

CSI 3140 - LECTURE 02

HTML

OBJECTIVES

- ▶ Textbook Chapter 2 and 3
- ▶ Introduction to HTML
- ▶ Making a “web page”
- ▶ Adding images,
 - ▶ hyperlinks,
 - ▶ lists
 - ▶ tables
 - ▶ Forms / data entry

LATEX

HTML



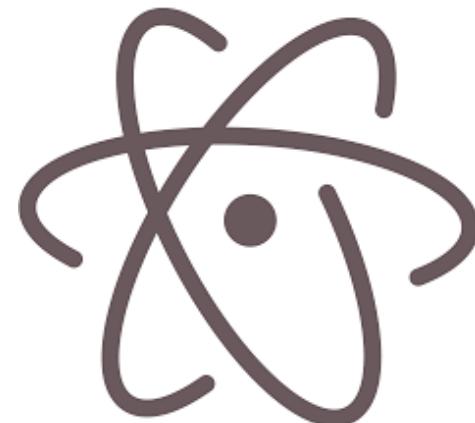
EXAMPLE MARKUP LANGUAGES

Abbr	Nom
LaTeX	Lamport Tex
XML	Extensible Markup Language
SGML	Standard Generalized Markup Language
HTML	Hypertext Markup Language
HAML	HTML Abstraction Markup Language
EEx	Embedded Elixir (HTML + Elixir)
Erb	Embedded Ruby (HTML + Ruby)

```
<h3>
    Prerequisites:
</h3>
<p>CSI2110, CSI2132</p>

<h3>Professeur</h3>
<table>
    <tr>
        <th>Professor</th>
        <td>Andrew Forward</td>
        <t
</t>table          Tag
<t>tbody          Tag
    <td          Tag
        <template      Tag
        <textarea      Tag
    </td>tbody          Tag
<t>tfoot          Tag
    <th          Tag
    <thead      Tag
        <a href="https://uottawa.brightspace.com/d2l/le/cont
            </td>
        </tr>
    <tr>
```

EXAMPLE TEXT EDITORS



BOILERPLATE HTML

```
01_boilerplate.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Welcome</title>
6   </head>
7   <body>
8     <p>Welcome to HTML5!</p>
9   </body>
10 </html>
```

Declaring "Document Type" required

Meta-data about the document

Empty tags without an "closing" tag

Double, not single quotes

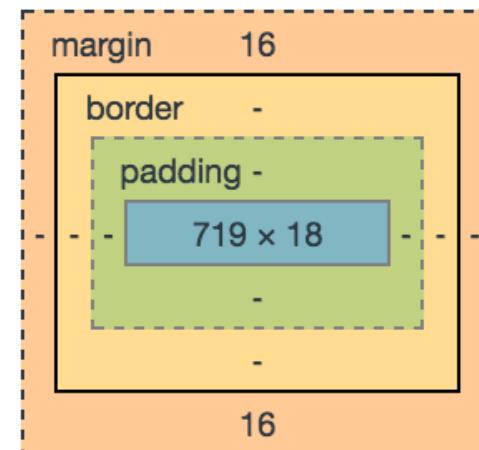
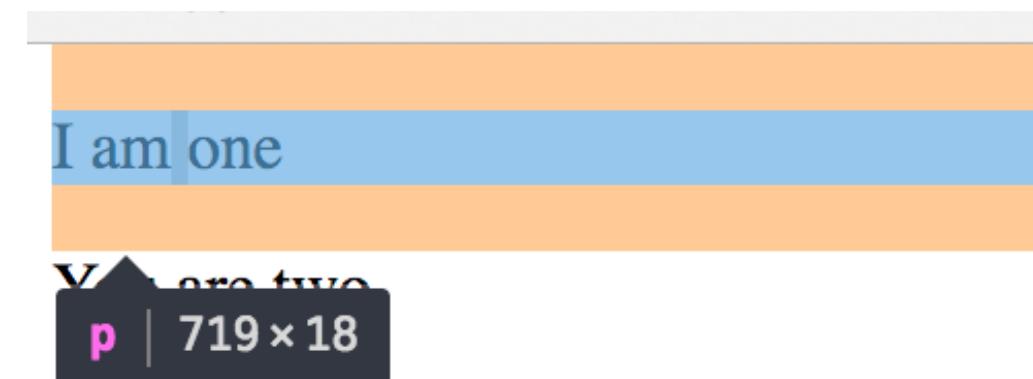
The content (aka body) of the document

Like XML usually tags need a start and a end

Tags should be lowercase

<P>PARAGRAPH</P>

```
    <body>
        <p>I am one</p>
        <p>You are two</p>
    </body>
```



VALIDATING HTML

<http://validator.w3.org/#validate-by-upload>

Validate by URI Validate by File Upload Validate by Direct Input

Validate by File Upload

Upload a document for validation:

File: 02_paragraph.html

► More Options

Check

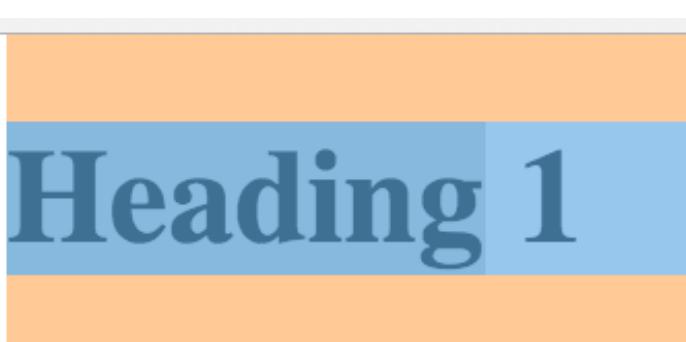
Document checking completed. No errors or warnings to show.

Used the HTML parser.

Total execution time 2 milliseconds.

<H1>HEADINGS</H1> ... <H6></H6>

```
</head>
<body>
  <h1>Heading 1</h1>
  <h2>Heading 2</h2>
  <h3>Heading 3</h3>
  <h4>Heading 4</h4>
  <h5>Heading 5</h5>
  <h6>Heading 6</h6>
  <h7>Heading 7</h7> <!-- Not a valid tag -->
</body>
</html>
```



Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

Heading 7

<A>HYPERLINKS (OR ANCHORS)

```
<body>
  <a href="http://a4word.com">a4word</a><br>
  <a href="https://google.ca">Google Search</a><br>
  <a href="mailto:aforward@gmail.com">Email me</a><br>
  <a href="01_boilerplate.html">Boilerplate</a><br>
  <a href="//01_boilerplate.html">Boilerplate (Absolute)</a> <!-- invalid -->
```

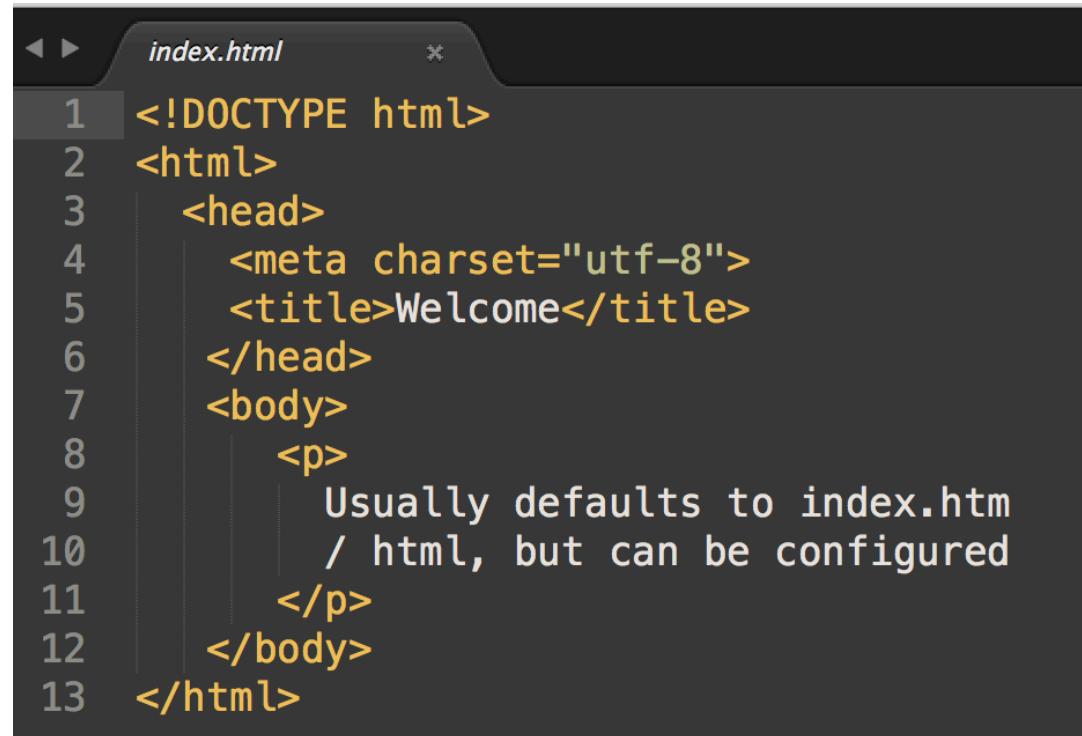
Relative link
Absolute link
Maintain same protocol
(i.e. https stays https)

External link

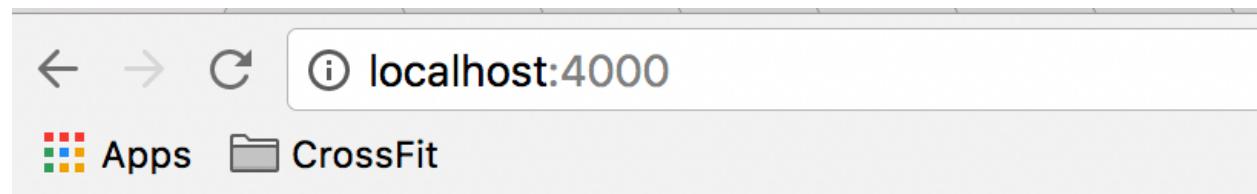
a4word
Google Search
Email me
Boilerplate
Boilerplate (Absolute)

Hypertext reference

DOCUMENT DOCUMENT (INDEX.HTM)



```
index.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8">
5     <title>Welcome</title>
6   </head>
7   <body>
8     <p>
9       Usually defaults to index.htm
10      / html, but can be configured
11    </p>
12  </body>
13 </html>
```



Usually defaults to index.htm / html, but can be configured

 IMAGES

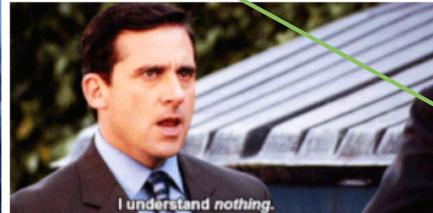
```
<style>
|   .skinny {width: 200px;}
</style>
</head>
<body>
|   
|   
|   
|   
</body>
```

"source" (src) not href

Attribute for accessibility

Named style
(i.e. a class)

Inline style



ACCESSIBILITY

- ▶ A browser may not be able to render the image
- ▶ The alt tag can also be read to the user
- ▶ ARIA (Accessible Rich Internet Applications Suite) as an alternative to pure HTML

TEXT + STYLE > IMAGES

```
<br />
<a href="http://a4word.com" class="button">My Fancy Button</a>
</body>
```



My Fancy Button

		
	!	"	#	\$	%	&	'
		!	" "	#	$	%	& &	'
()	*	+	,	.	/	:	;
()	* *	+	,	.	/	:	;
<	=	>	?	@	[\]	^
< <	=	> >	?	@	[&lbrcak;	\]]	^
—	`	{		}	 	¡	¢	£
_	` `	&lcurb; &brace;	| | |	&rcurb; }				

<HR> HORIZONTAL RULE

```
<style>
  .divider {
    border-top: 1px solid black;
    padding-top: 10px;
  }
</style>
</head>
<body>
  <p>Above</p>
  <hr>
  <p>Below</p>
  <br />
  <p>Better Above</p>
  <p class="divider">Better Below</p>
</body>
```

Above

Below

Better Above

Better Below

LIST ITEMS

```
<ul>
  <li>Apples</li>
  <li>Bananas</li>
  <li>Carrots</li>
</ul>

<ol>
  <li>Study</li>
  <li>Write Exam</li>
  <li>Graduate</li>
</ol>
</body>
```

- Apples
 - Bananas
 - Carrots
-
1. Study
 2. Write Exam
 3. Graduate

<TABLE></TABLE>

```
<table>
  <thead>
    <tr>
      <th>Name</th>
      <th>Grade</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>Andrew</td>
      <td>92%</td>
    </tr>
    <tr>
      <td>James</td>
      <td>78%</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th>Average</th>
      <th>85%</th>
    </tr>
  </tfoot>
</table>
```

Header information

Body of the table

Rows (table-row)

Cell (table-data)

Footer information

Header cell (table-header)

Name	Grade
Andrew	92%
James	78%
Average	85%

ROWSPAN / COLSPAN

```
<table>
<thead>
  <tr>
    <th colspan="2">Name</th>
    <th>&nbsp</th>
  </tr>
  <tr>
    <th>First</th>
    <th>Last</th>
    <th>Grade</th>
  </tr>
</thead>
<tbody>
  <tr>
    <td rowspan="2">Andrew</td>
    <td rowspan="2">Forward</td>
    <td>92%</td>
  </tr>
  <tr>
    <td>78%</td>
  </tr>
</tbody>
<tfoot>
  <tr>
    <th colspan="2">Average</th>
    <th>85%</th>
  </tr>
</tfoot>
</table>
```

```
<style>
  table { border-collapse: collapse; }
  table td, table th { border: 1px solid black; padding: 10px; }
  table thead tr { background-color: #bdc3c7; }
  table tfoot tr { background-color: #ecf0f1; }
</style>
<table>
```

Name		
First	Last	Grade
Andrew	Forward	92%
		78%
Average		85%

<FORM>

Where should we send the data
(defaults to itself)

The "id" is for the client (JavaScript)

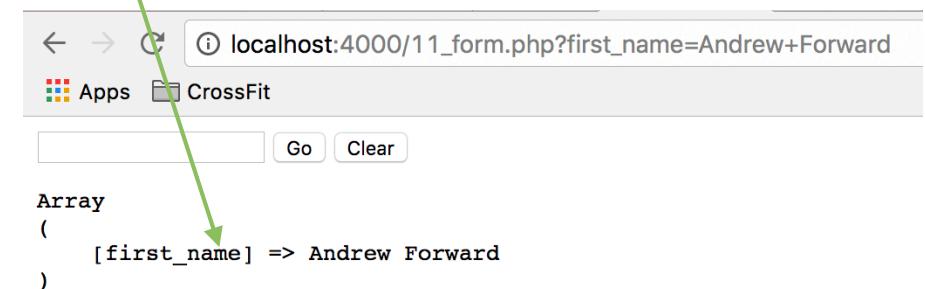
```
<body>
  <form method="get" action="process.php">
    <input type="text" name="first_name" id="firstName" />
    <input type="submit" value="Go" />
    <input type="reset" value="Clear" /> <!-- rarely used -->
  </form>
</body>
```

"get" et "post" supported

"Submit" the form

Don't use

The "name" is for the server



REQUEST VARIABLES

```
<body>
<?php
if (!empty($_POST)) {
    ?>
    <h1>Hello <?php echo $_POST["first_name"] ?></h1>
    <a href="11b_form.php">Try Again</a>
    ?>
} else {
    ?>
    <form method="post">
        <input type="text" name="first_name" id="firstName" />
        <input type="submit" value="Go" />
    </form>
    ?>
}
?>
```

<INPUT TYPE="X">

```
<input type="text" name="my_name" placeholder="Enter name">  
<input type="email" name="my_email" placeholder="Enter email">  
<input type="password" name="my_password" placeholder="Password">
```

Mask inputs on the screen

Great usability attribute
when the field is empty

Enter name

aforward@gmail.com

••••••••••••

HIDDEN

- ▶ Can store data without being displayed
- ▶ Sent in plain-text by the client
- ▶ No security, more convenience
- ▶ <input type = "hidden" name = "_method" value = "put">



A trick by web frameworks to
support all HTTP verbs

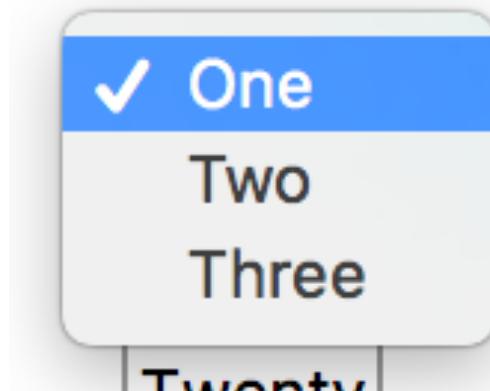
SELECT

```
<select name="my_select">  
    <option value="1">One</option>  
    <option value="2">Two</option>  
    <option>Three</option>  
</select>
```

Only one value sent to the server

The "name" of the option

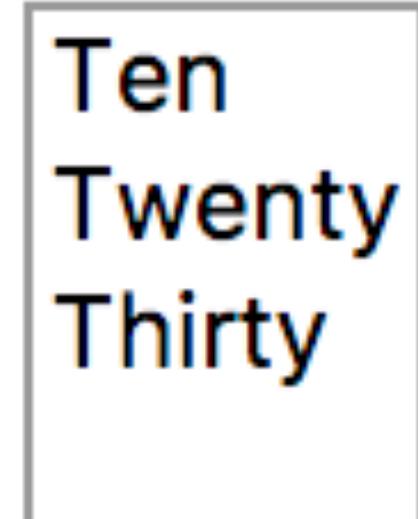
If not value attribute, use the name



SELECT MULTIPLE

Note the square brackets in the name

```
<select multiple="multiple" name="my_many_select[]">  
  <option>Ten</option>  
  <option>Twenty</option>  
  <option>Thirty</option>  
</select>
```



LABEL

```
<label for="firstName">Enter your first name</label>
```

```
<input type="text" name="first_name" id="firstName" placeholder="Your name" />
```

Associate the "label" with the input for improved usability

The labels increase the clickable area for the checkbox (more improved usability)

Enter your first name

Your name



Option 1



Option 2



Check 1



Check 2

FRAGMENTS

```
<div class="block" id="myA" >A  
  <a href="#myB">to B</a>  
</div>  
  
<div class="block" id="myB" >B  
  <a href="#myA">to A</a>  
</div>
```

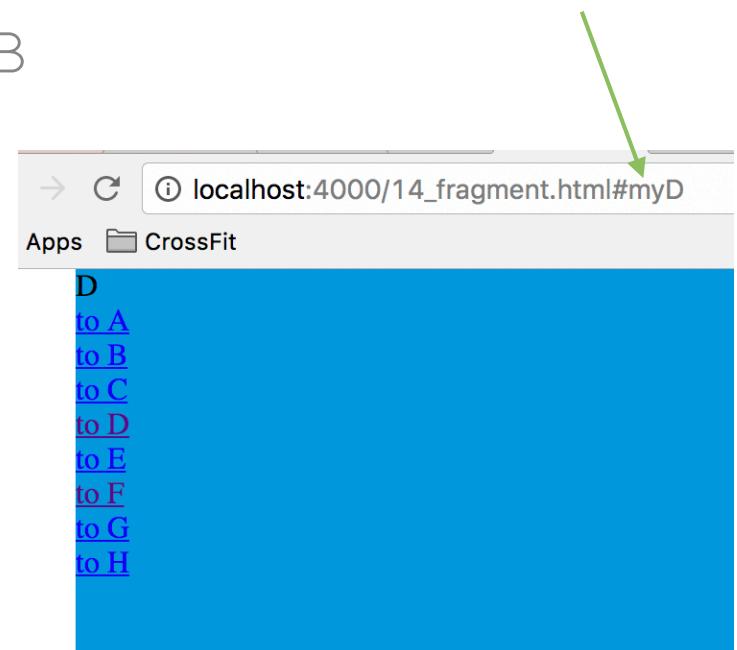
Fragments not sent to the server

The anchors (i.e. fragments) can be used for in-page navigation

Called a fragment

Fragments allow for deep links into a page

Careful, they are heavily used in client side framework



META

- ▶ Provides useful advice to search engines on the nature of your page
- ▶ Search Engine Optimization (SEO) could be an entire course
- ▶ The rules are constantly changing because once you know how you are assessed, some (all?) will game the system
 - ▶ Example, keywords are no longer relevant
 - ▶ Page size and loading speed are now relevant
- ▶ Before Google, Search Engines Were Nothing More Than Web Directories
 - ▶ This is where Yahoo shined (with Altavista), but are now missing or irrelevant

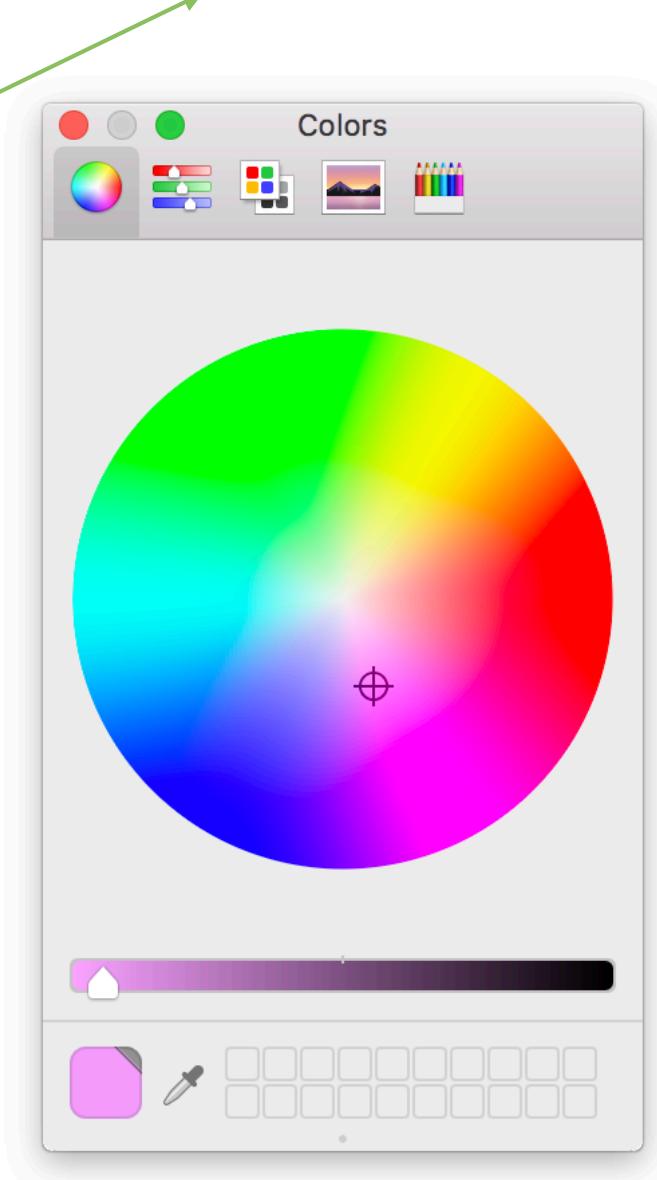
EXAMPLE OF META DATA

```
<head>
  <meta charset="utf-8">
  <title>Examples of relevant metatags</title>
  <meta name="description" content="An introduction to meta tags">
  <meta name="author" content="Andrew Forward">
  <!-- obsolete for SEO -->
  <meta name="keywords" content="HTML,How-To,MetaTags">
  <link rel="author" href="[YOUR PERSONAL PAGE]">
</head>
<body>
  <meta property="og:title" content="describe the block of text"/>
  <meta property="og:type" content="article"/>
  <p>Welcome to Meta</p>
</body>
</html>
```

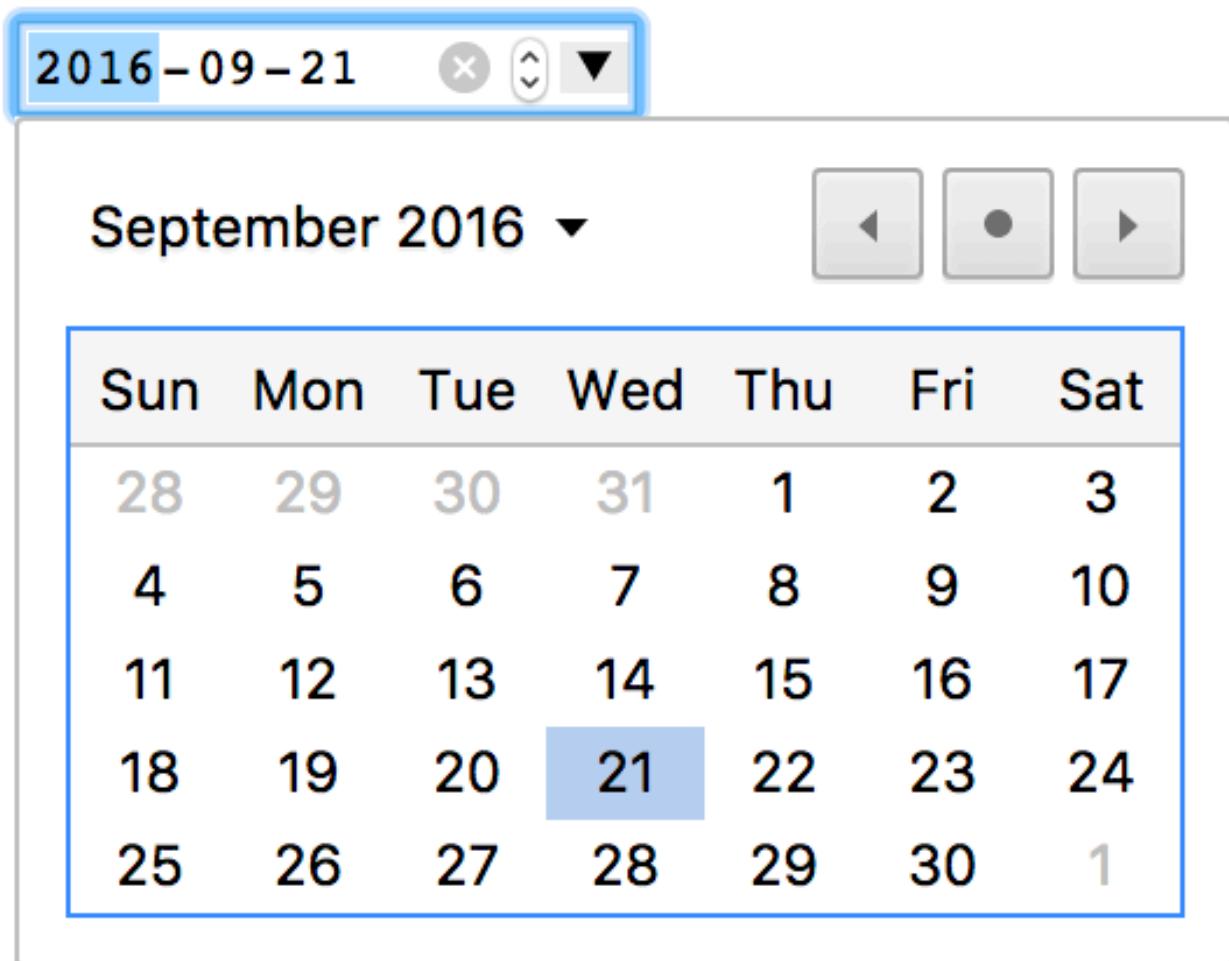
```
<input type="color" name="my_colour" id="myColour"  
value="#f39efb">
```

Code HEX

Just showing two naming styles of names
(server side use) and ids (client side use).



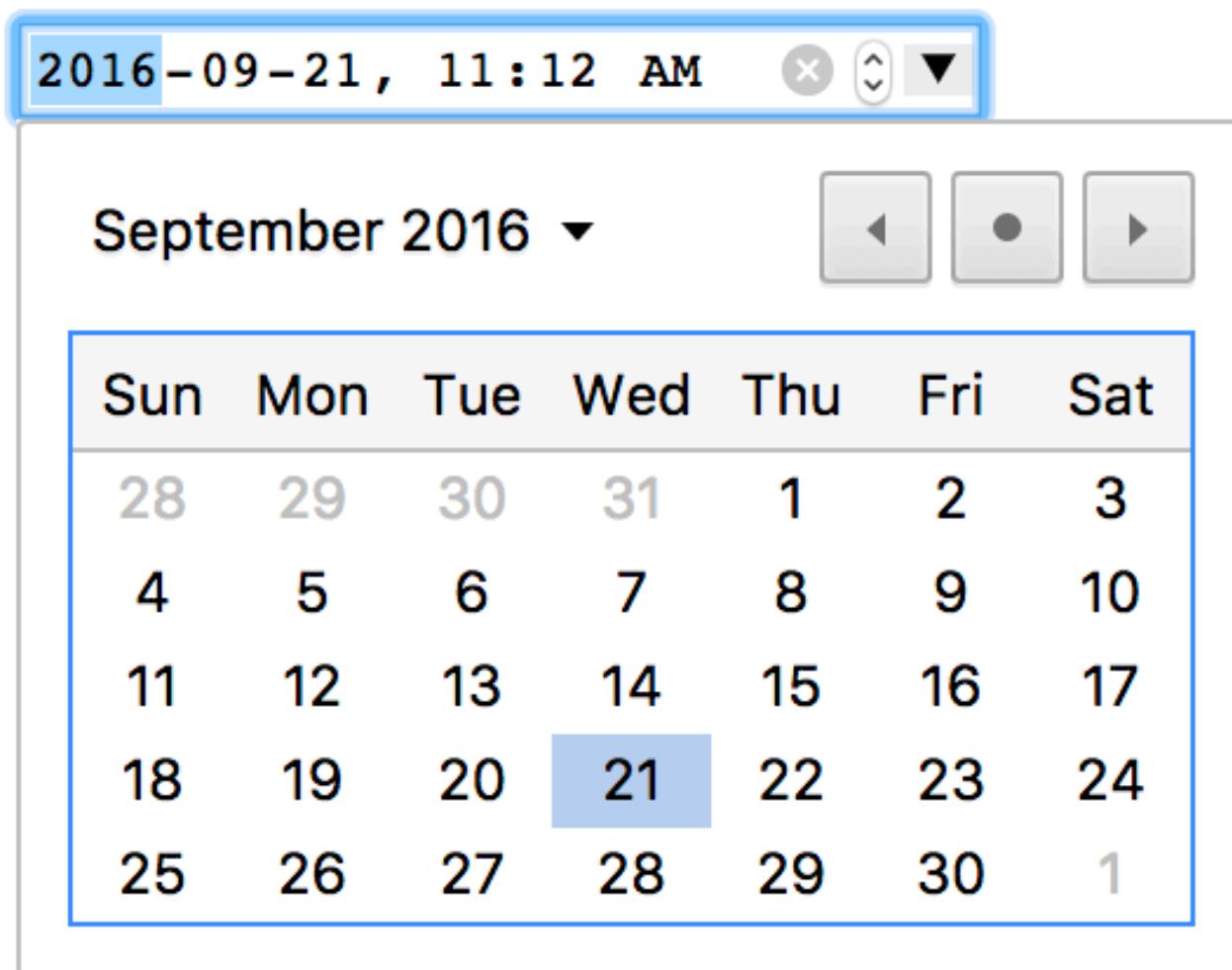
```
<input type="date" name="my_date" id="myDate"  
value="2016-09-21">
```



```
<input  
    type="datetime"  
    name="my_date_time"  
    id="myDateTime"  
    value="2016-09-21T10:11">
```

2016-09-21T10:11

```
<input type="datetime-local" name="my_date_time_local"  
id="myDateTimeLocal" value="2016-09-21T11:12">
```



```
<input type="time" name="my_time"  
id="myTime" value="10:11:12">
```



10 : 11 : 12 AM

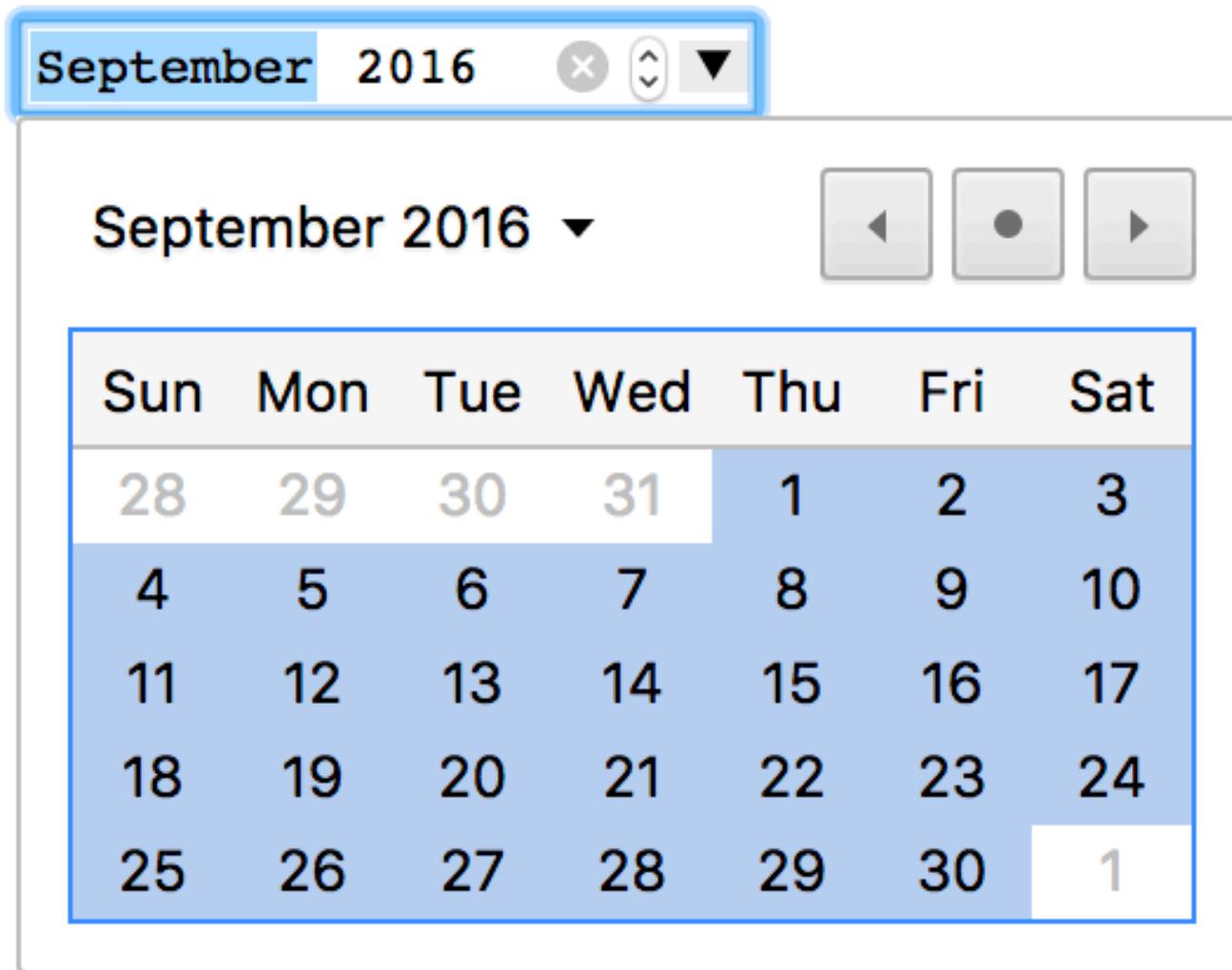
```
<input type="week" name="my_week" id="myWeek" >
```

Week 49, 2016 X ^ ▼

December 2016 ◀ ● ▶

Week	Sun	Mon	Tue	Wed	Thu	Fri	Sat
48	27	28	29	30	1	2	3
49	4	5	6	7	8	9	10
50	11	12	13	14	15	16	17
51	18	19	20	21	22	23	24
52	25	26	27	28	29	30	31

```
<input type="month" name="my_month" id="myMonth"  
value="2016-09">
```



```
<input type="number" name="my_number"  
id="myNumber" min="40" max="50" step="2" value="48">
```

A blurred screenshot of a web browser window. Inside the window, there is a form element consisting of an input field and a button. The input field contains the number '46'. To the right of the input field is a circular button containing a checkmark symbol (✓). The entire image is heavily blurred, obscuring specific details of the browser interface.

```
<input type="range" name="my_range"  
id="myRange" min="70" max="80"  
step="2" value="78">
```



```
<input type="search" name="my_search"  
id="mySearch" value="ponies">
```



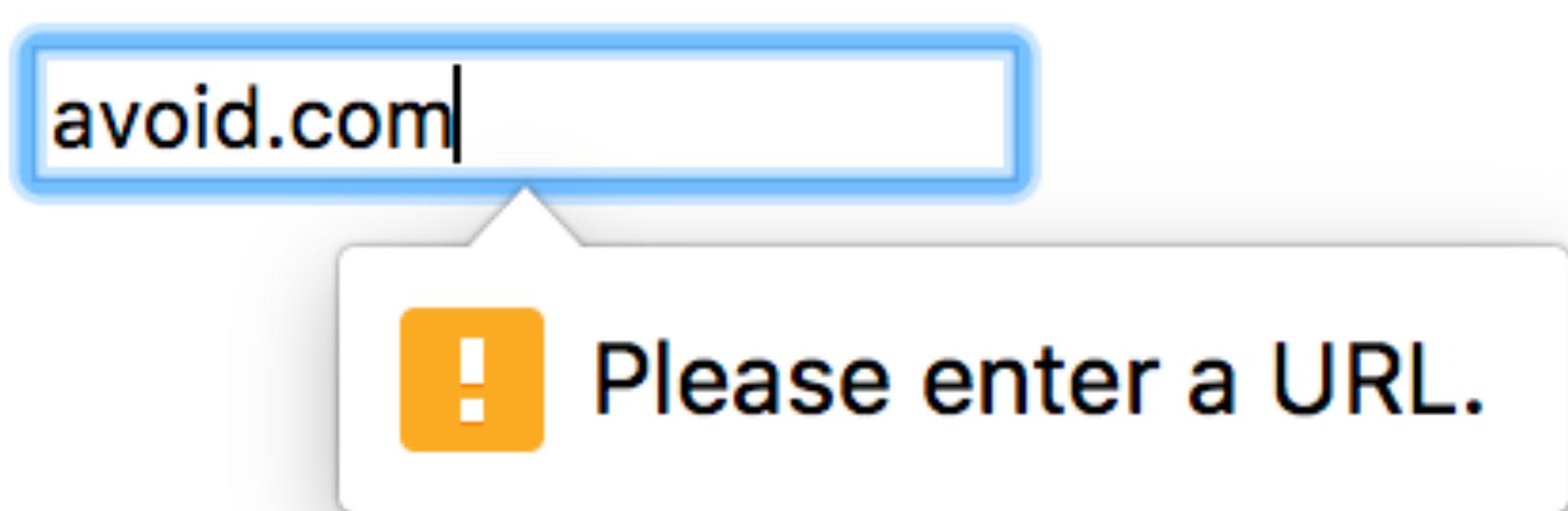
```
<input type="tel" name="my_tel"  
id="myTel" value="(613) 555-5555">
```



(613) 555-5555

```
<input type="url" name="my_url" id="myUrl"  
value="http://a4word.com">
```

http://a4word.com



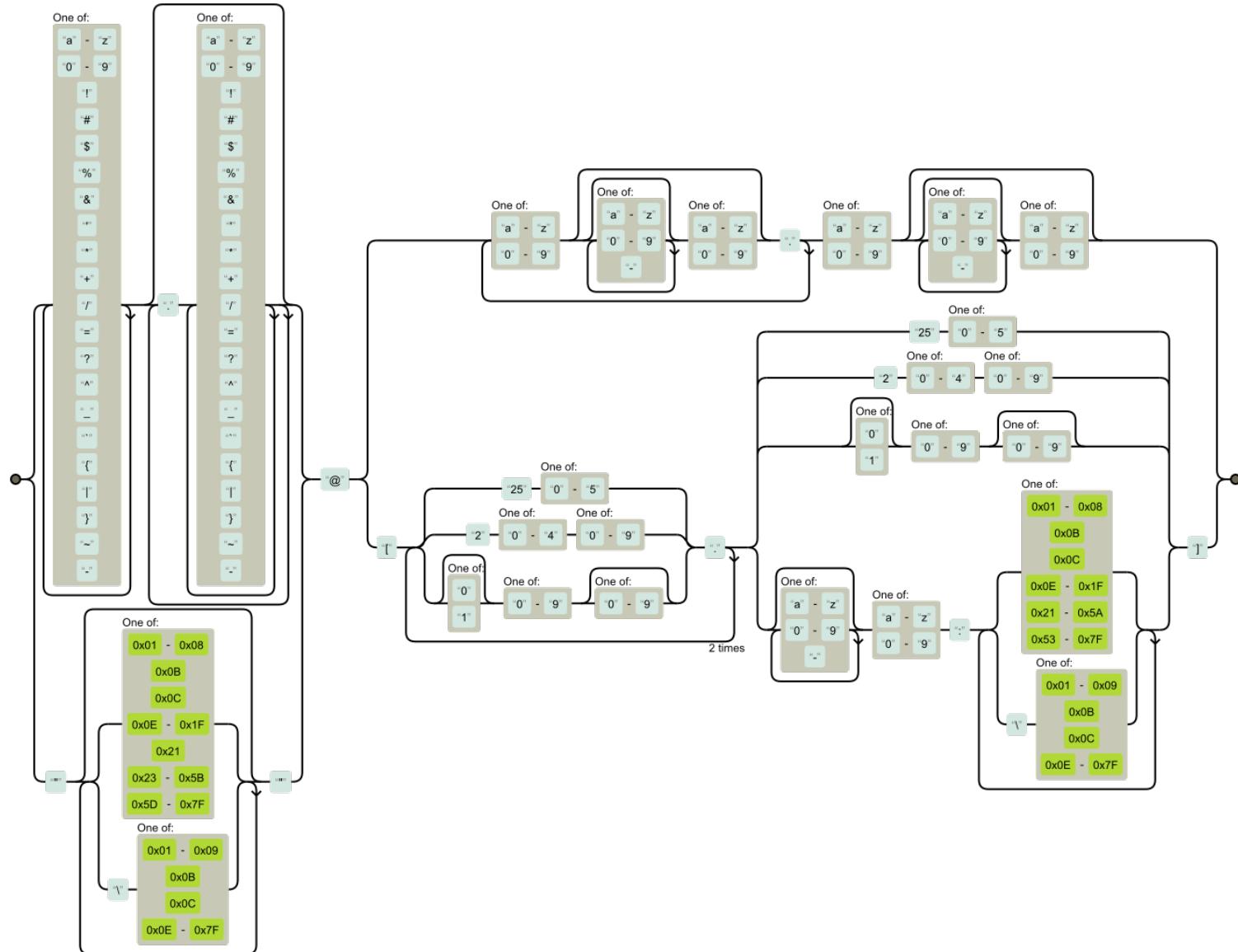
bob."@".smith@mywebsite.com

.smith@mywebsite.com



A part followed by '@' should not contain the symbol "".

<https://www.w3.org/Protocols/rfc822/>



A valid email is not
the same as a
correct email

Email address *

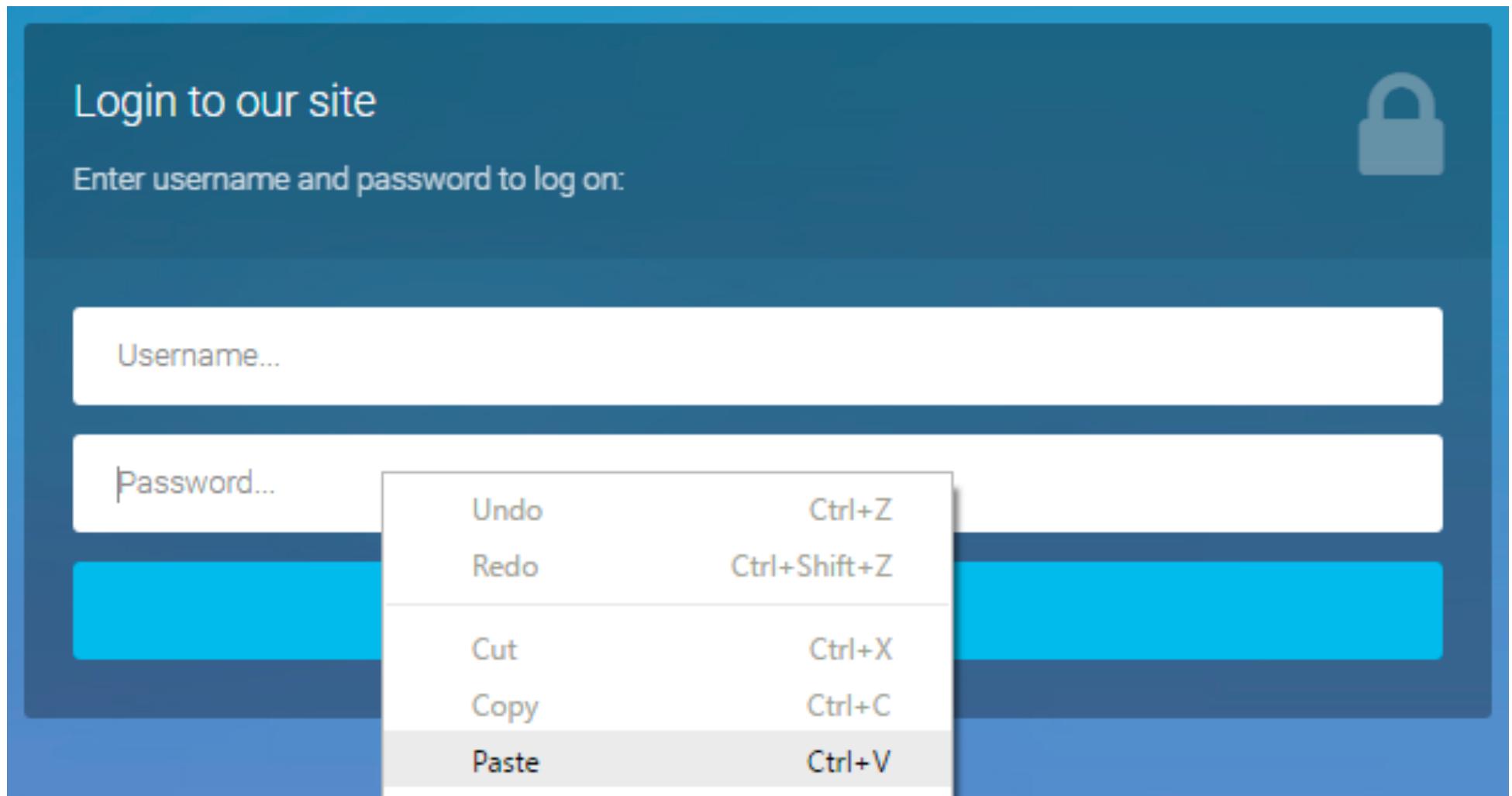
email1@example.com

Confirm email *

email2@example.com



The email addresses do not match



<https://paul.reviews/dont-let-them-paste-passwords/>

- Just stop, y'all. It's a waste of your time and your effort. Put down your Google search for an **email regular expression**, take a step back, and breathe. There's a famous quote that goes:

“

Some people, when confronted with a problem, think, “I know, I’ll use regular expressions.” Now they have two problems.

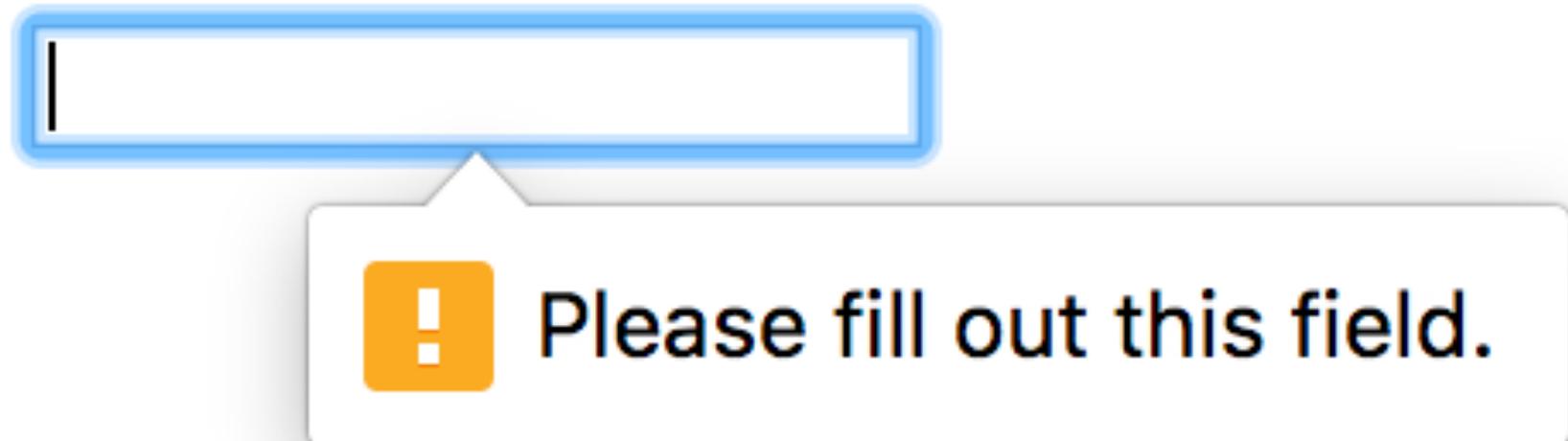
— Jamie Zawinski

<https://davidcelis.com/posts/stop-validating-email-addresses-with-regex/>

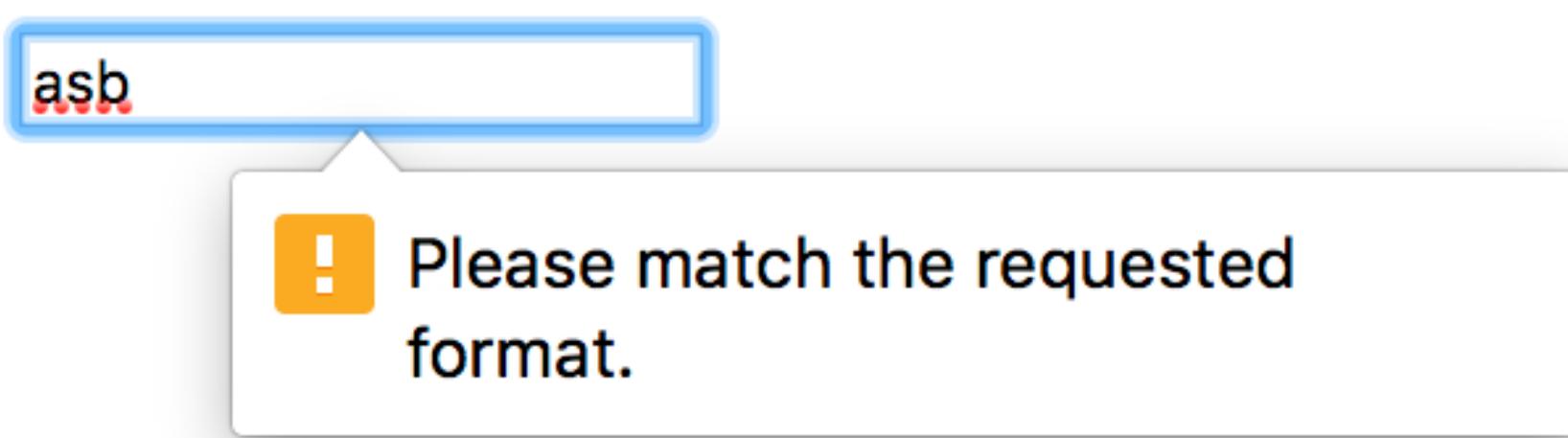
```
<input type="text" name="my_text" id="myText" autofocus>
```


|

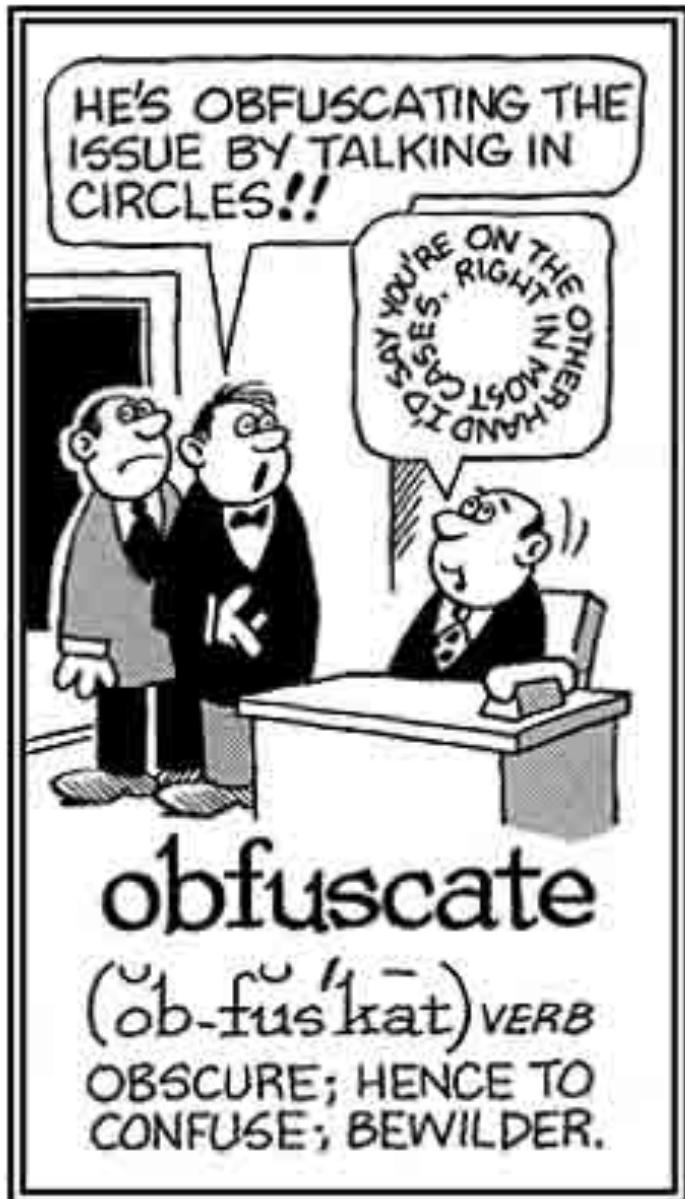
```
<input type="text" name="my_text" id="myText" required>
```



```
<input type="text" name="my_text" id="myText"  
pattern="[a-zA-Z]{4}" placeholder="4 letters">
```



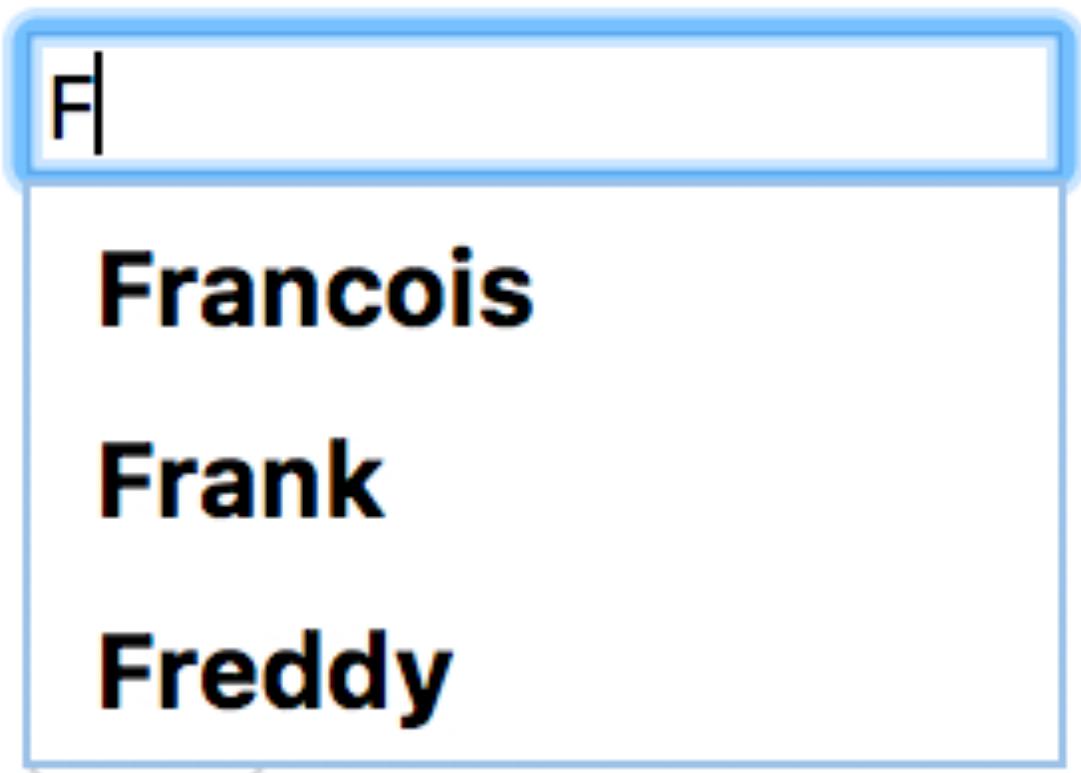
Validation in the
client is great, but



**Obfuscation is
not security.**
Please always
validate on the
server.

Type	Format	Type	Format
color	Hex code	month	yyyy-mm
date	yyyy-mm-dd	number	\d*
datetime*	yyyy-mm-ddThh:mm:ss	email	good luck!
datetime-local	yyyy-mm-ddThh:mm	url	nice try!
time	hh:mm	week	yyyy-Wnn

```
<form method="post" autocomplete="on"  
  <input type="text" name="my_text" id="myText" placeholder="people">  
  
  <input type="text" name="your_text" id="yourText" placeholder="shapes">  
  
  <button type="submit">Go</button>  
  
</form>
```

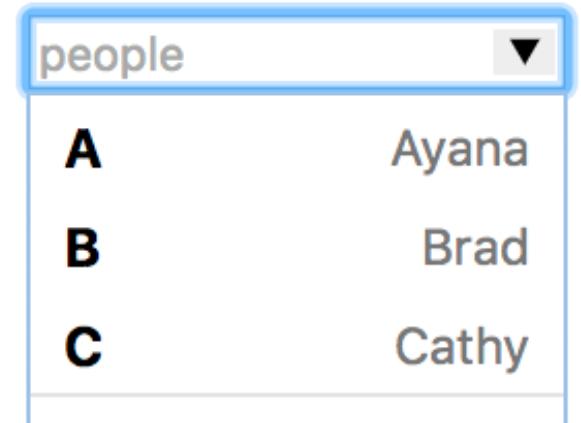


Autocomplete can be managed
on individual elements

```
<form method="post" >  
  
    <input type="text" name="my_text" id="myText"  
          placeholder="people" list="myPeople"  
          autocomplete="off">  
  
    <datalist id="myPeople">  
  
        <option value="A">Ayana</option>  
        <option value="B">Brad</option>  
        <option value="C">Cathy</option>  
    </datalist>  
  
    <input type="text" name="your_text"  
          placeholder="random thoughts"  
          autocomplete="on" >  
  
    <button type="submit">Go</button>  
  
</form>
```

The name is
optional

The value is what is sent to the server



Here's the old way to specify sections

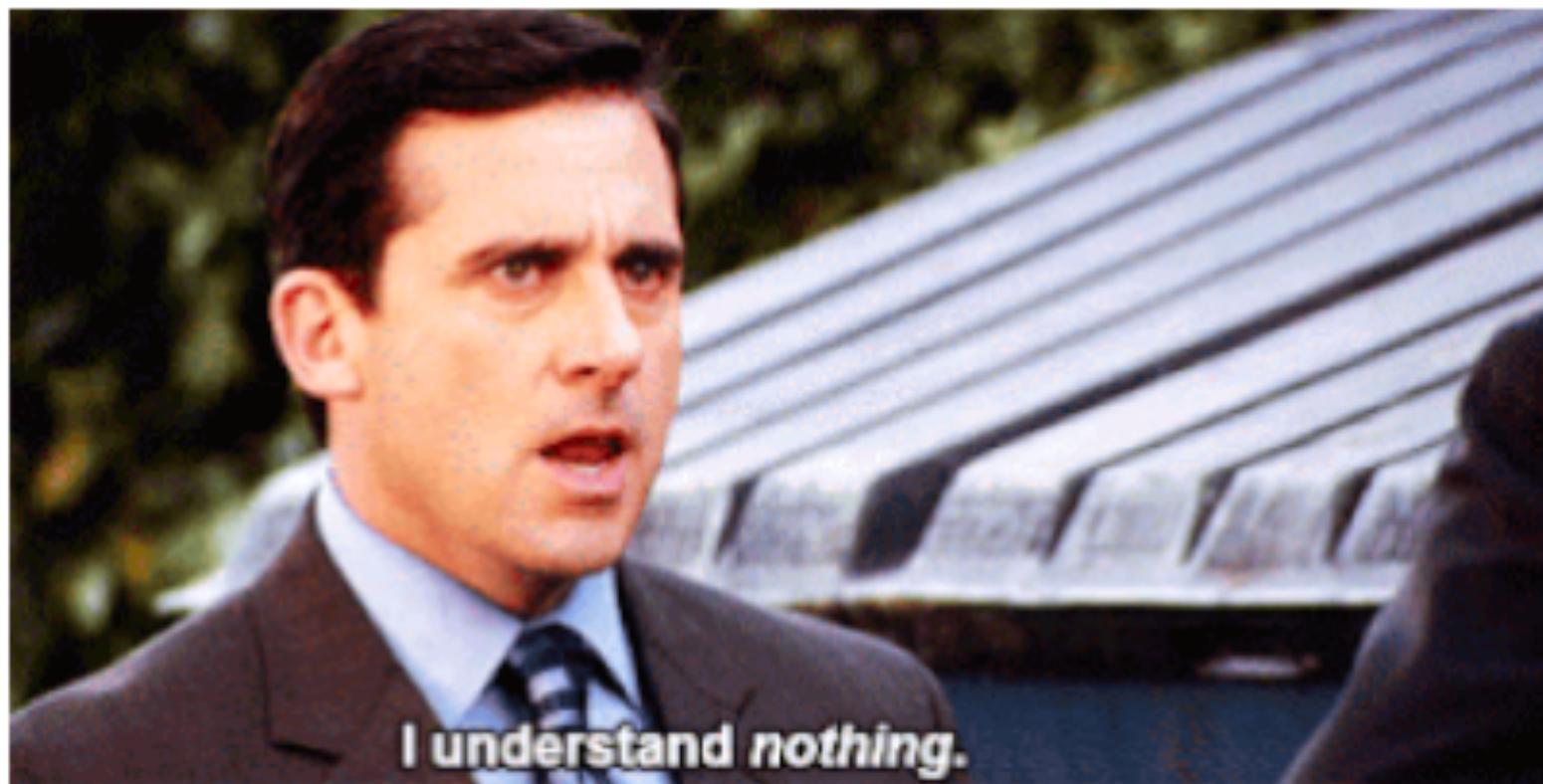
```
<body>
  <header>The Top</header>
  <section>Section 1</section>
  <section>Section 2</section>
  <footer>The Bottom</footer>
</body>
```



New(ish) semantic way

```
<div class="header">The Top</div>
<div class="section">Section 1</div>
<div class="section">Section 2</div>
<div class="footer">The Bottom</div>
```

```
body>
  <figure>
    
    <figcaption>I understand nothing</figcaption>
  </figure>
/boby>
```



I understand nothing

```
<p>I <mark>understand</mark> nothing</p>
```

I **understand** nothing

```
<body>
  <details>
    <summary>Lorem Ipsum is simply dummy text</summary>
    <p>
      Lorem Ipsum is simply dummy text of the printing and typesetting industry. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
    </p>
  </details>
</body>
```

- **Lorem Ipsum is simply dummy text**

```
<body>
  <details>
    <summary>Lorem Ipsum is simply dummy text</summary>
    <p>
      Lorem Ipsum is simply dummy text of the printing and typesetting industry. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.
    </p>
  </details>
</body>
```

▼ Lorem Ipsum is simply dummy text

Lorem Ipsum is simply dummy text of the printing and typesetting industry. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

```
<article>
  <h1>Lorem Ipsum is simply dummy text</h1>
  <p>
    Lorem Ipsum is simply dummy text of the p
    been the industry's standard dummy text e
    a galley of type and scrambled it to make
    five centuries, but also the leap into ele
    unchanged. It was popularised in the 1960
    Lorem Ipsum passages, and more recently w
    PageMaker including versions of Lorem Ips
  </p>
</article>
```

Lorem Ipsum is simply dummy text

No change to UX, just
better semantics



Lorem Ipsum is simply dummy text of the printing and typesetting indust
dummy text ever since the 1500s, when an unknown printer took a galley
book. It has survived not only five centuries, but also the leap into electro
It was popularised in the 1960s with the release of Letraset sheets contain
with desktop publishing software like Aldus PageMaker including versio

```
<p>  
    "There is no one who loves pain itself,  
    who seeks after it and wants to have it,  
    simply because it is pain..."  
</p>  
<aside>For more great quotes checkout myquotes.com</aside>  
</body>
```

The "aside" is indirectly
linked to the content above

Again, no change to UX, just better semantics

"There is no one who loves pain itself, who seeks after it and wants to have it, simply because it is pain..."

For more great quotes checkout myquotes.com

```
<body>
  <div id="myStatus"></div>
  <meter min="0" max="20" value="0" id="myMeter">
</body>
<script>
  var m = document.getElementById("myMeter");
  var s = document.getElementById("myStatus");
  var tick = function() {
    if (m.value < m.max) {
      m.value = m.value + 1;
      s.innerHTML = "Done " + m.value + " out of " + m.max;
      setTimeout(tick, 100);
    } else {
      s.innerHTML = "Finished!";
    }
  }
  tick();
</script>
```

Attaching to HTML
DOM elements

Increasing the value

Every 100ms

Start the counter

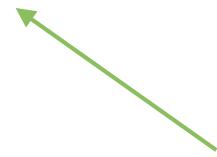
Done 19 out of 20



Finished!

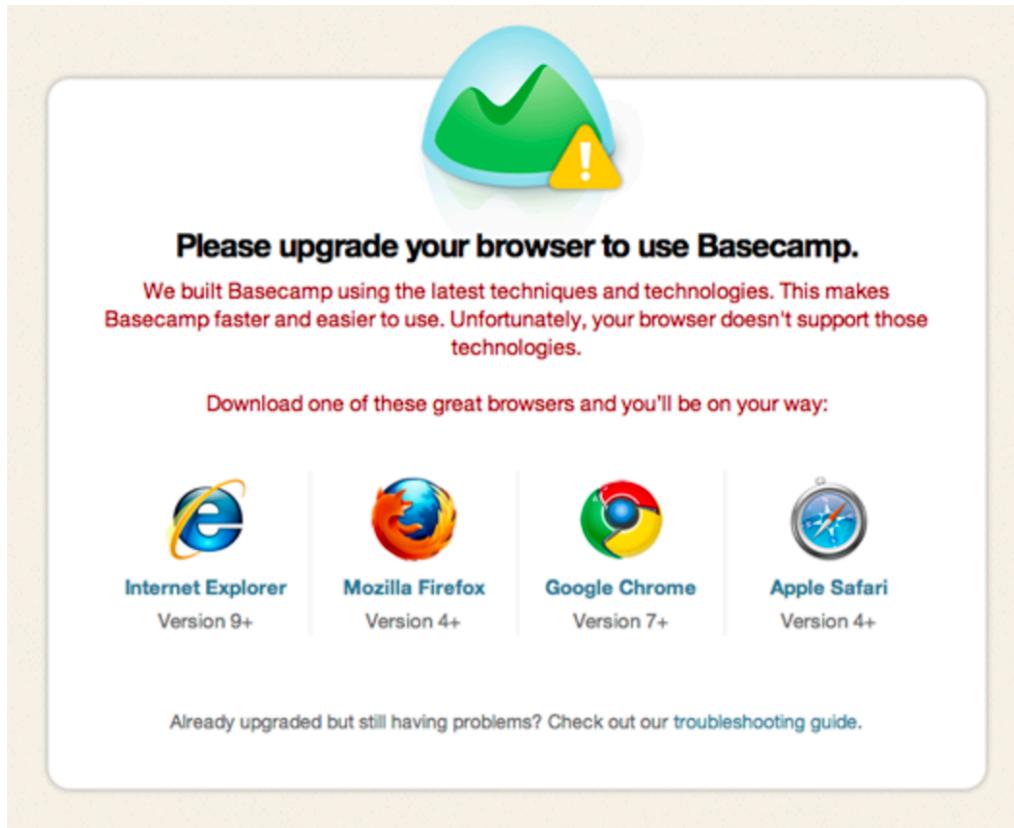


```
<span>How do you say  
Supercalifragilisticwbrexpialidocious?</span>
```



Tells the browser a good place to line
break (word break)

How do you say Supercalifragilistic
expialidocious?



A screenshot of a Basecamp error message. At the top is a green mountain icon with a yellow exclamation mark. Below it is the text "Please upgrade your browser to use Basecamp." A red paragraph explains that Basecamp uses modern technologies and the user's browser doesn't support them. A blue link below suggests downloading one of four supported browsers: Internet Explorer, Mozilla Firefox, Google Chrome, or Apple Safari. At the bottom, a blue link offers a troubleshooting guide.

Please upgrade your browser to use Basecamp.

We built Basecamp using the latest techniques and technologies. This makes Basecamp faster and easier to use. Unfortunately, your browser doesn't support those technologies.

Download one of these great browsers and you'll be on your way:

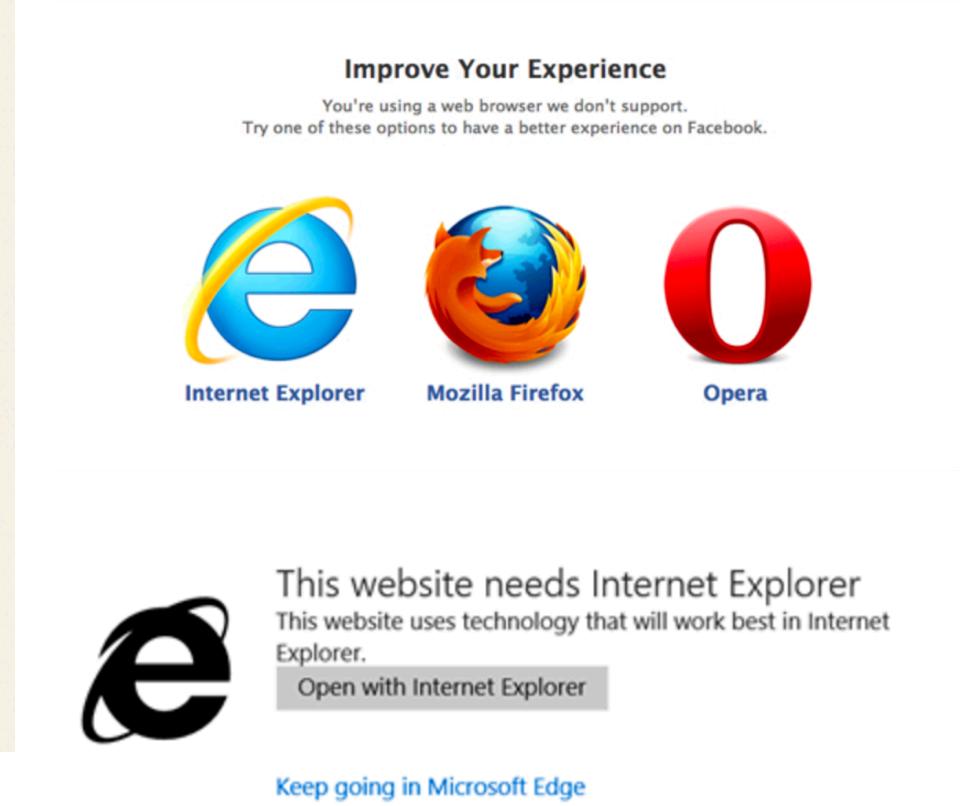
 **Internet Explorer**
Version 9+

 **Mozilla Firefox**
Version 4+

 **Google Chrome**
Version 7+

 **Apple Safari**
Version 4+

Already upgraded but still having problems? Check out our [troubleshooting guide](#).



A screenshot of a Facebook page. It starts with a "Improve Your Experience" section, noting that the user is using a unsupported browser and suggesting alternatives. Below are logos for Internet Explorer, Mozilla Firefox, and Opera. Further down, a large Internet Explorer logo is shown with text explaining the need for IE and a button to open it.

Improve Your Experience

You're using a web browser we don't support.
Try one of these options to have a better experience on Facebook.

 **Internet Explorer**

 **Mozilla Firefox**

 **Opera**

This website needs Internet Explorer
This website uses technology that will work best in Internet Explorer.
[Open with Internet Explorer](#)

[Keep going in Microsoft Edge](#)



Internet Explorer 6, you cannot be serious?

We don't support old technologies and neither should you!

Please upgrade to the latest version of Internet Explorer or use a different browser to visit our site.

REFERENCES

- ▶ <http://www.creativebloq.com/netmag/10-great-text-editors-web-designers-71412411>
- ▶ <http://www.clarissapeterson.com/2012/11/html5-accessibility/>
- ▶ <https://css-tricks.com/examples/ButtonMaker/#>
- ▶ <https://dev.w3.org/html5/html-author/charref>
- ▶ <http://www.iacquire.com/blog/18-meta-tags-every-webpage-should-have-in-2013>
- ▶ <https://developers.facebook.com/docs/sharing/opengraph>

REFERENCES

- ▶ <https://elliot.land/post/its-impossible-to-validate-an-email-address>
- ▶ <https://mathiasbynens.be/demo/url-regex>
- ▶ <https://code.tutsplus.com/articles/10-rare-html-tags-you-really-should-know--net-3908>