

Lecture 11

408/508 *Computational Techniques for Linguists*

Hypertext Transport Protocol (http)

404 Not Found

The HTTP **404 Not Found** [client error response](#) status code indicates that the server cannot find the requested resource. Links that lead to a 404 page are often called broken or dead links and can be subject to [link rot](#).

A 404 status code only indicates that the resource is missing without indicating if this is temporary or permanent. If a resource is permanently removed, servers should send the [410 Gone](#) status instead.

404 errors on a website can lead to a poor user experience for your visitors, so the number of broken links (internal and external) should be minimized to prevent frustration for readers. Common causes of 404 responses are mistyped URLs or pages that are moved or deleted without redirection. For more information, see the [Redirections in HTTP](#) guide.

Status

HTTP

Copy

404 Not Found

451 Unavailable For Legal Reasons

The HTTP **451 Unavailable For Legal Reasons** [client error response](#)

HTTP 451

Article [Talk](#)

From Wikipedia, the free encyclopedia

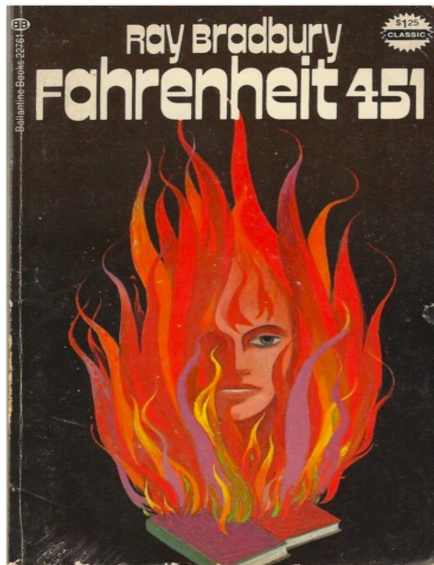
In [HTTP](#), **HTTP 451 Unavailable For Legal Reasons** is a [response status code](#) that indicates that a request cannot be satisfied for legal reasons, such as a web page censored by a government. The code value refers to [Ray Bradbury](#)'s 1953 *dystopian* novel *Fahrenheit 451*, in which books are outlawed.^[2] 451 provides more information than [HTTP 403](#), which is often used for the same purpose.^[3] This status code is currently a proposed standard in [RFC 7725](#), which updated the IANA HTTP Status Codes Registry to include 451.^[4]

Examples of situations where an 451 could be used include web pages deemed a danger to national security, or web pages deemed to violate copyright, privacy, [blasphemy laws](#), or any other law or court order.

Hypertext Transport Protocol (http)

Firemen of America

1953



"Established, 1790, to burn English-influenced books in the Colonies. First Fireman: Benjamin Franklin."

RULE 1. Answer the alarm swiftly.

2. Start the fire swiftly.

3. Burn everything.

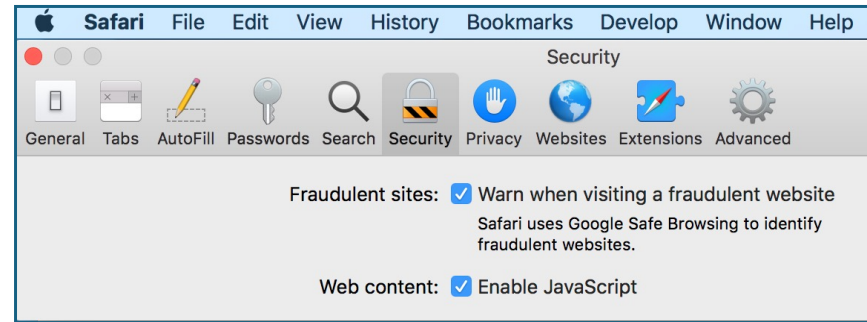
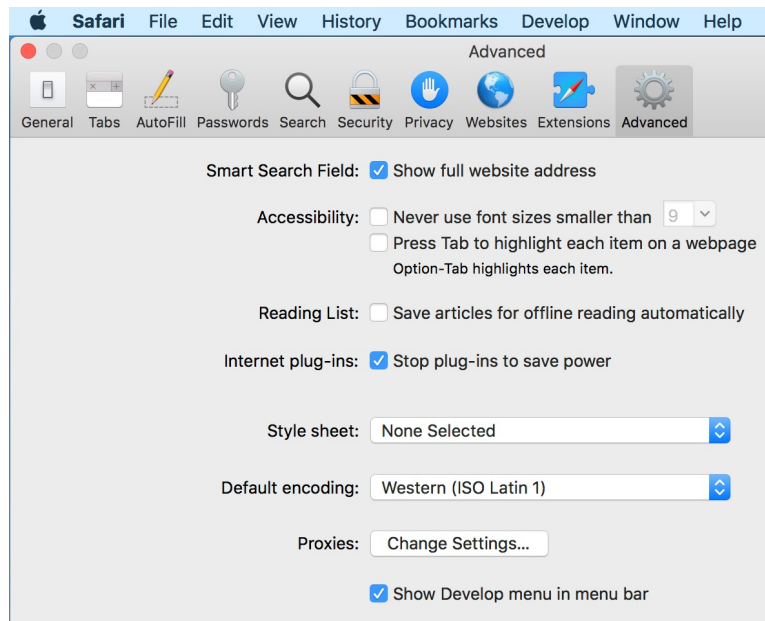
4. Report back to firehouse immediately.

5. Stand alert for other alarms.

FAHRENHEIT 451—
*the temperature at which book-paper
catches fire, and burns . . .*

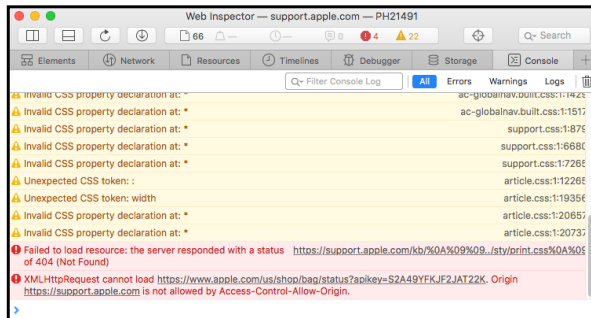
Debugging

- In Safari:
 - Preferences > Advanced > **Show Develop menu in menu bar**



Debugging

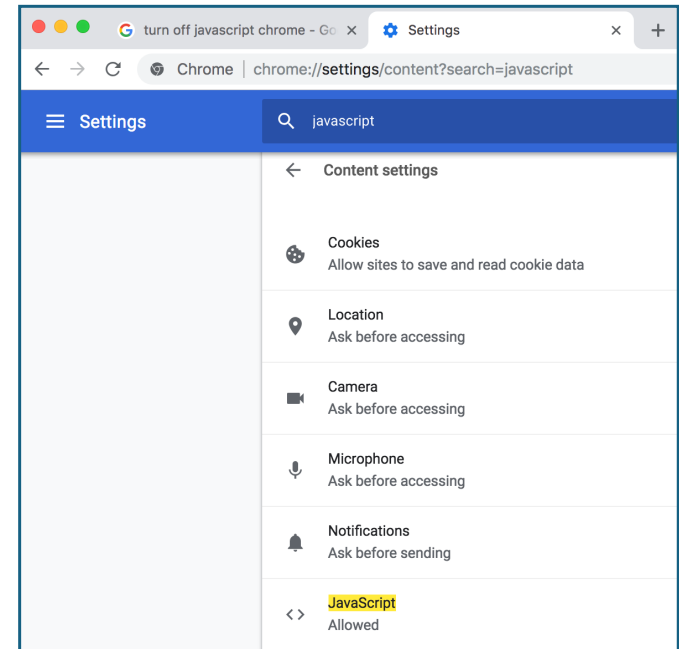
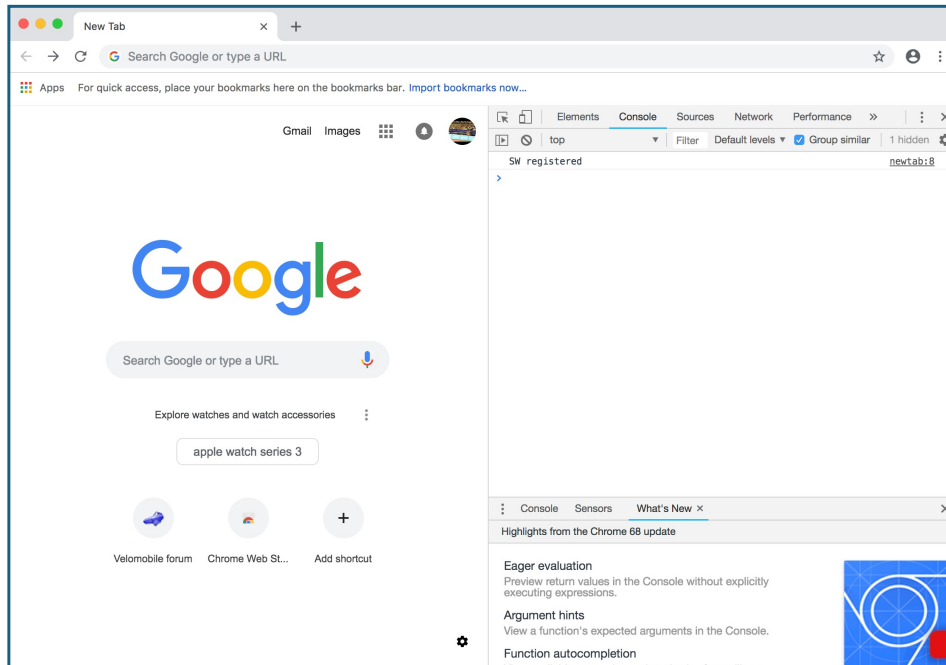
- In the Develop menu:
 - *Allow JavaScript from Smart Search Field*
 - **javascript:** ok in address bar
 - Show Page Source
 - Show Web Inspector



Develop		Window	Help
Open Page With User Agent			
Hide Web Inspector		⌘⇧⌘I	
Show Error Console		⌘⇧⌘C	
Show Page Source		⌘⇧⌘U	
Show Page Resources		⌘⇧⌘A	
Show Snippet Editor			
Show Extension Builder			
Start Timeline Recording		⌘⇧⌘T	
Empty Caches		⌘⇧⌘E	
Disable Caches			
Disable Images			
Disable Styles			
Disable JavaScript			
Disable Site-specific Hacks			
Disable Local File Restrictions			
Enable WebGL			
Allow JavaScript from Smart Search Field			

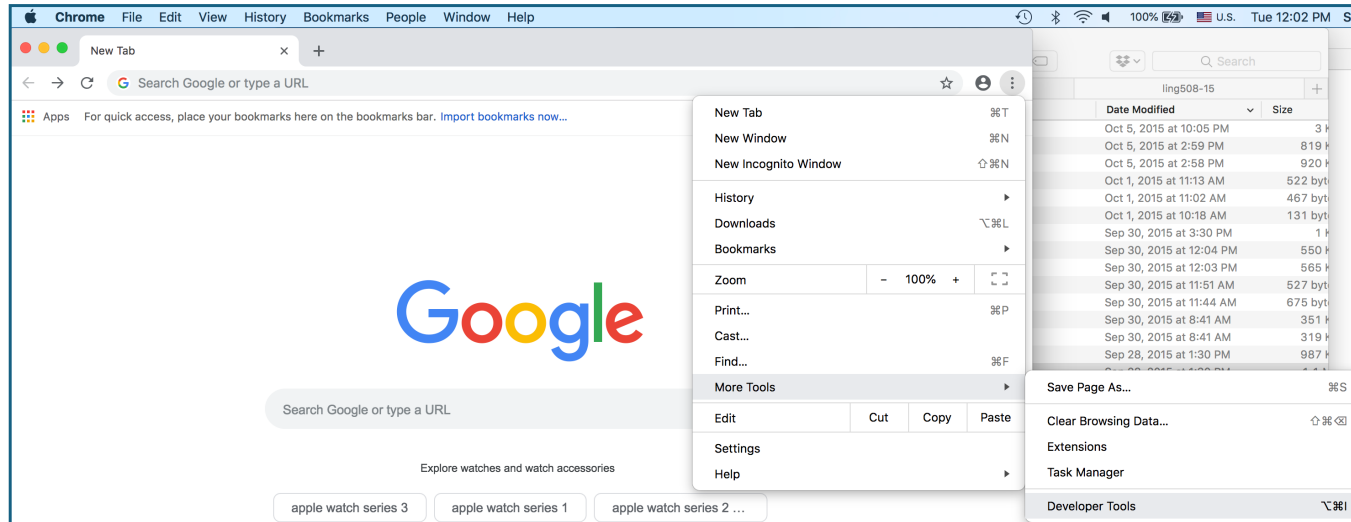
Debugging

- Developer Tools and Javascript in Google Chrome:



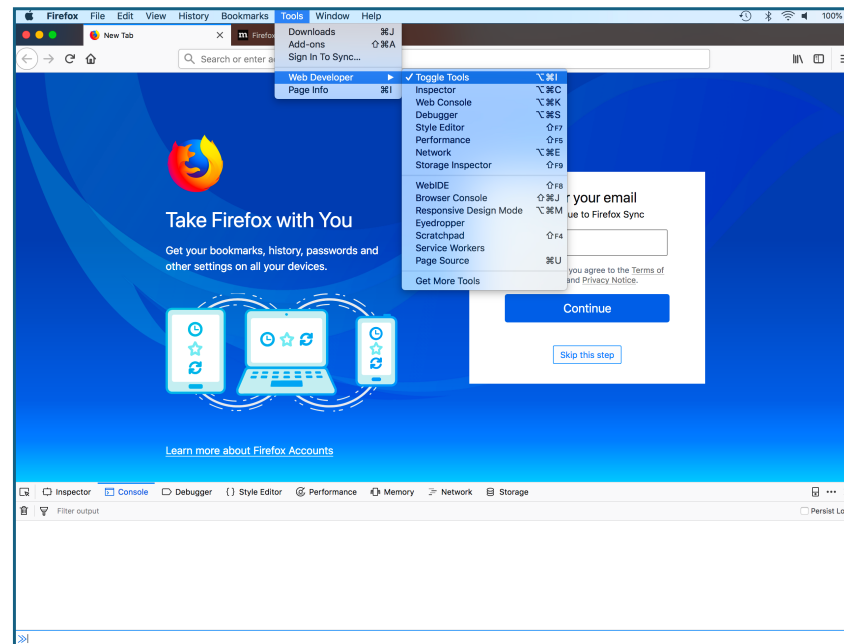
Debugging

- In Google Chrome:



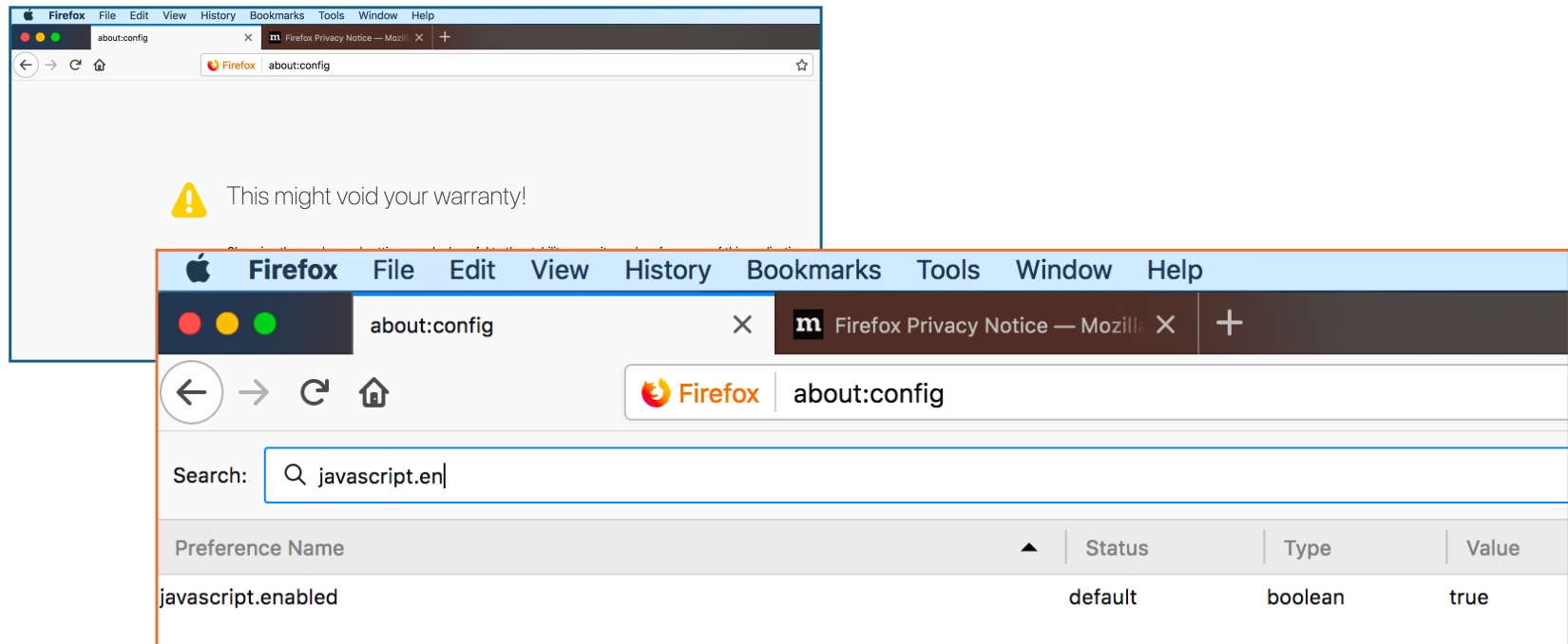
Debugging

- Firefox:



Debugging

- Firefox:



Example

- See example **source** for =>

508 Demo Page



This is the *first* paragraph of my demo page.

2nd paragraph

My webpage is at [here](#)

```
$ base64 -i airbnb.png -o encoded.txt
```

- One
- Two
- Three

A
A A A
A A A

Last modified: Tue Sep 30 15:19:01 MST 2025

HTML

- inline styling applied to text elements:
- e.g. `<span>` (*inline text*) vs. `<div>` (*block text*)
 - <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/span>
 - `<span style="...; ..."> ... </span>`
 - style attributes:
 - font-size: Xpx (px = pixel)
 - font-family: name, name ...
 - color: name (or hex RGB) e.g. #00CC00 for hex
 - background-color: name (or RGB)
 - text-align: left|right|center
 - **note:** serif, sans-serif, monospace are generic font families
 - **note:** X11 color names are okay,
http://en.wikipedia.org/wiki/X11_color_names

X11 Colors

- https://en.wikipedia.org/wiki/X11_color_names#Color_name_chart

	Name ↕	Hex (RGB) ↕	Red (RGB) ↕	Green (RGB) ↕	Blue (RGB) ↕	Hue (HSL/HSV) ↕	Satur. (HSL) ↕	Light (HSL) ↕	Satur. (HSV) ↕	Value (HSV) ↕
	Alice Blue	#F0F8FF	94%	97%	100%	208°	100%	97%	6%	100%
	Antique White	#FAEBD7	98%	92%	84%	34°	78%	91%	14%	98%
	Aqua	#00FFFF	0%	100%	100%	180°	100%	50%	100%	100%
	Aquamarine	#7FFFD4	50%	100%	83%	160°	100%	75%	50%	100%
	Azure	#F0FFFF	94%	100%	100%	180°	100%	97%	6%	100%
	Beige	#F5F5DC	96%	96%	86%	60°	56%	91%	10%	96%
	Bisque	#FFE4C4	100%	89%	77%	33°	100%	88%	23%	100%
	Black	#000000	0%	0%	0%	0°	0%	0%	0%	0%
	Blanched Almond	#FFEBCD	100%	92%	80%	36°	100%	90%	20%	100%
	Blue	#0000FF	0%	0%	100%	240°	100%	50%	100%	100%
	Blue Violet	#8A2BE2	54%	17%	89%	271°	76%	53%	81%	89%
	Brown	#A52A2A	65%	16%	16%	0°	59%	41%	75%	65%
	Burlywood	#DEB887	87%	72%	53%	34°	57%	70%	39%	87%

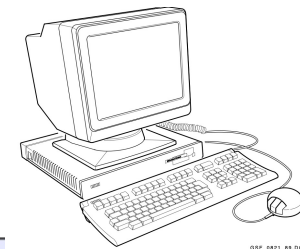
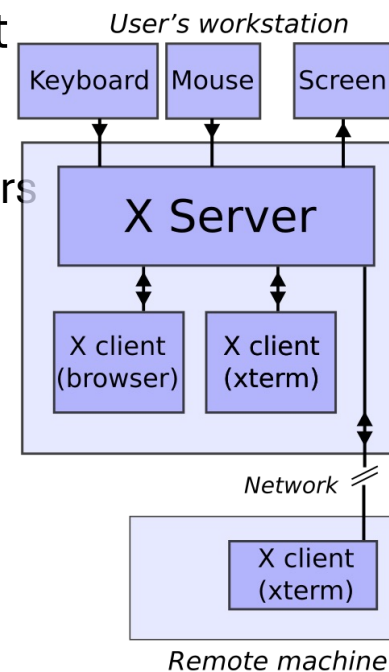
What is X11?

- **X Window System: Project Athena** at MIT (1983)
 - Distributed computing with graphics terminals connected to bigger computers
 - stuck at version 11
 - On Linux, X11 is still in operation

From: rws@mit-bold (Robert W. Scheifler)
To: window@athena
Subject: window system X
Date: 19 June 1984 0907-EDT (Tuesday)

I've spent the last couple weeks writing a window system for the VS100. I stole a fair amount of code from W, surrounded it with an asynchronous rather than a synchronous interface, and called it X. Overall performance appears to be about twice that of W. The code seems fairly solid at this point, although there are still some deficiencies to be fixed up.

We at LCS have stopped using W, and are now actively building applications on X. Anyone else using W should seriously consider switching. This is not the ultimate window system, but I believe it is a good starting point for experimentation. Right at the moment



Originally,
X terminal
was a real thing.
Now a PC/mac



UTF-8

```
<head>
...
<meta charset="utf-8">
...
</head>
```

- The <meta> [HTML](#) element represents [metadata](#) that cannot be represented by other HTML meta-related elements, like [<base>](#), [<link>](#), [<script>](#), [<style>](#) or [<title>](#).
- [charset](#)
 - This attribute declares the document's character encoding. If the attribute is present, its value must be an ASCII case-insensitive match for the string "utf-8", because **UTF-8 is the only valid encoding for HTML5** documents. <meta> elements which declare a character encoding must be located entirely within the first 1024 bytes of the document.

HTML

https://developer.mozilla.org/en-US/docs/Web/HTML/Inline_elements

HTML (**H**ypertext **M**arkup **L**anguage) elements historically were categorized as either "block-level" elements or "inline-level" elements. Since this is a presentational characteristic it is nowadays specified by CSS in the [Flow Layout](#). Inline elements are those which only occupy the space bounded by the tags defining the element, instead of breaking the flow of the content. In this article, we'll examine HTML inline-level elements and how they differ from [block-level elements](#).



An inline element does not start on a new line and only takes up as much width as necessary.

HTML

Conceptual differences

In brief, here are the basic conceptual differences between inline and block-level elements:

Content model

Generally, inline elements may contain only data and other inline elements. You can't put block elements inside inline elements.

Formatting

By default, inline elements do not force a new line to begin in the document flow. Block elements, on the other hand, typically cause a line break to occur (although, as usual, this can be changed using CSS).

- Block:
 - lists
 - headings:
 - `<h1> ... <h4>`
 - `<p>`
 - `<div>`
 - general container

HTML

- **Tables:**

- `<table> ... </table>`
- `<tr> ... </tr>`
- `<th> ... </th>`
- `<td> .. </td>`
- **Attributes:**
 - `border="size"`
 - `colspan="number"`
 - `style="...;..."`
 - `border: width style color`
 - `border-width: top right bottom left`
 - `border-style: top right bottom left`
 - `border-color: top right bottom left`
 - `border-collapse: collapse|separate`
 - `padding: size`
 - `text-align: left|center|right`
 - `vertical-align: top|middle|bottom`
 - `width, height`
- **Newer stuff:**
 - `<thead> ... </thead>`
 - `<tbody> .. </tbody>`

table row

table heading element

table data (one cell)

e.g. 1px

e.g. 2 (span next two columns)

also border-left, border-top, border-right, etc.

e.g. type=solid|dotted|dashed|double|none

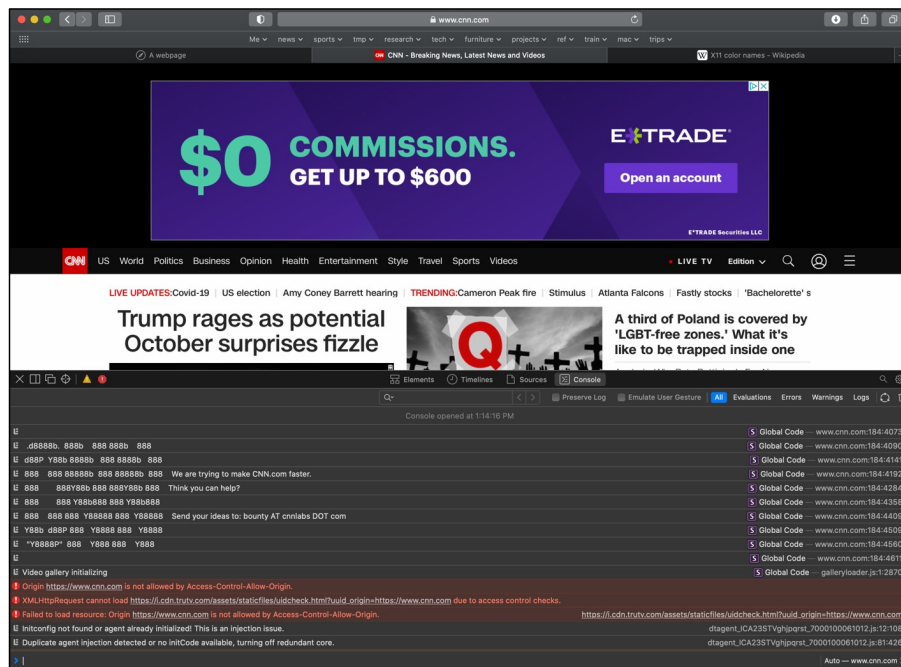
e.g. 100px or 100%

HTML

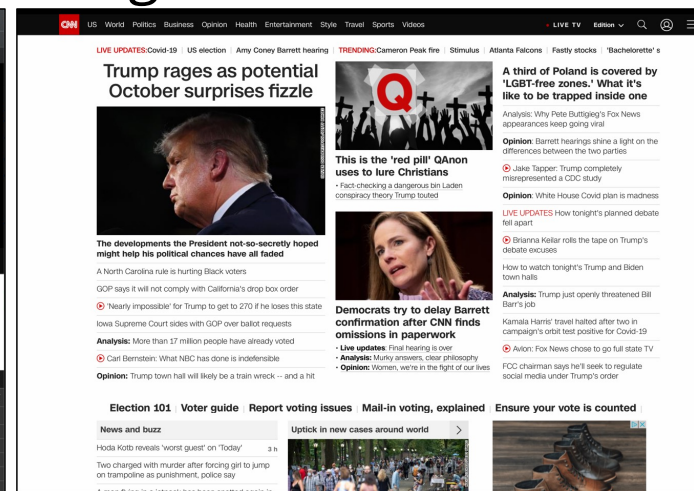
- General **chunks** of html:
 - `<div style="..."> ... </div>`
 - a division (block-level)
 - `<span style="..."> ... </span>`
 - small chunks (inline)
- (optional) **unique identifier** `id=`
 - used to refer to the "chunk" in CSS or DOM
 - `<div style="..." id="..."> ... </div>` a **division**
 - `<span style="..." id="..."> ... </span>` a **span** for inline text

CNN Webpage

Look at the errors on loading cnn.com!



Look at the bounty offered to developers! ⬆



CSS

```
body {  
    background-color: lightblue;  
}  
  
h1 {  
    color: white;  
    text-align: center;  
}  
  
p {  
    font-family: verdana;  
    font-size: 20px;  
}
```

Selector	Example	Example description
<u>.class</u>	.intro	Selects all elements with class="intro"
<u>#id</u>	#firstname	Selects the element with id="firstname"
<u>*</u>	*	Selects all elements
<u>element</u>	p	Selects all <p> elements
<u>element,element</u>	div, p	Selects all <div> elements and all <p> elements
<u>element element</u>	div p	Selects all <p> elements inside <div> elements
<u>element>element</u>	div > p	Selects all <p> elements where the parent is a <div> element

<https://www.w3schools.com/css/>