

Hell is other browsers - *Sartre*

JavaScript Events

Peter-Paul Koch (ppk)
<http://quirksmode.org>
<http://twitter.com/ppk>
Yahoo!, 23 April 2009

Event	IE 5.5	IE 6	IE 7	IE8b1	FF 2	FF 3b5	Saf 3.0 Win	Saf 3.1 Win	Opera 9.26	Opera 9.5b	Konqueror 3.5.7
<p>blur</p> <p>When an element loses the focus.</p> <ul style="list-style-type: none"> Firefox 2 fires too many events in a variety of circumstances. Firefox 3 fires too many events when blurring the window. Safari and Opera don't support these events on links and/or form fields in all circumstances. Konqueror doesn't support these events on the browser window. 	yes	yes	yes	yes	too many	almost	incomplete	almost	incomplete	incomplete	incomplete
<p>change</p> <p>When a form field value changes.</p> <ul style="list-style-type: none"> IE has a serious bug in its handling of this event on checkboxes and radios. 	buggy	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
<p>click</p> <p>When a mousedown and mouseup event occur on the same element OR an element is activated by the keyboard.</p>	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
Event	IE 5.5	IE 6	IE 7	IE8b1	FF 2	FF 3b5	Saf 3.0 Win	Saf 3.1 Win	Opera 9.26	Opera 9.5b	Konqueror 3.5.7
<p>contextmenu</p> <p>When the user right-clicks to get the context menu.</p> <p>Preventing the default (i.e. preventing the context menu from appearing) is the whole point of this event.</p>	yes	yes	minimal	yes	yes	buggy	yes	yes	no	no	no

Event	IE 5.5	IE 6	IE 7	IE8b1	FF 2	FF 3b5	Saf 3.0 Win	Saf 3.1 Win	Opera 9.26	Opera 9.5b	Konqueror 3.5.7
<u>blur</u>		yes			too many	almost	incomplete	almost	incomplete		incomplete

When an element loses the focus.

- Firefox 2 fires too many events in a variety of circumstances.
- Firefox 3 fires too many events when blurring the window.
- Safari and Opera don't support these events on links and/or form fields in all circumstances.
- Konqueror doesn't support these events on the browser window.

<u>change</u>	buggy				yes		yes		yes		yes
---------------	-------	--	--	--	-----	--	-----	--	-----	--	-----

When a form field value changes.

- IE has a serious bug in its handling of this event on checkboxes and radios.

<http://quirksmode.org/dom/events/>

<u>click</u>	yes				yes		yes		yes		yes
--------------	-----	--	--	--	-----	--	-----	--	-----	--	-----

When a mousedown and mouseup event occur on the same element OR an element is activated by the keyboard.

Event	IE 5.5	IE 6	IE 7	IE8b1	FF 2	FF 3b5	Saf 3.0 Win	Saf 3.1 Win	Opera 9.26	Opera 9.5b	Konqueror 3.5.7
<u>contextmenu</u>		yes		minimal	yes	buggy	yes		no		no

When the user right-clicks to get the context menu.

Preventing the default (i.e. preventing the context menu from appearing) is the whole point of this event.

Today's program:

- the key events
- the change event
- delegating the focus event
- first results of mobile event tests

Event	IE 5.5	IE 6	IE 7	IE8b1	FF 2	FF 3b5	Saf 3.0 Win	Saf 3.1 Win	Opera 9.26	Opera 9.5b	Konqueror 3.5.7
-------	--------	------	------	-------	------	--------	----------------	----------------	---------------	---------------	-----------------

blur	yes	yes	yes	yes	yes	yes	incomplete	almost	incomplete	incomplete	incomplete
------	-----	-----	-----	-----	-----	-----	------------	--------	------------	------------	------------

- Firefox 2 fires too many events in a variety of circumstances.
- Firefox 3 fires too many events when blurring the window.
- Safari and Opera don't support these events on links and/or form fields in all circumstances.
- Konqueror doesn't support these events on the browser window.

change	buggy	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
--------	-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

- IE has a serious bug in its handling of this event on checkboxes and radios.

click	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
-------	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

Event	IE 5.5	IE 6	IE 7	IE8b1	FF 2	FF 3b5	Saf 3.0 Win	Saf 3.1 Win	Opera 9.26	Opera 9.5b	Konqueror 3.5.7
-------	--------	------	------	-------	------	--------	----------------	----------------	---------------	---------------	-----------------

contextmenu

contextmenu	yes	yes	minimal	yes	buggy	yes	yes	yes	no	no	no
-------------	-----	-----	---------	-----	-------	-----	-----	-----	----	----	----

When the user right-clicks to get the context menu.

Preventing the default (i.e. preventing the context menu from appearing) is the whole point of this event.



The key events

A dark, close-up photograph of a computer keyboard, showing several keys in detail. The keys are dark with light-colored characters. The background is a dark, slightly blurred grid of keys.

keydown

When a key is depressed.

Repeats.

keypress

keyup



keydown

When a key is depressed.

Repeats.

keypress

When a *character* key is depressed.

Repeats.

keyup



keydown

When a key is depressed.

Repeats.

keypress

When a *character* key is depressed.

Repeats.

keyup

When a key is released.



keydown and keypress

keydown only



Originally this theory was created by Microsoft.

Safari has copied it.

It's the only theory; Firefox and Opera just fire some random events.

keydown

When a key is depressed.

Repeats.

keypress

When a *character* key is depressed.

Repeats.





Which key did my user press?

Two properties:
keyCode and charCode

Two bits of data:
- the key code
- the character code

Which key did my user press?

Obviously, having one property contain one bit of data and the other property the other

would be far too simple.

Which key did my user press?

keyCode

- onkeydown: key code
- onkeypress: character code



Which key did my user press?

charCode

- onkeydown: 0
- onkeypress: character code



Which key did my user press?

If you need the key:

```
el.onkeydown = function (e) {  
  e = e || window.event;  
  var realKey = e.keyCode;  
}
```

Which key did my user press?

If you need the key:

```
el.onkeydown = function (e) {  
  e = e || window.event;  
  var realKey = e.keyCode;  
}
```



Which key did my user press?

If you need the character:

```
el.onkeypress = function (e) {  
  e = e || window.event;  
  var char = e.keyCode || e.charCode;  
}
```

Which key did my user press?

If you need the character:

```
el.onkeypress = function (e) {  
  e = e || window.event;  
  var char = e.keyCode || e.charCode;  
}
```



How can I prevent the default action?

```
el.onkeydown = function (e) {  
  e = e || window.event;  
  var key = e.keyCode;  
  if (key is incorrect) {  
    // cancel default action  
  }  
}
```

How can I prevent the default action?

```
el.onkeydown = function (e) {  
  e = e || window.event;  
  var key = e.keyCode;  
  if (key is incorrect) {  
    // cancel default action  
  }  
}
```

W3C



Registration

- Traditional
- addEventListener (capture)
- addEventListener (bubble)

Events

The following events are registered:

- change

Elements

The events are registered on the following elements:

- document
- form
- text
- checkbox

Miscellaneous

- Prevent default action
- Cancel bubble

Event properties

- Show event properties

change

The change event fires when the value of a form field is changed.

This could be a very useful event; after all it fires only when the user actually changes something instead of whenever he focuses on a form field



- text fields
- select boxes
- checkboxes and radios

show page contents

Registration

- Traditional
- addEventListener (capture)
- addEventListener (bubble)

The following events are registered:

- change

Elements

The events are registered on the following elements:

- window
- #document
- form
- text
- checkbox

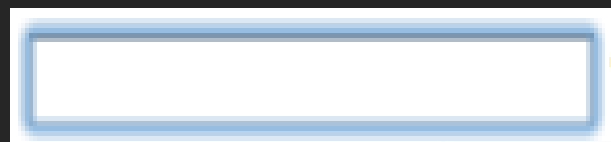
Miscellaneous

- Prevent default action
- Cancel bubble

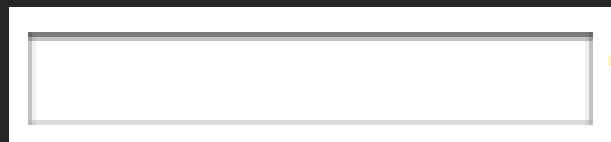
Event properties

- Show event properties

- text fields
- select boxes
- checkboxes and radios



focus



blur

No change event. The value hasn't been modified.

show page contents

Registration

- Traditional
- addEventListener (capture)
- addEventListener (bubble)

Events

The following events are registered:

- change

Elements

The events are registered on the following elements:

- window
- #document
- form
- text
- checkbox

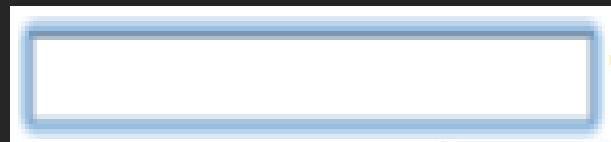
Miscellaneous

- Prevent default action
- Do not focus

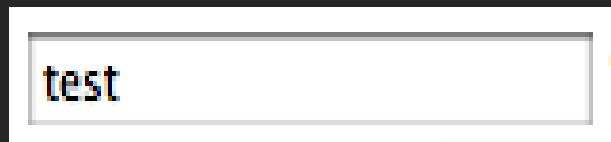
Event properties

- Show event properties

- text fields
- select boxes
- checkboxes and radios



focus



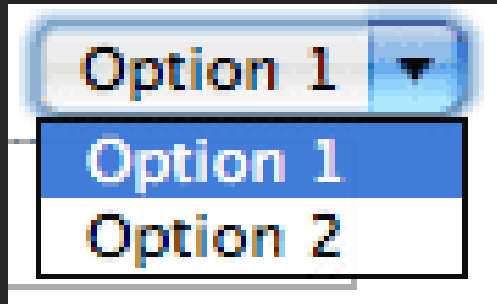
blur

Change event. The value has been modified.



- text fields
- select boxes
- checkboxes and radios

Mouse:



show page contents

Registration

- Traditional
- addEventListener (capture)
- addEventListener (bubble)

Events

The following events are registered:

- change

Elements

The events are registered on the following elements:

- window
- #document
- form
- text
- checkbox

Miscellaneous

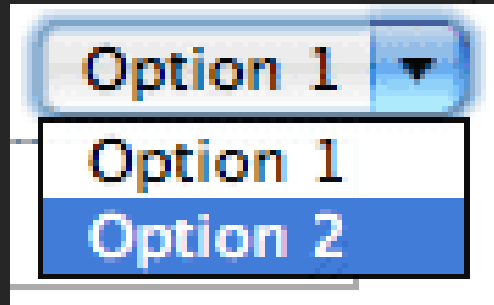
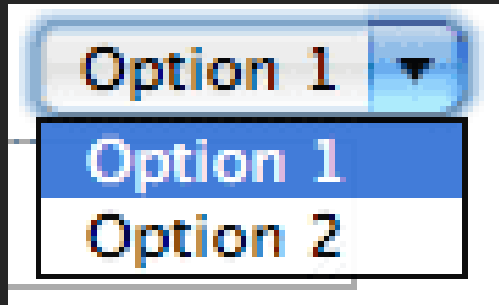
- Prevent default action
- Cancel bubble

Event properties

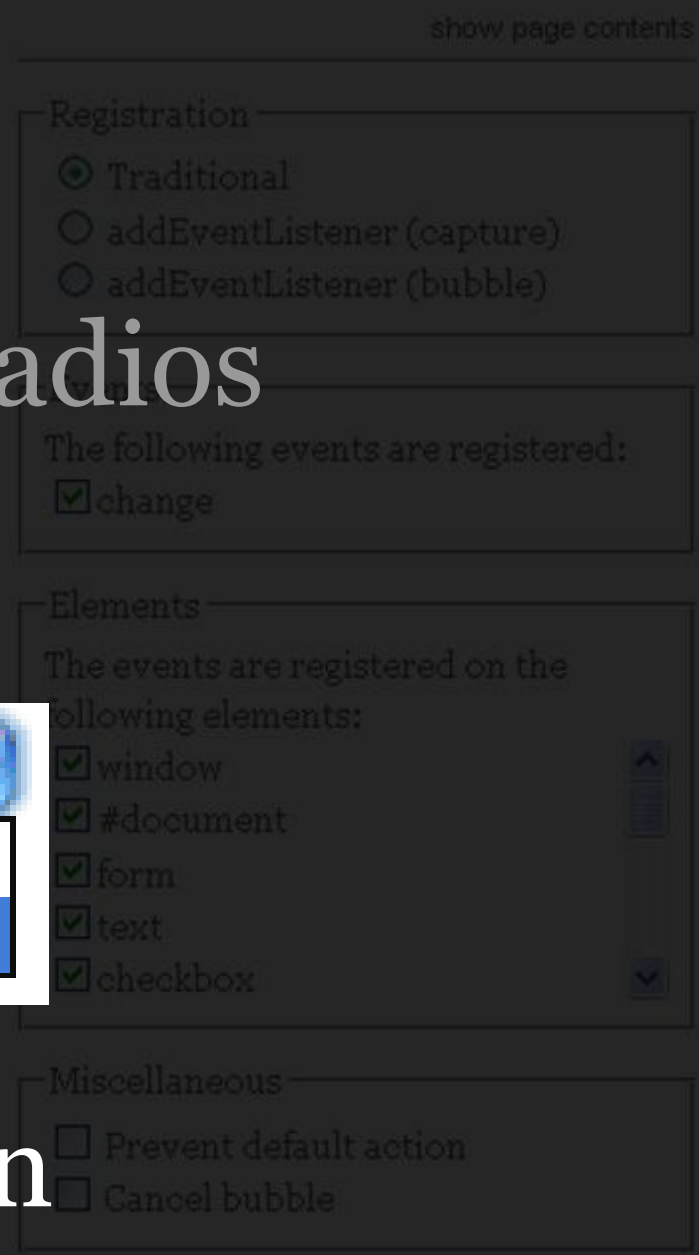
- Show event properties

- text fields
- select boxes
- checkboxes and radios

Mouse:

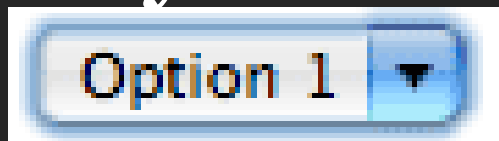


Click on new option
CHANGE



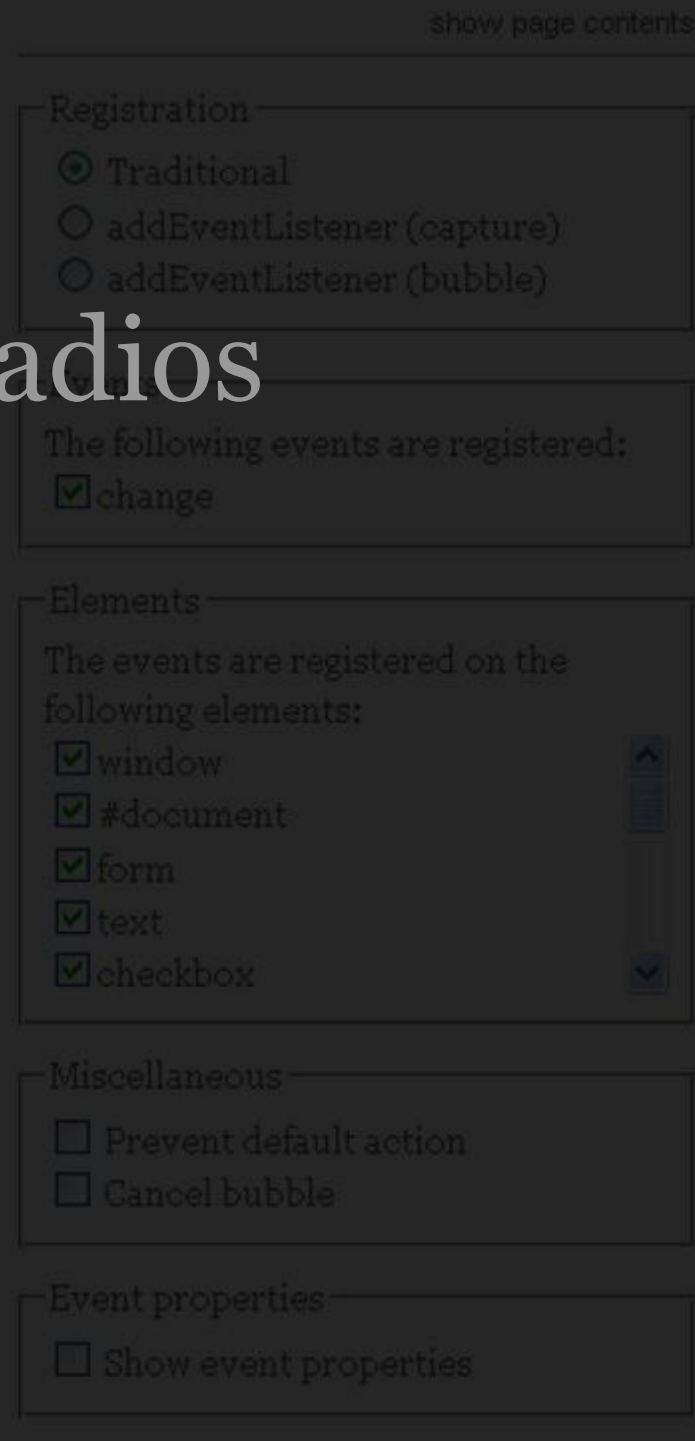
- text fields
- select boxes
- checkboxes and radios

Keyboard:



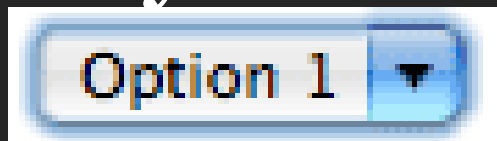
focus

Focus on select

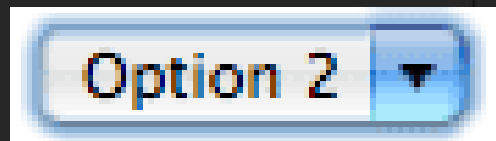


- text fields
- select boxes
- checkboxes and radios

Keyboard:

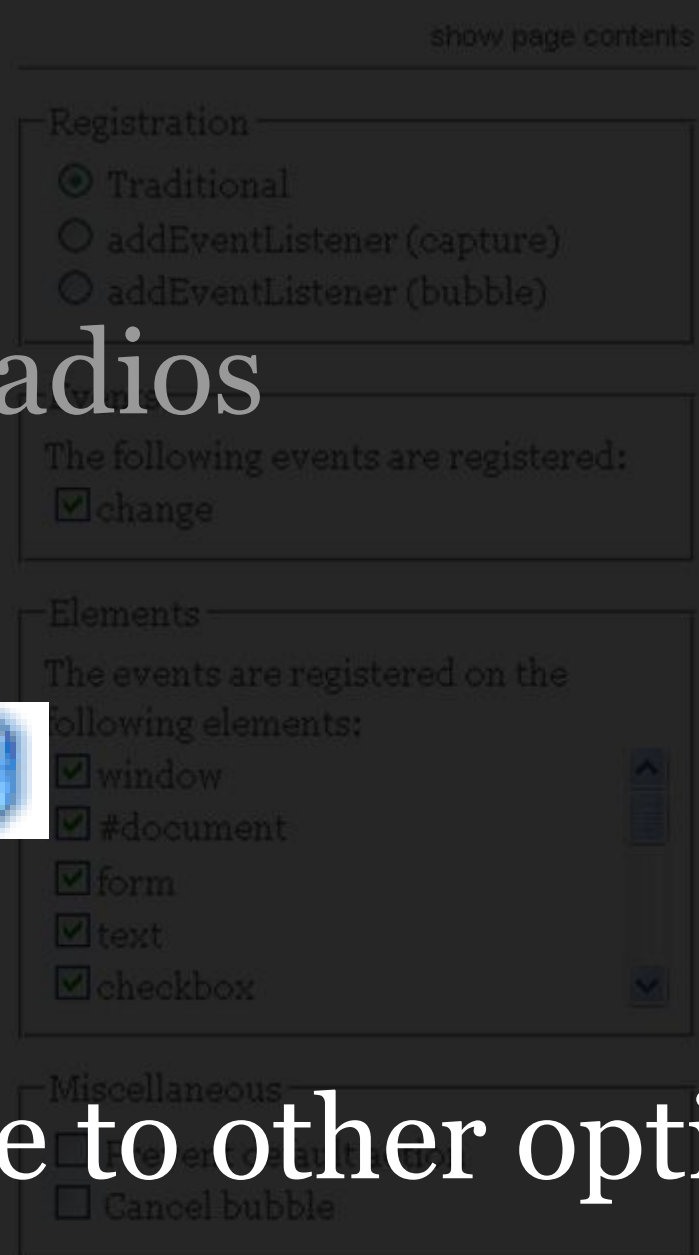
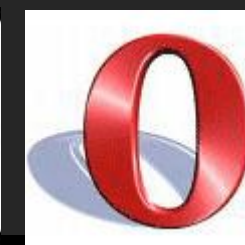


focus



arrow

Arrow keys to move to other option
CHANGE



- text fields
- select boxes
- checkboxes and radios

This is a
BUG!

Arrow keys to move to other option
CHANGE



show page contents

Registration

- Traditional
- addEventListener (capture)
- addEventListener (bubble)

The following events are registered:

- change

Elements

The following events are registered on the following elements:

- window
- #document
- form
- text
- checkbox

Miscellaneous

- Prevent default
- Cancel bubble

Properties

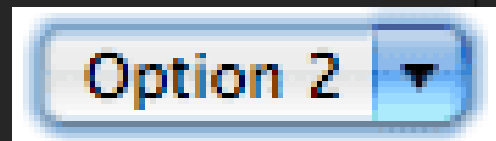
Event properties

- text fields
- select boxes
- checkboxes and radios

Keyboard:

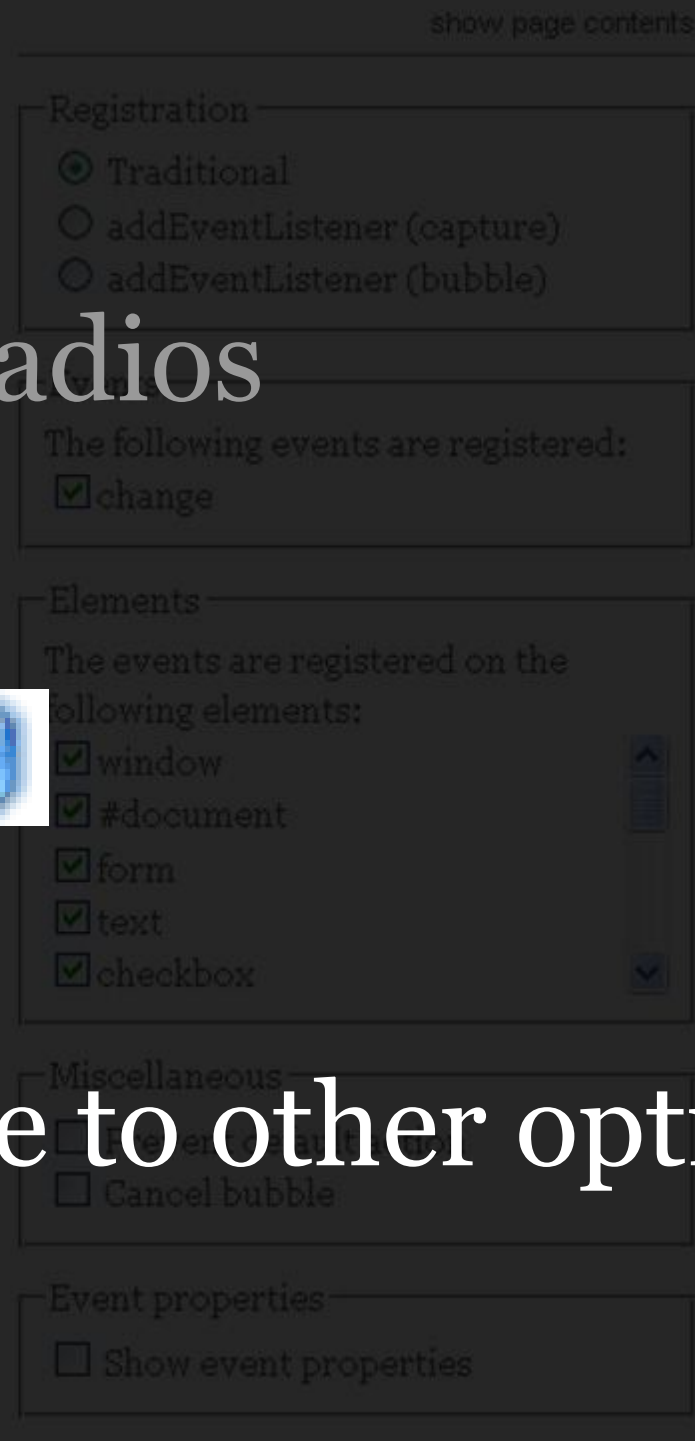


focus



arrow

Arrow keys to move to other option



- text fields
- select boxes
- checkboxes and radios

Keyboard:



focus



arrow



blur

Blur select box.
CHANGE



show page contents

Registration

- Traditional
- addEventListener (capture)
- addEventListener (bubble)

The following events are registered:

- change

Elements

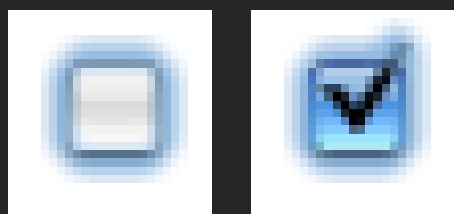
The events are registered on the

- form
- text
- checkbox

Miscellaneous

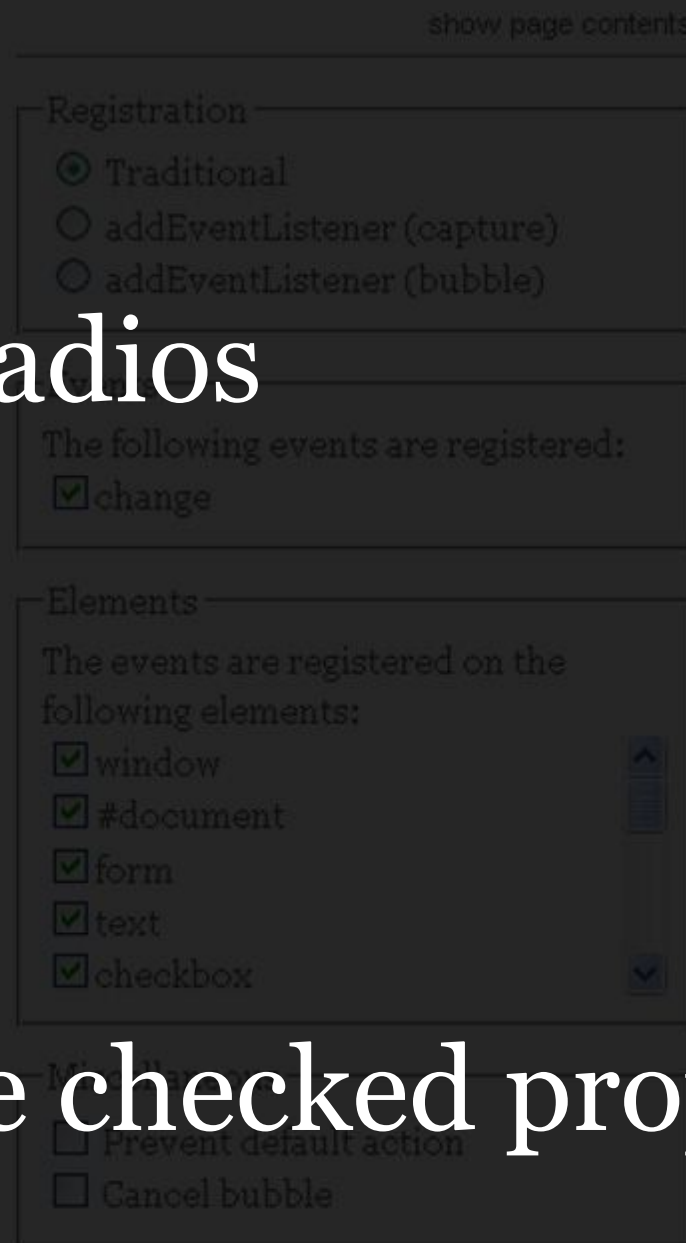
- Prevent default action
- Cancel bubble

- text fields
- select boxes
- checkboxes and radios

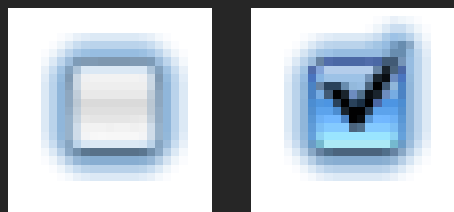


click

CHANGE when the checked property changes.



- text fields
- select boxes
- checkboxes and radios



click

...

show page contents

Registration

- Traditional
- addEventListener (capture)
- addEventListener (bubble)

The following events are registered:

- change

Elements

The events are registered on the following elements:

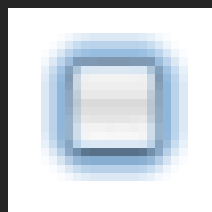
- window
- #document
- form
- text
- checkbox

Miscellaneous

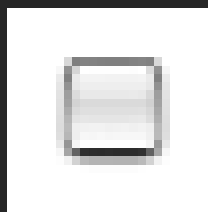
- Prevent default action
- Cancel bubble



- text fields
- select boxes
- checkboxes and radios



click



blur

CHANGE when the element loses the focus.

show page contents

Registration

- Traditional
- addEventListener (capture)
- addEventListener (bubble)

The following events are registered:

- change

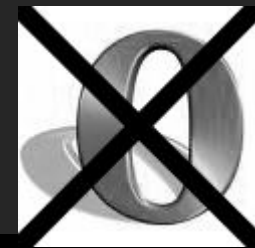
Elements

The events are registered on the following elements:

- window
- #document
- form
- text
- checkbox

Prevent default action

Cancel bubble



- text fields
- select boxes
- checkboxes and radios

This is a
BUG!

CHANGE when the element loses the focus.



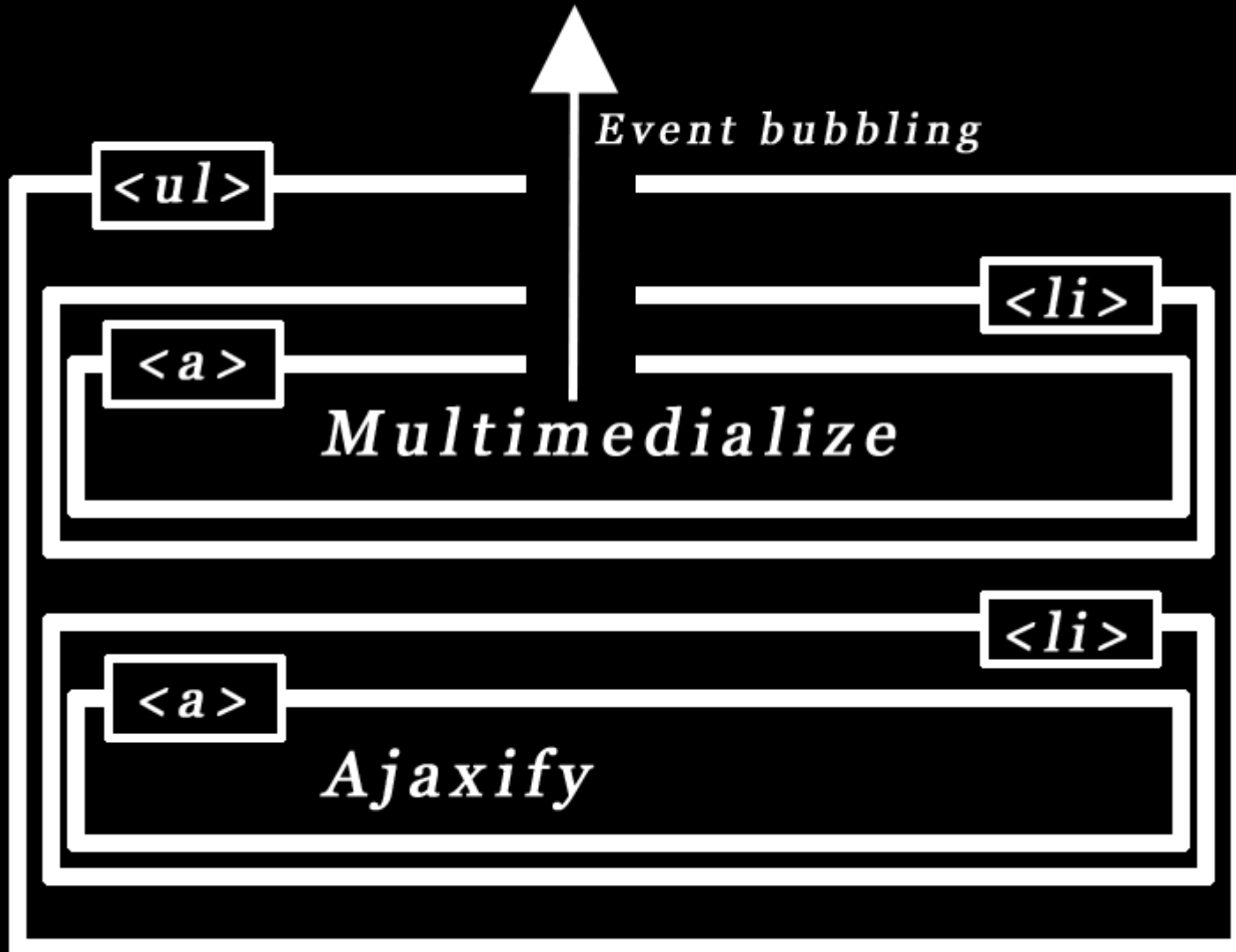


Event delegation

Event delegation

```
<ul>
  <li><a href="#">Multimedialize</a>
    <ul>
      <li><a href="#">Sound</a></li>
      <li><a href="#">Java applets</a></li>
    </ul></li>
  <li><a href="#">Ajaxify</a>
    <ul>
      <li><a href="#">Web 2.0</a></li>
      <li><a href="#">Web 3.0</a></li>
      <li><a href="#">Web 4.0b</a></li>
    </ul></li>
</ul>
```


Event delegation



Event delegation

The event bubbles up to the `` anyway.

```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover = mouseOver;  
    dropdown.onmouseout = mouseOut;  
  }  
}
```

So why not handle it at that level?

Saves a lot of event handlers.

Event delegation

```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover = mouseOver;  
    dropdown.onmouseout = mouseOut;  
  }  
}
```

Works in all browsers.

The logo for the World Wide Web Consortium (W3C), consisting of the letters 'W3C' in a blue, sans-serif font.




Event delegation

But suppose someone doesn't use a mouse at all,

but the keyboard

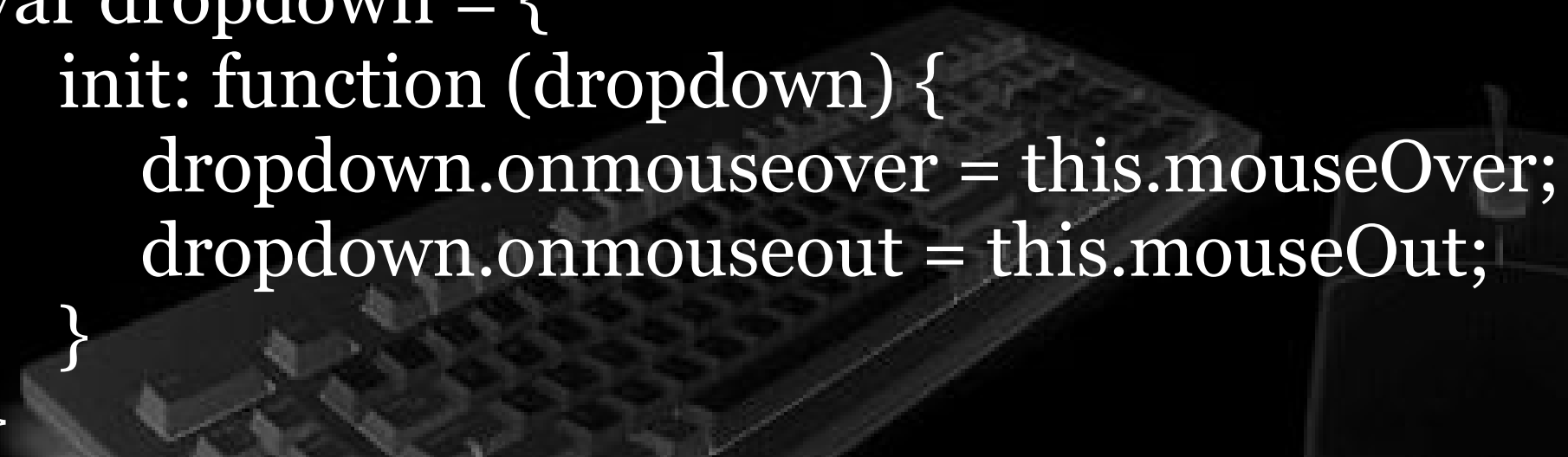
how does the menu fold out?

A computer keyboard and mouse are shown on a black background. The keyboard is on the left, and the mouse is on the right. The text "Device independence" is overlaid in the center in a white serif font.

Device
independence

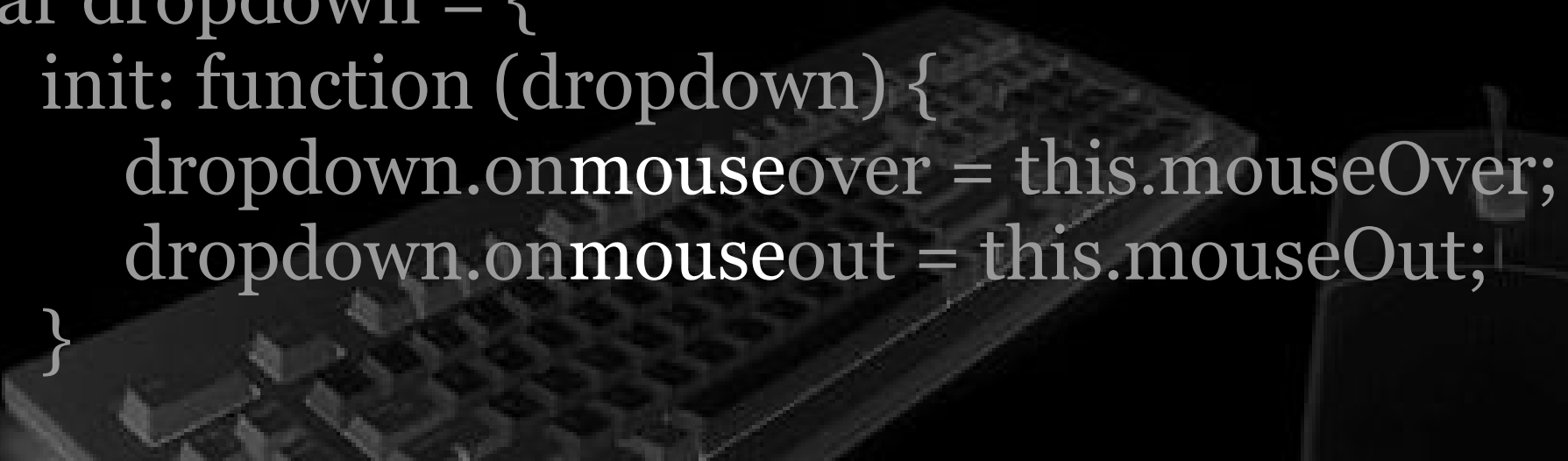
Event delegation

```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover = this.mouseOver;  
    dropdown.onmouseout = this.mouseOut;  
  }  
}
```

A grayscale image of a computer keyboard and a mouse, positioned diagonally across the lower half of the slide. The keyboard is on the left, and the mouse is on the right. The background is dark, making the lighter-colored devices stand out.

Event delegation


```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover = this.mouseOver;  
    dropdown.onmouseout = this.mouseOut;  
  }  
}
```



Doesn't work without a mouse.

Event delegation

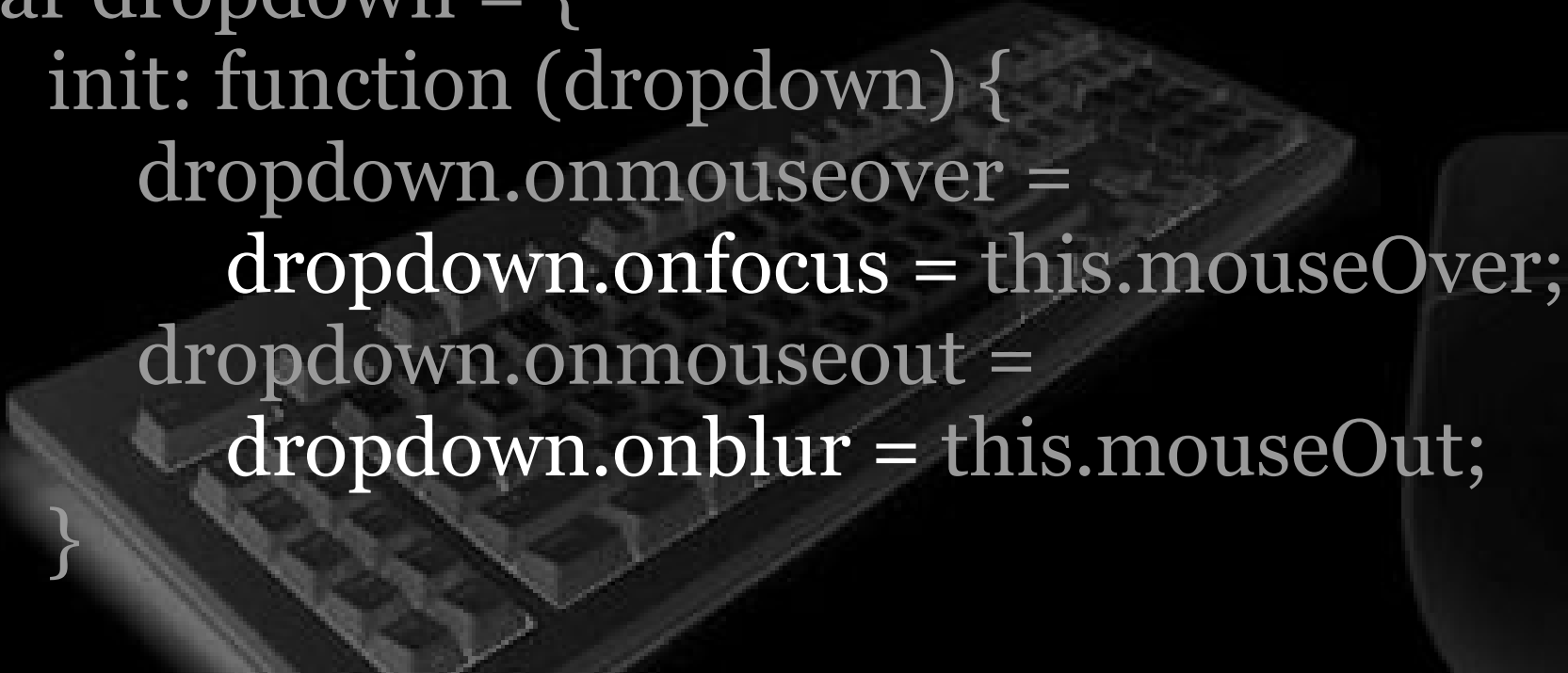
```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover = this.mouseOver;  
    dropdown.onmouseout = this.mouseOut;  
  }  
}
```



We need events that tell us whether the user enters or leaves a link.
focus and blur

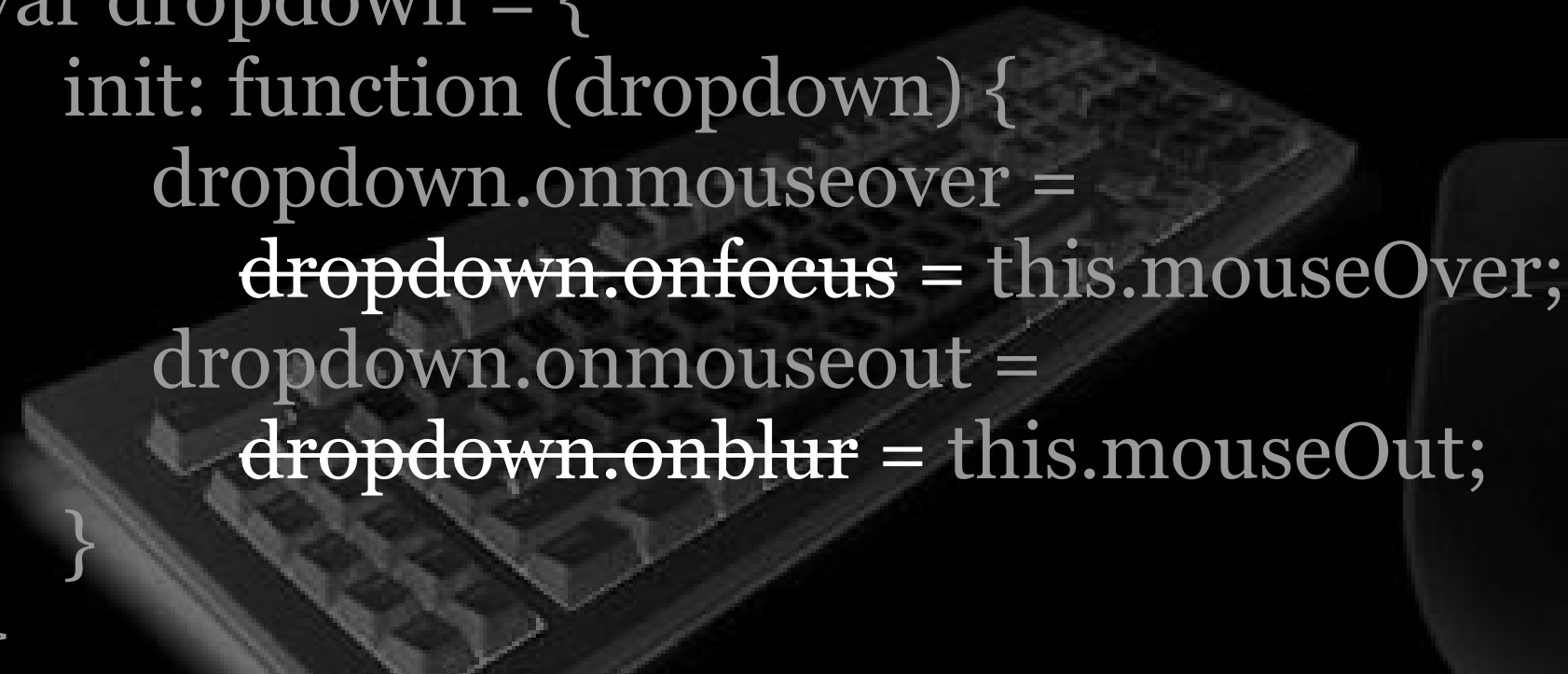
Event delegation

```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover =  
      dropdown.onfocus = this.mouseOver;  
    dropdown.onmouseout =  
      dropdown.onblur = this.mouseOut;  
  }  
}
```



Event delegation

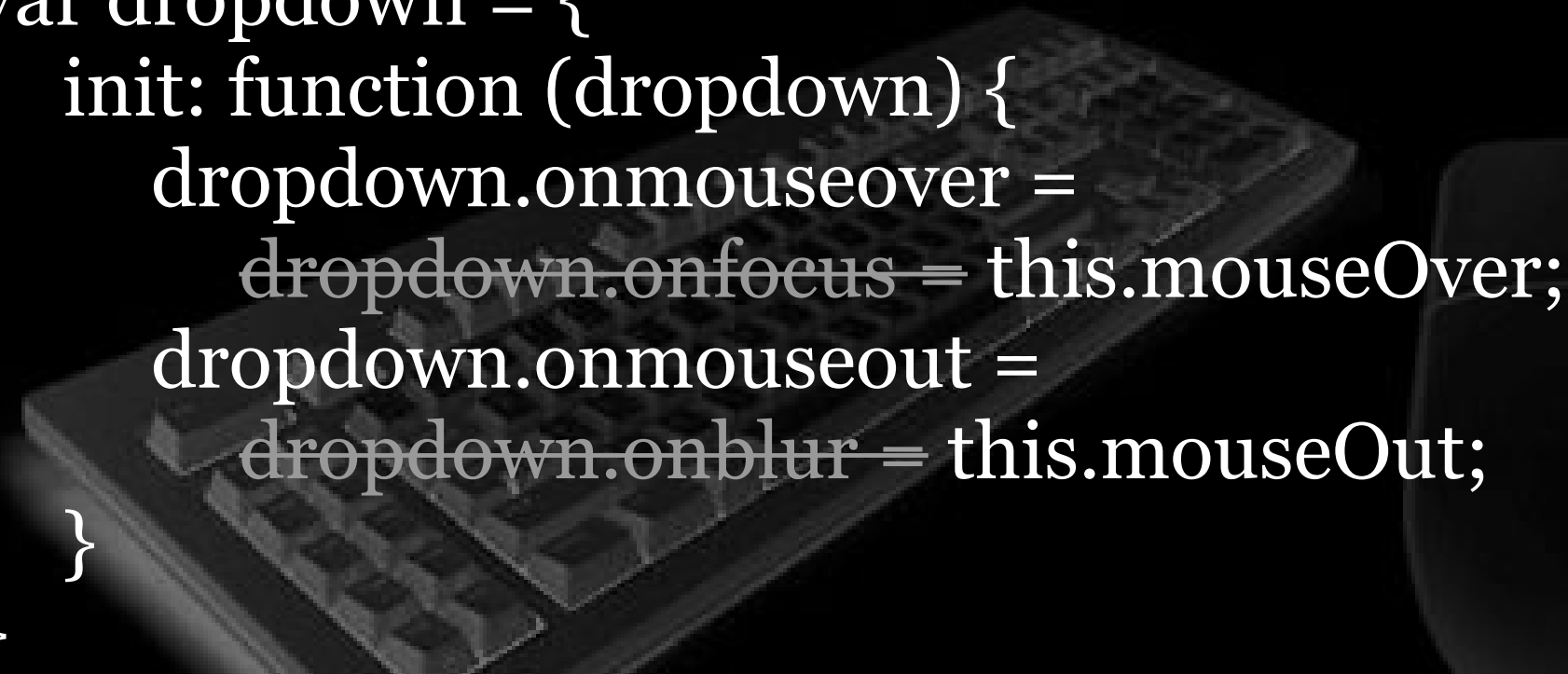
```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover =  
    dropdown.onfocus = this.mouseOver;  
    dropdown.onmouseout =  
    dropdown.onblur = this.mouseOut;  
  }  
}
```



Doesn't work.

Event delegation

```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover =  
      dropdown.onfocus = this.mouseOver;  
    dropdown.onmouseout =  
      dropdown.onblur = this.mouseOut;  
  }  
}
```



Focus and blur don't bubble.

To bubble or not to bubble

Two kinds of events:

- 1) Mouse and key events
- 2) Interface events



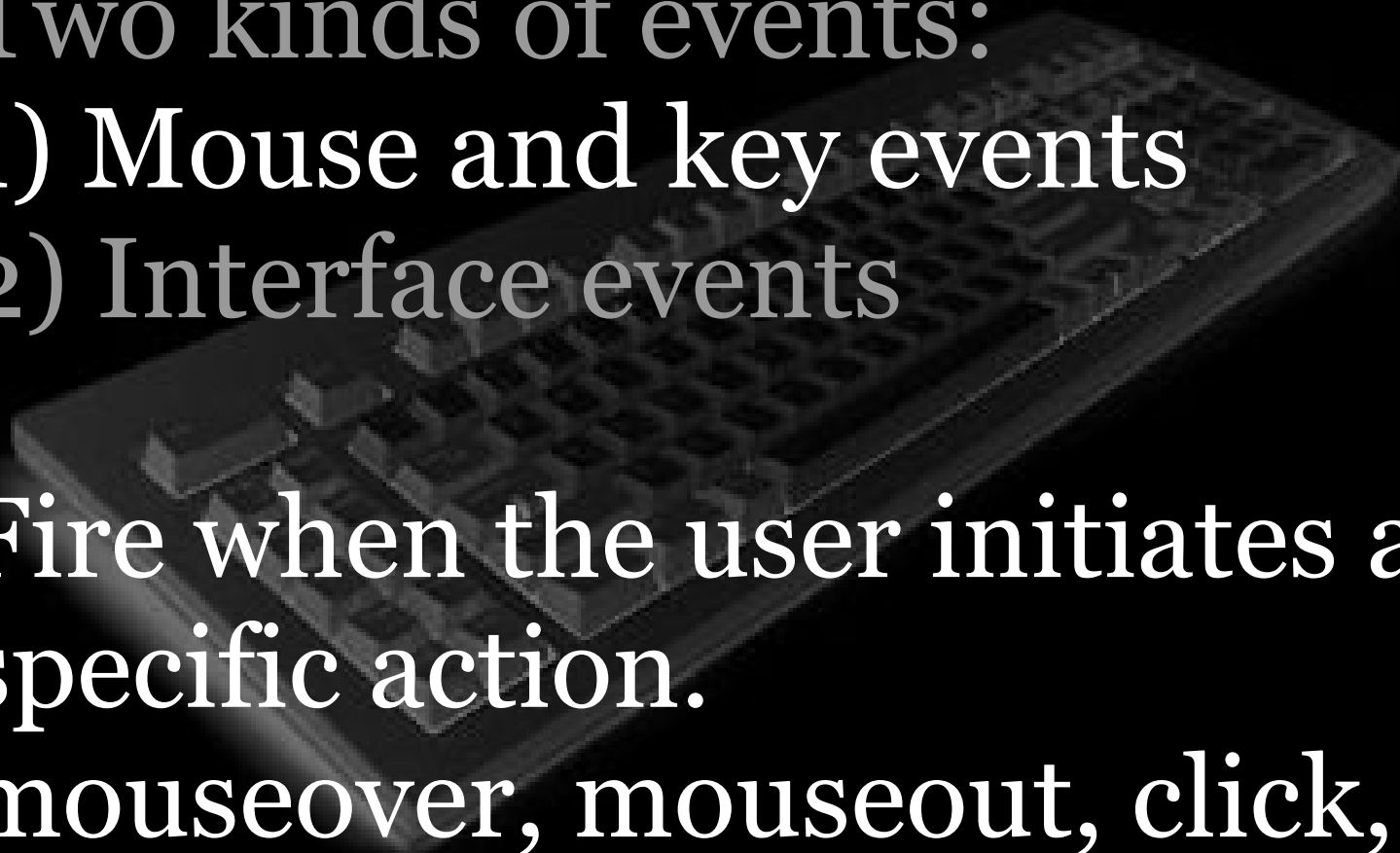
To bubble or not to bubble

Two kinds of events:

- 1) Mouse and key events
- 2) Interface events

Fire when the user initiates a device-specific action.

mouseover, mouseout, click, keydown, keypress

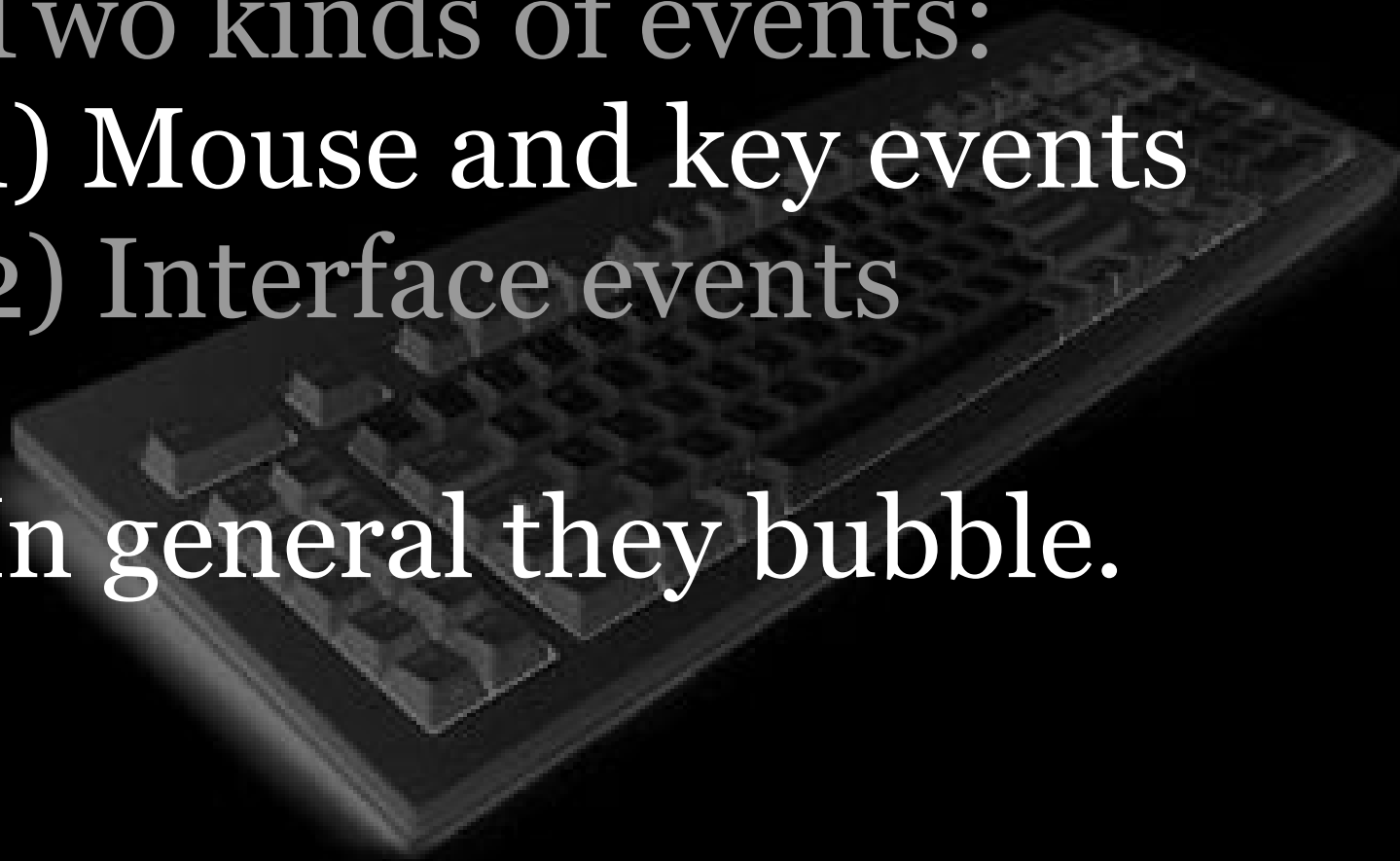


To bubble or not to bubble

Two kinds of events:

- 1) Mouse and key events
- 2) Interface events

In general they bubble.

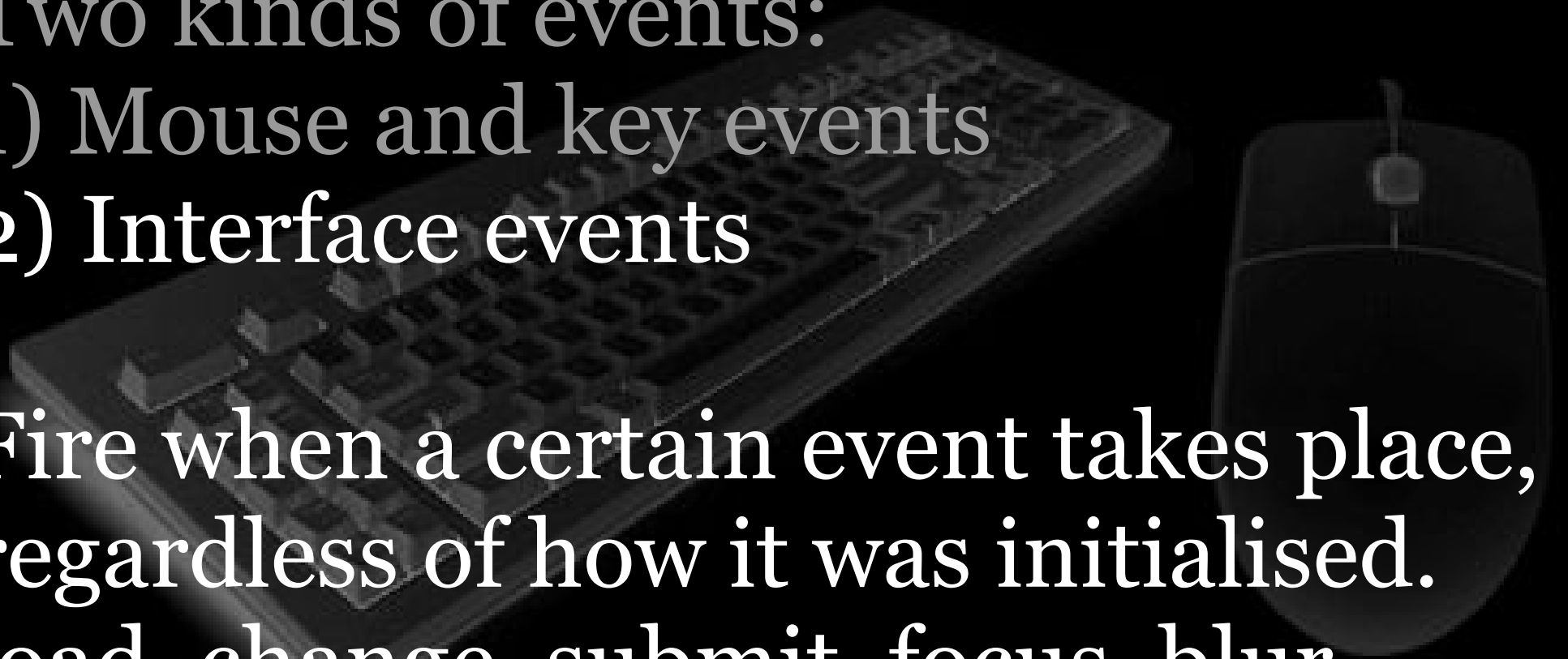


To bubble or not to bubble

Two kinds of events:

- 1) Mouse and key events
- 2) Interface events

Fire when a certain event takes place,
regardless of how it was initialised.
load, change, submit, focus, blur

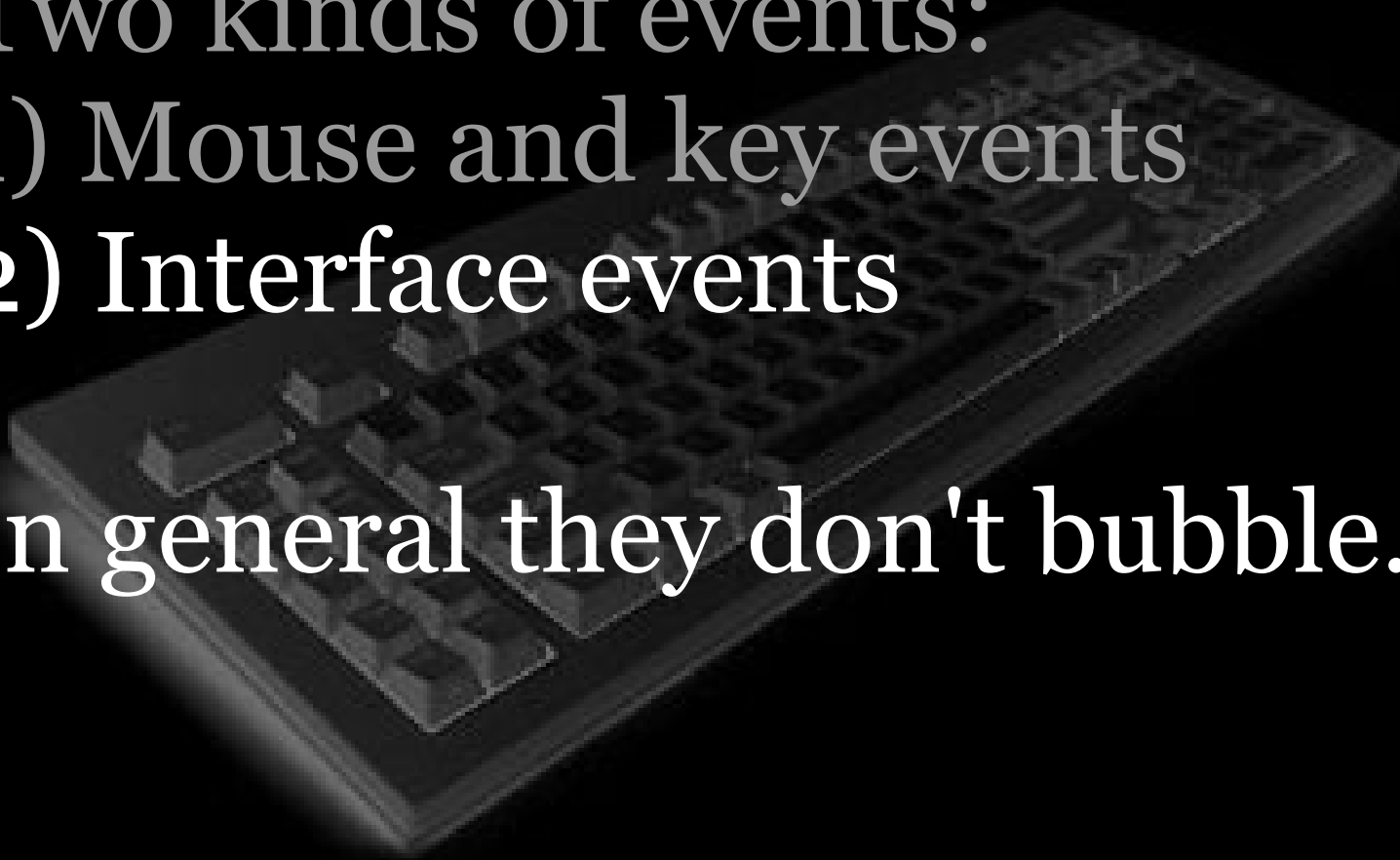


To bubble or not to bubble

Two kinds of events:

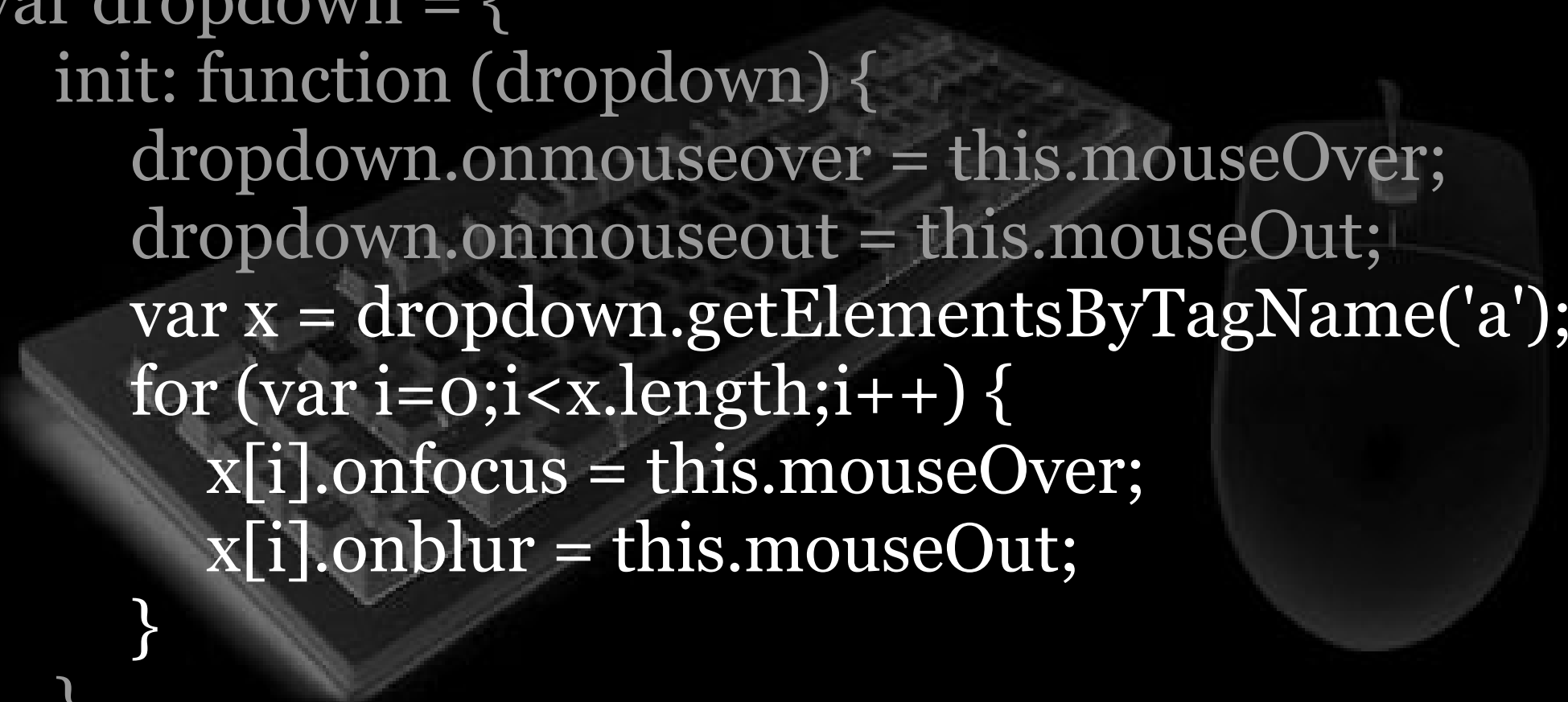
- 1) Mouse and key events
- 2) Interface events

In general they don't bubble.



Event delegation

```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover = this.mouseOver;  
    dropdown.onmouseout = this.mouseOut;  
    var x = dropdown.getElementsByTagName('a');  
    for (var i=0;i<x.length;i++) {  
      x[i].onfocus = this.mouseOver;  
      x[i].onblur = this.mouseOut;  
    }  
  }  
}
```



Event delegation

So we're stuck with setting a focus and blur event on every single link.

Or are we ... ?



Event delegation

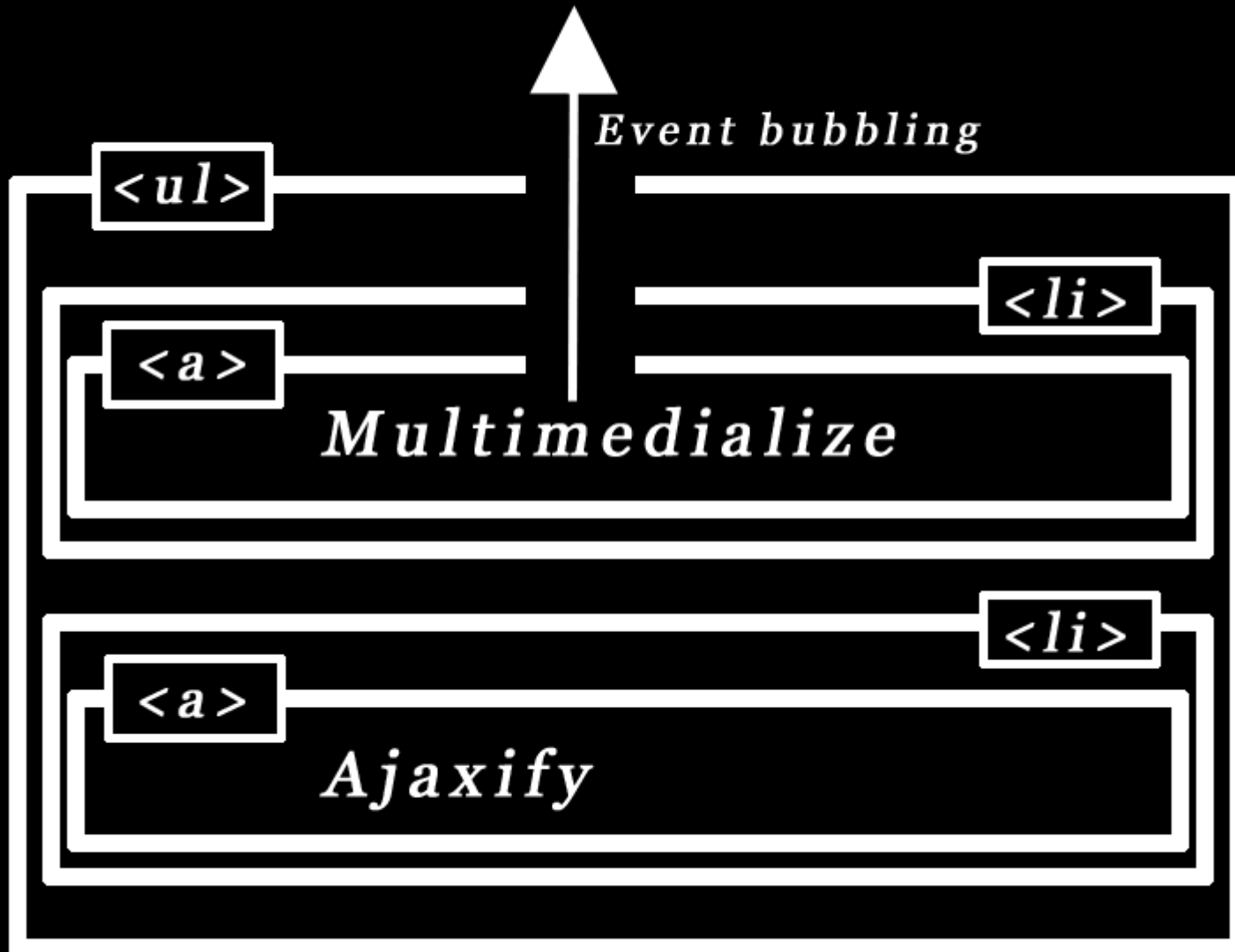
Event capturing to the rescue.

Event capturing is the opposite of event bubbling, and it is supported in all W3C-compliant browsers.



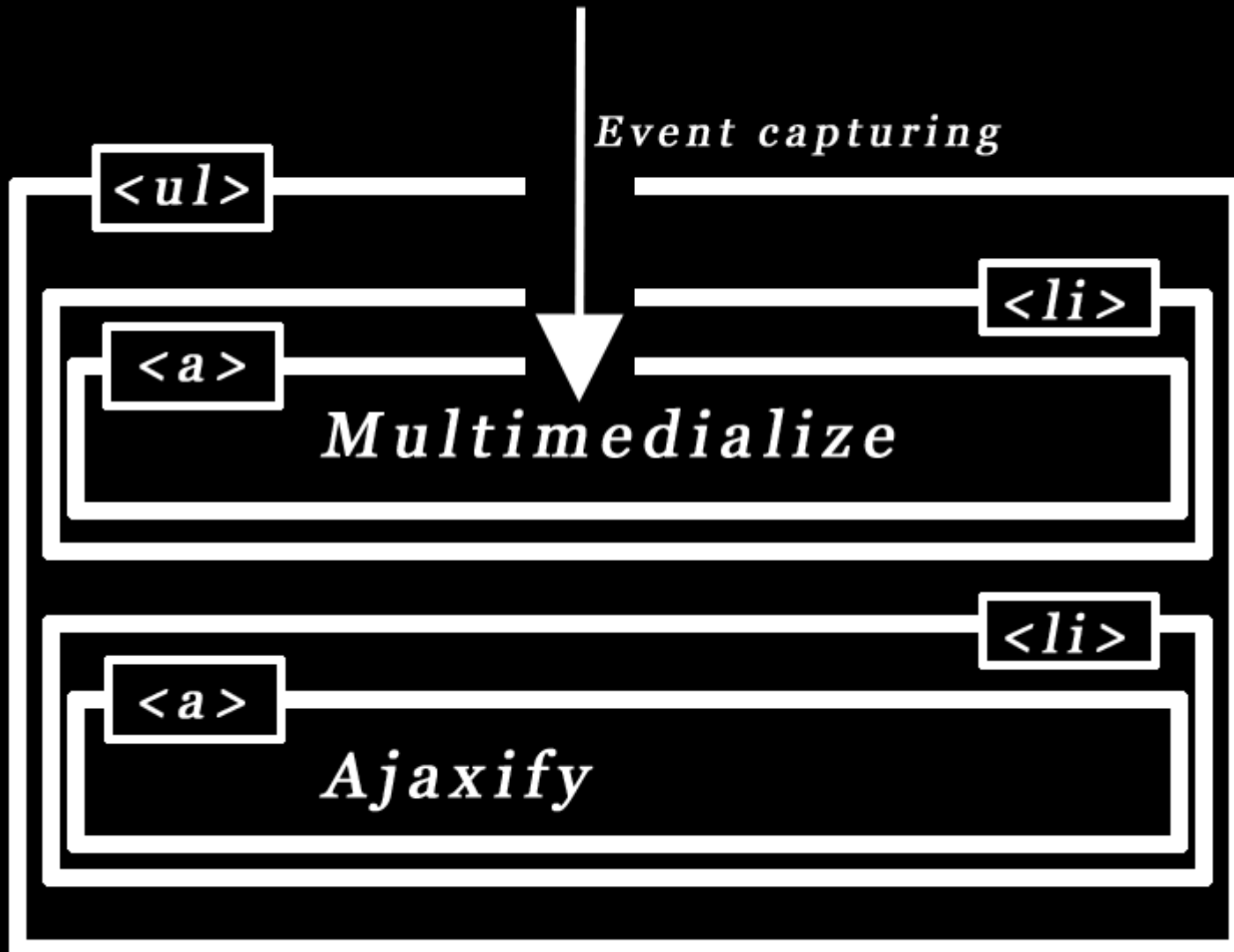
Event bubbling

```
addEventListener('click',fn,false)
```



Event capturing

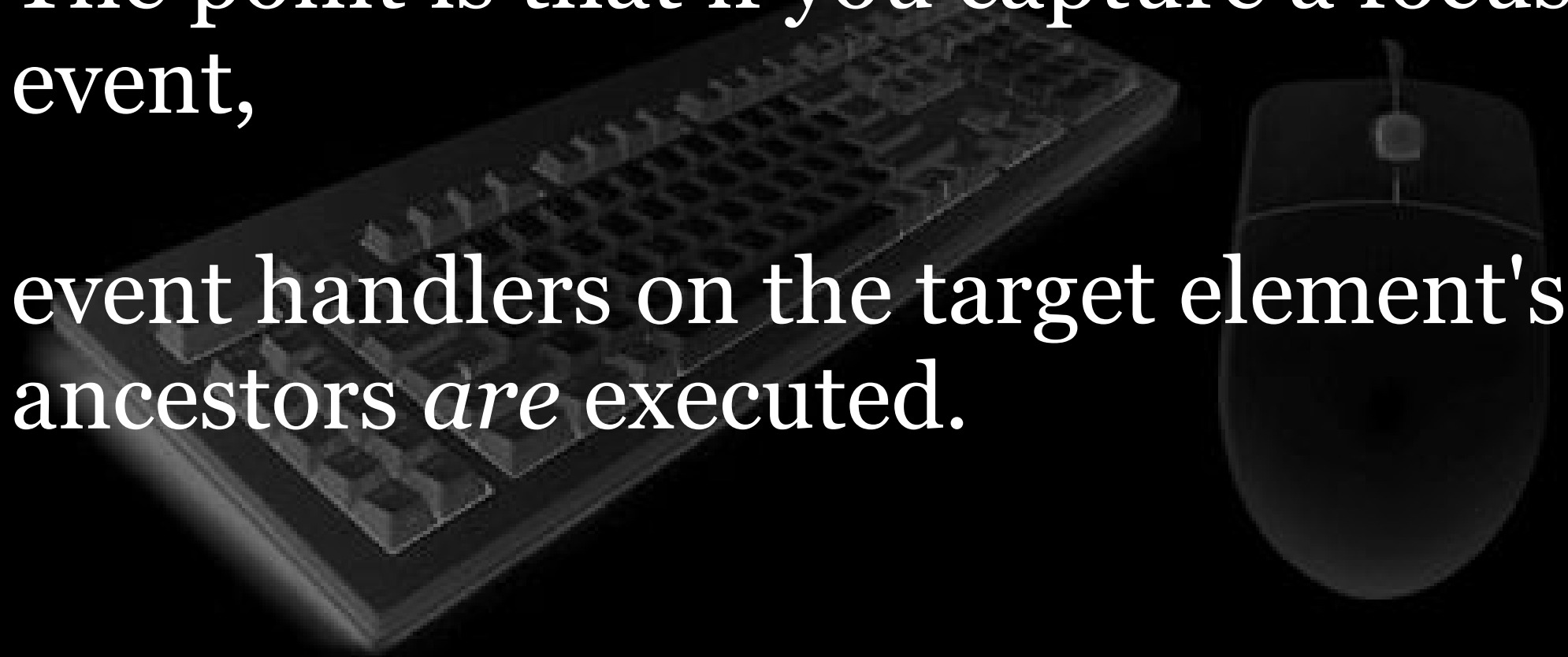
```
addEventListener('click',fn,true)
```



Event delegation

The point is that if you capture a focus event,

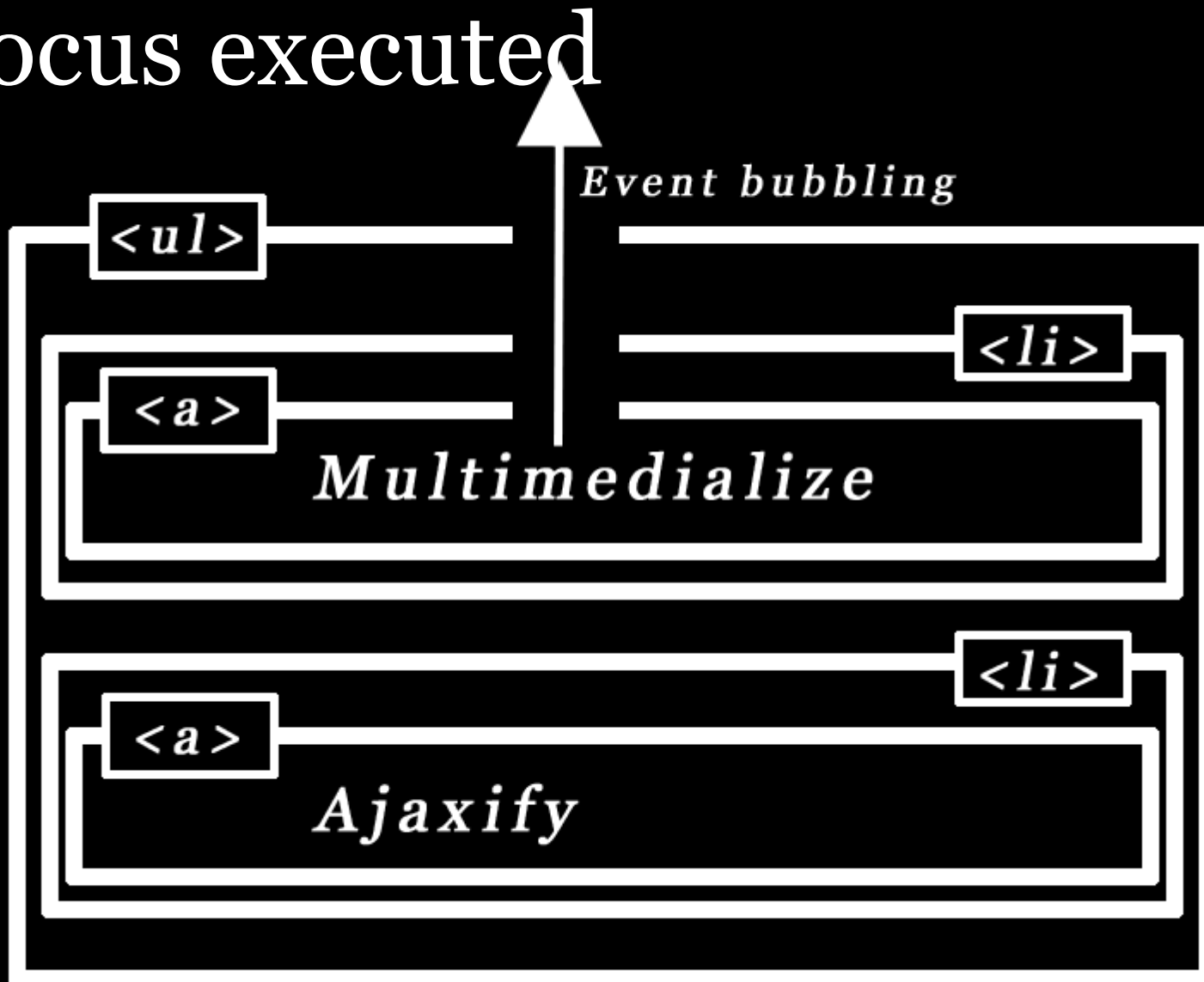
event handlers on the target element's ancestors *are* executed.



Event bubbling

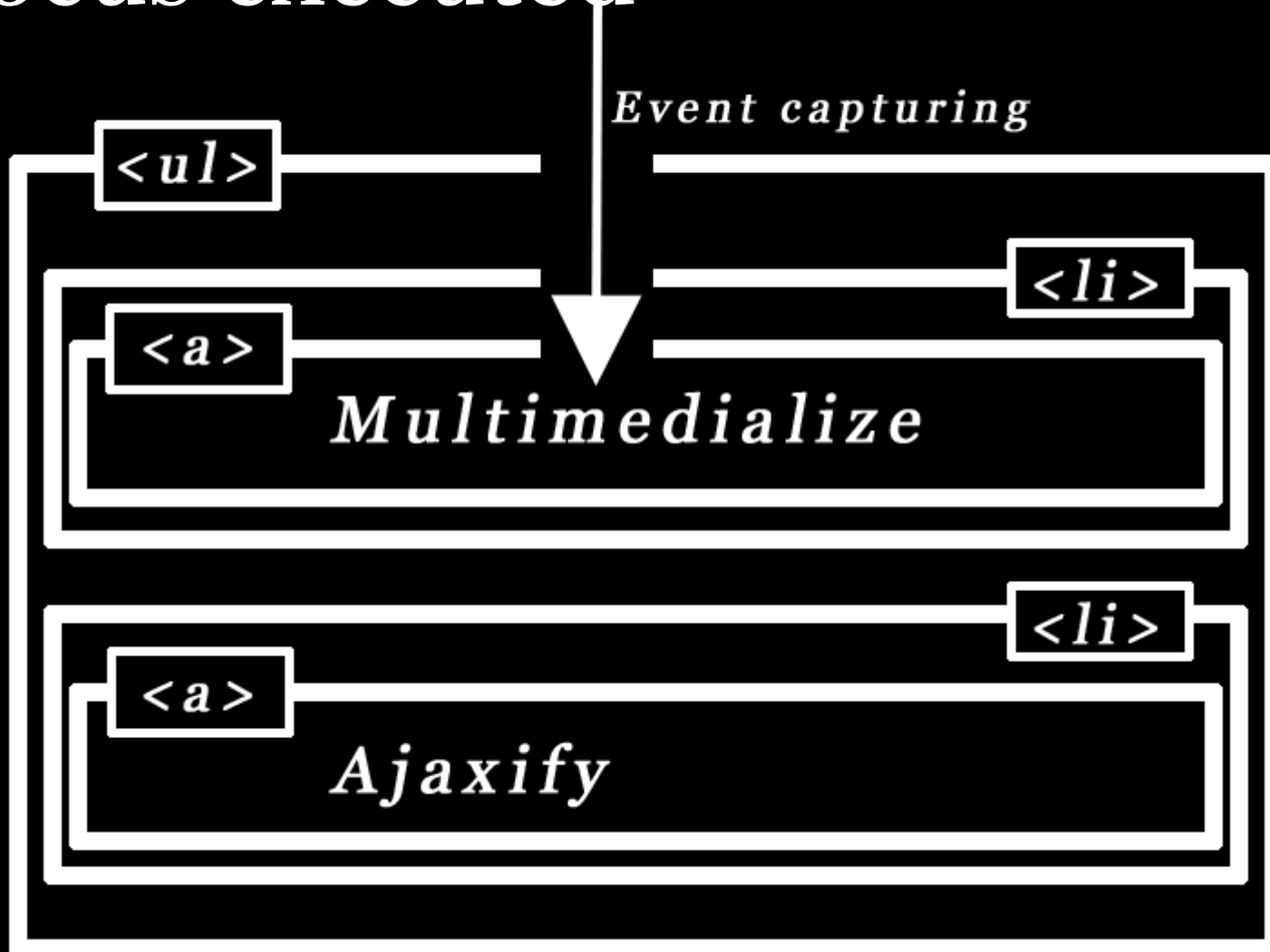
Focus on a:

a.onfocus executed



Event capturing

Focus on a: ul.onfocus, li.onfocus and a.onfocus executed



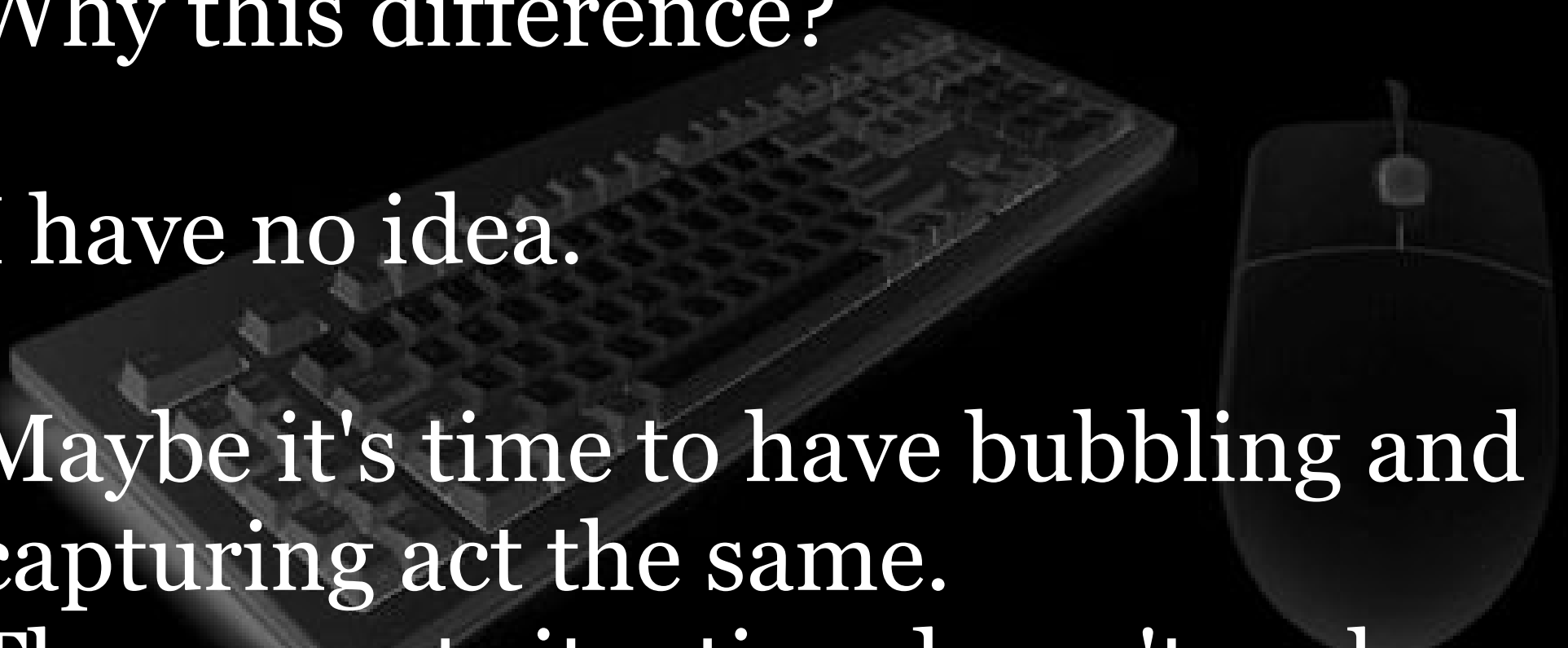
Event capturing and bubbling

Why this difference?

I have no idea.

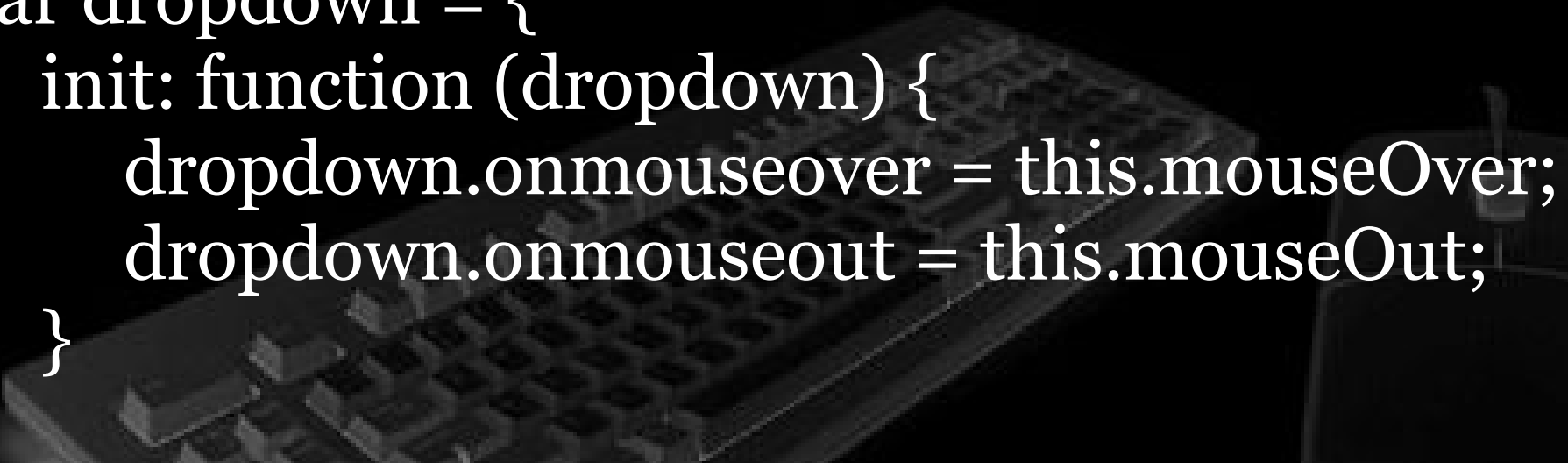
Maybe it's time to have bubbling and capturing act the same.

The current situation doesn't make sense (though it's useful).



Event delegation

```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover = this.mouseOver;  
    dropdown.onmouseout = this.mouseOut;  
  }  
}
```

A grayscale image of a computer keyboard and a mouse, positioned diagonally across the lower half of the slide. The keyboard is on the left, and the mouse is on the right. The background is dark, making the lighter-colored devices stand out.

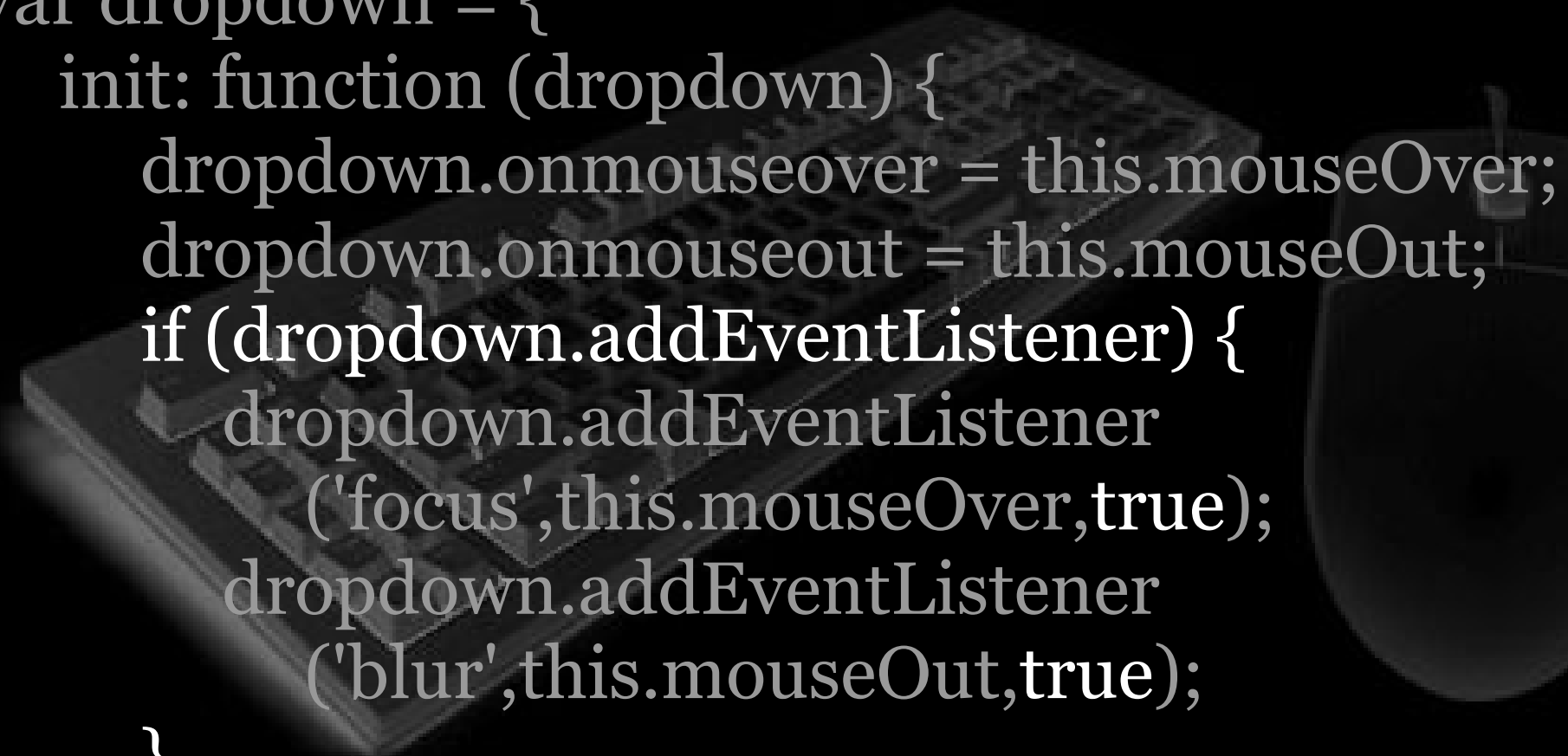
Event delegation

```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover = this.mouseOver;  
    dropdown.onmouseout = this.mouseOut;  
    dropdown.addEventListener  
      ('focus',this.mouseOver,true);  
    dropdown.addEventListener  
      ('blur',this.mouseOut,true);  
  }  
}
```



Event delegation

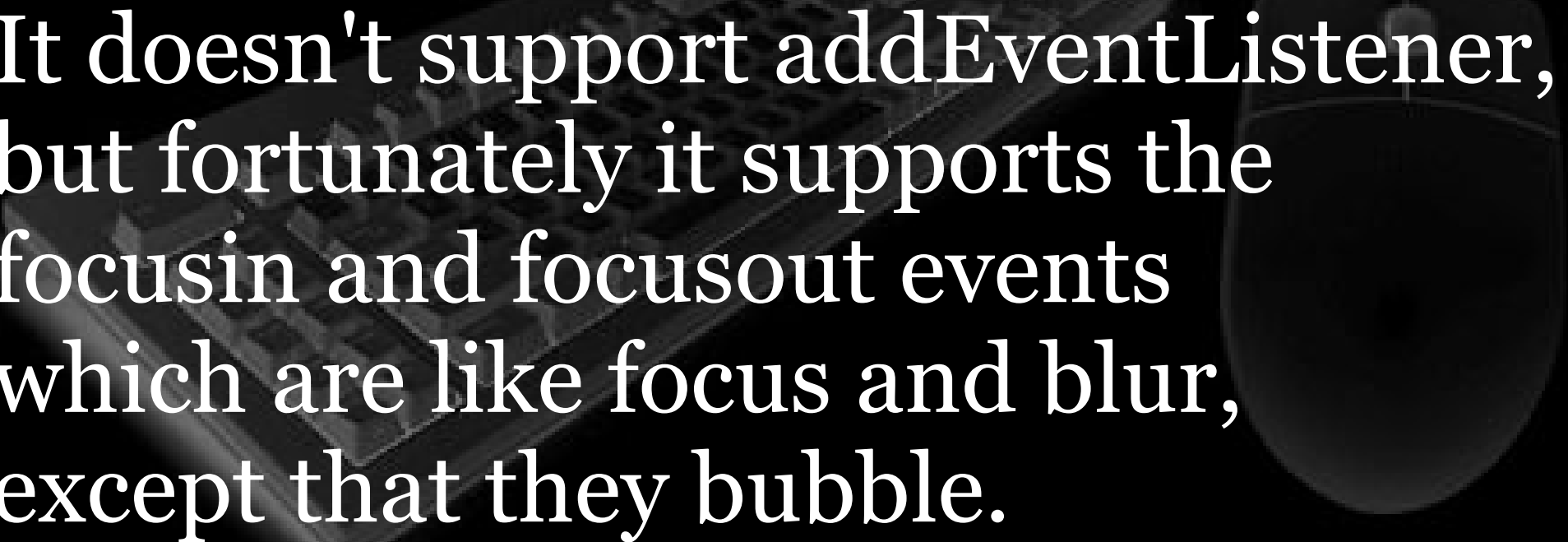
```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover = this.mouseOver;  
    dropdown.onmouseout = this.mouseOut;  
    if (dropdown.addEventListener) {  
      dropdown.addEventListener  
        ('focus',this.mouseOver,true);  
      dropdown.addEventListener  
        ('blur',this.mouseOut,true);  
    }  
  }  
}
```



Event delegation

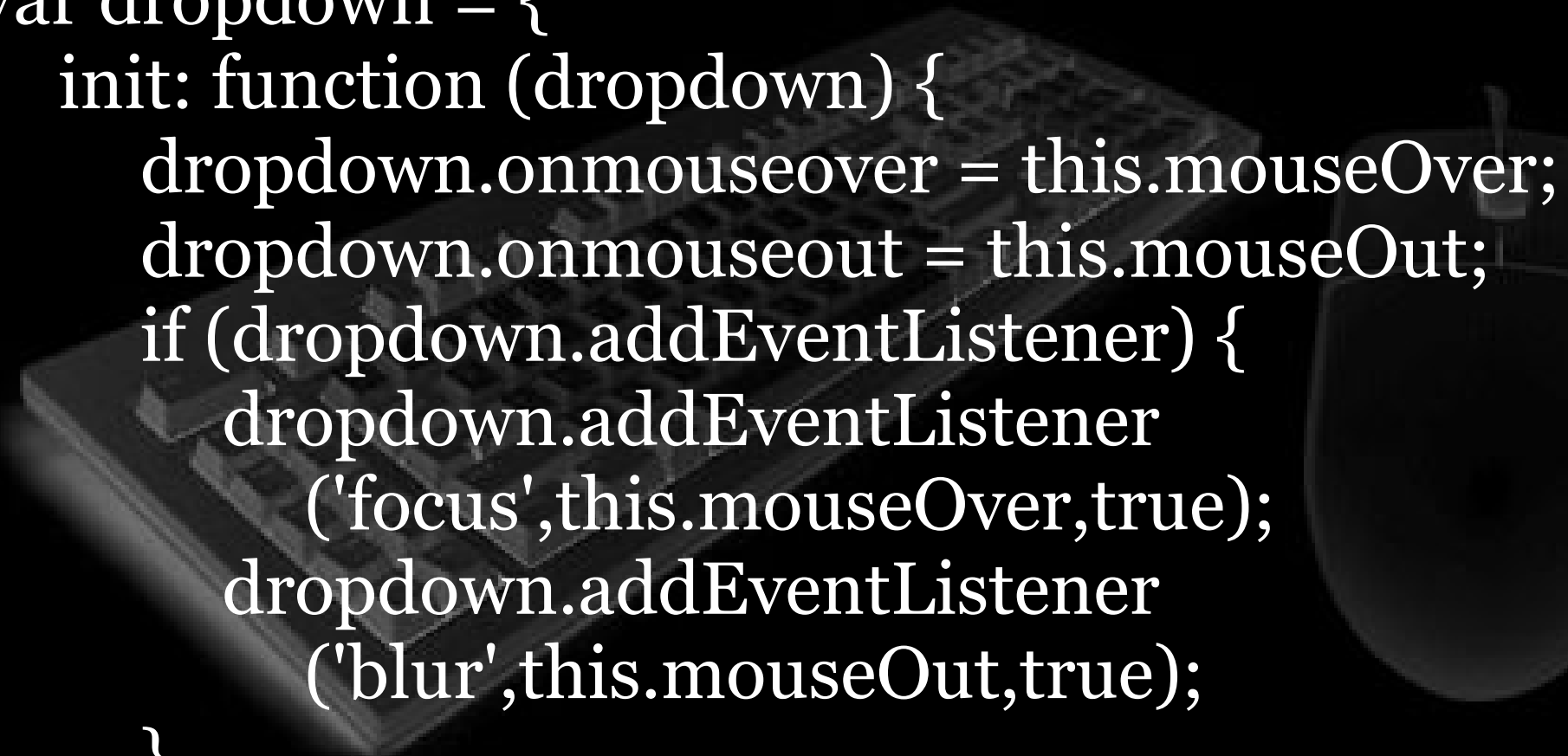
And what about IE?

It doesn't support `addEventListener`, but fortunately it supports the `focusin` and `focusout` events which are like `focus` and `blur`, except that they bubble.

A faint, semi-transparent image of a computer keyboard and mouse is visible in the background. The keyboard is on the left and the mouse is on the right, both appearing as light gray shapes against the dark background.

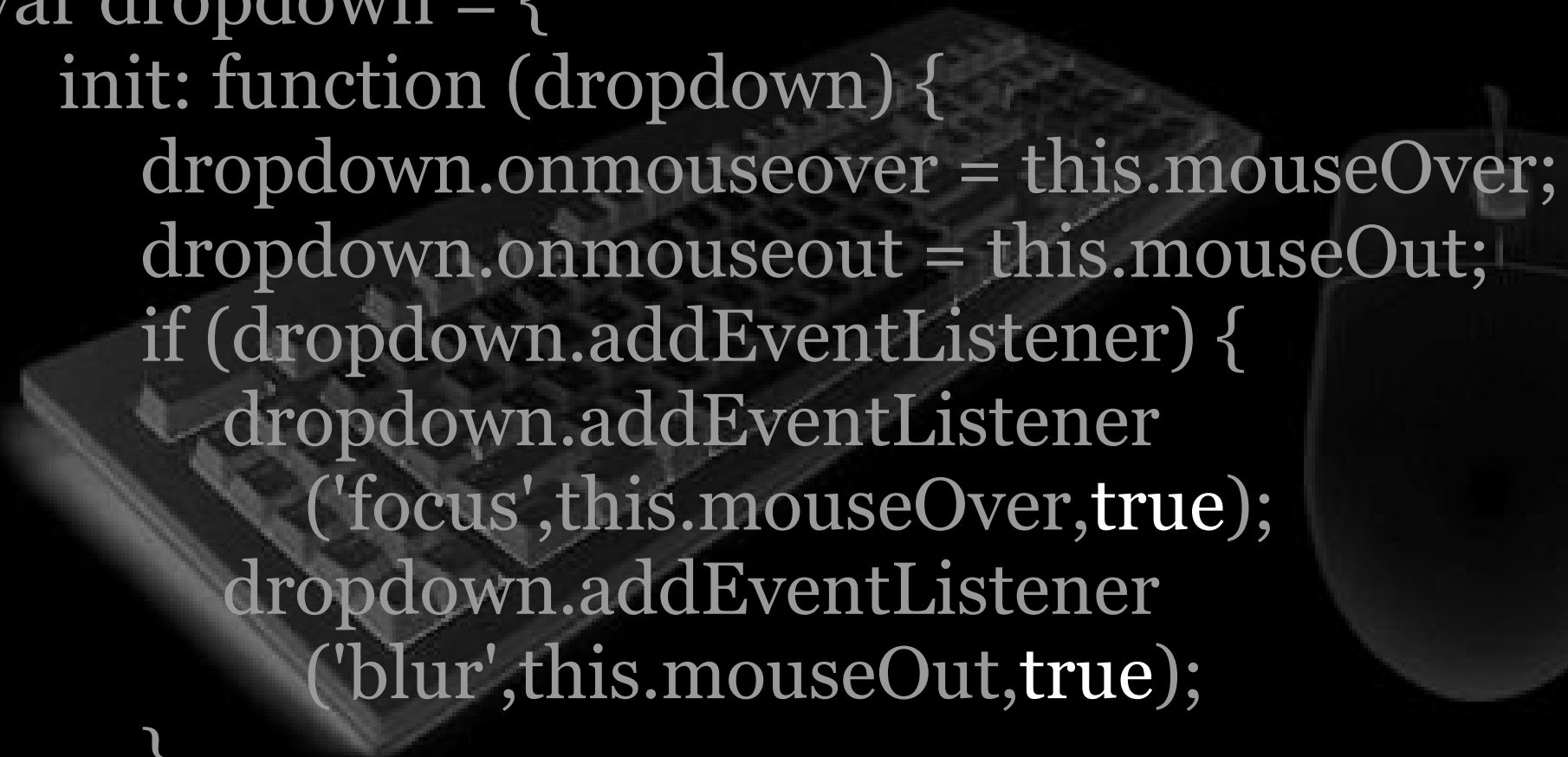
Event delegation

```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover = this.mouseOver;  
    dropdown.onmouseout = this.mouseOut;  
    if (dropdown.addEventListener) {  
      dropdown.addEventListener  
        ('focus',this.mouseOver,true);  
      dropdown.addEventListener  
        ('blur',this.mouseOut,true);  
    }  
  }  
}
```



Event delegation


```
var dropdown = {  
  init: function (dropdown) {  
    dropdown.onmouseover = this.mouseOver;  
    dropdown.onmouseout = this.mouseOut;  
    if (dropdown.addEventListener) {  
      dropdown.addEventListener  
        ('focus',this.mouseOver,true);  
      dropdown.addEventListener  
        ('blur',this.mouseOut,true);  
    }  
    dropdown.onfocusin = this.mouseOver;  
    dropdown.onfocusout = this.mouseOut  
  }  
}
```



Event delegation

```
dropdown.onmouseover = this.mouseOver;  
dropdown.onmouseout = this.mouseOut;  
if (dropdown.addEventListener) {  
  dropdown.addEventListener  
    ('focus',this.mouseOver,true);  
  dropdown.addEventListener  
    ('blur',this.mouseOut,true);  
}  
dropdown.onfocusin = this.mouseOver;  
dropdown.onfocusout = this.mouseOut;
```



A computer keyboard and mouse are shown on a black background. The keyboard is on the left, and the mouse is on the right. The text "Device independence" is overlaid in the center in a white serif font.

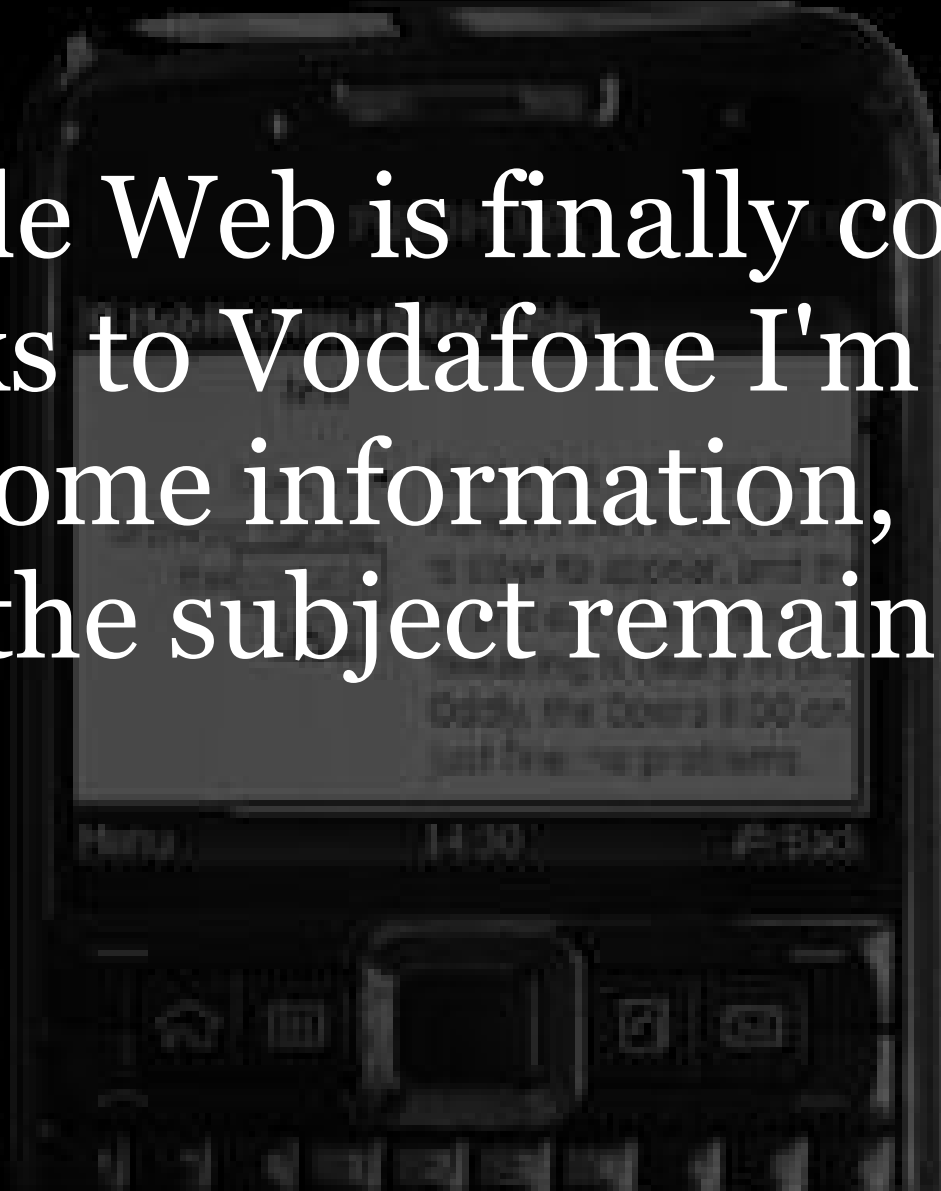
Device independence



Mobile phones



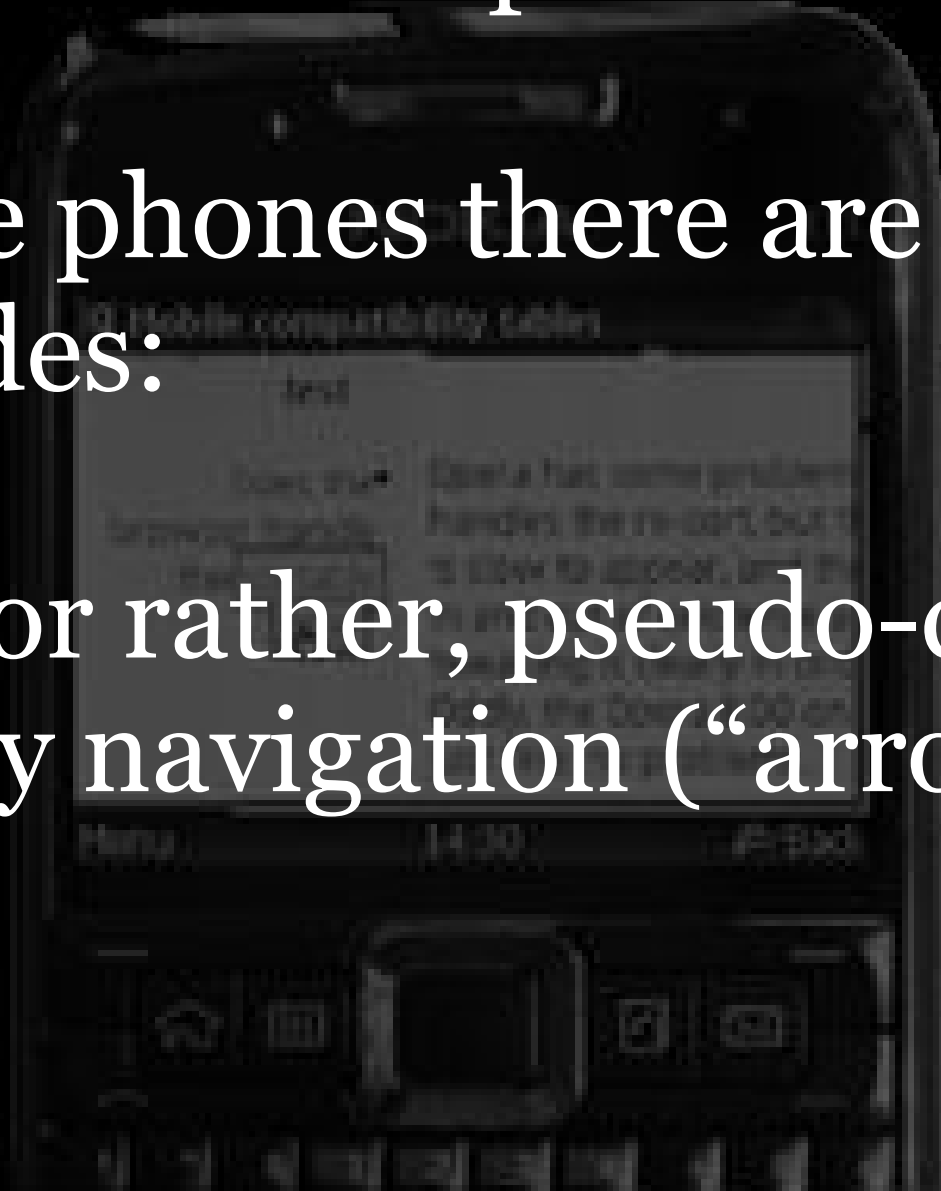
The Mobile Web is finally coming, and thanks to Vodafone I'm able to give you some information, although the subject remains a tricky one.

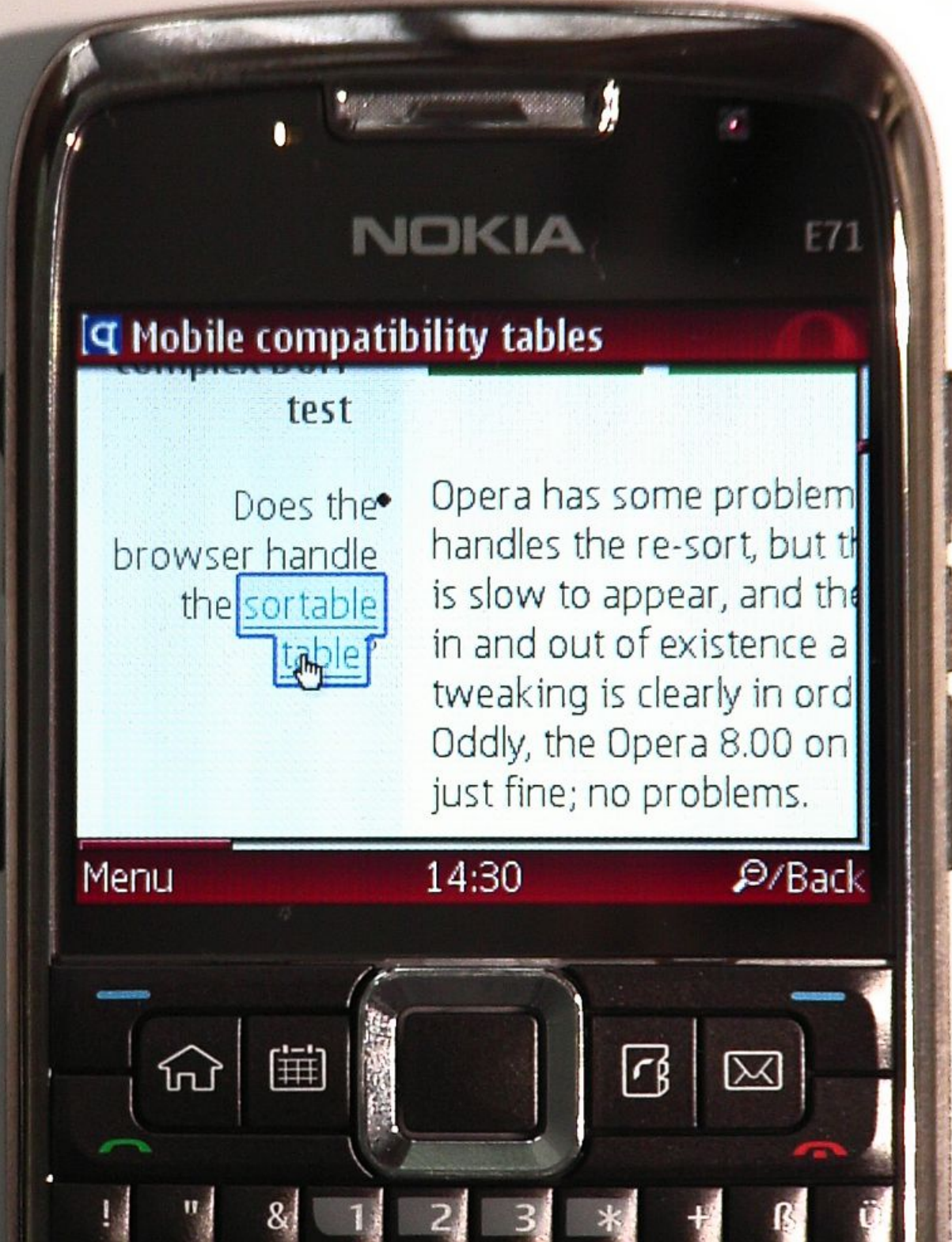


Mobile phones – input modes

On mobile phones there are three input modes:

- Touch
- Cursor (or rather, pseudo-cursor)
- Four-way navigation (“arrow” keys)





Opera Mini 4.2
on Nokia E71

Pseudo-
cursor input
mode

MOTOROLA

Mobile compatibility ta...

Medium-complex DOM test

Does the browser handle the [Edit Text](#) script?

Medium-complex DOM test

Does the browser handle the [getElementsByTagName](#) script?

yes

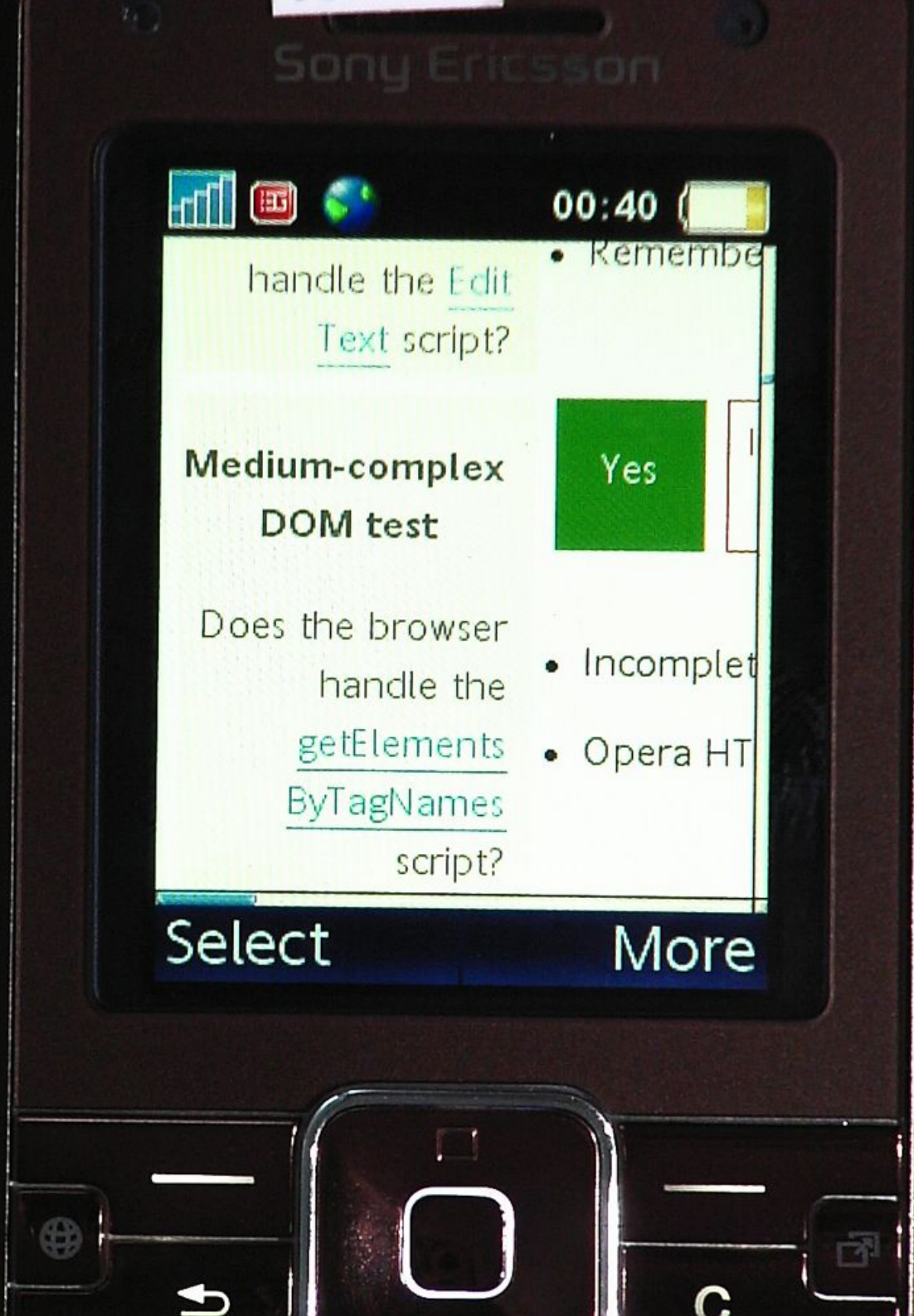
yes

incorrect

Options Back

Opera Mobile
8.00 on
Motorola V3xx

Four-way
navigation



NetFront on Sony Ericsson K770i

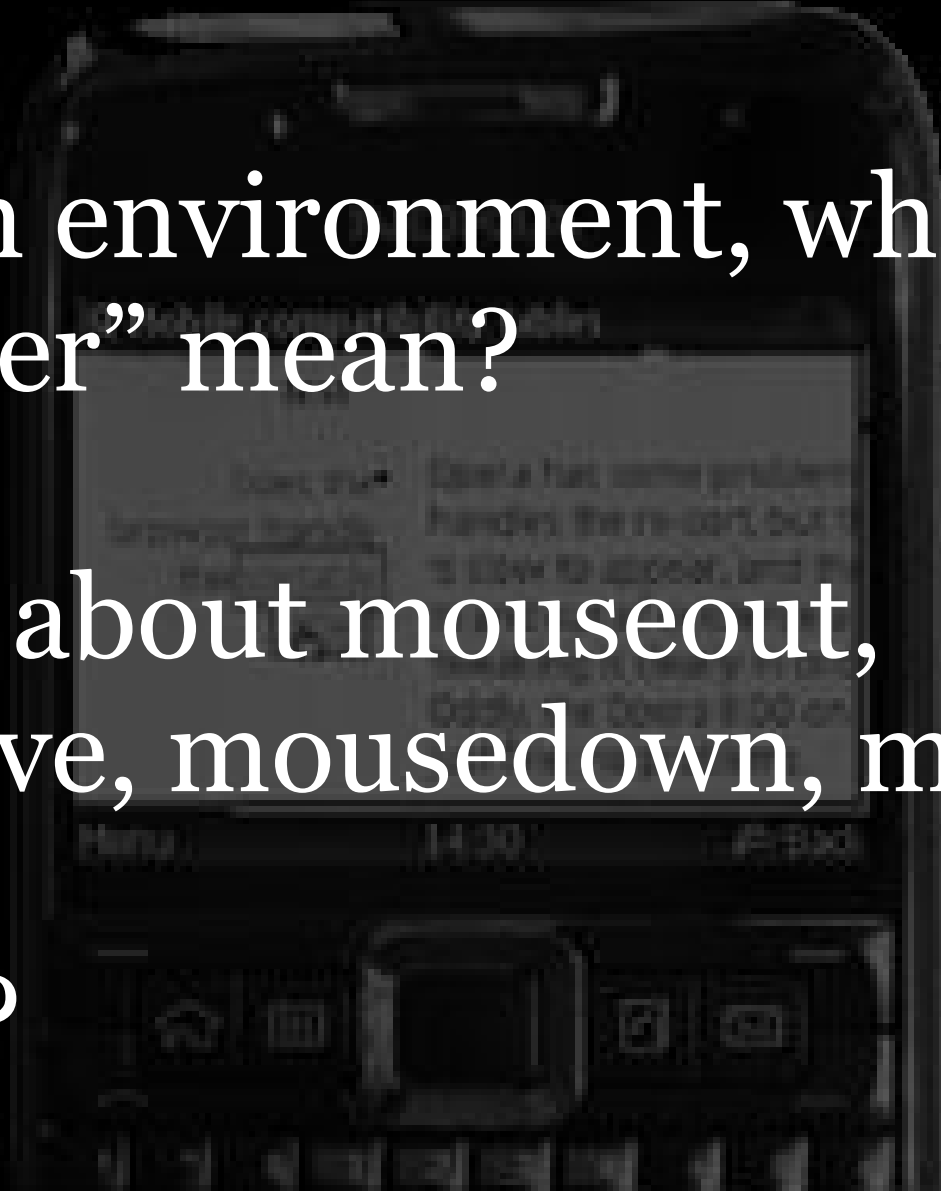
Four-way navigation, but which link do you follow when you click?

Mobile phones – events

In such an environment, what does “mouseover” mean?

And what about mouseout, mousemove, mousedown, mouseup?

And click?

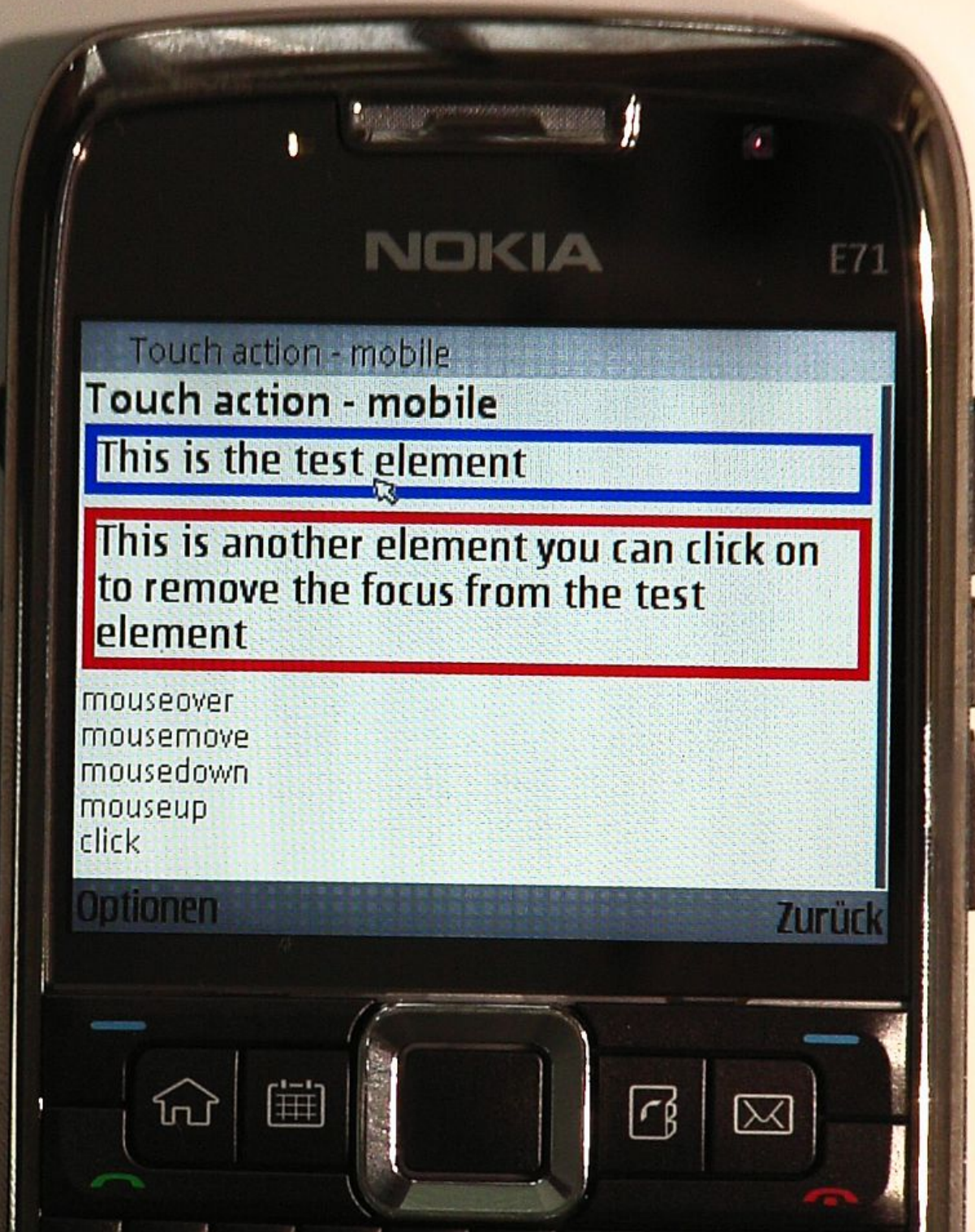


Mobile phones – events

I set up a test in which I “click” on a `<div>` element and see which events take place.

First some good news:





S60 WebKit on
Nokia E71
Input: cursor

The same as
desktop
browsers

Touch action - mobile

This is the test element

This is another element you can click on to remove the focus from the test element

- mousemove
- mouseover
- mousedown
- mouseup
- click



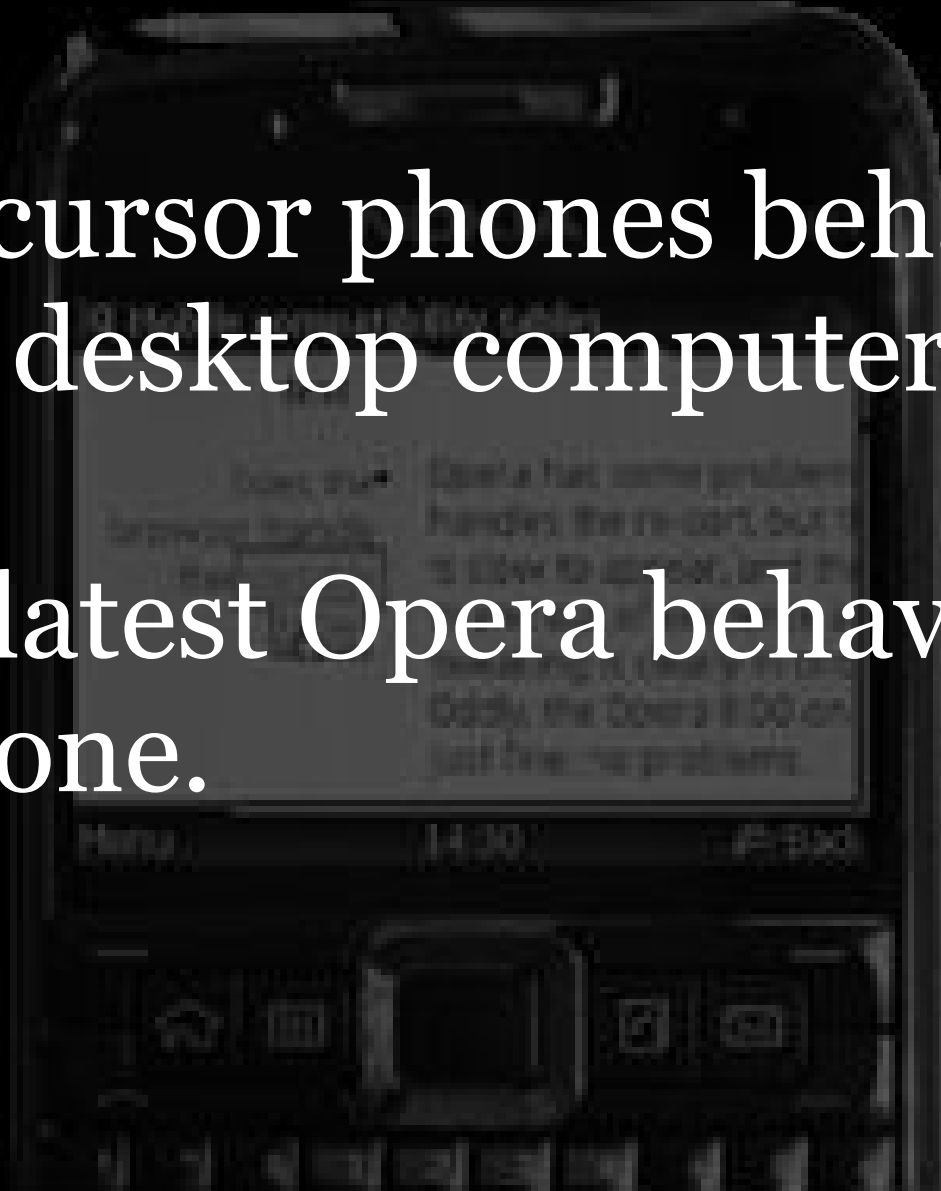
Opera Mobile
9.5 on HTC
Touch Diamond
Input: touch

This is the
same as the
iPhone does.

Mobile phones – events

So Nokia cursor phones behave exactly as desktop computers,

while the latest Opera behaves exactly as the iPhone.



Mobile phones – events

iPhone/Opera model:

As soon as the user touches an element
mouseover, mousemove, mousedown,
mouseup, and click fire,
and the :hover styles are applied.

When user touches another element, mouseout
fires and :hover styles are removed

Mobile phones – events

Now for some bad news.



Touch action - m

This is the test element

This is another element you
on to remove the focus fro
element

mousedown
mouseup
click

Blackberry
Storm

Input: touch

No
mouseover,
-out, or
-move

Touch action - mobile

This is the test element

This is another element you
can click on to remove the
focus from the test element

mousedown

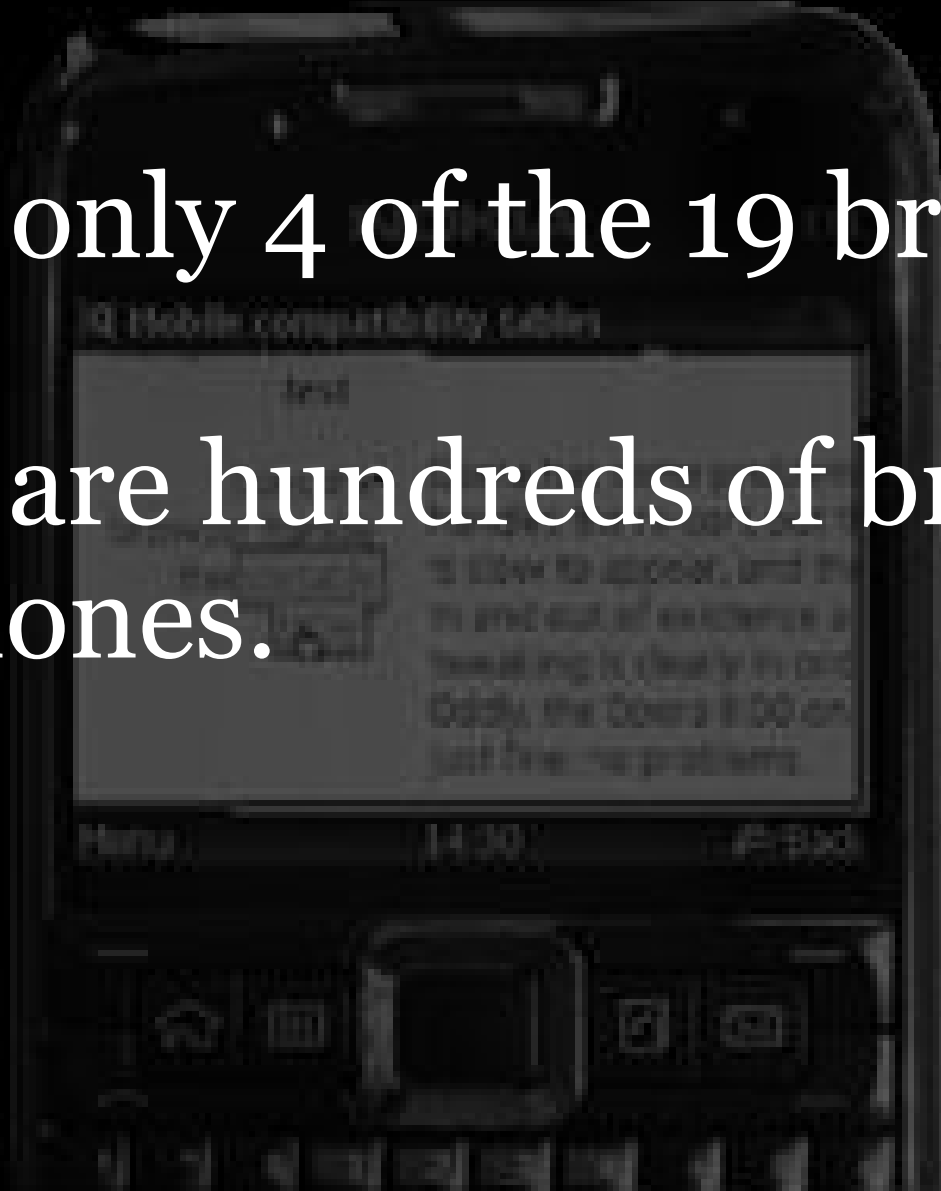
mouseup

NetFront on
Samsung F700
Input: touch

Where's the
click?

Mobile phones – events

These are only 4 of the 19 browsers I tested,
and there are hundreds of browsers on
mobile phones.



Mobile phones – events

General rules for events on mobile phones 1/3:

- use click (and let's politely but firmly give the finger to browsers that do not support it)
- forget about the mouse events

In fact, I think the time has come to retire the mouse events on all devices (with a few exceptions)

Mobile phones – events

General rules for events on mobile phones 2/3:

- use the `resize` AND the `orientationchange` event

`orientationchange` is supported only by iPhone and Blackberry

`resize` is supported by Opera and all WebKits
NetFront doesn't support either – too bad

Mobile phones – events

General rules for events on mobile phones 3/3:

- use key events only for setting general access keys; and NOT for reading out user input in a form field

You should read out the field's value instead.

Event compatibility for desktop:

When an element loses the focus.

- Firefox 2 fires too many events in a variety of circumstances.
- Firefox 3 fires too many events when blurring the window.

<http://quirksmode.org/dom/events>

Mobile compatibility (work in progress)

<http://quirksmode.org/m/>

contextmenu

When the user right-clicks to get the context menu.

Preventing the default (i.e. preventing the context menu from appearing) is the whole point of this event.

Event	IE 5.5	IE 6	IE 7	IE8b1	FF 2	FF 3b5	Saf 3.0 Win	Saf 3.1 Win	Opera 9.26	Opera 9.5b	Konqueror 3.5.7
<u>blur</u>		yes			too many	almost	incomplete	almost	incomplete		incomplete

When an element loses the focus.

- Firefox 2 fires too many events in a variety of circumstances.
- Firefox 3 fires too many events when blurring the window.
- Safari and Opera don't support these events on links and/or form fields in all circumstances.
- Konqueror doesn't support these events on the browser window.

Event	IE 5.5	IE 6	IE 7	IE8b1	FF 2	FF 3b5	Saf 3.0 Win	Saf 3.1 Win	Opera 9.26	Opera 9.5b	Konqueror 3.5.7
<u>change</u>			buggy		yes		yes		yes		yes

When a form field value changes.

- IE has a serious bug in its handling of this event on checkboxes and radios.

Thank you for your attention

Event	IE 5.5	IE 6	IE 7	IE8b1	FF 2	FF 3b5	Saf 3.0 Win	Saf 3.1 Win	Opera 9.26	Opera 9.5b	Konqueror 3.5.7
<u>click</u>			yes		yes		yes		yes		yes

When a mousedown and mouseup event occur on the same element OR an element is activated by the keyboard.

Event	IE 5.5	IE 6	IE 7	IE8b1	FF 2	FF 3b5	Saf 3.0 Win	Saf 3.1 Win	Opera 9.26	Opera 9.5b	Konqueror 3.5.7
<u>contextmenu</u>		yes		minimal	yes	buggy	yes		no		no

When the user right-clicks to get the context menu.

Preventing the default (i.e. preventing the context menu from appearing) is the whole point of this event.

Questions?

- Firefox 2 fires too many events in a variety of circumstances.
- Firefox 3 fires too many events when blurring the window.
- Safari and Opera don't support these events on links and/or form fields in all circumstances.
- Konqueror doesn't support these events on the browser window.

Ask away.

- IE has a serious bug in its handling of this event on checkboxes and radios.

Or ask me on Twitter

<http://twitter.com/ppk>

or on my site

<http://quirksmode.org>