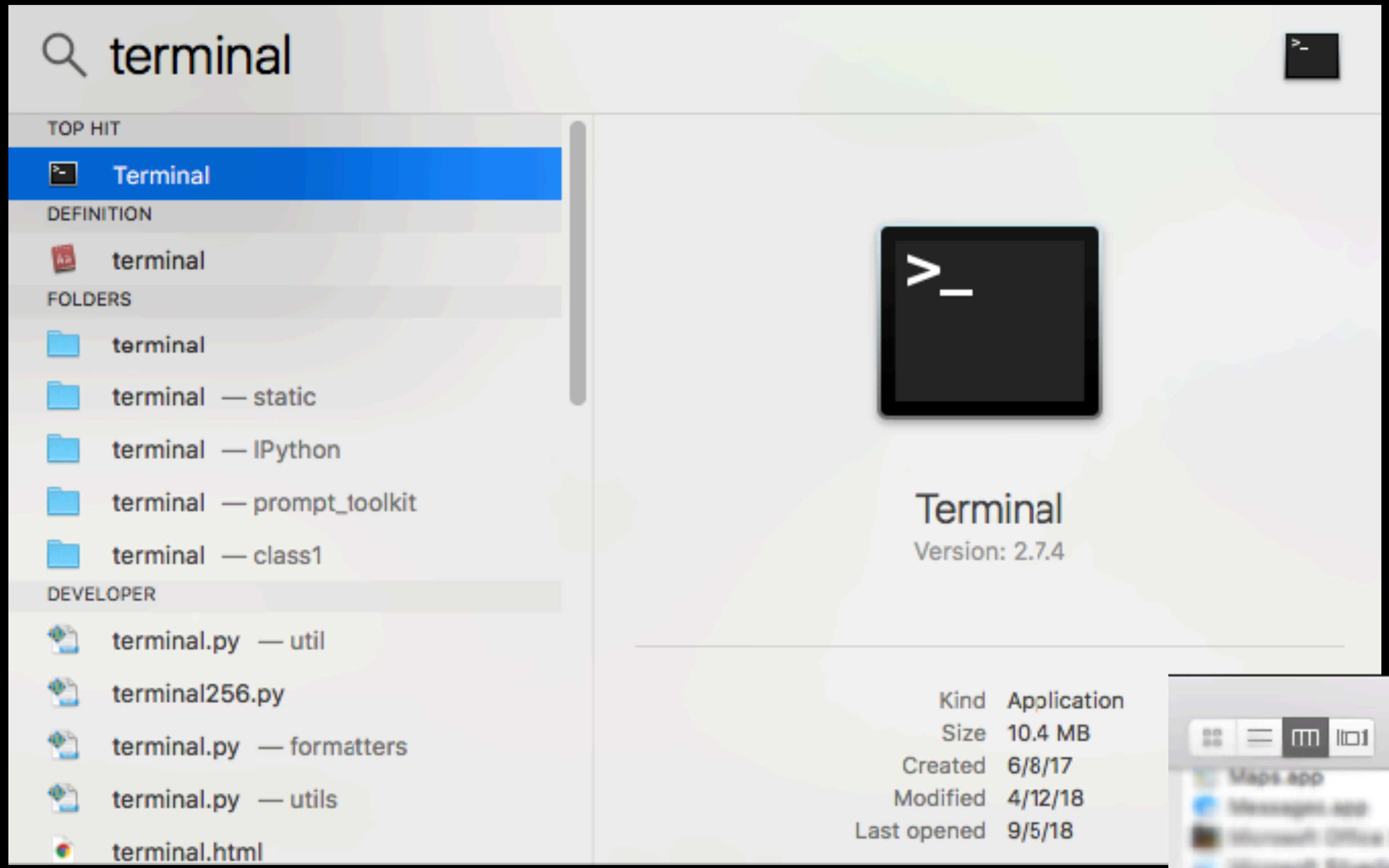
prototyping yr webpage

for those who like to search:

Hold down COMMAND + press SPACEBAR, and type TERMINAL



terminal

TOP HIT

Terminal

DEFINITION

terminal

FOLDERS

terminal

terminal — static

terminal — IPython

terminal — prompt_toolkit

terminal — class1

DEVELOPER

terminal.py — util

terminal256.py

terminal.py — formatters

terminal.py — utils

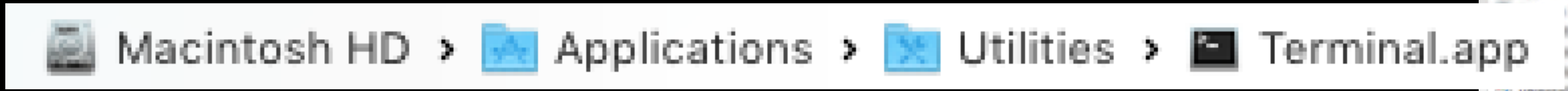
terminal.html

Terminal
Version: 2.7.4

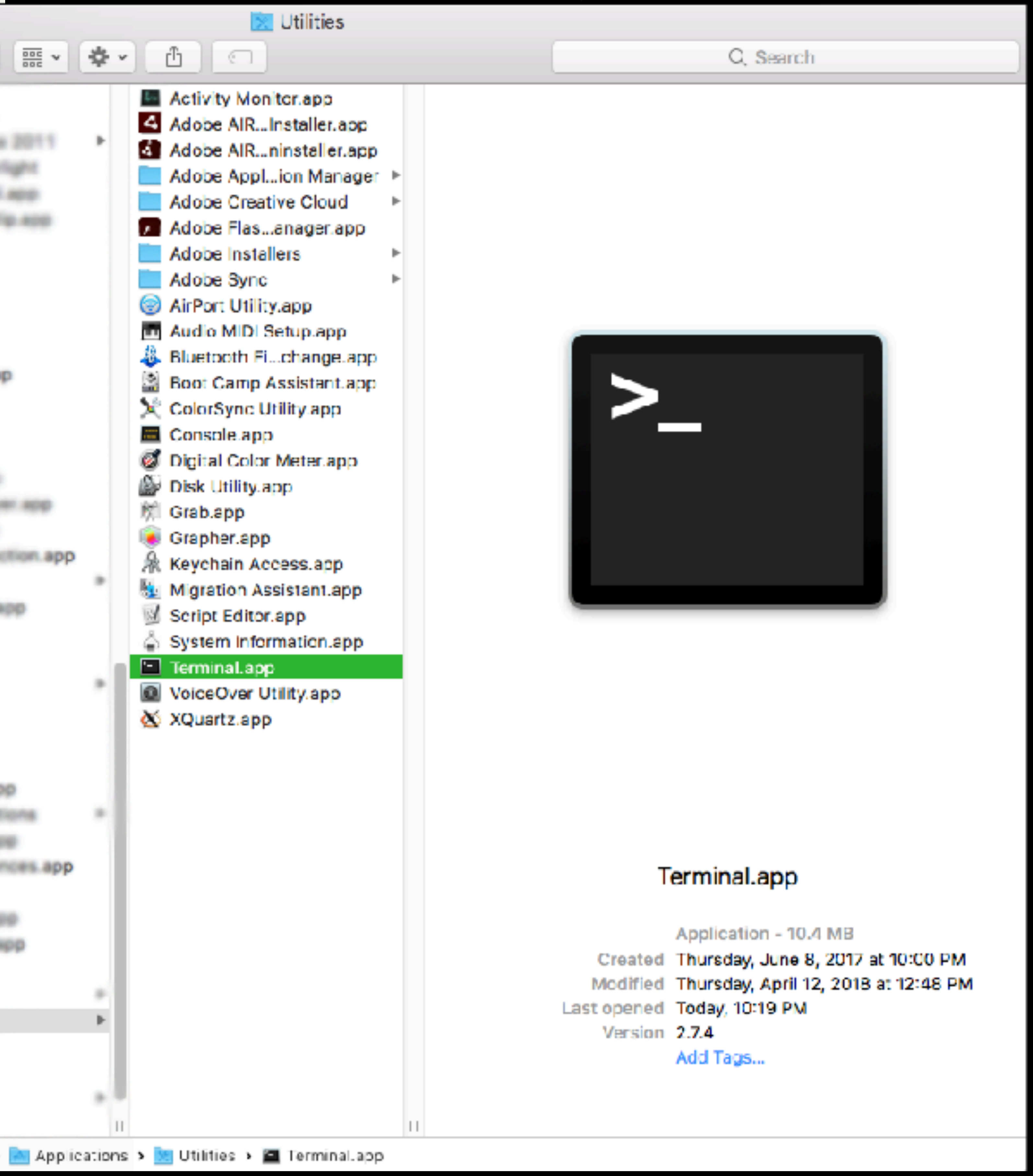
Kind	Application
Size	10.4 MB
Created	6/8/17
Modified	4/12/18
Last opened	9/5/18

for those who like to find:

the file path is:



Macintosh HD › Applications › Utilities › Terminal.app



Utilities

Activity Monitor.app
Adobe AIR...Installer.app
Adobe AIR...ninstaller.app
Adobe Appl...ion Manager
Adobe Creative Cloud
Adobe Flas...anager.app
Adobe Installers
Adobe Sync
AirPort Utility.app
Audio MIDI Setup.app
Bluetooth Fi...change.app
Boot Camp Assistant.app
ColorSync Utility.app
Console.app
Digital Color Meter.app
Disk Utility.app
Grab.app
Grapher.app
Keychain Access.app
Migration Assistant.app
Script Editor.app
System Information.app
Terminal.app
VoiceOver Utility.app
XQuartz.app

Terminal.app

Application - 10.4 MB

Created	Thursday, June 8, 2017 at 10:00 PM
Modified	Thursday, April 12, 2018 at 12:46 PM
Last opened	Today, 10:19 PM
Version	2.7.4

Add Tags...

Macintosh HD › Applications › Utilities › Terminal.app in Terminal we are speaking **Unix** :

- **cd** - “change directory”
- **ls** - “list items in this directory”
- **pwd** - “present working directory”

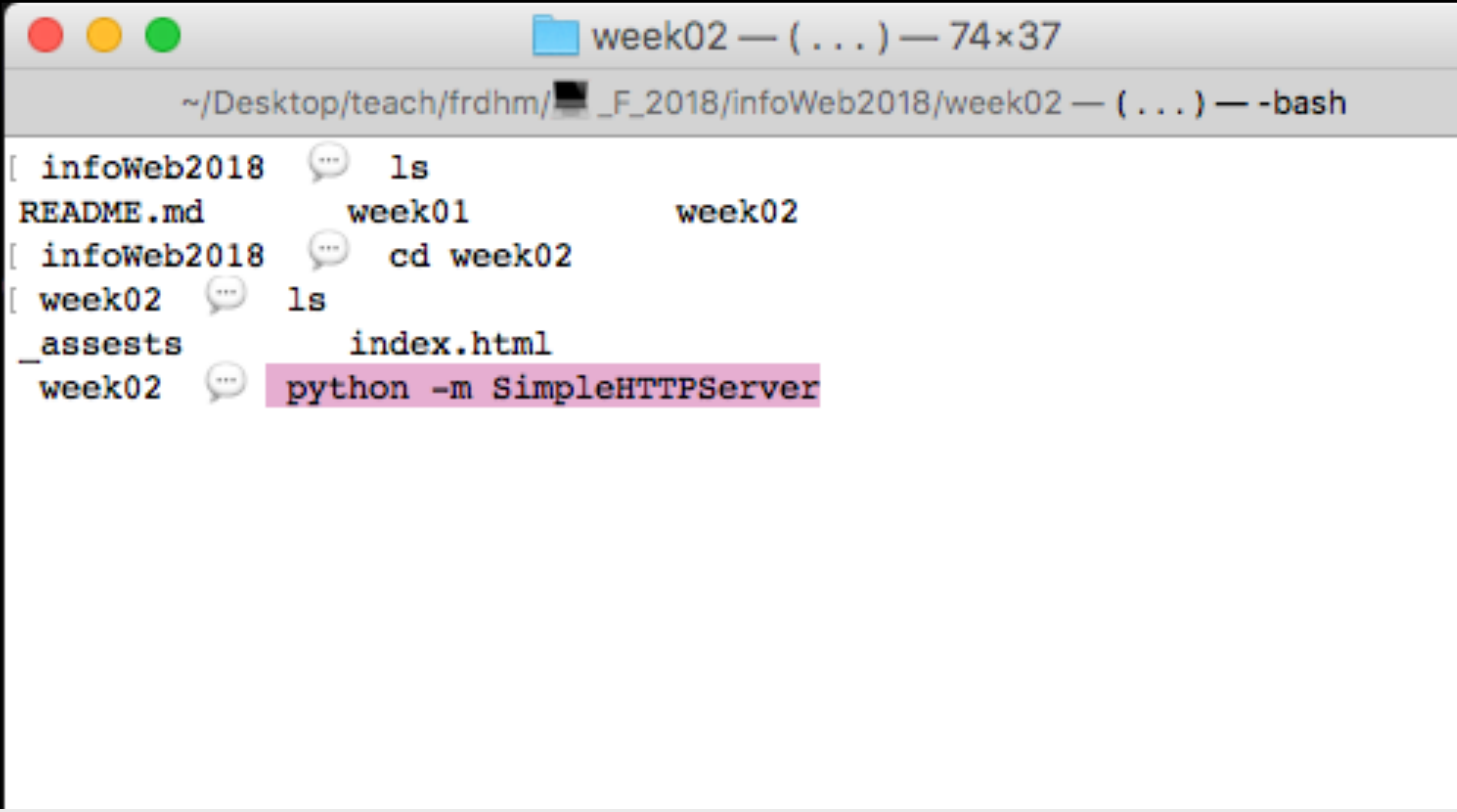
Running a local **Python** HTTP Server
in Mac OS - this is very simple :

When inside yr project folder simply type the following command:

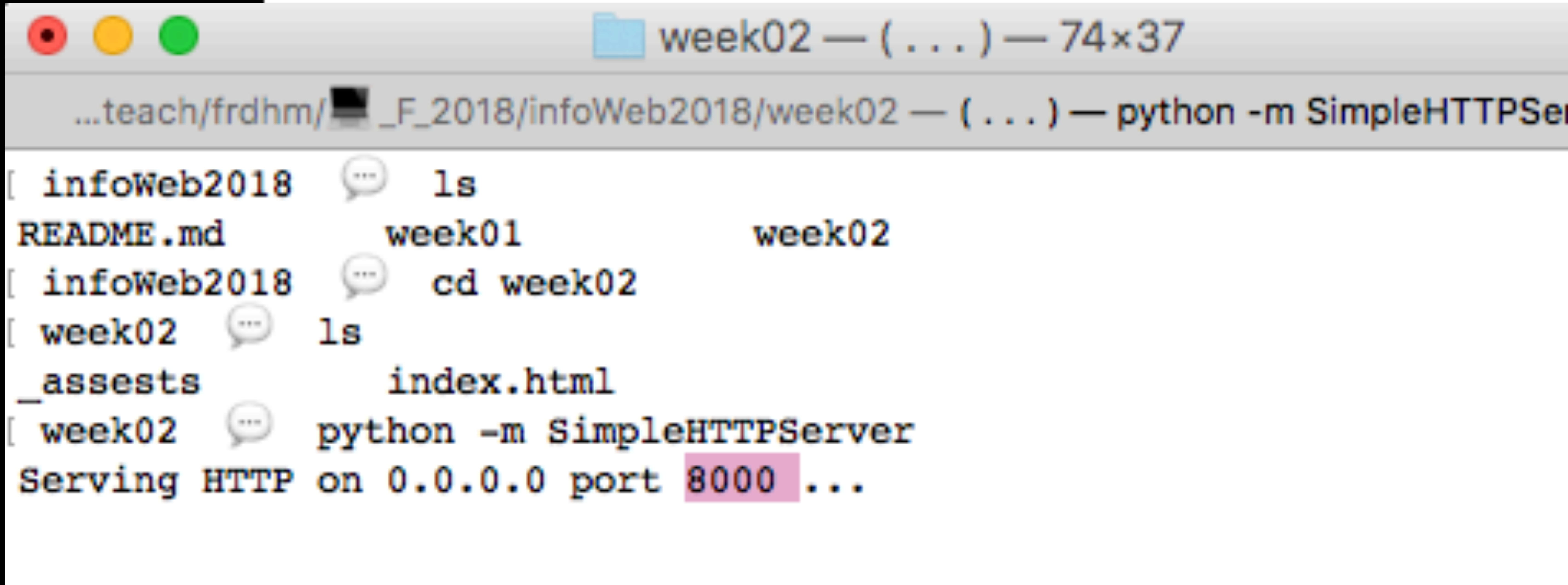
“**python -m SimpleHTTPServer**”

– defaults to port 8000

if we wrote:
“python -m SimpleHTTPServer 12345”
- we would go to port 12345


week02 — (. . .) — 74×37
~/Desktop/teach/frdhm/[redacted] _F_2018/infoWeb2018/week02 — (. . .) — -bash

```
[ infoWeb2018 💬 ls
README.md          week01          week02
[ infoWeb2018 💬 cd week02
[ week02 💬 ls
_assests           index.html
[ week02 💬 python -m SimpleHTTPServer
```


week02 — (. . .) — 74×37
...teach/frdhm/[redacted] _F_2018/infoWeb2018/week02 — (. . .) — python -m SimpleHTTPSe

```
[ infoWeb2018 💬 ls
README.md          week01          week02
[ infoWeb2018 💬 cd week02
[ week02 💬 ls
_assests           index.html
[ week02 💬 python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

for those working on a windows machine:

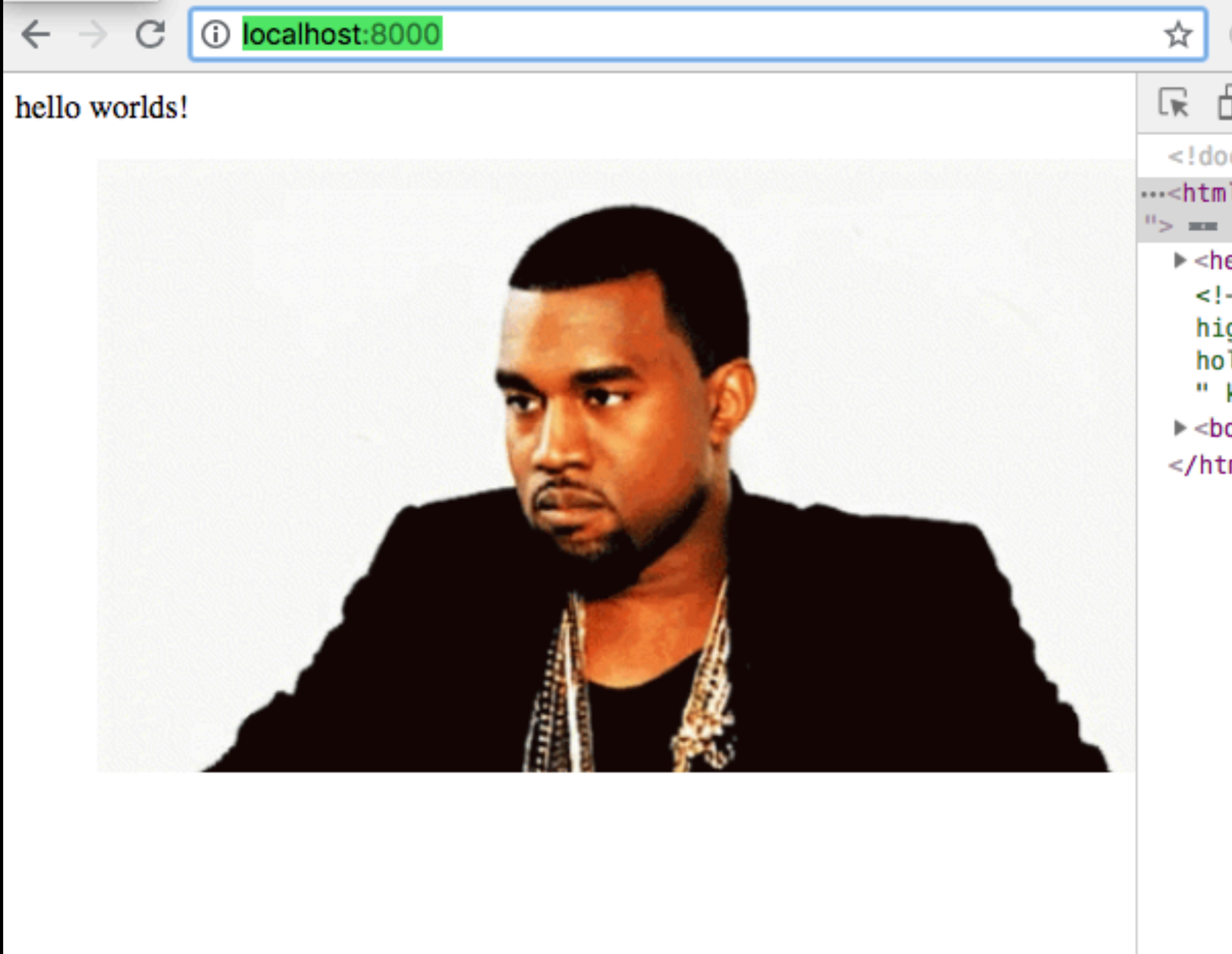
Starting a local http server from the command line…

If you have installed Python 3.0+: `python -m http.server`

in browser address bar: `localhost:8000`

Mac - to close the server: **COMMAND C**
Wndws - to close the server: **CNTRL C**

how to install python: https://github.com/rebleo/intrntWeb_LC/blob/master/week02/week2windows.pdf



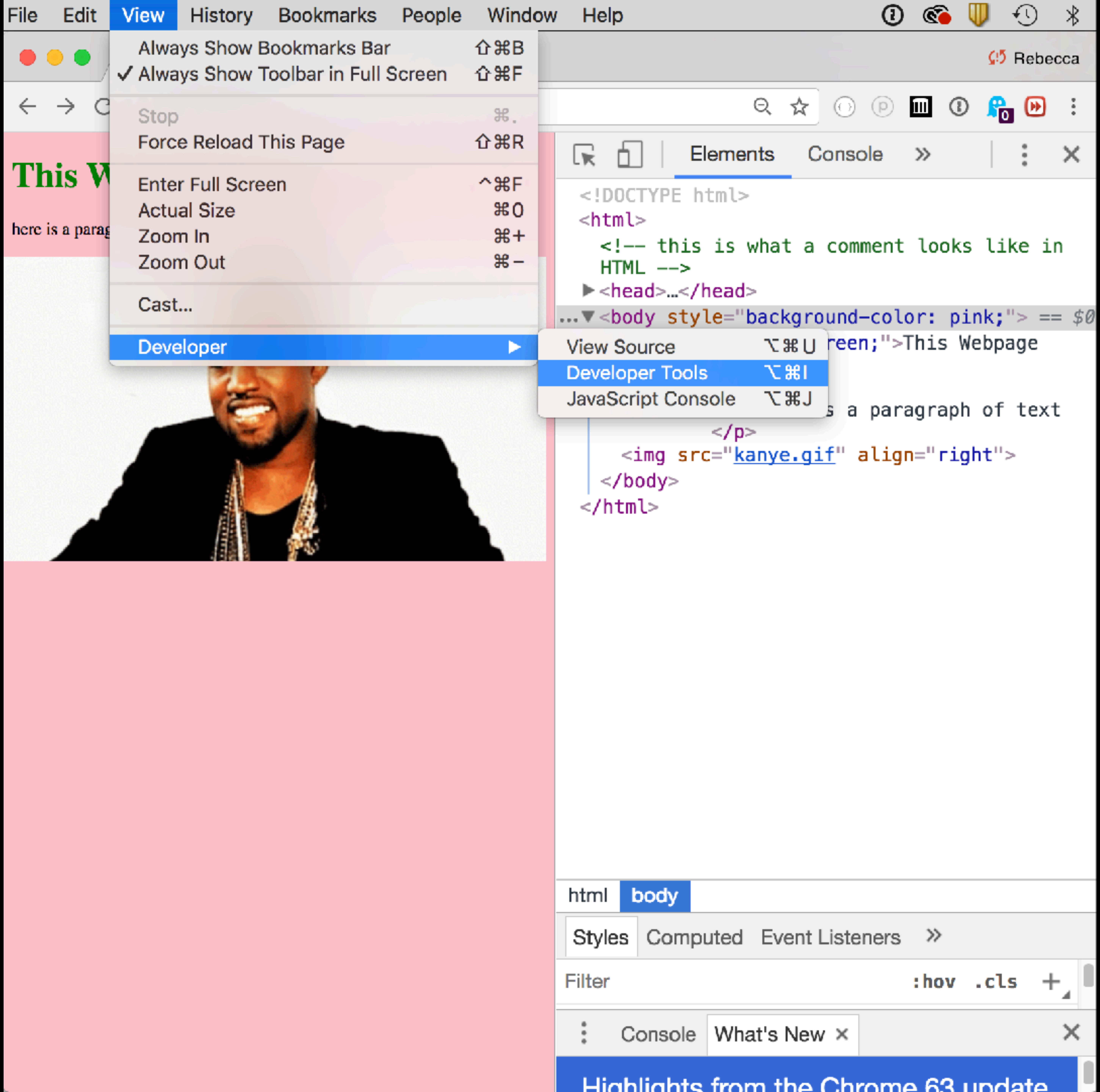
localhost:8000

hello worlds!

url is:

localhost:8000

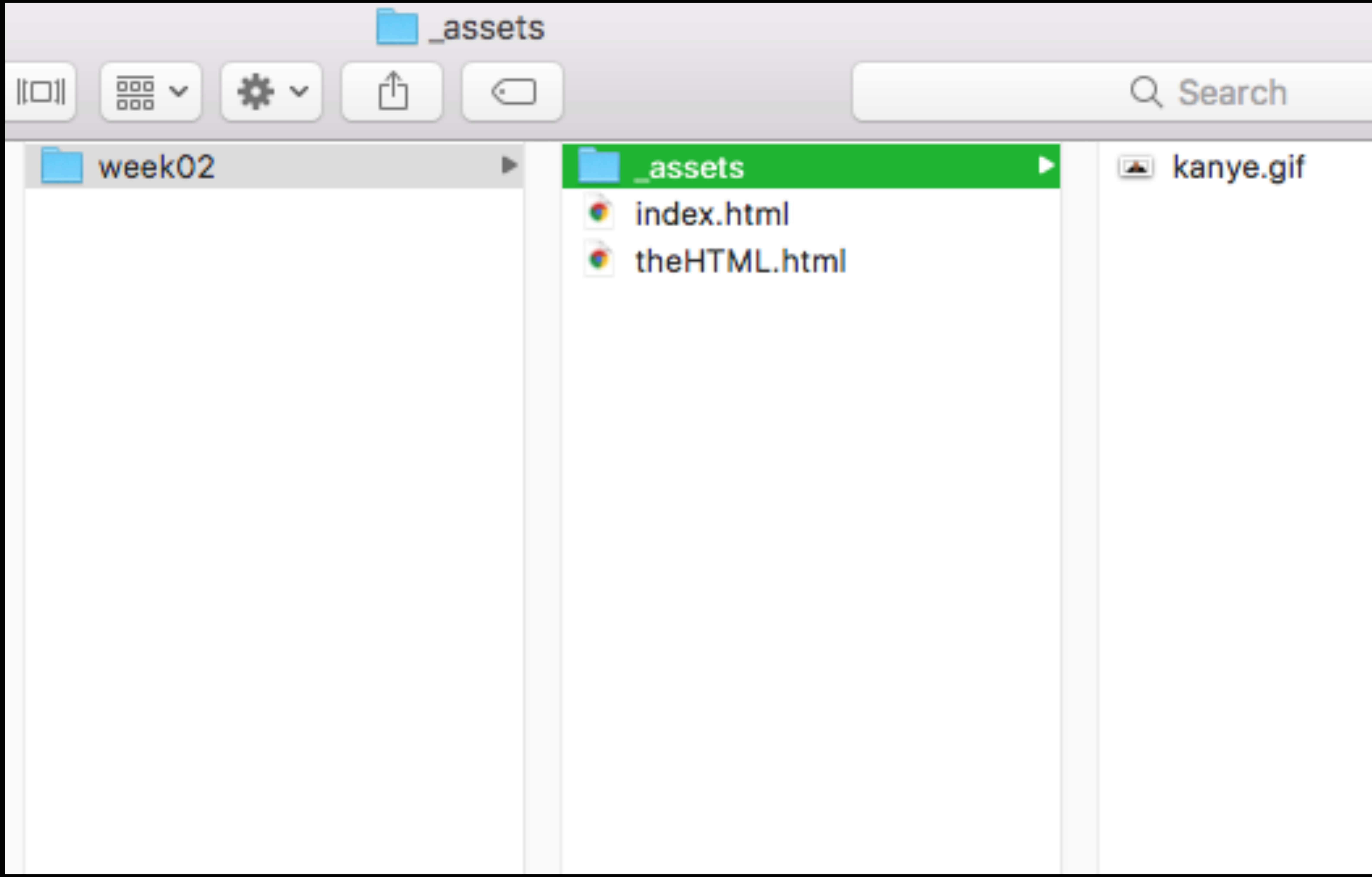
- **Google Chrome** Browser
  - Dev Tools (cmmd i)



```
<!DOCTYPE html>
<html>
  <!-- this is what a comment looks like in
  HTML -->
 ▶<head>…</head>
…▼<body style="background-color: pink;"> == $0
                          reen;">This Webpage

                        s a paragraph of text
        </p>
     <img src="kanye.gif" align="right">
   </body>
</html>
```

publishing yr webpage

Parent + Child File Structure
or: **File Paths**
or: **URL ADDRESSES!**

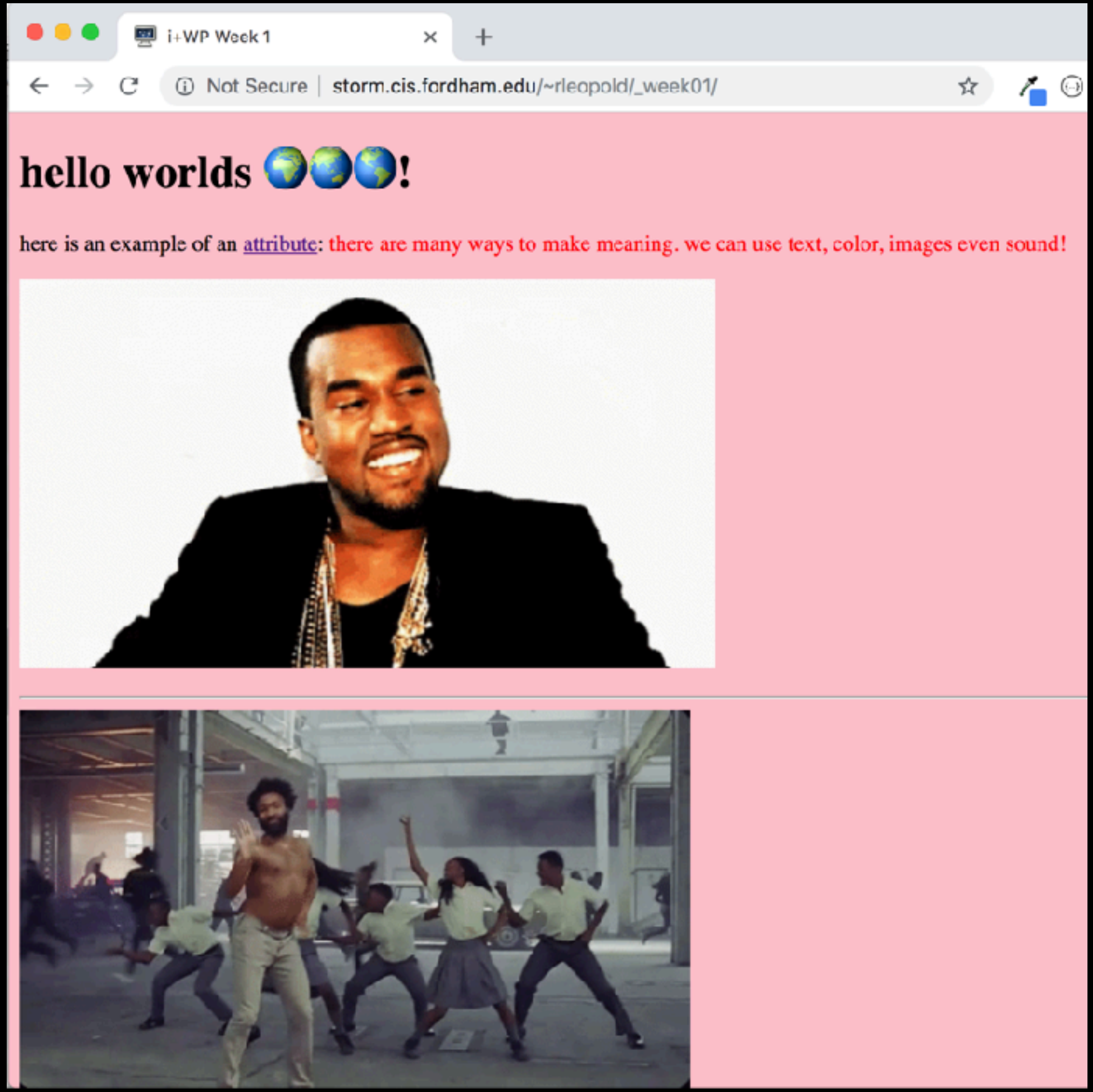


intrntWeb_LC — rleopold@storm:~/public_html — 79×31
~/Desktop/teach/frdhm/_Spring-2019/intrntWeb_LC — rleopold@storm:~/public_html — ssh rleopold@storm.cis.fordham.edu

```
 intrntWeb_LC    ssh rleopold@storm.cis.fordham.edu
Password:
Last failed login: Thu Jan 17 18:35:02 EST 2019 from [redacted]61 on ssh:notty
There was 1 failed login attempt since the last successful login.
Last login: Thu Jan 17 18:32:32 2019 from 10.[redacted]
[rleopold@storm ~]$ ls
private  public_html
[rleopold@storm ~]$ cd public_html/
[rleopold@storm public_html]$ ls
_week01  index.html  week01  week02  week03  week04
[rleopold@storm public_html]$ 
```

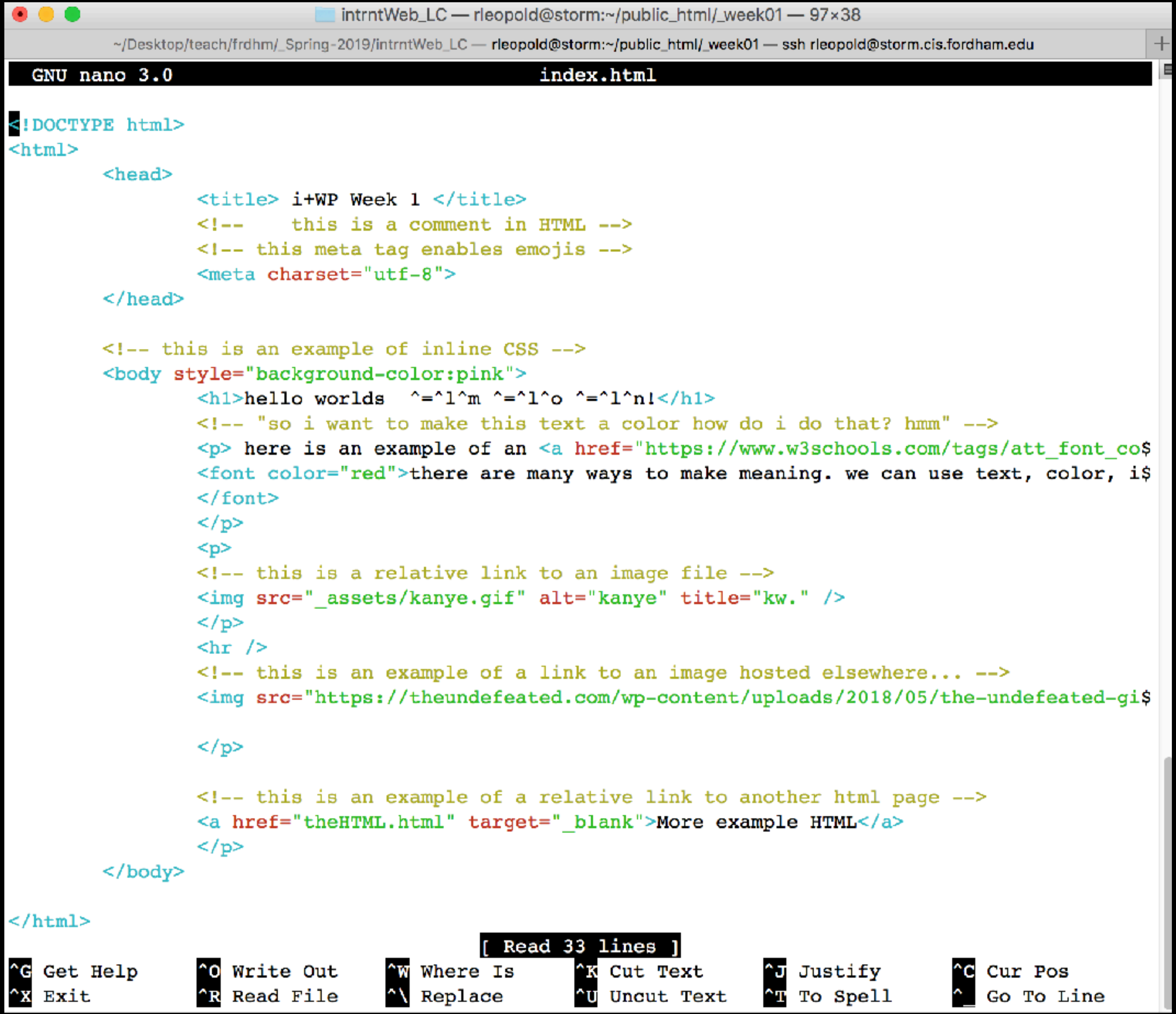
option 1:

ssh using command line: ssh username @ storm.cis.fordham.edu



hello worlds 🌍🌏🌎!

here is an example of an attribute: there are many ways to make meaning. we can use text, color, images even sound!



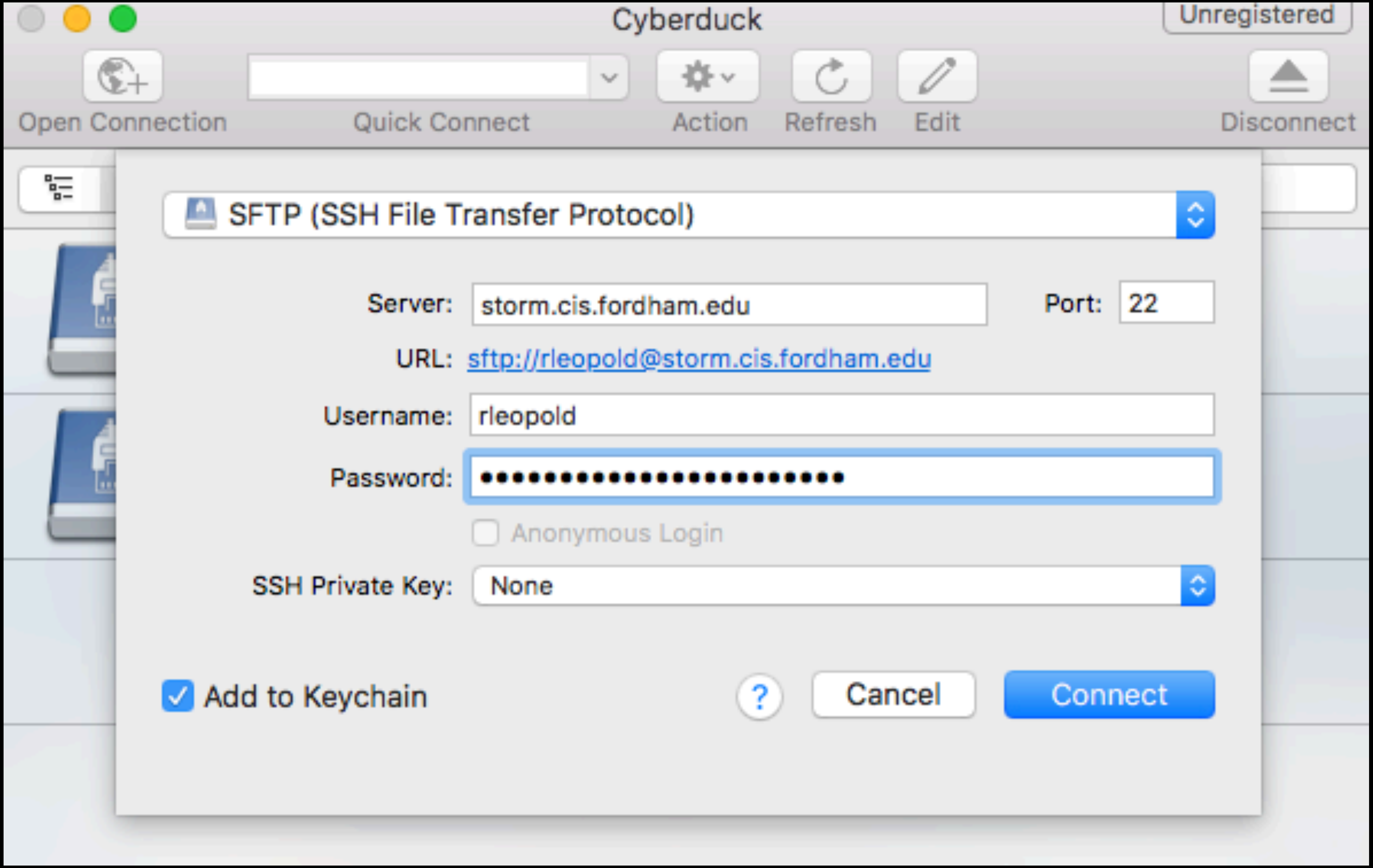
intrntWeb_LC — rleopold@storm:~/public_html/_week01 — 97×38
~/Desktop/teach/frdhm/_Spring-2019/intrntWeb_LC — rleopold@storm:~/public_html/_week01 — ssh rleopold@storm.cis.fordham.edu

```
 GNU nano 3.0                                    index.html

<!DOCTYPE html>
<html>
        <head>
                <title> i+WP Week 1 </title>
                <!--    this is a comment in HTML -->
                <!-- this meta tag enables emojis -->
                <meta charset="utf-8">
        </head>

        <!-- this is an example of inline CSS -->
        <body style="background-color:pink">
                <h1>hello worlds  ^=^l^m ^=^l^o ^=^l^n!</h1>
                <!-- "so i want to make this text a color how do i do that? hmm" -->
                <p> here is an example of an <a href="https://www.w3schools.com/tags/att_font_co$
                <font color="red">there are many ways to make meaning. we can use text, color, i$
                </font>
                </p>
                <p>
                <!-- this is a relative link to an image file -->
                <img src="_assets/kanye.gif" alt="kanye" title="kw." />
                </p>
                <hr />
                <!-- this is an example of a link to an image hosted elsewhere... -->
                <img src="https://theundefeated.com/wp-content/uploads/2018/05/the-undefeated-gi$

                </p>

                <!-- this is an example of a relative link to another html page -->
                <a href="theHTML.html" target="_blank">More example HTML</a>
                </p>
        </body>

</html>
                                      [ Read 33 lines ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^_ Go To Line
```

option 1:

edit web files w/ Nano, Vim, etc.

this means copying + pasting from yr local machine.



SFTP (SSH File Transfer Protocol)

Server: storm.cis.fordham.edu

Port: 22

URL: sftp://rleopold@storm.cis.fordham.edu

Username: rleopold

Password: ••••••••••••••••••••••••

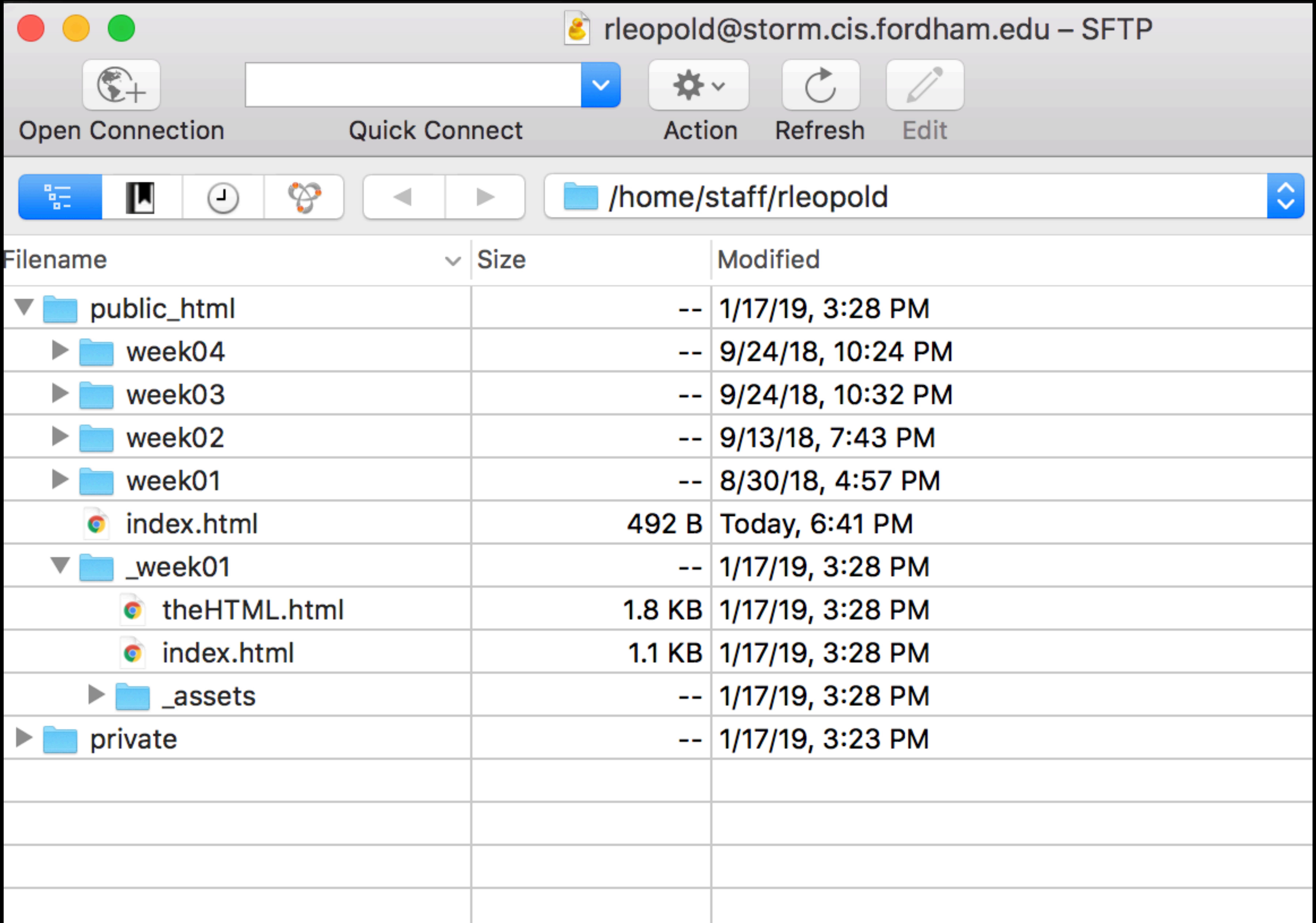
□ Anonymous Login

SSH Private Key: None

☑ Add to Keychain

option 2:

Download an SFTP client software like Cyberduck or Fetch or Filezilla



rleopold@storm.cis.fordham.edu – SFTP

/home/staff/rleopold

Filename	Size	Modified
public_html	--	1/17/19, 3:28 PM
week04	--	9/24/18, 10:24 PM
week03	--	9/24/18, 10:32 PM
week02	--	9/13/18, 7:43 PM
week01	--	8/30/18, 4:57 PM
index.html	492 B	Today, 6:41 PM
_week01	--	1/17/19, 3:28 PM
theHTML.html	1.8 KB	1/17/19, 3:28 PM
index.html	1.1 KB	1/17/19, 3:28 PM
_assets	--	1/17/19, 3:28 PM
private	--	1/17/19, 3:23 PM

option 2:

ftp client software. Drag + drop or upload buttons.

i+WP Week 1
Not Secure
storm.cis.fordham.edu/~rleopold/_week01/

hello worlds 🌍🌏🌎!

http://storm.cis.fordham.edu/~username

< HTML >

3 categories of html elements:

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

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

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

<a>, **<em>**, **<strong>**, **<br>**

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

<img>

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

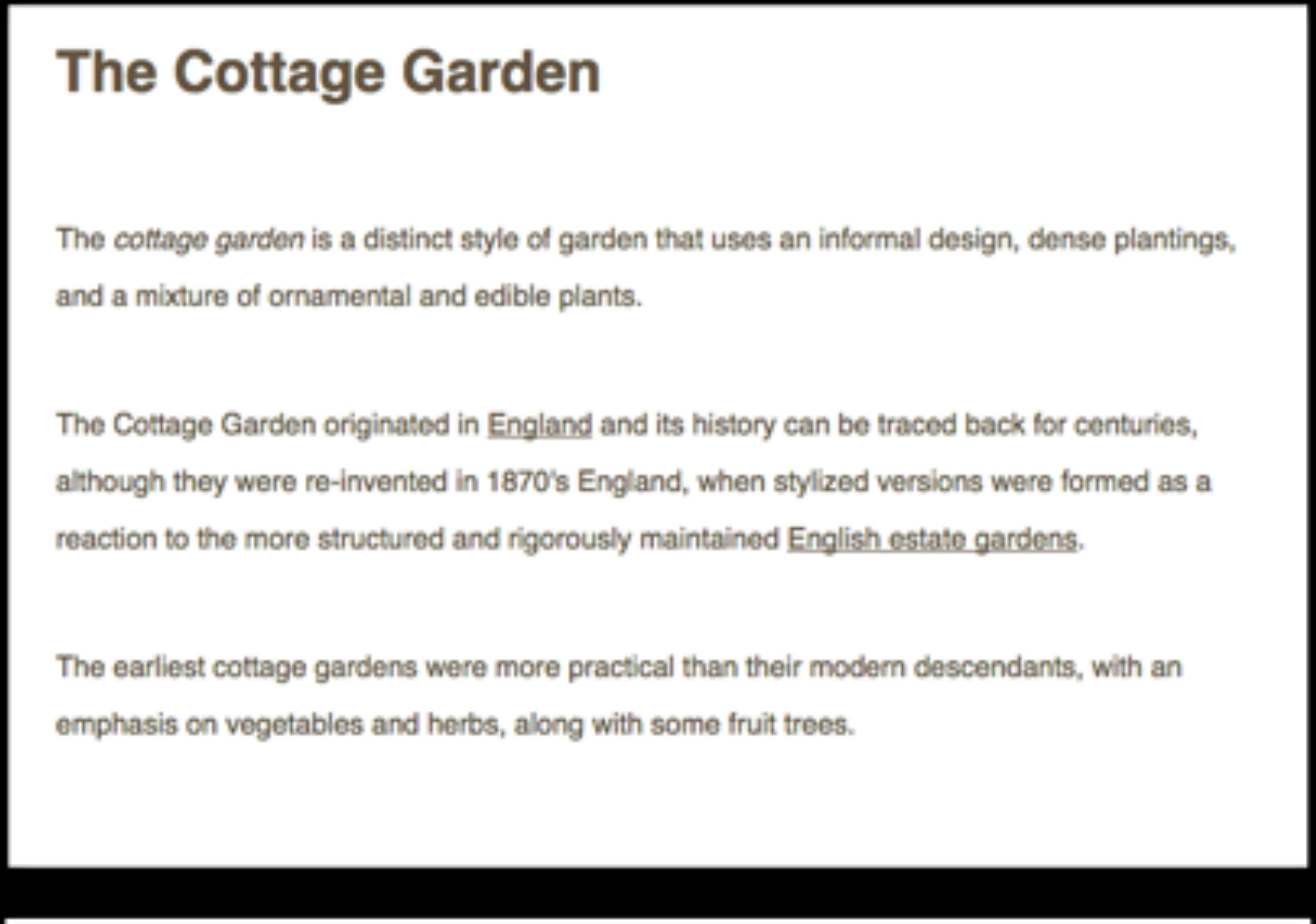
<title>, **<meta>**, **<script>**

CSS works by associating rules with HTML elements. These rules govern how the content of specified elements should be displayed.

A CSS rule contains two parts: a selector and a declaration.

** pro tip: It takes 5% to learn how to write CSS rule and 95% to learn different properties that you can use.

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



The Cottage Garden

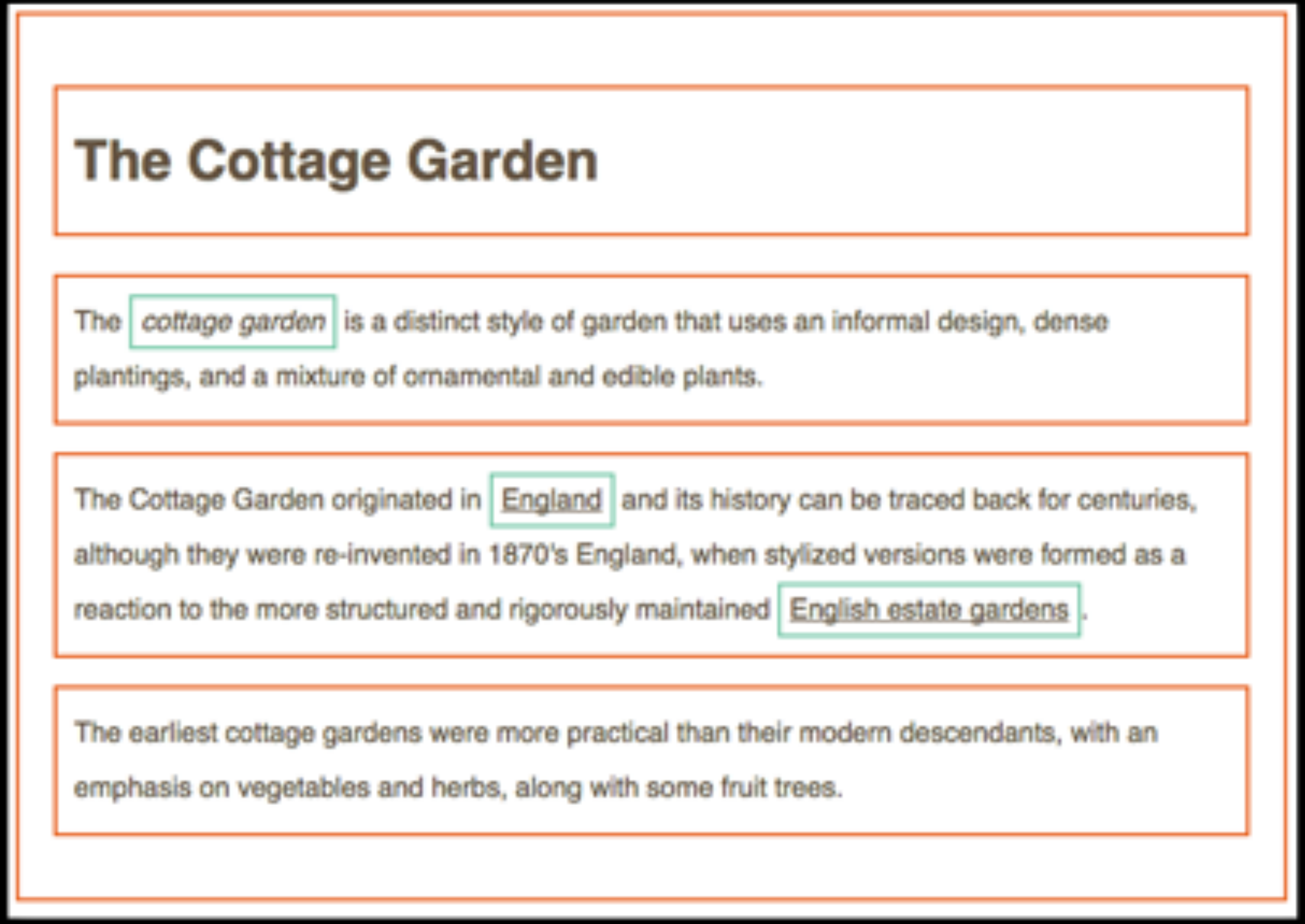
The *cottage garden* is a distinct style of garden that uses an informal design, dense plantings, and a mixture of ornamental and edible plants.

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.

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.

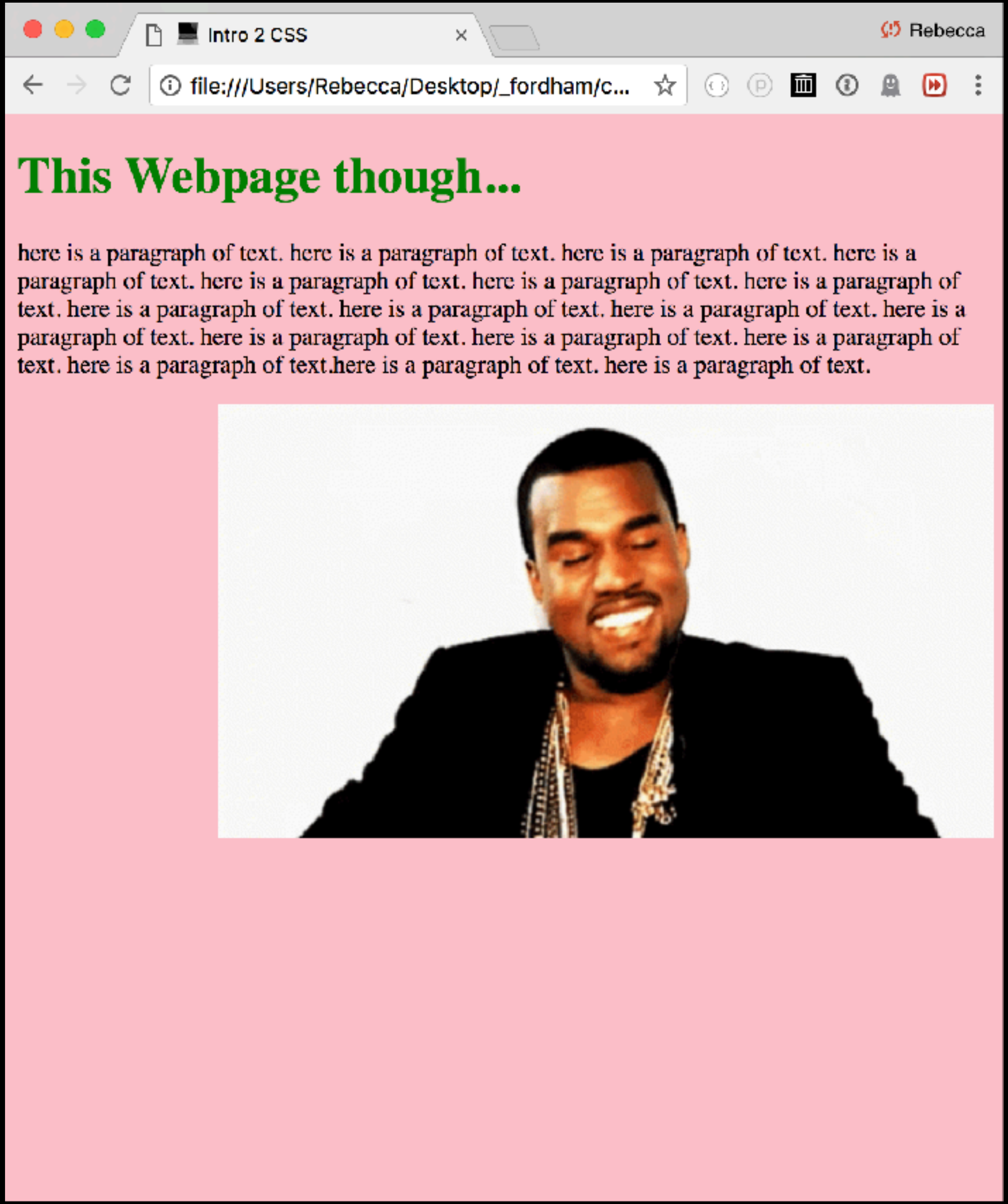
You can write CSS 3 Different Ways:

Inline Styles

```
<h1 style="color:green;">This Webpage though...</h1>
<body style="background-color: pink;">
```

Embedded Styles

Externals Styles



This Webpage though...

here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text.here is a paragraph of text. here is a paragraph of text.

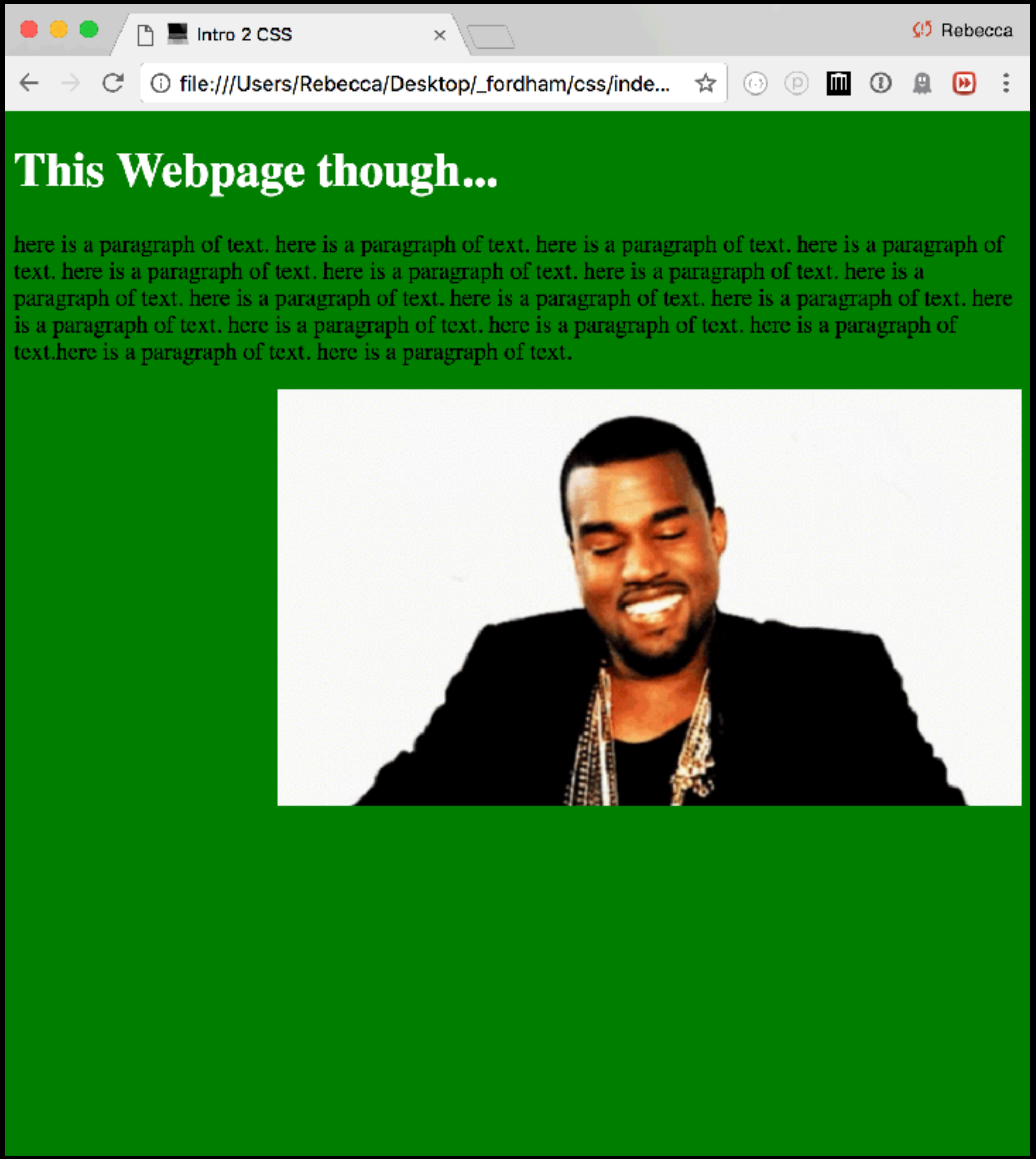
Inline Styles

```
<h1 style=“color:white;">This Webpage though…</h1>
<body style="background-color: green;”>
```

Embedded Styles

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

              body {
                    background: green;
              }

        </style>
   </head>
```

Externals Styles



This Webpage though...

here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text.here is a paragraph of text. here is a paragraph of text.

Inline Styles

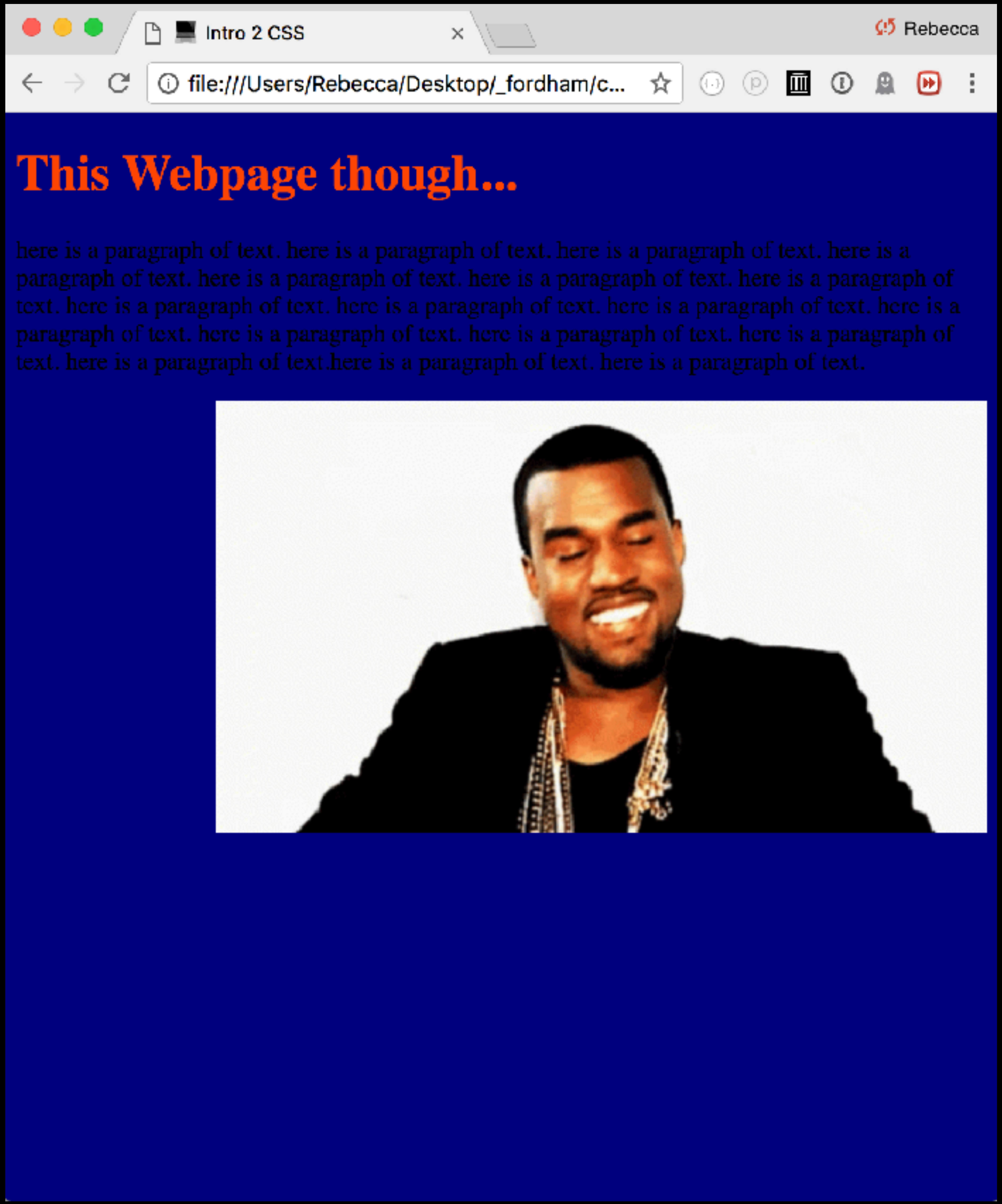
```
  <h1 style="color:#FF4500;">This Webpage though...</h1>
  <body style="background-color: #000080;">
```

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>
```



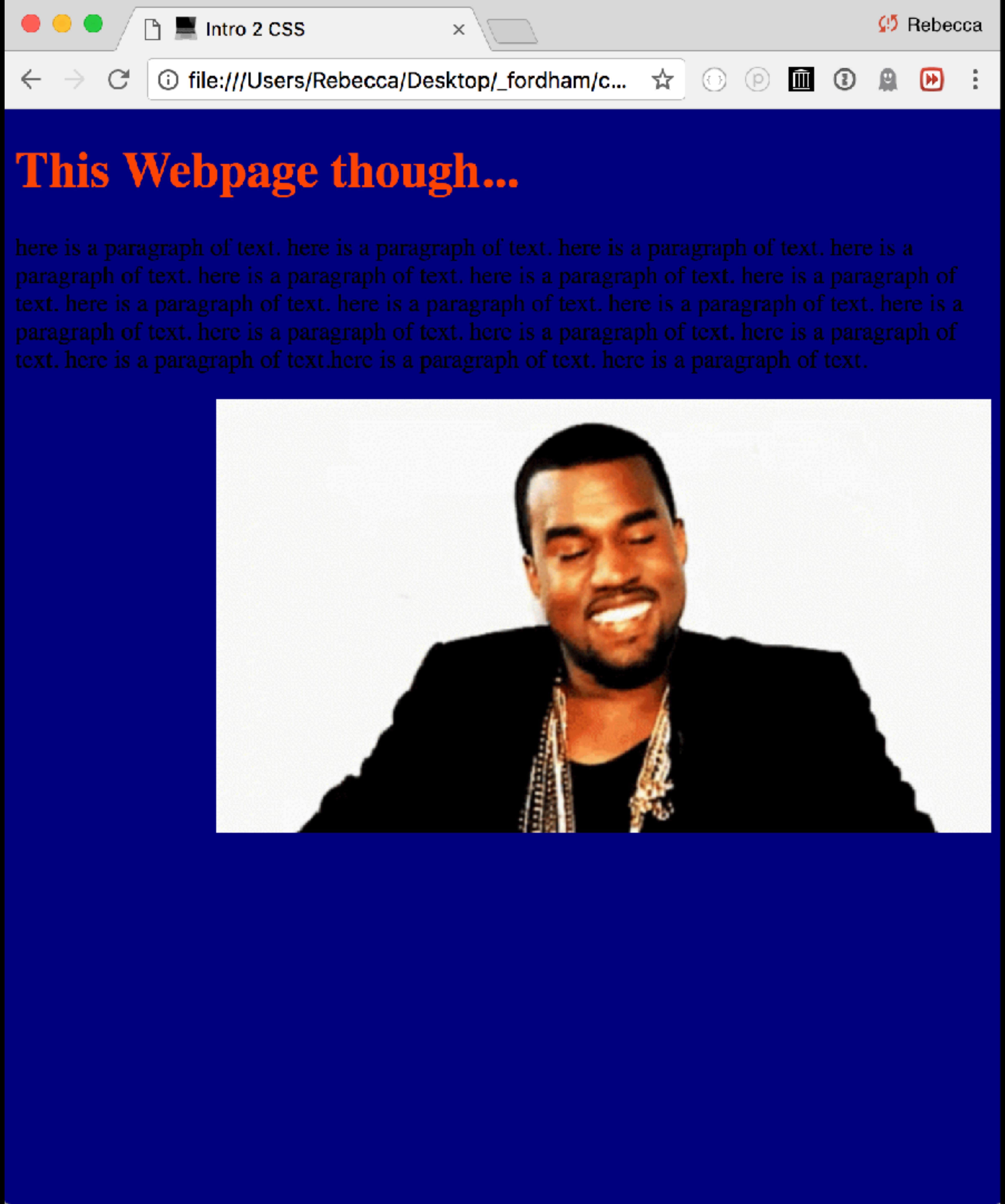
This Webpage though...

here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text.here is a paragraph of text. here is a paragraph of text.

CSS SYNTAX:

selectors are used to find (select) HTML elements based on their element name, id, etc…

```
selector {

 property: value ;

}
```



This Webpage though...

here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a paragraph of text.here is a paragraph of text. here is a paragraph of text.

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	.note { }
		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 <li> 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;
}

/* class attribute
selector */
.blue-text {
    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;


}
```

Units + Dimensions

Length & Size

The most common units we'll be using for setting the of an element or property are:

- `px`
- `%`
- `em`
- `rem`

```
 {
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> + <div>, 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>, <em> or <strong>. 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 {

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.

: active {

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

: focus {

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.

}

HTML comments are written like this

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

CSS comments are written like this

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

More on Cascading

From John Duckett book:

Last rule
If the two selectors are identical, the latter of the two will take precedence. For example, if there were two **i** elements in style sheet, the second one would take precedence over the first.

Specificity
If one selector is more specific than the others, the more specific rule will take precedence over more general ones. For example, **h1** is more specific than body tag and so one.

Important
You can add **!important** after any property value to indicate that it should be considered more important than other rules that apply to the same element.