HTML 5 Workshop

Nathaniel T. Schutta
@ntschutta

The Plan

- What is HTML 5?
- Why do I care?
- What can I do?
- How do I do it?

Assume you have:
browser, text editor.

Assume you have:
browser, text editor.

Copy zip.

Or clone it from github.

git clone git://github.com/ntschutta/html5_workshop.git

Extract to...somewhere ;)

Shout if you have ??s

What is it?

There was an HTML 4?

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
        "http://www.w3.org/TR/html4/loose.dtd">
```

Remember XHTML?

Lack of features.

Browsers are forgiving.

Web flourished
because of it.

Lots of “broken” pages.

HTML little long
in the tooth...

Draconian error handling.

And wasn’t designed
for applications.

Not backwards
compatible.

Pushing boundaries.

Syntax was adopted.

De facto standards...

XHR anyone?

Well not entirely.

Standards aren't always clear cut.

Evolve over time.

Can contradict.

A conversation.

Not a conspiracy against developers.

Browser implementors, designers, standardistas.

Often a reaction to what we're doing.

2004 W3C workshop.

We say we want standards...

What should we do?

Really want browser consistency.

Evolving HTML lost.

Pain isn't standards, it's implementation.

Formed a new group.

WHAT Working Group.

Canvas, audio, video tags.

Web apps!

Work continued on XHTML 2.0.

Reversed engineered, and documented, parsing.

You probably didn't notice.

Web forms.

WHAT WG had momentum.

W3C joined the effort.

Paving of cowpaths.

Thus was born HTML5.

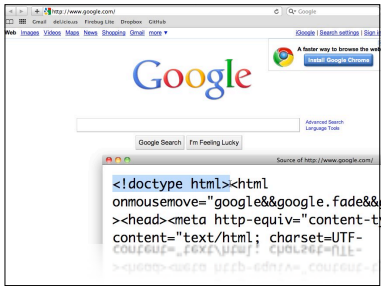
Evolutionary step.

So what is it again?

Why should I care?

HTML5 is a collection of features...

Market is moving.



Next big thing.

Millions of mobile devices...more daily.

Customers.

Mobile.

i* - iPad, iPhone etc.

Not just Apple...

Great! HTML5.

Don't want to build a native app?

Mobile browsers excellent HTML5 support.

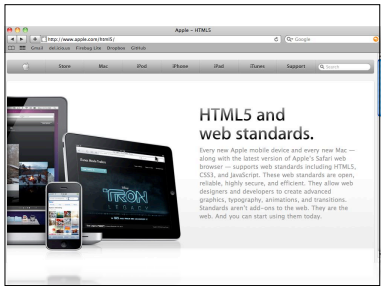
Don't want to support multiple OS?

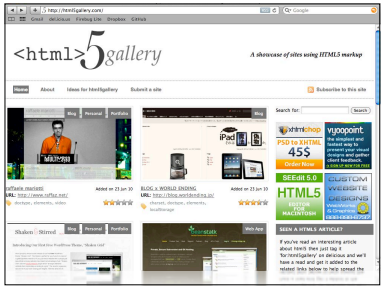
More likely than desktop peers.

Android fragmentation, Black Berry, Palm...

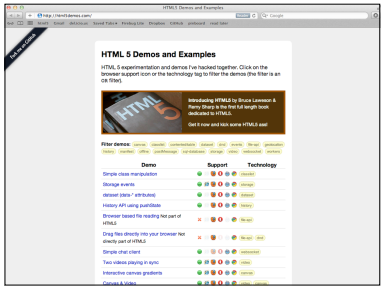
Flash is dying.

Lead largely by Apple and...Google.





YouTube



HTML5 gaining steam.

Customers want sites that work on today's devices.

HTML5 is the answer.

What can I do?

Respects what we're actually doing.

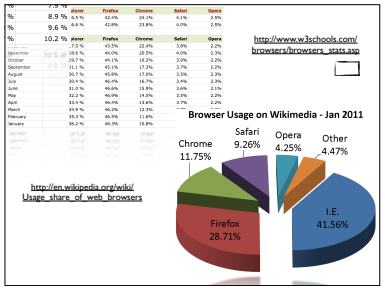
Browser support isn't universal.

Forms, controls... well worn hacks.

Shocking.

Helps us build applications!

Older browsers pervasive.



Google Apps...sorry IE 6.

http://googledocs.blogspot.com/2010/01/web-browser-support-for-docs-and-sites.html

IE 6 is dying...

Agent sniffing?

Even MS wants it gone.

NO!

SharePoint 2010 - no IE 6.

http://blogs.msdn.com/sharepoint/archive/2009/05/07/announcing-sharepoint-server-2010-preliminary-system-requirements.aspx

Browsers lie...

This browser's user agent:
Mozilla/5.0 (Macintosh; U; Intel Mac OS X

This browser's user agent:
Mozilla/5.0 (Macintosh; Intel Mac OS X

This browser's user agent:
Mozilla/5.0 (Macintosh; Intel Mac OS X

This browser's user agent:
Opera/9.80 (Macintosh; Intel Mac OS X

And you can always mimic another user agent...

Feature detection.

The DOM is your answer key.

Four approaches.

1. Ask the global object if a property exists.

```
function supports_geolocation() {
    return !!navigator.geolocation;
}
```

Create an element:

2. Look for a property.

```
function supports_canvas() {
    return !!document.createElement('canvas').getContext;
}
```

3. Look for a method.

```
function supports_video() {
    return !!document.createElement('video').canPlayType;
}
```

4. Set a property and see if the value sticks.

```
function supports_input (input_type) {
    var input = document.createElement("input");
    input.setAttribute("type", input_type);
    return input.type !== "text";
}
```

Modern browsers are evolving with spec.

Support is quite good.

Lab time!

Feature Detection

- Using the four detection techniques, add the proper code to the four empty methods
- ${extract}/html5_workshop/labs/detection.html

Modernizer.

http://www.modernizr.com/

```
function check_geo_modernizer() {
    if (Modernizr.geolocation) {
        var message = function supports_geolocation() {
            return !!navigator.geolocation;
        }
    } else {
        message = no_support_mod;
    }
    show_results(message);
}
```

```
function check_canvas_modernizer() {
    if (Modernizr.canvas) {
        var message = "modernizer says you've got canvas!";
    } else {
        message = no_support_mod;
    }
    show_results(message);
}
```

```
function check_video_formats() {
    if (Modernizr.video) {
        var message = "yep, we can do some kind of video!";
        if (Modernizr.video.ogg) {
            message = "actually, we can play ogg";
        } else if (Modernizr.video.h264) {
            message = "we prefer 264 thanks";
        }
    }
    show_results(message);
}
```

Ogg? 264? WebM?

Patents...

http://daringfireball.net/2010/03/gif_h264_patents

http://daringfireball.net/2010/03/on_submarine_patents

Browser support falls on philosophical lines.

Not sure what your browser can do?

```
function check_for_email_modernizer () {
// ...
function supports_input (input_type) {
    var input = document.createElement("input");
    input.setAttribute("type", input_type);
    return input.type !== "text";
}
}
```

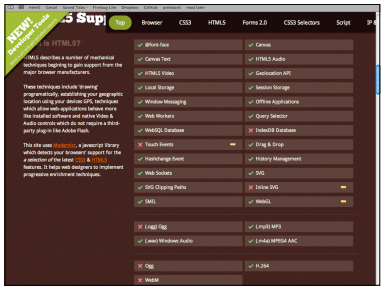
http://caniuse.com/

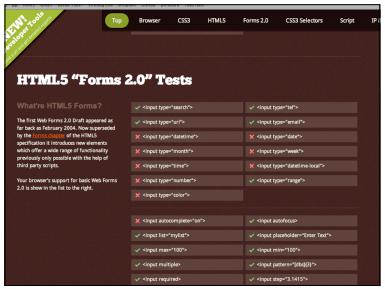
http://www.findmebyip.com/#target-selector

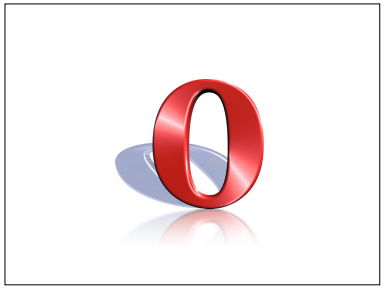
Lab time!

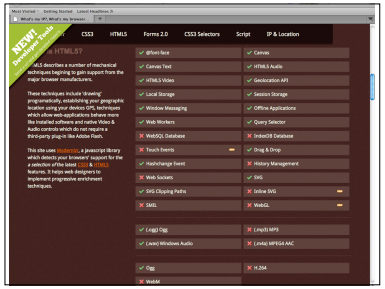
Feature Detection - 2

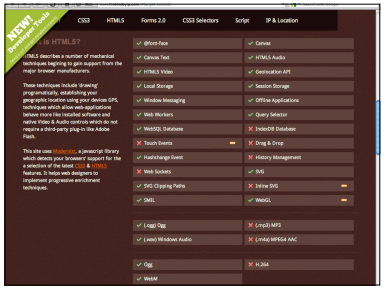
- Using Modernizr, add the proper code to the four empty methods
- ${extract}/html5_workshop/labs/detection_modernizr.html

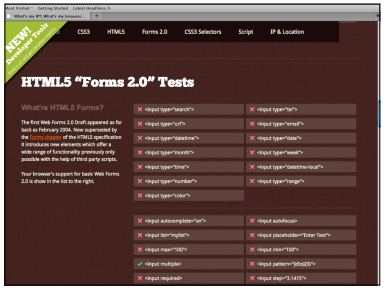


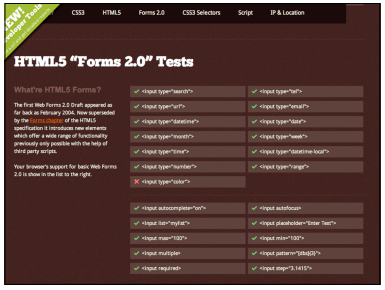

Top
Browser
CSS3
HTML5
Forms 2.0
CSS3 Selectors
Script
HTML5 “Forms 2.0” Tests
What’re HTML5 Forms?




CSS3
HTML5
Forms 2.0
CSS3 Selectors
Script
IP & Location

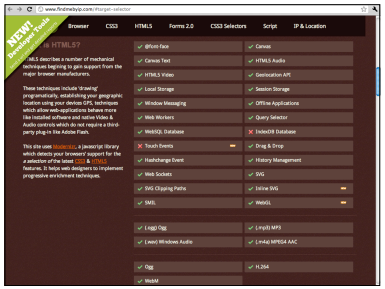

CSS3
HTML5
Forms 2.0
CSS3 Selectors
Script
IP & Location

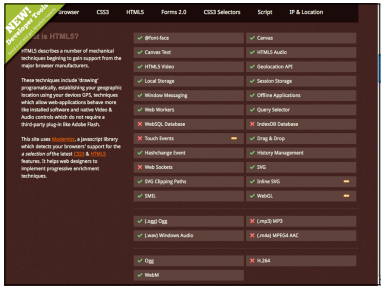

CSS3
HTML5
Forms 2.0
CSS3 Selectors
Script
IP & Location
HTML5 “Forms 2.0” Tests
What’re HTML5 Forms?

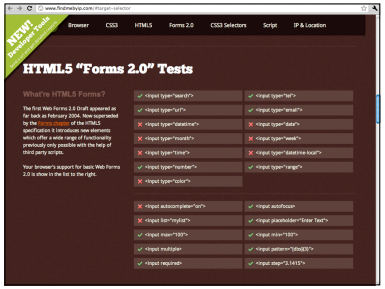

CSS3
HTML5
Forms 2.0
CSS3 Selectors
Script
IP & Location
HTML5 “Forms 2.0” Tests
What’re HTML5 Forms?

That’s 3.6.13...

Here's 4...

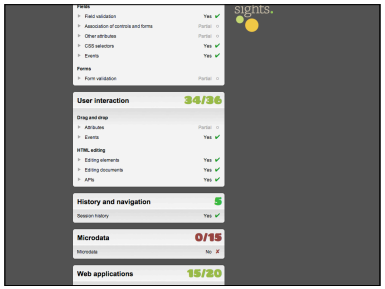


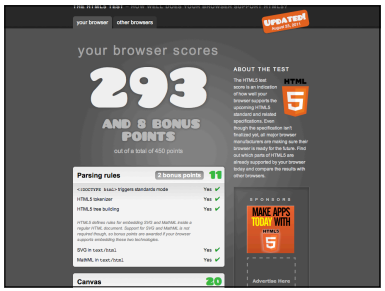


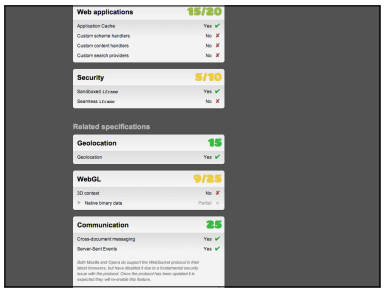


Another view...

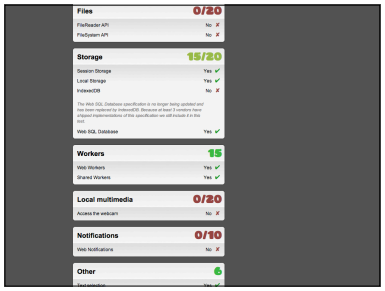
http://html5test.com/index.html

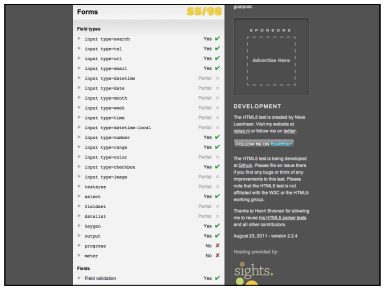

Forms
Field validation Yes
Association of controls and forms Partial
Other attributes Partial
CSS selectors Yes
Events Yes
Forms
Form validation Partial
User interaction 34/36
Drag and drop
Attributes Partial
Events Yes
HTML editing
Editing elements Yes
Editing documents Yes
APIs Yes
History and navigation 5
Session history Yes
Microdata 0/15
Microdata No
Web applications 15/20

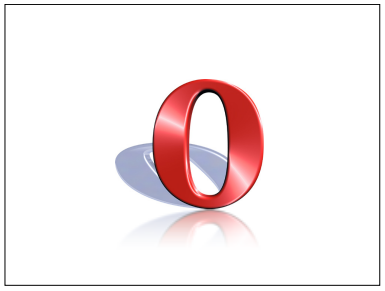

your browser
other browsers
UPDATED
your browser scores
293
AND 8 BONUS POINTS
out of a total of 450 points
ABOUT THE TEST
HTML 5
Parsing rules
3 bonus points
11
Canvas
20
SPONSORS
MAKE APPS TODAY WITH HTML5
Advertise Here

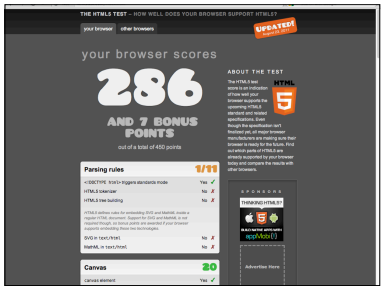

Web applications 15/20
Application Cache Yes
Custom scheme handlers No
Custom content handlers No
Custom search providers No
Security 5/10
Related specifications
Geolocation 15
Geolocation Yes
WebGL 9/25
3D context No
Native binary data Partial
Communication 25
Cross-document messaging Yes
Server-Sent Events Yes


Video
4 bonus points
21/31
video element Yes
Subtitle support No
Poster image support Yes
MPEG-4 support Yes
H.264 support Yes
Ogg Theora support No
WebM support No
Audio
2 bonus points
20
audio element Yes
PCM audio support No
MP3 support Yes
AAC support Yes
Ogg Vorbis support No
WebM support No
Elements
22/25
Embedding custom non-visible data Yes
New or modified elements

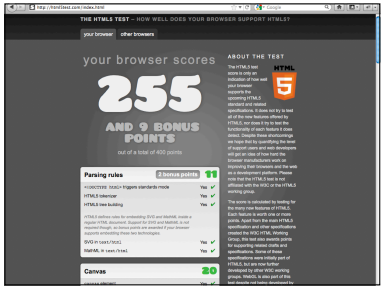

Files 0/20
FileReader API No
FileSystem API No
Storage 15/20
Session Storage Yes
Local Storage Yes
IndexedDB No
Web SQL Database Yes
Workers 15
Web Workers Yes
Shared Workers Yes
Local multimedia 0/20
Access the webcam No
Notifications 0/10
Web Notifications No
Other


Forms 55/98
Field types
input type=search Yes
input type=tel Yes
input type=url Yes
input type=email Yes
input type=datetime Partial
input type=date Partial
input type=month Partial
input type=week Partial
input type=time Partial
input type=datetime-local Partial
input type=number Yes
input type=range Yes
input type=color Partial
input type=checkbox Yes
input type=image Partial
textarea Yes
select Yes
fieldset Partial
datalist Partial
keygen Yes
output Yes
progress No
meter No
Fields
Field validation Yes
SPONSORS
Advertise Here
DEVELOPMENT
sights.



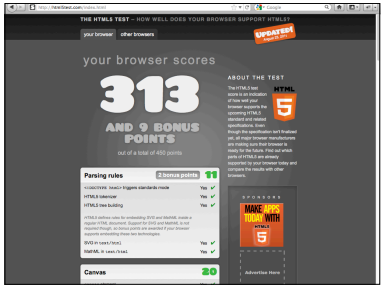


Here's 4.



And 6...

Again, that's 3.6.13.



Let's go!

Doctype.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
        "http://www.w3.org/TR/html4/loose.dtd">
```

Trans-what-now?

Quite a mouth full...

People designed expecting poor rendering.

Browsers standards support improved.

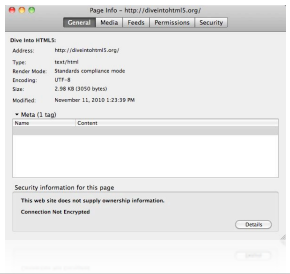
And almost standards mode.

And they broke the web.

Seriously.

Quirks mode vs. standards mode.

Let developers opt in.



HTML5 simplifies life.

```
<!DOCTYPE HTML>
```

Blank line can kick you into quirks mode...

It's almost too easy!

New semantic elements.

And you can type it.

HTML5 adds new elements.

Oh, must be the first line.

- section
- nav
- article
- aside
- hgroup
- header
- footer
- time
- mark

Defines things we've been doing for years.

Again, nod to what we're actually doing.

With divs and ids.

More meaningful than divs!

It works, but lacks meaning.

So what do these elements mean?

Common markup.

`<section>`

Thematic grouping of content.

<nav>

Might have heading or an outline.

Section with links.

Chapters, tabs.

Major navigation blocks.

Intro, part 1, part 2...part N, conclusion.

Common in footers.

Screen readers,
keyboard only users.

Nothing tells you that's
navigation though.

<article>

Common yes...

Reusable or
distributable.

Accessibility.

Post, blog entry, comment.

Think syndication.

<hgroup>

<aside>

Group of set of headings (h1-h6).

Tangential content.

<header>

Sidebars, pull quotes.

Introduction.

Could contain `hgroup` or headings.

```
<header>
...
</header>
```

Does't create a new section.

```
<footer>
```

Not a new scope for headers/footers.

Usually at the bottom of a section.

```
<div id="header">
...
</div>
```

Often contains copyright, contact info, help, privacy, etc.

Whatever lives in the `div id="footer"` ;)

Meetings, birthdays, anniversaries ;)

Does't create a new section.

```
<time datetime="2011-02-22" pubdate>February 22, 2011</time>
```

`<time>`

3 parts.

Encode time/date for machine use.

1. Machine readable.

```
<time datetime="2011-02-22" pubdate>February 22, 2011</time>
```

YYYY-MM-DD

Quite flexible.

http://www.whatwg.org/specs/web-apps/current-work/multipage/common-microsyntaxes.html#valid-global-date-and-time-string

Want time?

Add T, time in 24 hour, timezone offset.

```
datetime="2011-02-22T11:21:37-07:00"
```

2. Human readable.


```
<time datetime="2011-02-22" pubdate>February 22, 2011</time>
```

Text doesn't have to match the datetime attribute.

pubdate flag.

It's human readable!

```
<time datetime="2011-02-22" pubdate>February 22, 2011</time>
```

Next Sunday, tomorrow, in three days...

Boolean.

Could even be empty.

Says timestamp is publication date.

For article or
the document...

<mark>

Think highlight.

Call attention to something.

And if your browser
doesn't support it?

Unknown elements
rendered inline.

However, many of these
elements are block.

In older browsers...

Style them as block.

Also affects the DOM.

HTML5 Reset.

http://html5doctor.com/html-5-reset-stylesheet/

The workaround?

Oh, before 9, IE won't style unknown elements.

Create the element in JavaScript.

Despite your CSS.

IE will allow you to style it.

Don't want to do that yourself?

No worries.

HTML5 enabling script.

http://remysharp.com/2009/01/07/html5-enabling-script/

What's all the fuss about?

Divs work, right?

Document outline.

http://gsnedders.html5.org/outliner/

Before, headings were our only hope.

Sectioning content (article, aside, nav, section)...

Create new nodes.

Each has its
own hierarchy.

Aids composability.

Lab time!

Semantic Elements

- Take the sample web page and "convert" it to use HTML5 semantic elements
- View the page in various browsers
- ${extract}/html5_workshop/labs/semantic_elements.html

New Input Types.

We've spent a lot of
time developing apps.

With a really
limited palette.

Text box, text area, drop down, radio button...

Pretty limited.

Libraries help!

But why doesn't the browser do more?

Now it can!

HTML5 adds 13 new types.

And if your browser doesn't support it?

No worries.

Unknown types
treated as text.

Even works in IE 6!

So what's been added?

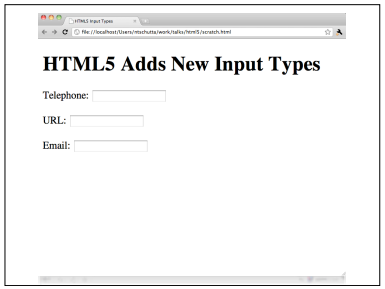
- search
- spinner
- slider
- color picker
- telephone number
- url
- email
- date, month, week, timestamp
- datetime

What do they do?

The spec doesn't say.

In many cases, they
look just a text box.

For example...



So what's the point?

```
<!DOCTYPE HTML>
<html>
<head>
    <title>HTML5 Input Types</title>
</head>
<body>
    <h1>HTML5 Adds New Input Types</h1>

    Telephone: <input type="tel"> <br/> <br/>
    URL: <input type="url"> <br/> <br/>
    Email: <input type="email"> <br/> <br/>

</body>
</html>
```

What about the iPhone?

Impressed?

No keyboard.

Yeah...

"Need a keyboard."

Really?

That's useful!

Can't reconfigure a physical keyboard.

Frustrating when sites don't.

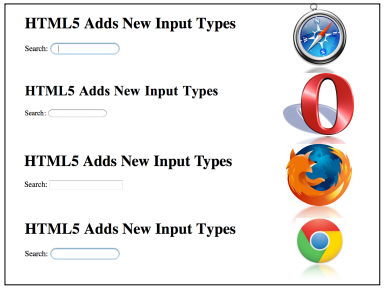
But when it's software...

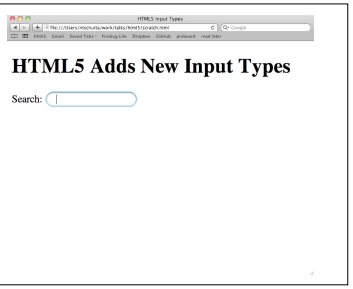
And it costs you nothing.

Search.

Speaking of inputs that don't look much different...

Oh, and an X...






```
<!DOCTYPE HTML>
<html>
<head>
    <title>HTML5 Input Types</title>
</head>
<body>
    <h1>HTML5 Adds New Input Types</h1>

    Search: <input type="search" autofocus> <br/><br/>

</body>
</html>
```

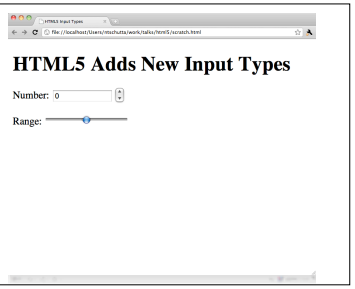
Numbers.

Rounded corners!

Like email addresses & URLs, they're special.

How about spinners and ranges?

Default step value is 1.



Ranges.

```
<!DOCTYPE HTML>
<html>
<head>
    <title>HTML5 Input Types</title>
</head>
<body>
    <h1>HTML5 Adds New Input Types</h1>

    Number: <input type="number"
                   step="2"
                   value="0"
                   min="0"
                   max="10"> <br/> <br/>

    Range: <input type="range"> <br/> <br/>

</body>
</html>
```

Slider control.

Attributes are optional.

Shades of thick clients!

```
<!DOCTYPE HTML>
<html>
<head>
    <title>HTML5 Input Types</title>
</head>
<body>
    <h1>HTML5 Adds New Input Types</h1>

    Number: <input type="number"
                   step="2"
                   value="0"
                   min="0"
                   max="10"> <br/> <br/>

    Range: <input type="range"> <br/> <br/>

</body>
</html>
```

valueAsNumber

Also includes some handy JavaScript.

Returns value as a number!

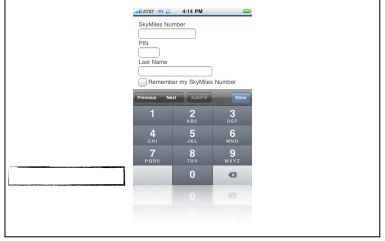
stepUp(n)
stepDown(n)

The value attribute is a string...

Increase/decrease the value by n.

Useful?

Back to the iPhone...



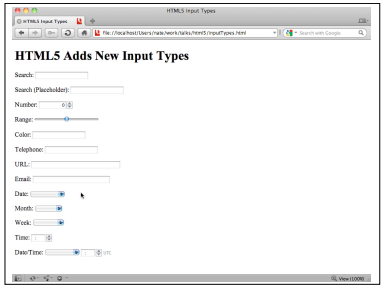
Date pickers.

Why don't we have a native date picker?

Now we do!

But it's only in Opera.

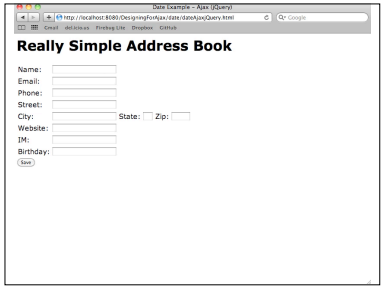
And you may not like it.



```
<!DOCTYPE HTML>
<html>
<head>
    <title>HTML5 Input Types</title>
</head>
<body>
    <h1>HTML5 Adds New Input Types</h1>

    Date: <input type="date"> <br/> <br/>
    Month: <input type="month"> <br/> <br/>
    Week: <input type="week"> <br/> <br/>
    Time: <input type="time"> <br/> <br/>
    Date/Time: <input type="datetime"> <br/> <br/>

</body>
</html>
```



CSS could be improved.

But which would your users' prefer?

Fallback to a library.

Speaking of pickers.

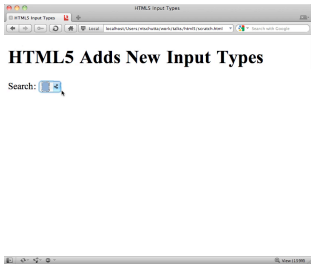
Color!

Pick a color, get a hex value!

Cool!

Except on Linux.

Only works in Opera.



Bummer.

```
<!DOCTYPE HTML>
<html>
<head>
    <title>HTML5 Input Types</title>
</head>
<body>
    <h1>HTML5 Adds New Input Types</h1>

    Color Picker: <input type="color">

</body>
</html>
```

Uses native picker.

Autofocus.

```
<!DOCTYPE HTML>
<html>
<head>
    <title>HTML5 Input Types</title>
</head>
<body>
<h1>HTML5 Adds New Input Types</h1>
Search: <input type="search" autofocus> <br/><br/>
Search (Placeholder): <input type="search" placeholder="search"> <br/> <br/>
Number: <input type="number" step="2" value="0"> <br/> <br/>
Range: <input type="range"> <br/> <br/>
Color: <input type="color"> <br/> <br/>
Telephone: <input type="tel"> <br/> <br/>
URL: <input type="url"> <br/> <br/>
Email: <input type="email"> <br/> <br/>
Date: <input type="date"> <br/> <br/>
Month: <input type="month"> <br/> <br/>
Week: <input type="week"> <br/> <br/>
Time: <input type="time"> <br/> <br/>
Date/Time: <input type="datetime"> <br/> <br/>
</body>
</html>
```

Today we use JavaScript.

Edge cases.

Consistency.

Can be disabled.

Validation!

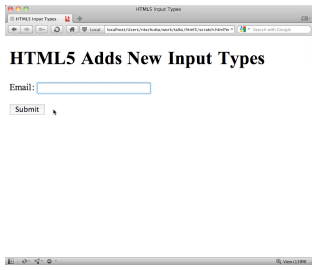
Email address.

Yes, you can do this
in JavaScript today.

Maybe you already do.

HTML5 to the rescue!

It's hard!



What if JS is disabled?

Validation is on by default.

And you're still validating on the server right?

Also works for url and numbers.

Even respects min/max.

Safari and Chrome,
no error messages.

Don’t want to validate?

Just doesn’t submit ;)

Use novalidate attribute.

Very user friendly.

Support is...soft.

Required fields.

Add the required attribute!

```
<!DOCTYPE HTML>
<html>
<head>
    <title>HTML5 Input Types</title>
</head>
<body>
    <h1>HTML5 Adds New Input Types</h1>
    <form>
        This field required: <input name="foo" required>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

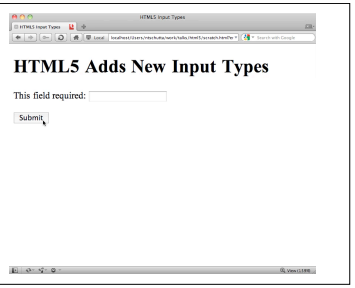
Appearance varies by browser.

Lab time!

For example...

Forms

- Take the sample web page and "convert" it to use HTML5 form elements
- Make a field required
- Put focus in the first field
- Add placeholder text
- Submit the form in various browsers
- ${extract}/html5_workshop/labs/forms.html



Fun With Numbers

- Convert the text field to a number
- Write a function that adds 1 to the number
- Write a function that subtracts 1 from the number
- Write a function to display the type of the input field using valueAsNumber
- ${extract}/html5_workshop/labs/numbers.html

Canvas.

Graphics!

Graphs, shapes,
animations, etc.

Controlled via scripting.

Pretty simple.

```
<canvas id="canvas" width="800" height="800"></canvas>
```

That's it?

Only two attributes:
width and *height*.

Optional, default is
300 pixels by 150 pixels.

```
<canvas id="fallback" width="800" height="800">
   Put fallback content here...perhaps an image.
</canvas>
```

CSS sizing as well.

Content between tag.

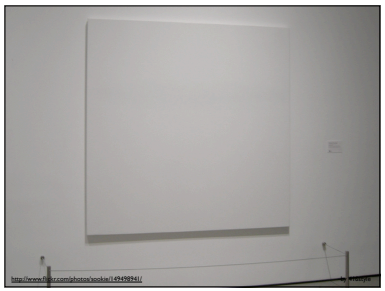
Can be styled like an image
- boarder, margin, etc.

Ignored by browsers
supporting canvas...

Specifying fallback content.

Canvas tag is ignored by
browsers lacking support.

Canvas starts like any canvas...



Up to you to fill it!

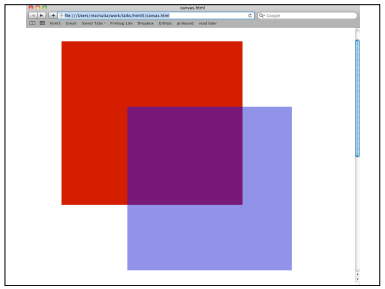
To draw, we need a context.

```
var ctx = canvas.getContext("2d");
```

For now, just 2D...

Likely 3D in the future.

Once we have a context, we can draw!



```
fillRect(x, y, width, height);
strokeRect(x, y, width, height);
clearRect(x, y, width, height);
```

```
<head>
    <script type="application/javascript">
        function draw() {
            var canvas = document.getElementById("canvas");
            var ctx = canvas.getContext("2d");

            ctx.fillStyle = "rgb(200,0,0)";
            ctx.fillRect(100, 100, 550, 500);

            ctx.fillStyle = "rgba(0, 0, 200, 0.5)";
            ctx.fillRect(300, 300, 550, 500);
        }
    </script>
</head>
<body onload="draw()">
<canvas id="canvas" width="800" height="800"></canvas>

<canvas id="fallback" width="800" height="800">
    Put fallback content here...perhaps an image.
</canvas>
```

All take the same arguments...

One primitive - rectangle.

x and y position of left corner of rectangle...

Three methods.

Width and height.

fillRect - filled rectangle.

strokeRect - outline.

clearRect - clears area, makes it transparent.

So that's it? Rectangles?

No!

Paths.

We can draw shapes.

- *beginPath* - creates path
- *closePath* - tries to close the shape
- *stroke* - draws an outlined shape
- *fill* - draws a solid shape
- *moveTo* - moves the "pen", doesn't draw

https://developer.mozilla.org/en/Canvas_tutorial%3aDrawing_shapes

```
function draw() {
    var canvas = document.getElementById("canvas");
    var ctx = canvas.getContext("2d");

    ctx.beginPath();
    ctx.arc(75,75,50,0,Math.PI*2,true); // Outer circle
    ctx.moveTo(110,75);
    ctx.arc(75,75,35,0,Math.PI,false);   // Mouth (clockwise)
    ctx.moveTo(65,65);
    ctx.arc(60,65,5,0,Math.PI*2,true);  // Left eye
    ctx.moveTo(95,65);
    ctx.arc(90,65,5,0,Math.PI*2,true);  // Right eye
    ctx.stroke();

}
```

Arc? Draws circles.

- x
- y
- radius
- startAngle - start point, radians
- endAngle - end point, radians
- anticlockwise - boolean, clockwise or not

lineTo(x, y) - Straight lines

Curves.

quadraticCurveTo,
bezierCurveTo

And of course you can combine these...

You can also use images.

Get an image - from page of from scratch.

`drawImage(image, x, y)`

Useful as backdrops...

Can also scale images.

Add width and height.

You can also crop...

And on and on!

Colors, gradients, line styles, patterns...

Rotating, scaling, transforms, compositing.

Animations!

Whew!

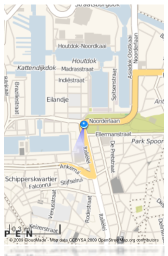
Lab time!

Olympic Rings

- Using the various canvas methods, draw the Olympic Rings
- ${extract}/html5_workshop/labs/canvas_rings.html

Geolocation.

Where in the world are you?



http://www.offmaps.com/

Very helpful on phones!

Technically not a part of HTML5.

Geolocation Working Group.

Wide browser support.

Your browser doesn't support it?

Device specific options.

Privacy issue?

Opt-in.

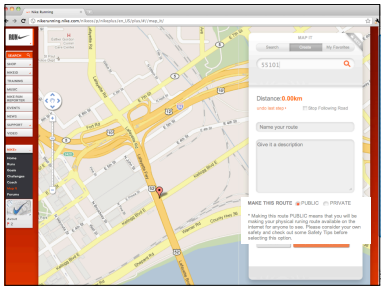
http://www.w3.org/TR/geolocation-API/#security

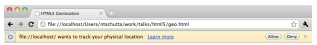
Absolutely.

Browsers tell you
before data is sent.

If you know where
the device is...

http://icanstalku.com/why.php



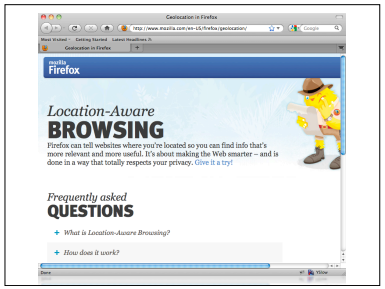


HTML5 Adds Geolocation

Where am I?

HTML5 Adds Geolocation

Where am I?



Tab specific.

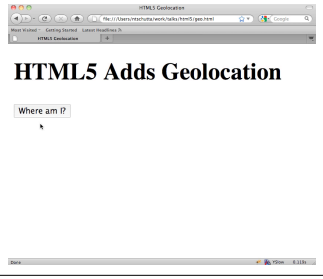
Infobars are smart.

Blocks.

Often give link to further information.

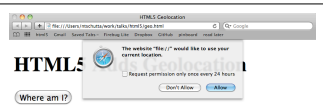
You can go about your business in other tabs.

Aren't modal.



How does this work?

navigator.geolocation



Simple API.

```
function whereAmI() {
  if (Modernizr.geolocation) {
    navigator.geolocation.getCurrentPosition(showLocation);
  } else {
    alert("this browser doesn't support geolocation, sorry!");
  }
}

function showLocation(position) {
  var latitude = position.coords.latitude;
  var longitude = position.coords.longitude;
  $("#location").html("lat: " + latitude + " and long: " + longitude);
}
```

getCurrentPosition()

New property.

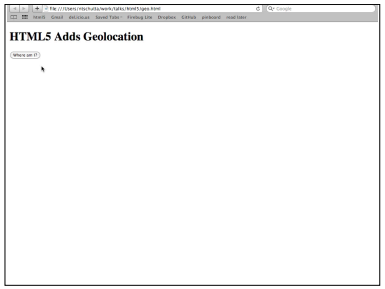
Browser "determines location," creates `Position`.

`Position` contains `Coordinates` & timestamp.

Coordinates

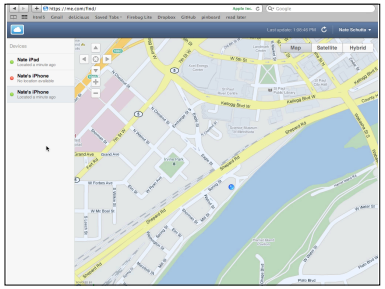
- latitude (double)
- longitude (double)
- altitude (double or `null`)
- accuracy (double)
- altitudeAccuracy (double or `null`)
- heading (double or `null`)
- speed (double or `null`)

You supply a callback function.



How accurate is it?

Can be *very* accurate.



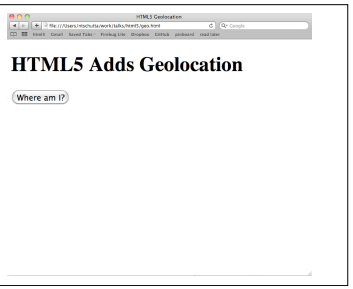
Your function receives a Position object with:

coords & timestamp

Can take some time ;)

getCurrentPosition()

Optional second arg:
error callback function.




```
function whereAmI() {
  if (Modernizr.geolocation) {
    navigator.geolocation.getCurrentPosition(showLocation, handleError);
  } else {
    alert("this browser doesn't support geolocation, sorry!");
  }
}

function showLocation(position) {
  var latitude = position.coords.latitude;
  var longitude = position.coords.longitude;
  $("#location").html("lat: " + latitude + " and long: " + longitude);
}

function handleError(error) {
  if(error.code == 1) {
    var message = "You want to keep your location private, that's OK!"
  }
  $("#location").html("Ooops! " + message);
}
```

Callback gets a
`PositionError` object.

Two attributes.

code & message

Code values...

- PERMISSION_DENIED (1)
- POSITION_UNAVAILABLE (2)
- TIMEOUT (3)
- UNKNOWN_ERROR (0)

getCurrentPosition()

Optional third argument.

PositionOptions

- enableHighAccuracy (boolean)
- timeout (long)
- maximumAge (long)

All attributes are optional.

Higher accuracy
may be slower.

Some devices have
separate permissions.

Timeout is based on network time, not user.

Age allows you to cache positions.

IE? Out of luck < 9.

Are other options.

Gears, device specific.

geo.js

http://code.google.com/p/geo-location-javascript/

Layer over various approaches.

Lab time!

Geolocation

- Using the geolocation API, create a function that displays your current latitude and longitude
- Handle the situation where a user opts out
- Handle the situation where location cannot be determined
- ${extract}/html5_workshop/labs/geolocation.html

Some browsers call it DOM Storage.

Local storage.

Simple way to store *key/value* pairs.

Technically web storage.

Like cookies...

Split into separate spec.

But bigger, stays local.

Very wide support.

Key is a *string*.

Data can be any JavaScript datatype...

But it's stored as a *string*.

Don't forget to parse...

Simple interface.

getItem(key)
setItem(key, value)

setItem() *silently overwrite* old values.

getItem() with unused key *returns null*...

Can remove items: *removeItem(key)*

Can treat *localStorage* as an *associative array*.

Called on a nonexistent key does nothing.

In other words, *bracket notation*.

localStorage.length gets number of stored values.

localStorage.getItem("key")
localStorage["key"]

key(index) retrieves key at that index.

Index out of bounds
returns null.

StorageEvent

- key
- oldValue
- newValue
- url

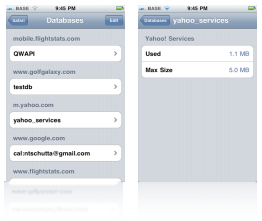
Also a *storage* event.

Can't cancel it.

Calls to setItem,
removeItem or clear...

Default max:
5 megabytes.

Provided *something changes.*



Exceed that...

QUOTA_EXCEEDED_ERR

Can you ask for more?

No.

Some browsers allow *the user* to control quota.

Lab time!

Local Storage

- Add a function that stores the current values locally
- Add a function that retrieves the values in local storage
- Add a function that displays the locally stored values
- Add a function that clears local storage
- ${extract}/html5_workshop/labs/localstorage_form.html

There's more...

Offline.

Google Apps, GMail rock.

Some flights have wifi.

Not cheap.

Application cache.

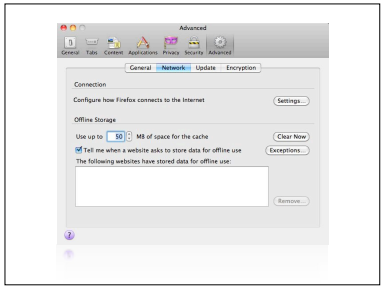
Web server tells client what it needs.

Application works when disconnected.

When it connects, changes are uploaded.

http://demos.hacks.mozilla.org/openweb/todo/

Multiple threads.



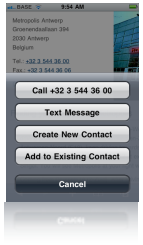
Microdata.

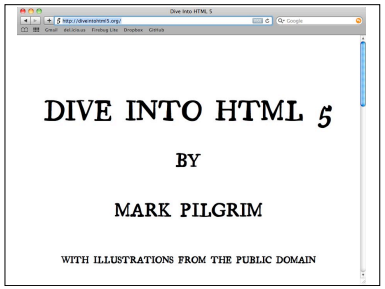
Web workers.

Add semantics
to your pages.

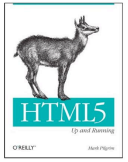
Background JavaScript.

Licensing info, vCard.





For search and browsers.



Books coming.

http://books.alistapart.com/product/html5-for-web-designers

http://diveintohtml5.org/

Plus *many* more.

Questions?!?

Thanks!

Please complete your surveys.