

Standardizing `<select>`:

What the future holds for HTML Controls

Hi! I'm Stephanie.

Program Manager for
Microsoft Edge
Developer Experiences

@seaotta

Hi! I'm Stephanie.

Designer

Front-end developer

Dev and Designer Advocate

@seaotta



Designer



Developer



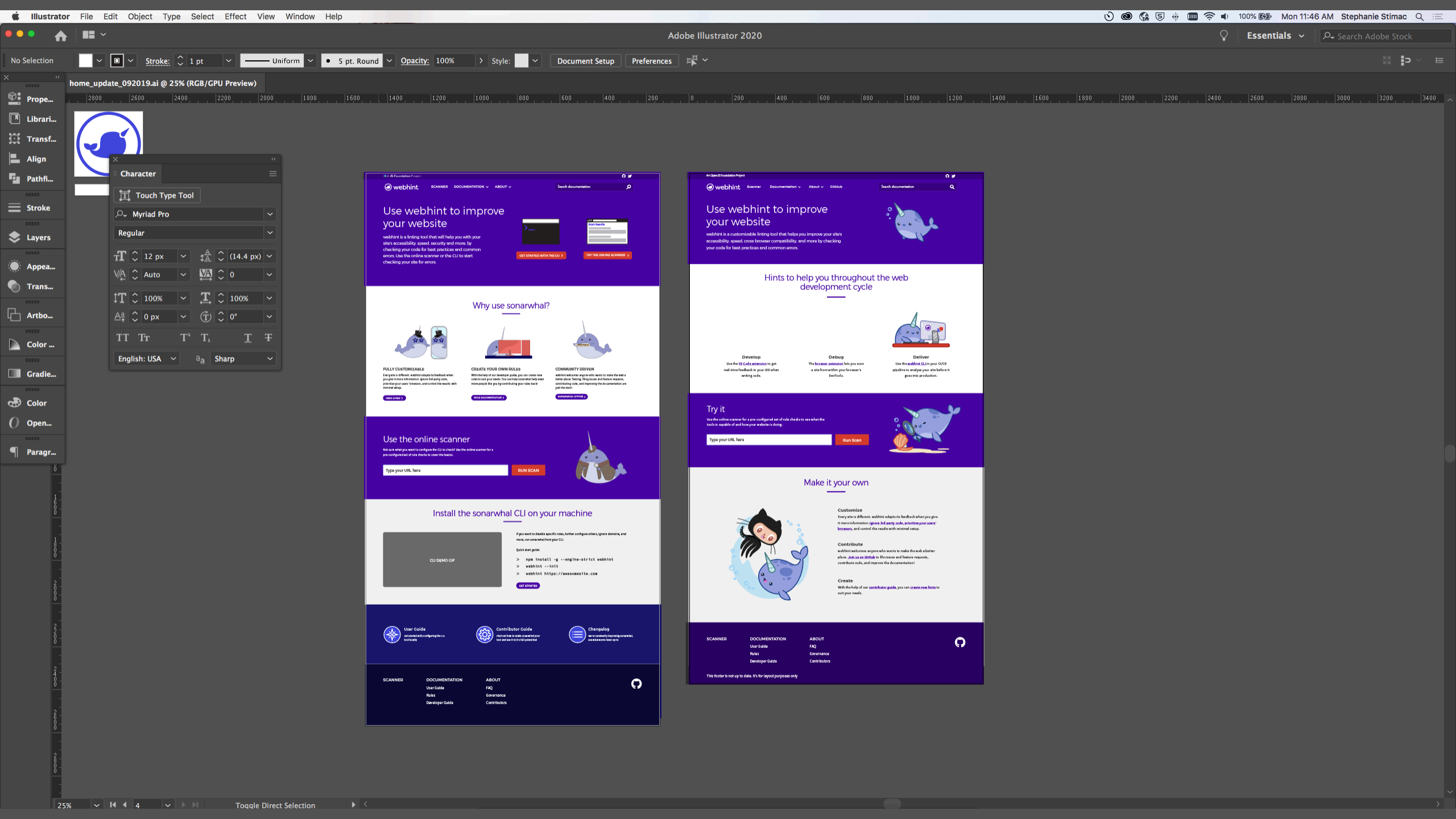


ACTIONSCRIPT 3.0



*Learn ActionScript the
Quick and Easy Way!*

DERRICK YPENBURG



```
class="portfolio">  
  id="0" class="portfolio-item wide picture one" startburg/><  
  class="portfolio-item tall picture two" startburg/><  
  class="portfolio-item large picture three" startburg three/><  
  class="portfolio-item tall picture four" summer/><  
  class="portfolio-item small picture insta" inst/><  
  class="portfolio-item small picture kontakt" kontakt/><  
  class="portfolio-item small picture ich" ich/><  
  class="portfolio-item superwide picture shooting" anden  
</div>  
</div>  
</div>
```





Mock up



Production Site

Form field widgets to easily allow users to enter a date, time or both, generally by using a calendar/time input widget. Refers to supporting the following input types: `date`, `time`, `datetime-local`, `month` & `week`.

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Opera Mobile	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baic Brow
		2-52	4-19					2.1-3							
	¹ 12	⁴ 53-56	⁵ 20-24			3.2-4.3		³ 4-4.3							
6-10	13-79	⁵ 57-72	25-79	3.1-12.1	10-65	² 5-13.1		4.4-4.4.4	12-12.1				4-9.2		
11	80	⁵ 73	80	13	66	² 13.2	all	80	46	80	68	12.12	10.1	1.2	7.1
		⁵ 74-75	81-83	TP		² 13.3									

Notes

Known issues (0)

Resources (10)

Feedback

There used to also be a `datetime` type, but it was **dropped from the HTML spec**.

¹ Partial support in Microsoft Edge refers to supporting `date`, `week`, and `month` input types, and not `time` and `datetime-local`.

² Partial support in iOS Safari refers to not supporting the `week` input type, nor the `min`, `max` or `step` attributes

³ Some modified versions of the Android 4.x browser do have support for date/time fields.

⁴ Can be enabled in Firefox using the `dom.forms.datetime` flag.

⁵ Partial support refers to supporting `date` and `time` input types, but not `datetime-local`, `month` or `week`.



HTML Controls

Standardizing <select>

The Past: why HTML Controls are the way they are

The Present: where we're at now

The Future: what's to come with HTML Controls (select!)

Some History

HTML Form Controls

1995

1995



1995

Hypertext Markup Language - 2.0

September 22, 1995

T. Berners-Lee (1)

D. Connolly (1)

(1) MIT/W3C

- [Forms](#)
 - [Form Elements](#)
 - [Form: FORM](#)
 - [Input Field: INPUT](#)
 - [Text Field: INPUT TYPE=TEXT](#)
 - [Password Field: INPUT TYPE=PASSWORD](#)
 - [Check Box: INPUT TYPE=CHECKBOX](#)
 - [Radio Button: INPUT TYPE=RADIO](#)
 - [Image Pixel: INPUT TYPE=IMAGE](#)
 - [Hidden Field: INPUT TYPE=HIDDEN](#)
 - [Submit Button: INPUT TYPE=SUBMIT](#)
 - [Reset Button: INPUT TYPE=RESET](#)
 - [Selection: SELECT](#)
 - [Option: OPTION](#)
 - [Text Area: TEXTAREA](#)
 - [Form Submission](#)
 - [The form-urlencoded Media Type](#)
 - [Query Forms: METHOD=GET](#)
 - [Forms with Side-Effects: METHOD=POST](#)
 - [Example Form Submission: Questionnaire Form](#)

1995

Hypertext Markup Language - 2.0

September 22, 1995

T. Berners-Lee (1)

D. Connolly (1)

(1) MIT/W3C

- [Forms](#)
 - [Form Elements](#)
 - [Form: FORM](#)
 - [Input Field: INPUT](#)
 - [Text Field: INPUT TYPE=TEXT](#)
 - [Password Field: INPUT TYPE=PASSWORD](#)
 - [Check Box: INPUT TYPE=CHECKBOX](#)
 - [Radio Button: INPUT TYPE=RADIO](#)
 - [Image Pixel: INPUT TYPE=IMAGE](#)
 - [Hidden Field: INPUT TYPE=HIDDEN](#)
 - [Submit Button: INPUT TYPE=SUBMIT](#)
 - [Reset Button: INPUT TYPE=RESET](#)
 - [Selection: SELECT](#)
 - [Option: OPTION](#)
 - [Text Area: TEXTAREA](#)
 - [Form Submission](#)
 - [The form-urlencoded Media Type](#)
 - [Query Forms: METHOD=GET](#)
 - [Forms with Side-Effects: METHOD=POST](#)
 - [Example Form Submission: Questionnaire Form](#)

Pre-
1995

<HTML>
1.0

Pre-
1995

<HTML>
1.0



The primary focus of the specification draft was to capture common HTML practice in web browsers as of June 1994.

Web Browsers 1991-1994

WorldWideWeb
(Nexus)

ViolaWWW

Erwise

MidasWWW

MacWWW

Mosaic

Cello

Lynx 2.0

Arena

AMosaic 1.0

IBM WebExplorer

Netscape Navigator

SlipKnot 1.0

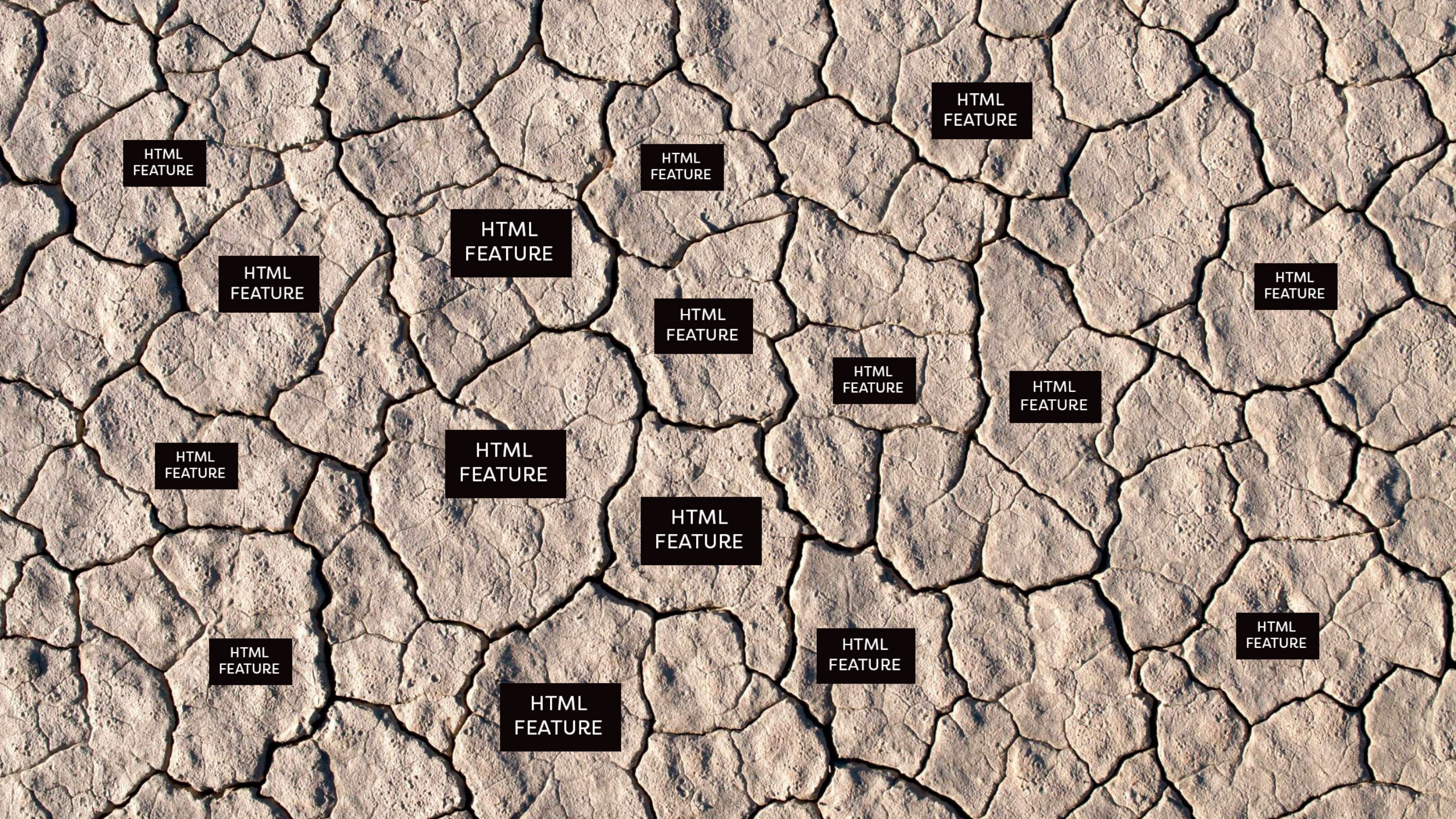
MacWeb

lbrowse

Agora (Argo)

Minuet





HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

HTML
FEATURE

1995

Selection: SELECT

The *SELECT* [element](#) constrains the form field to an enumerated list of values. The values are given in *OPTION* elements. Attributes are:

MULTIPLE

indicates that more than one option [may](#) be included in the value.

NAME

specifies the name of the form field.

SIZE

specifies the number of visible items. Select fields of size one are typically pop-down menus, whereas select fields with size greater than one are typically lists.

For example:

```
<SELECT NAME="flavor">
<OPTION>Vanilla
<OPTION>Strawberry
<OPTION value="RumRasin">Rum and Raisin
<OPTION selected>Peach and Orange
</SELECT>
```

The initial state has the first option selected, unless a *SELECTED* attribute is present on any of the *OPTION* elements.

Option: OPTION

The Option [element](#) can only occur within a Select [element](#). It represents one choice, and has the following attributes:

SELECTED

Indicates that this option is initially selected.

VALUE

indicates the value to be returned if this option is chosen. The field value defaults to the content of the *OPTION* [element](#).

The content of the *OPTION* [element](#) is presented to the user to represent the option. It is used as a returned value if the *VALUE* attribute is not present.

1995

First name:

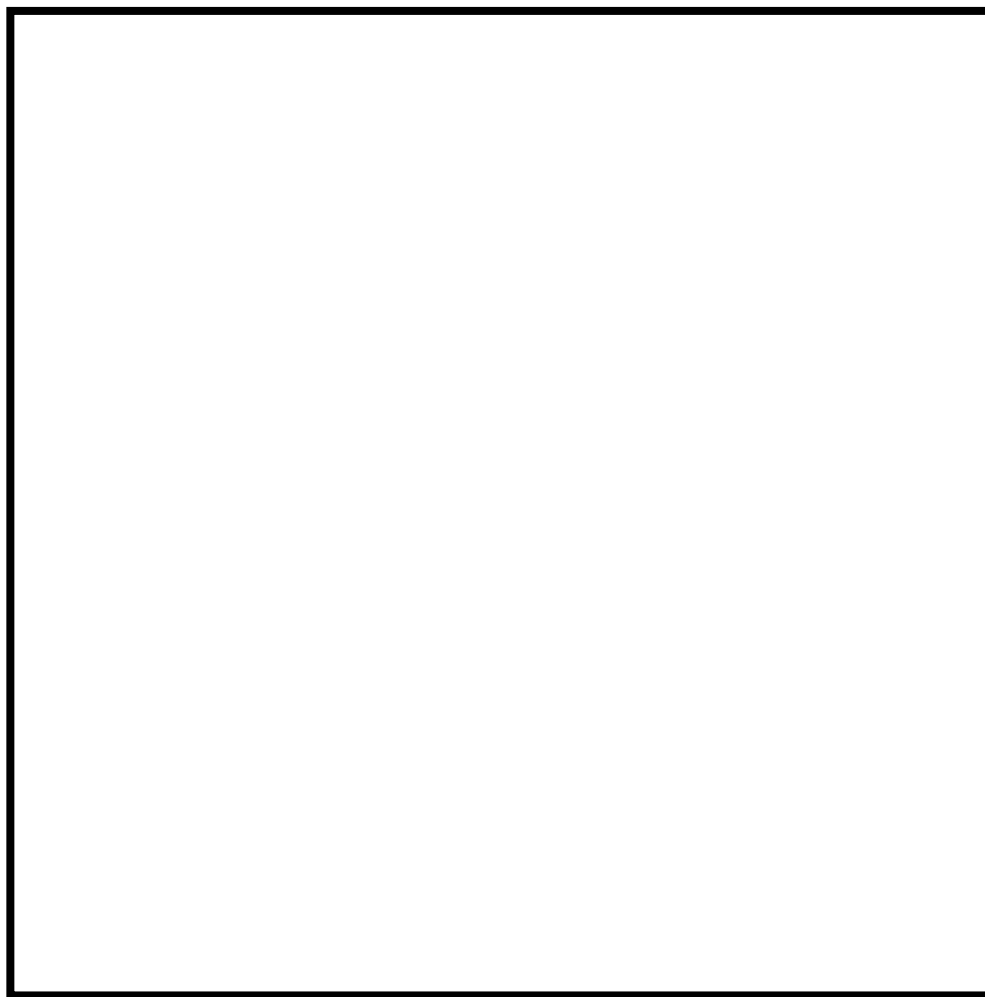
John

Last name:

Doe

Submit

1995



CSS

IS

AWESOME

1997

CSS

IS

AWESOME

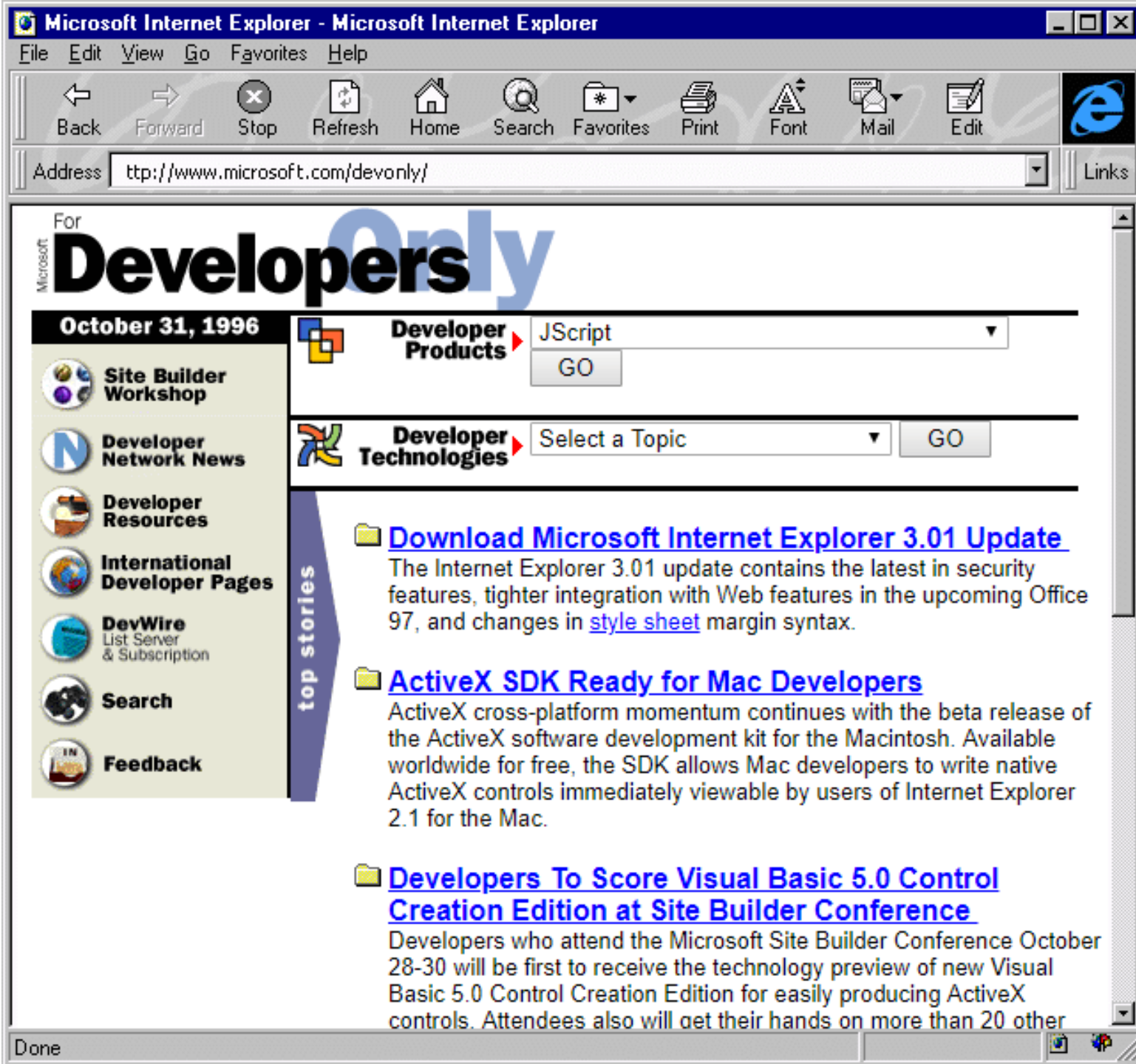
1999

<HTML>
4.01



CSS
IS
AWESOME

1995



Option ▴ ▾

Safari

Option ▴ ▾

IE Mac

Option ▾

Moz/FF Mac

Option ▴ ▾

Opera Mac

Option ▴ ▾

Camino

Option ▾

IE Win 2k

Option ▾

Moz/FF Win 2k

Option ▾

Opera Win 2k

Option ▾

IE Win XP

Option ▾

Moz/FF Win XP

Option ▾

Opera Win XP

```
.form {  
    -webkit-appearance: value;  
    -moz-appearance: value;  
    appearance: value;  
}
```

Can I use

? ⚙ Settings

x | Feature: CSS Appearance

CSS Appearance 📄 - WD

The **appearance** property defines how elements (particularly form controls) appear by default. By setting the value to **none** the default appearance can be entirely redefined using other CSS properties.

Usage % of all users ⌵ ?
Global 0% + 95.39% = 95.39%
unprefixed: 0% + 2.04% = 2.04%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	
		1 3 2-34			10-12.1										
6-10	1 2 12-18	1 35-71	1 4-78	1 3.1-12.1	1 15-63	1 3.2-13.1		1 2.1-4.4.4	12-12.1				1 4-9.2		
11	1 79	1 72	1 79	1 13	1 64	1 13.2	all	1 76	1 46	1 79	1 68	1 12.12	1 10.1	1 1.2	1
		1 73-74	1 80-82	1 TP		1 13.3									

Notes Known issues (0) Resources (2) Feedback

- ¹ The appearance property is supported with the **none** value, but not **auto**. WebKit, Blink, and Gecko browsers also support additional vendor specific values.
- ² Microsoft Edge and IE Mobile support this property with the `-webkit-` prefix, rather than `-ms-` for interop reasons.
- ³ `-moz-appearance:none` doesn't remove the dropdown arrow in select tag

Recap

Pre-1995: Lots of browsers pop up

1994: HTML 1.0 draft expires

1995: HTML 2.0 becomes standardized spec

- Basic HTML form controls standardized

- No standard for styling

- Operating system dependency

1999: CSS Supported by HTML 4.01 embraced by more browsers

1999 – present

The Current State of Styling Native Controls

It's not bad...

But it's not great either.

Feasibility of Styling Form Controls with CSS

Feasibility of Styling Form Controls with CSS

Can be styled with
few problems

<form>

<fieldset>

<label>

<output>

Text-field (<input>)

Buttons

Feasibility of Styling Form Controls with CSS

Can be styled with few problems

<form>
<fieldset>
<label>
<output>
Text-field (<input>)
Buttons

Can be styled with complex CSS and hacks

Checkboxes
Radios
<legend>

Feasibility of Styling Form Controls with CSS

Can be styled with few problems

<form>
<fieldset>
<label>
<output>
Text-field (<input>)
Buttons

Can be styled with complex CSS and hacks

Checkboxes
Radios
<legend>

Good night and good luck.

<select>
<option>
<optgroup>
<datalist>
<progress>
<meter>
Color picker
Date controls
Dropdown widgets
Range
File picker

Feasibility of Styling Form Controls with CSS

Can be styled with few problems

<form>

<fieldset>

<label>

<output>

Text-field (<input>)

Buttons

Can be styled with complex CSS and hacks

Checkboxes

Radios

<legend>

Good night and good luck.

<select>

<option>

<optgroup>

<datalist>

<progress>

<meter>

Color picker

Date controls

Dropdown widgets

Range

File picker

“CSS 2.1 does not define which properties apply to form controls and frames, or how CSS can be used to style them. User agents may apply CSS properties to these elements. Authors are recommended to treat such support as experimental. A future level of CSS may specify this further.”

Browser Inconsistencies

Browser Inconsistencies

Chrome

No file chosen

EdgeHTML

Firefox

No file selected.

Safari

no file selected

No extensibility



Scott Jehl

@scottjehl



you have one problem:
you want icons in your <select> menu options.

you decide to make a custom select menu:
you now have at least 75 problems.

11:50 AM · Feb 6, 2020 · [Twitter Web App](#)

41 Retweets **199** Likes

```
<video controls width="1080">  
</video>
```

▶ 0:00



```
<video controls width="1080">  
</video>
```

▶ 0:00



```
<video width="1080">  
</video>
```

UI Components

Dropdown

↓

Secondary

Radio Group

Jotaro Kujo

Joseph Joestar

Dio Brando

☒☒☐

Value Slider

At least 8,000

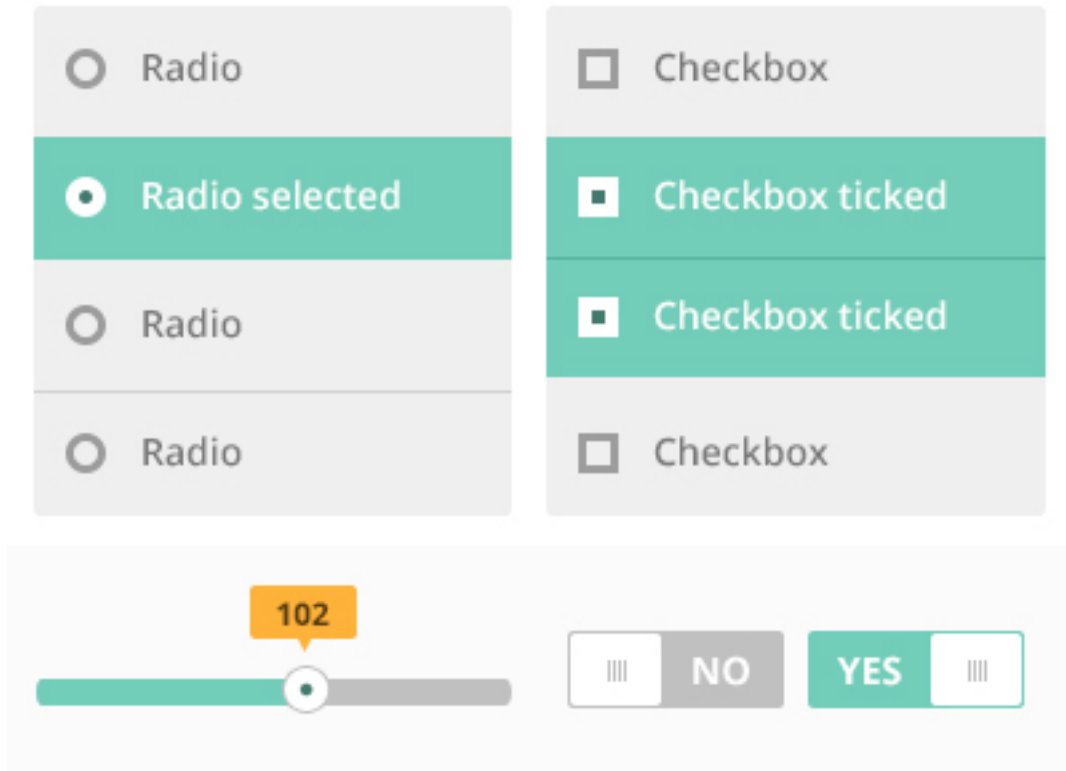
ZeroTen Thousand

Value Range

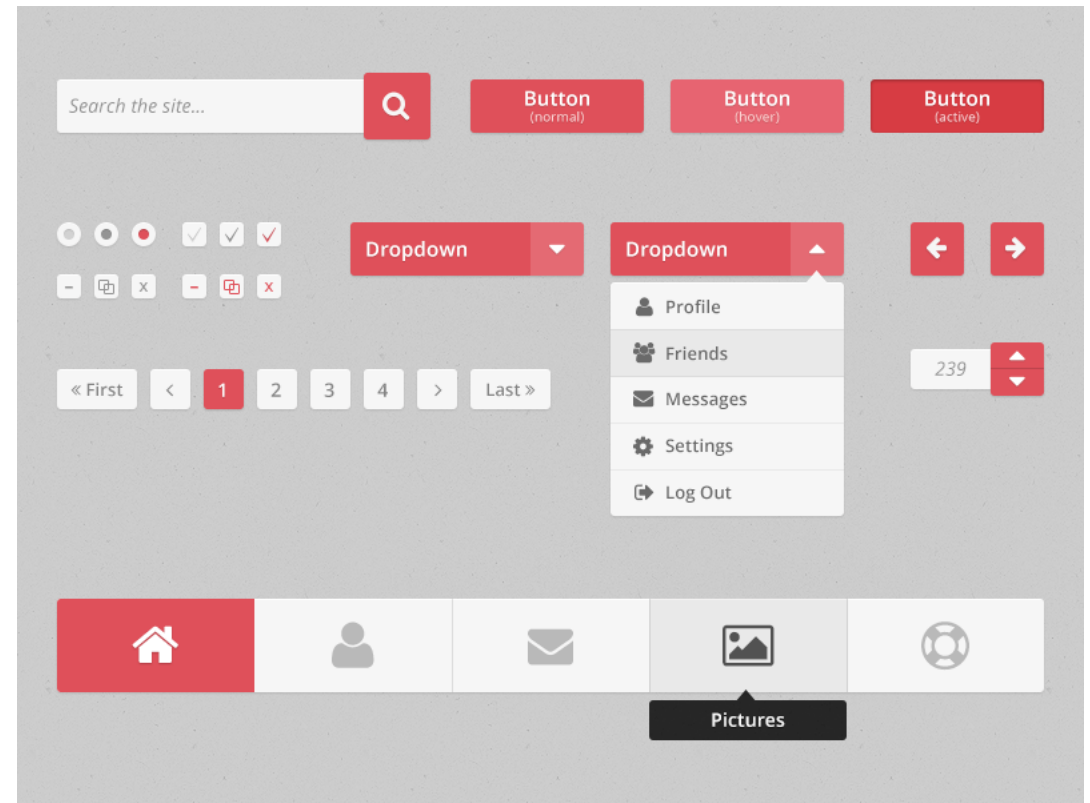
Between 4,000 and 9,000

ZeroTen Thousand

Dribbble: Jordan Ranson



Dribbble: Mark Hendriks



Dribbble: Seb kay

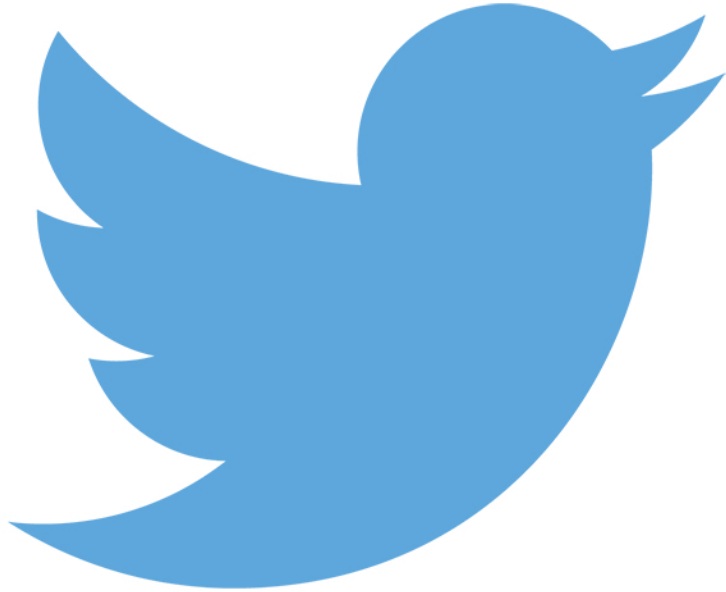
What do
developers
really want?

Do they really
want better
form controls?



OH YES!

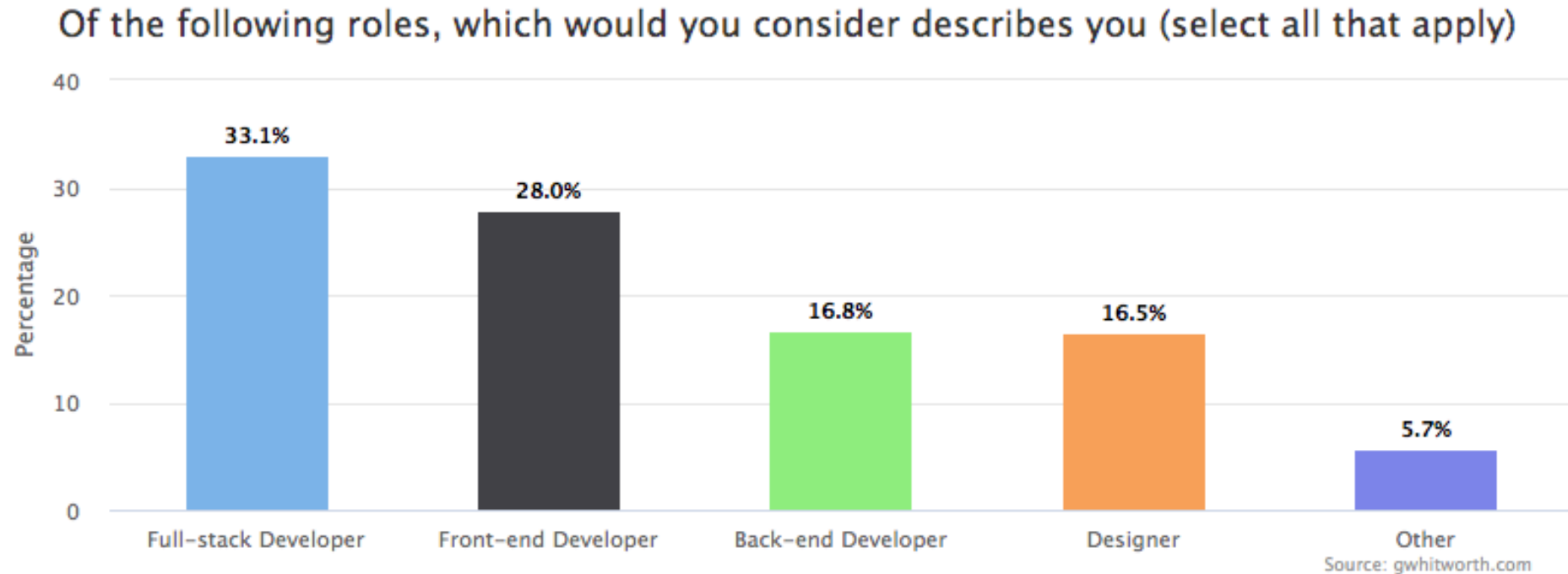
Initial Research



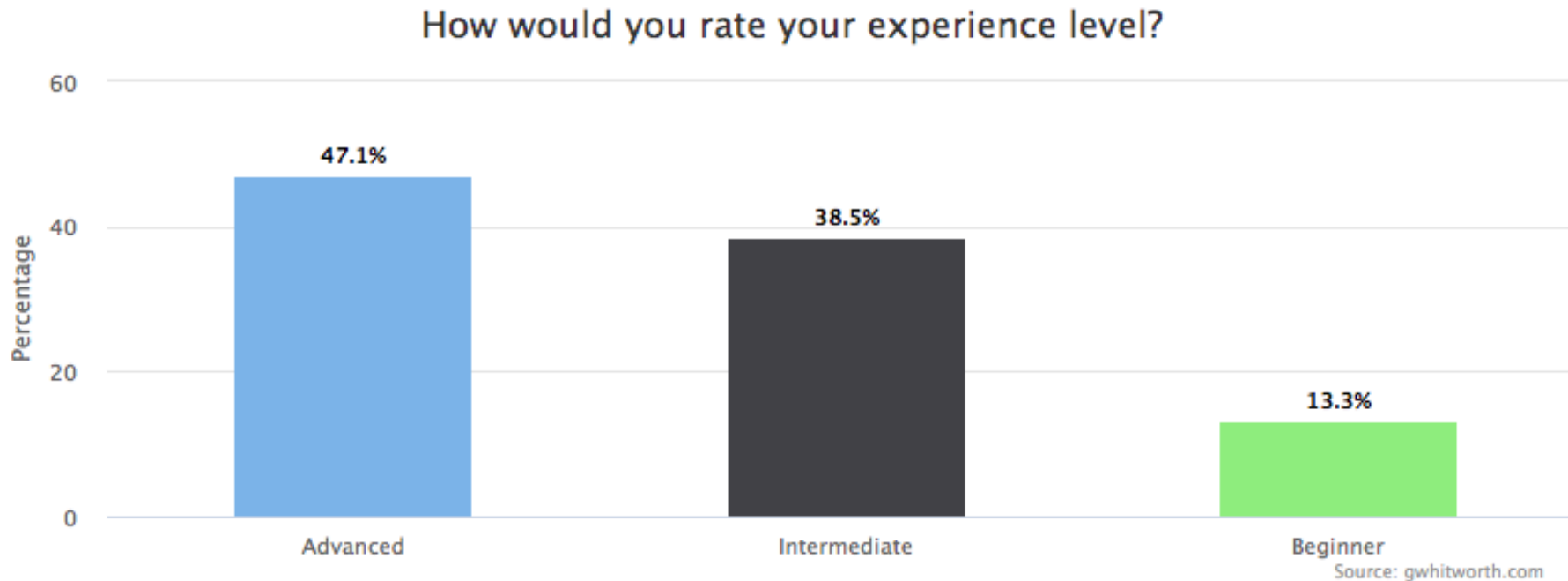
1400
respondents

Initial Research: Demographics

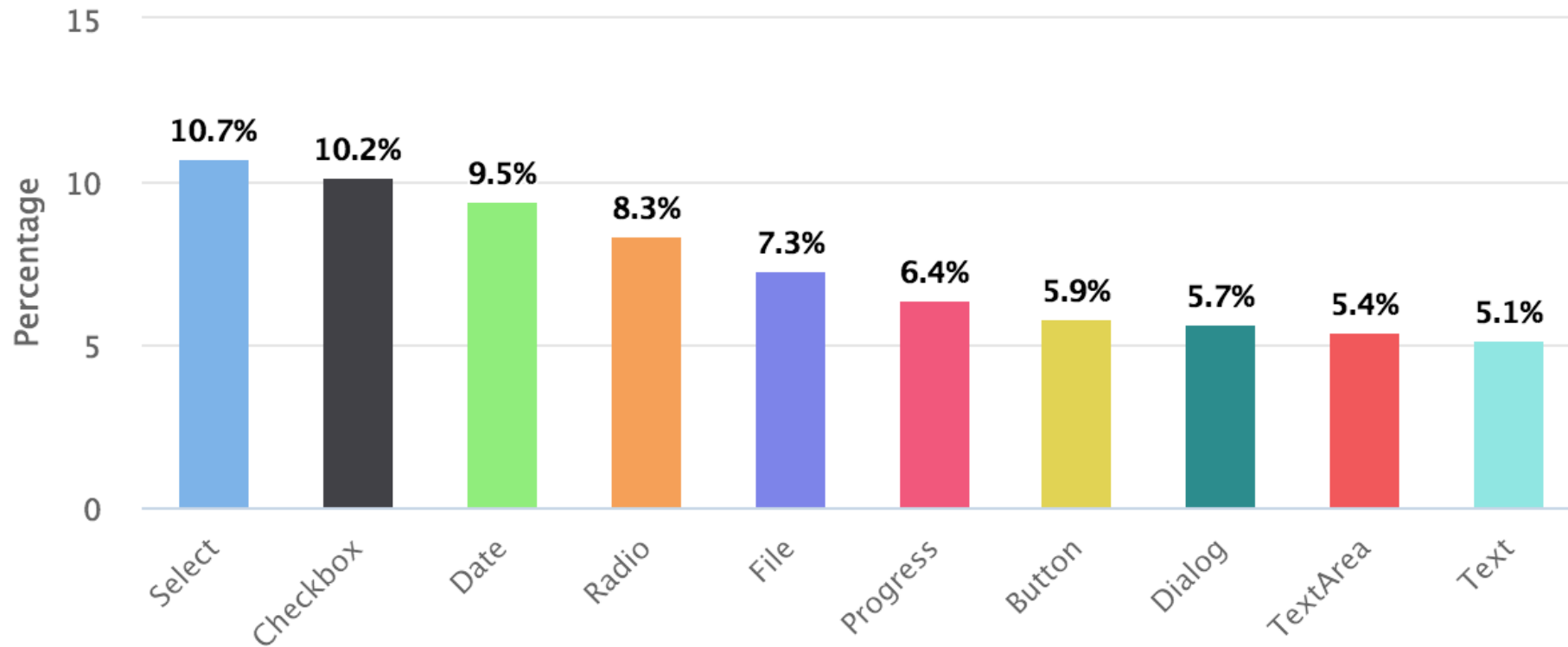
Demographic of respondents



Initial Research: Demographics



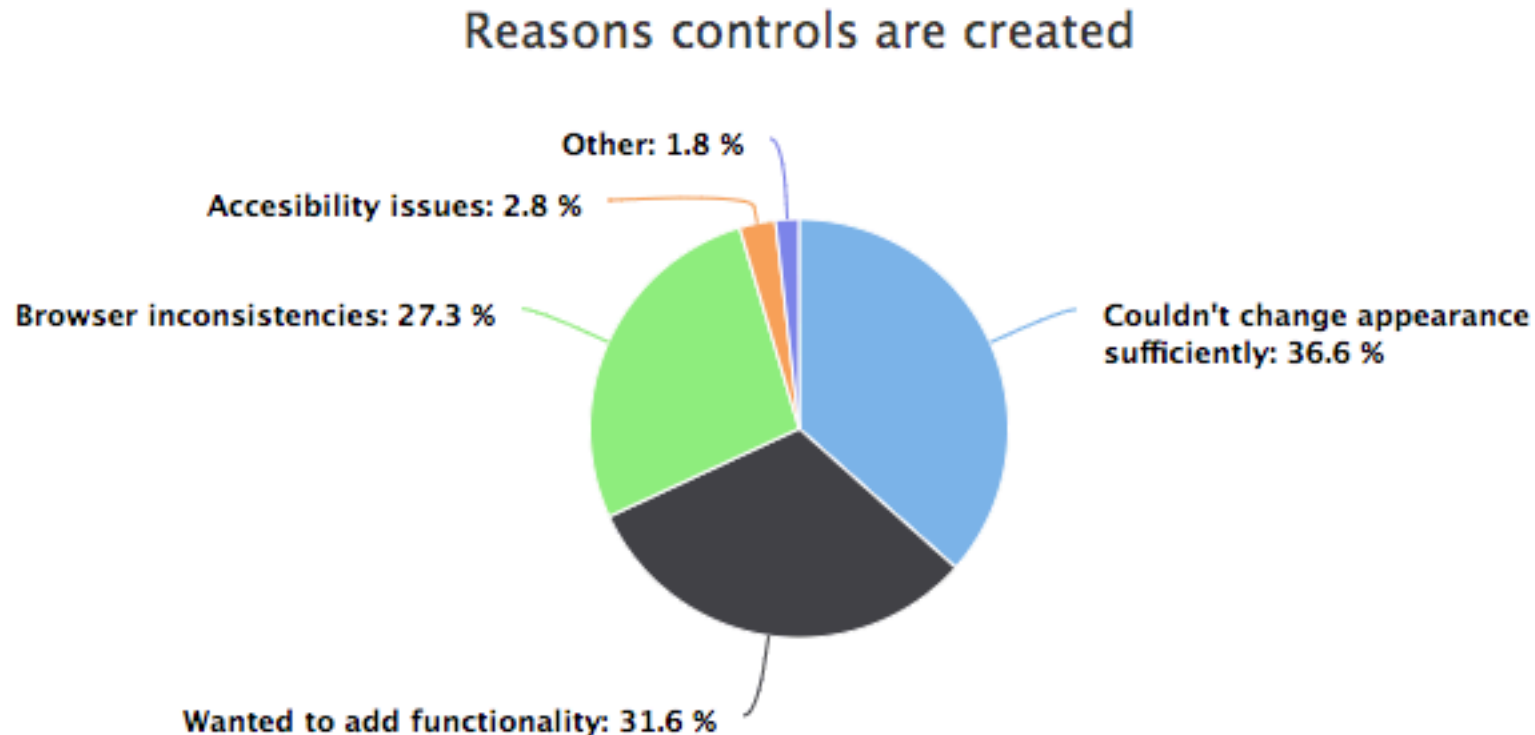
Top 10 Re-created Form Controls



Source: gwhitworth.com

Full results: <https://aka.ms/controls-survey>

Reasons Controls Are Created



Source: gwhitworth.com

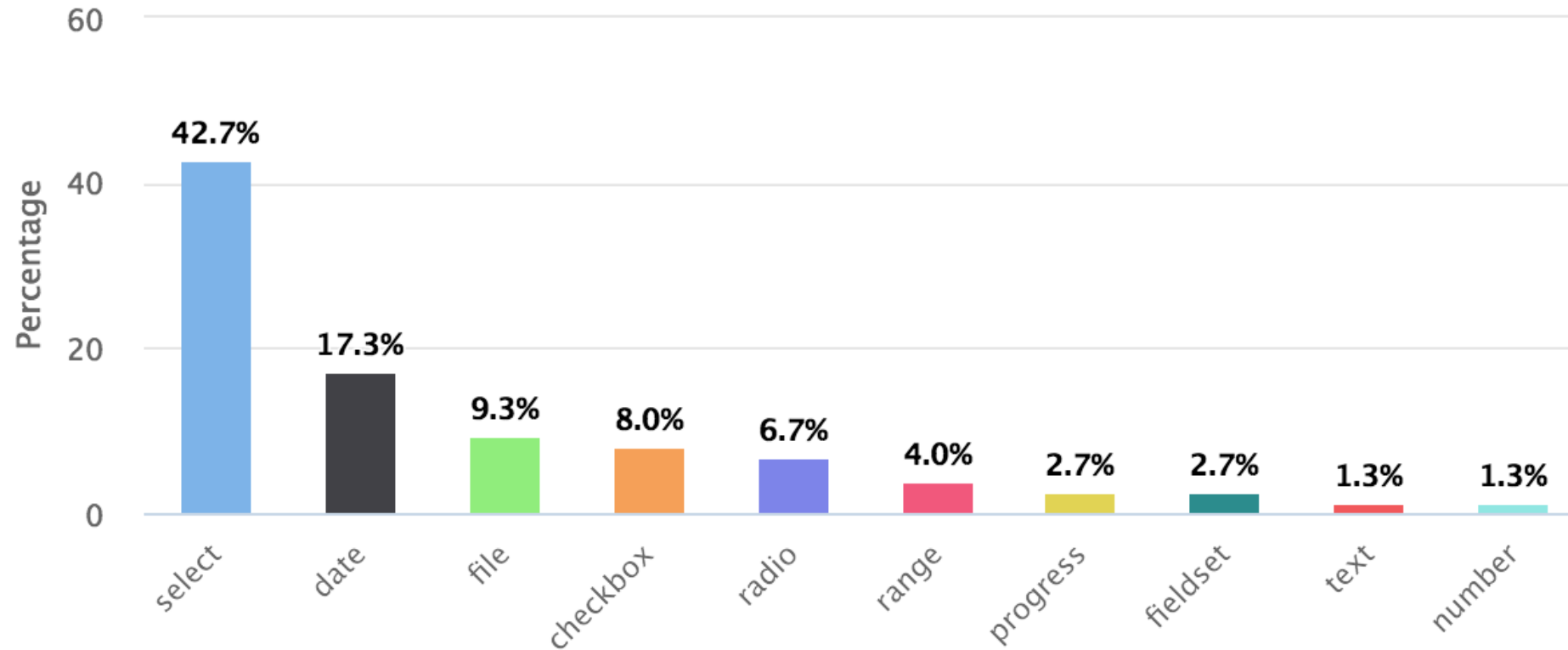
JSConfEU Survey

Which form control gives you the most frustration?

Why?

JSConfEU Survey

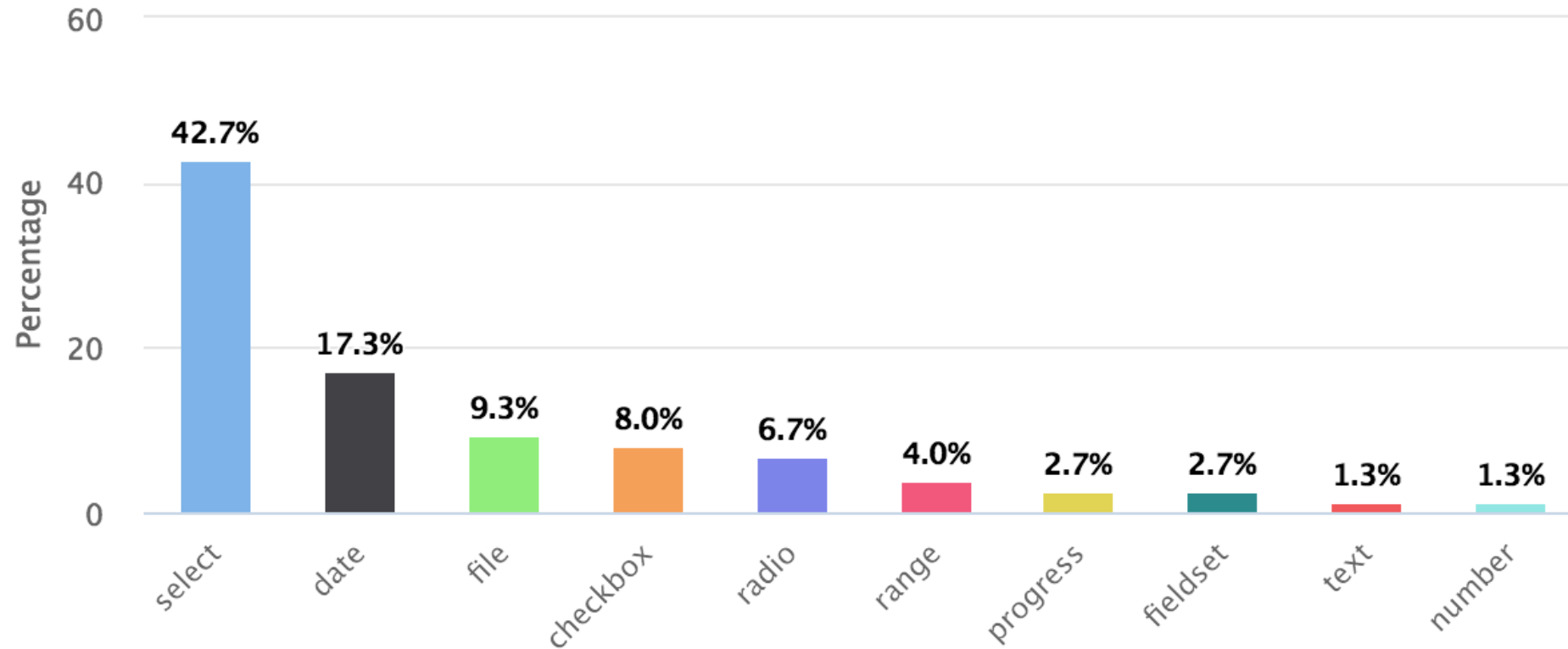
Which form control gives you the most frustration?



Source: gwhitworth.com

JSConfEU Survey

Why?



Source: gwhitworth.com

“

Requires
hacky tricks



Can't
style `<option>`
elements at all
to the extent we
need to



**...but the amount of
work it takes to
implement an
accessible alternative
with complete feature
parity is massive.**

How painful is it?

Important™ Research



Stephanie Stimac 🧙 Casting Spells
@seaotta



Dear devs and front-end designers, important research for a talk I'm giving: please fill in the blank:

"I would rather _____ than attempt to style a native `<select>` element"

6:40 PM · Jan 6, 2020 · [Twitter Web App](#)





dilski  @dilski · Jan 7

Replying to @seaotta

Call each person attempting to use the form and ask them what option they would like

1



36



Hagbard Celine @asciidisco · Jan 6

Replying to @seaotta

Convince my fellow programmers to reinstall Silverlight.



2



Martin Adams @Martin_Adams · Jan 6

Replying to @seaotta

Build the entire site in Flash

2



212



Ed Henderson @edhenderson · Jan 6

Replying to @seaotta

Support IE6



1



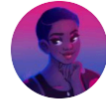
marty  @Marty331 · Jan 7

Replying to @seaotta

I would rather use vim.



1



Keziah @KeziahL · Jan 7

Replying to @seaotta

Breakdance barefoot on a pile of Legos

1



12



Ben Horst @benhorst · Jan 6

Replying to @seaotta

Grind my teeth to the jawbone with rough sandpaper



1



Iarf @Iarf2k · Jan 7

Replying to @seaotta

chew on glass



1



Chelsea Adams @thecityinspeech · Jan 7

Replying to @seaotta

Boil my toes in lava



1



Mike Francis @_mikefrancis · Jan 6

Replying to @seaotta

Maybe a bit melodramatic but heat up a rusty old fork, with a few tines (yes I had to google that) snapped off and broken, then with both arms thrust it into my inner thigh



2



**Clearly, there is
an issue**

The Future

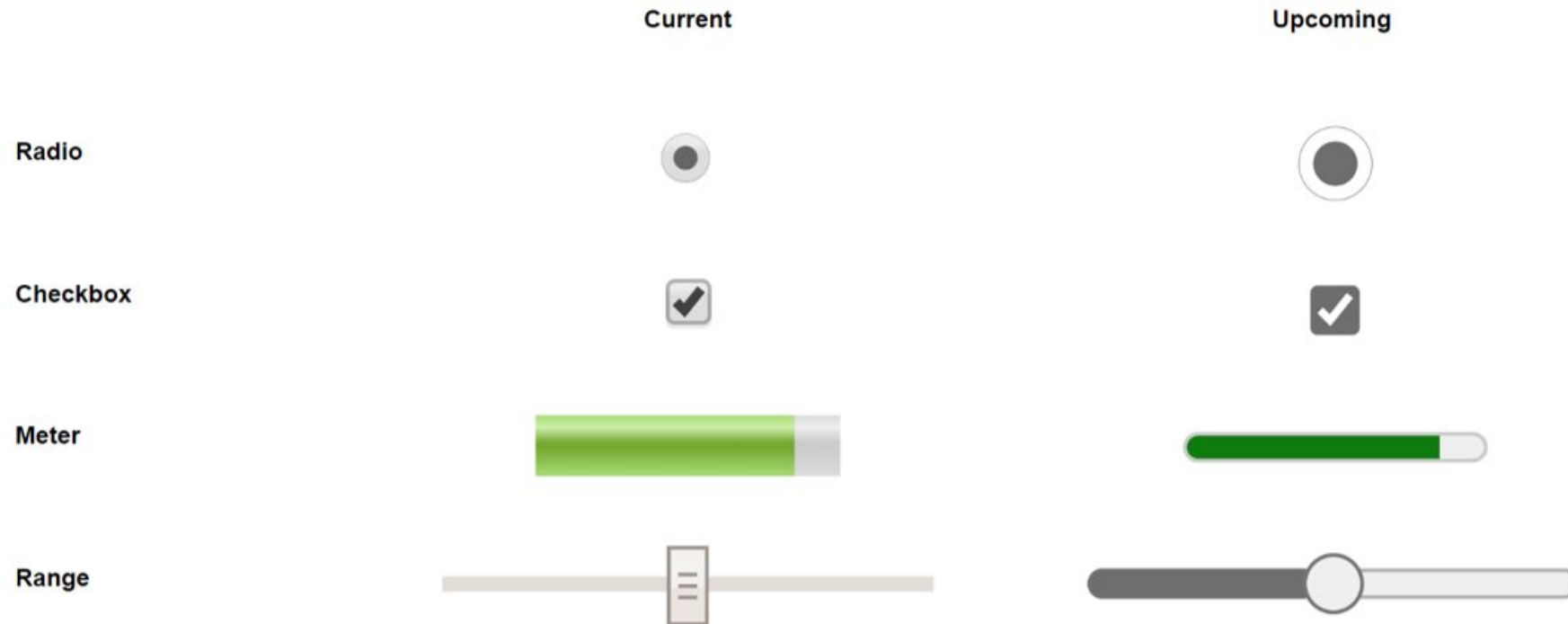


HTML isn't done



New Styles & Accessibility Improvements

Chromium Controls Visual Refresh



New Native Components

New native components

<virtual-list>

<https://aka.ms/virtual-scroller>

<toggle-switch>

<https://aka.ms/toggle-switch>

New native components

<virtual-list>

More info:

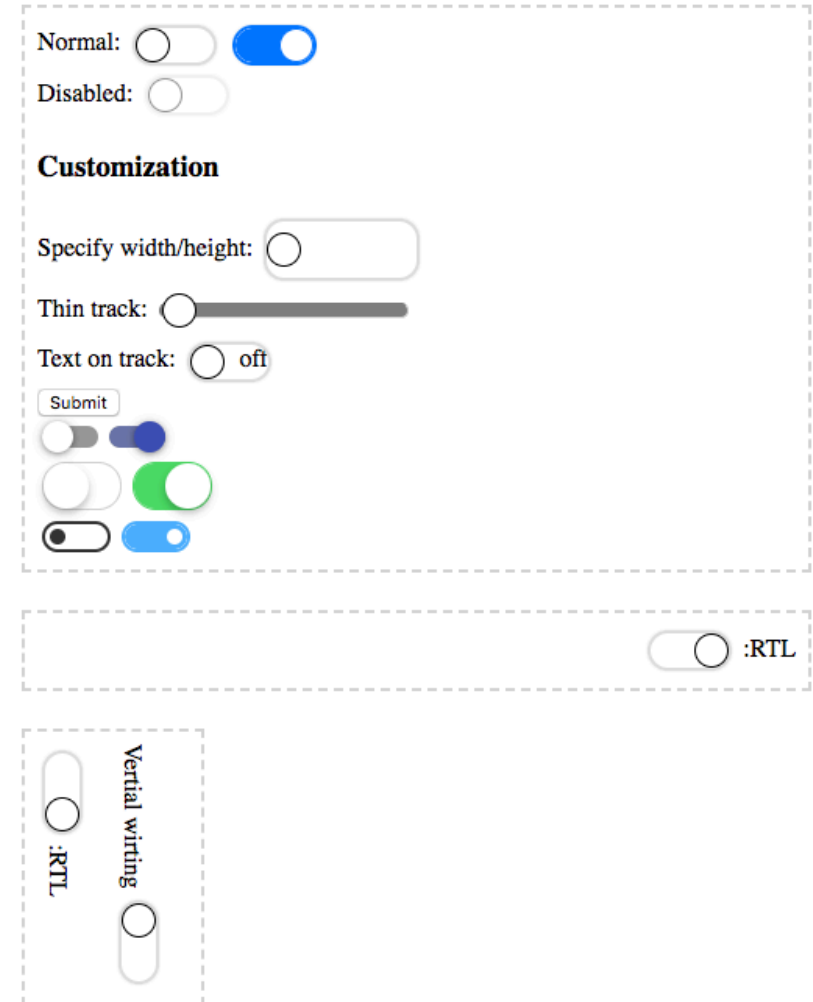
<https://aka.ms/virtual-scroller>

New native components

`<toggle-switch>`

More info + links to
prototype:

<https://aka.ms/toggle-switch>



From proposal to standard

1. Proposal incubates in the WICG
2. After incubation, multi-implementer interest is required before graduating to an HTML standard

Tell other implementers what you want!

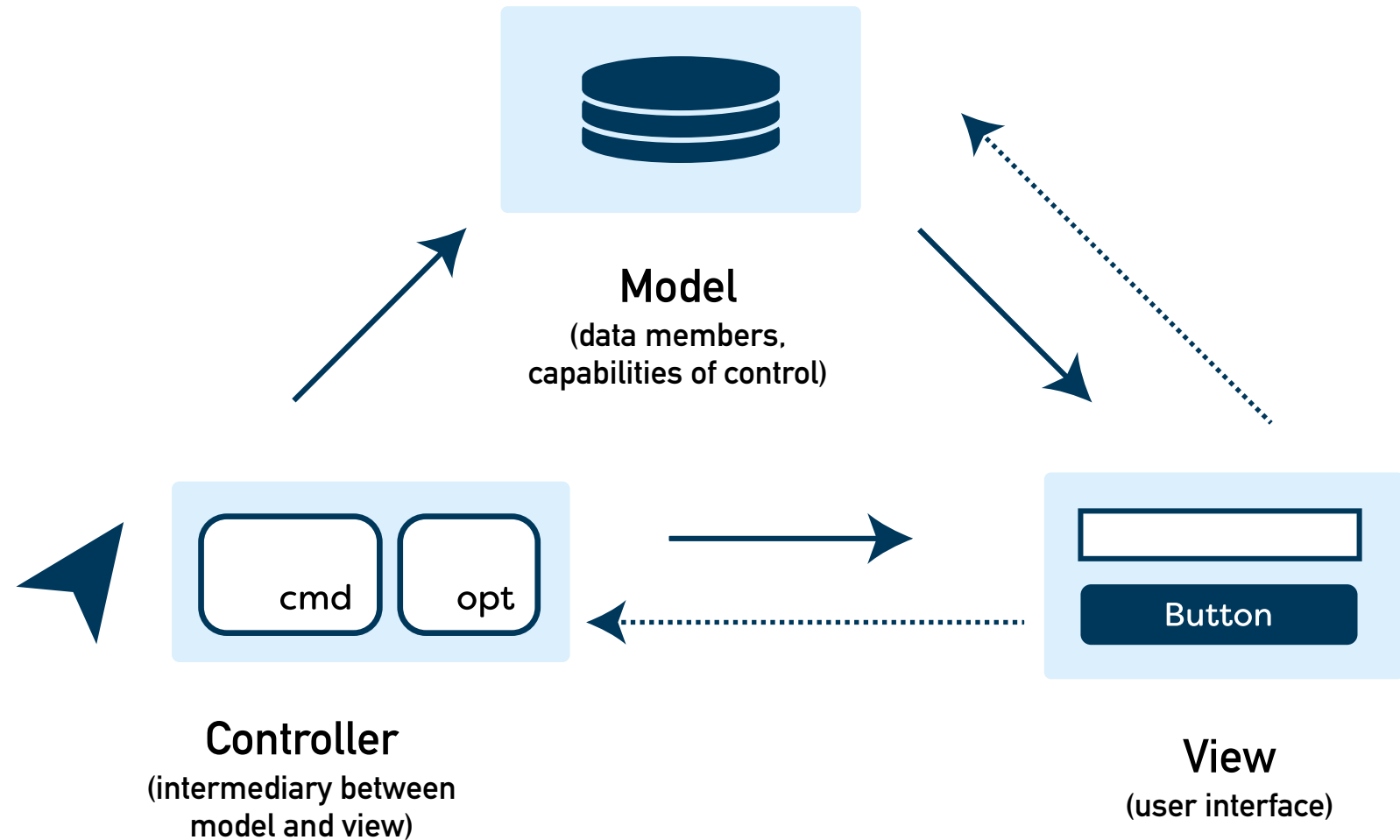
1. @moz hacks on Twitter
2. Mozilla Firefox Platform Status:
<https://github.com/mozilla/platform-status>
3. @webkit on Twitter
4. WebKit Platform Status: <https://webkit.org/status/>

Fixing the current
problems with controls

Customizing Controls UI

Customizing Controls UI Explainer

MVC Design Pattern



Goals

- Style any arbitrary part of a native control
- Add arbitrary content into a control
- Style a particular part without rewriting the whole UI
- Customize UI without reimplementing the data model and code for reacting to user input
- Customized controls are accessible by default

3 Customization Options

- Using standardized parts and states to override the native control styles via pseudo-classes and pseudo-elements.
- Using named <slots> to replace parts of the native control UI with developer content, while optionally leaving some native parts in place.
- Replacing the entire UI of the native control with an author-supplied shadow root.

Solution 1:

**Standardized
Control UI anatomy, parts
and behavior**

Solution 1:

**Standardized
Control UI
anatomy,
parts and
behavior**



aka.ms/open-ui

Goals of Open UI

- Document component names as they exist today
- Define a common language for describing UIs and design systems
- Eventual browser standards for web app components
- Converging designer processes and developer workflows

<select>
investigation

[Home](#)[Charter](#)[Contribute](#)[Analysis](#)[Button](#)[Card](#)[Component Name Matrix](#)[Datepicker](#)[Skeleton](#)[Slider](#)[Switch](#)[Text](#)[Components](#)[Checkbox \(Working Draft\)](#)[Table \(Editor's Draft\)](#)[Select \(Editor's Draft\)](#)

Select (Editor's Draft)

[View Research](#)

Overview

The `<select>` is a control that provides a list of options for the user to select from.

Use Cases

The `<select>` control is primarily leveraged to select an option for within a form. For example when your buying a shirt you may be provided with a `<select>` that has options for sizes that you then select the appropriate one for you.

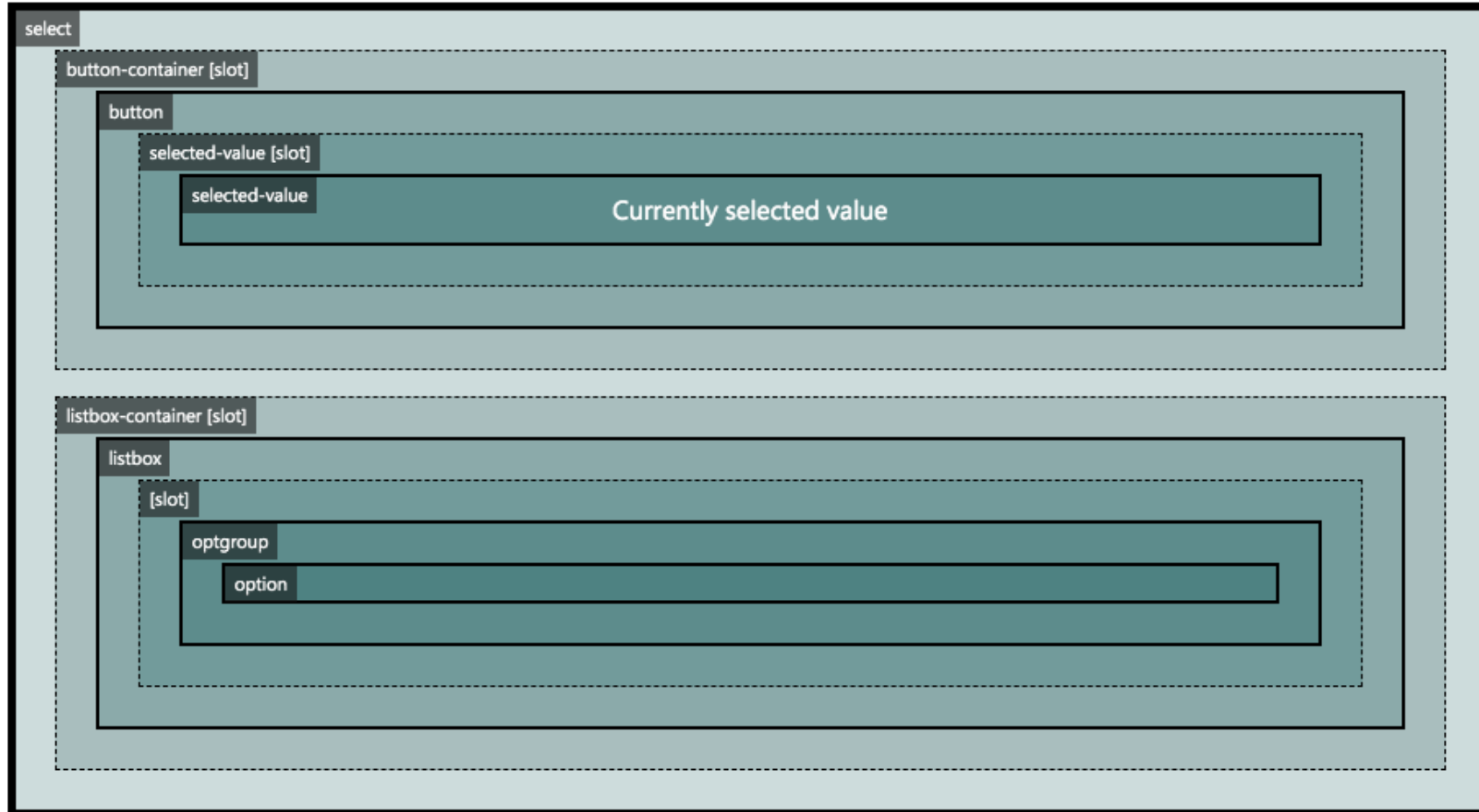
Prior Art/Examples

- [Ant Design](#)
- [Carbon Design System](#)
- [FAST DNA](#)
- [Atlas Kit](#)

`<select>` Properties

Attribute Name	Type	Default Value	Description
<code>autocomplete</code>	<code>string</code>	<code>off</code>	Allows the developer to provide a hint on how to search the content within the <code><option></code> (s)
<code>autofocus</code>	<code>bool</code>	<code>false</code>	If set to <code>true</code> the <code>input</code> will have focus set on page load
<code>disabled</code>	<code>bool</code>	inherited from containing element	If set to <code>true</code> the user will not be able to interact with the control
<code>form</code>	<code>string</code>	the form that contains the	Represents the form owner, via the <code>form</code> attribute being

The anatomy of a `<select>`



```
<style>
  .styled-select::part(button) {
    background-color: red;
  }
</style>
```

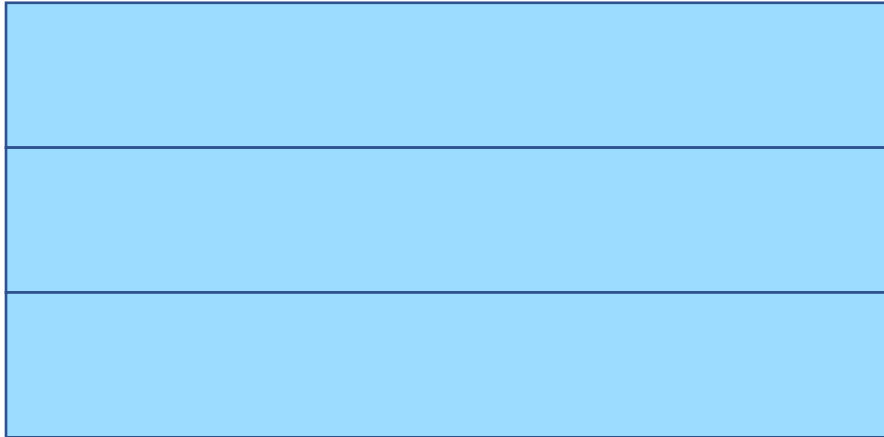
```
<select class="styled-select">
  <option>choice 1</option>
  <option>choice 2</option>
</select>
```

```
<style>  
  .styled-select:open::part(button) {  
    background-color: lightgray;  
  }  
</style>
```

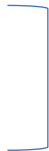
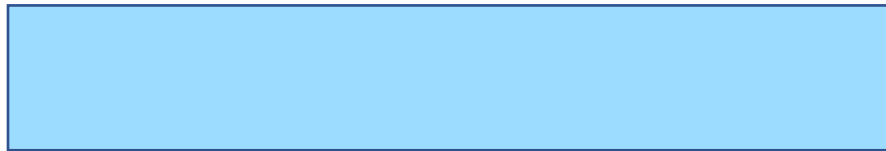
Solution 2:

Named Slots

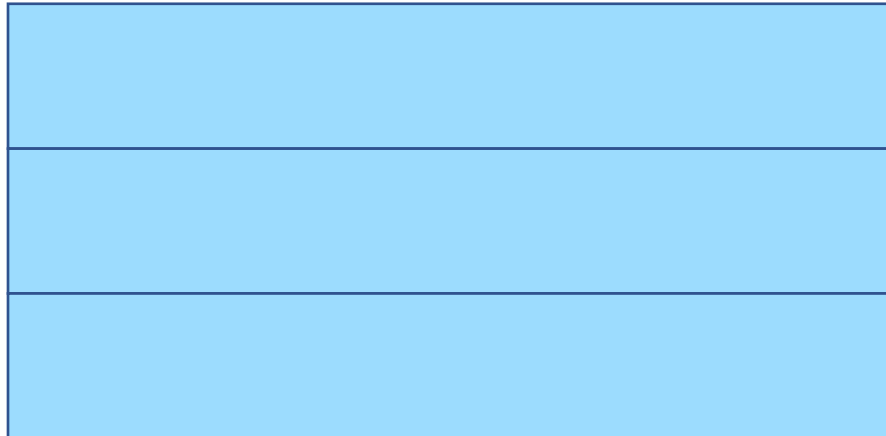
Standardized Slots



Standardized Slots



slot=button



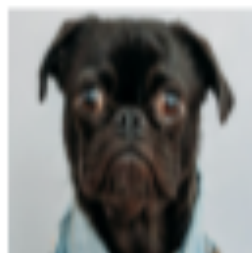
slot=listbox

```
<select>
  <div slot="button" part="button" class="custom-button">Choose a pet</div>
  <div slot="listbox" part="listbox" class="custom-listbox">
    <option>
      
      <div class="option-text">Cat</div>
    </option>
    <option>
      
      <div class="option-text">Dog</div>
    </option>
  </div>
</select>
```

Choose a pet



Cat



Dog

Customize selectively

```
<input type="range">
```

Customize selectively

```
<range>  
  <div slot="thumb" part="thumb">  
    <svg><!-- Use SVG to draw the thumb icon... --></svg>  
  </div>  
</range>
```

Parts and slots

Native controller code + user-provided UI Parts

Web Platform
Code

Native event
handlers

Part
Attributes /
element types

Solution 3:

Shadow DOM replacement

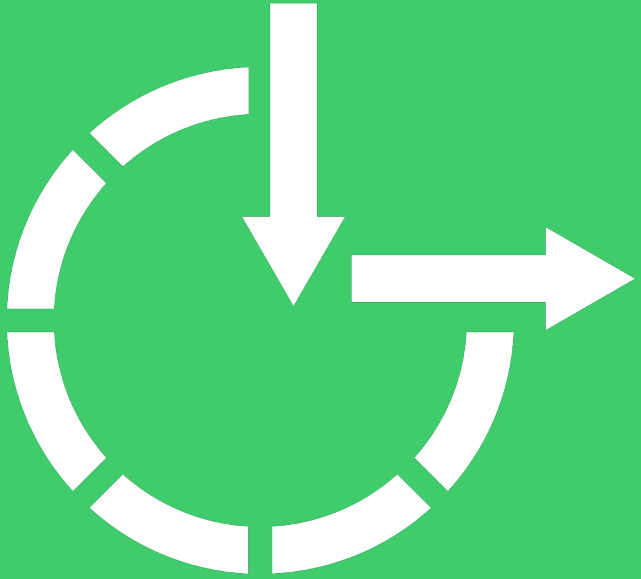
attachShadow()

~~attachShadow()~~

```
let customSelect = document.createElement('select');
customSelect.setAttribute("custom", "");
let selectShadow = customSelect.attachShadow({ mode: 'open' });
selectShadow.innerHTML = `My custom select UI`;
document.body.appendChild(customSelect);
```

Don't forget your parts!

```
part="button"
```



Accessibility



Enabling Custom Control UI

Authors:

- [Dan Clark](#) (Microsoft)
- [Bo Cupp](#) (Microsoft)
- [Mason Freed](#) (Google)
- [Ionel Popescu](#) (Microsoft)
- [Melanie Richards](#) (Microsoft)
- [Greg Whitworth](#) (Salesforce)

Status of this Document

This document is intended as a starting point for engaging the community and standards bodies in developing collaborative solutions fit for standardization. As the solutions to problems described in this document progress along the standards-track, we will retain this document as an archive and use this section to keep the community up-to-date with the most current standards venue and content location of future work and discussions.

- This document status: **Active**
- Expected venue: [W3C Web Incubator Community Group](#)
- Current version: this document

Introduction

<https://aka.ms/controls-explainer>

We need you!

We need you!

- Provide feedback on the current HTML Control prototypes that Chrome has
- Contribute to the form control investigations on Open UI
- Tell browser vendors what you need from your form controls

Follow these folks

- @SoMelanieSaid – Microsoft Edge PM
- @gregwhitworth – Salesforce / Open UI
- @stubbornella – Google Chrome PM
- @seaotta – Microsoft Edge PM

We're here to listen...

...because these
improvements are for you.

Thank you!

<https://noti.st/seaotta>

@seaotta

ststimac@microsoft.com

