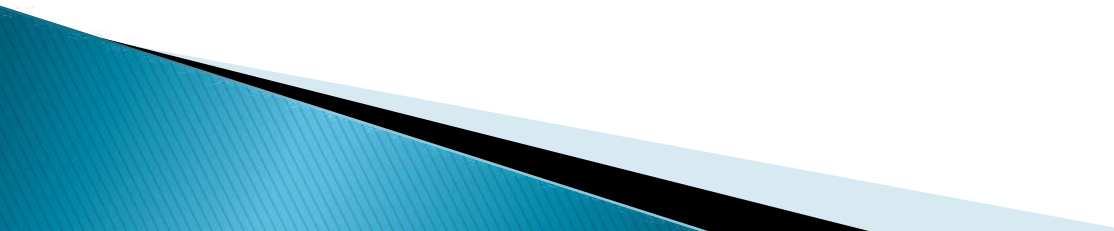


HTTP Client Hints

This specification defines a set of HTTP request header fields, known as Client Hints, that are intended to be used as input to proactive content negotiation; just as the accept header allows clients to indicate what formats they prefer, Client Hints allow clients to indicate a list of device and agent specific preferences.



HTTP Client Hints

▶ Available hints

Current list includes DPR (device pixel ratio), Width (resource width), Viewport-Width (layout viewport width), and Dwnlink(maximum downlink speed) request headers, and Content-DPR response header that is used to confirm the DPR of selected image resources.

HTTP Client Hints

- ▶ **Delivering DPR-aware images**

DPR hint automates device-pixel-ratio-based selection and enables delivery of optimal image variant without any changes in markup. For example, given the following HTML markup:

```

```

HTTP Client Hints

GET /img.jpg HTTP/1.1

User-Agent: Awesome Browser Accept: image/webp, image/jpg

DPR: 2.0

CH-RW: 160

HTTP/1.1 200 OK

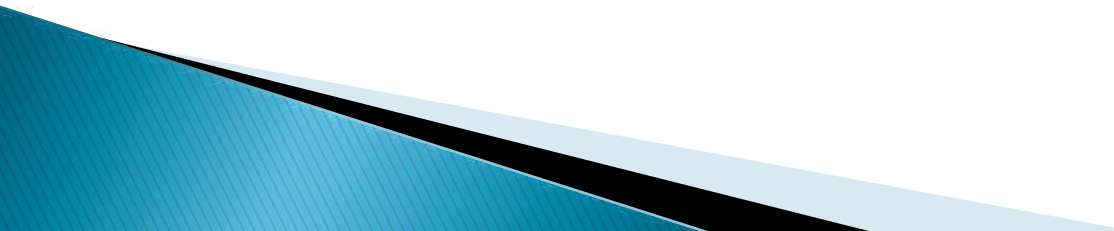
Server: Awesome Server

Content-Type: image/jpg

Content-Length: 124523

Vary: CHoDPR, CH-RW

DPR: 2.0

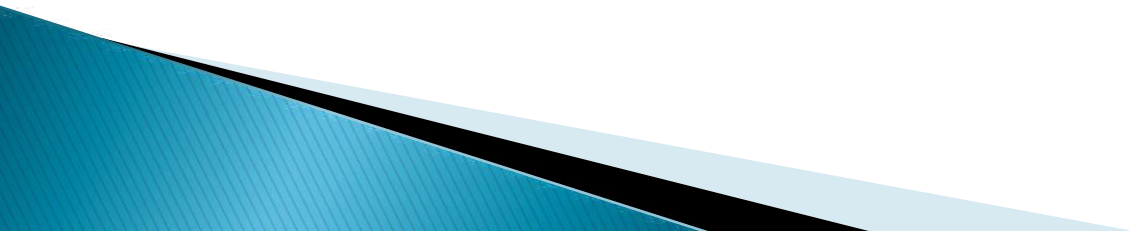


Do the right selection of quality image

- ❑ Should we make several copies of the image and conditionally request the right one from the client?

Responsive Images

Responsive images is a method for providing the browser with multiple image sources depending on display density, size of the image element in the page, or any number of other factors to make possible flexible approach to images.

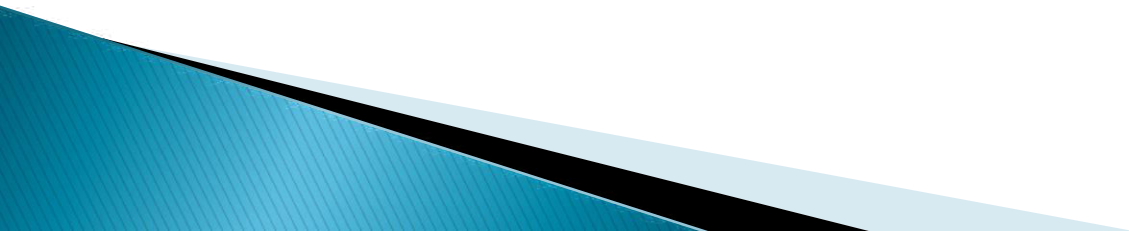


Responsive Images

Graphics are often a problem in responsive layouts designed to adapt automatically to different device sizes from desktop monitors to hand-held smartphones.

Traditionally, graphics have fixed dimensions that break responsive layouts on screens that are smaller than the graphic.

Length is not a problem, because pages can always scroll vertically. However, you never want the page to have to scroll horizontally on mobile devices with fluid layouts.



Problems in Responsive Images

▶ **What Are The Problems?**

One major factor in the need for responsive images is overall website size - even today, a huge percentage of mobile-version sites are as large (or even larger) than their desktop versions, thereby affecting performance negatively

Problems in Responsive Images

► Problem 1 - Semantics

Making responsive images work cleanly and reliably across multiple platforms sometimes involves using techniques that aren't 'semantic'. Why? Because when the source of an image points to a real image, with an alt text to describe it, even if the source is the smallest image possible, it means you're downloading unnecessary data.

Problems in Responsive Images

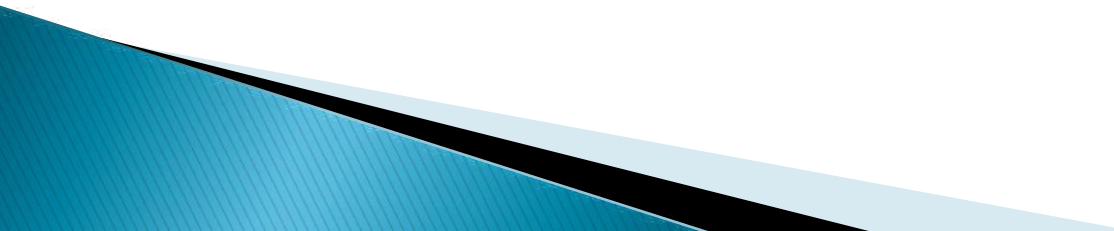
▶ Problem 2 - Art Direction

A picture tells a thousand words ... most of the time. Many responsive image techniques enable you to provide several resolution versions of an image, which can then be used accordingly for suit a given platform. However, this can sometimes have a negative effect where the message of the image is diluted or lost altogether. What's the minimum size for an image? The answer, of course, depends on context.

Problems in Responsive Images

► Problem 3 - Validity

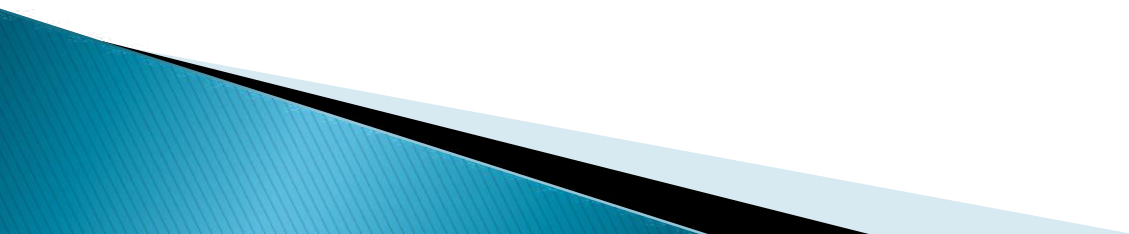
If you are customizing markup, checking markup validity using a validation service such as W3C ensures that a construct follows the correct syntax of whatever language it works with (for example, HTML). While most of the time browsers are able to think ahead to present Web pages that aren't strictly 'valid' without any problems, these results can sometimes vary between one browser and another. In rare cases, it can cause confused layouts or even crashes. 'Validating' every page is a bulletproof way of identifying these problems before they occur.



Problems in Responsive Images

Problem 4 - Bandwidth Capabilities

Much as we'd all like to say otherwise, a world of super-speed connections and lighting-fast load times for everyone is still some way off. That's why in many developers' opinions, responsive images ought to consider the bandwidth available on any given device, and adjust the image size accordingly to cut out excess downloads.



Problems in Responsive Images

► Problem 5 - Server Side Components

The bulk of Adaptive Images' work is done through .htaccess and PHP 5.x, which offers a great solution without complete dependency on JavaScript. The problem with .htaccess is that it assumes your website is running on an Apache server. If you're using something else, such as Nginx, you can port the .htaccess over to Nginx syntax - which is similar to Apache's but this still involves some work. Additionally, adding PHP files to your site's root directory may not be possible if you're running on other technologies, such as Ruby or Python. Clearly this isn't ideal for everyone, so be sure to check first before you dive in.

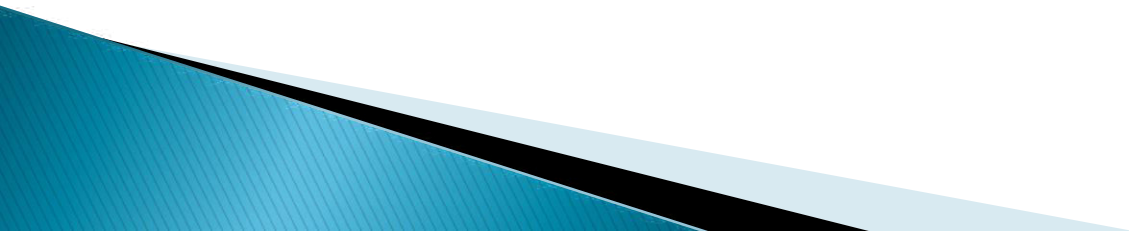
Problems in Responsive Images

▶ **Problem 6 - JavaScript**

As you've probably noticed, many of the responsive image techniques above require some JavaScript. If you'd prefer to keep things as simple as possible and want to avoid using JavaScript you can use the third-party Sencha.io solution.

Responsive images

Responsive images is a method for providing the browser with multiple image sources depending on display density, size of the image element in the page, or any number of other factors to make possible flexible approach to images.



Responsive Images using CSS3

Using The width Property:

If the width property is set to 100%, the image will be responsive and scale up and down:

```
img {  
    width: 100%;  
    height: auto;  
}
```

Responsive Images using CSS3

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
img {
    width: 100%;
    height: auto;
}
</style>
</head>
<body>

<p>Resize the browser window to see how the image will scale.</p>
</body>
</html>
```

Responsive Images using CSS3



Resize the browser window to see how the image will scale.

Responsive Images using CSS3

► Using The max-width Property

If the max-width property is set to 100%, the image will scale down if it has to, but never scale up to be larger than its original size:

```
img {  
  
    max-width: 100%;  
    height: auto;  
  
}
```

Responsive Images using CSS3

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<style>
img {
    max-width: 100%;
    height: auto;
}
</style>
</head>
<body>

<p>Resize the browser window to see how the image wil scale when the width is less
than 460px.</p>
</body>
</html>
```

Responsive Images using CSS3



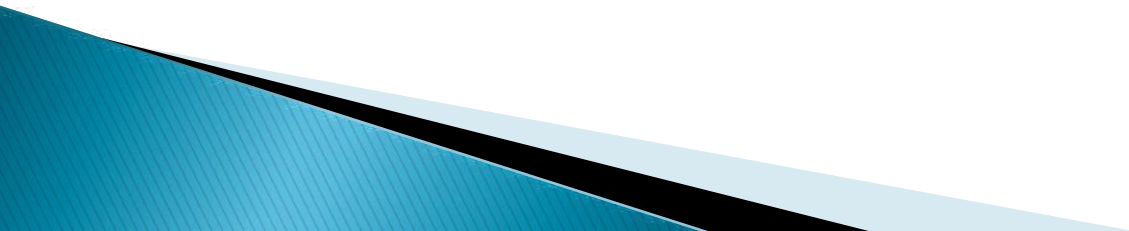
Resize the browser window to see how the image wil scale when the width is less than 460px.

Use of high quality and different sized copies of image

In order to do responsive images , we need to request different images based on the device pixel ratio, but also possibly based on the layout width.

These images may be the same image transcoded to different sizes.

They may be actually different images depending on the layout size.



Use of high quality and different sized copies of image

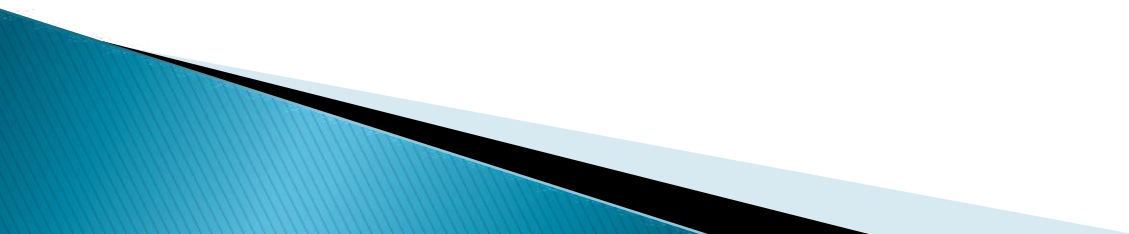


Use of high quality and different sized copies of image

We shall need to provide multiple copies of this image so that low pixel count devices can save bandwidth or high pixel count devices can get a crisp image.

It is possible to manage on server side in some cases.

In fact, some content systems let you simply specify what size you want a given source image app.



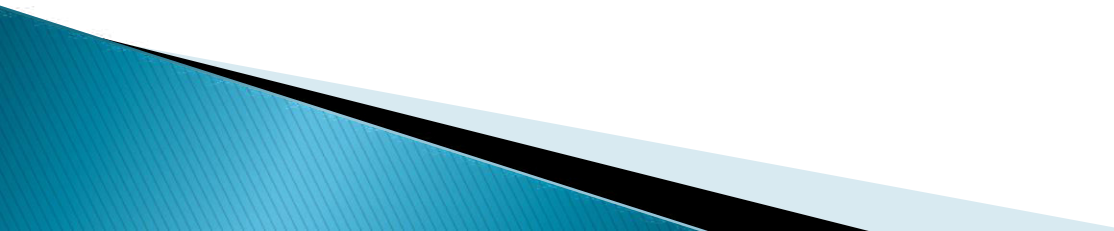
Use of high quality and different sized copies of image

If you want the images to actually represent different art, different crops for example you'll have to do that manually.

For each of these images though you should remember to tune the quality parameters to minimize file size while maximizing quality.

We'll need at least 1x and 2x versions of images as 1x or 2x for device pixel ratio purpose.

For responsive images, We may request different images based on available layout width.



Request Right Image

- ▶ Client should request for the “right” image.
- ▶ We need to get the client to ask for the right image to be downloaded.

Request Right Image

- ▶ It needs to know what device pixel ratio the device is using and potentially the layout size too, if it's responsive to the layout width and then request only the appropriate image.

Request Right Image

- ▶ This last part is the hard part, because the web browser tries to download image as early as possible which of course is a good thing for getting content on the screen quickly, but bad if the client asks for more than one version particularly on a mobile device with limited bandwidth.

Request Right Image

So we should try to find the solution for the problem of right image at the right time.

Ideal Image Solution

- ▶ What we really need is the ability for our client to selectively choose what image to use.

Ideal Image Solution

- Ideal solution would be semantic and validate properly. It would only load one copy of the image and work and be accessible across all browsers.

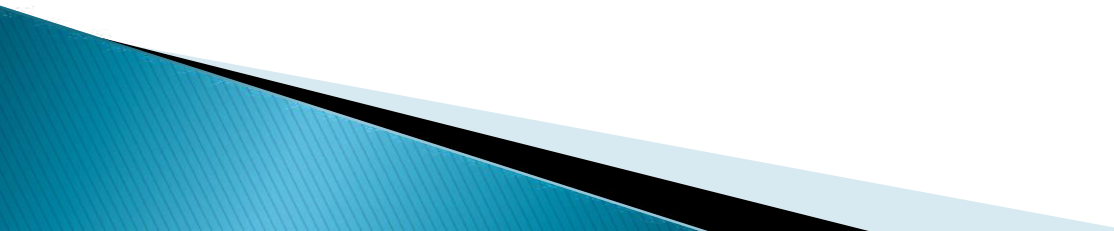
Ideal Image Solution

- ▶ Unfortunately that solution does not exist.

Background Image

The CSS background properties are used to define the background effects for elements.

We shall discuss following properties of background images:

- ▶ Background Image - Repeat Horizontally or Vertically
 - ▶ Background Image - Set position and no-repeat
 - ▶ Background Image -background-position:
- 

Background Image

- ▶ Background Image:

The background-image property specifies an image to use as the background of an element. By default, the image is repeated so it covers the entire element.

Background Image

Example:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {  
    background-image: url ("paper.gif");  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>Hello World!</h1>
```

```
<p>This page has an image as the background!</p>
```

```
</body>
```

```
</html>
```

Background Image

Result:



Hello World!

This page has an image as the background!

↓ [Show all downloads...](#)



Background Image

- ▶ Background Image – Repeat Horizontally or Vertically

By default, the background-image property repeats an image both horizontally and vertically. Some images should be repeated only horizontally or vertically, or they will look strange, like this:

Background Image

- ▶ **Background Image - Repeat Horizontally or Vertically:**

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("gradient_bg.png");
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>Strange background image...</p>

</body>
</html>
```

Background Image

Result:



Hello World!

Strange background image...

Background Image

- ▶ **Background Image - Set position and no-repeat:**

Showing the background image only once is also specified by the background-repeat property:

Background Image

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
    background-image: url("img_tree.png");
    background-repeat: no-repeat;
}
</style>
</head>
<body>

<h1>Hello World!</h1>
<p>Background image example.</p>
<p>The background image is only showing once, but it is disturbing the reader!</p>
</body>
</html>
```

Background Image

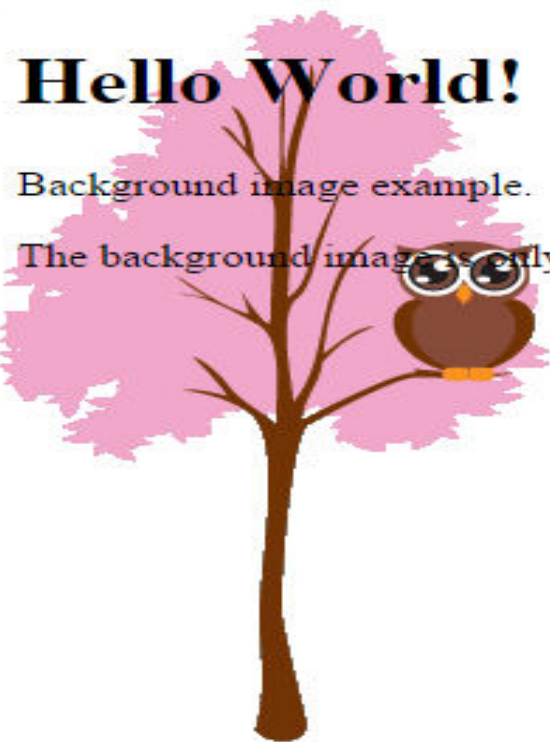
Result:



Hello World!

Background image example.

The background image is only showing once, but it is disturbing the reader!



Background Image

background-position:

In the example above, the background image is shown in the same place as the text. We want to change the position of the image, so that it does not disturb the text too much.

The position of the image is specified by the background-position property:

Example

```
body {  
    background-image: url("img_tree.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

Background Image

Result:

Hello World!

Background no-repeat, set position example.

Now the background image is only shown once, and positioned away from the text.

In this example we have also added a margin on the right side, so the background image will never disturb the text.

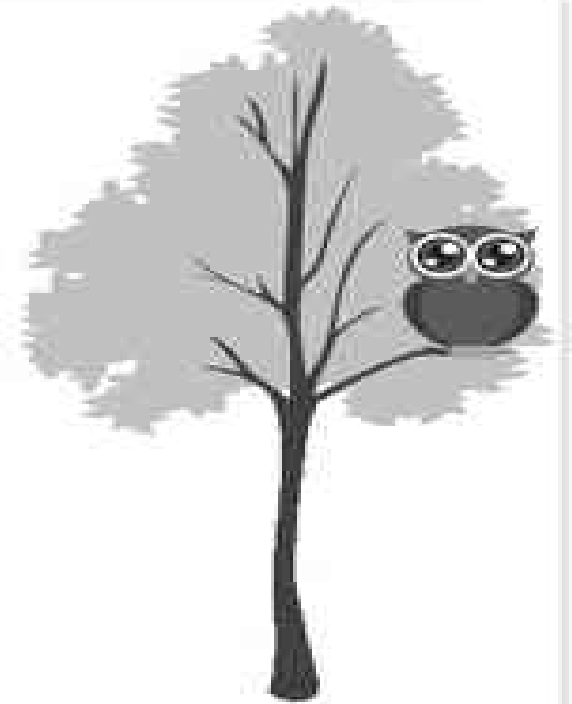


Image-set

Why we use image-set instead of media queries?

- I. The problem with media queries for higher resolution displays is they don't give the browser any discretion. They say explicitly that if the pixel density equals 1 or 2 or whatever, that the browser must use a particular image source.
 - II. The various resolutions of the image are all in the same place in css. When we use media queries to specify different image resolutions, we can end up with the image sources separated by dozens of lines of css which makes tracking down the different image sources more challenging.
- 

Image-set

- ▶ The emerging trend on HD screens for computer devices has changed the way we build websites, including how we add images. In the past, we can simply use background-image and set the image URL.
- ▶ Today, this is no longer relevant, since we also need to provide an image that is optimized for higher resolution displays or the image will look pixelated or blurred. So, technically we need two sets of image, one in regular size and another twice the size.

Image-set



Image-set

▶ CSS Image Set Function:

By using this function, we can insert multiple images which will be set for normal and high-res display.

Apart from that, this function is also trying to deliver the most appropriate image resolution based on the connection speed. So, regardless of the screen resolution, if the user accesses the image through a slow Internet connection.

CSS Image Set is experimental. It is only supported in Safari 6 and Google Chrome 21 and is prefixed — `-webkit-image-set()`. In action, the smaller-sized image will be delivered.

Image-set

- ▶ This function is declared within the background-image property, while the background URL is added within the function followed by the resolution parameter (1x for normal display and 2x is for high-res display), like so.

Image-set

```
.selector {  
    background-image: -webkit-image-set(url('../img/image-  
1x.jpg') 1x, url('../img/image-2x.jpg') 2x);  
}
```

Image-set

```
body {  
    background-image: url('../img/bg.png');  
}  
.demo-wrapper {  
    width: 600px;  
    margin: 0 auto;  
}  
.demo-wrapper .theimage {  
    margin-top: 100px;  
    border: solid 8px #f3f3f3;  
    width: 100%;  
    height: 360px;  
    background-size: 100% auto;  
    background-repeat: no-repeat;  
    background-position: center center;  
    background-image: url('../img/image-1x.jpg');  
    background-image: -webkit-image-set(url('../img/image-1x.jpg')  
1x, url('../img/image-2x.jpg') 2x);  
}
```

Image-set



Image-set

► Finally:

`image-set()` function seems to be a promising solution for serving images in high-res. But, with the current limited browser support, `image-set()` is not ready to be implemented in a live site. In addition, the implementation could be changed along the way in the near future.

Importance of Performance



Why performance

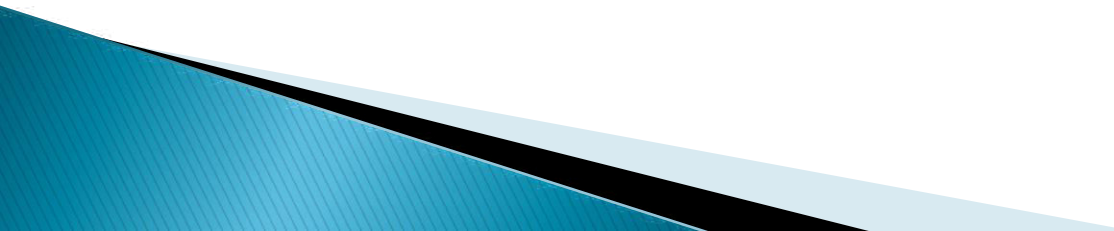
- ▶ A slow site means that you will have less user engagement.

Example

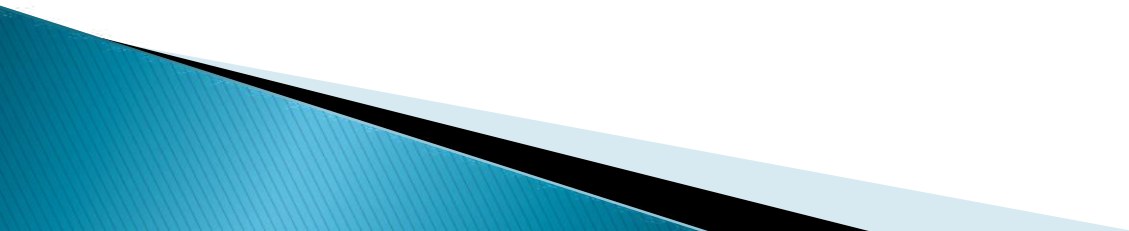
let's say you accidentally, introduce a performance issue, that causes your page to take an extra half second to load. That's just 500 milliseconds.

That's a pretty small difference.

What kind of difference, in user engagement ,do you think this would cause?

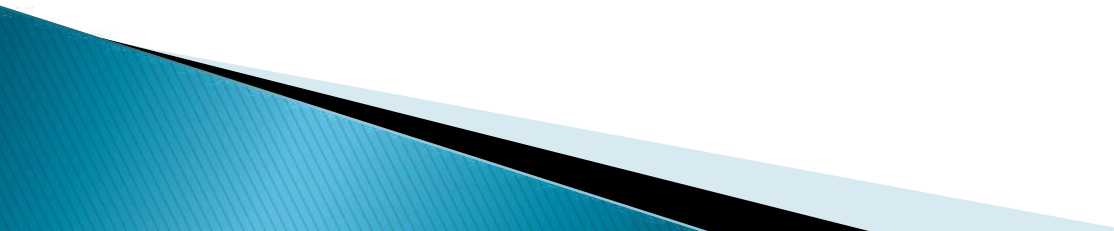


Even such a small change, causes a slightly more than one percent drop in user engagement, and income from your site.



It is a requirement

Many well publicized studies from Google, Microsoft, Amazon, Facebook, and others, all show that UI performance translates directly to profit. A drop in performance directly drives a drop in user engagement. In short, speed is not a feature, it's a requirement. Users won't use your app if it's not fast.

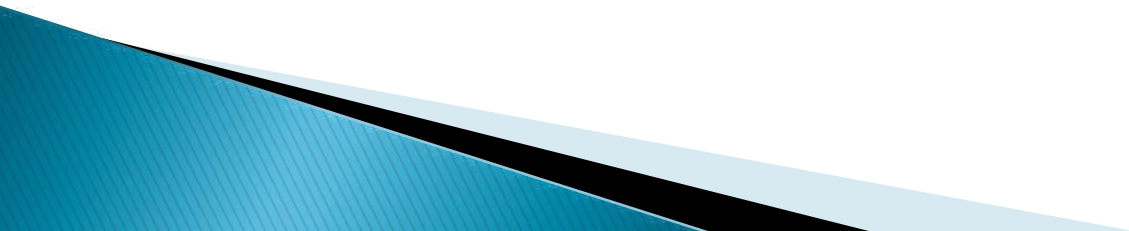


Analyzing Performance

When to consider Performance

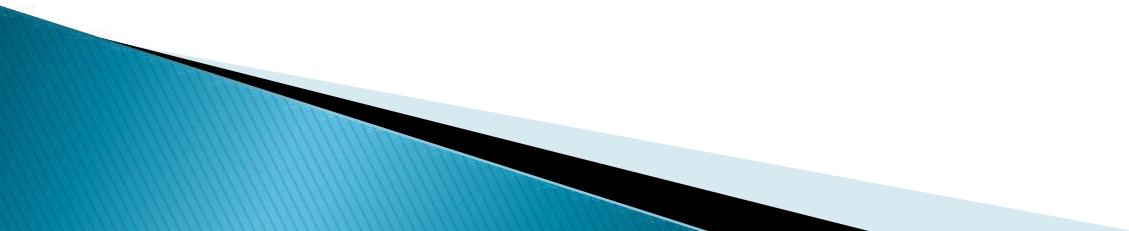
Of course, it's nearly impossible to graph performance into your application after it's already been built, but before we get into good performance patterns, the first step to insuring great performance, is being able to analyze your performance

#perfmatters

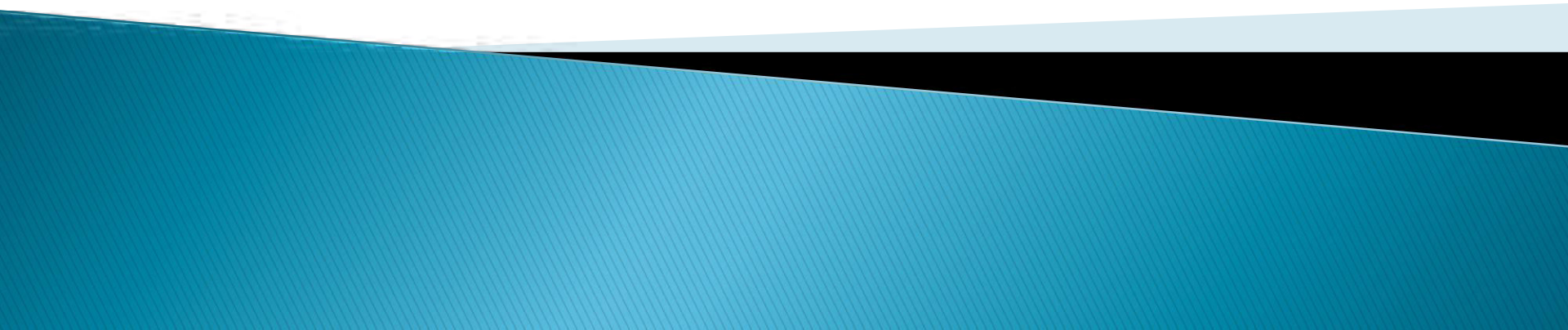


Chrome Development Tools

Chrome Development Tools



Understanding Network



Pillars of Performance

- ▶ Three pillars of performance are
 - ▶ Network
 - ▶ Render
 - ▶ Compute

Network

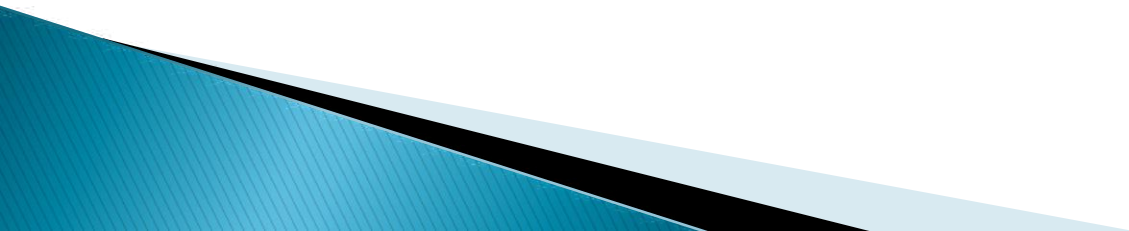
► Examples

1. www.vu.edu.pk
2. <http://playbiolab.com/>
3. <https://www.eecs.mit.edu/>

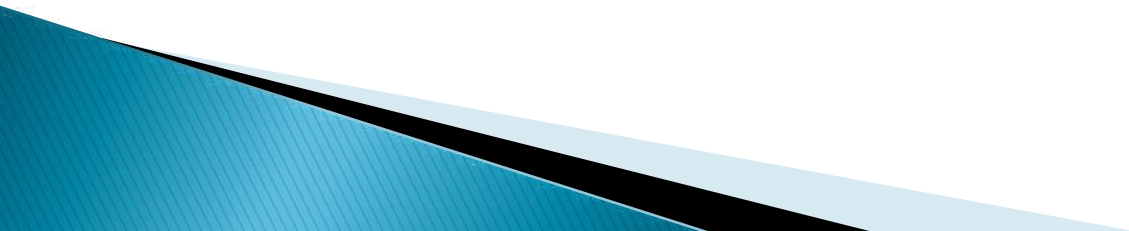
Touch Interaction



One of the most important factors in an immersive user experience on a mobile device is touch interaction. You expect a responsive interaction from a device that you hold in the palm of your hand, and with most mobile screens, your primary pointing device is your finger.



As a mobile web developer, you need to be sure that you're building great touch enabled user devices

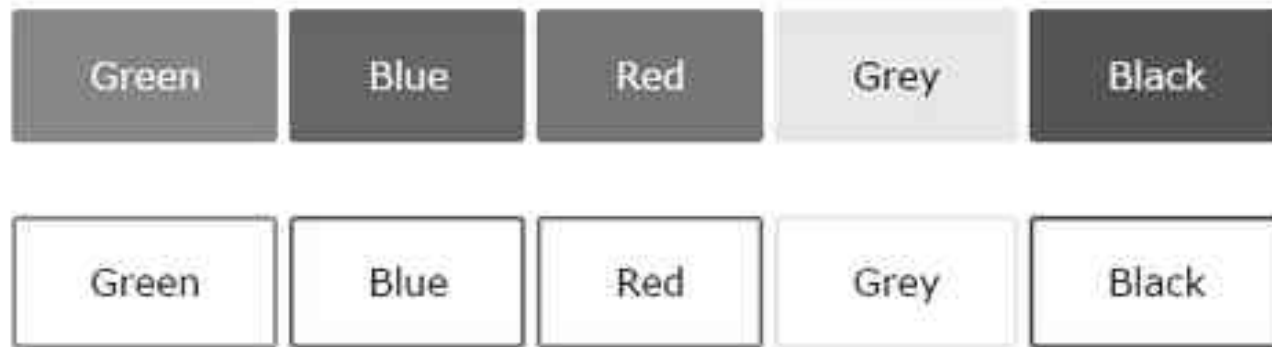


Simple rules for touch UI

Rule 1

Do not relay on Hover Effect

Hoverable Buttons

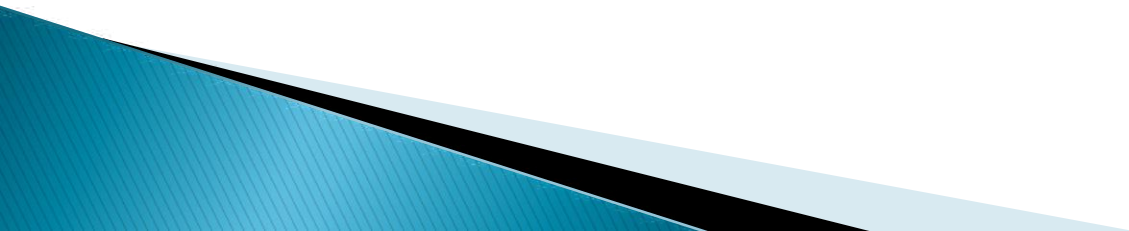


Buttons Source: http://www.w3schools.com/css/css3_buttons.asp

Rule 2

Size of Touch Area should be at-least

10mm X 10 mm



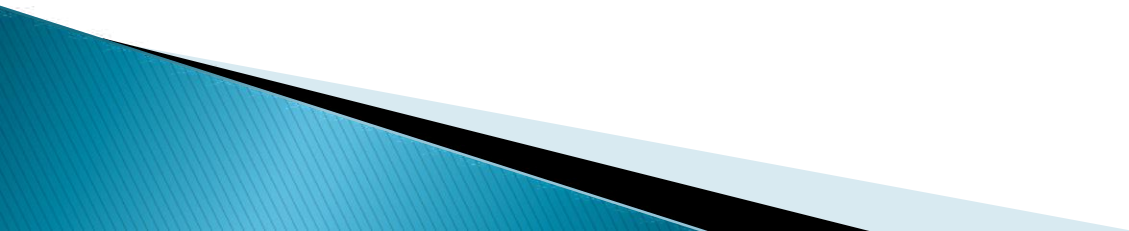
Rule 3

Touch and mouse are not mutually exclusive



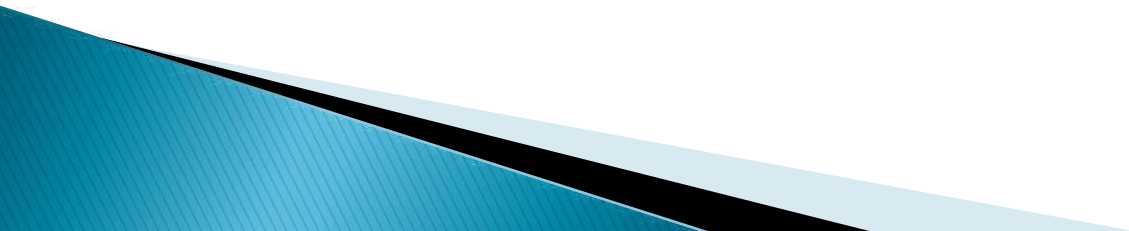
Rule 4

Provide Mouse support as well.



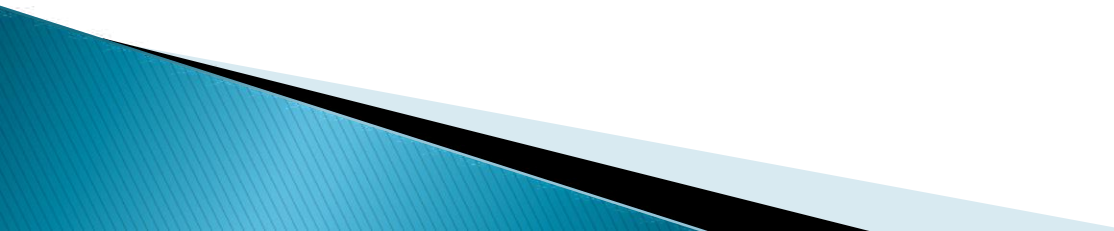
Don't do anything

It is important to understand that you may not even need to do anything in order to support touch.



Touch events emulate mouse clicks already, and the web browser builds in some gestures like scrolling and zooming.

So, you only need to implement something here in cases where the gesture support Or mouse emulation does not give an optimal user experience.



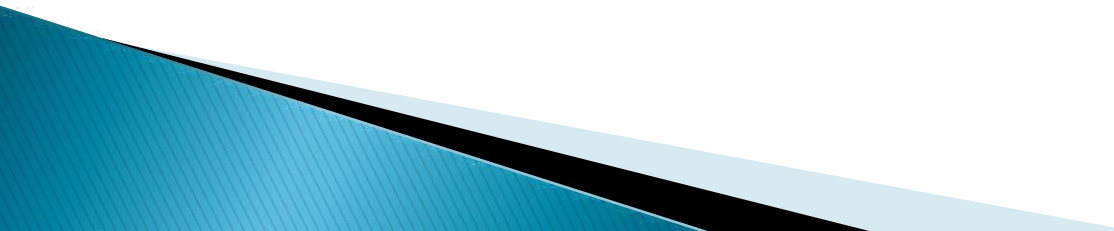
Example

- ▶ http://www.w3schools.com/css/css3_buttons.asp

Click Delay

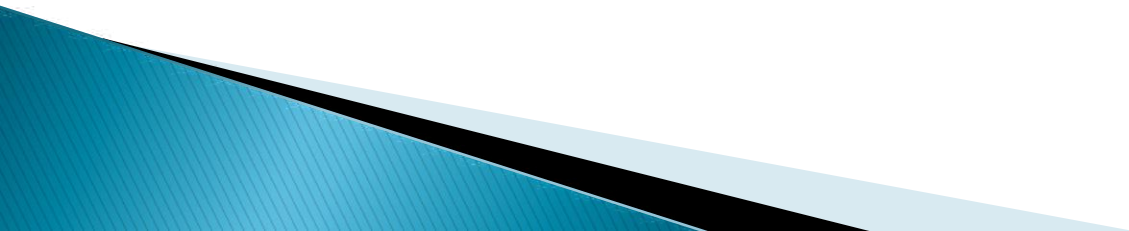
Touch supporting mobile browsers, usually interpret double tapping, as a zoom into this element gesture. Unfortunately, this means the platform, delays firing click of it until it can decide whether, the user is double tapping.

This delay is about a third of a second.



Chrome has recently checked in some changes, that minimize this behavior.

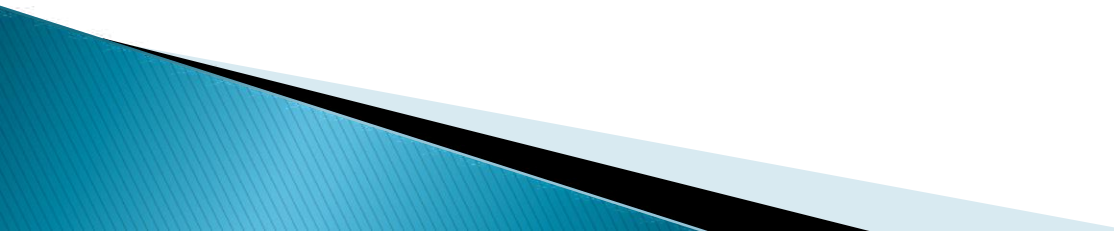
But, pretty much, all mobile browsers, have this click delay feature, to some degree.



Fix

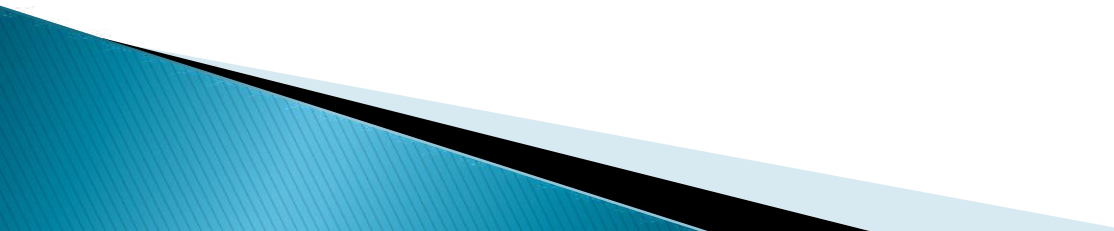
There are a few ways to fix this problem, and, get clicks without a delay,

1) You can set the view port, to be non-scalable. Either, set user scalable, to no, or set minimum scale and maximum scale, to one, which has the same effect. This can cause accessibility problems, though, so be careful jumping, to this solution. You need to make sure, if you do this, that your site will never, need to be zoomed.

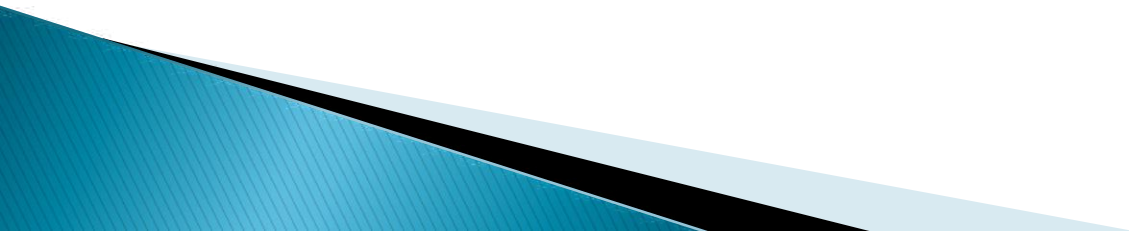


2) You can use a fast click library, like the ft labs one, that we've linked down below. But ,you do have to be careful, about how this impacts, your scrolling performance, be sure to read the directions, very carefully.

<https://github.com/ftlabs/fastclick>



3) Where finally, of course, you can implement tech support yourself directly, if you consume the touch events, you won't have this issue.



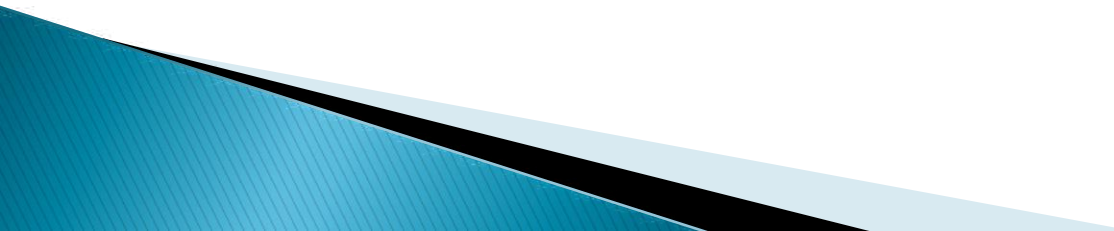
Supporting Touch



Go to www.caniuse.com

And search for “touch”

Or go to <http://caniuse.com/#feat=touch>



Touch Events

- ▶ touchstart -> mousedown
- ▶ touchmove -> mousemove
- ▶ touchend -> mouseup
- ▶ touchcancel

Touch Events

```
interface TouchEvent : UIEvent {  
    readonly attribute TouchList touches;  
    readonly attribute TouchList targetTouches;  
    readonly attribute TouchList changedTouches;  
    readonly attribute boolean altKey;  
    readonly attribute boolean metaKey;  
    readonly attribute boolean ctrlKey;  
    readonly attribute boolean shiftKey;  
};
```

Source: <https://www.w3.org/TR/touch-events/#touchevent-interface>

```
interface Touch {  
    readonly attribute long identifier;  
    readonly attribute EventTarget target;  
    readonly attribute long screenX;  
    readonly attribute long screenY;  
    readonly attribute long clientX;  
    readonly attribute long clientY;  
    readonly attribute long pageX;  
    readonly attribute long pageY;  
};
```

Source: <https://www.w3.org/TR/touch-events/#touch-interface>

Touch Events are Fired on The Original Target

Touch events are always delivered to the element that first received that Touch.

They don't walk across boundaries like Mouse Events do. So it's important to hook the right elements. In fact, it's a good idea to add the touch end handler during your touch start handler. And, keep in mind too that even if you remove a DOM element from your tree, it still gets the events until the touch ends.

DOM http://www.w3schools.com/js/js_htmlDOM.asp



Mouse Emulation

Example:

<https://webaudiodemos.appspot.com/midi-synth/>



Synthesizer Application

Application Link

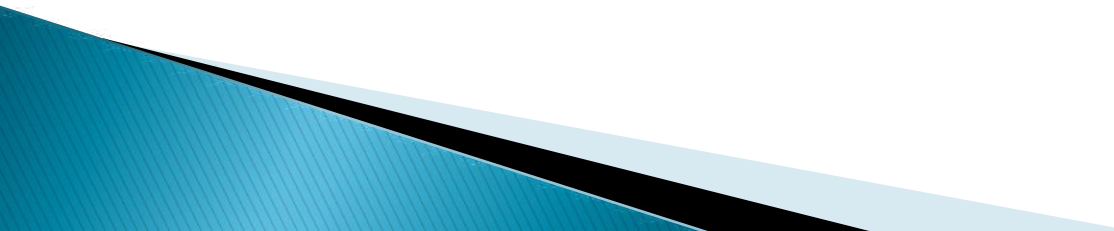
<https://webaudiodemos.appspot.com/>

Source Code Download Link

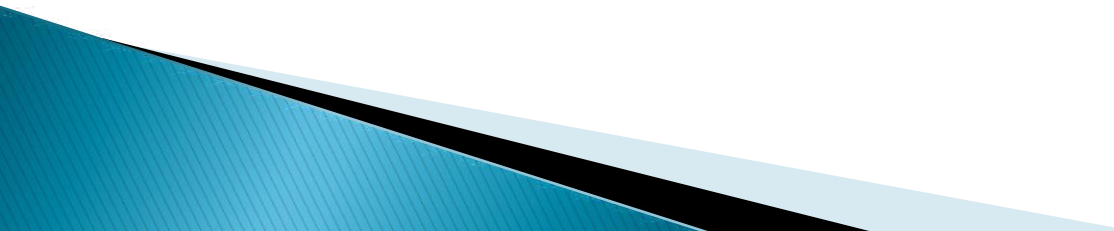
<https://github.com/cwilso/midi-synth>

We will Initially start with touch events.

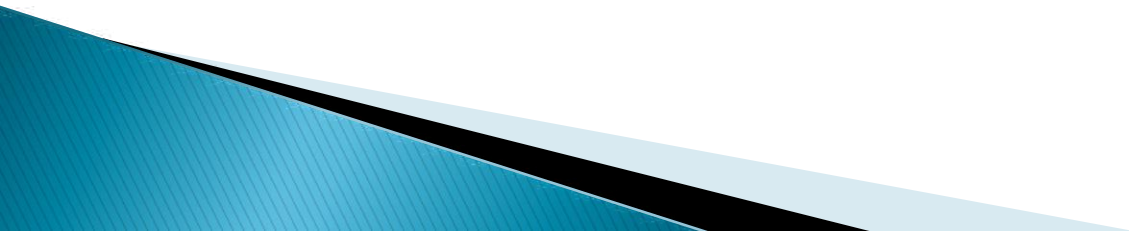
There were two challenges, though. First, you'll notice that on the desktop, if I don't have that touch event emulation turned on, it doesn't actually work at all when I use the mouse.



Of course, this is expected. But I didn't really want two code paths, one for mouse and one for touch. Secondly, I really wanted to be able to have a drag across the keyboard. Play each note as you slid across it in turn. But with the way touch delivers all the events to the originally touched down element, I would have had to calculate the hit testing myself for these. And that was kind of a pain.



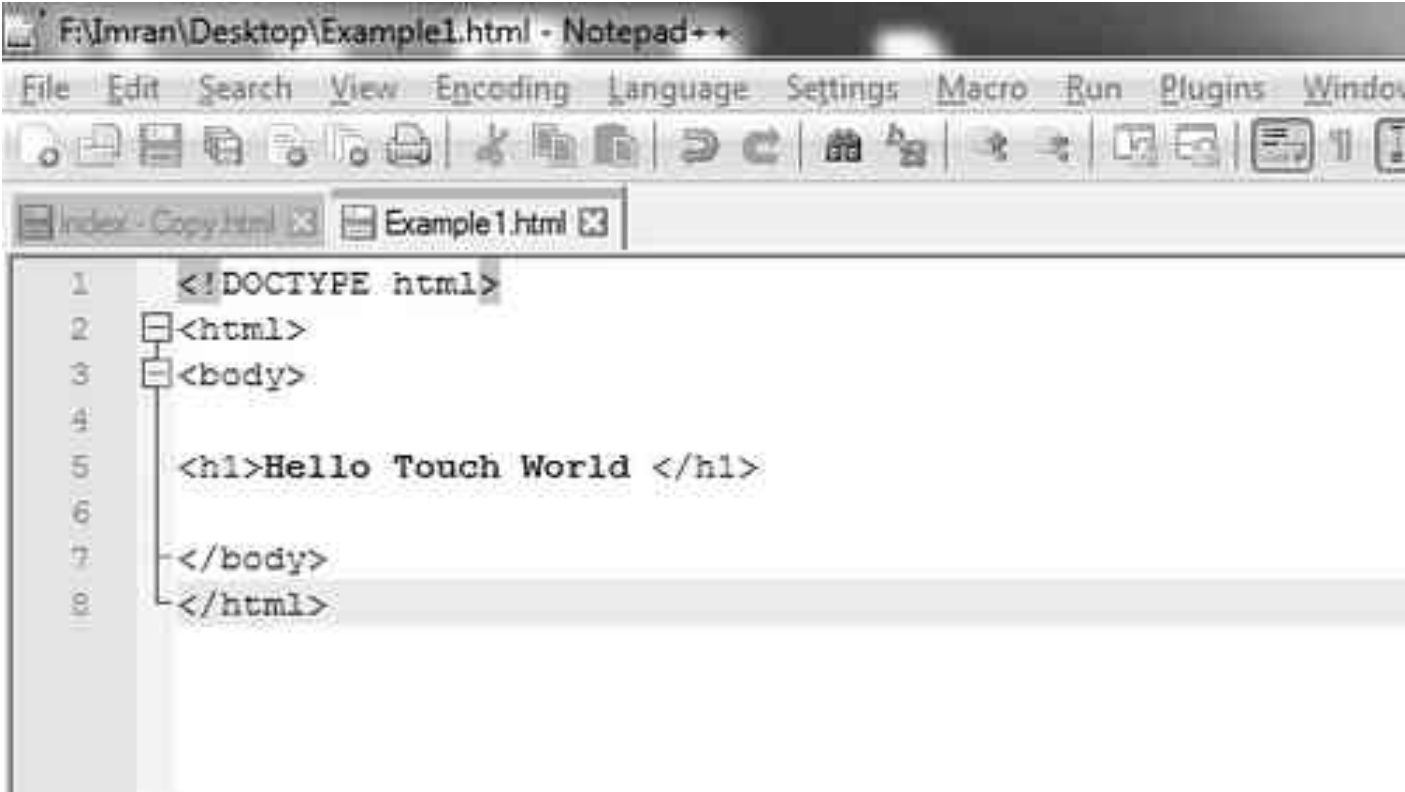
I really wanted these touch events to be delivered to the down element for each key. Instead what happens in this synth, is when I hit one key and then drag, it doesn't actually move where the events are delivered.



Getting Started With Touch Events



Example – 1



```
F:\Imran\Desktop\Example1.html - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window
Index - Copy.html Example1.html
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Hello Touch World </h1>
6
7 </body>
8 </html>
```

- ▶ We define a *touch-sensitive* div to which we will attach an event listener. We also define a div at the top where we will display the screen coordinates of the most recent touch.

```
1 <div id="coords"></div>
2 <div id="touchzone"></div>
```

F:\Imran\Desktop\Example1.html - Notepad++

File Edit Search View Encoding Language Settings Macro

index - Copy.html Example1.html

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Hello Touch World </h1>
6
7 <div id="coords"></div>
8 <div id="touchzone"></div>
9
10 </body>
11 </html>
```

- ▶ Next we add some JavaScript to register the event listener, and add a handler function to do something with the touch data:
- ▶ So, here we're simply identifying the touchzone div, attaching a listener for the touchstart event, and registering a handler function: touchHandler.

```
1 function init() {  
2     // Get a reference to our touch-sensitive element  
3     var touchzone = document.getElementById("touchzone");  
4     // Add an event handler for the touchstart event  
5     touchzone.addEventListener("touchstart", touchHandler, false);  
6 }
```

In the touchHandler function we grab the x and y coordinates of the touch, and write them to the coords div:

```
1 function touchHandler(event) {  
2   // Get a reference to our coordinates div  
3   var coords = document.getElementById("coords");  
4   // Write the coordinates of the touch to the div  
5   coords.innerHTML = 'x: ' + event.touches[0].pageX + ', y: ' + event.touches[0].pageY;  
6 }
```

Adding Mouse Support

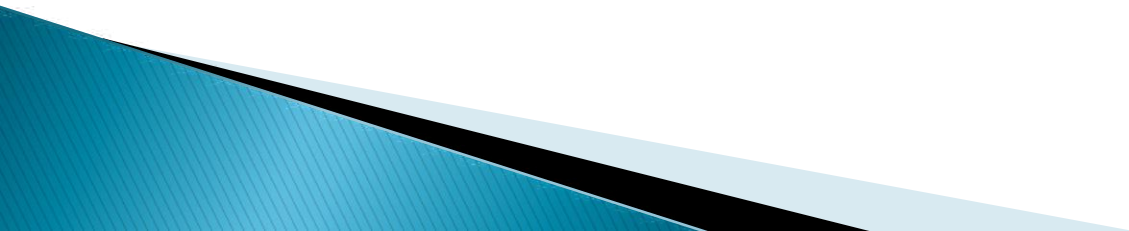


Pointer Events



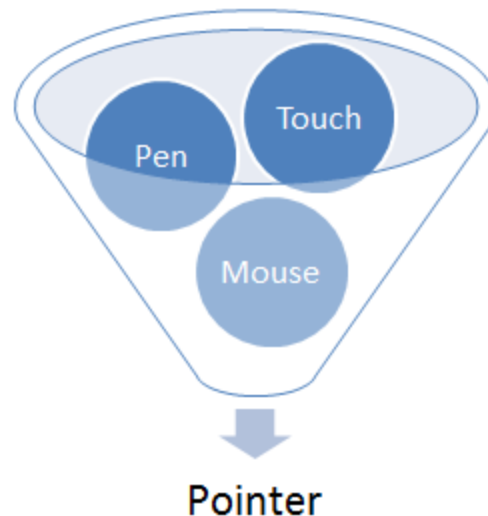
Problem

To handle the mouse events, touch events and pen events we have to write the code separately for each of these.



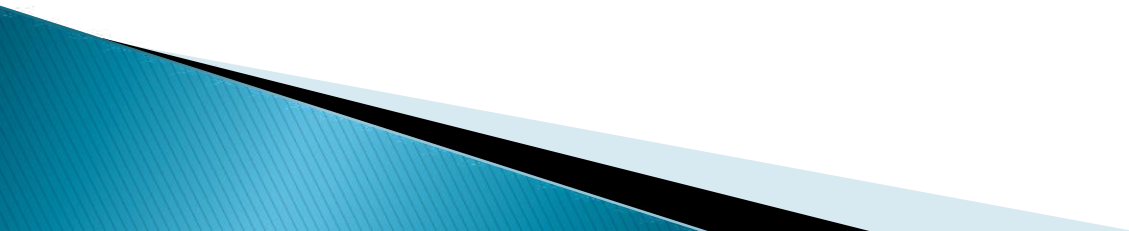
Solution

To rectify this situation, Microsoft made a standards proposal that reconciles touch events and mouse events and pen input too, into one model called *pointer events*.



Support in Browsers

Go to www.caniuse.com

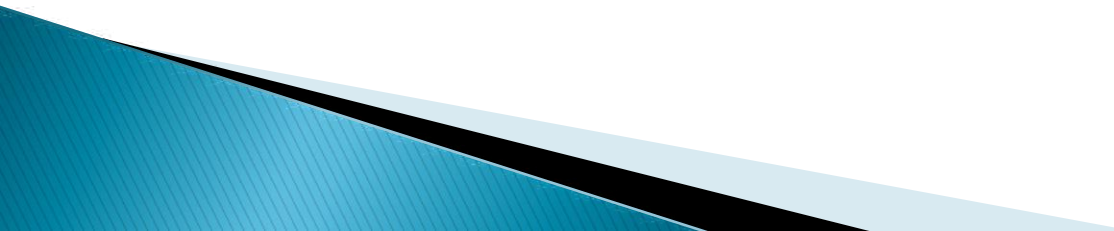


Pointer Events Object

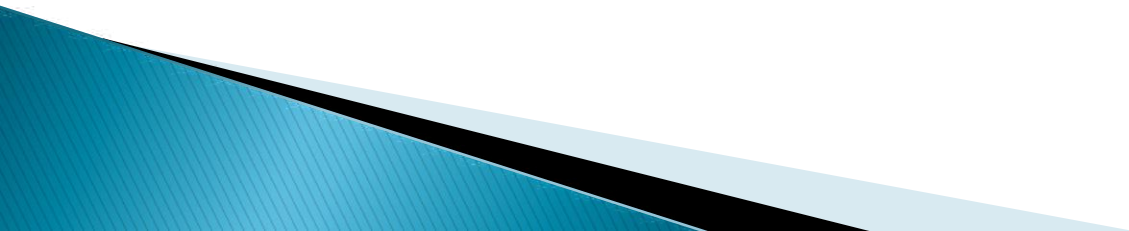
While the Touch Events API was defined in terms of *Touches*, the Pointer Events API is defined in terms of *Pointers*,

where a Pointer is defined as:

A hardware agnostic representation of input devices that can target a specific coordinate (or set of coordinates) on a screen



PointerEvent inherits and extends MouseEvent, so it has all the usual properties that MouseEvent has, such as clientX, clientY etc., as well as a few additional ones, such as tiltX, tiltY, and pressure



The pointer attributes

Attribute	Description
pointerId	unique numeric identifier
screenX	horizontal coordinate relative to screen
screenY	vertical coordinate relative to screen
clientX	horizontal coordinate of point relative to viewport, excluding scroll offset
clientY	vertical coordinate of point relative to viewport, excluding scroll offset
pageX	horizontal coordinate of point relative to page, including scroll offset
pageY	vertical coordinate of point relative to page, including scroll offset
width	width of pointer contact on screen
height	height of pointer contact on screen
tiltX	angle of tilt of stylus between Z and X axes, where X and Y plane is on the surface of the screen
tiltY	angle of tilt of stylus between Z and Y axes, where X and Y plane is on the surface of the screen
pressure	pressure of contact on screen
pointerType	class of Pointer: mouse, pen, or touch
isPrimary	is this the main Pointer for a pointer type

Event types defined by the PointerEvent interface

event type	fired when...
pointerover	pointer moves over an element (enters its hit test boundaries)
pointerenter	pointer moves over an element or one of its descendants. Differs to pointerover in that it doesn't bubble
pointerdown	<u>active buttons state</u> is entered: for touch and stylus, this is when contact is made with screen; for mouse, when a button is pressed
pointermove	pointer changes coordinates, or when pressure, tilt, or button changes fire no other event

pointerup	active buttons state is left: i.e. stylus or finger leaves the screen, or mouse button released
pointercancel	pointer is determined to have ended, e.g. in case of orientation change, accidental input e.g. palm rejection, or too many pointers
pointerout	pointer moves out of an element (leaves its hit test boundaries). Also fired after pointerup event for no-hover supported devices, and afterpointercancel event
pointerleave	pointer moves out of an element and its descendants
gotpointercapture	when an element becomes target of pointer
lostpointercapture	when element loses pointer capture

Mouse events, pointer events, and touch events equivalence

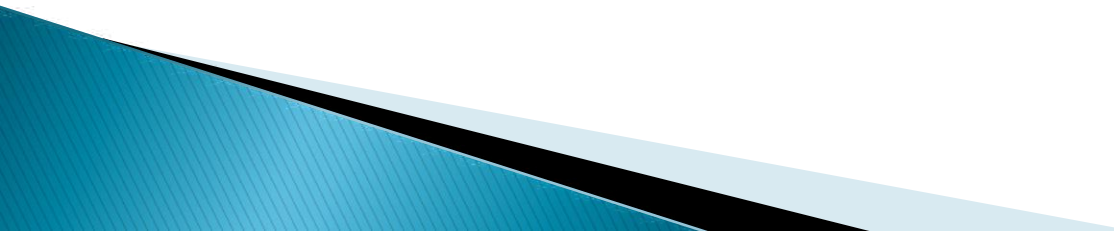
For comparison, the following table shows the corresponding events from each of these input related APIs

<u><i>Mouse event</i></u>	<u><i>Touch event</i></u>	<u><i>Pointer event</i></u>
mousedown	touchstart	pointerdown
mouseenter		pointerenter
mouseleave		pointerleave
mousemove	touchmove	pointermove
mouseout		pointerout
mouseover		pointerover
mouseup	touchend	pointerup

Polyfill Library

Polyfill Library

It refers to a JavaScript **library** that implements an **HTML5** web standard, either an established standard (supported by some browsers) on older browsers, or a proposed standard (not supported by any browsers) on existing browsers.



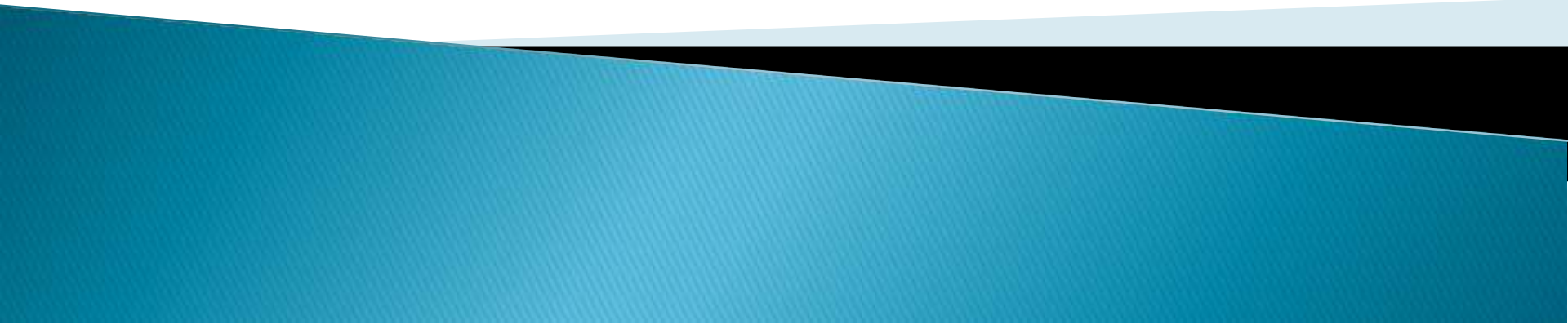
Pointer Code Example

Pointer Code Example

```
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>PEP (Pointer Events Polyfill)</title>
  <meta name="viewport" content="width=device-width">
  <!-- include PEP -->
  <script src="https://code.jquery.com/pep/0.4.1/pep.js"></script>
</head>
<body>
<button id="b" touch-action="none">Test button!</button>
<p><output id="o"></output></p>
<script>
document.getElementById( "b" ).addEventListener( "pointerdown", function( e ) {
  document.getElementById( "o" ).innerHTML = "that was a " +
    e.pointerType + " " + e.type + " on a " + e.target.nodeName;
});
</script>
</body>
</html>
```

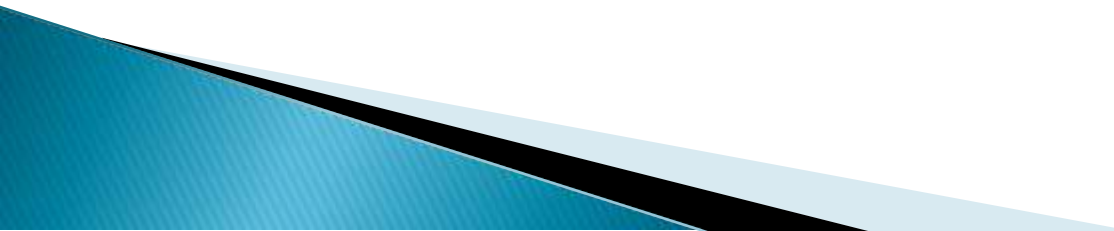
Pointer Code Example 2

Touch Handler Guide Lines



Touch Handler Guide Lines

It is good practice To use as few handlers as necessary in your application and keep the areas with touch handlers as tightly constrained as you can. That way the default scrolling mechanisms can take over as much as possible.



Input Introduction

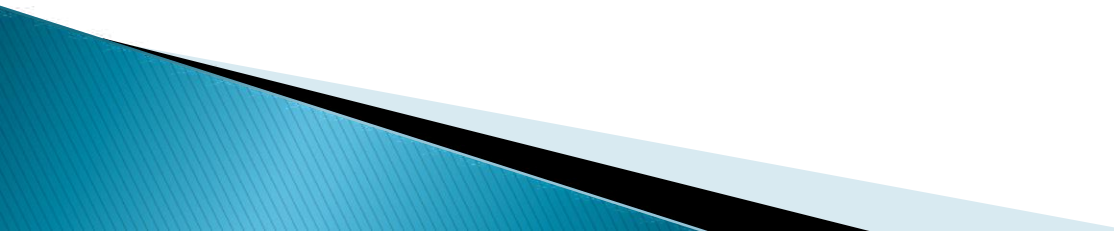
Input Introduction

- ▶ Some users can easily type longer text on mobile devices.
- ▶ But it is still very touch to just type with thumbs.

- ▶ So we must make of Input entry easy for all the users.
- ▶ Also while entering data it should be clearly visible. It is much difficult to enter the small font data.

Keyboard Input

Phones and tablets do not have the benefit most desktops and laptops do, which is a large input device called the "keyboard". Virtual keyboards offered by mobile devices are physically smaller and usually more tedious to operate, but the great news is that HTML5 has already standardized a handful of input types which make them easier to use.



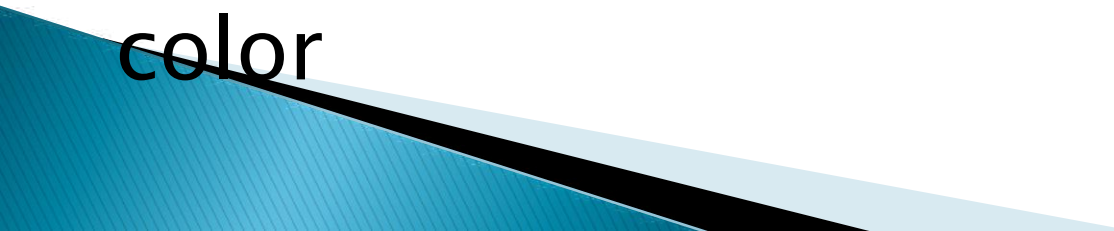
Input Types in HTML5

search email url tel

number range date month

week time datetime datetime-
local

color



Text

The most common input type

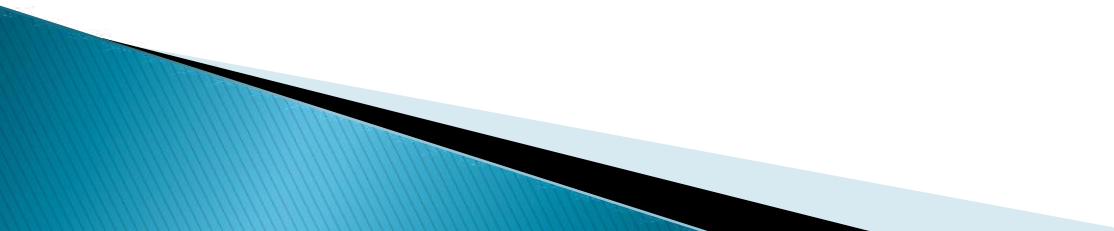
```
<input type="text">
```



Email

Entering an email on the standard keyboard is very difficult. When we use Input Type as email. We get a @ sign as well as the .com and other handy buttons to just make entering an email address as easy as possible.

```
<input type="email">
```





Tel

Tel input provides the user with ability to add the numbers , it provides ability to quickly add *, # , + signs as well.

```
<input type="tel">
```



O2-UK 17:23 99%

Tel

Input Tel

Previous

Next

Done

1

2

3

ABC

DEF

4

GHI

5

JKL

6

MNO

7

PQRS

8

TUV

9

WXYZ

+ * #

0

⌫

Number

Number input provides a key board to quickly add the numbers.

```
<input type="number">
```



Date

Date input types makes it very easy to enter the date on Webpage. Using Calendar can be little difficult for users

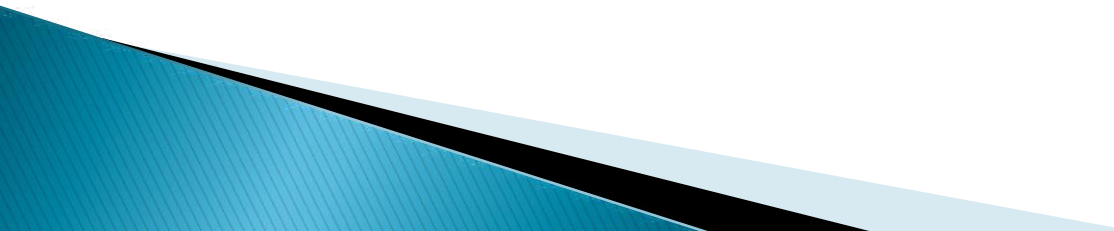
```
<input type="date">
```



Date & Time

Entering Date and Time is very Difficult but for mobile devices, using Date & Time input it becomes very easy to enter data.

```
<input type="datetime">
```



O2-UK



17:31



98%



Date & Time

2 Sep 2012 17:31 ▾

Previous

Next

Clear

Done

Fri 31 Aug 15 29

Sat 1 Sep 16 30

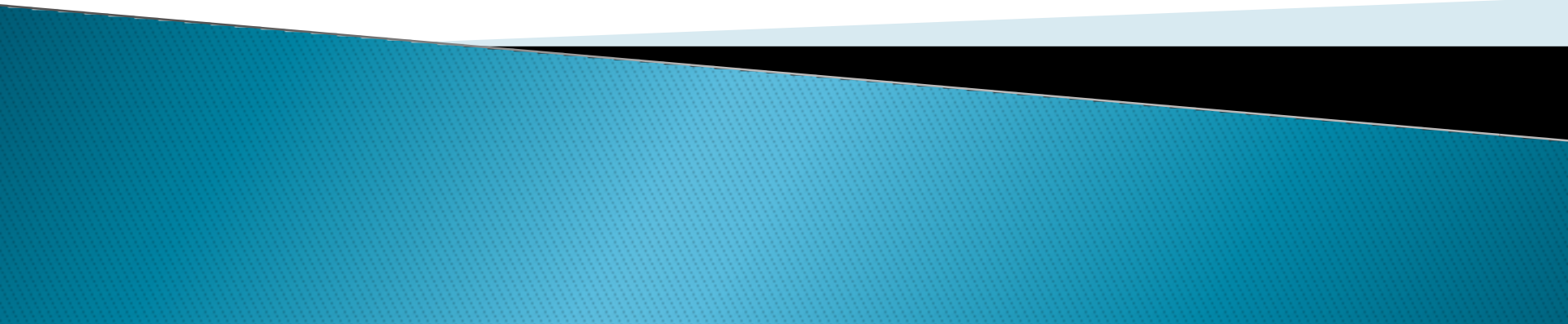
Today 17 31

Mon 3 Sep 18 32

Tue 4 Sep 19 33

- ▶ **Month**
- ▶ **Search**

Regular Expression Validation



For some types like e-mail, there's some basic validation. But this is not very smart validation.

If you want to do better client side validation though, HTML-5 supports that too.

There's also a pattern attribute on the input element, that you can use to validate on the client side.



The pattern element takes a regular expression, as expressed by the java script regex syntax.

Example

For example, let's presume we want to only allow Country code in a specific field in our mobile web application like PAK

```
<form action="demo_form.asp">  
  Country code: <input type="text" name="country_code" pattern="[A-Za-z]{3}"  
  title="Three letter country code">  
  <input type="submit">  
</form>
```

Country code:



Please match the requested
format.

Three letter country code

I can put in a regular expression for this, and now when I try to input a Country code of more than three letters and then submit, it pops up an error message that says I have to match the requested format.



Example 2

An `<input>` element with `type="password"` that must contain 6 or more characters:

```
<form action="demo_form.asp">  
  Password: <input type="password" name="pw" pattern=".{6,}" title="Six or more  
characters">  
  <input type="submit">  
</form>
```

Password:

Submit

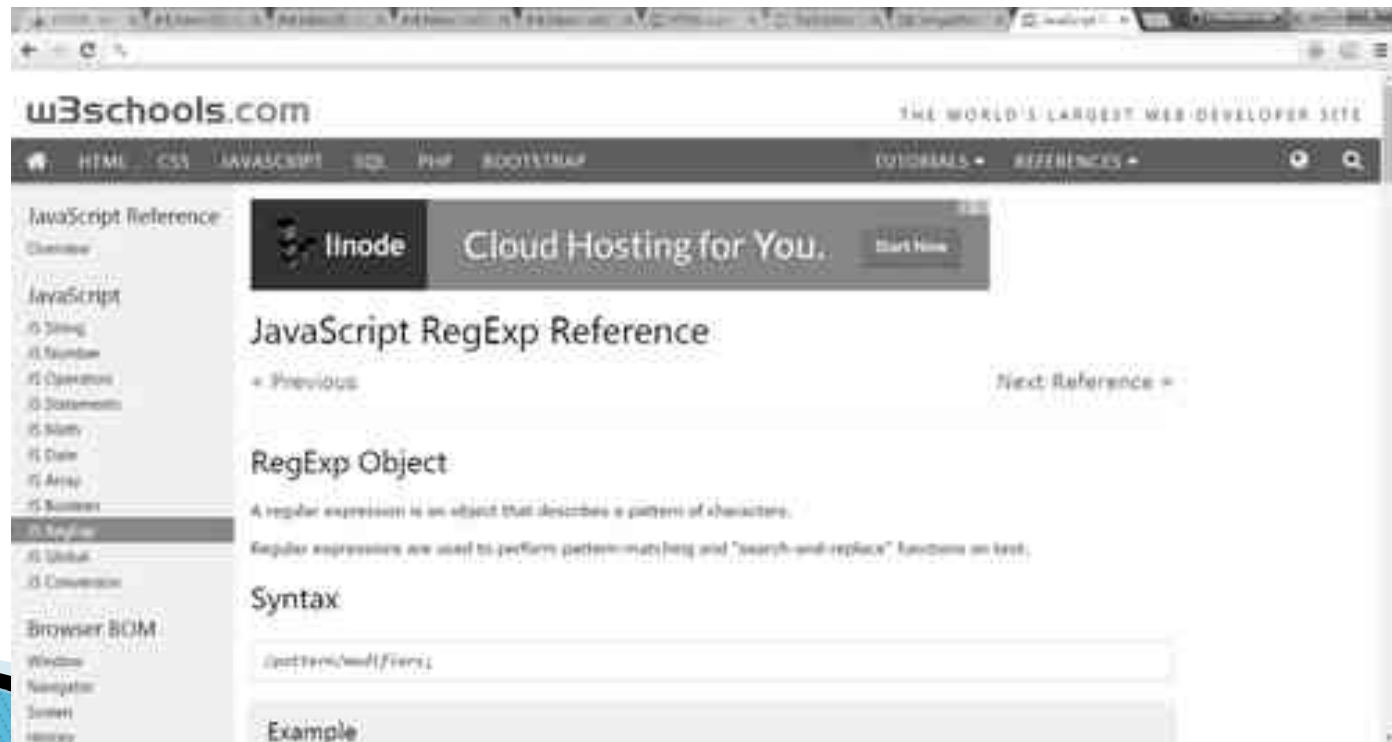


Please match the requested
format.

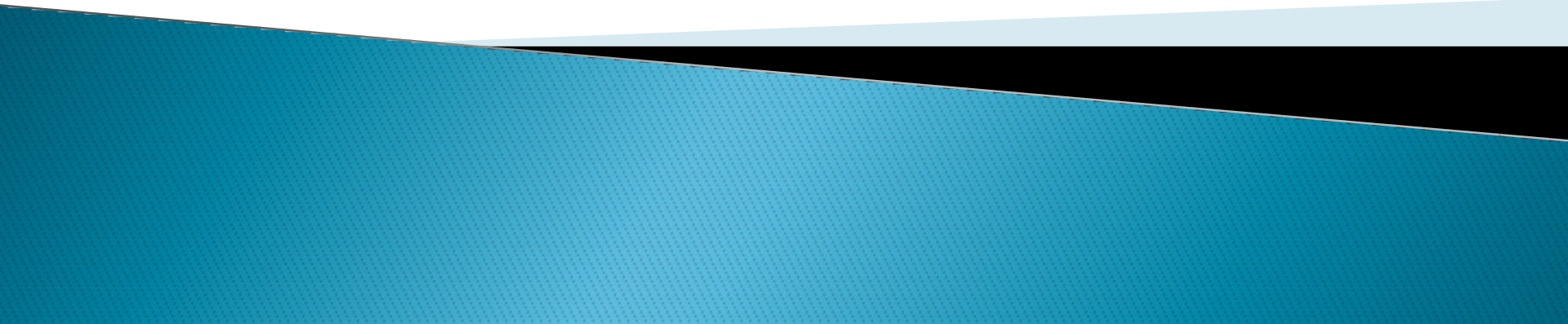
Six or more characters

To Learn more about Regular Expressions

- ▶ http://www.w3schools.com/jsref/jsref_obj_regexp.asp



Two Important Features of Input



Two Important Features of Input

- ▶ There are two more features I wanted to mention on inputs.
 - ▶ The required attribute and the place holder attribute.
 - ▶ Required just lets me say, this field is required before you can submit the form. So,
- 

Example

For example, if I want to make sure that a user enters the username,

```
<form action="demo_form.asp">  
  Username: <input type="text" name="usrname" required>  
  <input type="submit">  
</form>
```

Now if user will try to submit this form with out entering the username he will get the error message

Username:

! Please fill out this field.

Placeholder Text

The other feature, is the placeholder text. You may want to give hints to your users. In some fields, they go away when they start typing. You can do that pretty easily by just setting a placeholder attribute.



- ▶ The placeholder attribute specifies a short hint that describes the expected value of an input field (e.g. a sample value or a short description of the expected format).
 - ▶ The short hint is displayed in the input field before the user enters a value.
 - ▶ **Note:** The placeholder attribute works with the following input types: text, search, url, tel, email, and password.
- 

Example

```
<form action="demo_form.asp">  
  <input type="text" name="fname" placeholder="First name"><br>  
  <input type="text" name="lname" placeholder="Last name"><br>  
  <input type="submit" value="Submit">  
</form>
```

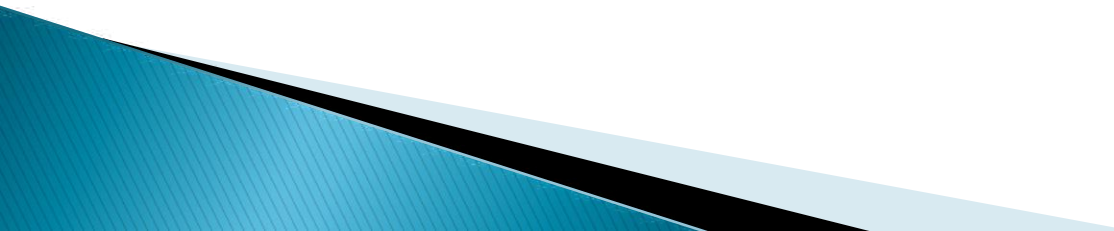
First name
Last name
Submit

Muhammad

Last name

Submit

Input in older browsers

- ▶ The HTML spec says that unrecognized input types are to be
 - ▶ treated as text. So, new input types will still work even if they are not supported on a particular browser.
 - ▶ They'll just function like text inputs. Not quite as useful, but still functional.
- 

Wrapping in Label

You should always wrap label elements around your input elements, and their associated labeling texts, because it increases the touchable area of the control.

