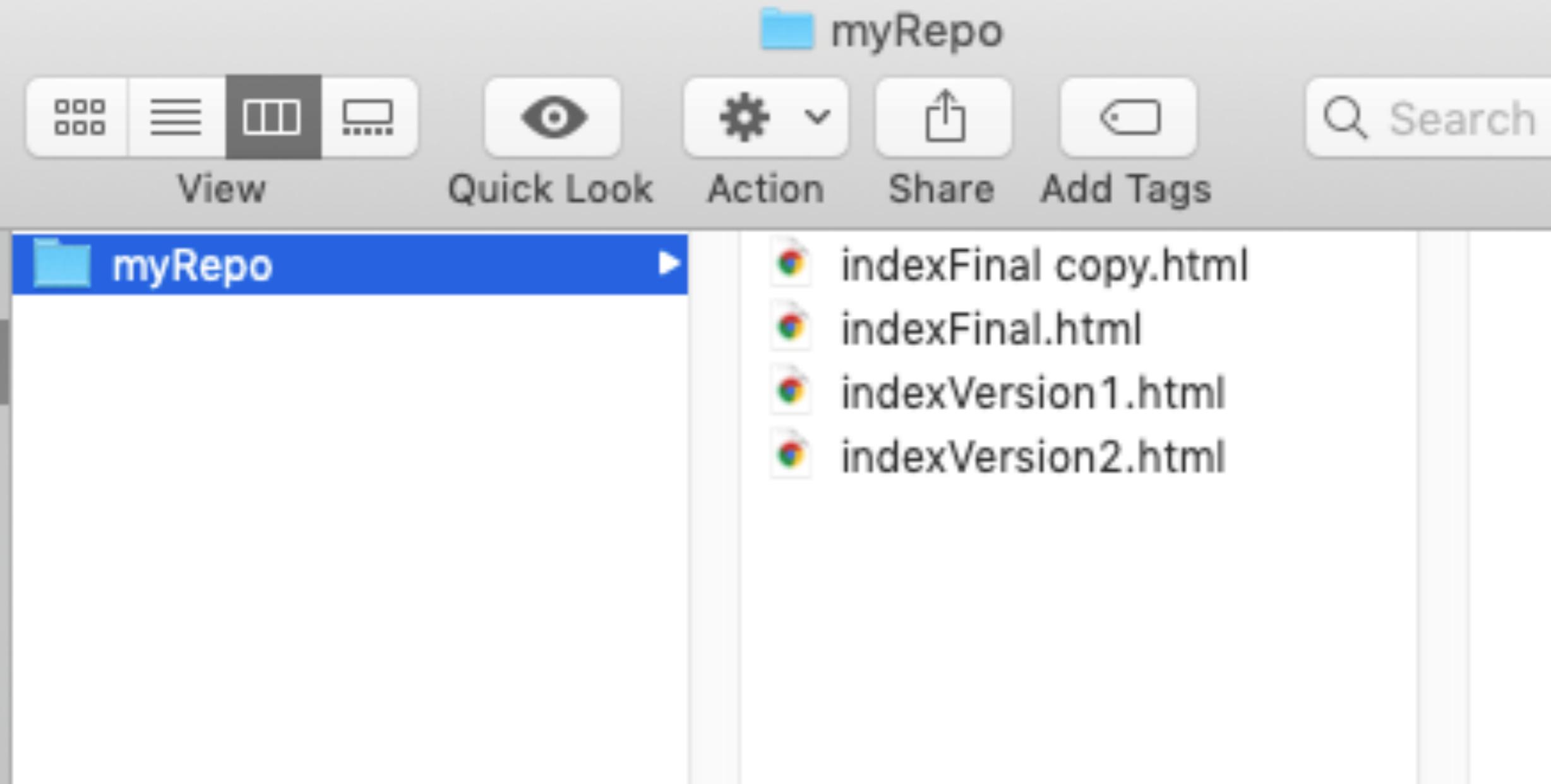


# Git

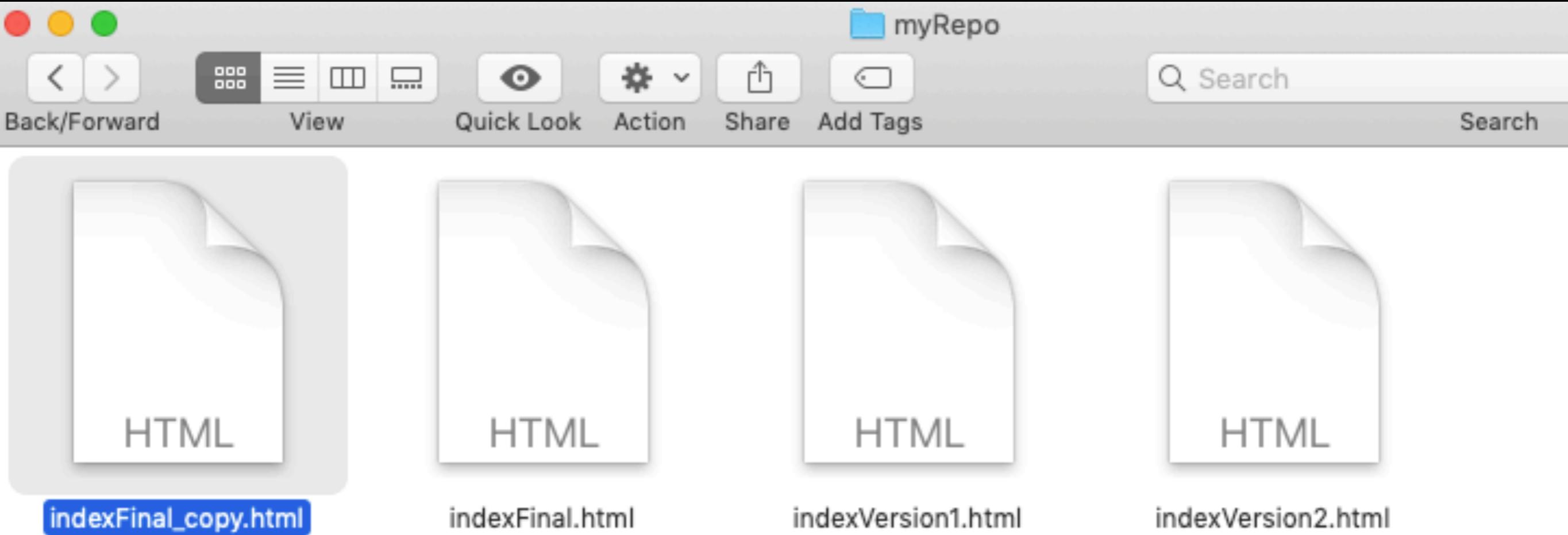
# What is Git?

A version control system meant to make it easier to have multiple versions of code, sometimes across multiple developers or teams

At its most simple, git helps with the following problem:



At its most simple, git helps with the following problem: **AKA**



At its most complex, git allows professional developers to work together worldwide on software projects without over writing each other's work, causing erased code, bugs and generally breaking collaborative projects.

It is also a great resource (**web site**) to find code examples and inspiration. If you haven't already, you will likely be introduced to it and be asked to implement existing code in your Creative Code class - among many other contexts. It's like a library - only instead of taking out books, you can take out software.

**\*\*Note:** Like a library in English class you should **NEVER** take credit for someone else's intellectual property. There is a grey area between **open source** and plagiarism.

Github

**Github is a service to host your code projects on the WWW in order to collaborate with other developers.**

Code is **pushed** (uploaded) from a local **directory** (folder) on your computer thus becoming what we call in GitHub speak: a **repository** or **repo**

example - our class site:

[http://www.github.com/rebleo/webDev\\_B\\_Fall2021](http://www.github.com/rebleo/webDev_B_Fall2021)

## Git vs GitHub.com

git is a version control system that takes snapshots of your code at certain points in development

These snapshots are stored on your **local machine's storage** in a **repo**, or **repository** (in git hub speak) this translates to **directory** in **Unix** speak or **folder** in general operating system speak

GitHub.com is a website that hosts git repositories on a remote server + is available for all the web to see, copy + implement.

# Git Terminology

**repository** - where data is managed. the directory containing your files.

**local** - the copy that exists on your machine, no one else can access this

**remote** - the copy in your github account, anyone with access to your github repo can access the remote instance (we won't be doing this!)

**push** - once you make changes to the local copy you \*upload the changes to the remote copy

**pull** - if someone else makes changes to the remote copy (we won't be doing this this semester)

**clone a repository** - download the entire codebase of the repo you can pull in changes + and push your own changes if you are given access

# Github pages

github.io

easily allows you to host web pages using github servers + workflow

url (uniform resource locator)

http://

yrUsername.github.io



**yrUsername.github.io**

prototype locally

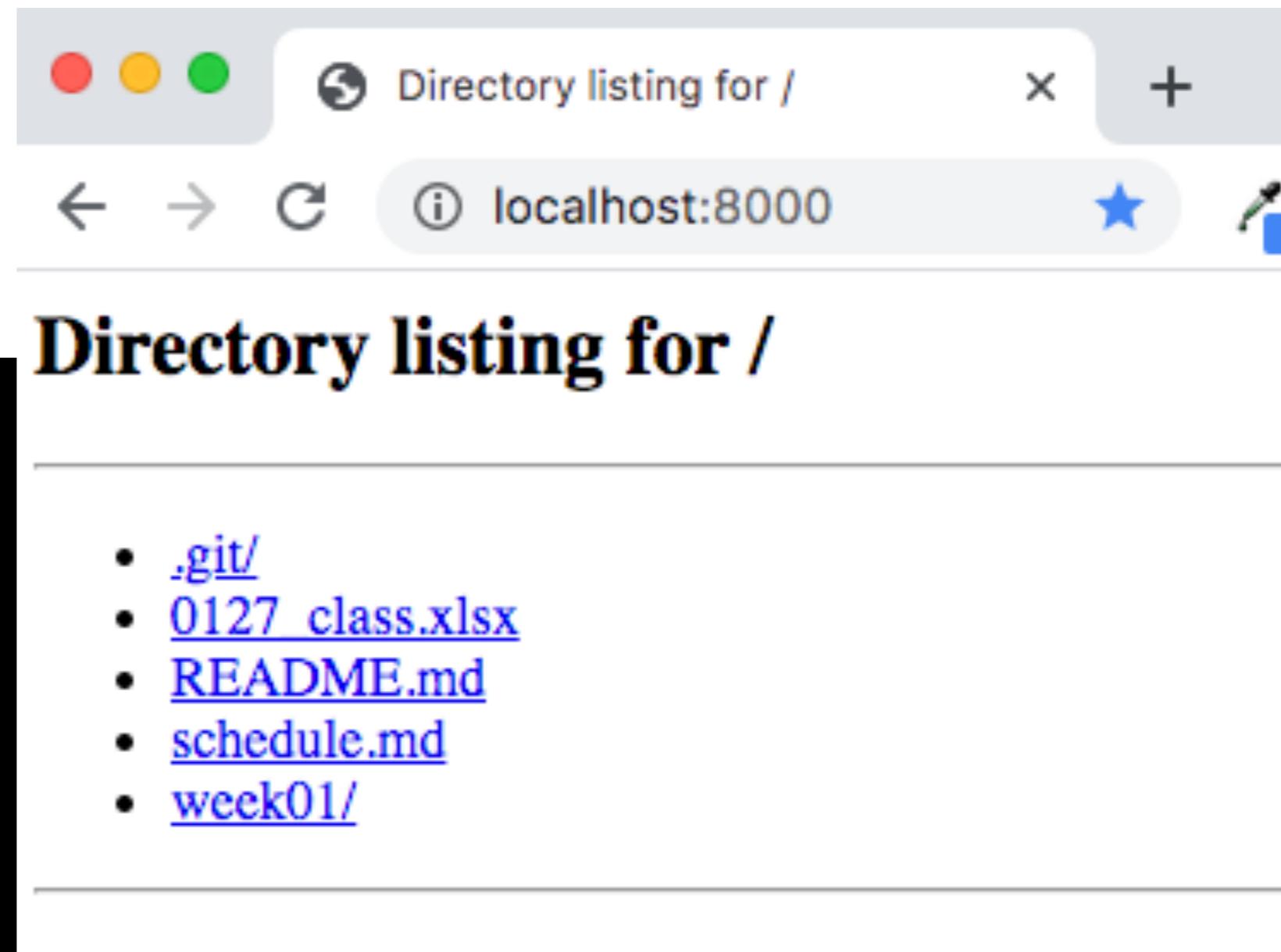
# prototype locally

w/ a local http server

**prototype:** local http server  
(using our local machine as a server!!)

**publish:** pushed to Github Pages

```
[ webDevSpring2020 ] python -m SimpleHTTPServer  
[ webDevSpring2020 ] Serving HTTP on 0.0.0.0 port 8000 ...
```



< HTML >

[ box model ]

**block** vs. **inline** display

The key to understanding how **HTML** + **CSS** works is to imagine that there is an invisible box around every **HTML** element.

Block level elements are outlined w/ red + inline elements in green.

**<body>** creates 1st box, then **<h1>, <h2>, <p>, <i> + <a>** each create their own boxes within it.

## The Cottage Garden

The *cottage garden* is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

The Cottage Garden originated in [England](#) and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained [English estate gardens](#).

The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

## The Cottage Garden

The *cottage garden* is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

The Cottage Garden originated in [England](#) and its history can be traced back for centuries, although they were re-invented in 1870's England, when stylized versions were formed as a reaction to the more structured and rigorously maintained [English estate gardens](#).

The earliest cottage gardens were more practical than their modern descendants, with an emphasis on vegetables and herbs, along with some fruit trees.

## < HTML >

3 categories of HTML elements

1 - **block**: large blocks of content has height + width

`<p>, <h1>, <blockquote>, <ol>, <ul>, <table>`

2 - **inline**: small about of content, no height or width

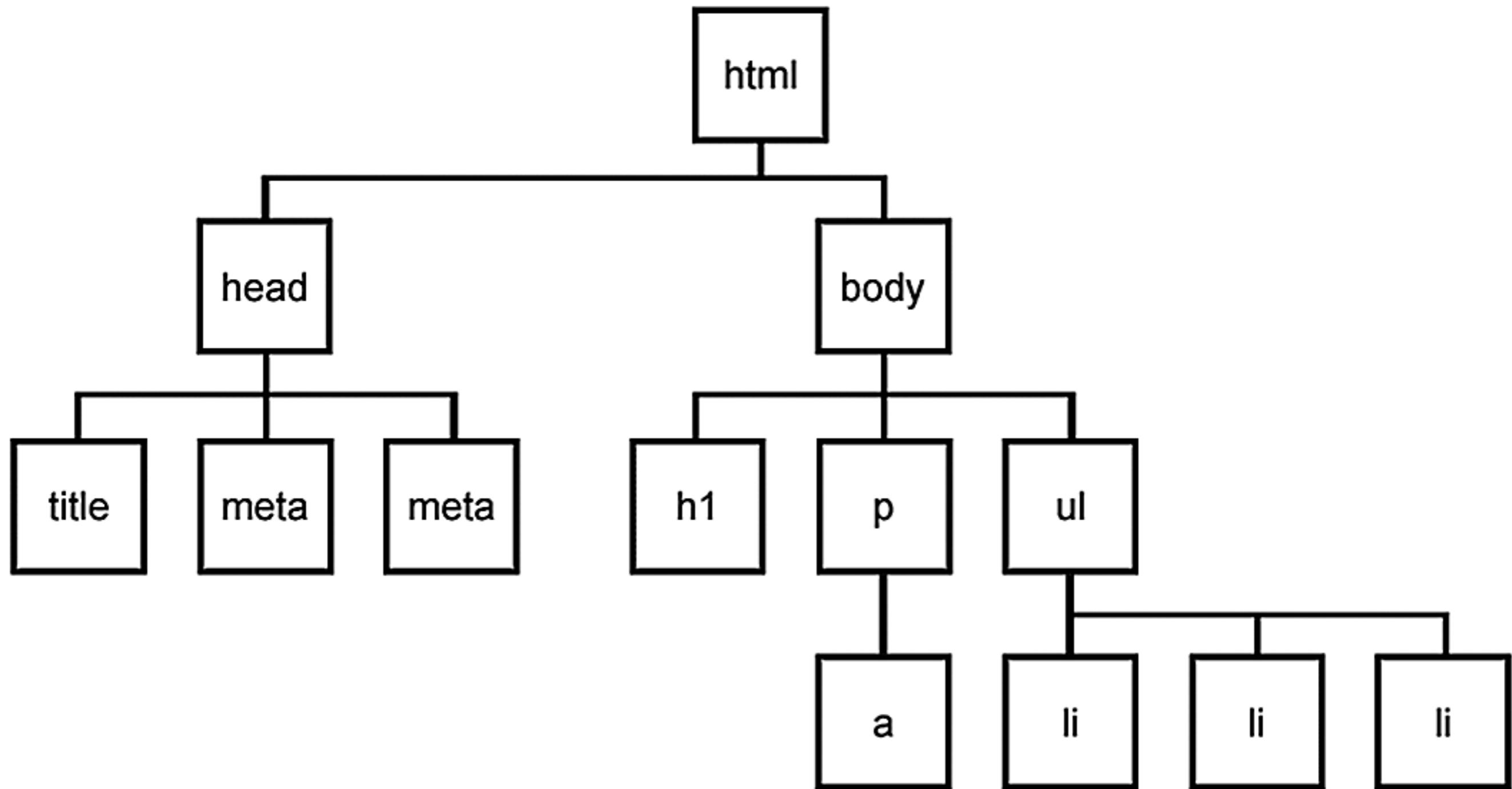
`<a>, <em>, <strong>, <br>, <span>, <time>`

a. **inline block**: inline content w/ height + width

3 - **metadata**: information abou the page, usually not visible

`<title>, <meta>, <script>`

# Parent / Child Element Structure



# Parent + Child

```
<!doctype html>
  <head>
    <title> Week 1 </title>
  </head>
  <body>
    <div>
      Here's a Great Site.
    </div>
  </body>
</html>
```

head is the parent of title

div is the child of body

body is the child of html

The `<head>` element contains the metadata for a web page. Metadata is information about the page that isn't displayed directly on the web page. Unlike the information inside of the `<body>` tag, the metadata in the head is information about the page itself.

# Text tags

**h1, h2, h3, h4, h5, h6** are text tags for headings

**p** is a tag for paragraphs

**b** is for bold, **i** is for italics

**<strong>** is for **bold** **<em>** is for *italics*

**ul, ol, li** are used for making lists

**ul**: unordered lists

**ol**: ordered lists

**li**: an individual list tag

**<br/>** will break to a new line

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

**<div>s can contain any text or other HTML elements, such as links, images, or videos. Remember to always add two spaces of indentation when you nest elements inside of <div>s for better readability.**

# Semantic HTML

HTML should be coded to represent the data that will be populated and not based on its default presentation styling. Presentation (how it should look), is the sole responsibility of CSS.

Some of the benefits from writing semantic markup are as follows:

- Search engines will consider its contents as important keywords to influence the page's search rankings (see SEO)
- Screen readers can use it as a signpost to help visually impaired users navigate a page
- Finding blocks of meaningful code is significantly easier than searching through endless divs with or without semantic or namespaced classes
- Suggests to the developer the type of data that will be populated
- Semantic naming mirrors proper custom element/component naming

# Semantic elements

<p>

<h1> - <h6>

<main>

dominant content of the <body> element

<article>

A document, page or site. This is usually a root container element after body

<section>

Generic section of a document

<header>

Intro section of a document

<footer>

Footer at end of a document or section

<nav>

Navigational section

Use these **before** div when appropriate.

# Semantic elements

represents a portion of a document whose content is only indirectly related to the document's main content. Asides are frequently presented as sidebars or call-out boxes.

**<details>** creates a disclosure widget in which information is visible only when the widget is toggled into an "open" state.

**<figcaption>** represents a caption or legend describing the rest of the contents of its parent **<figure>**

**<mark>** represents text which is marked or highlighted for reference or notation purposes, due to the marked passage's relevance or importance in the enclosing context.

**<summary>** element specifies a summary, caption, or legend for a **<details>** element's disclosure box. Clicking the **<summary>** element toggles the state of the parent **<details>** element open and closed.

**<time>** represents a specific period in time.

**tag    attribute    value**

<video src= "filepath/file.mov" alt= "this is the video" height="300"> </video>

<html attribute= "value" attribute= "value" attribute= "value"> </html>

# Absolute Links direct to another server

OPENING  
LINK TAG

URL WE ARE  
DIRECTED TO



TEXT WE  
CLICK ON

CLOSING  
TAG

```
<a href="https://www.youtube.com/watch?v=qcnnI6HD6DU"> absolute link</a>
```

< a href — stands for *hyperlink reference*

# RELATIVE Links

direct to a file on the same site /server

re: Unix!!

if the file is in the same folder:

```
<a href="index.html">Homepage</a>
```

if the file is in the parent folder:

```
<a href="../index.html">Homepage</a>
```

if the file is in the child folder:

```
<a href="images/photos.html">Photos</a>
```

id attribute: [Jump to a different element on page](#thisID)

```
<li><a href="#theFoot">id attribute link</a></li>
```

# RELATIVE Links

**direct to a file on the same site /server**

It's faster to simple direct to the file path.

id attribute: <a href="#thisID">Jump to a different element on page</a>

```
<li><a href="#">#theFoot">id attribute link</a></li>
```

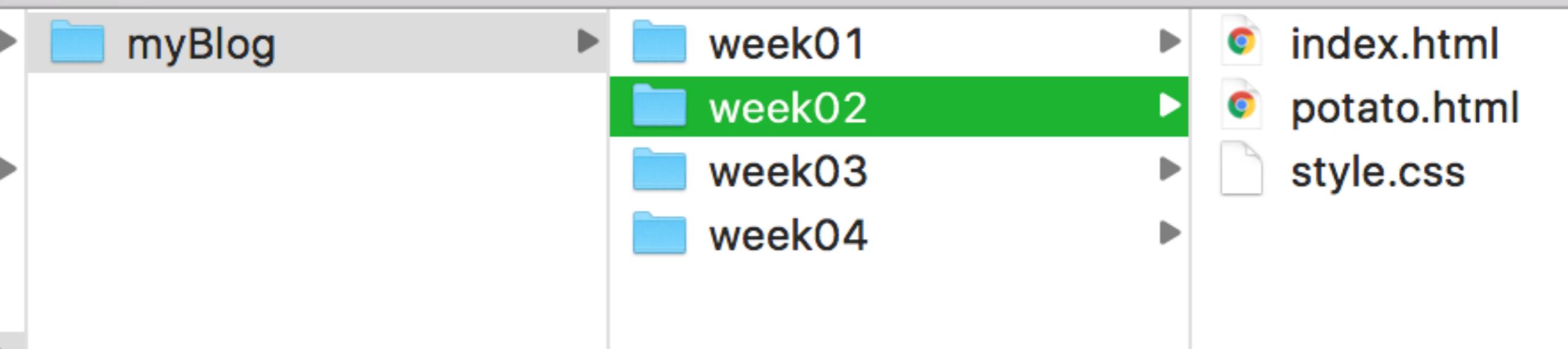
# Why index.html?

# Why index.html?

The main homepage of a site written in HTML (and the homepage of each section in a child folder) is called index.html.

Web servers are usually set up to return the index.html file if no file name is specified. Therefore, it's always a good idea to name your "home" page index.html

# Why index.html?



The **<img>** tag has a required attribute called **src**.

The **src** attribute must be set to the image's source, or the location of the image. In some cases, the value of **src** must be the *uniform resource locator* (URL) of the image. A URL is the web address or local address where a file is stored.

# Images: relative vs. absolute url

```
<img src= "images/potato07.png" alt= "spud" >
```

```
<img src= "https://pngriver.com/wp-content/uploads/2018/04/Download-Potato-PNG-Pic.png" alt= "spud" >
```

The **<img>** tag is for images, which can be on your local directory or on another webpage.  
Read all about **<img>** tag [here](#). The same goes for **<video>** + **<audio>** tags

The **alt** attribute, which means alternative text, brings meaning to the images on our sites. The **alt** attribute can be added to the image tag just like the **src** attribute. The value of **alt** should be a description of the image.

```

```

1. If an image fails to load on a web page, a user can mouse over the area originally intended for the image and read a brief description of the image. This is made possible by the description you provide in the **alt** attribute.
2. Visually impaired users often browse the web with the aid of screen reading software. When you include the **alt** attribute, the screen reading software can read the image's description out loud to the visually impaired user.
3. The **alt** attribute also plays a role in Search Engine Optimization (SEO), because search engines cannot "see" the images on websites as they crawl the internet. Having descriptive **alt** attributes can improve the ranking of your site.

Like the `<img>` tag, the `<video>` tag requires a `src` attribute with a link to the video source.

Unlike the `<img>` tag however, the `<video>` element requires an opening and a closing tag.

# <video /> structure

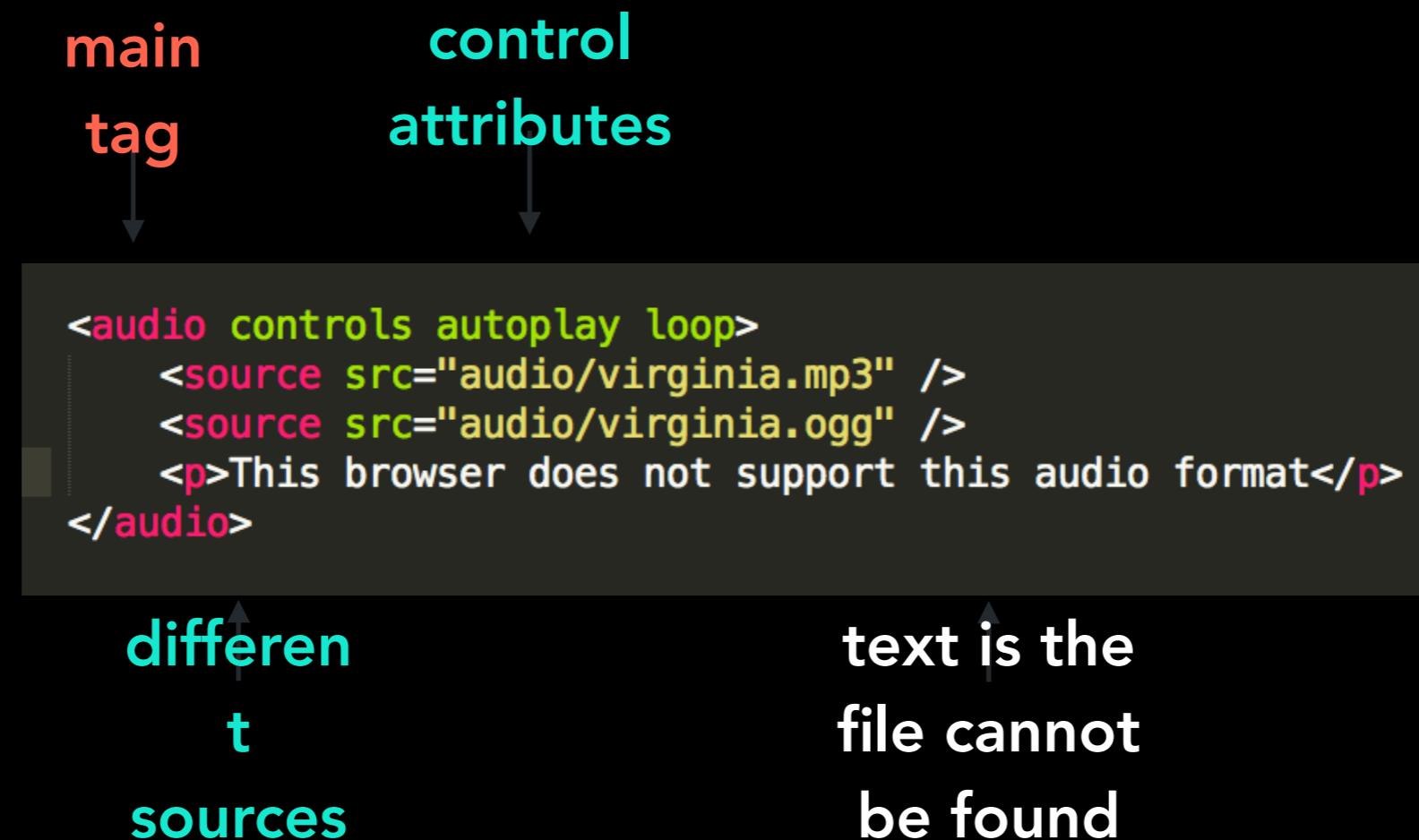


After the **src** attribute, the **width** and **height** attributes are used to set the size of the video displayed in the browser.

The **controls** attribute instructs the browser to include basic video controls: pause, play and skip. Unlike the **<img>** tag however, the **<video>** element requires an opening and a closing tag.

The text, "Video not supported", between the opening and closing video tags will only be displayed if the browser is unable to load the video.

# <audio /> structure



# Some Media Attributes

**Preload** - what preloads when the page loads

**Controls** - if the play/stop buttons are visible

**Autoplay** - if the video should start playing  
automatically

**Loop** - if the video should loop on completion

# Attributes

If we want to expand an element's tag, we can do so using an attribute. Attributes are content added to the opening tag of an element and can be used in several different ways, from providing information to changing styling. Attributes are made up of the following two parts:

- 1) The **name** of the attribute
- 2) The **value** of the attribute

One commonly used attribute is the **id**.

We can use the **id** attribute to specify different content (such as **<div>**s) and is really helpful when you use an element more than once.

```
<div id="intro">  
  <h1>Technology</h1>  
  </div>
```

**<span>** contains short pieces of text or other HTML. They are used to separate small pieces of content that are on the same line as other content.

```
<div>
    <h1>Technology</h1>
</div>
<div>
    <p> Wherever there's a
    <span>computer</span>, there's a skilled
    person developing, maintaining, hacking,
    advancing or simply using it.</p>
</div>
```

## <Text> input

Username:

## Password input

Username: Ruta

Password  .....

## Text area

What is your favorite movie to watch?

What is your favorite movie to watch?

## Checkbox

Select your favorite input type:

Radio  Checkbox  Text

## Drop down list

Select your favorite input type:

- Radio
- Checkbox
- Text

## Multiple select box

Select your favorite input type:

- Radio
- Checkbox
- Text

## Submit button

Are you ready to make that selection?

SUBMIT

**What are Ways of Seeing?**



Walter Benjamin  
*Art in the Age of Mechanical Reproduction*, 1935



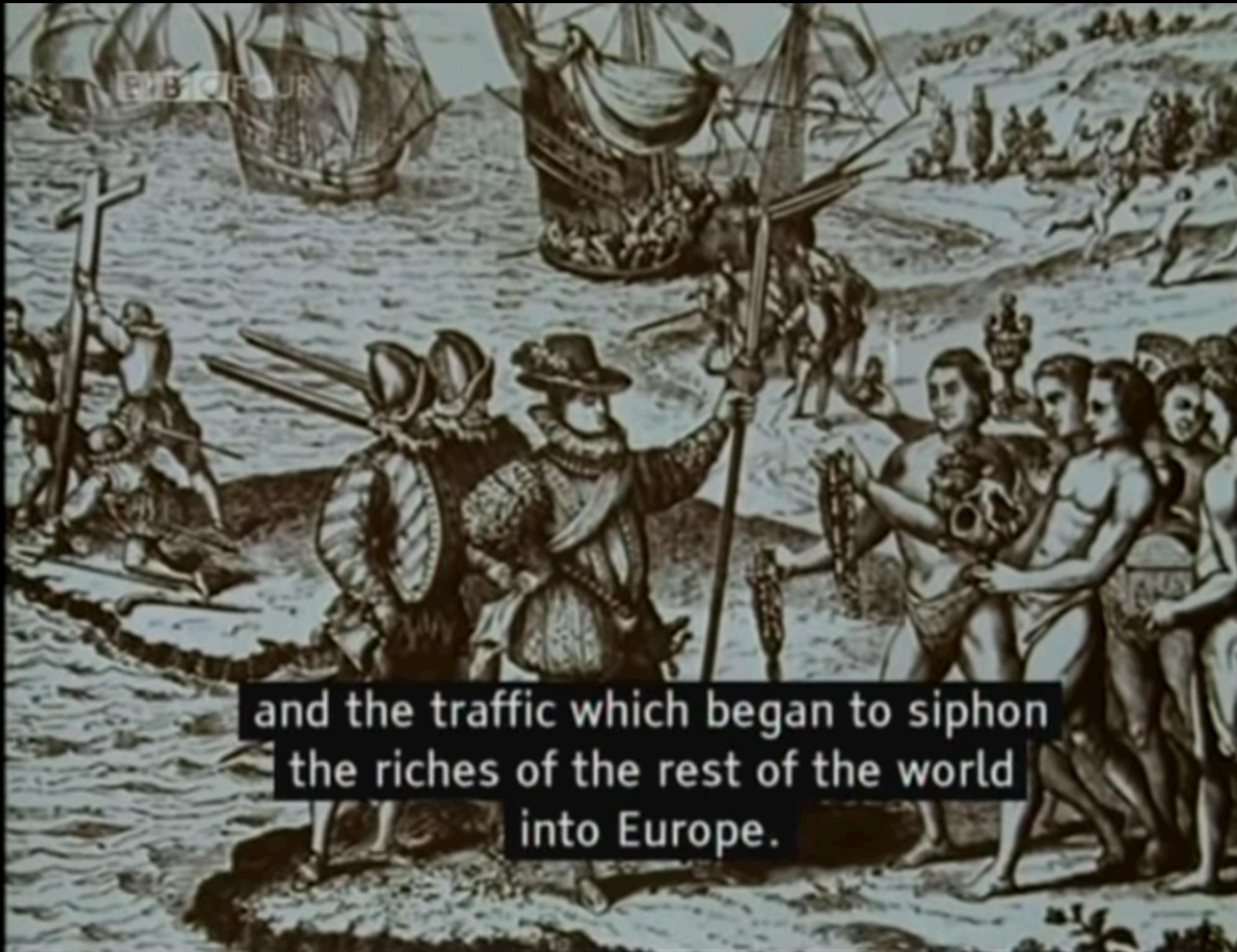
A portrait of John Berger, an elderly man with dark, wavy hair, looking slightly to his right. He is wearing a light-colored shirt with a pattern of small, stylized figures. The background is a clear blue sky.

you receive images and meanings  
which are arranged.

*Ways of Seeing* - John Berger, 1972



txt + 



and the traffic which began to siphon  
the riches of the rest of the world  
into Europe.

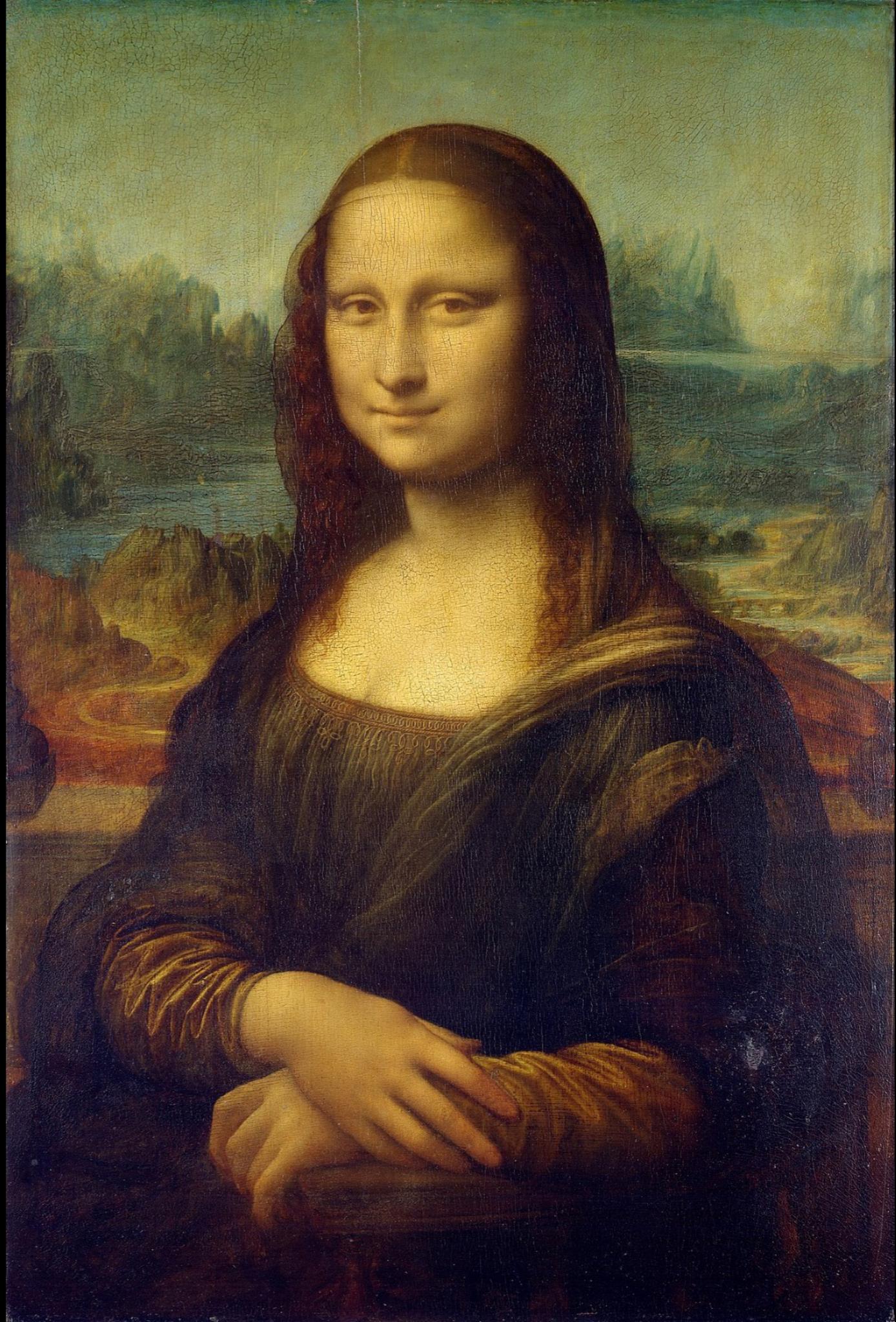
BBC FOUR

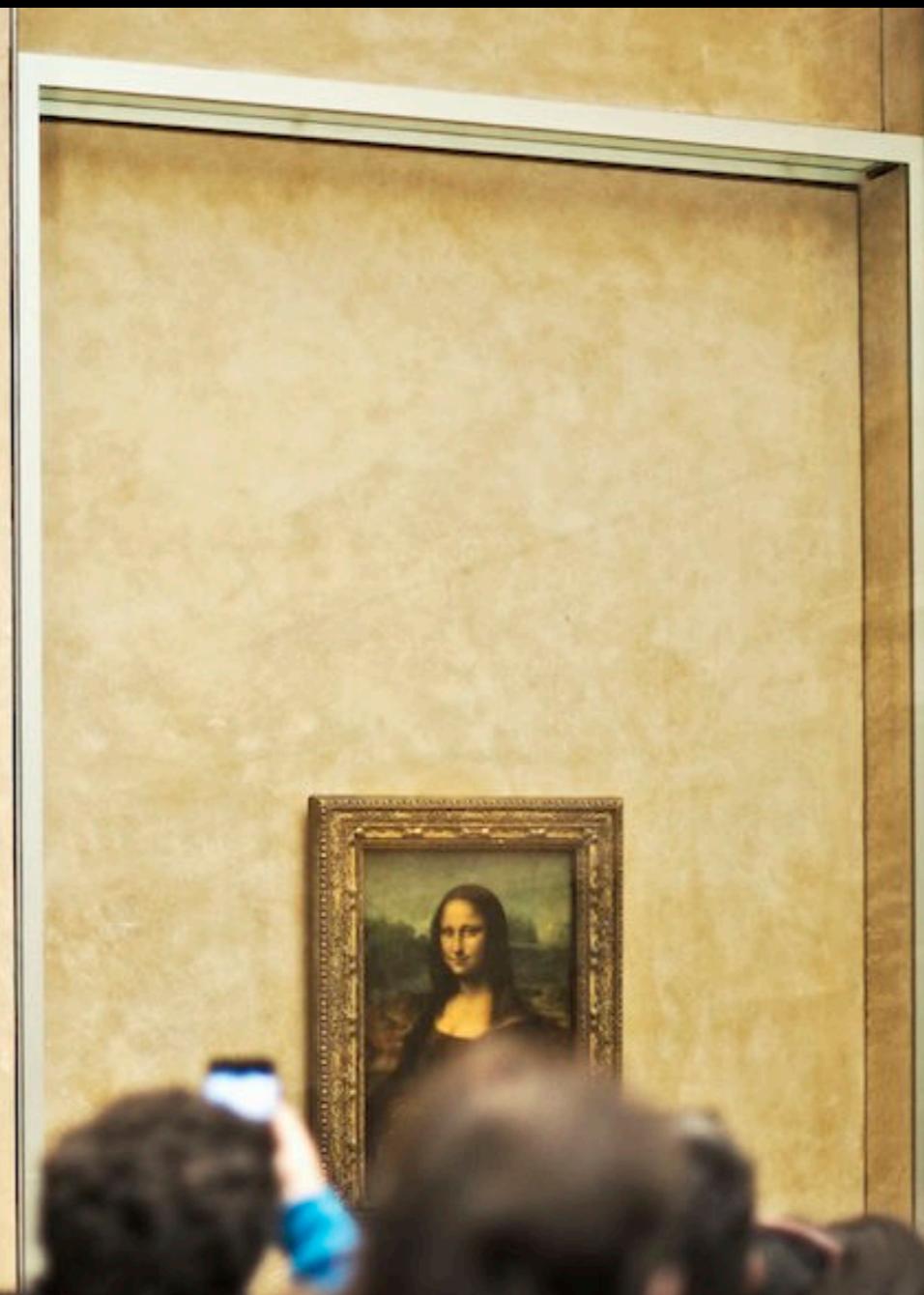
ghts  
ovement.

good kid

We take them away in our minds.

Ways of Seeing







beyonce • Follow

...



rihanna.always A bih reached there



96w 1 like Reply



pookymovesmusic 61'



95w Reply



raygoomes @manu\_henrique ela já estava planejando tudo



91w 3 likes Reply

— View replies (1)



adrianngaitan hi



832,538 likes

OCTOBER 11, 2014



Add a comment...

Post





1 inespgoncalves, cjey97, therunwayboyz and T others like this.

-  **aamuchick**  
Sweet Respect
-  **aamuchick**  
The Soft Smile of Happiness
-  **gaalllliiii**  
|||||
-  **roc4life3**  
45 LUV THE FAMILY VACATION PICS! KEEP THE COMING! ROCNATION4LIFE3!
-  **tmamie11**  
What is wrong with you too you guess are crazy
-  **parmanmarcus**  
Hov looking fresh
-  **jotieno**  
@sammialyse



Leave a comment...

PRICE \$8.99

# THE NEW YORKER

AUG. 29, 2022



the Box Model

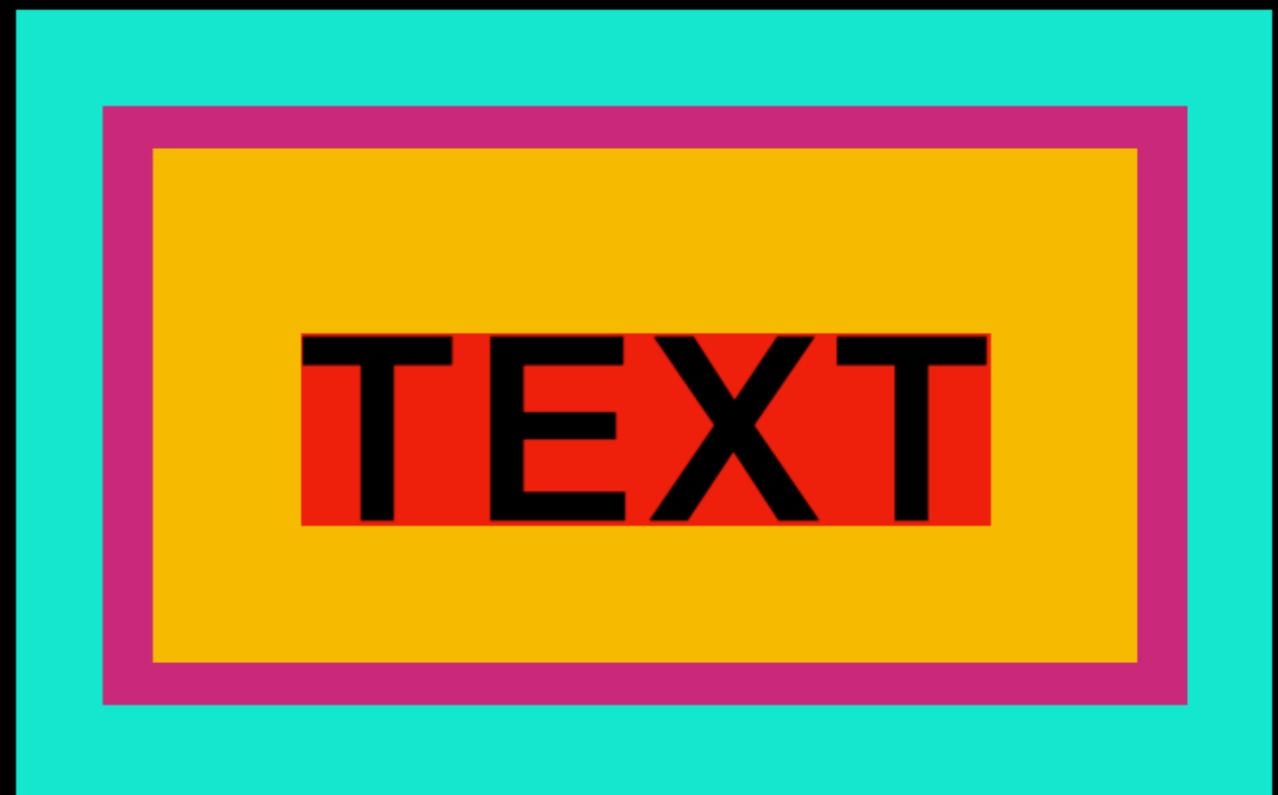
## Border

All boxes have borders even if invisible or 0px wide. It separates the edge of one box from another.

# the Box Model

## Padding

Padding is the space btw the border + any content contained within it. More padding increases the readability of its contents.



## Margin

Margins sit outside the edge of the border. You can set the width to create a gap btw borders of adjacent boxes.

## Content