Web components in Django

Xavier Dutreilh

DjangoCon Europe 2014

Who am I?

web developer

(not a *front-end guy*, not a *back-end guy*)

web developer

(neither a *Pythonist* nor a *Djangonaut*)

lead front-end
developer
at polyconseil

(autolib.eu, its variants and
many internal applications)

What is the problem?

Building front-ends
is hard and complex.

languages

(like html, css and javascript)

libraries

(like underscore.js, jquery and d3.js)

frameworks

(like bootstrap, handlebars and angularjs)

tools

(like bower, grunt, yeoman and jshint)

performance

(scalable code, smallest footprint)

usability

(usable by everyone, no dark UX patterns)

accessibility

(accessible to everyone, wai and aria)

cross-browser functionality

(like chrome, firefox and internet explorer)

cross-platform functionality

(like windows, os x and android)

cross-device functionality

(desktops, laptops, mobiles, tablets)

How do we face it?

We ignore everything that I just said.

We look for ready-to-use frameworks and libraries.

We pile them one over another.

We add a ton of glue code.

We hope that everything is going to work *as expected*.

Does it really work?

It seems so
but it is not.

technical debt

(hydra code)

user adaptation

(compliance with the code)

What can we do?

Stop creating mess.

Think about requirements.

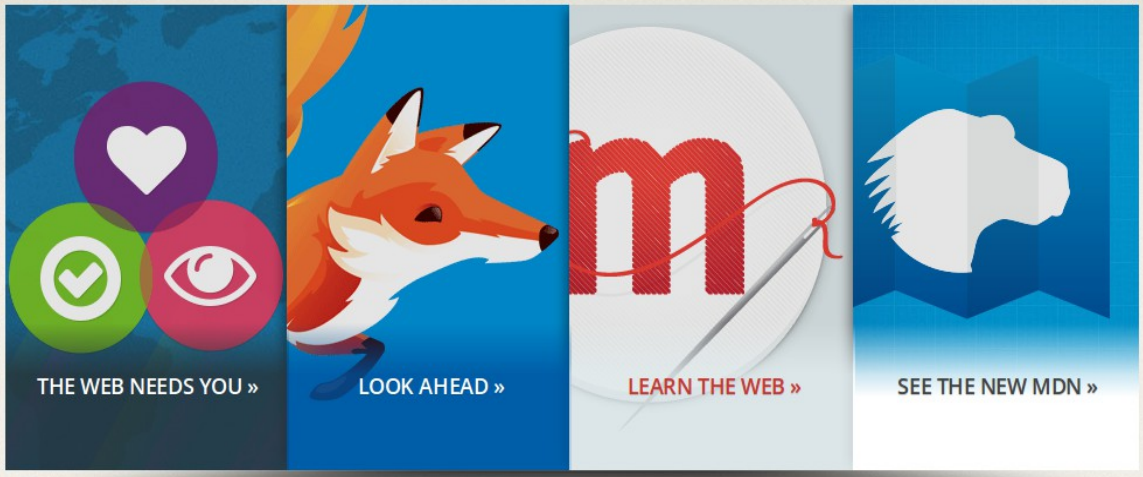
Create reusable elements.

Use web components.

What is a web page?

We are mozilla

Doing good is part of our code



Firefox | Different by design

FREE DOWNLOAD
English (US) • Linux 64-bit

Systems & Languages | What's New | Privacy

In the news

Firefox OS Unleashes the Future of Mobile »

See all news »

Get Mozilla updates

YOUR EMAIL HERE

France

I'm okay with Mozilla handling my info as explained in this Privacy Policy

Sign me up »

mozilla

Portions of this content are ©1998–2014 by individual mozilla.org contributors. Content available under a Creative Commons license.

Contact Us · Partner with Us
Donate · Firefox Affiliates
Contribute to this page

Privacy Policy · Legal Notices
Report Trademark Abuse

Mozilla: Twitter · Facebook
Firefox: Twitter · Facebook · YouTube

Other languages: English (US)

```
<!doctype html>
<html class="windows x86 no-js" lang="en-US" dir="ltr">
  <head>
    <meta charset="utf-8">
<!--
                _.-~-.
            7''  Q..\
          _7         (_
        _7  _/     _q.  /
      _7 .  ___   /VVvv-'_                                          .
     7/ / /~- \_\\      '-._      .-'                        /      //
    ./ ( /-~-/||'=.__   '::. '-~'' {                  ___    /  //     ./{
  V   V-~-~| ||    __''_    ':::.    ''~-~.___.-'' _/  // / {_   /  {  /
   VV/-~-~-|/ \ .'__'. '.     '::                          _ _ _          ''.
   / /~~~~||VVV/ /  \ )  \              _ __ ___   ___ ___(_) | | __ _    .::'
  / (~-~-~\\.-' /    \'   \:::::. | '_ ` _ \ / _ \_  / | | |/ _` | :::'
 /..\    /..\__/      '     ':::: | | | | | | (_) / /| | | | (_| | ::'
 vVVv    vVVv                 ': |_| |_| |_|\___/___|_|_|_|\__,_| ''

Hi there, nice to meet you!

Interested in having a direct impact on hundreds of millions of users? Join
Mozilla, and become part of a global community that's helping to build a
brighter future for the Web.

Visit https://careers.mozilla.org to learn about our current job openings.
Visit https://www.mozilla.org/contribute for more ways to get involved and
help support Mozilla.-->

        <meta name="viewport" content="width=device-width, initial-scale=1">
    <!-- validates bing webmaster tools -->
<meta name="msvalidate.01" content="B7B177115A634927D608514DA17B2574" />
<!-- YouTube Verification -->
```

It is a document embedding elements.

What is an element?

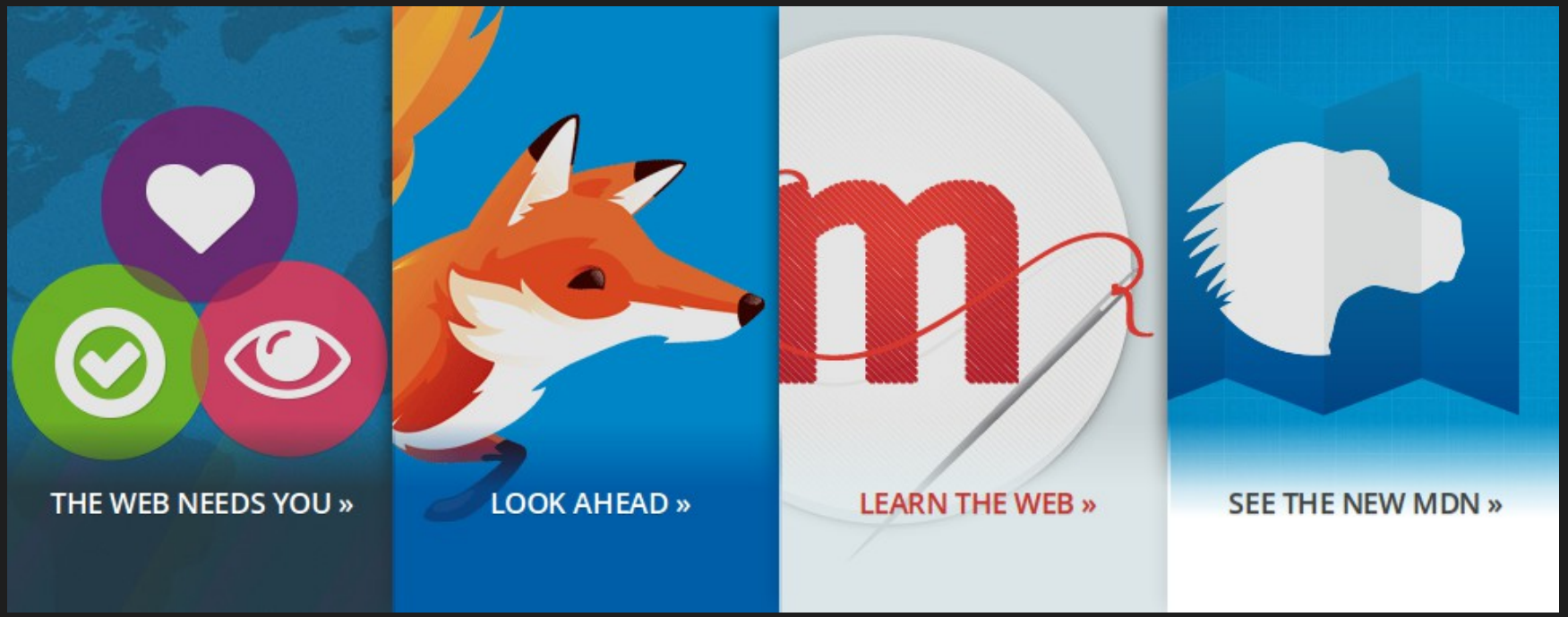
```
<select name="country">
  <option value=""></option>
  <option value="BE">Belgium</option>
  <option value="FR">France</option>
  <option value="DE">Germany</option>
  <option value="IT">Italy</option>
  <option value="ES">Spain</option>
</select>
```

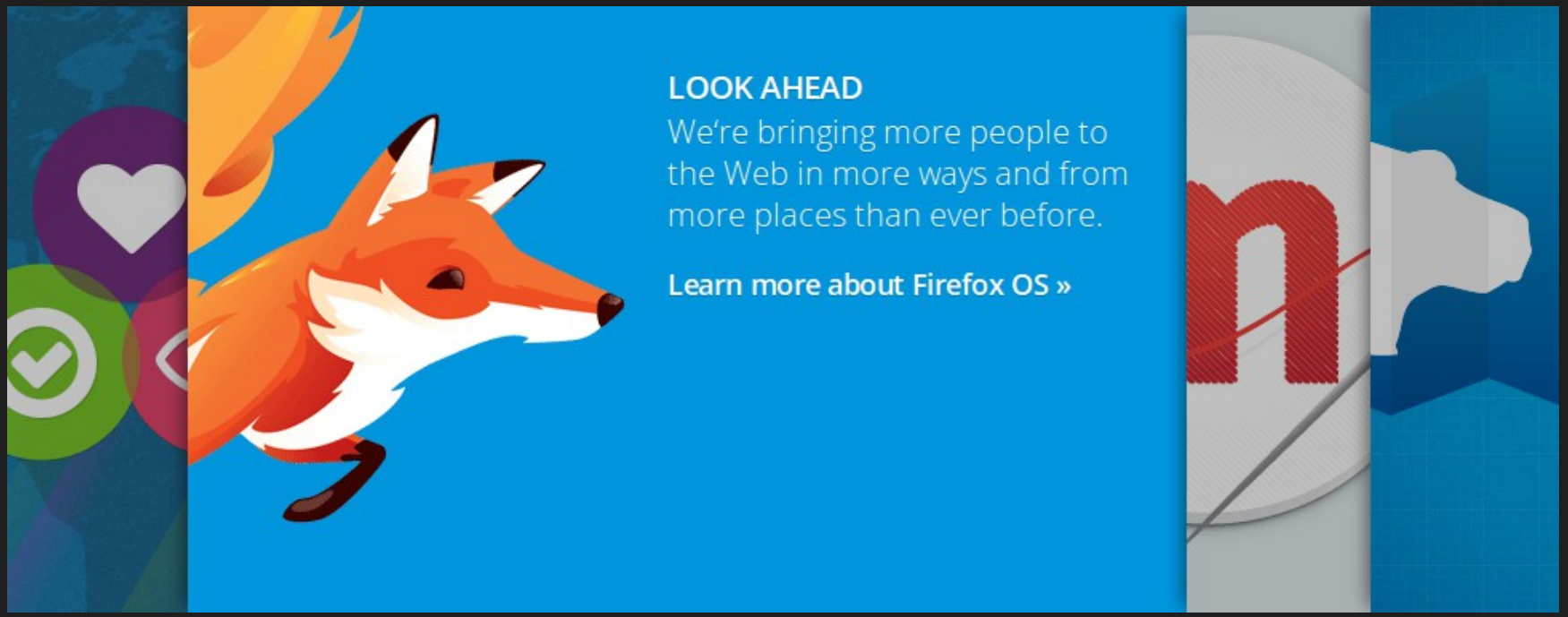
Elements are defined by the specification.

Custom elements cannot be added.

What happens when we need custom elements?

We hack with HTML, CSS and JavaScript.


THE WEB NEEDS YOU »
LOOK AHEAD »
LEARN THE WEB »
SEE THE NEW MDN »


LOOK AHEAD
We're bringing more people to the Web in more ways and from more places than ever before.
Learn more about Firefox OS »

```
<ul class="accordion">
  <li class="panel">
    <h2 class="panel-title">Title</h2>
    <div class="panel-inner">
      <div class="panel-content">
        <p>Text</p>
      </div>
    </div>
  </li>
  ...
</ul>
```

```
jQuery('.panel').on('click focus', function() {
  if (!jQuery(this).hasClass('expanded')) {
    if (horizontal) {
      panel.expandHorz(jQuery(this));
    } else {
      panel.expandVert(jQuery(this));
    }
  }
});
...
```

What are web components?

It is a set of emerging standards to extend HTML.

templates

(markup defined into <**template**/>)

```
<template id="accordion-template">
  <style>...</style>
  <ul class="accordion">
    <content></content>
  </ul>
</template>
```

```
<template id="panel-template">
  <style>...</style>
  <li class="panel">
    <content></content>
  </li>
</template>
```

```
<template id="panel-heading-template">
  <style>...</style>
  <h2 class="panel-title">
    <content></content>
  </h2>
</template>
```

```
<template id="panel-content-template">
  <style>...</style>
  <div class="panel-inner">
    <div class="panel-content">
      <p>
        <content></content>
      </p>
    </div>
  </div>
</template>
```

custom elements

(elements created with <**element**/>
or the javascript api)

```
<element name="accordion">
</element>

document.createElement('accordion');

document.registerElement('accordion');
```

shadow dom

(isolated subtree, javascript api)

```
var proto = Object.create(
  HTMLElement.prototype, {
    createdCallback: {
      value: function() {
        var s = '#accordion-template';
        var t = document.querySelector(s);
        this.createShadowRoot().appendChild(
          t.content.cloneNode(true);
        );
      }
    }
  }
);
```

```
var myPanel = document.registerElement(
  'panel', { prototype: proto });
```

html imports

(loading via <**link**>)

```
<link rel="import" href="accordion.html">
```

How can we use web components?

```
<link rel="import" href="accordion.html">

<accordion>
  <panel>
    <panel-heading>Title</panel-heading>
    <panel-content>Text</panel-content>
  </panel>
  ...
</accordion>
```

Is it ready yet?

The specification is still a draft.

But browser-vendors started to implement web components.

Templates

Custom elements

Shadow DOM

HTML imports

But it is not there yet.

polyfills

(polymer and x-tag)

polymer

(polyfills, custom elements,
reusable elements)

Templates

Custom elements

Shadow DOM

HTML imports

x-tag

(polyfills, custom elements and brick)

Templates

Custom elements

Shadow DOM

HTML imports

What are the tools?

The tools are the same as when building any other applications and widgets.

package manager

(bower, component, jam, volo, browserify)

libraries and frameworks

(jquery, angularjs, ember.js, backbone.js)

preprocessors

(coffeescript, sass, less, stylus)

testing tools

(karma, jasmine, mocha, sinon.js, zombie.js, protractor)

quality tools

(jshint, csslint)

Does it work good with Django?

Yes but the integration still remains basic.

tooling

(mostly node.js)

application structure

(web components vs django)

data sharing

(restful api)

security

(authentication, csrf, cors, local storage)

That is it.

What do we do next?

Do yourself a favor and try front-end development.

Review your web applications and improve your front-ends.

Share your knowledge and skills with the front-end community.

Hire people to build your front-ends.

Thank you!

Xavier Dutreilh
xavier@dutreilh.com
http://xavier.dutreilh.com/