

URI & HTML & GET/POST

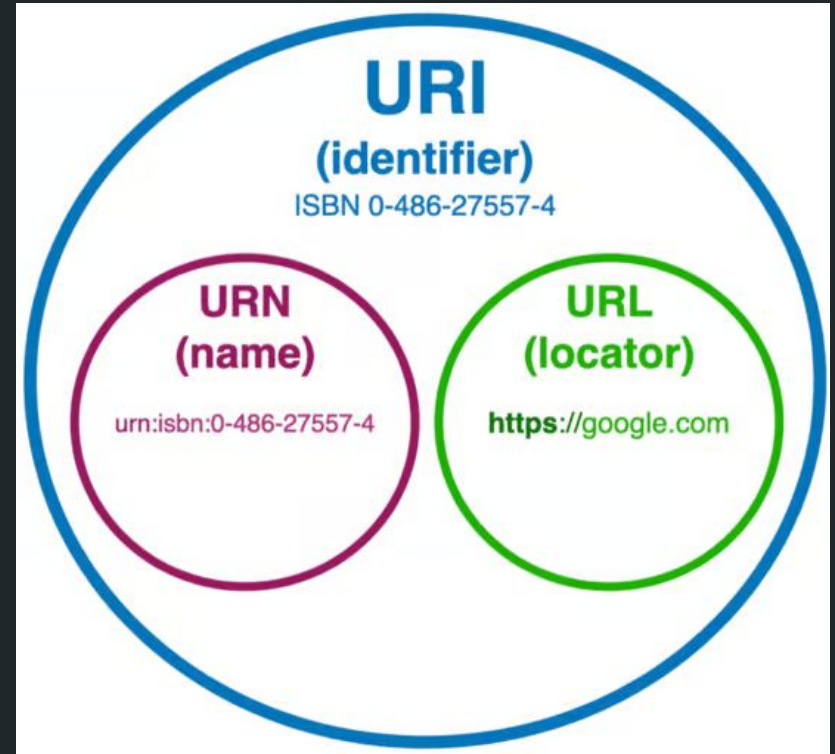
February 8th 2024

Introduction to URIs

- URI stands for Uniform Resource Identifier
- “A Uniform Resource Identifier (URI) is a compact sequence of characters that identifies an abstract or physical resource”

ISBN 0-486-27557-4

Types of Uniform Resource Identifiers



URLs

- Uniform Resource Locator
- Identifies and locates websites
- Specifies how a resource can be accessed
- location-dependent

Example:

<https://www.google.com/>

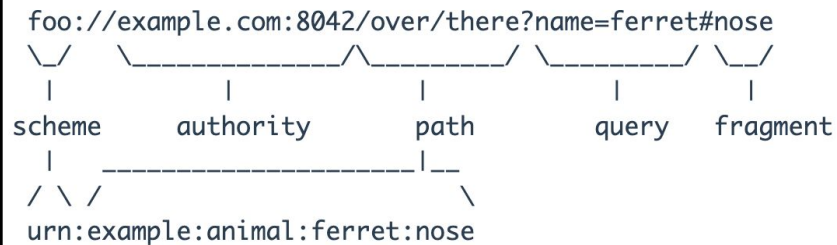
URNs

- Uniform Resource Name
- location independent and persistent
- A URN has three components:
 - The label "urn"
 - A colon
 - A character string as the unique identifier

Example:

Urn:oasis:names:specification:docbook:dtd:xml:4.1.2

URI Components



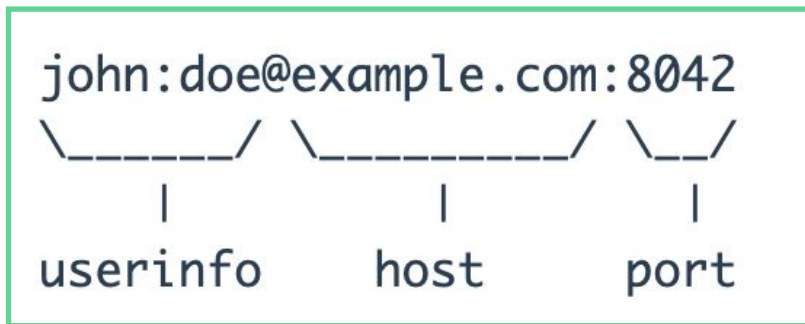
Scheme

- Scheme is the first element in the URI
- The scheme establishes the protocols for the URI
- Case Insensitive



Authority

- includes information about the server responsible for hosting the resource
- The authority component is optional



Path

- The path specifies the resource location on the server
- The path component is NOT optional



Query

- The query component is optional
- The query is most often a sequence of attribute/value pairs



Fragment

- The fragment is the last component and is optional
- The fragment provides a direction to a secondary resource
- If the URI refers to the whole object, the fragment ID is void



Hyper Text Markup Language (HTML)



Introduction to HTML

- HTML stands for Hyper Text Markup Language
- the standard markup language for creating Web pages
- describes the structure of a Web page
- consists of elements tell the browser how to display the content

HTML Elements

- An HTML element is defined by a start tag, content and then an end tag

Example:

```
<h1>My First Heading</h1>
```

HTML elements can be nested

Example: `<body>`

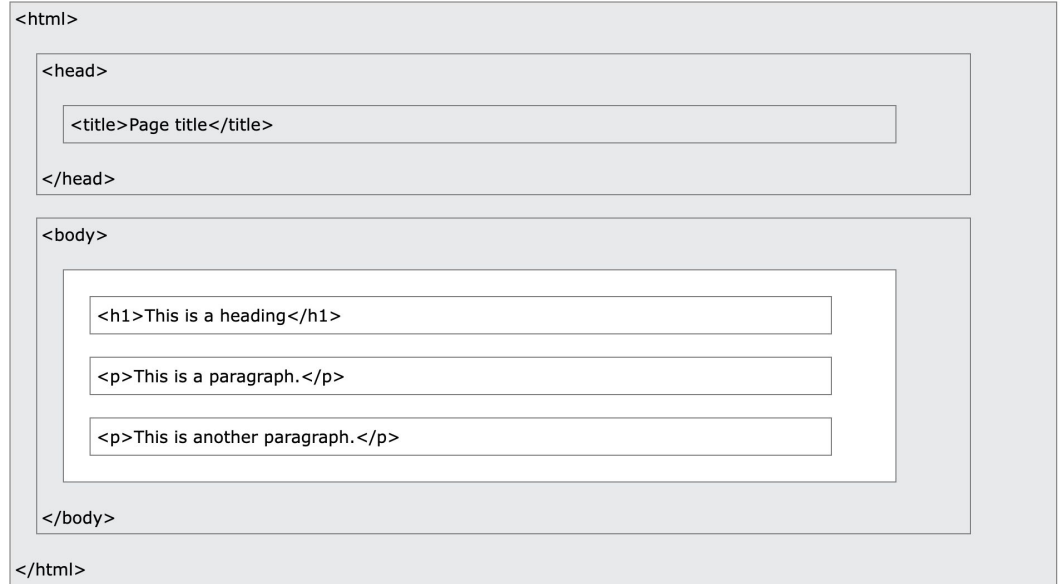
```
<h1>My First Heading</h1>
```

```
<p>My first paragraph.</p>
```

```
</body>
```

HTML Structure

- All HTML documents must start with a document type declaration: `<!DOCTYPE html>`.
- The HTML document always begins with `<html>` and ends with `</html>`
- The `<head>` and `<body>` contains all the visible contents

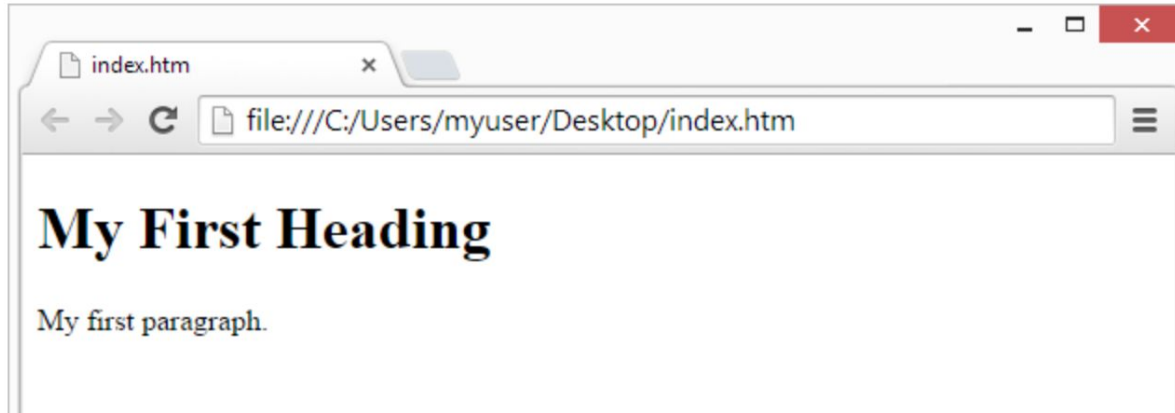


How is HTML Displayed?

A browser does not display the HTML tags, it uses tag them to determine the display of the document.

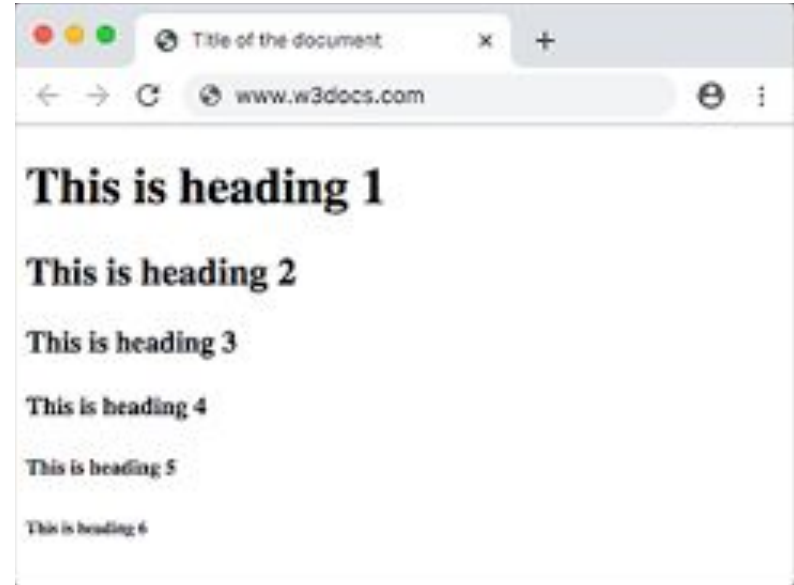
Example: `<h1>My First Heading</h1>`

`<p>My first paragraph.</p>`



HTML Headings

- HTML headings consists of **titles** or **subtitles**
- They are defined with the <h1> to <h6> tags.
 - <h1> defines the most important heading
 - <h6> defines the least important heading.
- Search engines use the headings to index the structure and content of a web page



HTML Paragraphs

- The HTML `<p>` element defines a paragraph
- With HTML, you cannot change the display by adding extra spaces or extra lines in your HTML code

```
<p>  
This paragraph  
contains a lot of lines  
in the source code,  
but the browser  
ignores it.  
</p>
```

```
<p>  
This paragraph  
contains      a lot of spaces  
in the source  code,  
but the      browser  
ignores it.  
</p>
```

This paragraph contains a lot of lines in the source code, but the browser ignores it.

This paragraph contains a lot of spaces in the source code, but the browser ignores it.

HTML Paragraphs - Formatting

- `<pre>` element defines preformatted text and it preserves both spaces and line breaks

```
<!DOCTYPE html>
<html>
<body>

<p>The pre tag preserves both spaces and line breaks:</p>

<pre>
  My Bonnie lies over the ocean.
  My Bonnie lies over the sea.
  My Bonnie lies over the ocean.
  Oh, bring back my Bonnie to me.
</pre>

</body>
</html>
```

The pre tag preserves both spaces and line breaks:

```
My Bonnie lies over the ocean.
My Bonnie lies over the sea.
My Bonnie lies over the ocean.
Oh, bring back my Bonnie to me.
```

HTML Attributes

- All HTML elements can have attributes
 - They are always specified in the start tag
- Attributes provide additional information about elements
- Attributes usually come in name/value pairs like: name="value"

The href Attribute

- The <a> tag defines a hyperlink
- The href attribute specifies the URL of the page the link goes to

Example:

```
<a href="https://www.w3schools.com">Visit W3Schools</a>
```

[Visit W3Schools](https://www.w3schools.com)

The src Attribute

- The tag is used to embed an image
- The src attribute specifies the path to the image to be displayed

Example:

```

```

The style Attribute

- The style attribute is used to add styles to an element

Example:

```
<!DOCTYPE html>
<html>
<body>

<h1 style="color:blue;">This heading is blue</h1>
<p style="color:red;">This paragraph is red.</p>

</body>
</html>
```

This heading is blue

This paragraph is red.

- Use **color** for text colors
- Use **font-family** for text fonts
- Use **font-size** for text sizes
- Use **text-align** for text alignment

HTML Formatting

- `` - Bold text
- `` - Important text
- `<i>` - Italic text
- `` - Emphasized text
- `<mark>` - Marked text
- `<small>` - Smaller text
- `` - Deleted text
- `<ins>` - Inserted text
- `<sub>` - Subscript text
- `<sup>` - Superscript text

Example:

```
<p><b>This text is bold</b></p>
```

This text is bold

While you can use the style attribute and formatting to design the page, using CSS is better for styling web pages.

HTTP Get and Post

Introduction to HTTP

- HTTP stands for Hypertext Transfer Protocol
- it's the communication protocol used with websites
 - operates as a request-response protocol between a browser (client) and server

The GET Method

- The GET method is used to retrieve data from a specific resource
- With GET the data is not modified in anyway
- GET requests have length restrictions
- GET parameters are all sent in the URL

Example:

```
/test/demo_form.php?name1=value1&name2=value2
```

The POST Method

- The purpose of the POST method is to create or update a resource
- POST parameters are sent in the body of the request
- POST requests do not have data length restrictions

Example:

```
POST /test/demo_form.php HTTP/1.1  
Host: w3schools.com  
  
name1=value1&name2=value2
```

Compare GET vs. POST

| | GET | POST |
|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| History | Parameters remain in browser history | Parameters are not saved in browser history |
| Restrictions on data length | Yes, when sending data, the GET method adds the data to the URL; and the length of a URL is limited (maximum URL length is 2048 characters) | No restrictions |
| Restrictions on data type | Only ASCII characters allowed | No restrictions. Binary data is also allowed |
| Security | GET is less secure compared to POST because data sent is part of the URL Never use GET when sending passwords or other sensitive information! | POST is a little safer than GET because the parameters are not stored in browser history or in web server logs |
| Visibility | Data is visible to everyone in the URL | Data is not displayed in the URL |

HTML Forms

- Forms are a type of html element that are used to collect information from the user
- The HTML `<form>` element defines a form
- It has a `method` attribute that specifies how to send form-data

```
<form>
```

```
▪
```

```
form elements
```

```
▪
```

```
</form>
```

Get Example

```
<!DOCTYPE html>
<html>
<body>

<h1>The form method="get" attribute</h1>

<form action="/action_page.php" method="get" target="_blank">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
</form>

<p>Click on the submit button!</p>

</body>
</html>
```

The form method="get" attribute

First name:

Last name:

Click on the submit button!

Post Example

```
<!DOCTYPE html>
<html>
<body>

<h1>The form method="post" attribute</h1>

<form action="/action_page.php" method="post" target="_blank">
  <label for="fname">First name:</label>
  <input type="text" id="fname" name="fname"><br><br>
  <label for="lname">Last name:</label>
  <input type="text" id="lname" name="lname"><br><br>
  <input type="submit" value="Submit">
</form>

<p>Click on the submit button! </p>

</body>
</html>
```

The form method="post" attribute

First name:

Last name:

Click on the submit button!

Sources

- Aviani, G. (2019, September 12). *An introduction to [http](#): Everything you need to know*. freeCodeCamp.org. <https://www.freecodecamp.org/news/http-and-everything-you-need-to-know-about-it/>
- Awati, R., & Churchville, F. (2021, October 15). *What is a uniform resource identifier (URI)?*. WhatIs. <https://www.techtarget.com/whatis/definition/URI-Uniform-Resource-Identifier>
- Berners-Lee, T., Fielding, R. T., & Masinter, L. M. (n.d.). *RFC 3986: Uniform resource identifier (URI): Generic syntax*. IETF Datatracker. <https://datatracker.ietf.org/doc/html/rfc3986#section-1.1>
- HTML introduction*. Introduction to HTML. (n.d.). https://www.w3schools.com/html/html_intro.asp
- Thinkful. (2023, May 30). *Learning the basics of HTML and HTML5*. <https://www.thinkful.com/blog/learn-the-basics-of-html-and-html5/>
- Uri parts and components*. URI. (n.d.). <https://uri.thephpleague.com/uri/4.0/components/overview/>
- Wikimedia Foundation. (2024, January 18). *Uniform resource identifier*. Wikipedia. https://en.wikipedia.org/wiki/Uniform_Resource_Identifier

Practice!

Create a Web Page With...

- A large red header with your name
- A paragraph with a sentence describing your favorite hobby
 - In the paragraph add a link to more information about your hobby
- Add an image at the bottom of your page