Ways of Seeing - John Berger, 1972

file paths

Documents

Desktop

idm

Spring2020

webDev

myClassDirectory

if you haven’t already - please take moment to create a directory for this class in a location that makes sense for you.

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 '**indexv1.html**, **indexv2.html**, **indexv3FINAL.html**' problem

At its most complex, git allows developers to work together worldwide on code without stepping on each other's toes

Github

Github is a service to host yr projects on the web.

Code is pushed (uploaded) from a local directory (folder) called a repository or rep.

example - our class site:

http://www.github.com/rebleo/webDevSpring2020

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 in a '**repo**', or '**repository**' on your local machine

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

HTTPS

Hypertext Transfer Protocol Secure - is an extension of the Hypertext Transfer Protocol (HTTP). It is used for secure communication over a computer network, and is widely used on the Internet.

SSH

Secure Shell or Secure Socket - a network protocol that gives users, particularly system administrators, a secure way to access a computer over an unsecured network. **SSH** also refers to the suite of utilities that implement the **SSH** protocol.

Git commands

Git Terminal Commands

clone - download a copy for repo to your local machine

status - view the change status of the repo

add - add changed files to be committed

commit - **-m** “My saved message here” (this message will be public)

push - send your committed updates to github

Example flow of operations

git status

git add .

git commit -m “i am saving my work to github. I am writing..”

git status

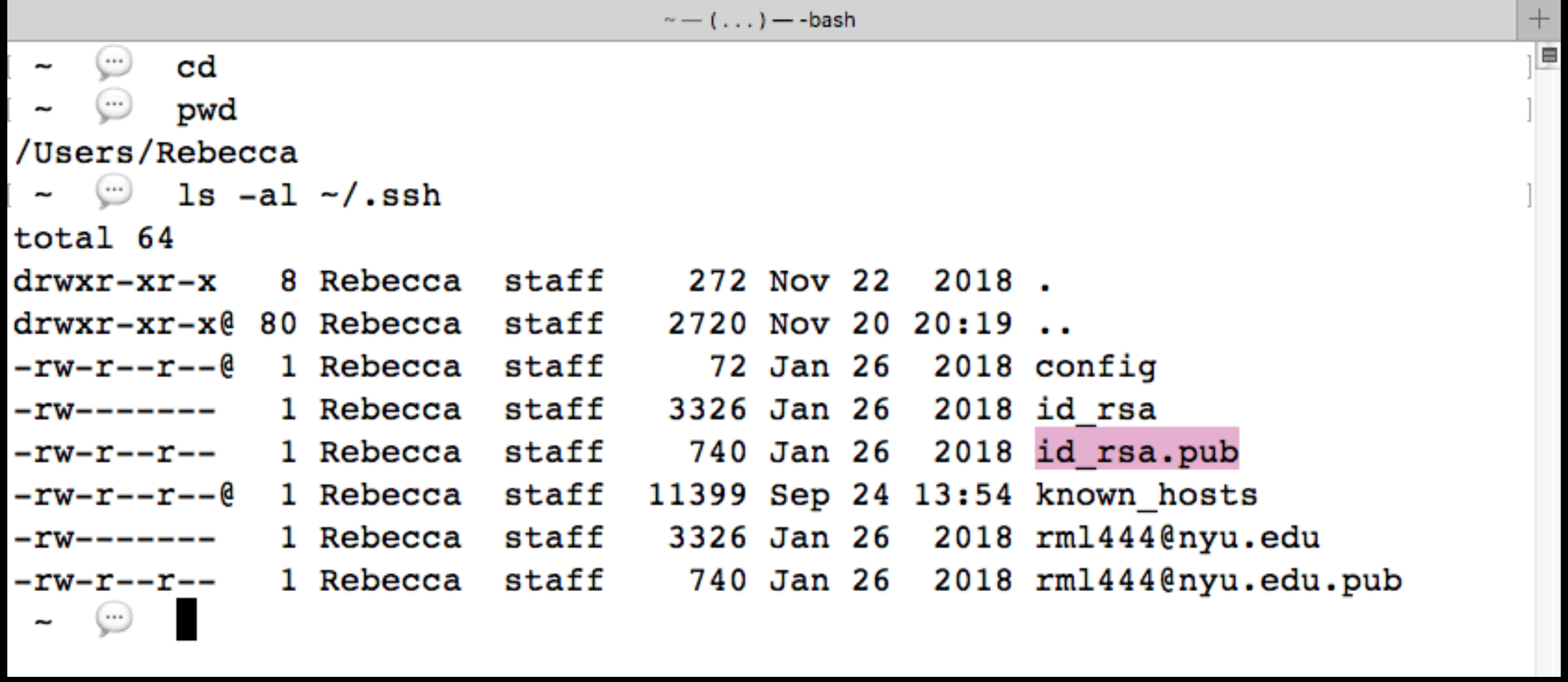
git push origin master

Github pages

https://pages.github.com

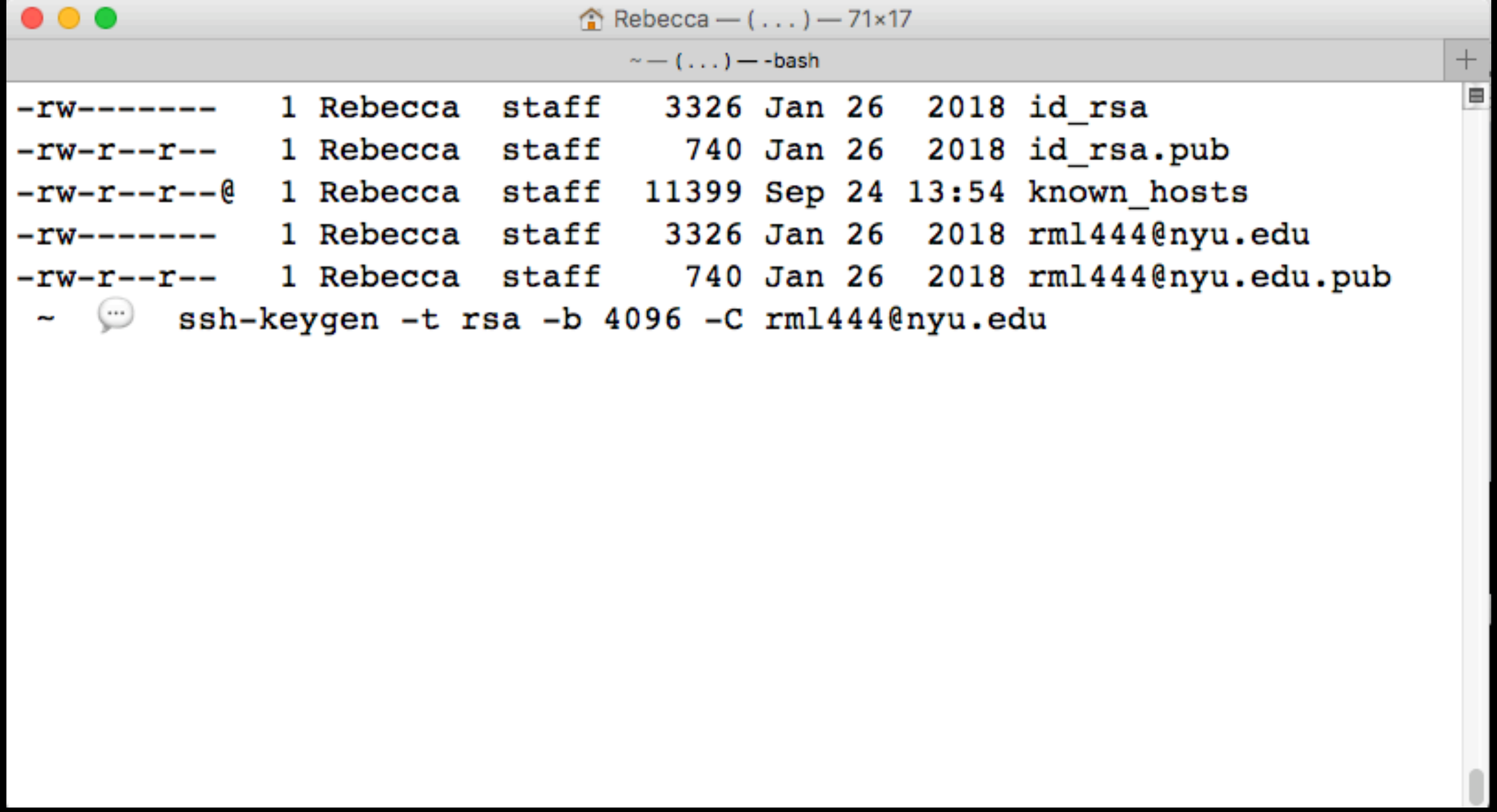
Git Steps

1. Create a new repo on yr Github (git init)
2. Clone yr new repo somewhere on yr local machine
3. Make yr first commit



Check for SSH Keys

At the root of your machine (your user) - in Bash: type **cd**. Then **pwd** to check…
Using **ls ~/.ssh** (list specific computer readable files w/ exertions .ssh). Look for **id_rsa.pub**, if it's not there we'll fix that!



Generate a new SSH Key

Make sure to use the same email you used to create your Github account

"Enter a file in which to save the key", just press **Enter** to continue.

```
Enter file in which to save the key (/Users/you/.ssh/id_rsa): [Press enter]
```

3 You'll be asked to enter a passphrase.

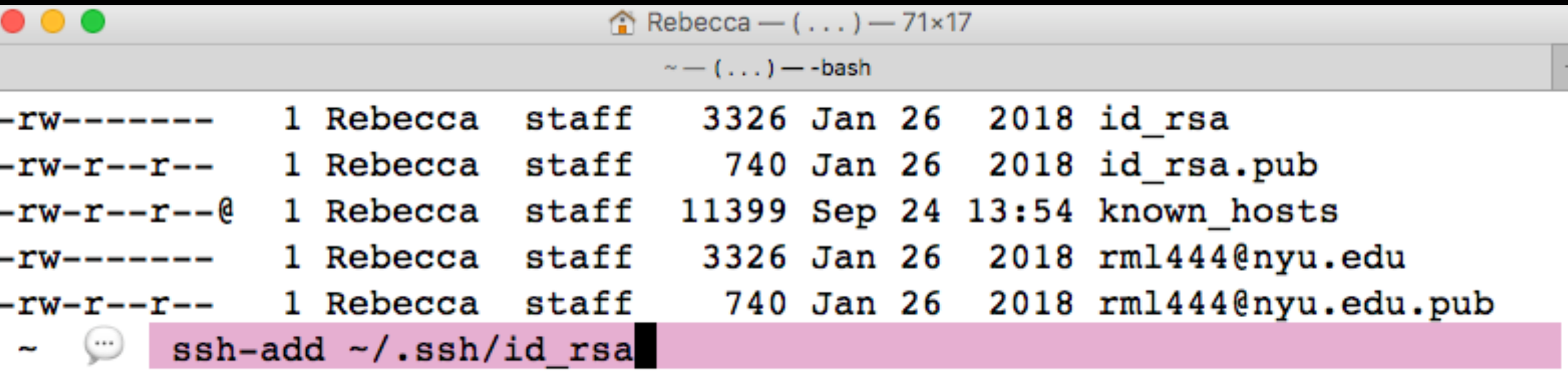
```
Enter passphrase (empty for no passphrase): [Type a passphrase]
Enter same passphrase again: [Type passphrase again]
```

Tip: We strongly recommend a very good, secure passphrase. For more information, see "Working with SSH key passphrases".

4 After you enter a passphrase, you'll be given the fingerprint, or *id*, of your SSH key. It will look something like this:

```
Your identification has been saved in /Users/you/.ssh/id_rsa.
Your public key has been saved in /Users/you/.ssh/id_rsa.pub.
The key fingerprint is:
01:0f:f4:3b:ca:85:d6:17:a1:7d:f0:68:9d:f0:a2:db your_email@example.com
```

Finish creating the key

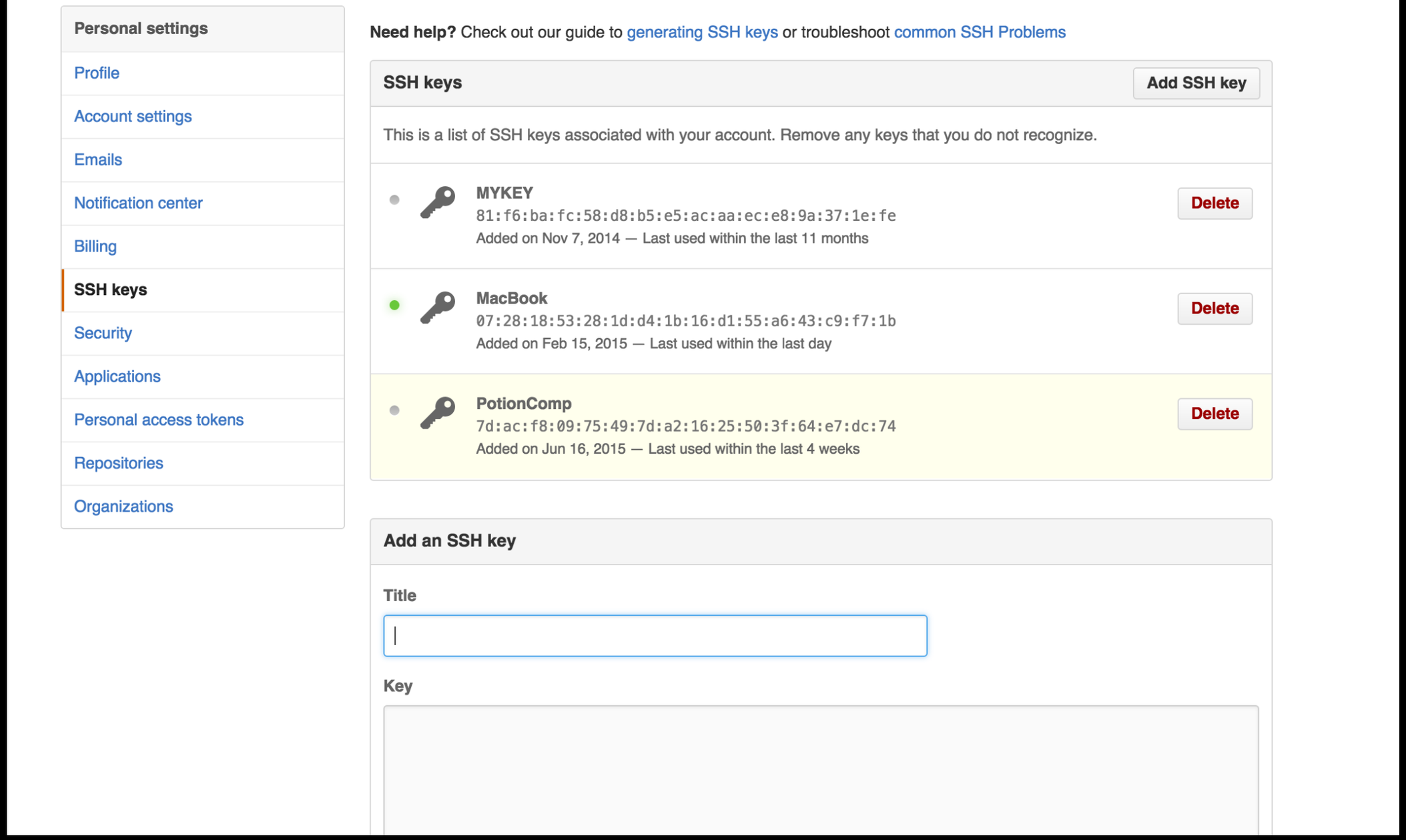


Add your key to the SSH Agent

```
$ pbcopy < ~/.ssh/id_rsa.pub
# Copies the contents of the id_rsa.pub file to your clipboard
```

copy key to clipboard

This command reads your key file (id_rsa.pub) and copies the contents to your clip board.



Add key to Github by going to Settings. Paste your key in the key section (command + v in the key section and it will appear)

now you can create new repos + clone them to your local machine!!

Example flow of operations

git status

git add .

git commit -m “i am saving my work to github. I am writing..”

git status

git push origin master

Where is git? Once you've installed git, you can verify it has been installed by

opening up the Terminal and typing git

```
$ git
```

If you see a 'command not recognized' error, you probably haven't installed git

Go through the process of creating a repo for your Github pages site. Clone it inside the webDev directory you made earlier. Ta Da!

http://yourUserName.github.io

yrUsername.github.io

Starting a local http server from the command line using Python...

If you have installed Python 3.0+:

```
python -m http.server
python -m http.server 12345
```

in browser address bar:

```
localhost:8000
localhost:12345
```

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

Starting a local http server from the command line using Node.js...

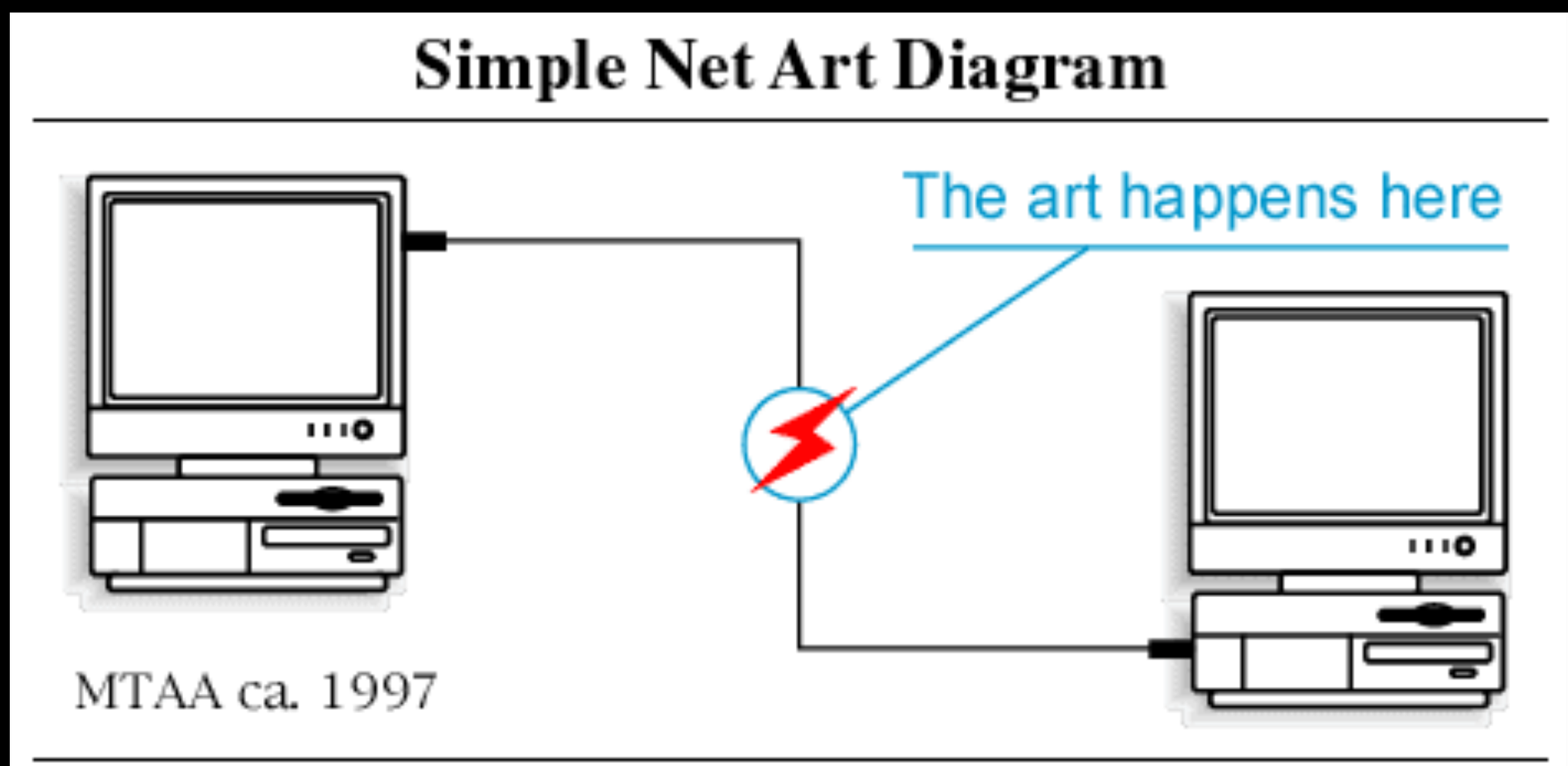
```
http-server
http-server -p 12345
```

in browser address bar:

```
localhost:8080
localhost:12345
```

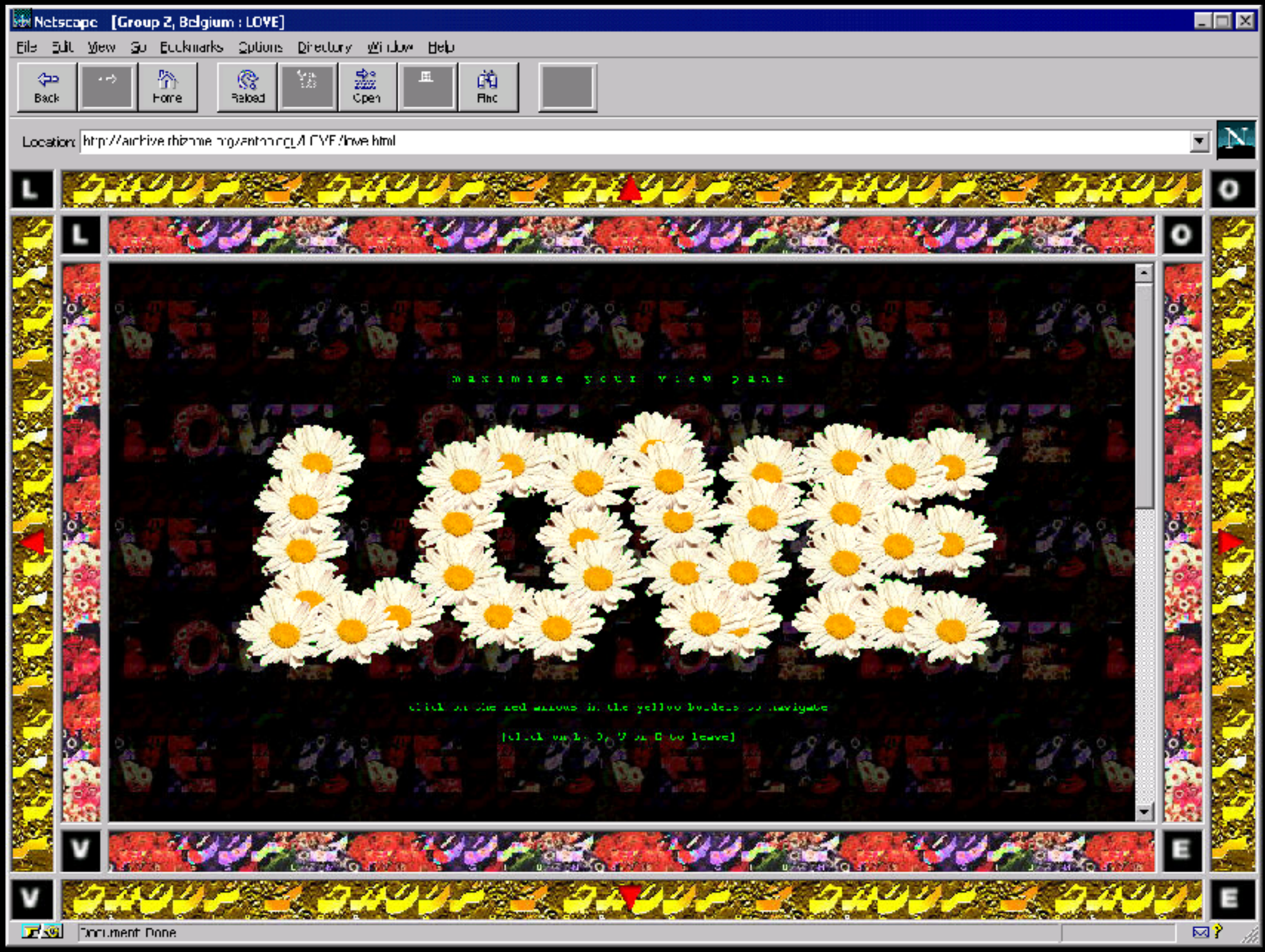
Mac - to close the server: **COMMAND C**
Wndws - to close the server: **CNTRL C**

Web as Medium

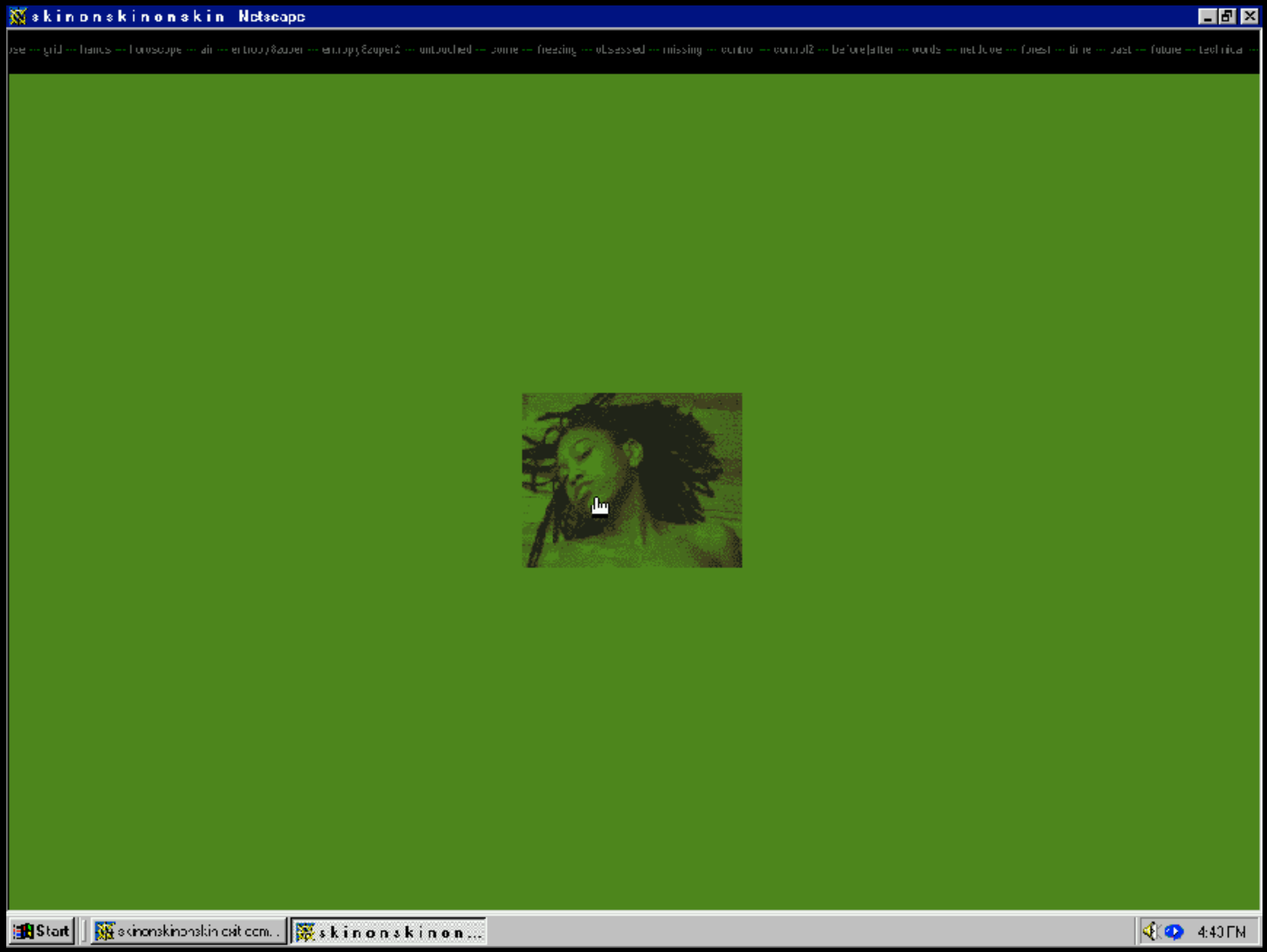


MTAA, 1997

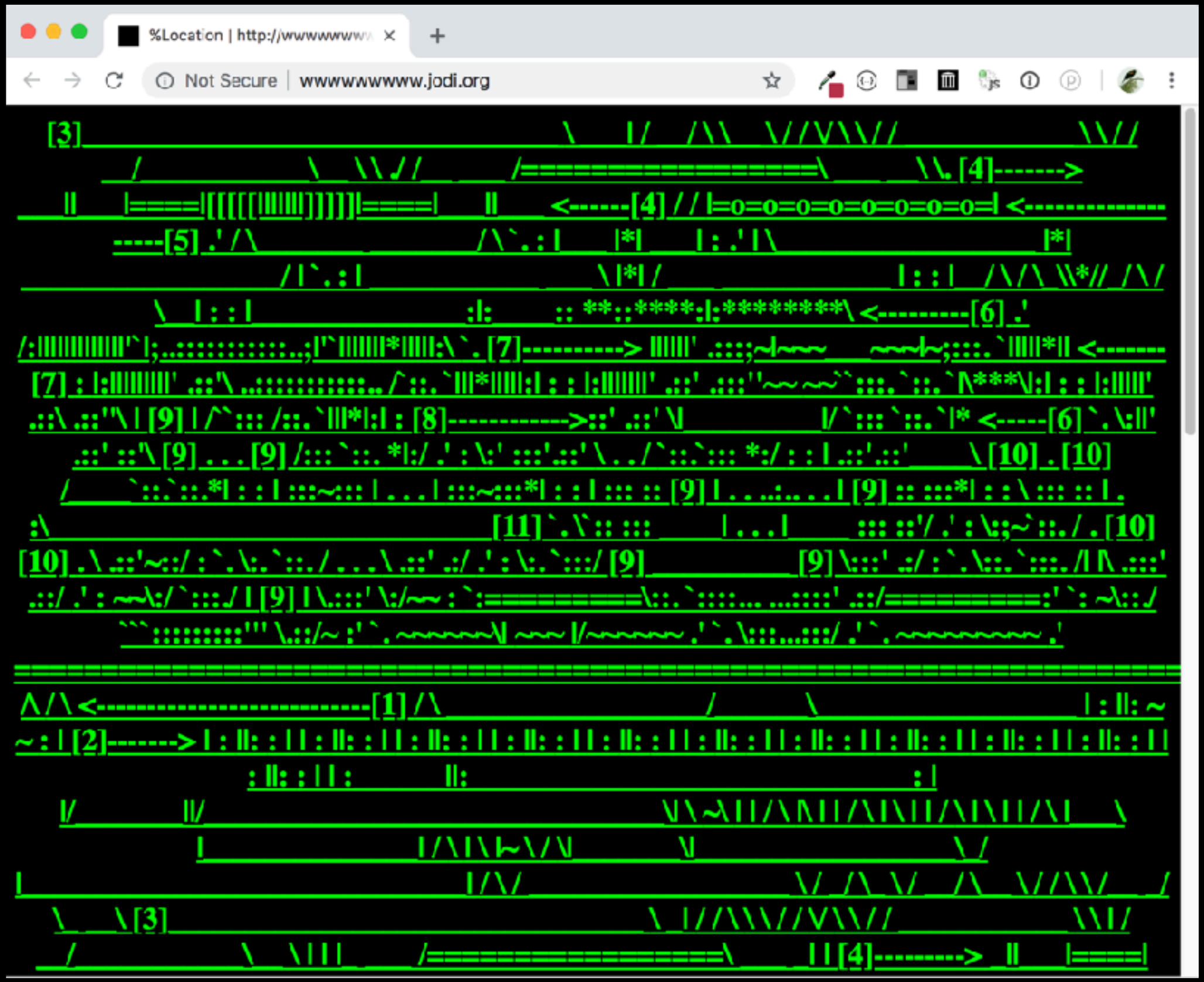
My Boyfriend Came Back From War, 1996
Olia Lialina



Love, 1995
Group Z (Michaël Samyn)



skinonskinonskin, 1999
Auriea Harvey (Entropy8) + Michaël Samyn (Zuper!)
Entropy8Zuper!



[jodi.org](), 1999 - 2001

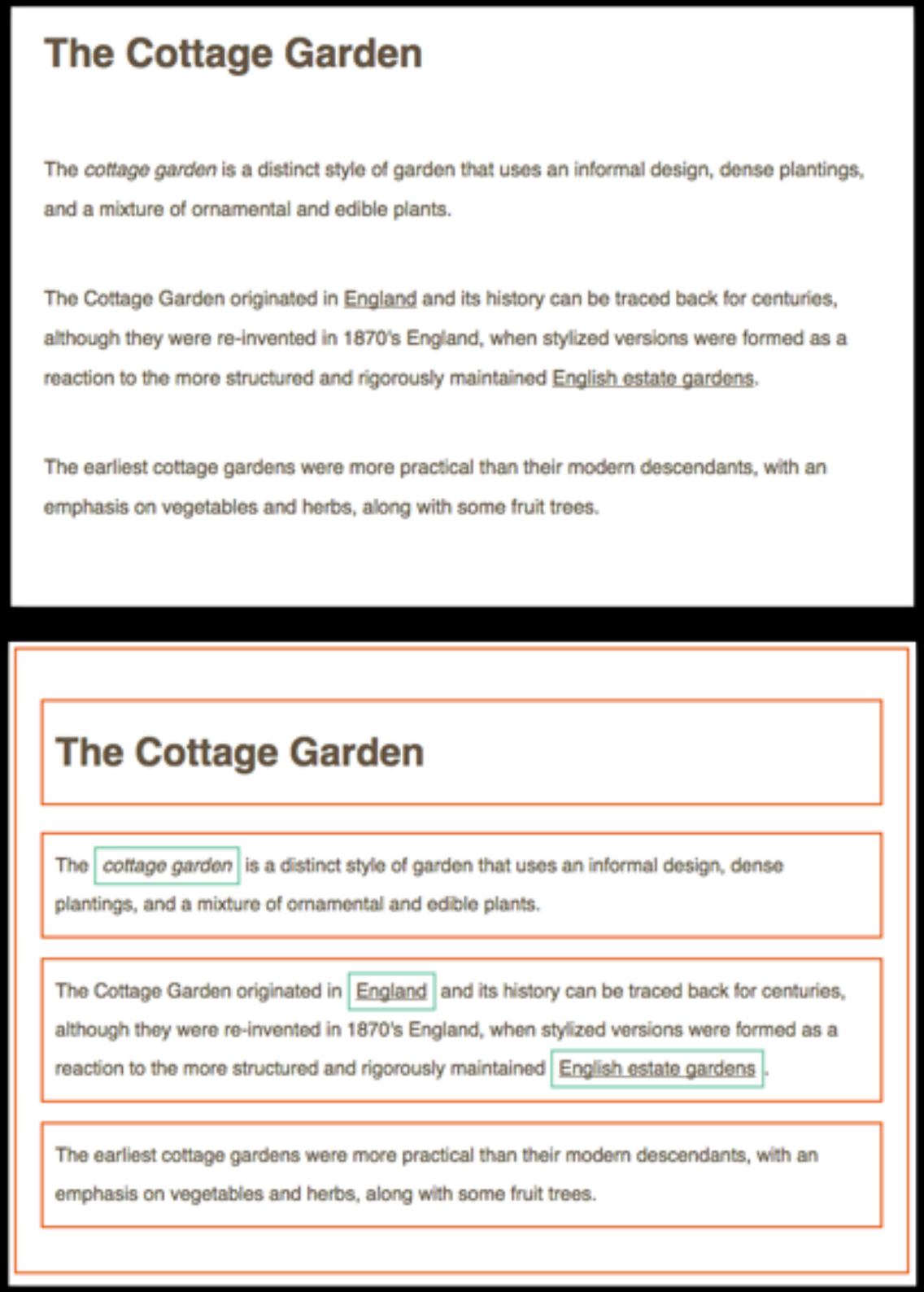
The Whale Hunt, 2007
Jonathan Harris

< HTML >

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.



< 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>, **, **, **
, , <time>**
a. **inline block**: inline content w/ height + width

3 - metadata: information abou the page, usually not visible
<title>, **<meta>**, **<script>**

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

Structure tags

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.

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 though 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

https://developer.mozilla.org/en-US/docs/Glossary/Semantics

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

<aside> 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>` element.

<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.

<a> links

OPENING LINK TAG

WE ARE DIRECTED TO

TEXT WE CLICK ON

 IDM - Tandon

< a href — stands for *hyperlink reference*

<a> relative urls

Linking to pages on the same site

Same Directory `<a href="week00.html" > Week 1 Page </a>`

Child Directory `<a href="myBlog/week00.html" > Week 1 Page </a>`

id attribute `<a href="#potatoGallery" > Click here for this week in potatoes! </a>`

Parent Directory `<a href="../index.html" > Home Page </a>`

< a href — stands for *hyperlink reference*

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 directories’ root webpages index.html

Images: relative vs. 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” >
```