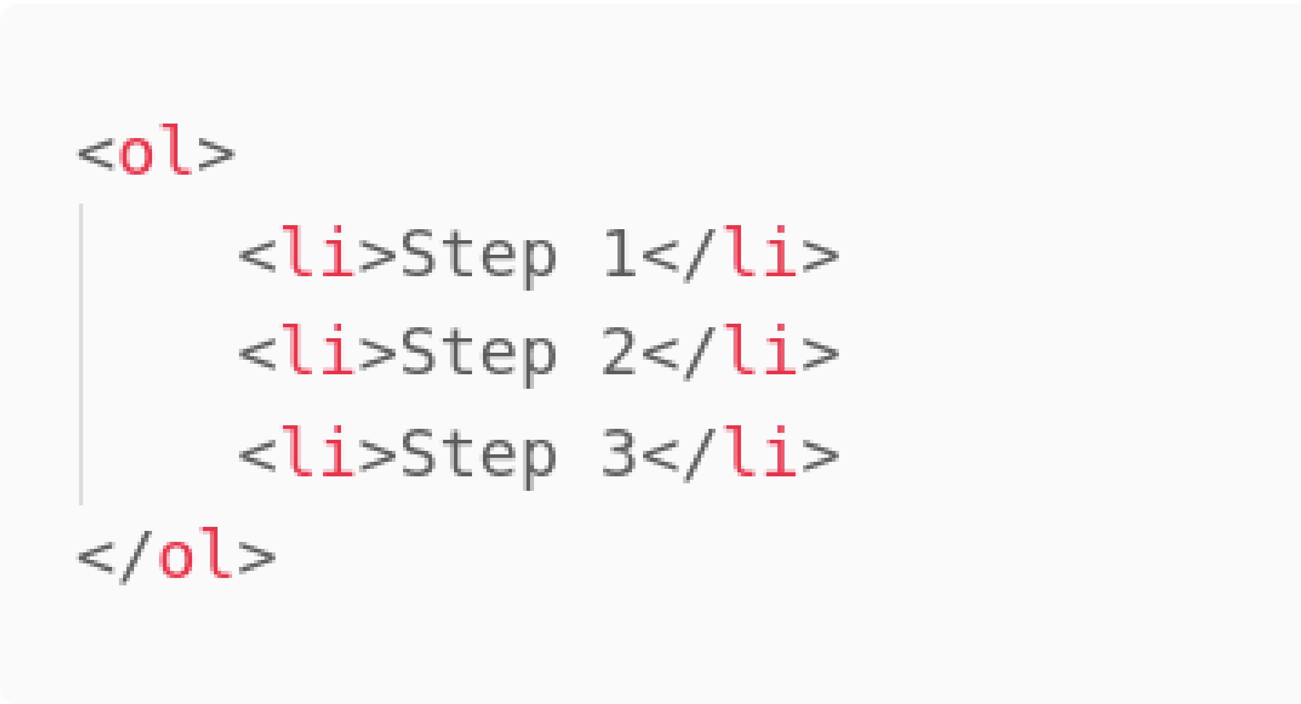
1. **Unordered list** - similar to to-do lists
2. **Ordered list** - steps for a recipe

Unordered List:



~~~
```html
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
    <li>Item 3</li>
</ul>
```
~~~

Ordered List:



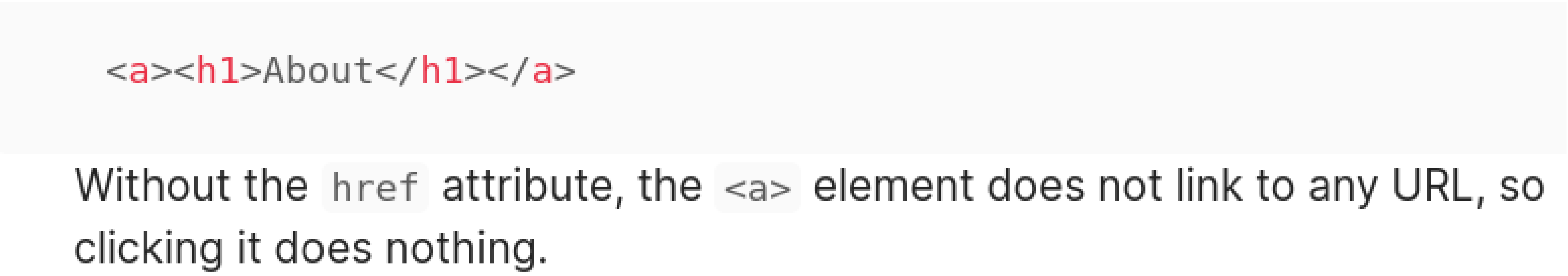
```
<ol>
    <li>Step 1</li>
    <li>Step 2</li>
    <li>Step 3</li>
</ol>
```

HTML Elements - Links

- We have been saying the web the web why is it called the web any way

- The reason it is called the web is the links the ability to link different web pages and resources to each other

- To create a link in HTML you use the anchor element `<a>` you put text inside it or another HTML element

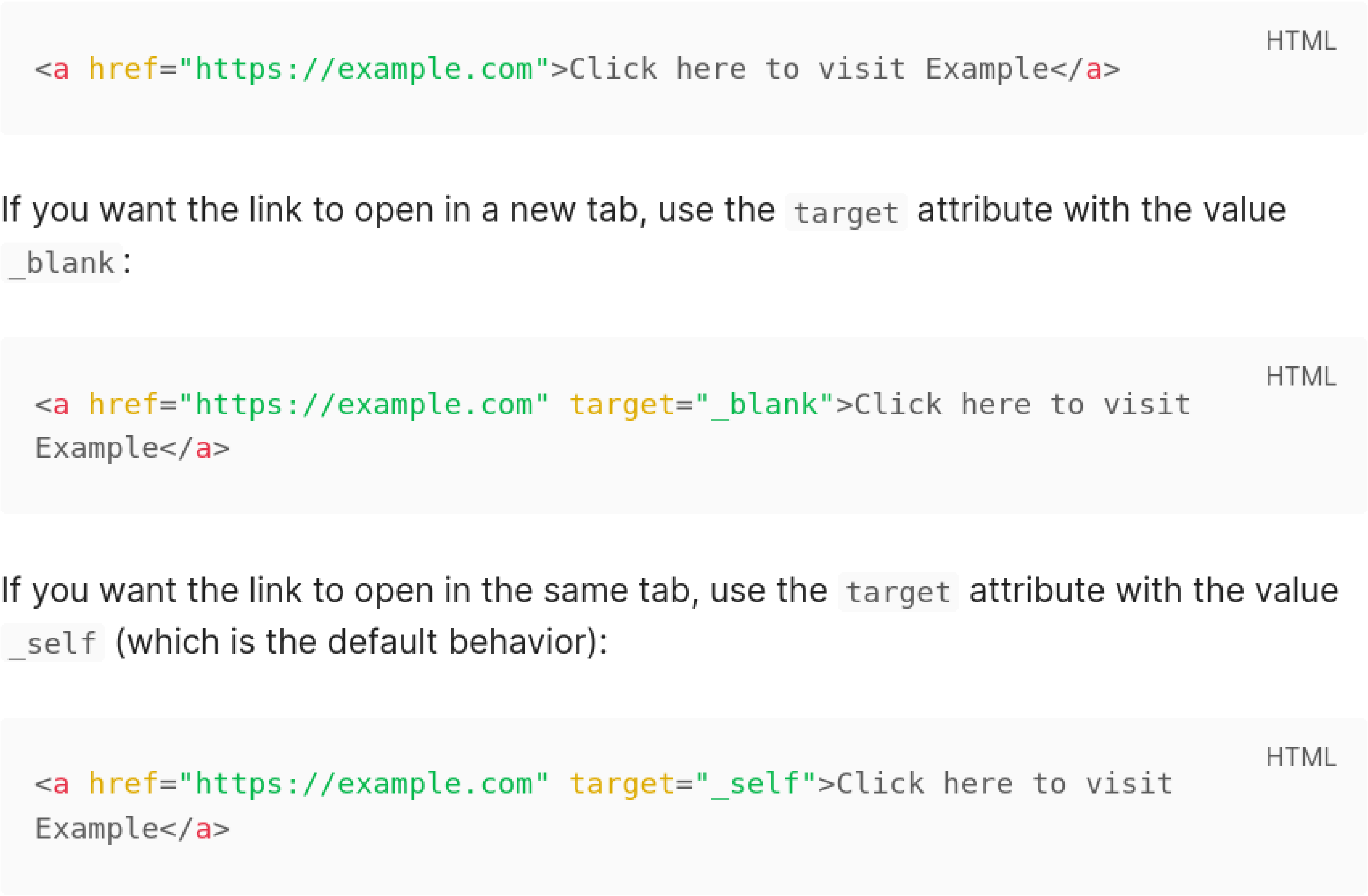


```
<a><h1>About</h1></a>
```

Without the `href` attribute, the `<a>` element does not link to any URL, so clicking it does nothing.

HTML Elements - Links

- Examples



```html
<a href="https://example.com">Click here to visit Example</a>
```

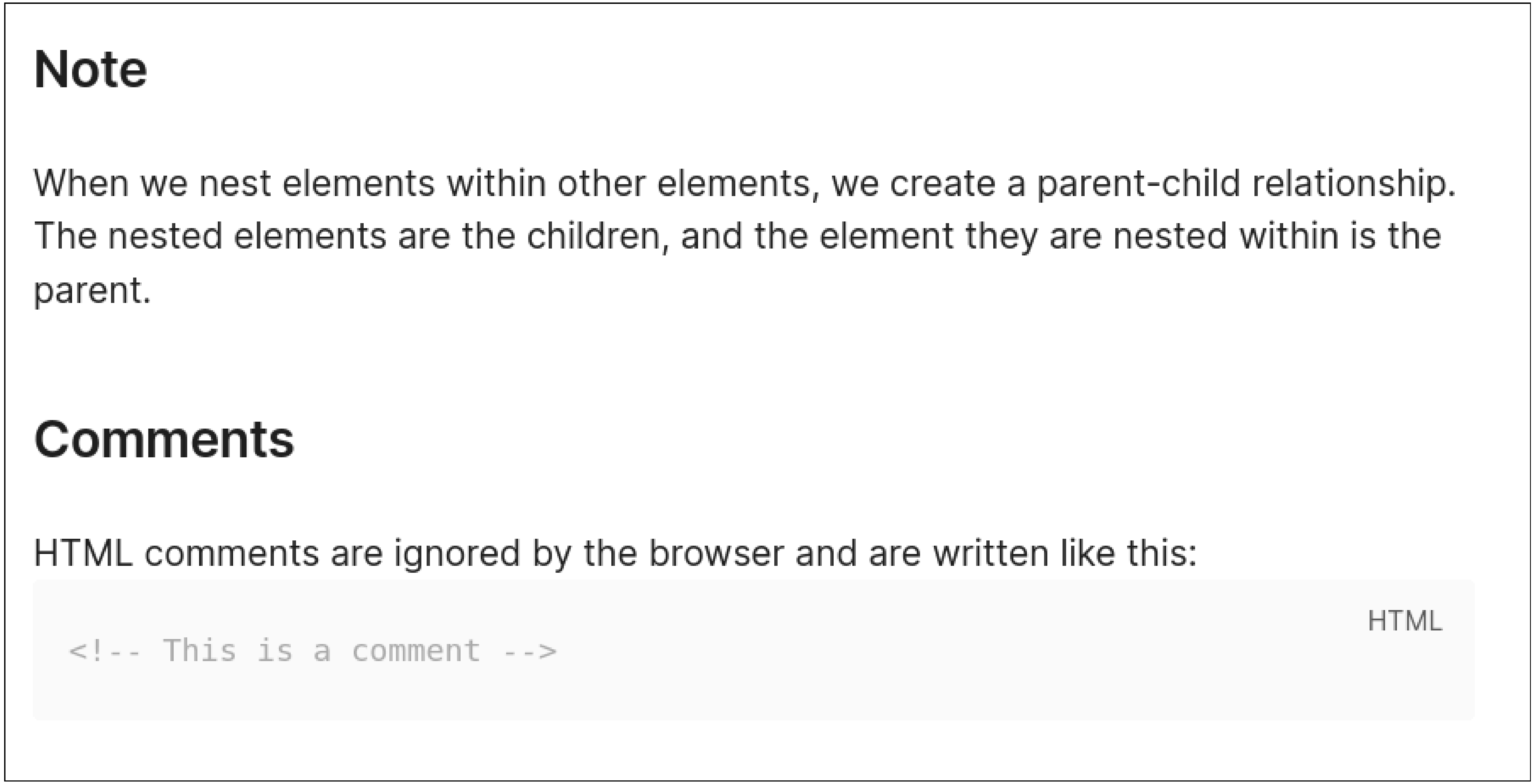
If you want the link to open in a new tab, use the `target` attribute with the value `_blank`:

```html
<a href="https://example.com" target="_blank">Click here to visit
Example</a>
```

If you want the link to open in the same tab, use the `target` attribute with the value `_self` (which is the default behavior):

```html
<a href="https://example.com" target="_self">Click here to visit
Example</a>
```

HTML Elements - Notes



Note

When we nest elements within other elements, we create a parent-child relationship. The nested elements are the children, and the element they are nested within is the parent.

Comments

HTML comments are ignored by the browser and are written like this:

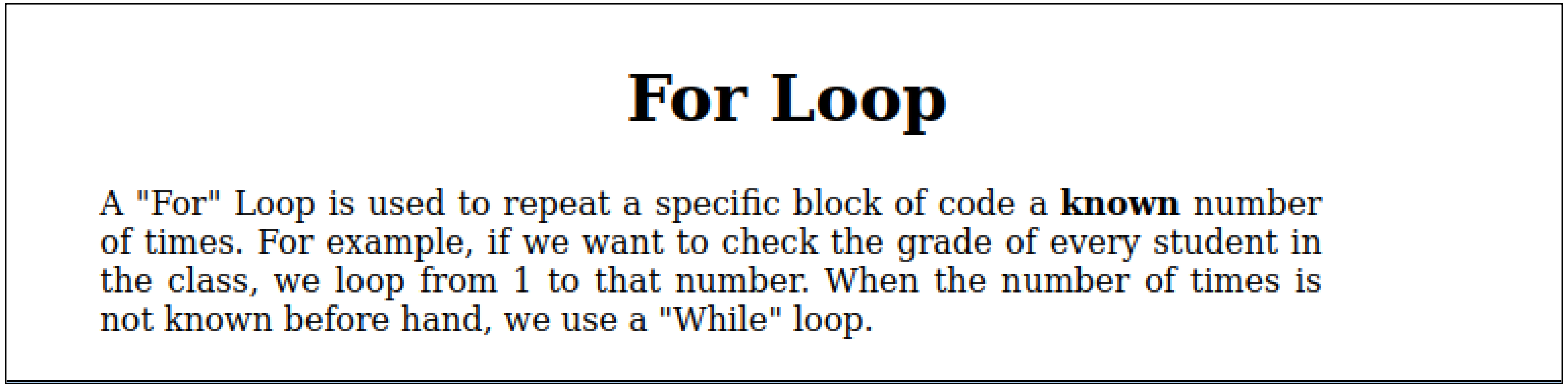
```html
<!-- This is a comment -->
```

Syntax Vs Semantics

- Syntax: Refer to the set of rules that define the grammer of the language (the structre and format).

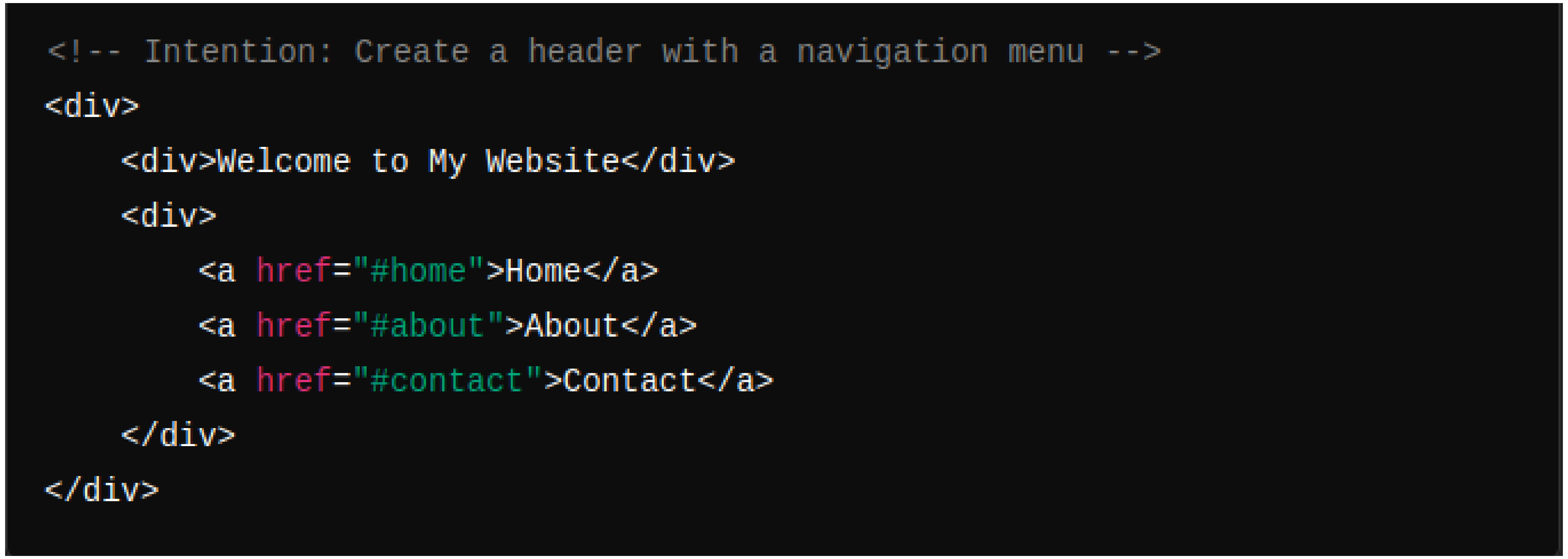
- Semantics: Refer to the meaning and purpose of the sentence

- You are expert in violating syntactic rules let’s talk semantics.



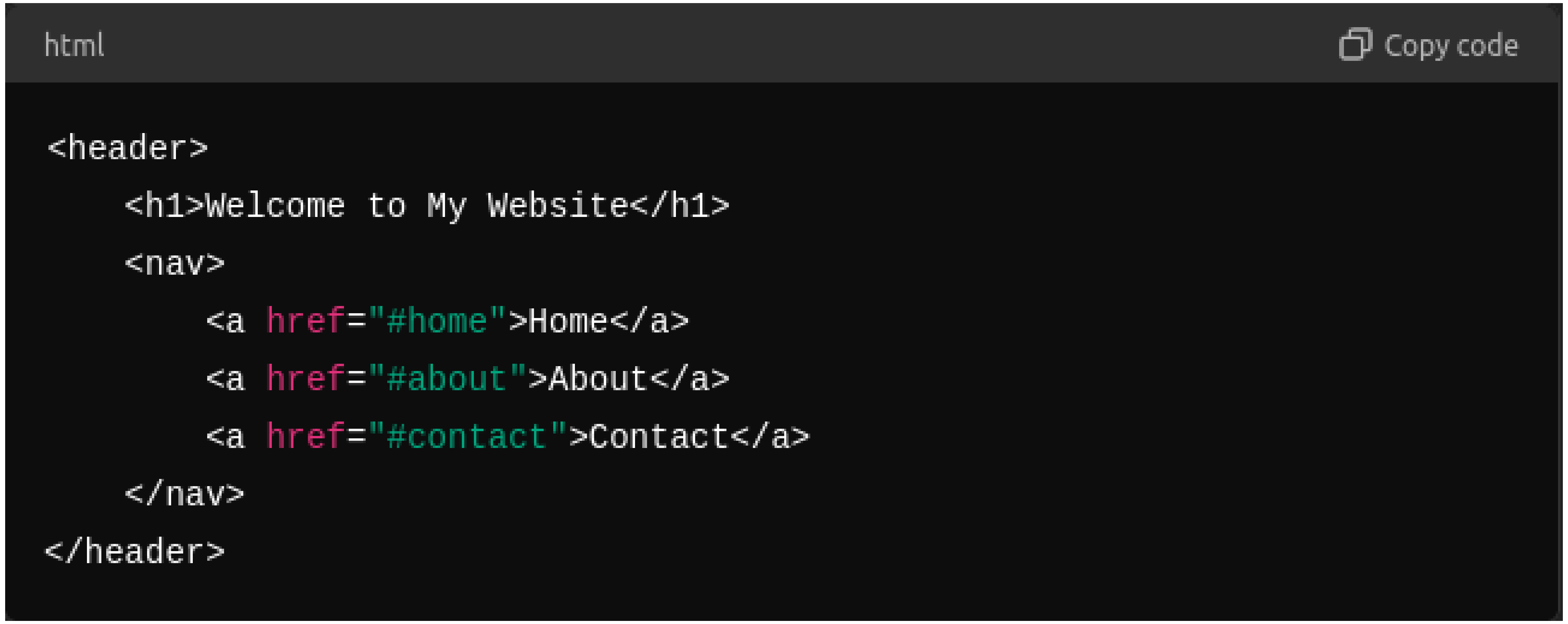
For Loop

A "For" Loop is used to repeat a specific block of code a **known** number of times. For example, if we want to check the grade of every student in the class, we loop from 1 to that number. When the number of times is not known before hand, we use a "While" loop.



```html
<!-- Intention: Create a header with a navigation menu -->
<div>
    <div>Welcome to My Website</div>
    <div>
        <a href="#home">Home</a>
        <a href="#about">About</a>
        <a href="#contact">Contact</a>
    </div>
</div>
```

Syntactically correct, semantically incorrect



```html
<header>
    <h1>Welcome to My Website</h1>
    <nav>
        <a href="#home">Home</a>
        <a href="#about">About</a>
        <a href="#contact">Contact</a>
    </nav>
</header>
```

Syntactically correct, semantically correct