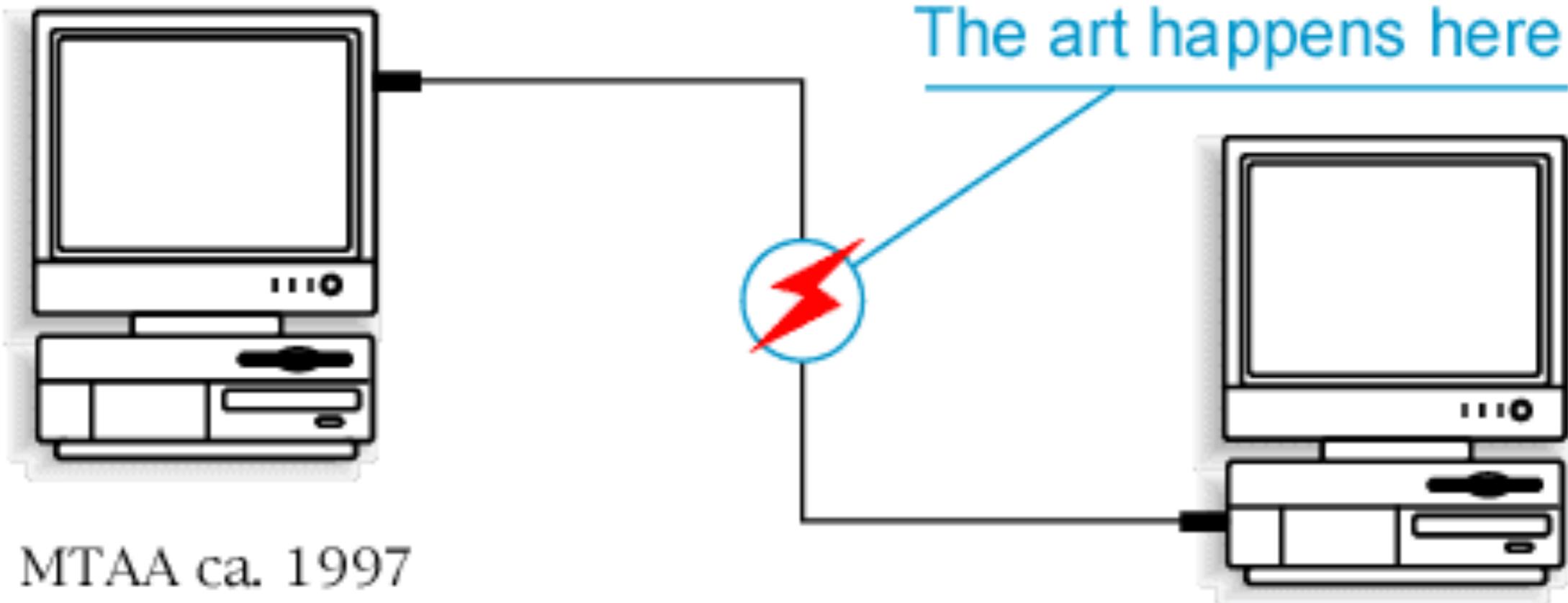
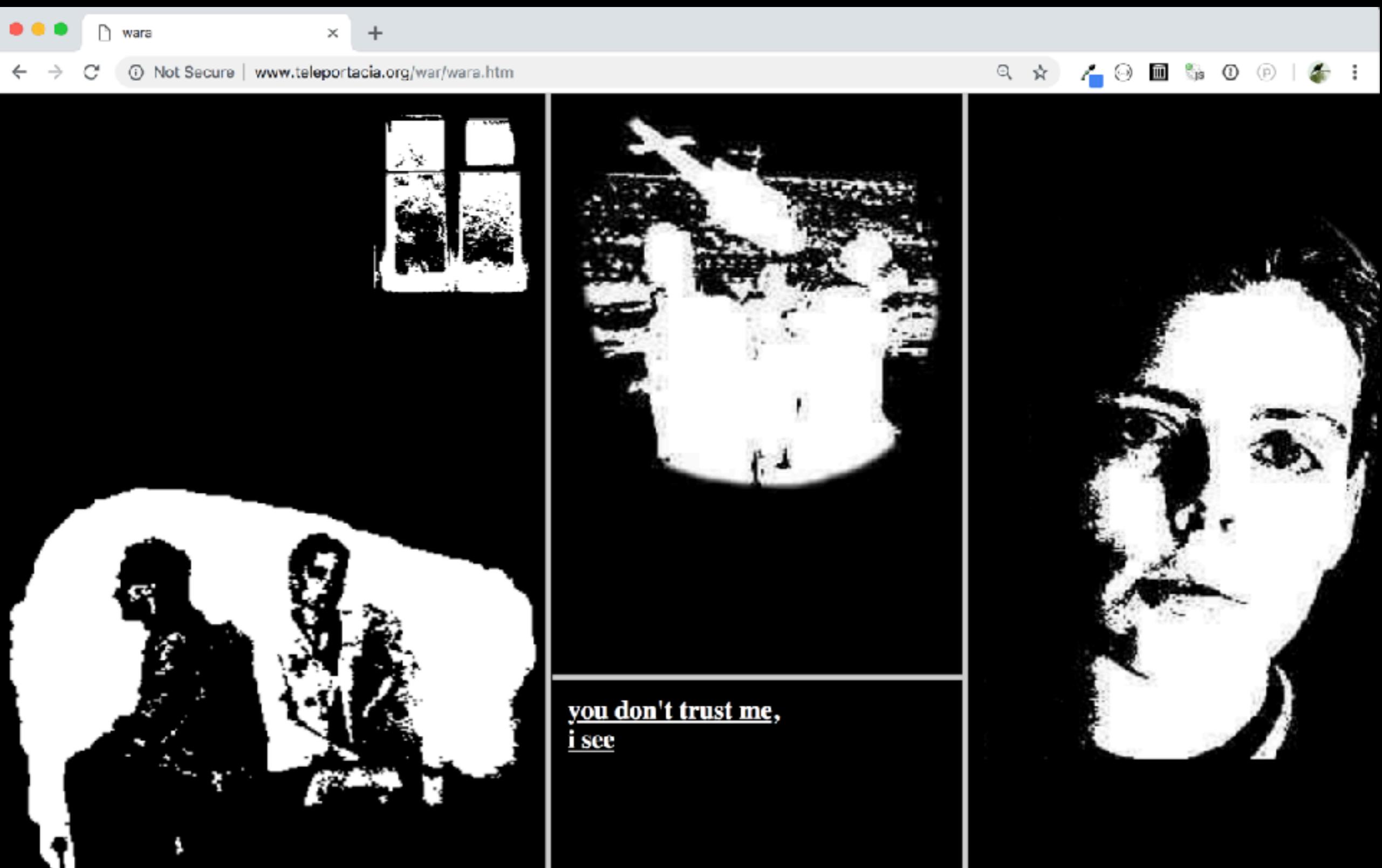


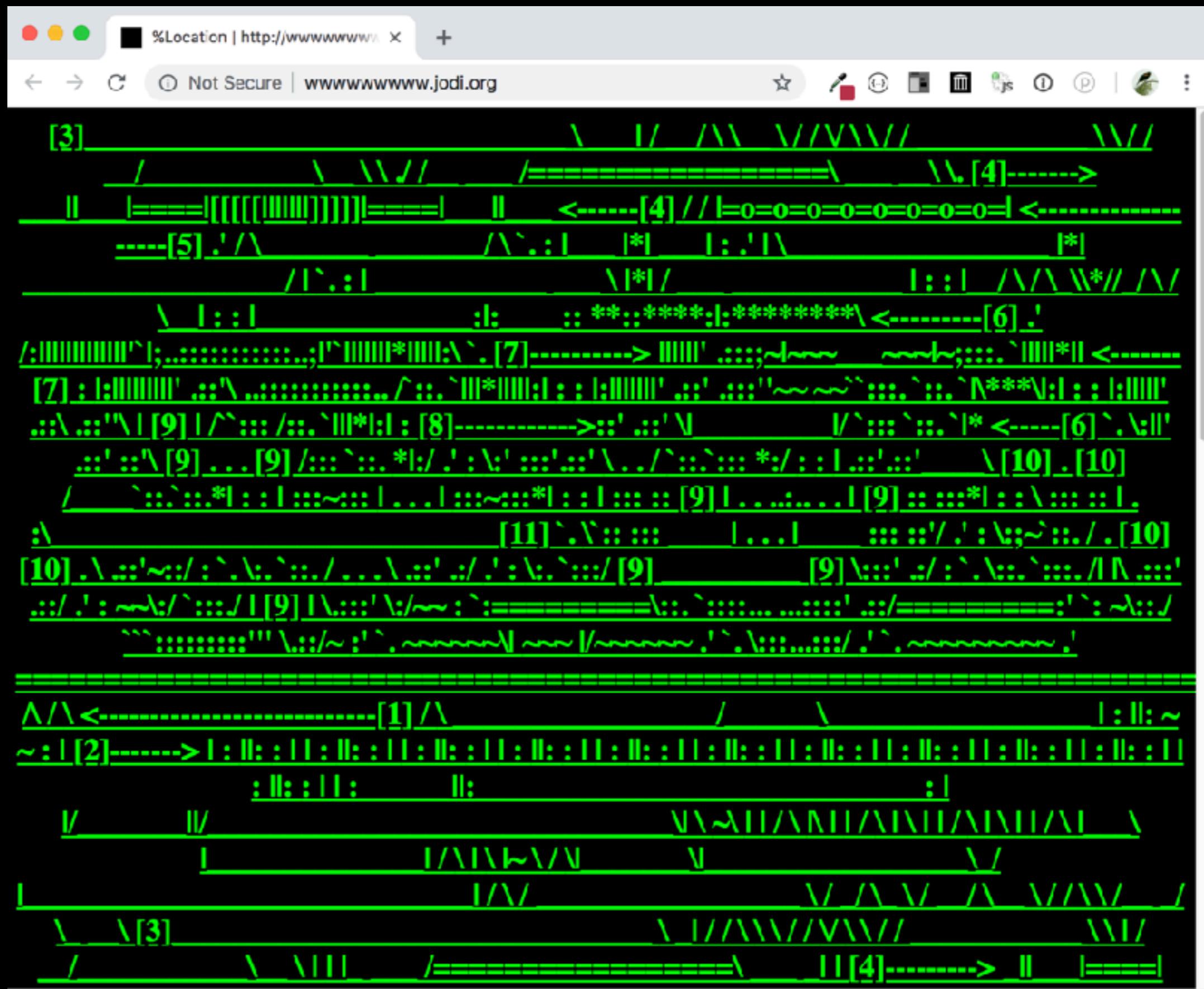
Simple Net Art Diagram





you don't trust me,
i see

My Boyfriend Came Back From War, 1996
Olia Lialina





The Whale Hunt / A storytelling X +

← → C ⓘ Not Secure | thewhalehunt.org/whalehunt.html



[The Whale Hunt](#), 2007
Jonathan Harris



89 Steps, 2014
Union Docs



Pool Poem, 2017
Dina Kelberman

index.html — _code UNREGISTERED

FOLDERS

_code

index.html

kanye.gif

theStyle.css

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title> Emerging Media 2: Week 2</title>
5     <!-- the following line of metadata
6       enables the browser to render emoji -->
7     <meta charset="utf-8">
8     <!-- <link rel="stylesheet" type="text/
9       css" href="theStyle.css"> -->
10
11   </head>
12   <body>
13     <h1>This Webpage though...</h1>
14     <h2>Is what the world is like with/ css</h2>
15     <p>
16       here is a paragraph of text. here is
17       a paragraph of text. here is a
18       paragraph of text. here is a
19       paragraph of text. <a href="
20         https://developer.mozilla.org/en-US/
21         docs/Web/CSS" target="_blank">here
```

week02 — (...) — 62x18

~/Desktop/teach/hntr/_s2019/week02 — (...) — python -m SimpleHTTPServer

```
15 Last login: Fri Feb 1 09:37:12 on console
16
17 <!
18   ~ cd /Users/Rebecca/Desktop/teach/hntr/_s2019/week02
19   week02 ls
20   th index.html
21   <P> week02 python -m SimpleHTTPServer
22   Serving HTTP on 0.0.0.0 port 8000 ...
23   127.0.0.1 - - [03/Feb/2019 17:56:31] "GET / HTTP/1.1" 200 -
24   127.0.0.1 - - [03/Feb/2019 17:56:31] code 404, message File no
25   t found
26   127.0.0.1 - - [03/Feb/2019 17:56:31] "GET /favicon.ico HTTP/1.
27   1" 404 -
28   127.0.0.1 - - [03/Feb/2019 18:01:16] "GET / HTTP/1.1" 200 -
29   127.0.0.1 - - [03/Feb/2019 18:01:16] "GET /favicon.ico HTTP/1.1" 200 -
```

Emerging Media 2: Week 2

localhost:8000/_code/

This Webpage though...

Is what the world is like with/ css

here is a paragraph of text. here is a paragraph of text. here is a paragraph of text. here is a sentence that is a link + some fine styling here is a paragraph of text. here is a paragraph of text. here is a paragraph of text.

this is a webpage of wonderful text

here is another paragraph of text. here is another paragraph of text.



Web Dev Workflow: Text Editor (Atom or Sublime), Local HTTP Server, Chrome

in Terminal we are speaking **Unix** :

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



```
week02 — (...) — 74x37
~/Desktop/teach/frdhm/_F_2018/infoWeb2018/week02 — (...) — bash
[ infoWeb2018     ls
 README.md       week01      week02
[ infoWeb2018     cd week02
[ week02        ls
 _assets         index.html
 week02        python -m SimpleHTTPServer
```

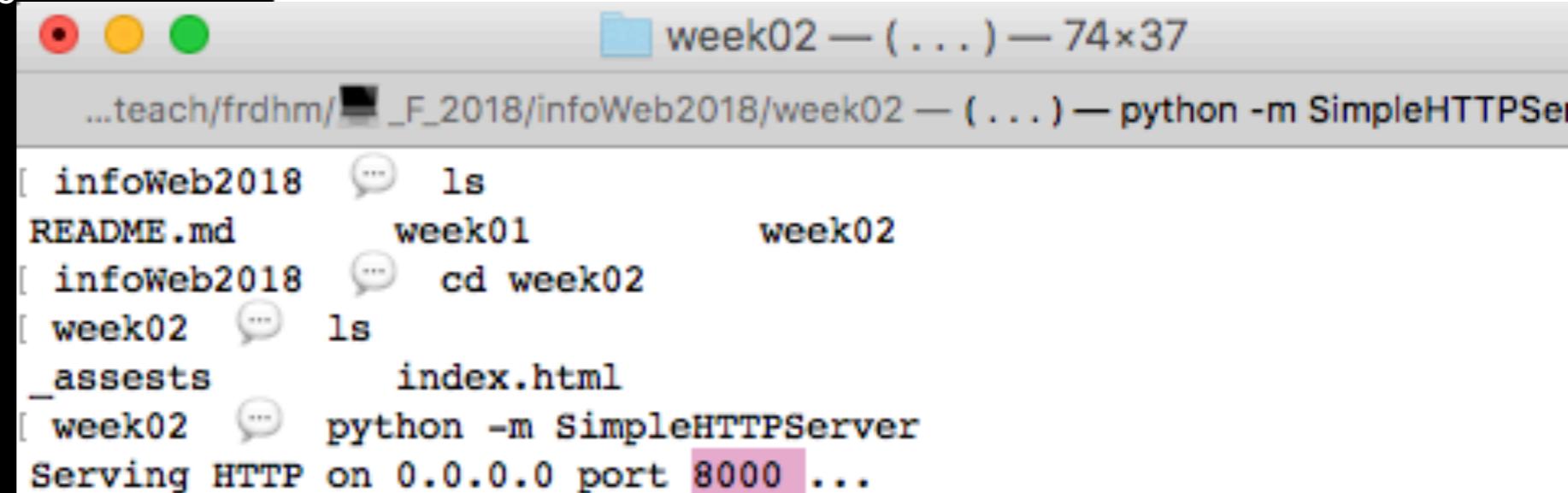
Running a local **Python** HTTP Server

in Mac OS - this is very simple :

When inside yr project folder simply
type the following command:

"python -m SimpleHTTPServer"

– defaults to port 8000



```
week02 — (...) — 74x37
...teach/frdhm/_F_2018/infoWeb2018/week02 — (...) — python -m SimpleHTTPServer
[ infoWeb2018     ls
 README.md       week01      week02
[ infoWeb2018     cd week02
[ week02        ls
 _assets         index.html
[ week02        python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
```

if we wrote:

"python -m SimpleHTTPServer 12345"

- we would go to port 12345

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`

Wnws - to close the server: `CNTRL C`

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

< HTML >

3 categories of html elements:

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

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

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

`<a>, , ,
`

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

``

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.

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 >

Hypertext Markup Langauge

Describes the **content + structure** of a web page;
NOT a programming language

< HTML >

3 categories of HTML elements

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

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

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

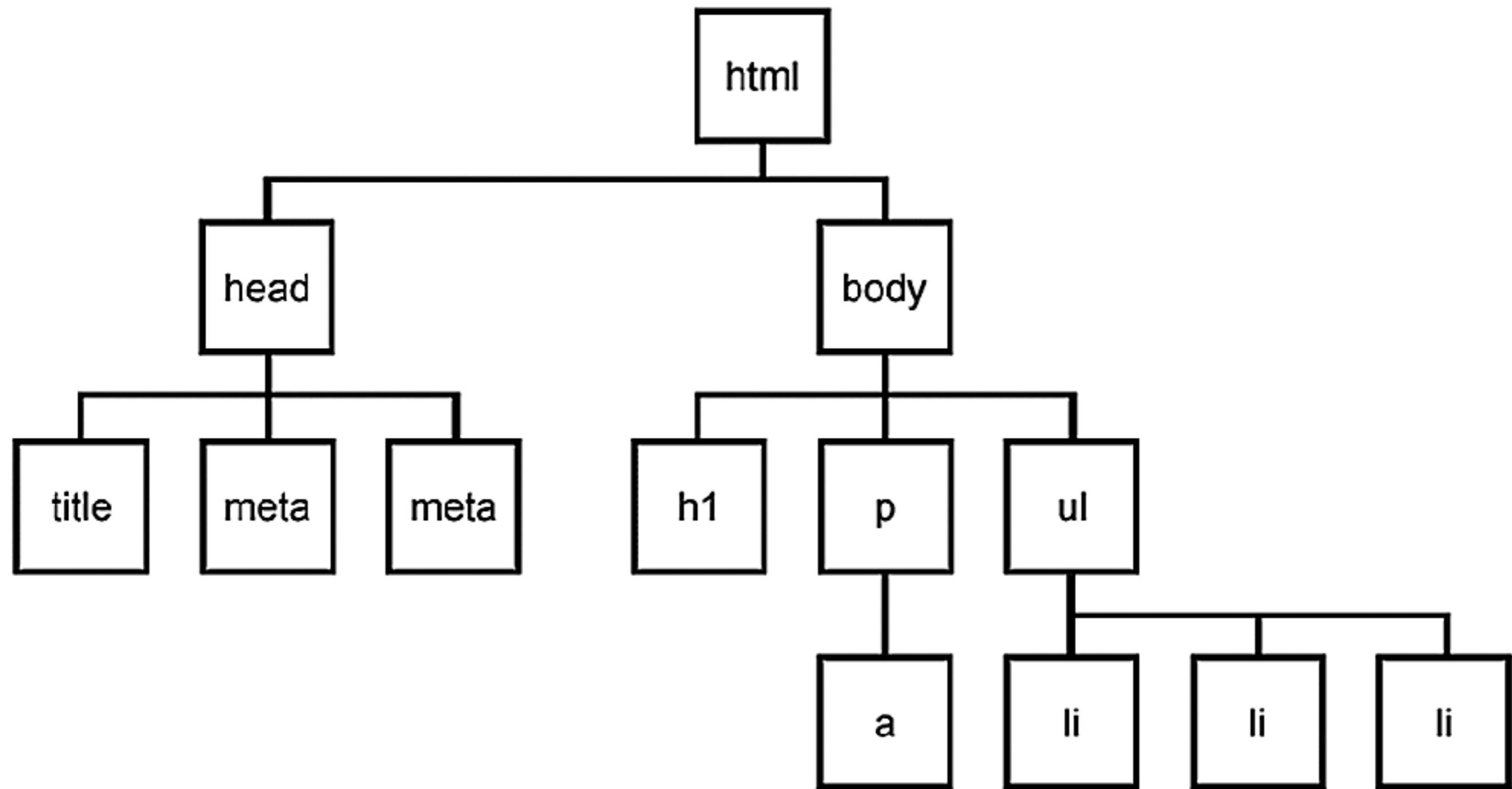
`<a>, , ,
`

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

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

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

Parent / Child Element Structure



HTML - Hyper Text Mark Up

```
<!DOCTYPE html>
<html>
  <head>
    <title>💻 Internet + Web Week 1</title>
  </head>
  <body>
    this is a webpage of wonderful text
  </body>
</html>
```

tag **attribute** **value**

< p id= "firstParagraph" > in between the tags is the content </p>

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

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

HTML Elements

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.

Structure tags

```
<!doctype html>
<html>
  <head>
    <title> Week 2 </title>
  </head>
  <body>
    <div>
      Here's a Great Site.
    </div>
  </body>
</html>
```

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

<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.