

Hell is other browsers - *Sartre*

The touch events

Peter-Paul Koch (ppk)

<http://quirksmode.org>

<http://twitter.com/ppk>

DIBI, 28 April 2010

The desktop web

Boring!

- Only five browsers
- with only one viewport each
- that support nearly everything
- Even IE? Yes, even IE.



The Mobile Web

Exciting!

- Fifteen browsers and counting
- ranging from great to lousy
- Interesting new bugs
- About five times as many users as the desktop web (eventually)
- New interaction modes

The Mobile Web

Exciting!

- Fifteen browsers and counting
- ranging from great to lousy
- Interesting new bugs
- About five times as many users as the desktop web (eventually)
- **New interaction modes**

Before we start

please open the following link on your iPhone or Android:

<http://quirksmode.org/touchevents>

It gives links to the test files.

Mouse



Mouse



Keyboard

Mouse

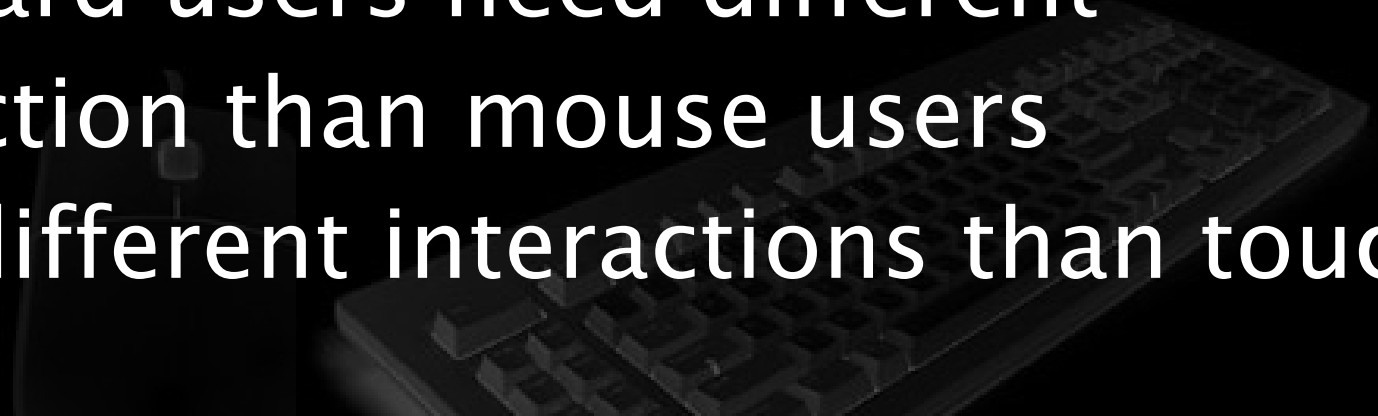


Keyboard



Touch

Keyboard users need different interaction than mouse users need different interactions than touch users.

A grayscale image of a computer keyboard and mouse, positioned in the upper half of the slide. The keyboard is a standard full-sized keyboard, and the mouse is a two-button mouse with a scroll wheel. The background is dark, making the white text stand out.

Your script accomodates all three modes, right?

A grayscale image of a hand holding a smartphone. The phone's screen displays a news article from The New York Times. The article title is "Chrysler on Verge of Sale to a Private Firm". The phone is held in a way that the hand is visible at the bottom and sides. The background is dark.

It's all a question of events.



keydown
keypress
keyup

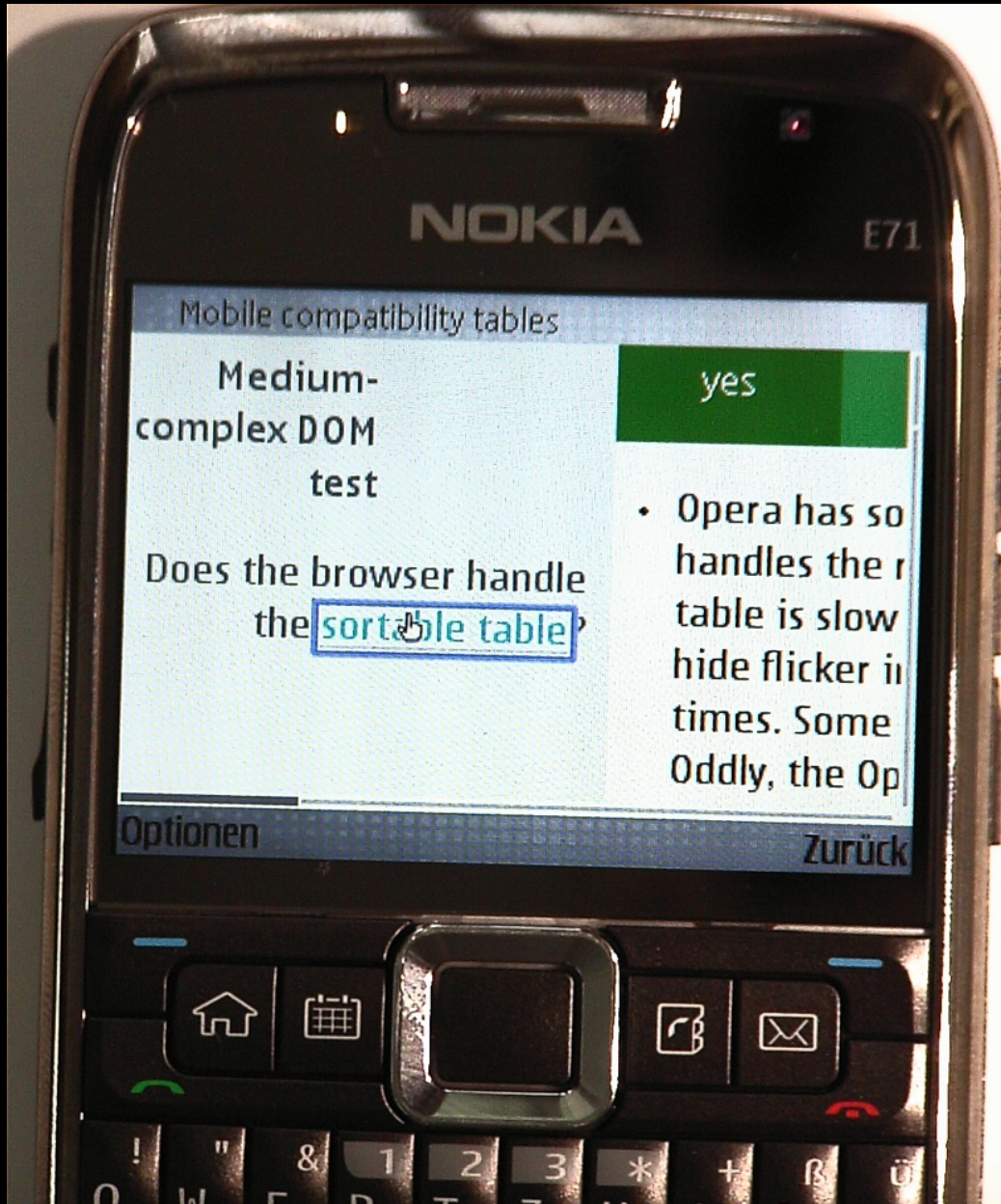


mouseover
mouseout
mousedown
mouseup
mousemove

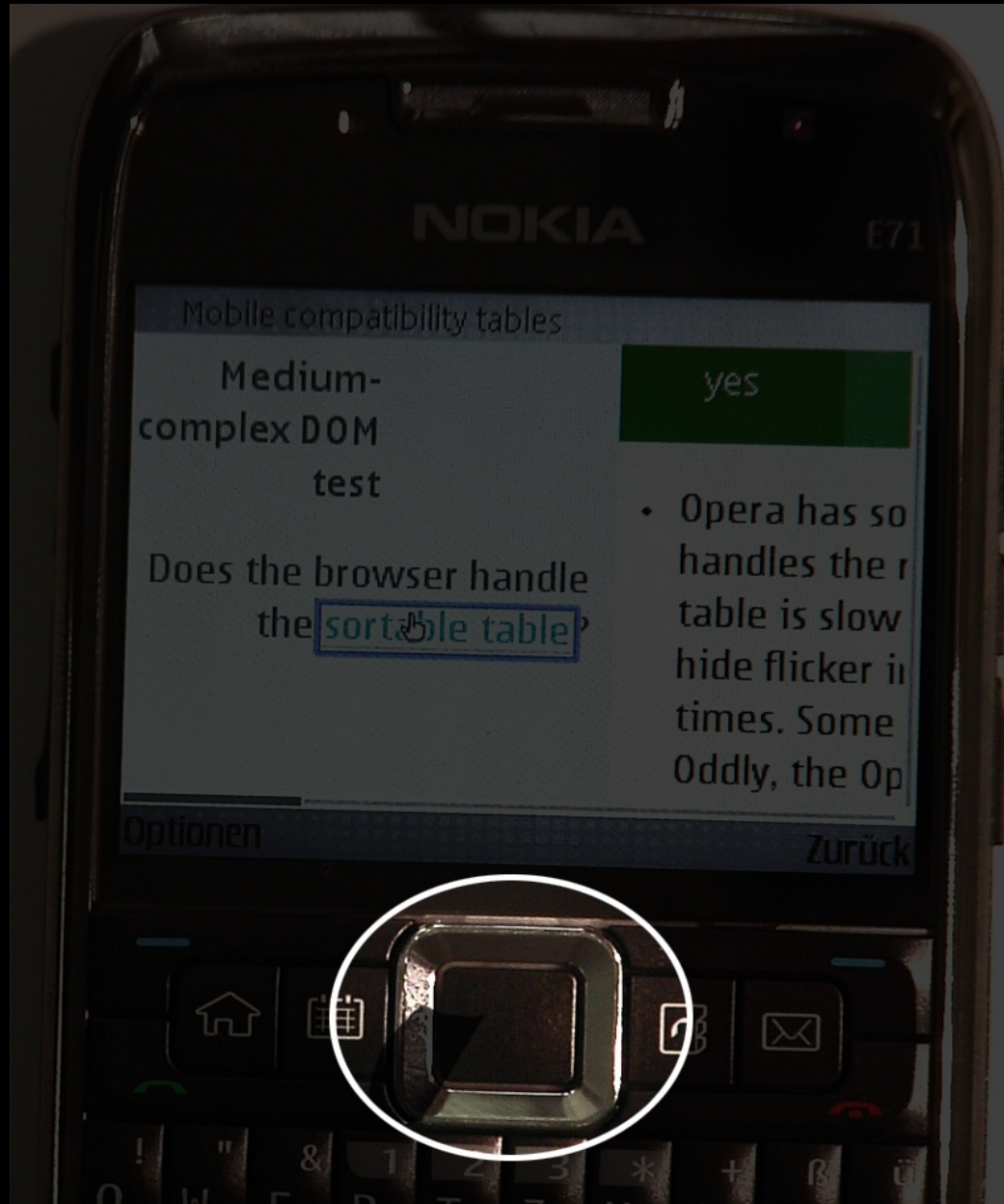
touchstart
touchmove
touchend
touchcancel



It's not an either-or proposition.



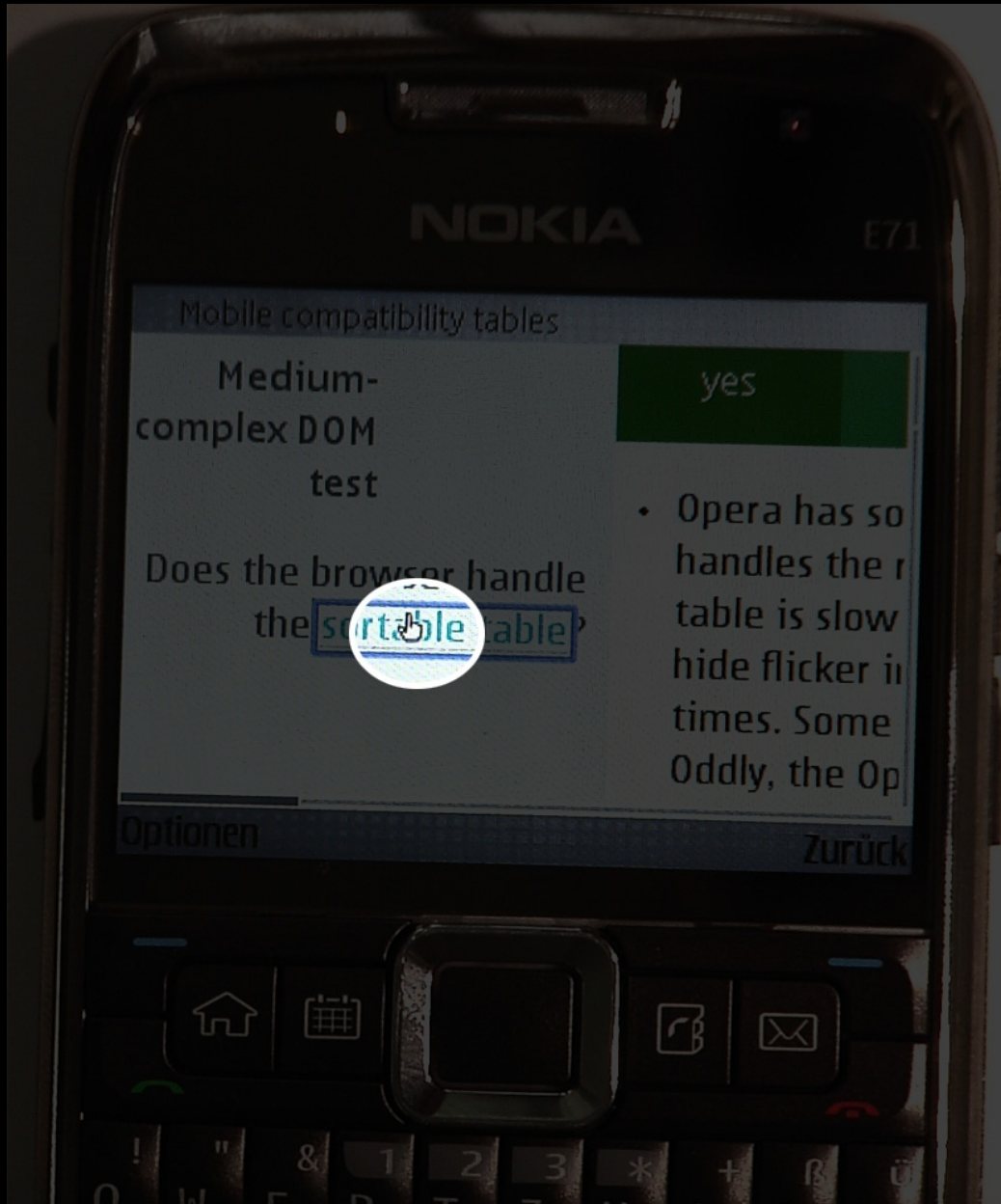
It's not an either-or proposition.



The Nokia E71 has a four-way navigation. Works like the arrow keys (including keycodes).

But...

It's not an either-or proposition.



But...
the “arrow keys”
steer a mouse
cursor.

Key events
and mouse
events

Today we'll concentrate on the touch events, though.



Touch !== mouse

- Area
- Pressure
- Temperature
- more than one touch



<http://quirksmode.org/touchevents>

Open the first dropdown example.

Task: Click on option 3.2

This is with traditional mouseover and mouseout; no touch-specific code.

Works (a bit oddly, but works).

```
dropdown.onmouseover = function (e) {  
  // open dropdown  
  dropdown.onmouseout = function (e) {  
    // close dropdown  
    // if appropriate  
    dropdown.onmouseout = null  
  }  
}
```

<http://quirksmode.org/touchevents>

Now open the second dropdown example.

Task: Click on option 3.2

Doesn't work.

```
dropdown.onmouseovertouchstart
= function (e) {
  // open dropdown
  dropdown.onmouseouttouchend
    = function (e) {
      // close dropdown
      // if appropriate
      dropdown.onmouseout = null
    }
}
```

Not an entirely fair comparison.

Not an entirely fair comparison.

Touchstart and touchend are not the equivalents of mouseover and mouseout.

Still, there is no better option.

Besides, it shows how different touch interaction is from mouse interaction.

Interaction modes

Mouse

mousedown

mousemove

mouseup

mouseover

mouseout

All

Keyboard

keydown

keypress

keyup

focus

blur

All

Touch

touchstart

touchmove

touchend

-

-

iPhone,

Android

There is no true hover on a touchscreen.

No way of saying “I might be interested in this element but I'm not sure yet.”

Instead, the mobile browsers fake mouseover/out and :hover.

(We'll see how later.)

Interaction modes

Mouse

Keyboard

Touch

mousedown

keydown

touchstart

mousemove

keypress

touchmove

mouseup

keyup

touchend

mouseover

focus

-

mouseout

blur

-

load, unload, click, submit, resize,
zoom, change etc. etc.

Interaction modes

Mouse

mousedown

mousemove

mouseup

mouseover

mouseout

Keyboard

keydown

keypress

keyup

focus

blur

Touch

touchstart

touchmove

touchend

-

-

load, unload, click, **submit**, resize,
zoom, change etc. etc.

Interaction modes

Mouse

mousedown

mousemove

mouseup

mouseover

mouseout

load, unload, click, submit, resize,
zoom, change etc. etc.

Keyboard

keydown

keypress

keyup

focus

blur

Touch

touchstart

touchmove

touchend

-

-

Interaction modes

Mouse

mousedown

mousemove

mouseup

mouseover

mouseout

load, unload, **click**, submit, resize,
zoom, change etc. etc.

Keyboard

keydown

keypress

keyup

focus

blur

Touch

touchstart

touchmove

touchend

-

-

In theory a touchscreen device should fire only the touch events, and not the mouse events.

However, too many websites depend on the mouse events, so touch browser vendors are forced to support them, too.

Therefore, when you touch the screen of a touchscreen, both touch and mouse events fire.

But the mouse events are a bit special. They all fire at the same time.

<http://quirksmode.org/touchevents>

You can test the events for yourself at the touch action test page.

touchstart

mouseover

mousemove (only one!)

mousedown

mouseup

click

:hover styles applied



When the user touches another element
mouseout
:hover styles removed

On the iPhone this element must listen
to events. If it doesn't, it's not clickable
and events do not fire.



touchstart If a DOM change occurs
mouseover onmouseover or
mousemove onmousemove, the rest
~~mousedown~~ of the events is cancelled.
~~mouseup~~ (iPhone and Symbian)
click

~~:hover styles applied~~



<http://quirksmode.org/touchevents>

Now open the first drag-and-drop example.

Should work fine; both on touch devices and with a mouse.

This is very simple.

```
element.onmousedown = function (e) {  
  // initialise  
  document.onmousemove = function (e) {  
    // move  
  }  
  document.onmouseup = function (e) {  
    document.onmousemove = null;  
    document.onmouseup = null;  
  }  
}
```

```
element.onmousedown = function (e) {  
  // initialise  
  document.onmousemove = function (e) {  
    // move  
  }  
  document.onmouseup = function (e) {  
    document.onmousemove = null;  
    document.onmouseup = null;  
  }  
}
```

Set `mousemove` and `mouseup` handlers only when `mousedown` has taken place.

May save some processing time; especially on mobile.

```
element.onmousedown = function (e) {  
  // initialise  
  document.onmousemove = function (e) {  
    // move  
  }  
  document.onmouseup = function (e) {  
    document.onmousemove = null;  
    document.onmouseup = null;  
  }  
}
```

Set mousemove and mouseup handlers on the document.

Helps when user moves too fast and “overshoots”: the script remains functional.

```
element.onmousedown = function (e) {  
  // initialise  
  document.onmousemove = function (e) {  
    // move  
  }  
  document.onmouseup = function (e) {  
    document.onmousemove = null;  
    document.onmouseup = null;  
  }  
}
```



```
element.ontouchstart = function (e) {  
  // initialise  
  document.ontouchmove = function (e) {  
    // move  
  }  
  document.ontouchend = function (e) {  
    document.ontouchmove = null;  
    document.ontouchend = null;  
  }  
}
```

But: how do we know which events to use? How do we know whether a user uses a mouse or a touchscreen?

```
element.onmousedown = function (e) {  
  document.onmousemove = etc.  
  document.onmouseup = etc.  
}
```

```
element.ontouchstart = function (e) {  
  document.ontouchmove = etc.  
  document.ontouchend = etc.  
}
```

```
element.onmousedown = function (e) {  
  document.onmousemove = etc.  
  document.onmouseup = etc.  
}
```

```
element.ontouchstart = function (e) {  
  element.onmousedown = null;  
  document.ontouchmove = etc.  
  document.ontouchend = etc.  
}
```

Remove the mousedown event handler when a touchstart takes place: now you're certain that the user uses a touchscreen.

<http://quirksmode.org/touchevents>

Now open the second drag-and-drop example.

iPhone only.

Try dragging two or all three layers simultaneously.

(A bit stilted, but you get the point.)

This is impossible on a desktop computer. Two mice?

Useful for games, maybe (especially on the iPad).

Does not work on Android: the browser doesn't (yet) support true multitouch.

<http://quirksmode.org/touchevents>

Now open the scrolling layer example.

Works fine – on mobile.

But how do we port this to the other interaction modes?

- keys: use arrow keys
- mouse: ???

Interaction modes

- mouse
- keyboard
- touch
- and a fourth....



Interaction modes

- mouse
- keyboard
- touch
- trackball

Generally fires a
mousemove event



Thank you!
Questions?

<http://quirksmode.org>

<http://twitter.com/ppk>