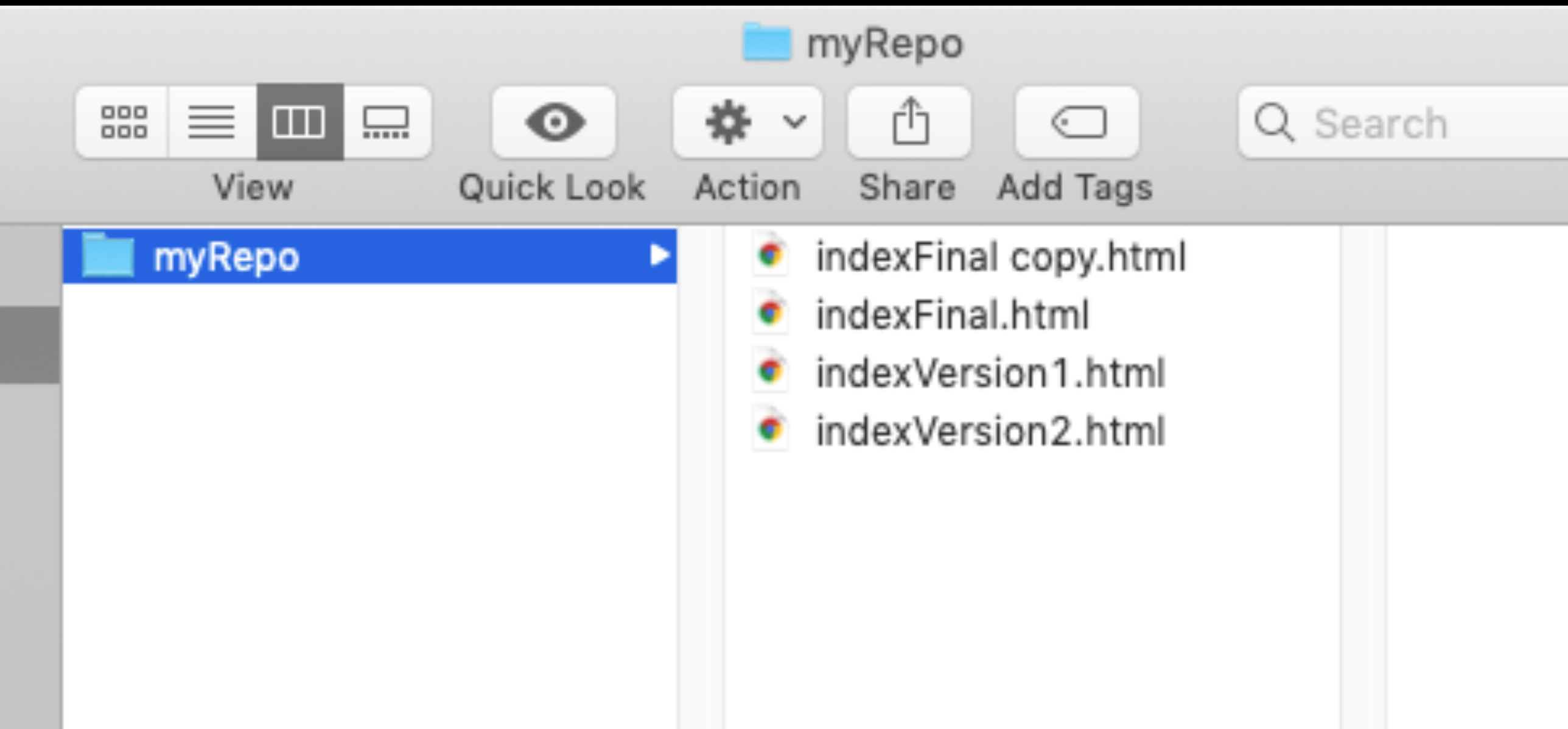
Git

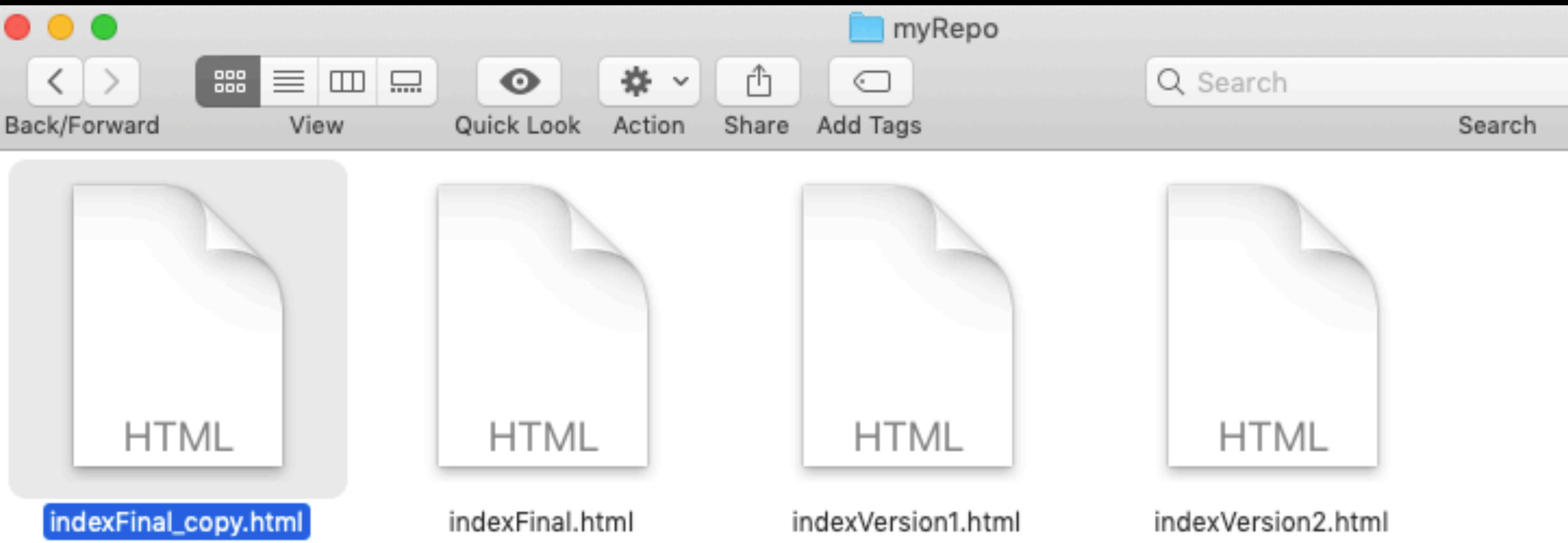
What is Git?

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 contxts. 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

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

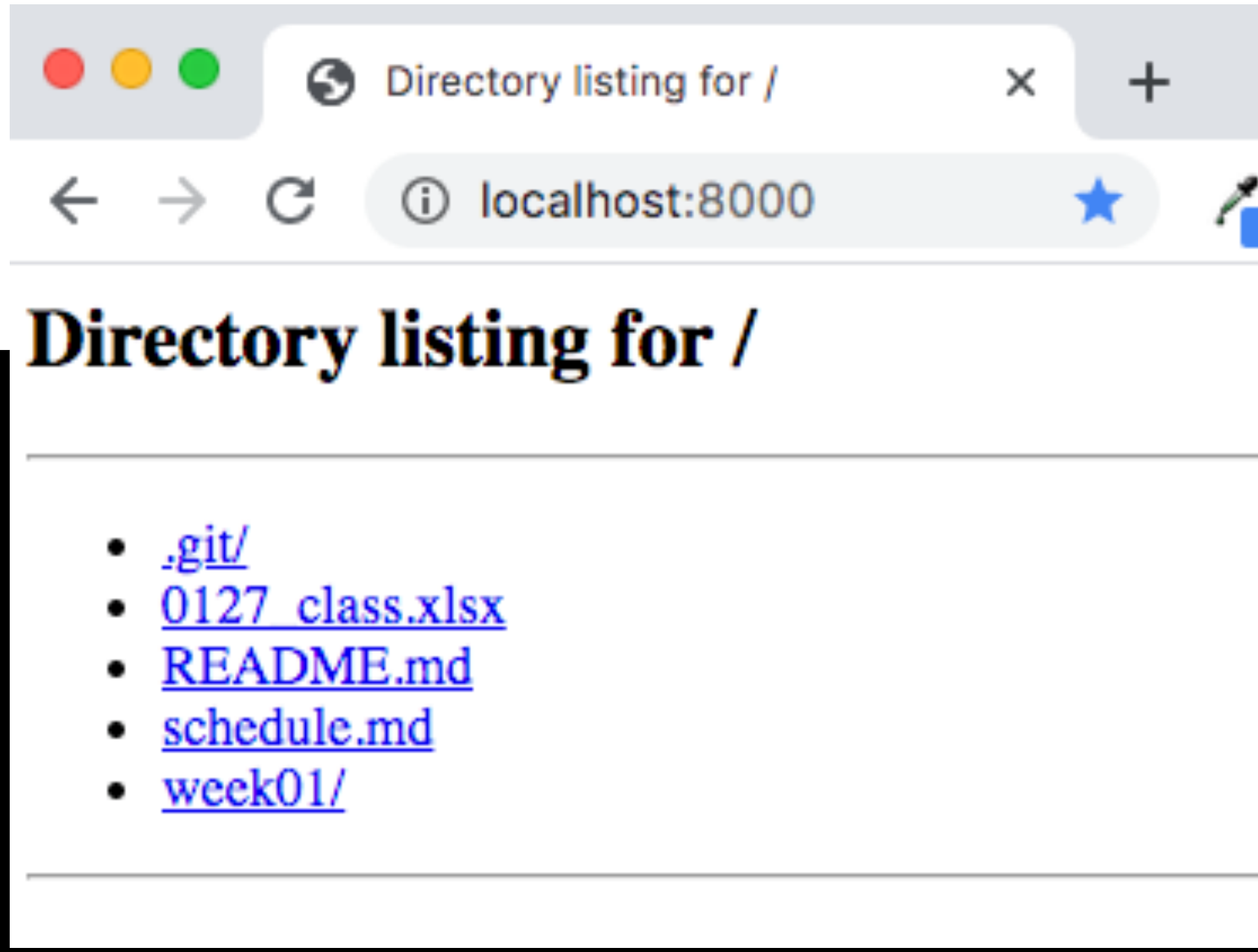
prototype: local http server
(using our local machine as a server!!)

publish: pushed to Github Pages

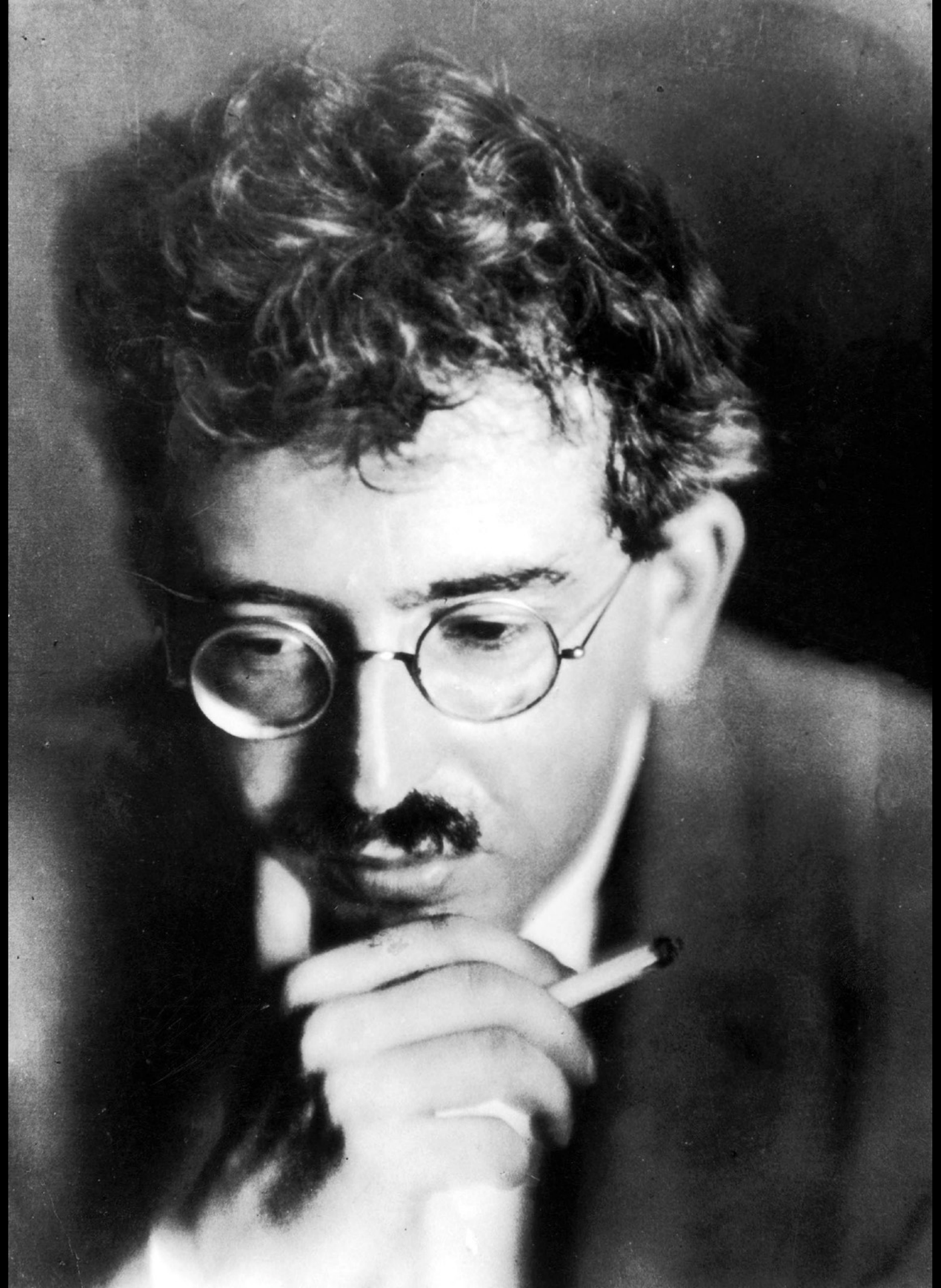
webDevSpring2020 — (. . .) — 100×40

~/Desktop/teach/nyu/2020/webDevSpring2020 — (. . .) — python -m SimpleHTTPServer

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



What are Ways of Seeing?



Walter Benjamin
Art in the Age of Mechanical Reproduction, 1935

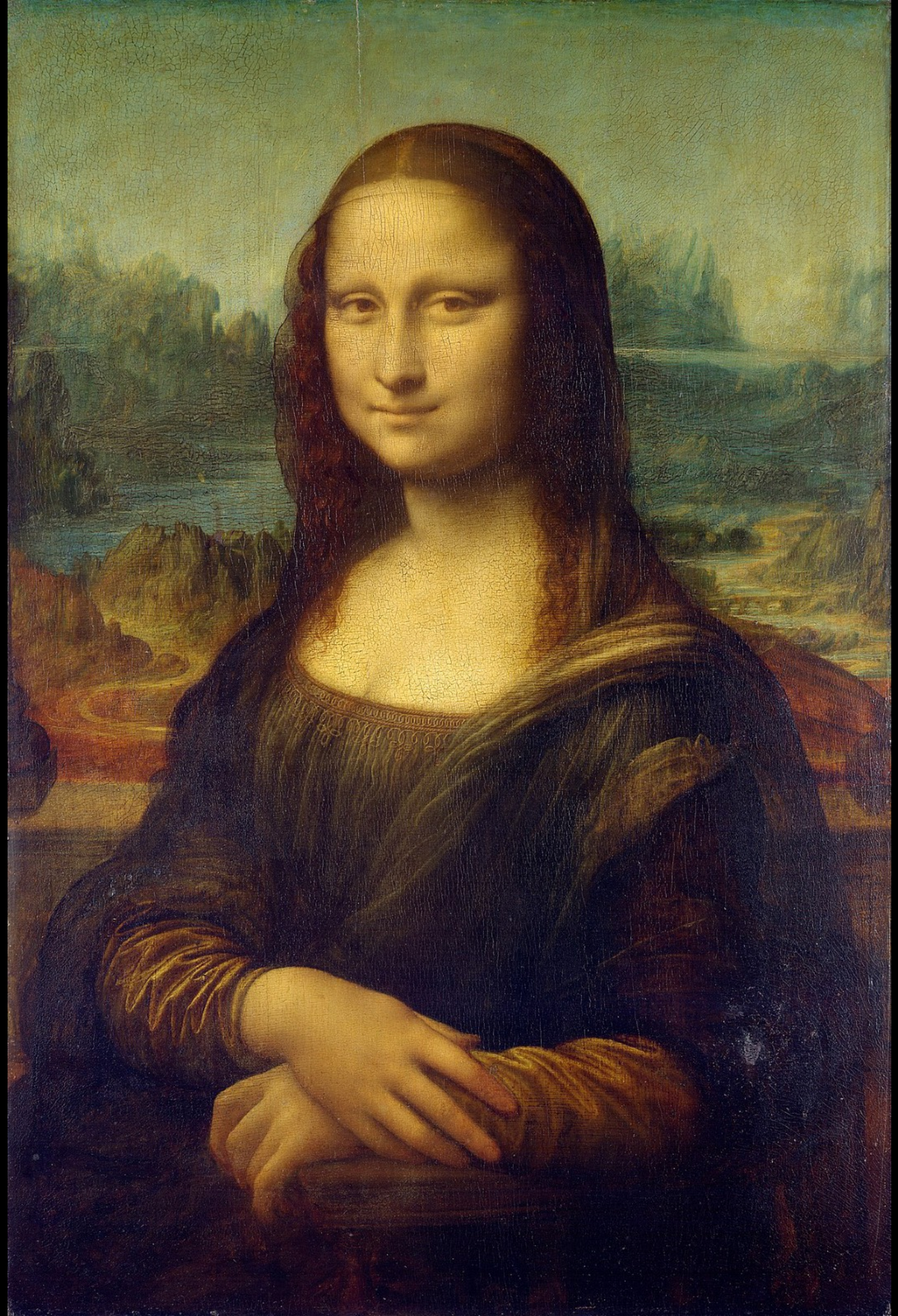
Ways of Seeing - John Berger, 1972

power, representation

txt + 

Ways of Seeing

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

"

inespgoncalves, cjey97, therunwayboyz and 1 others like this.

aamuchick
Sweet Respect

aamuchick
The Soft Smile of Happiness

gaaliiiii
🙌🙌🙌

roc4life3
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

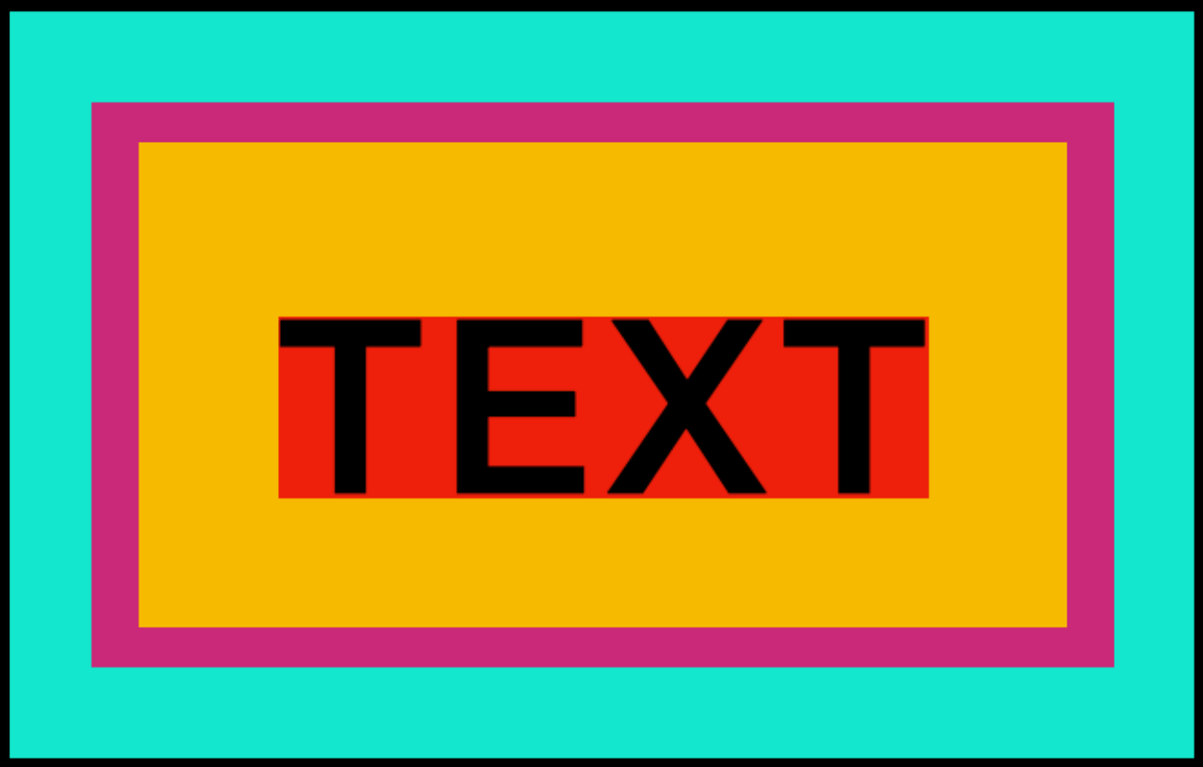
the Box Model

Border

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

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

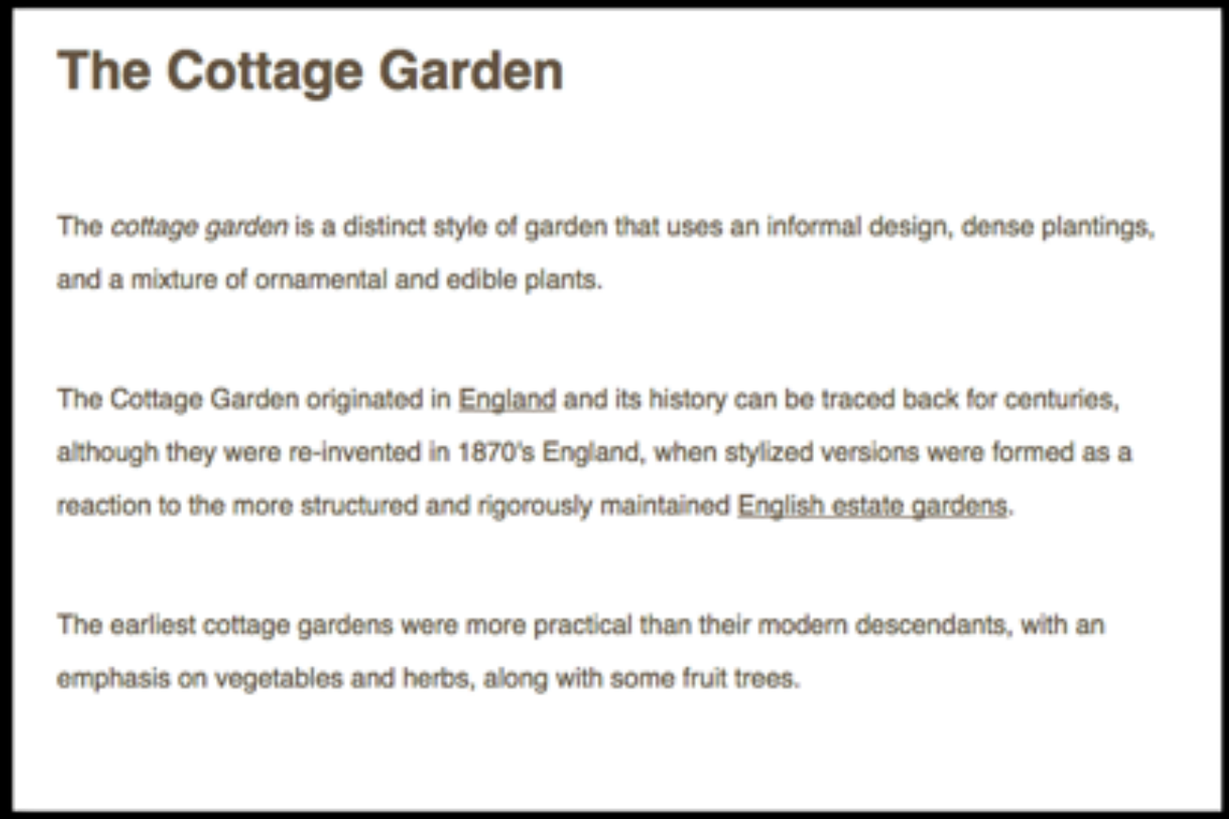
HTML - Hyper Text Mark Up

is a grammar for structuring web pages. It defines paragraphs, headings, data tables + media elements. HTML describes the content of the page - not how it looks.

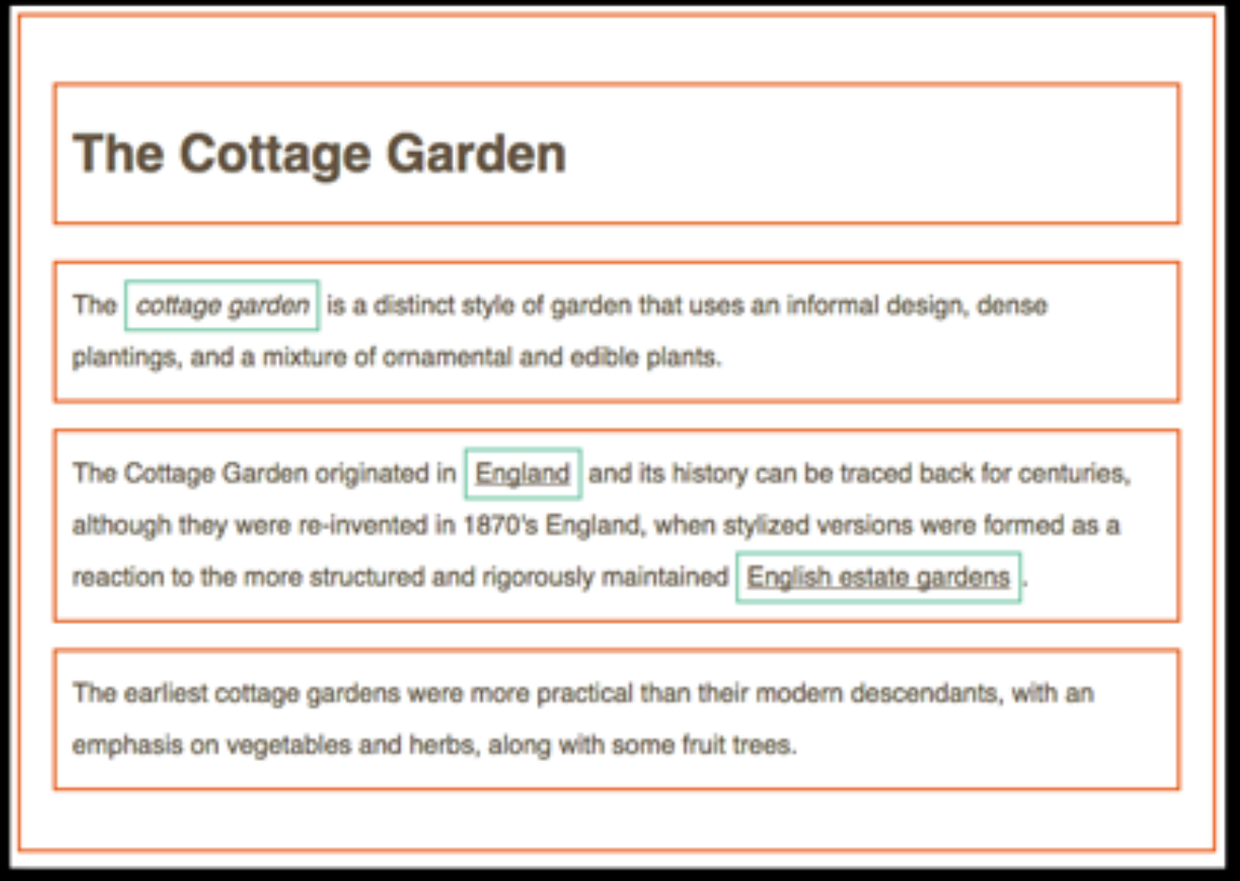
CSS - Cascading Style Sheet

rules for styling a web page. Setting colors, typeface, and the layout. It can be used to consider the design of your **page across different platforms and screen sizes.**

The key to understanding how **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.

Quick Getting Ahead of Ourselves: responsive web design

Metadata: `viewport`

The user's visible area of a web page

HTML5 introduced a method to let web designers take control over the viewport, through the **<meta>** tag.

```
<!
—  - Tells the browser to match the device's width for the viewport
     - Sets an initial zoom value -->

<meta name="viewport" content="width=device-width, initial-scale=1.0”>
```

<meta name="viewport" content="width=device-width, initial-scale=1.0”>

without

with

Inline Styles

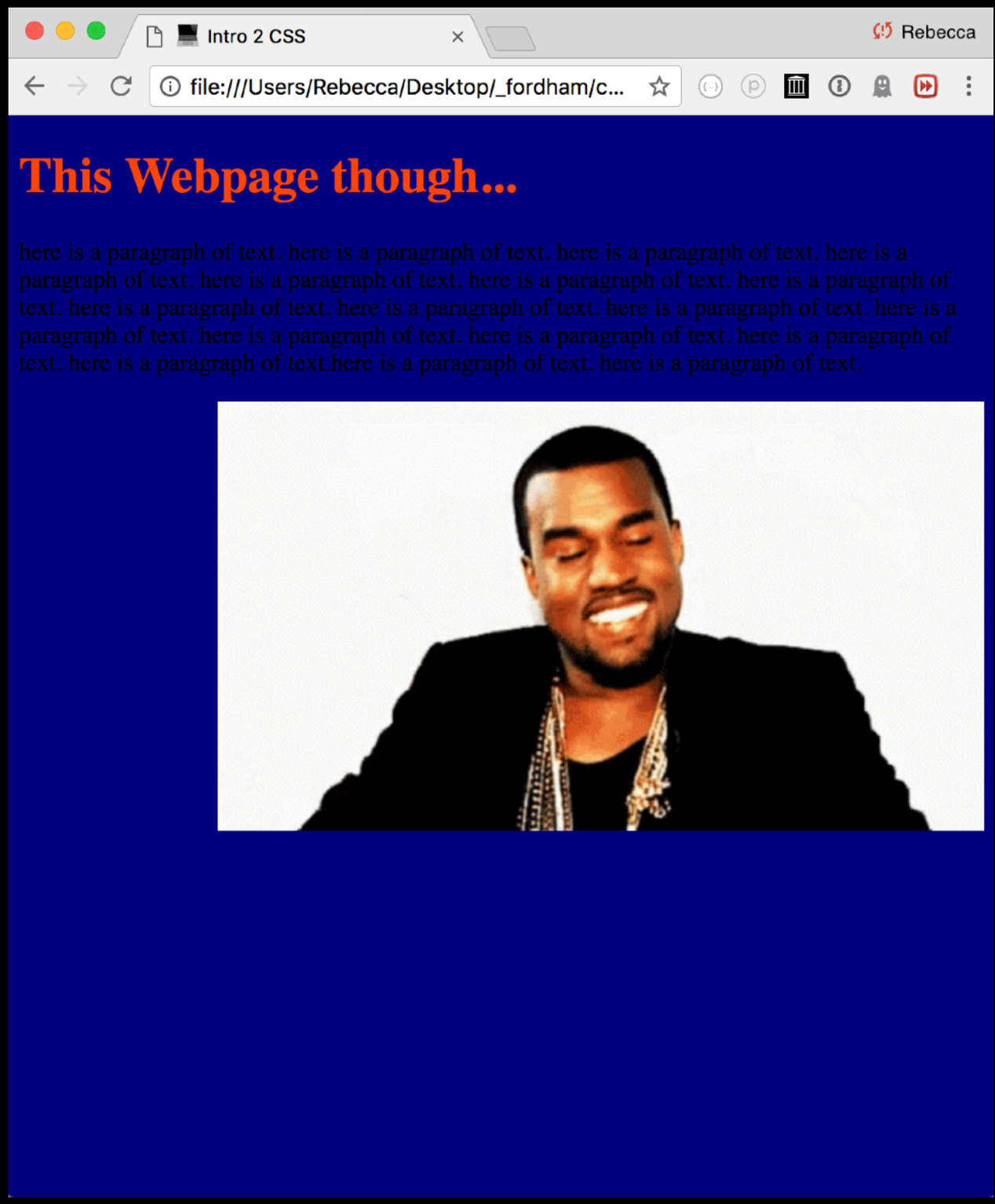
```
<h1 style="color:#FF4500;">This Webpage though...</h1>
<body style="background-color: #000080;">
```

Embedded Styles

```
<html>
  <head>
        <title> 💻 Intro 2 CSS </title>
        <style type- "text/css">
            h1 {
                color: #FF4500
            }

            body {
                background: #000080;
            }

        </style>
    </head>
```

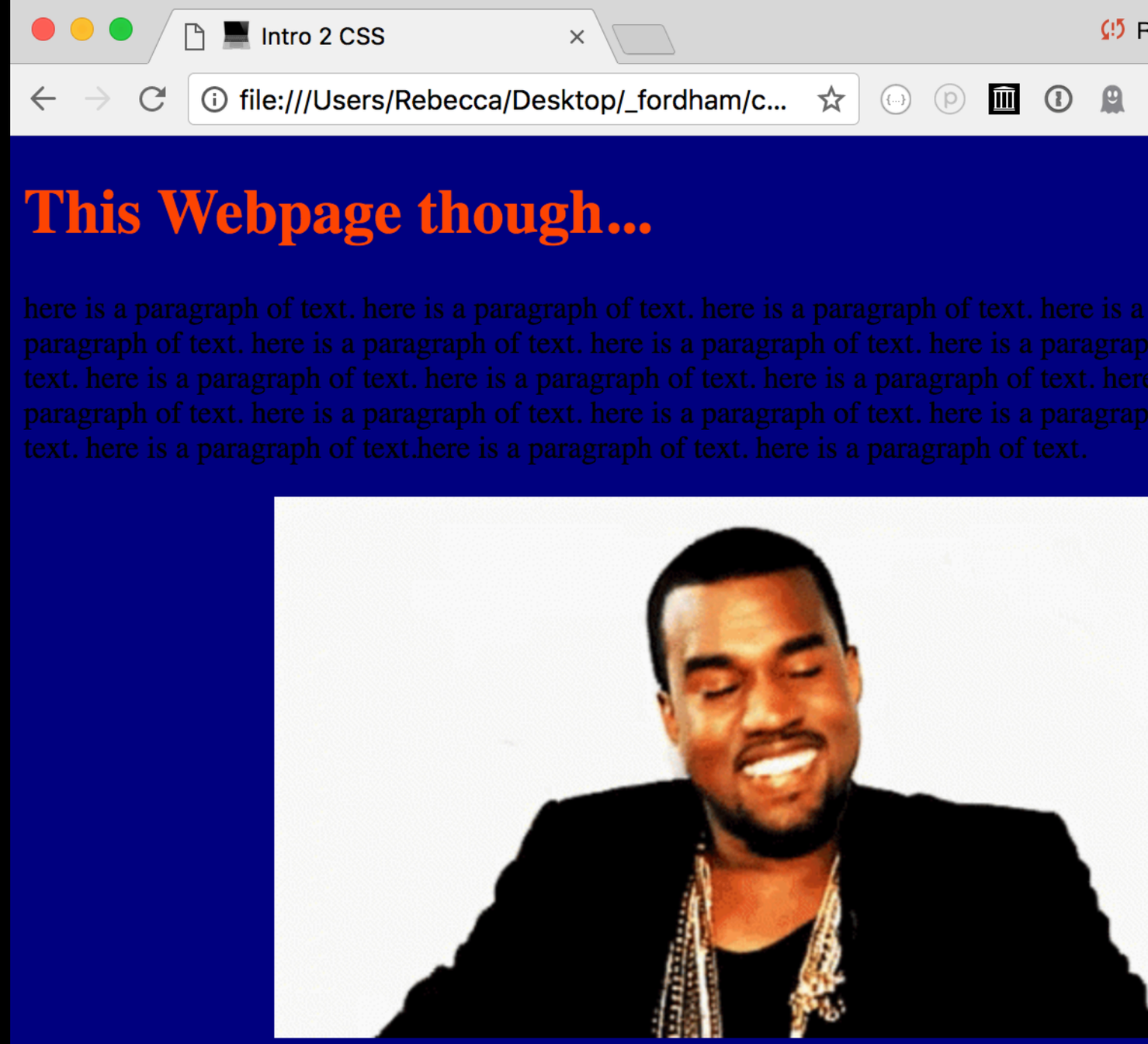
External Styles *

```
<head>
    <title> 💻 Intro 2 CSS </title>
  <link rel="stylesheet" type="text/css" href="theStyle.css">
 </head>
<body>

</body>
```



```
h1 {
    color: #FF4500
}

body {
    background-color: #000080;
}
```

```
selector {

  property: value ;

}
```



Selector is a term such as p, h1, div that identifies the HTML element you want to format or apply a rule to. You can add multiple selectors in a declaration.

Selector	Meaning	Example
Universal Selector	Applies to all elements in the document	* { }
Type Selector	Matches element names	h1, h2, h3 { }
Class Selector	Matches an element whose class attribute has a value that matches the one specified after the period (or full stop) symbol	.theNote { } targets any element whose class attribute has a value of "note}
		p.note { } targets only <p> elements whose class attribute has a value of "note"
ID Selector	Matches an element whose id attribute has a value that matches then specified after the # symbol	#introduction { } targets the element whose id attribute has value of "introduction"

Selector	Meaning	Example
Child Selector	**Matches an element that is a direct child of another**	li > a { } targets any <a> element that are children of an element (but not other <a> elements in the page.
Descendant Selector	**Matches an element that is a descendent of another specified element (not just a direct child of that element)**	p a { } targets any <a> elements that sit inside a <p> element, even if there are other elements nested btw them

Selector	Meaning	Example
Adjacent Sibling Selector	**Matches an element that is the next sibling of another**	**h1+p** { } targets the first **<p>** element after any **<h1>** element (but not other **<p>** elements)
General Sibling Selector	**Matches an element that is a sibling of another, although it does not have to be the directly preceding element**	**h1~p** { } tif you have two **<p>** elements that are siblings of an **<h1>** element, this rule would apply to both

```css
/* type/element selector */
p {
    color: blue;
    font-size: 50vh;
}

/* class attribute selector */
.myBlueText {
    color: blue;
}

/* id attribute selector */
#blue-par {
    color: blue;
}

/* BONUS: grouping
selector */
p,
.blue-text,
#blue-par {
    color: blue;
}
```

selecting multiple elements:

```
h1, h2, h3 {

    color: red;
    background-color: blue;
    width: 500px;

}
```

```
p,
li {

    background-color: red;
    font-color: blue;


}
```

HTML comments are written like this

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

CSS comments are written like this

```
/* This is a comment */
```

```
 {
text-align:

            left ;
            right ;
            center ;
            justify ;

}
```

```
{
vertical-align:

        baseline ;
        sub ;
        super ;
        top ;
        text-top ;
        middle ;
        bottom ;
        text-bottom ;


}
```

This property is NOT intended to allow you to vertically align text in the middle of a block level elements such as <p> + , although it does have this effect when used with table cells <td> + <th> elements.

It is more commonly used w/ inline elements such as <img>, or . When used with these elements, it performs a task very similar to the HTML align attribute used on the <img> element.

Interaction Design

```
a: link {

a: visited {

    : hover {

    : active {

    : focus {

}
```

Applied when a user hovers over an element w/ a mouse. This changes the appearance of links and buttons when a user places their cursor over them. Does not work on mobile.

Applied when an element is bingo activated by a user, like when a button is pressed or a link clicked. This added to UX.

Applied when an element has focus. Any thing you can interact with.

Focus occurs when a browser discovers that you are ready to interact w/ an element. For example when yr cursor is in an input - that element is said mohave focus.

Classes and IDs

Two common attributes used to single out certain HTML elements are **class** and **id**, both are used to identify particular elements when adding CSS styling rules.**You author class + id names!!** They have no particular meaning in themselves, besides a puzzle - or code - you are creating.

Use a **class** when you have more than one element you want to share the same styling - perhaps across multiple pages.

Use an **id** when there is only one element on the page with that id, for example id=”header”
With a class you can have as many elements with that styling as you like.

An element can have more than one **class**, but not more than one **id**. When there is more than one class, the class names are separated by spaces.

```
<h1 id="myHeader">Hello World!</h1>
```

IDs

Every HTML element can carry the id attribute. It is used to uniquely identify that element from other elements on the page.

Its value should start with a letter or an underscore (not a number or any other character). It is important that no two elements on the same page have the same value for their id attributes (otherwise the value is no longer unique).

More to read on ID naming: https://mathiasbynens.be/notes/css-escapes

IDs

To select these IDs in CSS you would do so with **#myHeader** syntax

(IDs may become particularly useful when it comes to media elements - photos, videos + sound files.)

```
#myHeader{
    color: blue;
}
```

Classes

```
<div class="theAuthor">
  -- from John Duckett's <span><a
  href="https://www.amazon.com/Web-Design-HTML-JavaScript-jQuery/dp/1118907442
  /ref=sr_1_3?ie=UTF8&qid=1526310943&sr=8-3&keywords=html+and+css"
  target="_blank">HTML + CSS</span></a>
  <br>
</div>
```

To select these classes in CSS you would do so with .theAuthor syntax

```
.theAuthor{
  background: rgb(255,255,255);
    /* HSL: Hue, Saturation + Lightness
    Hue - as an angle between 0 + 360
    Saturation - as a precentage
    Lightness - as a precentage: 0% = white, %50 = normal + 100% is bl
    Alpha - expressed btw 0 _ 1.0 : 0.5 = 50% transparency, .75 is 75%
    transparency*/
  background: hsl(0,100%,100%, 0.2);
  text-align: center;
}
```