

# [301] Web 2

Tyler Caraza-Harter

# Learning Objectives Today

Make your own website!

## Learn HTML Basics

- syntax for tags, attributes, etc
- hyperlinks, tables, images

## Generate HTML

- links
- tables
- multiple pages

# Outline

Hypertext

Tag Syntax

Hyperlinks and Attributes

Images

Tables

Self Learning

# Hypertext

index.html (common home page name)

**Welcome to the home page!**

**Please visit the other pages:**

- [about](#)
- [contact](#)

# Hypertext

index.html (common home page name)

**Welcome to the home page!**

Please visit the other pages:

- [about](#)
- [contact](#)

Hypertext is text with clickable links to other pages

# Hypertext

index.html (common home page name)

Welcome to the home page!

Please visit the other pages:

- [about](#)
- [contact](#)

contact.html

**Contact Us**

Email: [tylerharter@gmail.com](mailto:tylerharter@gmail.com)

Phone: 123-456-7890

Hypertext is text with clickable links to other pages

# Hypertext

index.html (common home page name)

Welcome to the home page!

Please visit the other pages:

- [about](#)
- [contact](#)

Two steps for dealing with Hypertext:

1. how to **retrieve** hypertext
2. how to **display** hypertext

contact.html

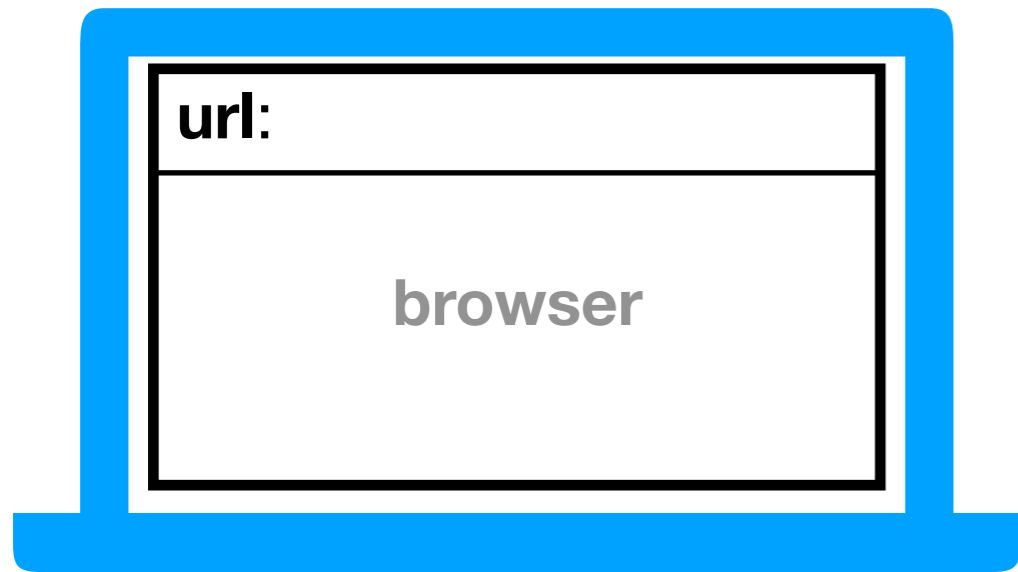
**Contact Us**

Email: [tylerharter@gmail.com](mailto:tylerharter@gmail.com)

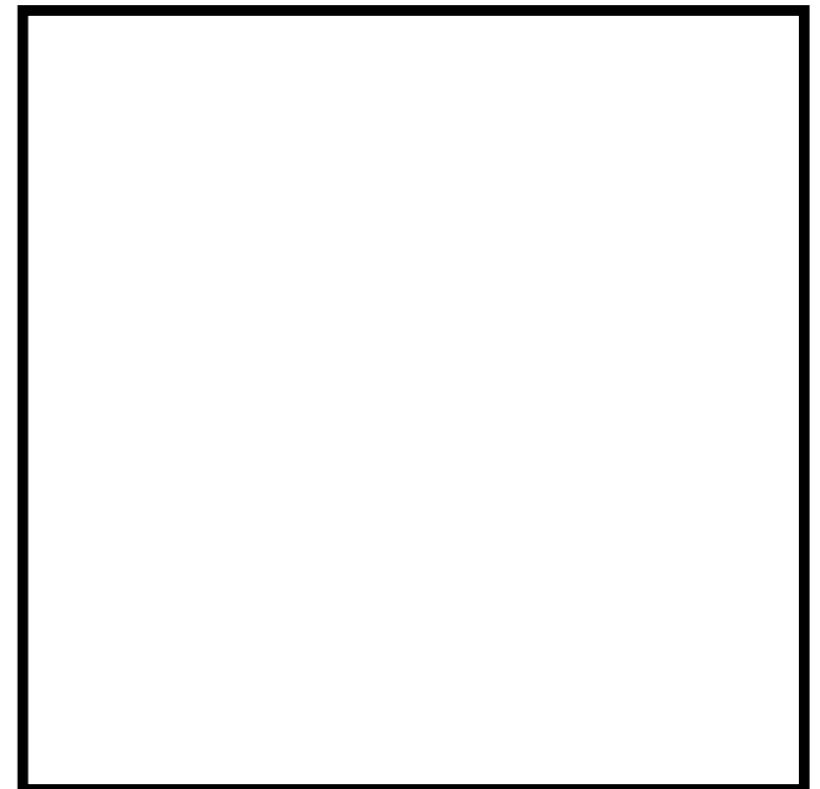
Phone: 123-456-7890

Hypertext is text with clickable links to other pages

# Hypertext



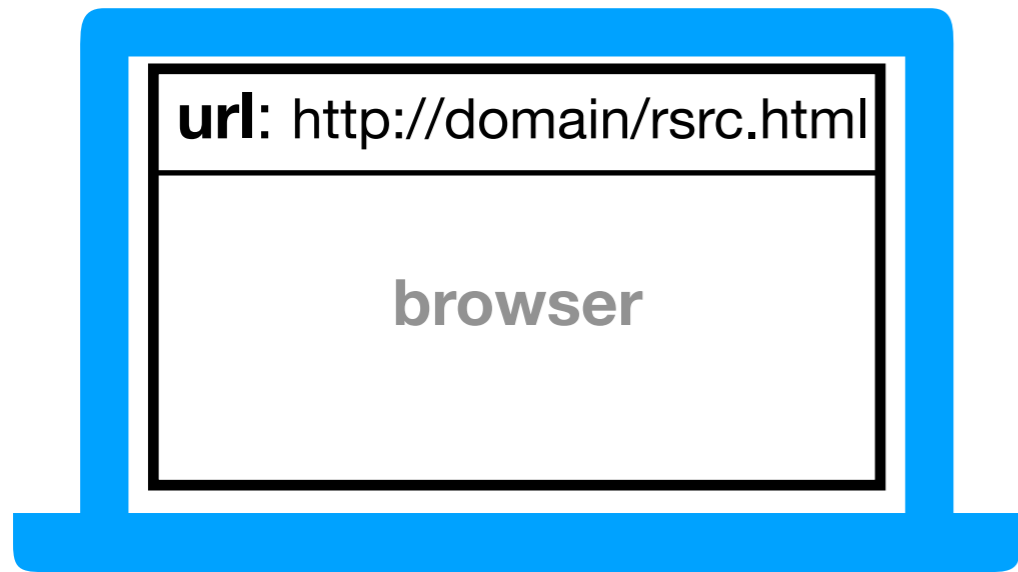
**computer 1**



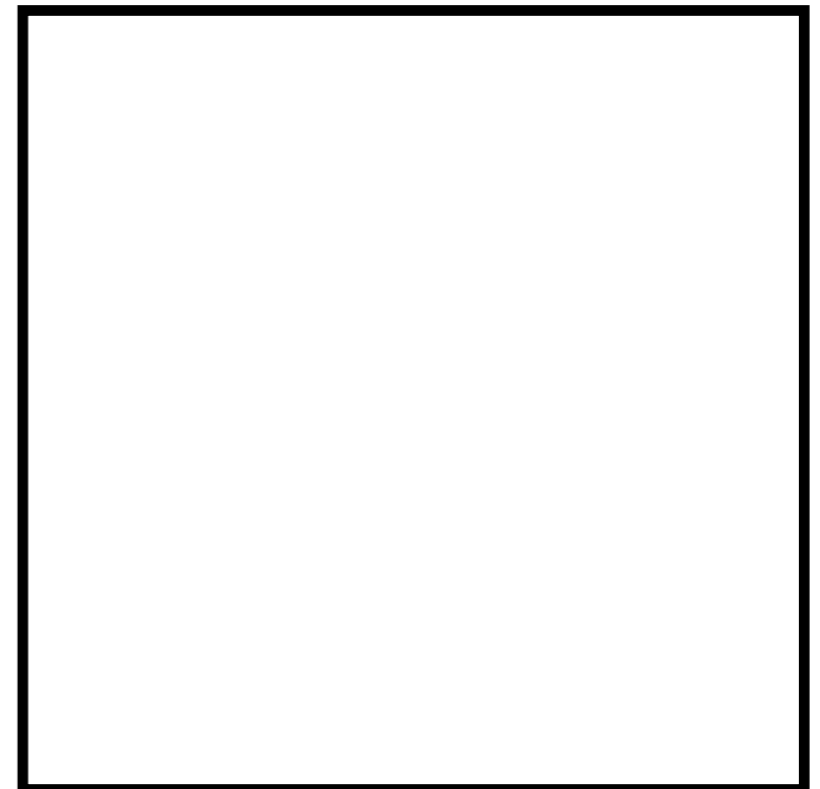
**computer 2**



# Hypertext

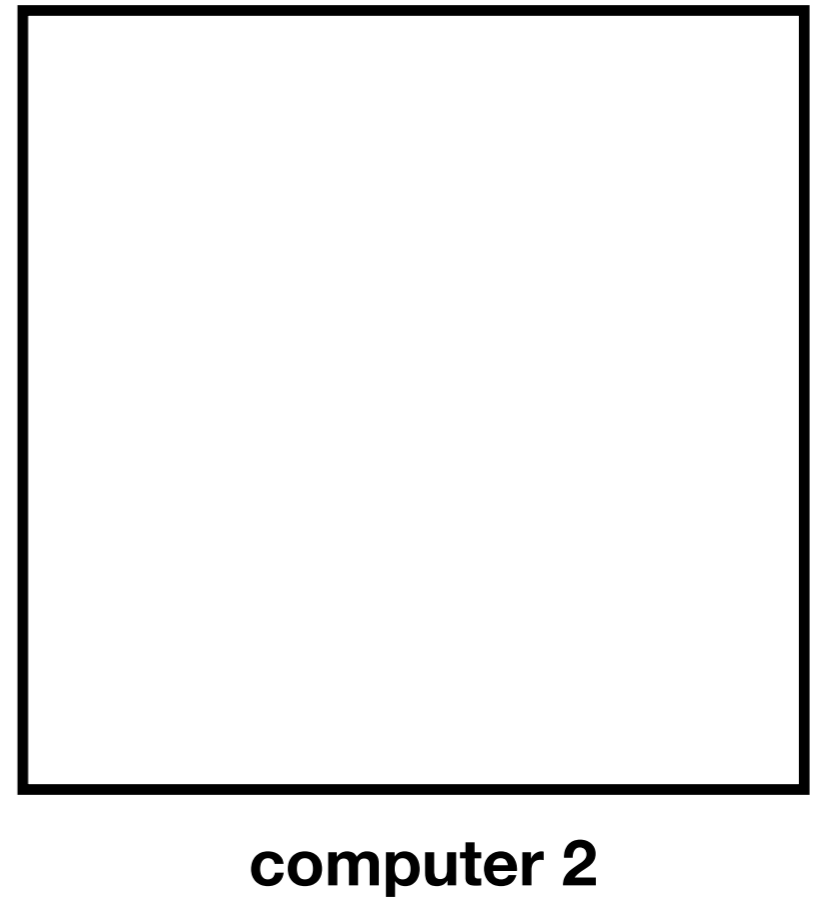
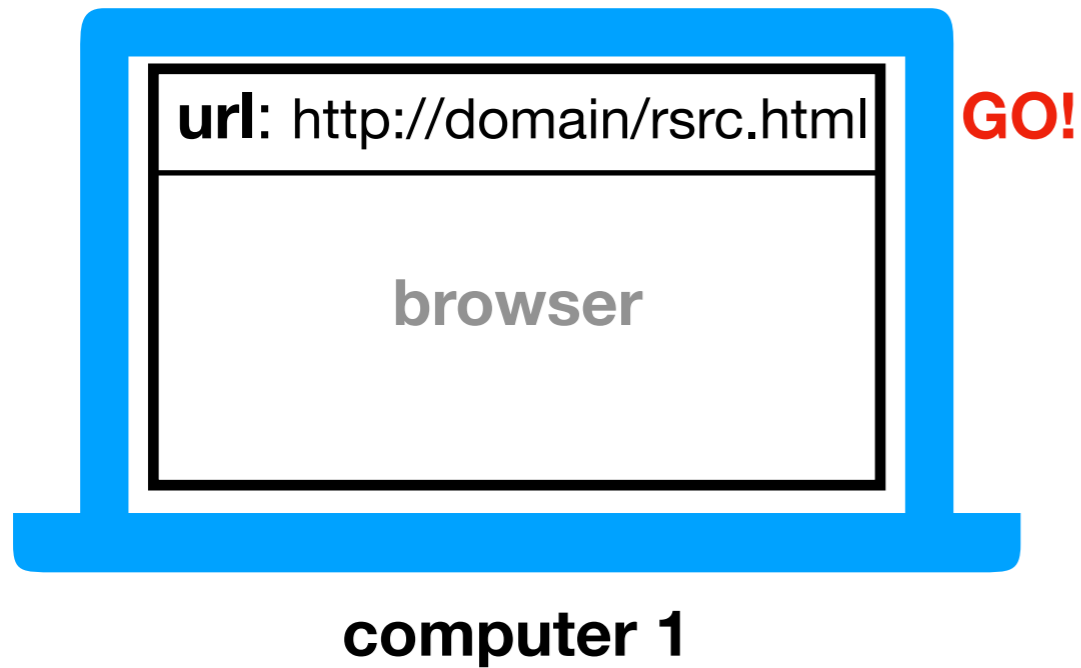


**computer 1**



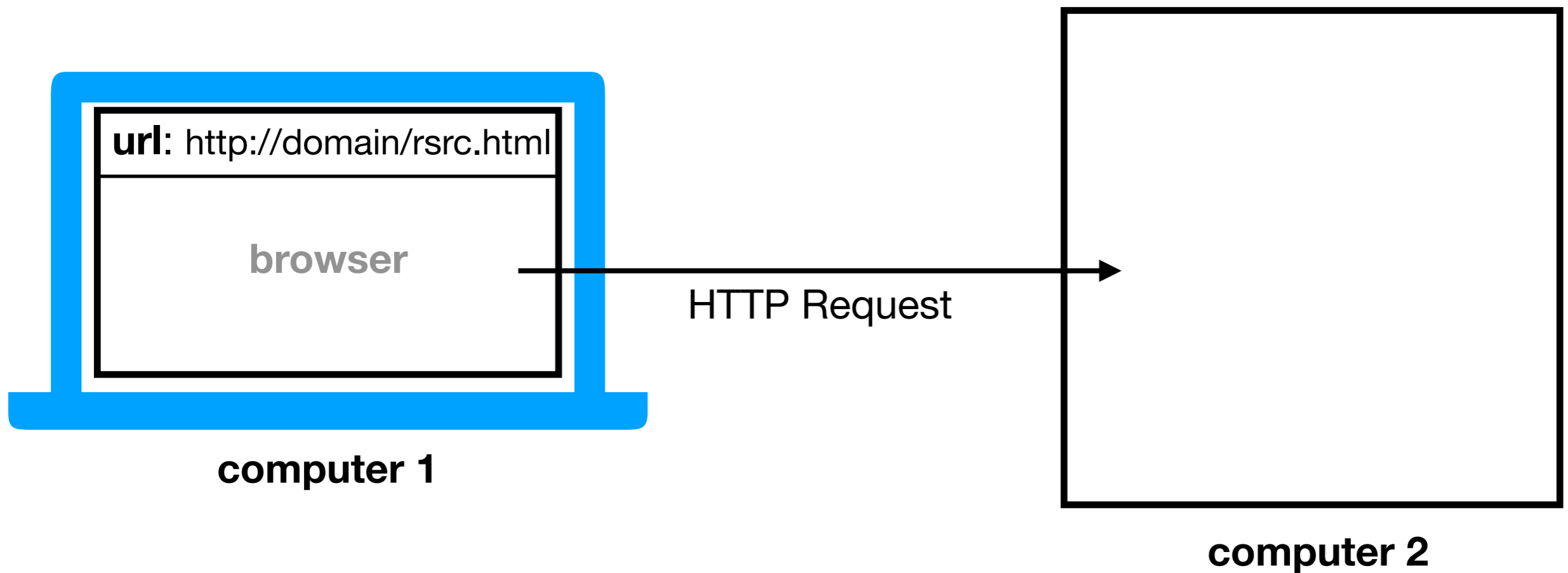
**computer 2**

# Hypertext



# Hypertext

**Step 1:** use HTTP (Hypertext Transfer Protocol) to **transfer** page

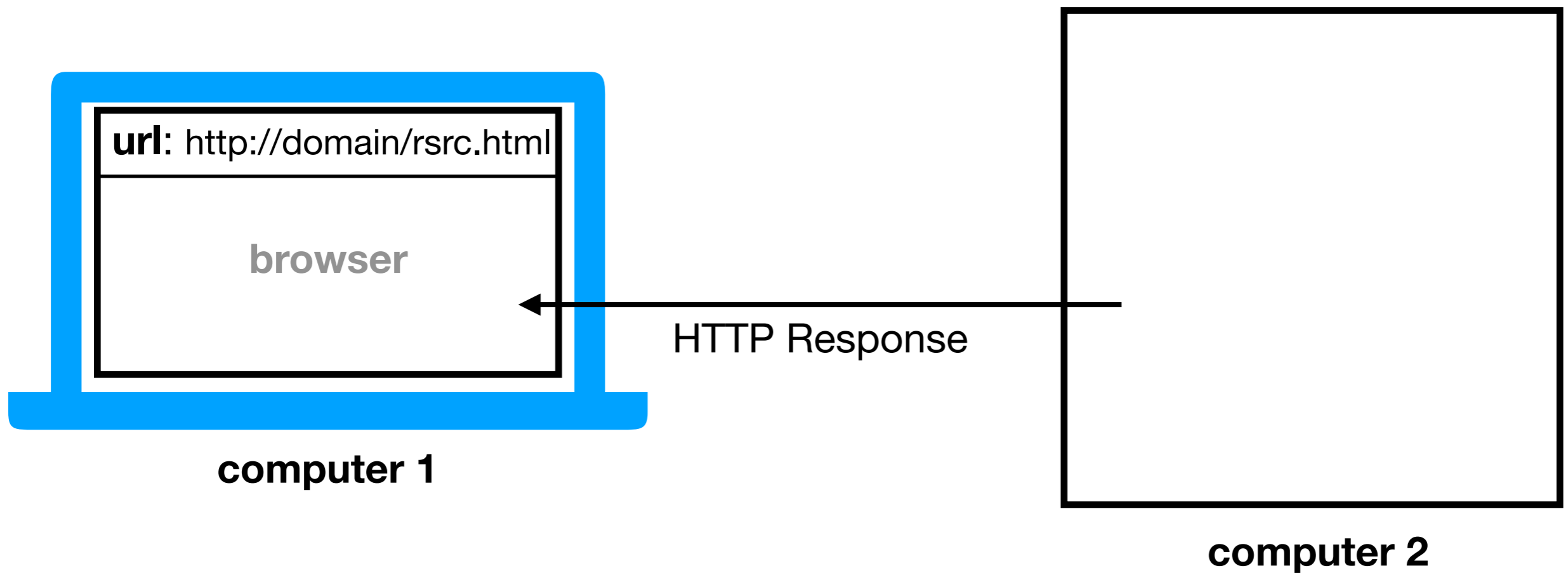


**HTTP Request:**

```
GET /rsrc.html HTTP/1.1  
Host: example.com  
User-Agent: ...  
Accept: */*
```

# Hypertext

Step 1: use HTTP (Hypertext Transfer Protocol) to **transfer** page



**HTTP Response:**

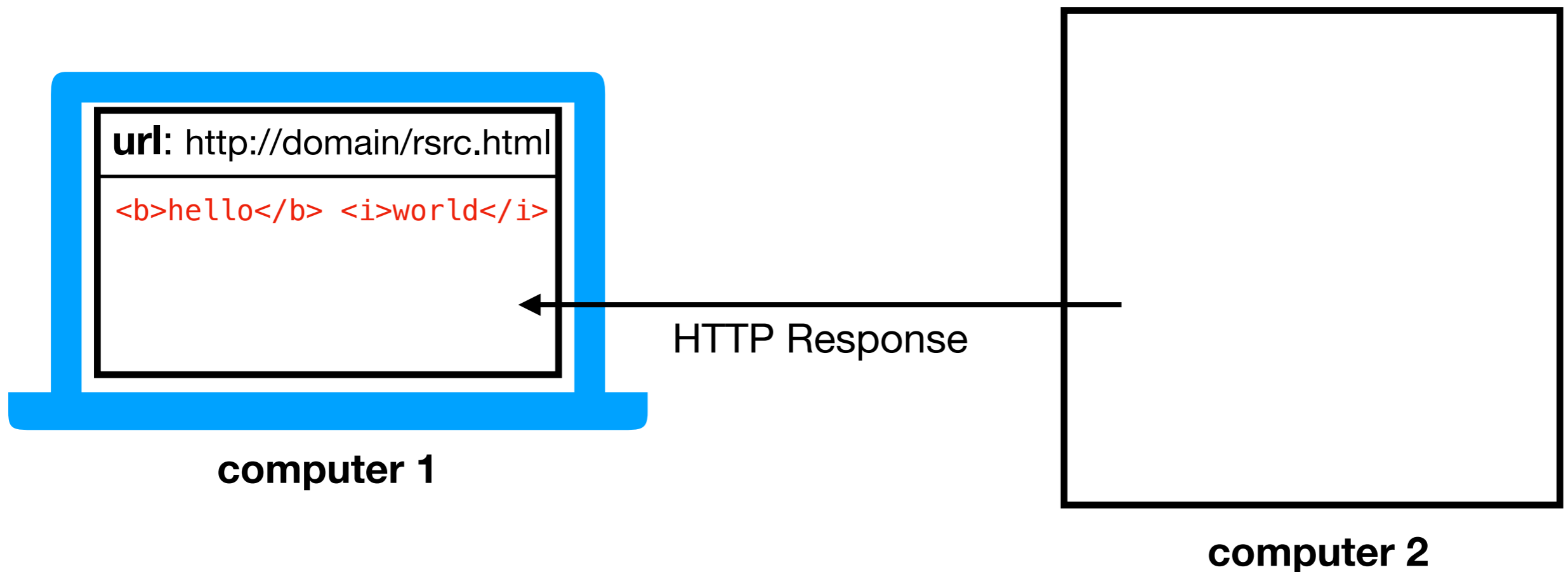
```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 74

<b>hello</b> <i>world</i>
```

# Hypertext

**Step 1:** use HTTP (Hypertext Transfer Protocol) to **transfer** page

**Step 2:** parse HTML (Hypertext Markup Language) to **render** page



**HTTP Response:**

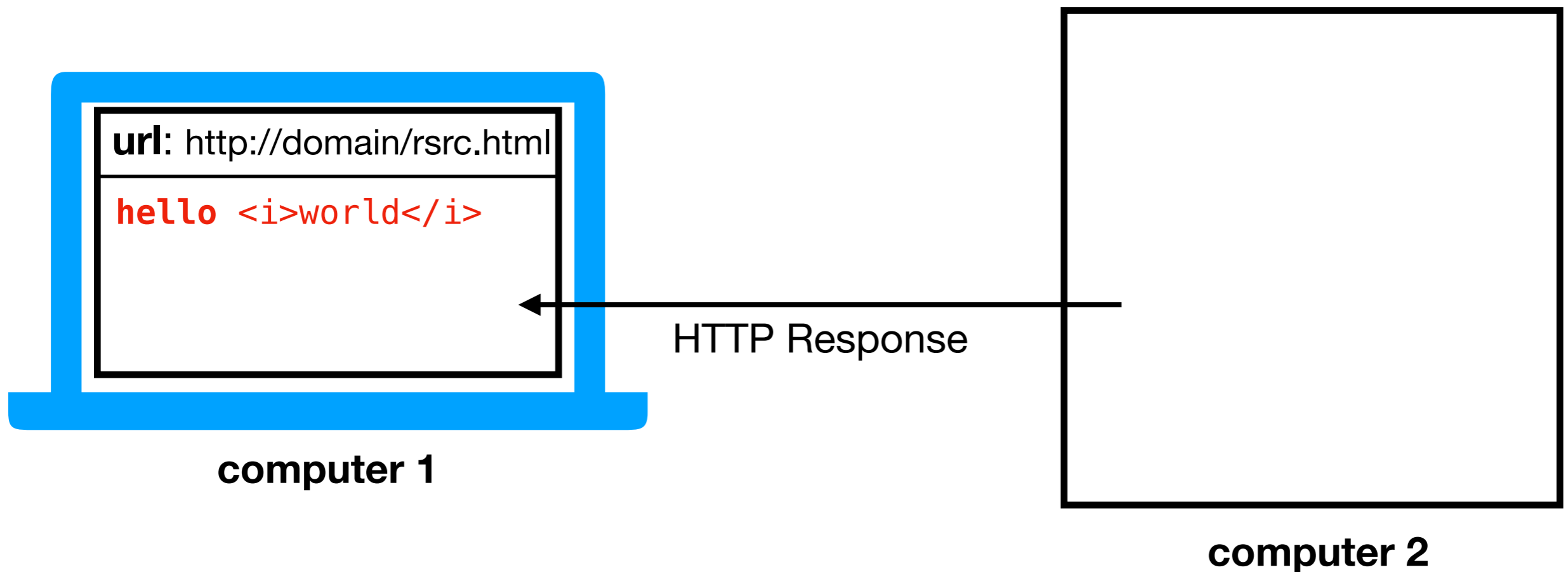
```
HTTP/1.0 200 OK  
Content-Type: text/html; charset=utf-8  
Content-Length: 74
```

```
<b>hello</b> <i>world</i>
```

# Hypertext

**Step 1:** use HTTP (Hypertext Transfer Protocol) to **transfer** page

**Step 2:** parse HTML (Hypertext Markup Language) to **render** page



**HTTP Response:**

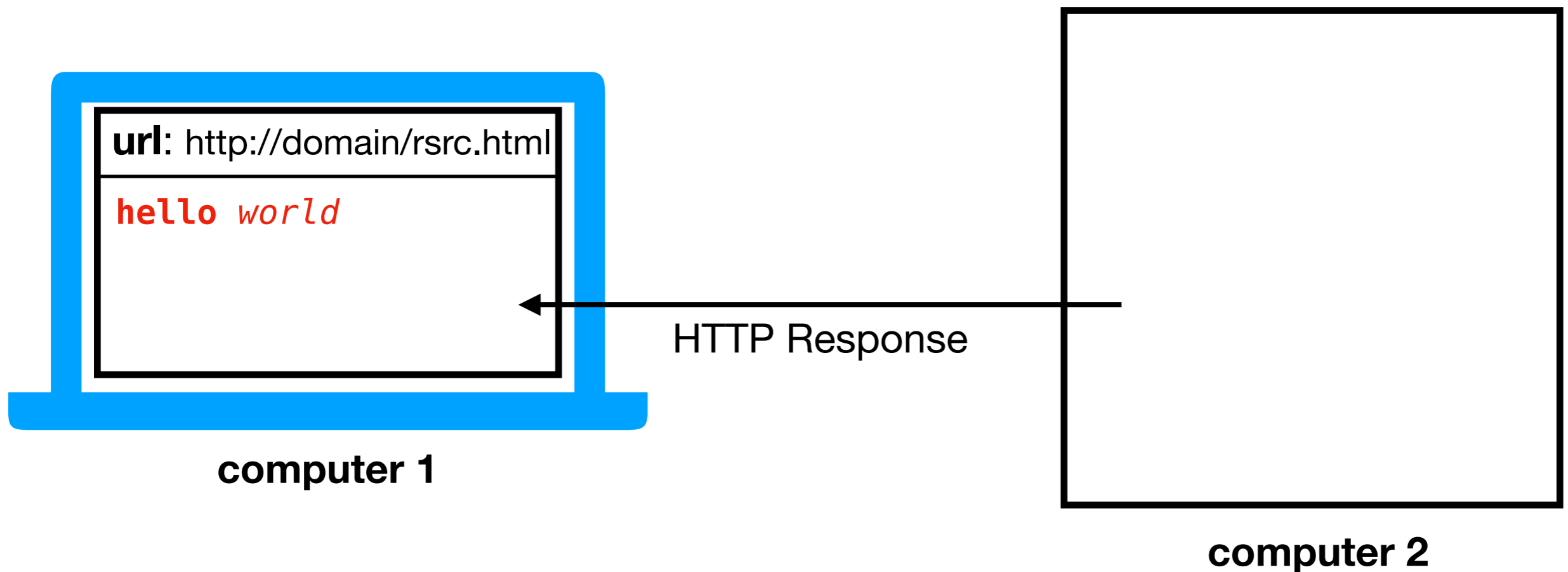
```
HTTP/1.0 200 OK  
Content-Type: text/html; charset=utf-8  
Content-Length: 74
```

```
<b>hello</b> <i>world</i>
```

# Hypertext

**Step 1:** use HTTP (Hypertext Transfer Protocol) to **transfer** page

**Step 2:** parse HTML (Hypertext Markup Language) to **render** page



**HTTP Response:**

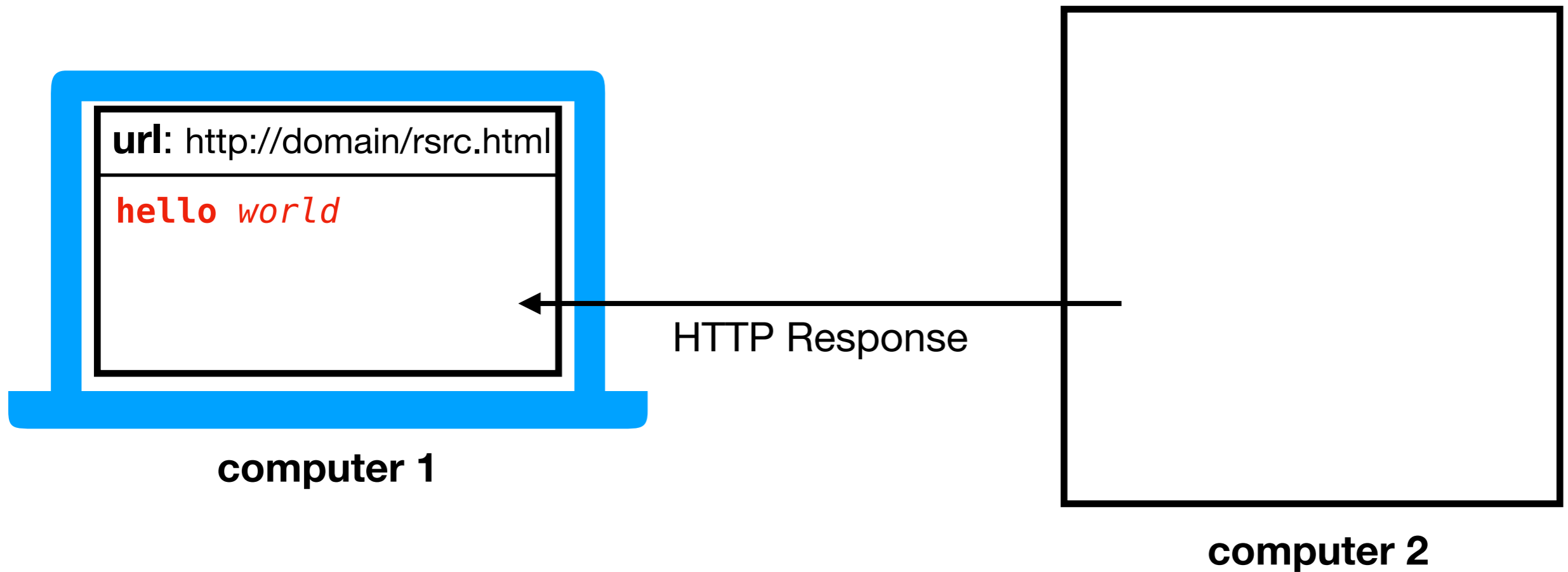
```
HTTP/1.0 200 OK  
Content-Type: text/html; charset=utf-8  
Content-Length: 74
```

```
<b>hello</b> <i>world</i>
```

# Hypertext

**Step 1:** use HTTP (Hypertext Transfer Protocol) to **transfer** page

**Step 2:** parse HTML (Hypertext Markup Language) to **render** page



**HTTP Response:**

```
HTTP/1.0 200 OK
Content-Type: text/html; charset=utf-8
Content-Length: 74
```

`<b>hello</b> <i>world</i>`

today we'll learn  
how to write HTML



# Text files vs HTML files

## Compare:

<https://tyler.caraza-harter.com/hello.txt>

<https://tyler.caraza-harter.com/hello.html>

## Inspecting:

- view source
- download page source (open locally)

# Outline

Hypertext

Tag Syntax

Hyperlinks and Attributes

Images

Tables

Self Learning

# HTML Tags

We can enclose text in “tags” to change how it is displayed

- often tags come in pairs (to indicate range of text to format)

this is regular text that is before `<TAG>` special text `</TAG>` followed by more regular text

# HTML Tags

We can enclose text in “tags” to change how it is displayed


- often tags come in pairs (to indicate range of text to format)

this is regular text that is before **<TAG>** special text **</TAG>** followed by more regular text

**opening tag:**  
tag name in angle brackets



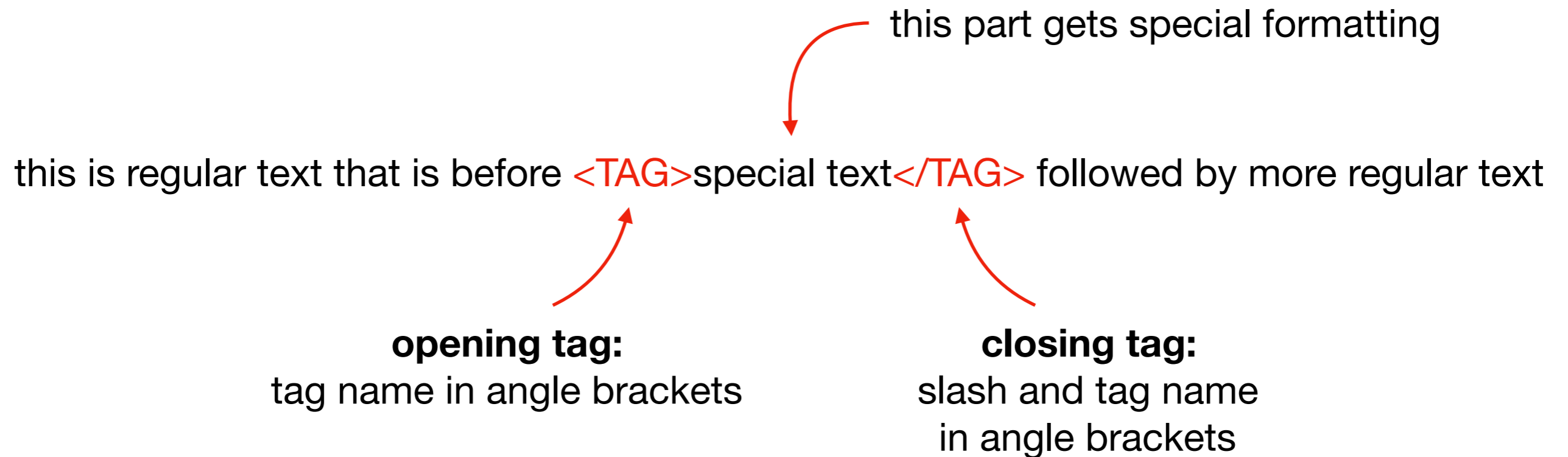
**closing tag:**  
slash and tag name  
in angle brackets



# HTML Tags

We can enclose text in “tags” to change how it is displayed

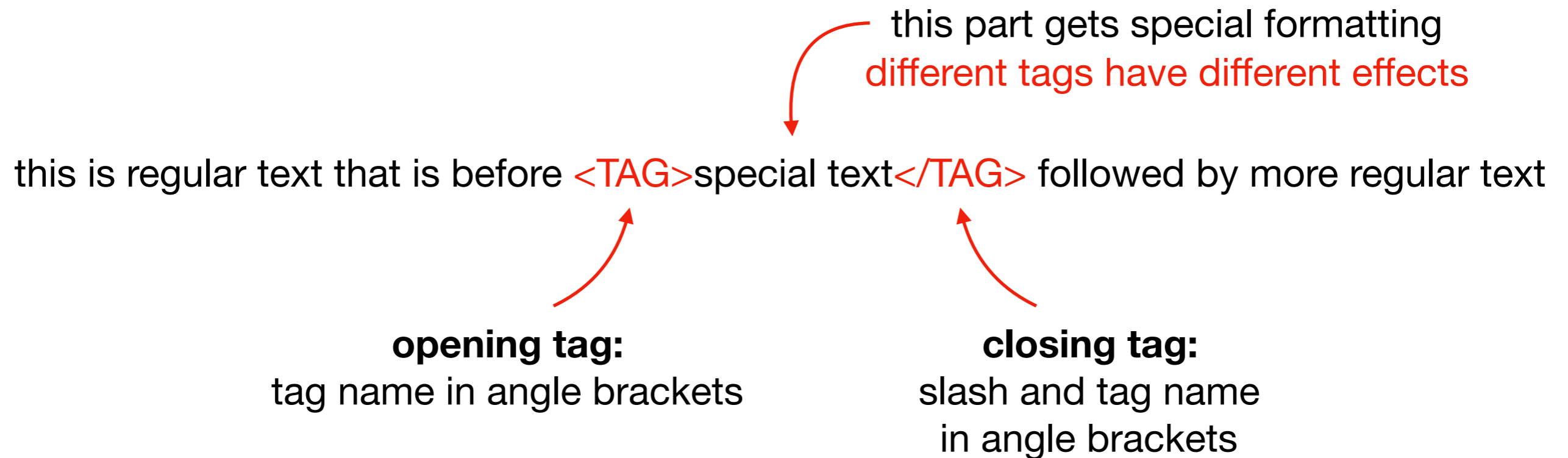
- often tags come in pairs (to indicate range of text to format)



# HTML Tags

We can enclose text in “tags” to change how it is displayed

- often tags come in pairs (to indicate range of text to format)



# Simple Tags

**test.html**

This is a test page

This is a test page

# Simple Tags

b: bold

test.html

This is a `<b>test</b>` page

This is a **test** page



# Simple Tags

b: bold

test.html

```
This is a <b>test page</b>
```

This is a **test page**

# Simple Tags

i: italic

test.html

```
This is a <i>test page</i>
```

This is a *test page*

# Simple Tags

i: italic

test.html

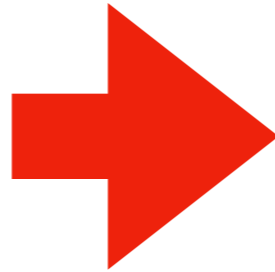
```
<i>This</i> is a <i>test page</i>
```



*This is a test page*

# Vertical Space

whitespace in  
html is ignored



**test.html**

This is a test page.

It is awesome.

This is a test page. It is  
awesome.

# Vertical Space

p: paragraph

test.html

```
<p>This is a test page.</p>
```

```
<p>It is awesome.</p>
```



This is a test page.

It is awesome.

# Vertical Space

h1: big header

test.html

```
<h1>Welcome</h1>  
  
<p>This is a test page.</p>  
  
<p>It is awesome.</p>
```



# Welcome

This is a test page.

It is awesome.

# Vertical Space

h1: big header

h2: smaller header

test.html

```
<h1>Welcome</h1>
```

```
<h2>This is a test page.</h2>
```

```
<p>It is awesome.</p>
```



**Welcome**

**This is a test page.**

It is awesome.

# Vertical Space

h1: big header

h2: smaller header

hN: etc

test.html

```
<h1>Welcome</h1>
```

```
<h2>This is a test page.</h2>
```

```
<p>It is awesome.</p>
```



**Welcome**

**This is a test page.**

It is awesome.



# Vertical Space

br: line break

test.html

This is a test page.

`<br>`

It is awesome.

This is a test page.

It is awesome.

# Vertical Space

br: line break

test.html

This is a test page.

`<br><br>`

It is awesome.

This is a test page.

It is awesome.

# Vertical Space

br: line break

test.html

This is a test page.

```
<br><br><br>
```

It is awesome.

This is a test page.

It is awesome.

# Vertical Space

br: line break

test.html

This is a test page.

```
<br><br><br><br>
```

It is awesome.

This is a test page.

It is awesome.

# Vertical Space

br: line break

test.html

```
This is a test page.  
<br><br><br><br>  
It is awesome.
```

note there is no closing tag  
(these are known as void elements)

This is a test page.

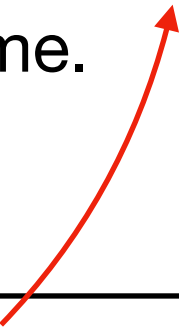
It is awesome.

# Vertical Space

br: line break

test.html

```
This is a test page.  
<br /><br /><br /><br />  
It is awesome.
```



sometimes you'll encounter it like this

This is a test page.

It is awesome.

# Lists

ul: unordered list

li: list item

test.html

Items:

```
<ul>  
  <li>Item X</li>  
</ul>
```

Items:

- Item X

# Lists

ul: unordered list

li: list item

test.html

Items:

```
<ul>  
  <li>Item X</li>  
  <li>Item Y</li>  
</ul>
```

Items:

- Item X
- Item Y



# Lists

ul: unordered list

li: list item

test.html

Items:

```
<ul>  
  <li>Item X</li>  
  <li>Item Y</li>  
  <li>Item Z</li>  
</ul>
```

Items:

- Item X
- Item Y
- Item Z

# Lists

ul: unordered list

li: list item

test.html

Items:

```
<ul>  
  <li>Item X  
  <li>Item Y  
  <li>Item Z  
</ul>
```

closing tags  
are optional  
for list items

Items:

- Item X
- Item Y
- Item Z

# Demo 1: List Visualization

Goal: convert Python list to bulleted list in HTML

## Input:

- arguments on command line

## Output:

- HTML file with bulleted list

## Example:

```
python bullets.py apple broccoli cabbage
```



Items:

- apple
- broccoli
- cabbage

# Complete Web Page

whole page is usually  
in html and body tags

**test.html**

This is a `<b>test</b>` page

This is a **test** page

# Complete Web Page

whole page is usually  
in html and body tags

**test.html**

```
<html>  
  <body>  
    This is a <b>test</b> page  
  </body>  
</html>
```



This is a **test** page

# Complete Web Page

you can also have  
a **head** tag with  
various metadata  
(e.g., title, keywords)

**test.html**

```
<html>  
  <head>  
  
  </head>  
  <body>  
    This is a <b>test</b> page  
  </body>  
</html>
```



This is a **test** page

# Complete Web Page

you can also have  
a **head** tag with  
various metadata  
(e.g., title, keywords)

**test.html**

```
<html>
  <head>
    <title>Test Page</title>
  </head>
  <body>
    This is a <b>test</b> page
  </body>
</html>
```



This is a **test** page

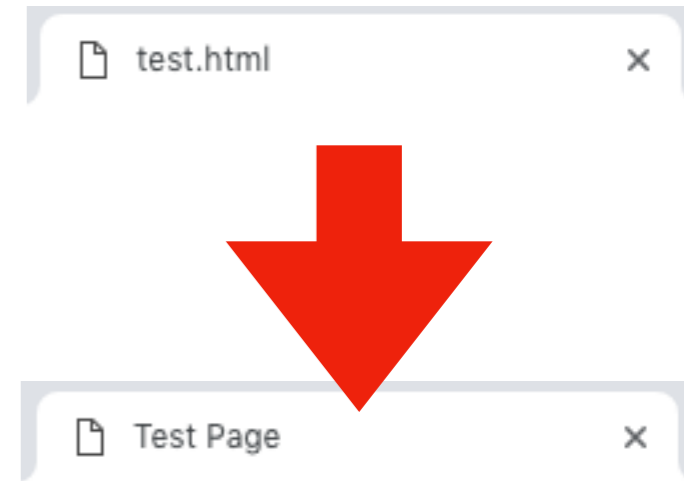
# Complete Web Page

you can also have  
a **head** tag with  
various metadata  
(e.g., title, keywords)

**test.html**

```
<html>
  <head>
    <title>Test Page</title>
  </head>
  <body>
    This is a <b>test</b> page
  </body>
</html>
```

This is a **test** page





# Outline

Hypertext

Tag Syntax

Hyperlinks and Attributes

Images

Tables

Self Learning

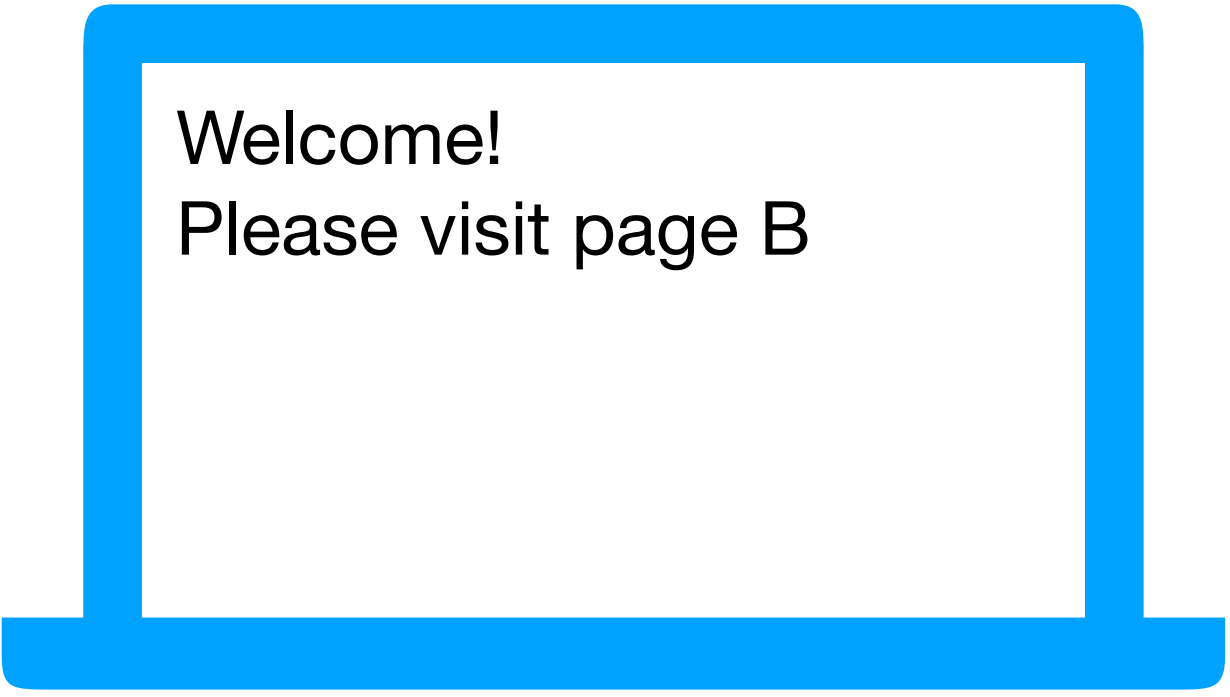
# Hyperlinks

## A.html

Welcome!<br>  
Please visit page B

## B.html

You're on page B.



Welcome!  
Please visit page B

# Hyperlinks

**we want this to be clickable**

- use `<a>` tag

**A.html**

```
Welcome!<br>
Please visit <a>page B</a>
```

**B.html**

```
You're on page B.
```



Welcome!  
Please visit page B

# Hyperlinks

**we want this to be clickable**

- use `<a>` tag

## A.html

Welcome!<br>

Please visit `<a>`page B`</a>`

**where does it go to?**

## B.html

You're on page B.

Welcome!

Please visit **page B**

# Hyperlinks

**we want this to be clickable**

- use `<a>` tag
- use href attribute

## A.html

```
Welcome!<br>
Please visit
<a href="B.html">page B</a>
```

## B.html

```
You're on page B.
```

page B' is on the second line. The text 'page B' is underlined and blue, indicating a clickable link." data-bbox="50 600 498 935"/>

Welcome!  
Please visit [page B](#)

# Hyperlinks

## we want this to be clickable

- use `<a>` tag
- use href attribute

inside an opening tag, we often set parameters by using this:

`attribute-name="attribute-value"`

## A.html

```
Welcome!<br>
Please visit
<a href="B.html">page B</a>
```

## B.html

```
You're on page B.
```



Welcome!

Please visit [page B](#)

# Hyperlinks

**we want this to be clickable**

- use `<a>` tag
- use href attribute

inside an opening tag, we often set parameters by using this:

`attribute-name="attribute-value"`

## A.html

```
Welcome!<br>
Please visit
<a href="B.html">page B</a>
```

## B.html

You're on page B.

Welcome!

Please visit [page B](#)  
click!

# Hyperlinks

## we want this to be clickable

- use `<a>` tag
- use href attribute

inside an opening tag, we often set parameters by using this:

`attribute-name="attribute-value"`

## A.html

```
Welcome!<br>
Please visit
<a href="B.html">page B</a>
```

## B.html

```
You're on page B.
```



You're on page B.



# Demo 2: Dictionary Visualization

Goal: generate HTML page for every dictionary value and have a keys.html page that links to each of them

## Input:

- a string containing a JSON dictionary

## Output:

- a keys.html file and an HTML file for each value

## Example:

```
python dict-vis.py '{"A": "val 1", "B": "val 2"}'
```



**keys.html**

Keys:

- [A](#)
- [B](#)

**A.html**

val 1

**B.html**

val 2

# Outline

Hypertext

Tag Syntax

Hyperlinks and Attributes

Images

Tables

Self Learning

# Images

img: image

test.html

```

```



# Images

img: image

test.html

```
W  
  
E
```

W  python™ E

# Images

img: image

test.html

```
N<br>
W

E<br>
S
```

N

W

S



python™

E

# Images

img: image

test.html

```
N<br>
W

E<br>
S
```



# Images

img: image

test.html

```
N<br>  
W  
  
E<br>  
S
```



# Images

img: image

test.html

```
N<br>  
W  
  
E<br>  
S
```





# Outline

Hypertext

Tag Syntax

Hyperlinks and Attributes

Images

Tables

Self Learning

# Tables


**table:** a table

**tr:** table row

**td:** table data (cell)

test.html

```
<table>
  <tr>
    <td>A</td>
    <td>B</td>
  </tr>
  <tr>
    <td>C</td>
    <td>D</td>
  </tr>
</table>
```



A	B
C	D

# Tables

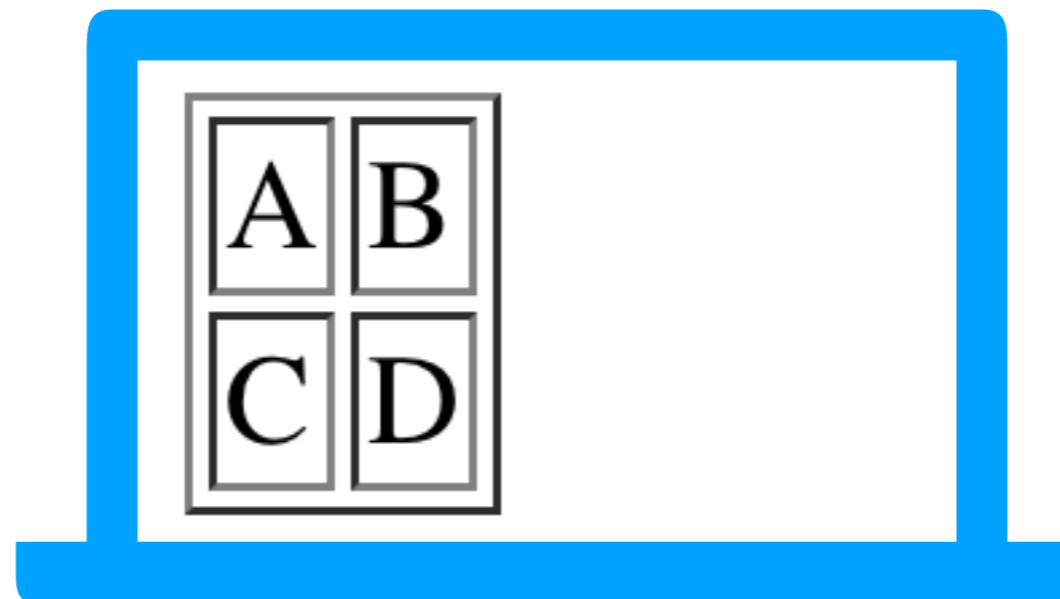
**table:** a table

**tr:** table row

**td:** table data (cell)

test.html

```
<table border="1">  
  <tr>  
    <td>A</td>  
    <td>B</td>  
  </tr>  
  <tr>  
    <td>C</td>  
    <td>D</td>  
  </tr>  
</table>
```



A	B
C	D

# Tables

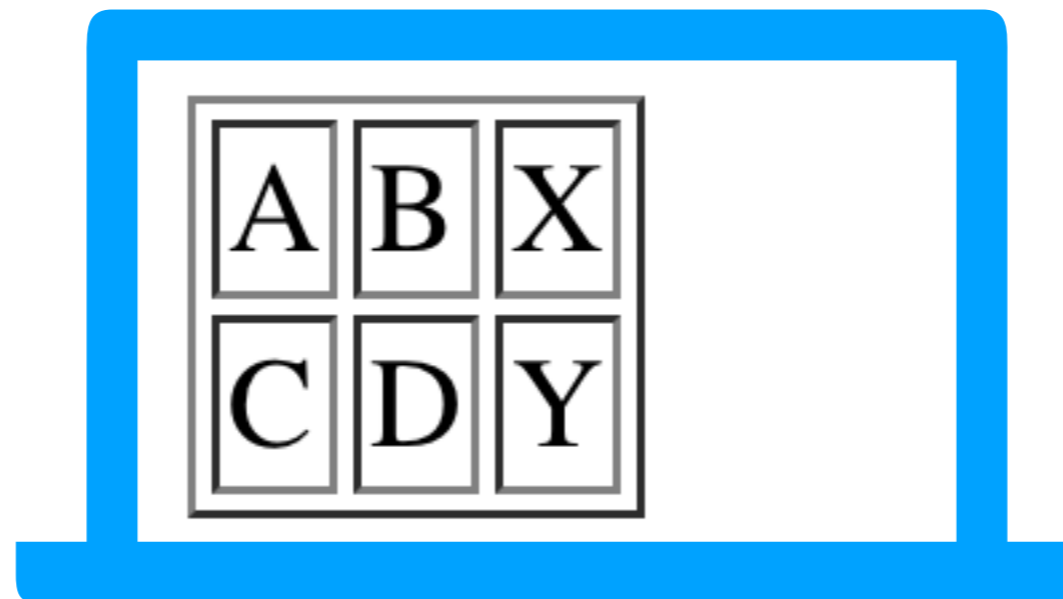
**table:** a table

**tr:** table row

**td:** table data (cell)

test.html

```
<table border="1">  
  <tr>  
    <td>A</td>  
    <td>B</td>  
    <td>X</td>  
  </tr>  
  <tr>  
    <td>C</td>  
    <td>D</td>  
    <td>Y</td>  
  </tr>  
</table>
```



A	B	X
C	D	Y

# Tables

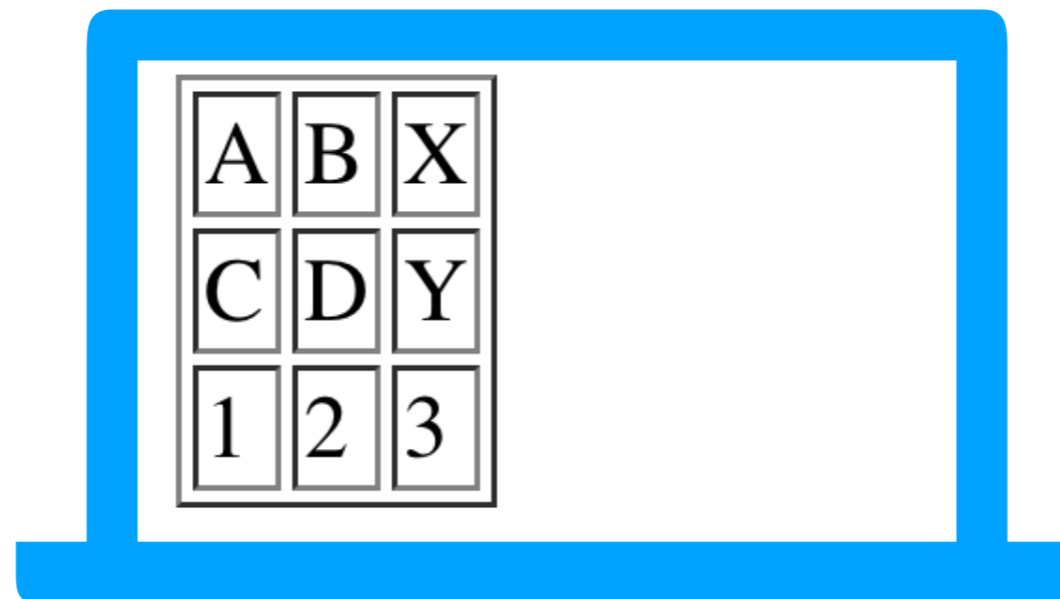
**table:** a table

**tr:** table row

**td:** table data (cell)

test.html

```
<table border="1">  
  <tr>  
    <td>A</td>  
    <td>B</td>  
    <td>X</td>  
  </tr>  
  <tr>  
    <td>C</td>  
    <td>D</td>  
    <td>Y</td>  
  </tr>  
  <tr><td>1</td>2<td></td><td>3</td></tr>  
</table>
```



A	B	X
C	D	Y
1	2	3

# Demo 3: DataFrame to HTML Table

Goal: convert a DataFrame to an HTML Table  
(note, you can do this more simply, but it's good practice)

## Input:

- a Pandas DataFrame

## Output:

- an HTML file

# Other Demos (time permitting)

## **Demo 4: Table Diff**

- convert two similar DataFrames to HTML tables, highlighting the differing cells with bold text

## **Demo 5: Search Highlighting**

- Download page and convert certain words (search terms) to bold text

## **Demo 6: Flash Cards**

- Show questions, and reveal answers when user clicks an answer hyperlink

# Outline

Hypertext

Tag Syntax

Hyperlinks and Attributes

Images

Tables

Self Learning



# Resources for Self Learning

There are many free online resources (e.g., w3schools) if you want to learn more about web development

## More **HTML**

- <https://www.w3schools.com/html/default.asp>
- More of what we learned today

## **CSS** (Cascading Style Sheets)

- <https://www.w3schools.com/html/default.asp>
- Control the aesthetics of your HTML with the CSS language

## **JavaScript**

- <https://www.w3schools.com/html/default.asp>
- Full programming language (like Python) for running code in a user's web browser