



Viewing Websites

What are Websites?

Websites are built using a variety of different tools, though the most common are HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and Javascript.

Browsers are built so that they can interpret these and display them nicely!



What are Websites?

Websites are built using a variety of different tools, though the most common are HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and Javascript.

Browsers like Firefox and Chrome are built so that they can interpret these and display them nicely!

How do browsers **find** web pages, though?



Step 1: The URL

Uniform Resource Locators (or URLs) are the paths used by the Internet to **Locate Resources** (wow!)

We access different URLs when we type them into the browser, when we click on links, and when we refresh the page we're currently on!



Step 1: The URL

www.example.com/homepage.html

A URL describes to your browser exactly what resource you're looking for, and where that resource is located.

Every URL is comprised of 2 different parts:

1. The Domain
2. The Path



Step 1: The URL

www.example.com/homepage.html

A URL describes to your browser exactly what resource you're looking for, and where that resource is located.

Every URL is comprised of 2 different parts:

1. The Domain
 - a. This tells you **where** on the Internet you need to search for the resource
2. The Path



Step 1: The URL

www.example.com/homepage.html

A URL describes to your browser exactly what resource you're looking for, and where that resource is located.

Every URL is comprised of 2 different parts:

1. The Domain
 - a. This tells you **where** on the Internet you need to search for the resource
2. The Path
 - a. This is the path to the actual resource that's being requested



Step 1: The URL

www.example.com/homepage.html

A URL describes to your browser exactly what resource you're looking for, and where that resource is located.

Every URL is comprised of 2 different parts:

1. The Domain
 - a. This tells you **where** on the Internet you need to search for the resource
2. The Path
 - a. This is the path to the actual resource that's being requested
 - i. This resource can be an HTML file, an image file, a music file, or any other thing being hosted on the Internet!



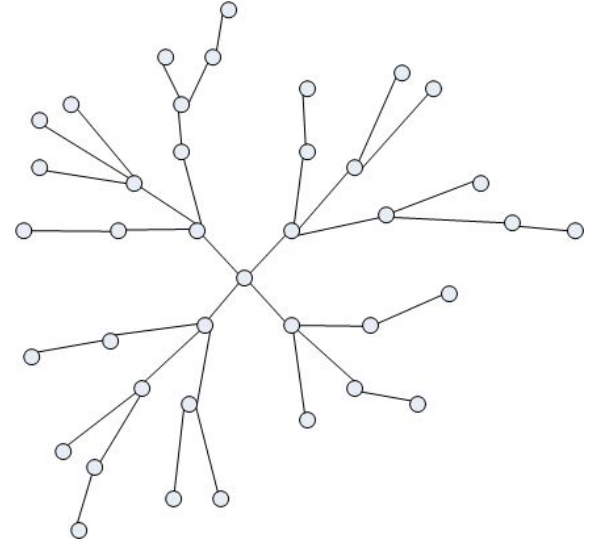
Step 1: The URL

www.example.com/homepage.html



Step 2: The Request

www.example.com/homepage.html



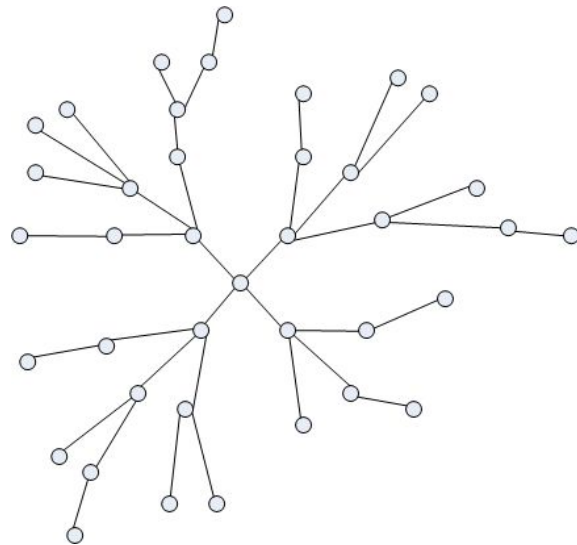
Step 2: The Request

www.example.com/homepage.html



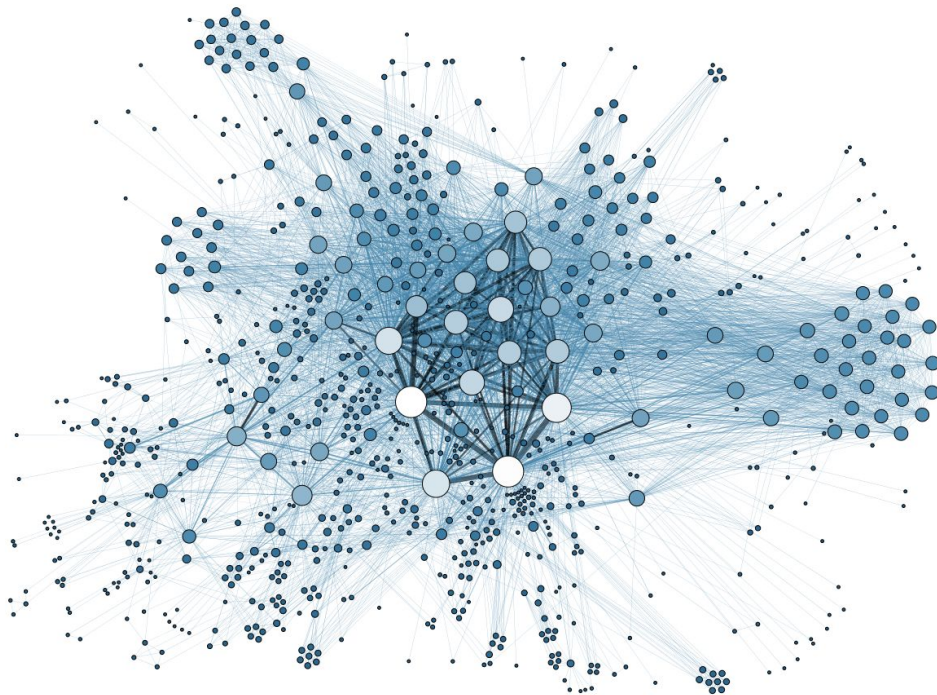
Request

"Hey, example.com! Please
send me your
homepage.html file!"

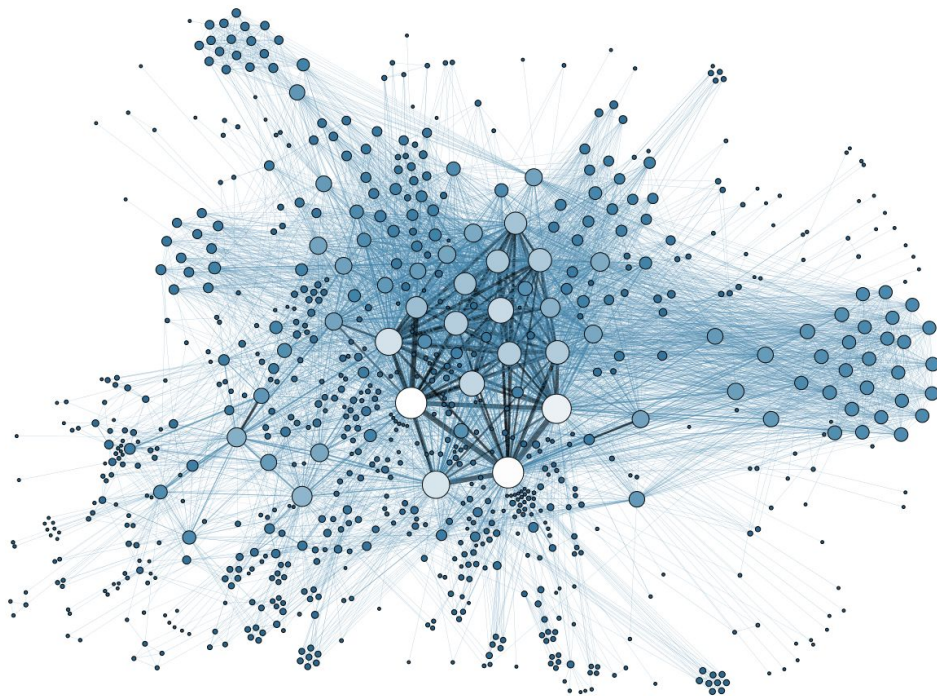
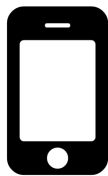


Step 2: The Request

The Internet is a massive connection of networks, all connected together!



Step 2: The Request

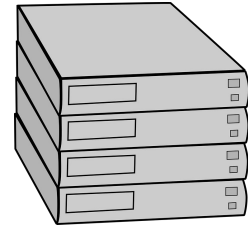
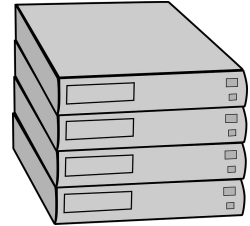
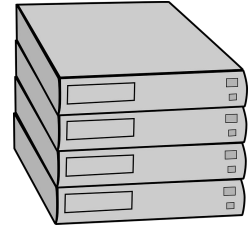


There are devices
of all sorts that are
connected to the
Internet -
computers,
telephones,
smartwatches, and
more!

Step 2: The Request

There is one other major type of device connected to the Internet, though:

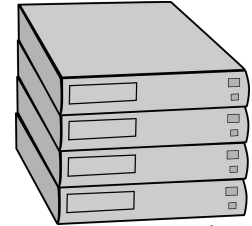
Servers!



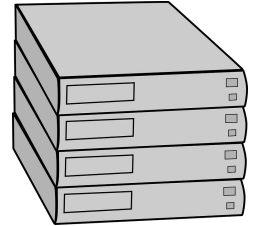
Step 2: The Request

Servers are used to host the information for a website, and **serve** it up to users upon request!

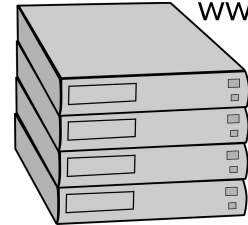
Each server belongs to a different website, and will sit and listen for requests.



www.google.com

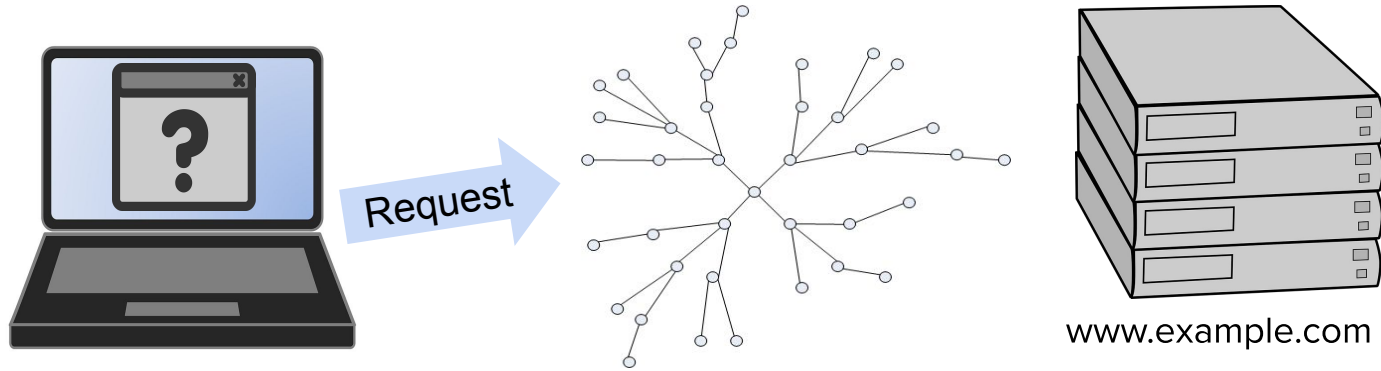


www.codehs.com



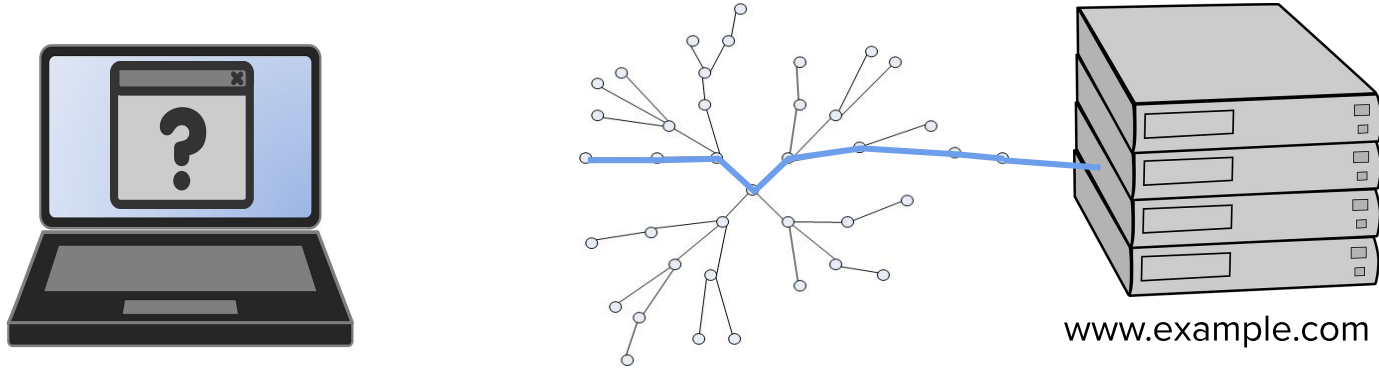
www.example.com

Step 2: The Request



Your computer will send a request to the server for the website in the **domain** of the URL you typed in, asking for the resource specified in the **path**!

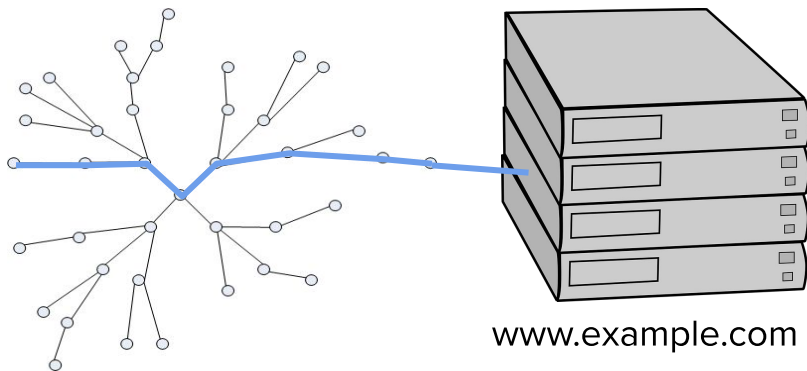
Step 2: The Request



Your computer will send a request to the server for the website in the **domain** of the URL you typed in, asking for the resource specified in the **path**!

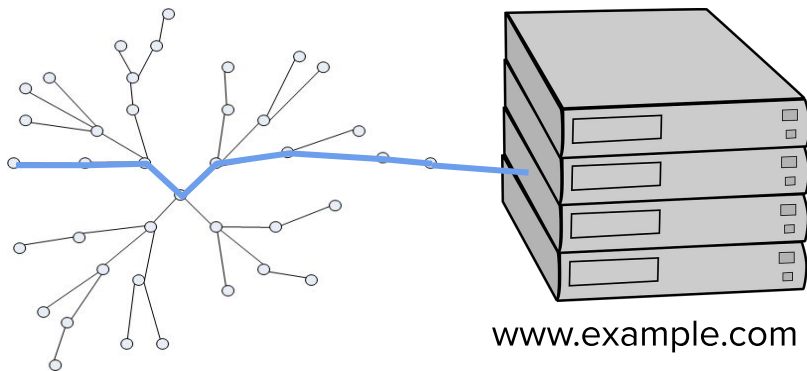
"Hey, example.com! Please send me your homepage.html file!"

Step 2: The Request



```
GET /homepage.html HTTP/1.1  
Host: www.example.com  
Content-Type: text/html  
Content-Language: en  
...
```

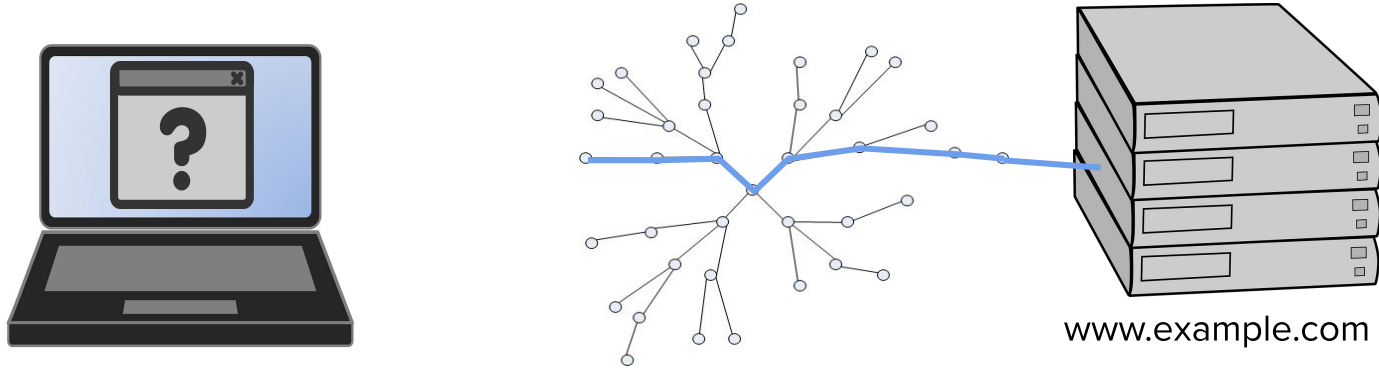
Step 2: The Request



```
GET /homepage.html HTTP/1.1  
Host: www.example.com  
Content-Type: text/html  
Content-Language: en  
...
```

The protocol that
defines how requests
should be structured is
called **HTTP**!

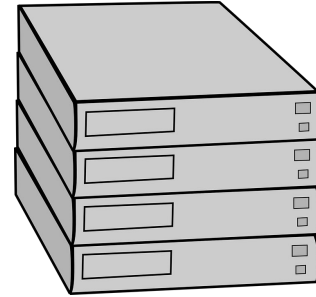
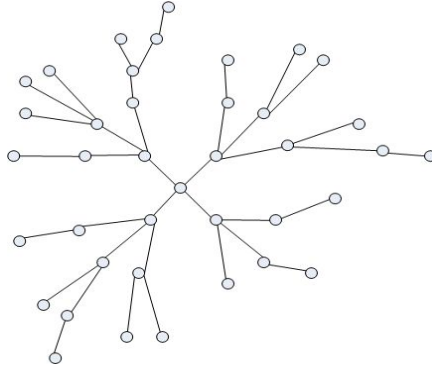
Step 2: The Request



"Hey, example.com! Please send me your homepage.html file!"

The systems in the Internet will forward this Request along until it reaches the server for example.com!

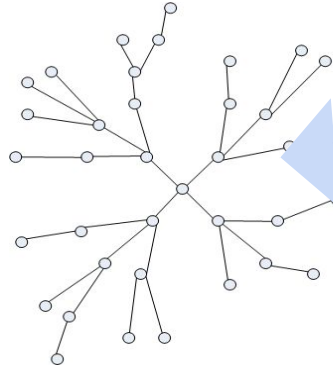
Step 3: The Response



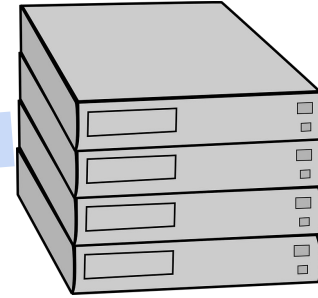
www.example.com

When the server receives the request, it will provide a **response**, usually containing the resource being requested!

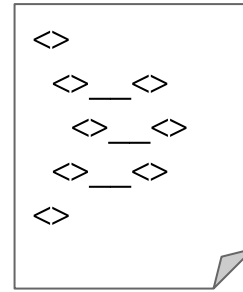
Step 3: The Response



Response



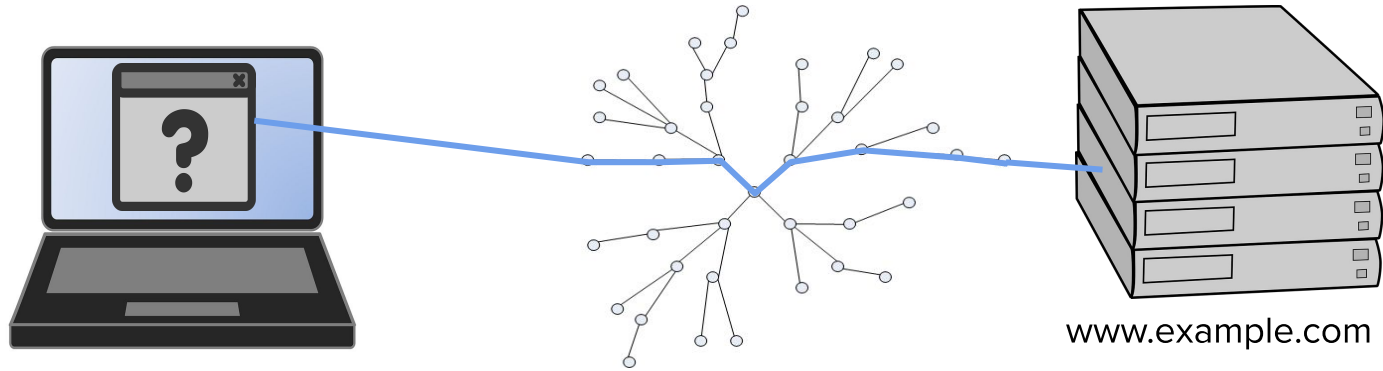
www.example.com



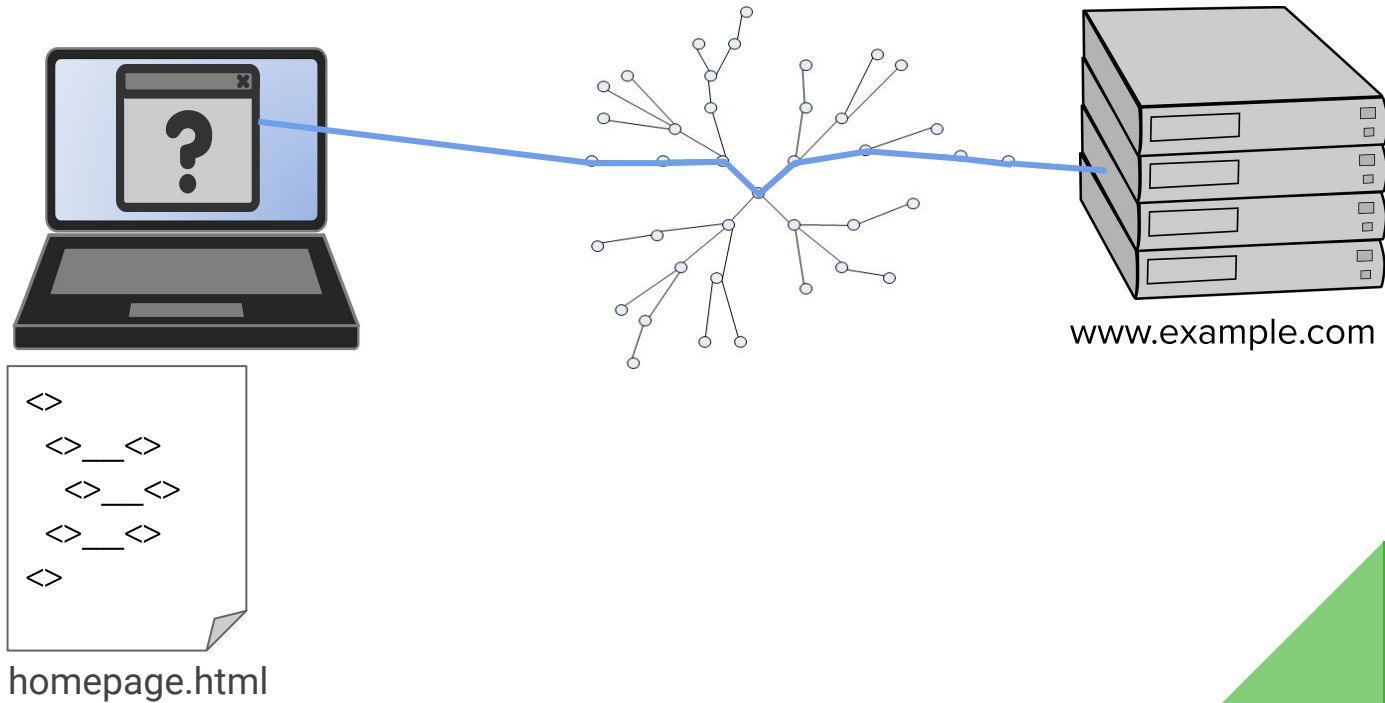
homepage.html

"Here you go!"

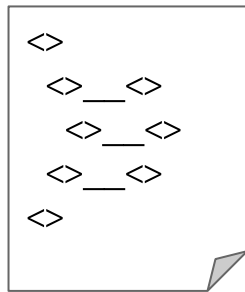
Step 3: The Response



Step 3: The Response



Step 4: Rendering the Page



homepage.html

Now that your browser has access to the homepage.html file, it can read through it and produce the beautiful webpage that it's encoding!

Step 4: Rendering the Page



Now that your browser has access to the homepage.html file, it can read through it and produce the beautiful webpage that it's encoding!

Once the page has been rendered, you can view it!



Step 4: Rendering the Page



This whole process - sending the request, getting a response, and rendering the webpage - happens in less than a second!

Hosting Our Own Sites

If we want to, we are able to host our own websites!

In order to accomplish this, there are a few steps:

1. Register a domain
2. Have a server listen to requests coming into that domain
3. Give the server files to respond with



Hosting Our Own Sites

If we want to, we are able to host our own websites!

In order to accomplish this, there are a few steps:

1. Register a domain
 - a. example.com, google.com, etc. (but you need to have your very own unique one!)
2. Have a server listen to requests coming into that domain
3. Give the server files to respond with



Hosting Our Own Sites

If we want to, we are able to host our own websites!

In order to accomplish this, there are a few steps:

1. Register a domain
 - a. example.com, google.com, etc. (but you need to have your very own unique one!)
2. Have a server listen to requests coming into that domain
 - a. Many services will allow you to host a website on their servers!
3. Give the server files to respond with



Hosting Our Own Sites

If we want to, we are able to host our own websites!

In order to accomplish this, there are a few steps:

1. Register a domain
 - a. example.com, google.com, etc. (but you need to have your very own unique one!)
 2. Have a server listen to requests coming into that domain
 - a. Many services will allow you to host a website on their servers!
 3. Give the server files to respond with
 - a. Some services will enable you to build a website using their tools, but you can write the HTML for your very own website all by yourself if you want!
- 