

Hell is other browsers - *Sartre*

# State of the Browsers

Peter-Paul Koch (ppk)

<http://quirksmode.org>

<http://twitter.com/ppk>

The Ajax Experience, Sept. 30, 2008

# quirksmode.org

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><a href="#">blur</a></p> <p>When an element loses the focus.</p> <ul style="list-style-type: none"> <li>Firefox 2 fires too many events in a variety of circumstances.</li> <li>Firefox 3 fires too many events when blurring the window.</li> <li>Safari and Opera don't support these events on links and/or form fields in all circumstances.</li> <li>Konqueror doesn't support these events on the browser window.</li> </ul>	yes	yes	yes	yes	too many	almost	incomplete	almost	incomplete	incomplete	incomplete
<p><a href="#">change</a></p> <p>When a form field value changes.</p> <ul style="list-style-type: none"> <li>IE has a serious bug in its handling of this event on checkboxes and radios.</li> </ul>	buggy	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
<p><a href="#">click</a></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><a href="#">contextmenu</a></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

# Today's topics

I'm not going to discuss many browser incompatibilities.

That means that your favorite bug will not be treated.

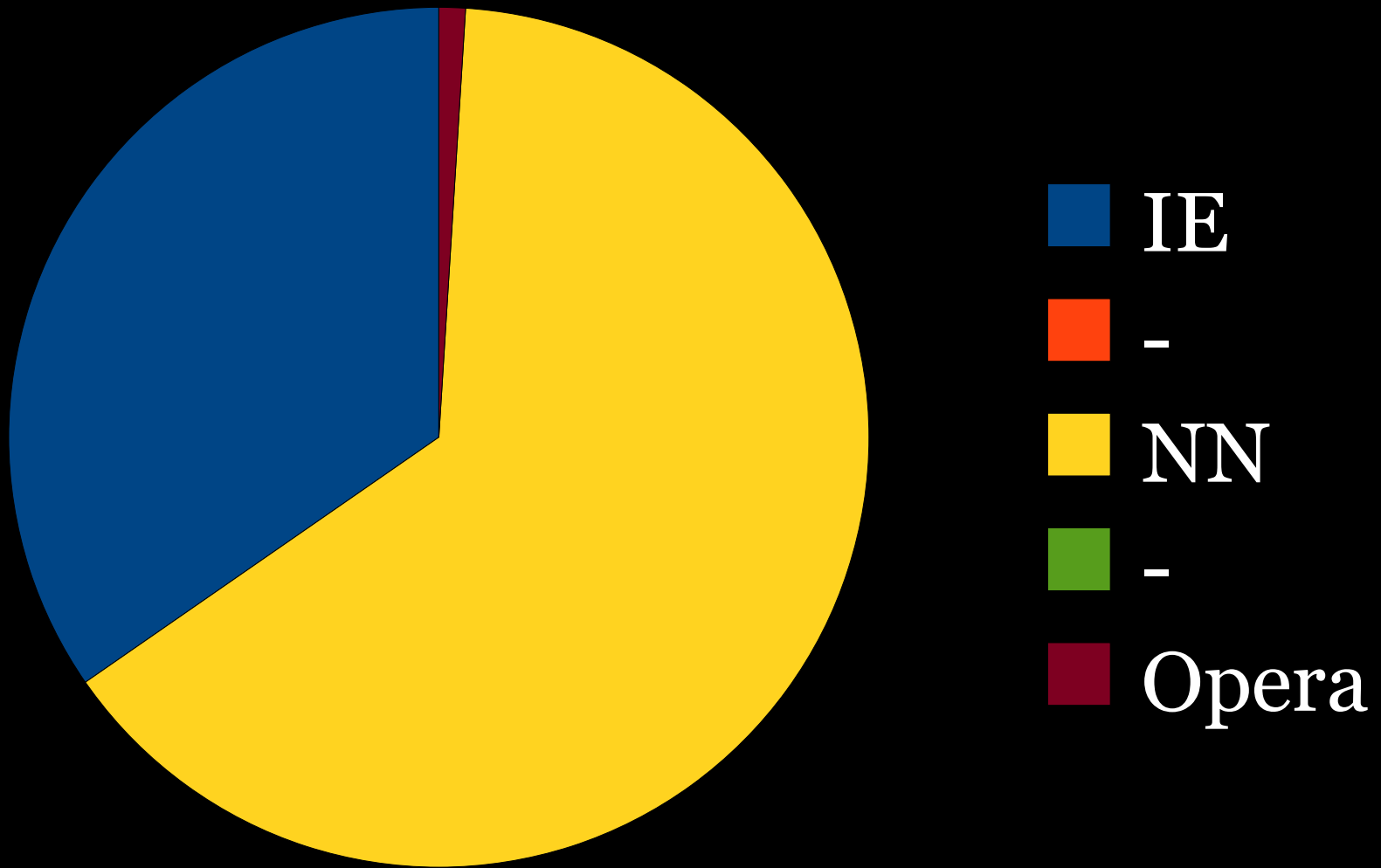
Sorry.

# Today's topics

- A bit of history
- Compatibility patterns, with examples
- Browser detects and how to do them right
- A peek into the future

# A bit of history

# The browser market long ago



# Browser Wars, 1996-1999



# Browser Wars Era

A world divided

- Netscape Navigator
- MS Internet Explorer

and they were incompatible.



A black and white historical photograph showing soldiers in a trench during a battle. The scene is filled with smoke and dust. In the foreground, a soldier in a helmet and uniform is crouching, holding a rifle. Behind him, another soldier is crawling. To the right, a third soldier is standing and looking towards the camera. The ground is uneven and rocky.

document.layers

document.all

# Browser Wars Era

- Deliberate incompatibilities

Why?

To leverage their competitor out of the market.

# Browser Wars Era

- Deliberate incompatibilities

That didn't work.

Why not?

Users don't care which  
browser they're using.

If a site doesn't work they blame  
the site owner.

Whether that's fair or not.

# Browser Wars Era

- Deliberate incompatibilities



**НАРОД И АРМИЯ ЕДИНЫ!**

# Browser Wars Era

- Deliberate incompatibilities
- Ideological in nature

Microsoft is Evil.

Why?

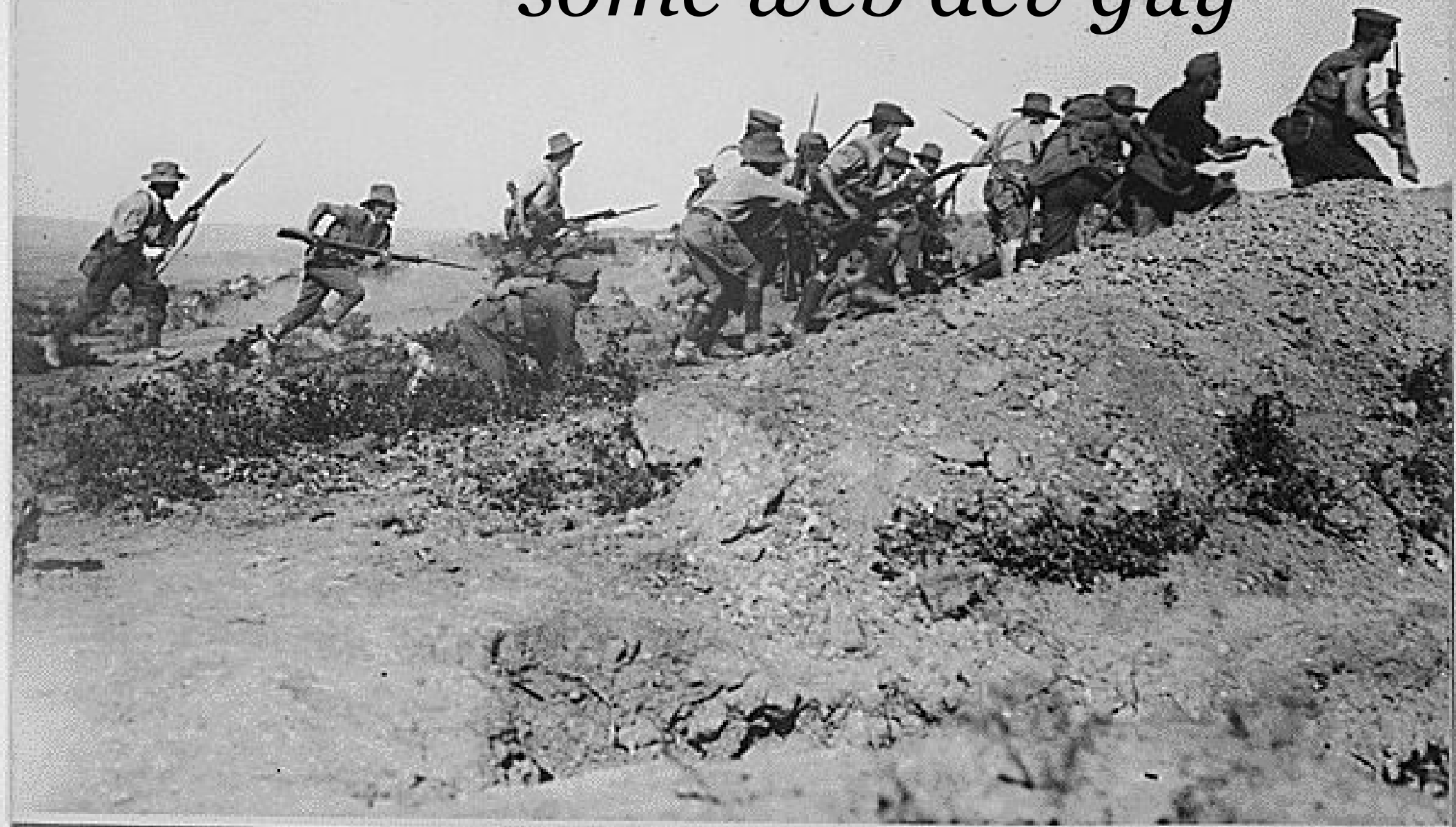
Well, just because.

# Browser Wars Era

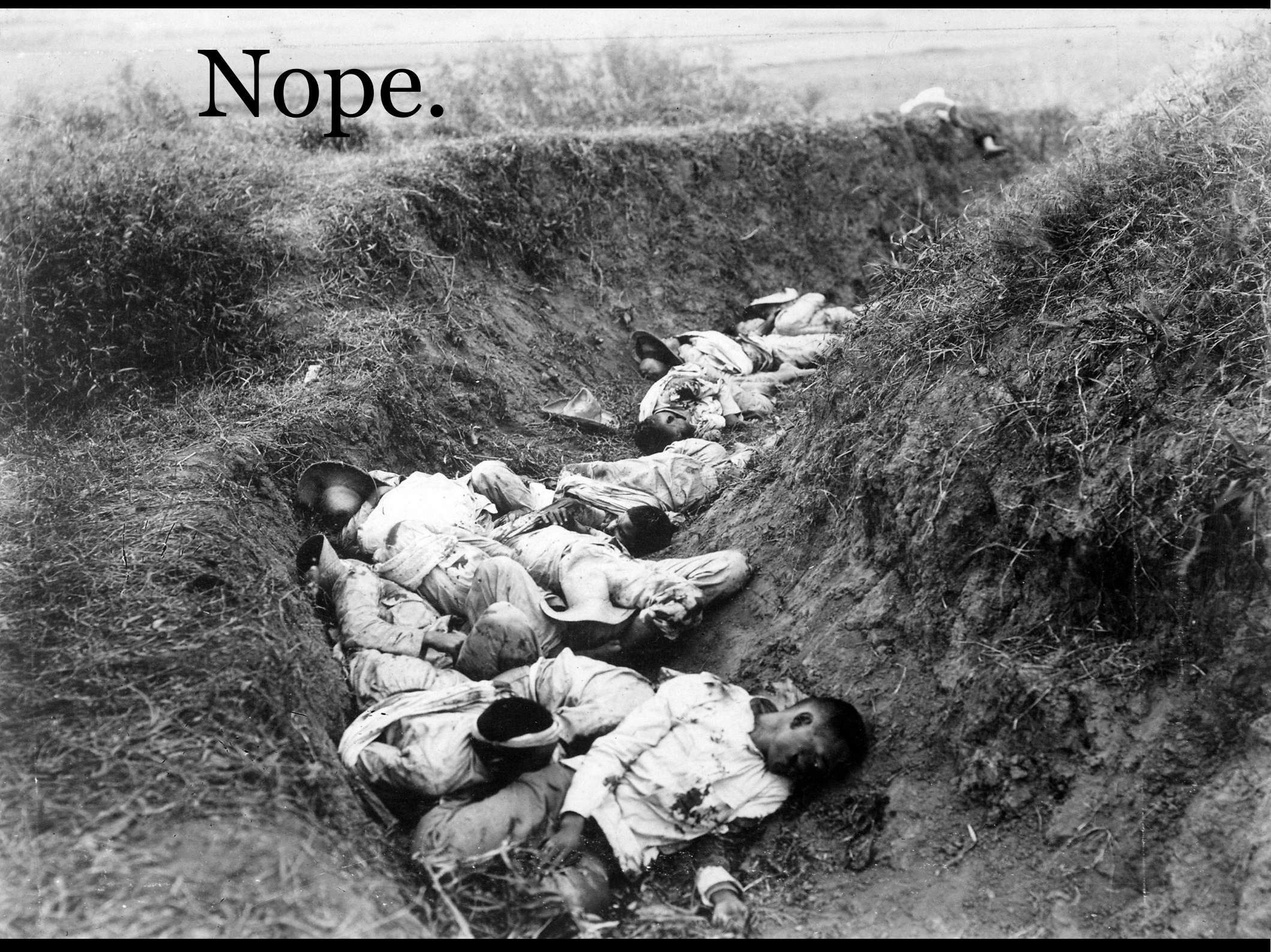
- Deliberate incompatibilities
- Ideological in nature
- Influence of web devs: zero



“Let's hope this works...”  
- *some web dev guy*



Nope.



# Browser Wars Era

- Deliberate incompatibilities
- Ideological in nature
- Influence of web devs: zero

Why?

Because they were supposed to choose sides and shut up.

# Browser Wars Era

- Deliberate incompatibilities
- Ideological in nature
- Influence of web devs: zero

But that didn't work because users don't care which browser they're using.

# Browser Wars Era

- Deliberate incompatibilities
- Ideological in nature
- Influence of web devs: zero

Until...



Until...



# Browser Peace

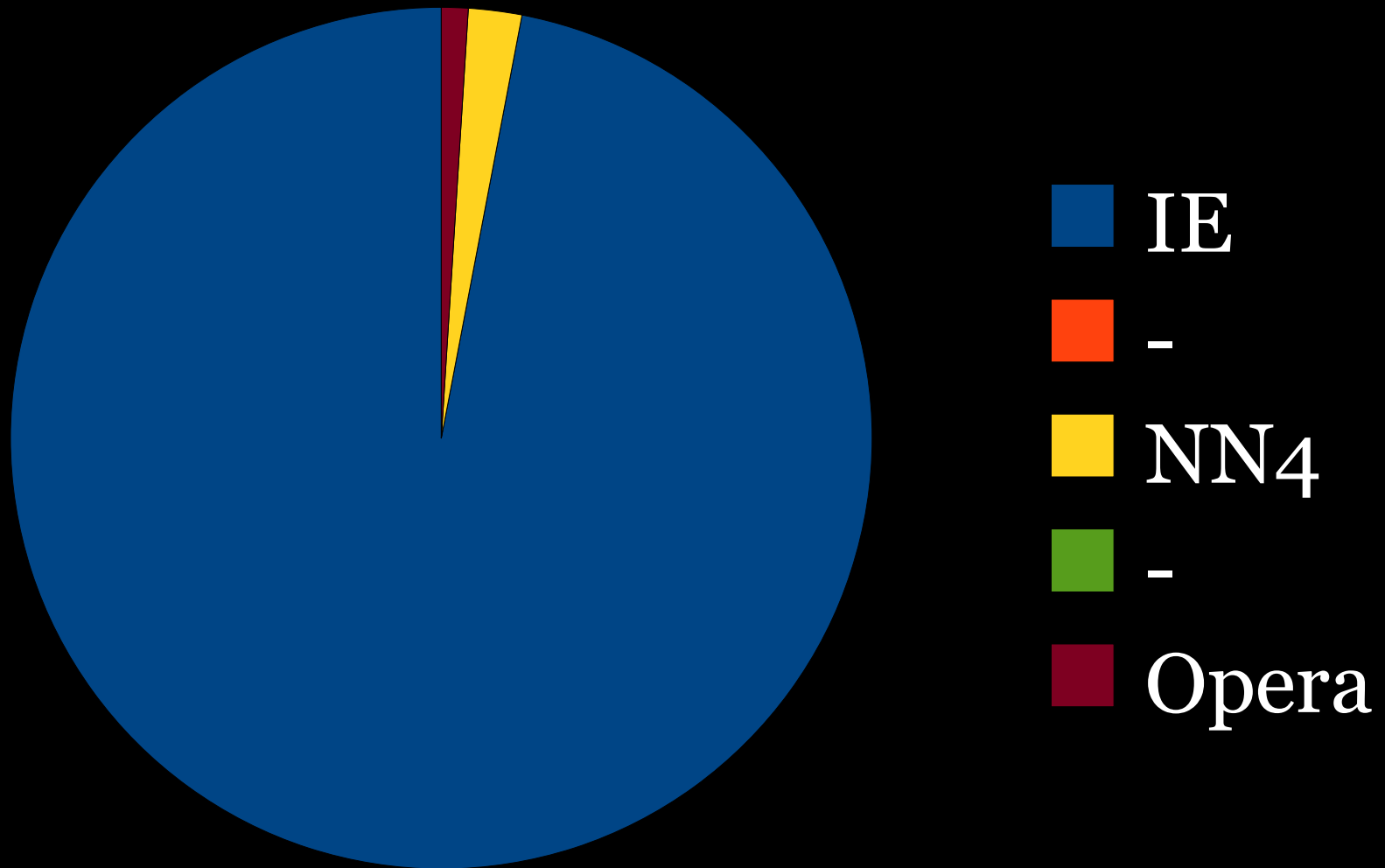
IE 5.0 Windows was the first browser to decently support the W3C DOM.

And IE5 Mac did the same for CSS1.

Back then, Microsoft *deserved* to win.



# The browser market back then



# Browser Peace 1999-2008?

Although Microsoft deserved to win back in 1999,  
it became complacent  
and didn't do anything for the next six  
years.

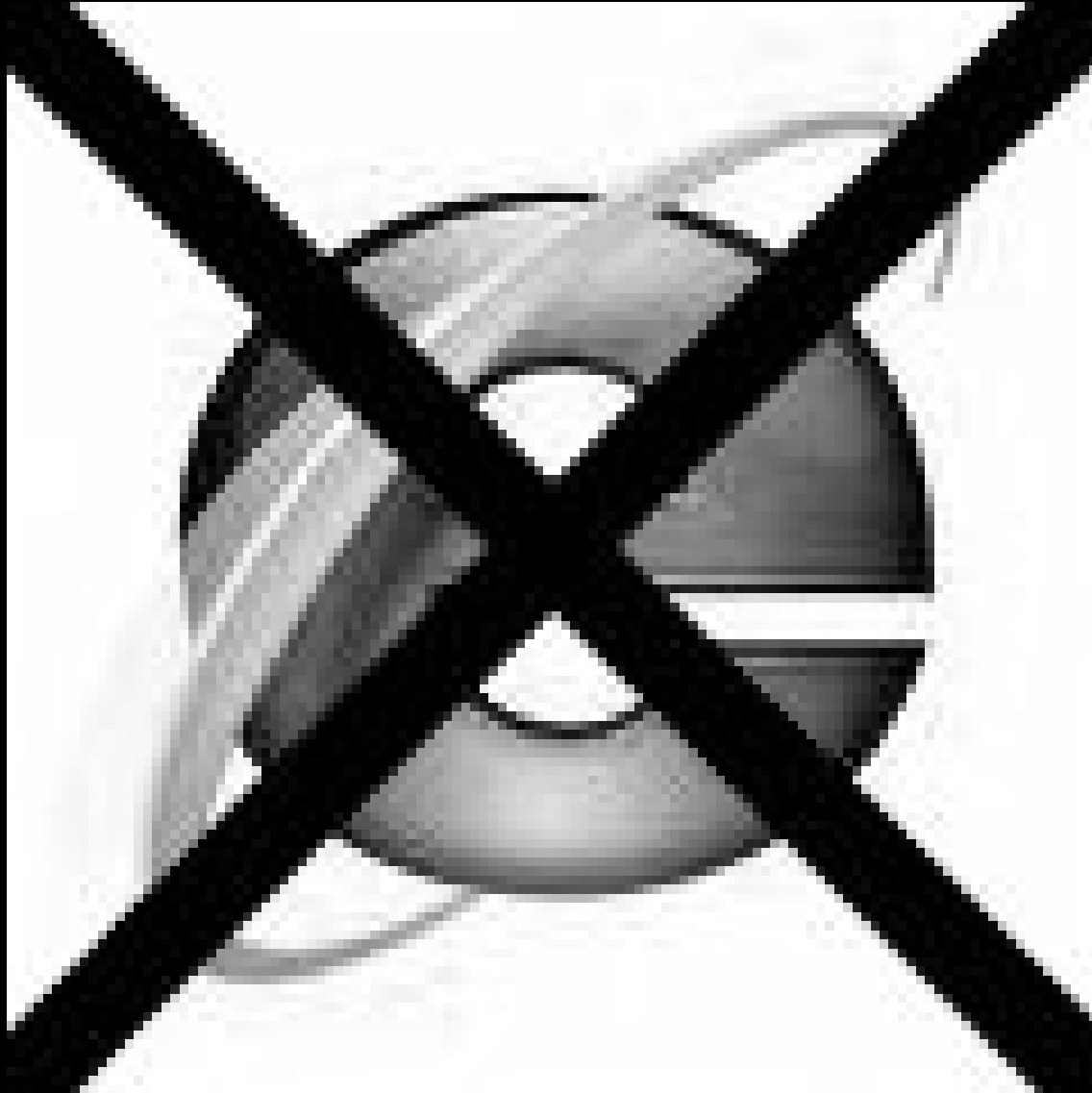
Other browsers profited from that.

# Today's browsers

- IE 7
- Firefox 3.0
- Safari 3.1
- Opera 9.5
- Chrome 0.2



# When IE gets it wrong



# When IE gets it wrong

<sigh />

The good news: Microsoft is actually working on it.

# Adding rules to style sheets

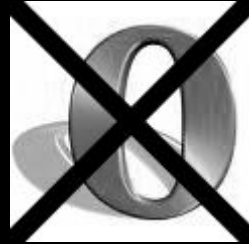
```
styleSheet.insertRule(...);
```



# Adding rules to style sheets

```
styleSheet.addRule(...);
```

Fortunately, this is easily solved.



# Adding rules to style sheets

```
if (styleSheet.insertRule) {  
    styleSheet.insertRule(...);  
}  
else if (styleSheet.addRule) {  
    styleSheet.addRule(...);  
}
```



# Compatibility patterns

#1

IE uses a different name for essentially the same functionality.

A simple code branch solves this problem.

# Event registration

```
element.addEventListener  
('click',someFn,false);
```



# Event registration

```
element.attachEvent('onclick',someFn);
```



# Compatibility patterns

#2

Minor browsers sometimes have to support major browsers' proprietary stuff.



# Event registration

```
element.attachEvent('onclick',someFn)
```

```
function someFn() {  
    this.style.backgroundColor = '#ffffff';  
}
```

# Event registration

```
element.attachEvent('onclick',someFn)
```

```
function someFn() {  
    this.style.backgroundColor = '#ffffff';  
}
```

You expect `this` to refer to the clicked element.

# Event registration

```
element.attachEvent('onclick',someFn)
```

```
function someFn() {  
    this.style.backgroundColor = '#ffffff';  
}
```

Unfortunately `this` refers to the window.  
Except in Opera.

# Event registration

In other words:

Opera is *not buggy enough*.

(Plz fix. Thx.)



# Compatibility patterns

#2b

Minor browsers sometimes have  
to support major browsers'  
proprietary stuff  
*and copy their bugs*



# Ranges

```
<body>
```

```
  <blockquote>
```

```
    Web developers! Web developers! Web  
    developers!
```

```
  </blockquote>
```

```
  <cite>Steve Ballmer</cite>
```

```
</body>
```

# Ranges

```
<body>  
  <blockquote>  
    Web developers! Web developers! Web  
    developers!  
  </blockquote>  
  <cite>Steve Ballmer</cite>  
</body>
```

# Ranges

```
<body>  
  <blockquote>  
    Web developers! Web developers! Web  
    developers!  
  </blockquote>  
  <cite>Steve Ballmer</cite>  
</body>
```

In which nodes does the selection start and end?

# Ranges

In which nodes does the selection start and end?

range.startContainer  
range.endContainer



# Ranges

In which nodes does the selection start and end?

Not possible in IE.

(Well, OK, you could parse the HTML to search for the selection.)

(Have fun.)



# Compatibility patterns

#3

Sometimes browsers just don't support stuff at all.

Can't be helped  
except by pressuring them.

# When IE gets it right





# Compatibility patterns

The difference between extensions and incompatibilities.

# When IE gets it right

Microsoft extensions.

99% are boring or useless

1% are stunningly brilliant

# When IE gets it right

Brilliant Microsoft extensions

- :hover
- innerHTML

Let's review a few more.

# Empty text nodes

```
<body>  
  <blockquote>  
    Web developers! Web developers! Web  
    developers!  
  </blockquote>  
  <cite>Steve Ballmer</cite>  
</body>
```

How many children does the body have?

# Empty text nodes

`<body>`  
One child

`<blockquote>`  
Two children

Web developers! Web developers! Web  
developers!  
Three children

`</blockquote>`  
Four children

`<cite>Steve Ballmer</cite>`  
Five children

How many children does the body have?

# Empty text nodes

`<body>`  
One child

`<blockquote>`  
Two children

Web developers! Web developers! Web  
developers!  
Three children

`</blockquote>`  
Four children

`<cite>Steve Ballmer</cite>`  
Five children

How many children does the body have?

# Empty text nodes

The first child is not the blockquote

```
<body>  
  <blockquote>  
    Web developers! Web developers! Web  
    developers!  
  </blockquote>  
  <cite>Steve Ballmer</cite>  
</body>
```

# Empty text nodes

but the *whitespace* between body and blockquote

```
<body>.....
```

```
....<blockquote>
```

```
  Web developers! Web developers! Web  
  developers!
```

```
</blockquote>
```

```
<cite>Steve Ballmer</cite>
```

```
</body>
```



# Empty text nodes

and yes, that's totally absurd.

Whitespace is unimportant in HTML.

```
<body>.....
```

```
....<blockquote>
```

```
  Web developers! Web developers! Web  
  developers!
```

```
</blockquote>
```

```
<cite>Steve Ballmer</cite>
```

```
</body>
```

# Empty text nodes

Fortunately IE gets it right. The body has two children: the blockquote and the cite

```
<body>
```

```
1) <blockquote>
```

```
    Web developers! Web developers! Web  
    developers!
```

```
    </blockquote>
```

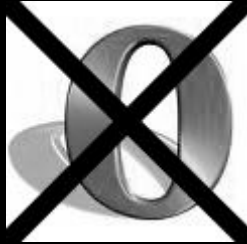
```
2) <cite>Steve Ballmer</cite>
```

```
</body>
```

# Empty text nodes

Everybody gets it wrong.

Except for IE.



# Compatibility patterns

#4

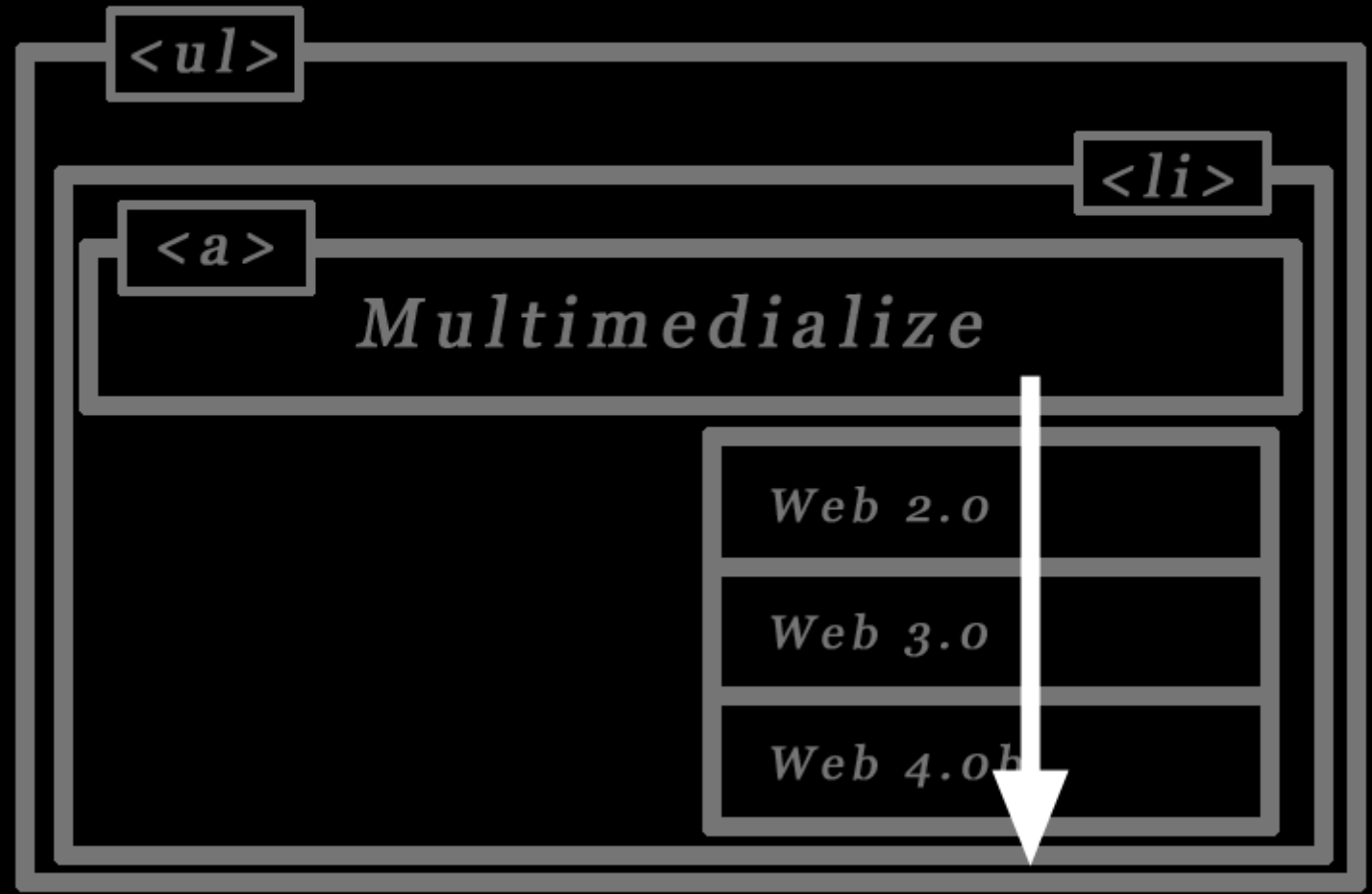
Deliberate incompatibilities

are wrong

except when you're right.

# Dropdown menu

You want to know when the mouse exits the opened submenu.



# Dropdown menu

a.onmouseout

a.onmouseover

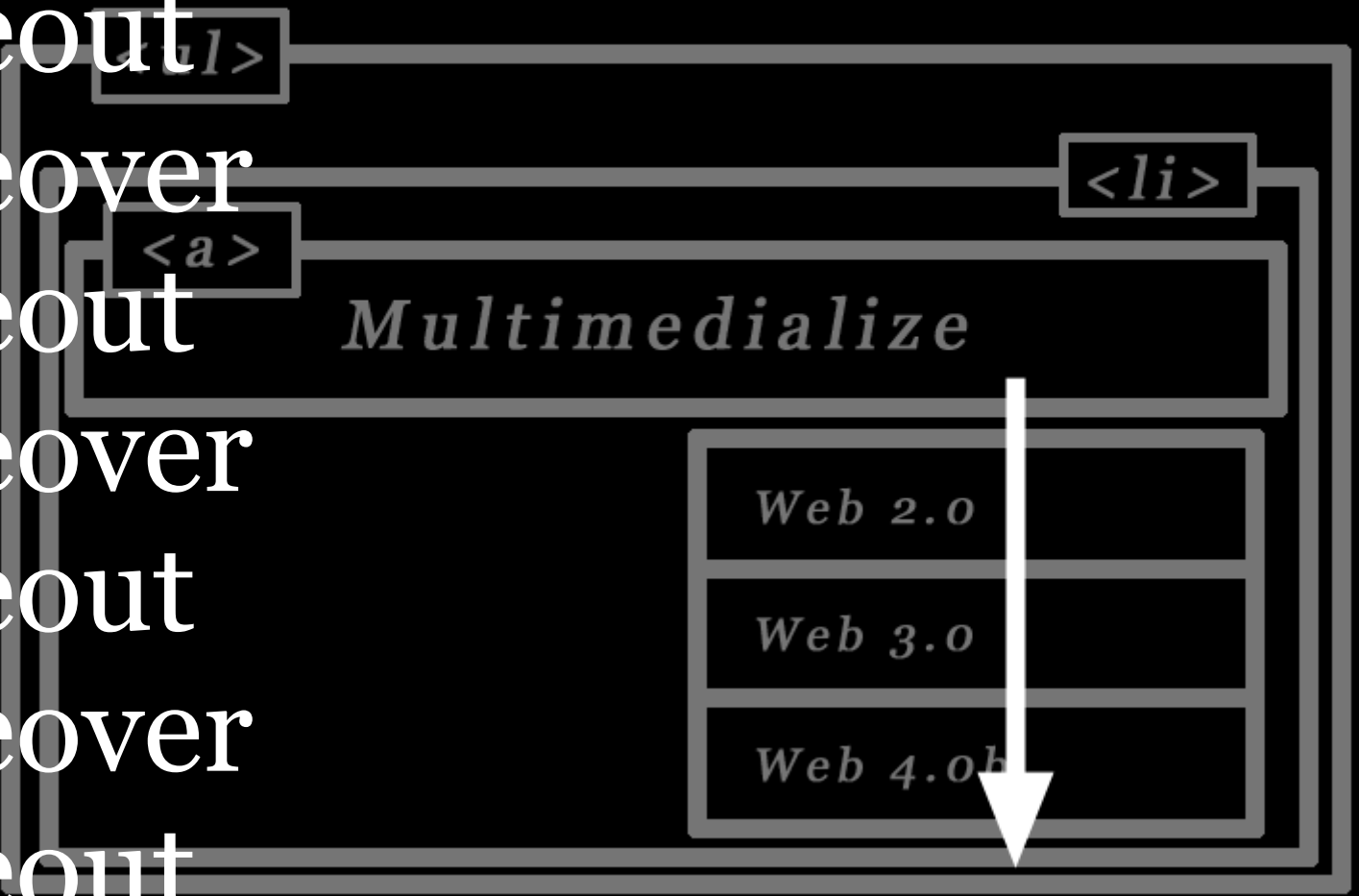
a.onmouseout

a.onmouseover

a.onmouseout

a.onmouseover

a.onmouseout



# Dropdown menu

a.onmouseout

Lots of stuff!

a.onmouseover

Fun!

a.onmouseout

a.onmouseover

But has the

a.onmouseout

mouse left

a.onmouseover

the submenu

a.onmouseout

yet?

# Dropdown menu

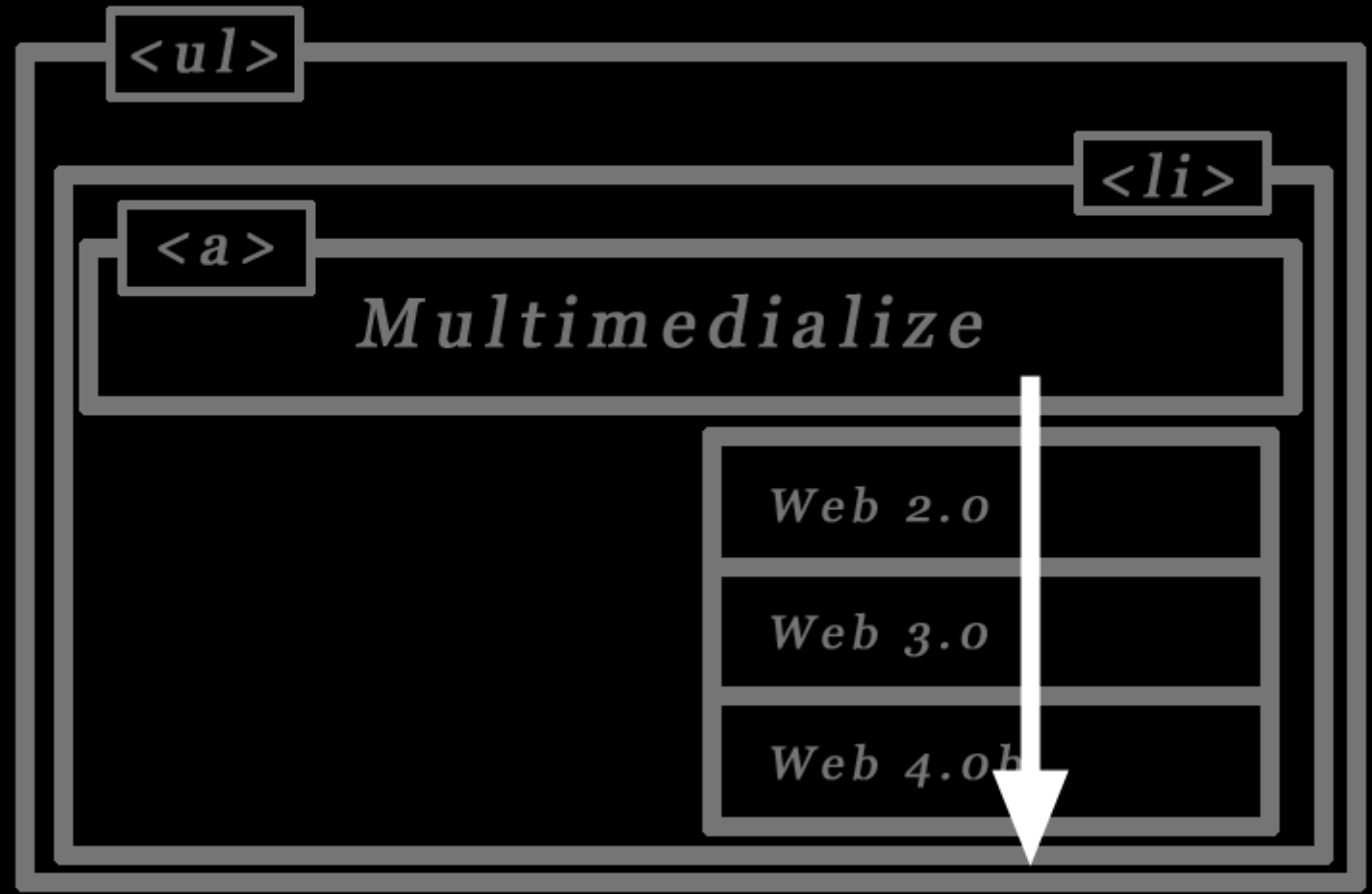
Mouseover and mouseout aren't very developer-friendly.

That's why IE added mouseenter and mouseleave.



# Dropdown menu

You want to know when the mouse exits the opened submenu.



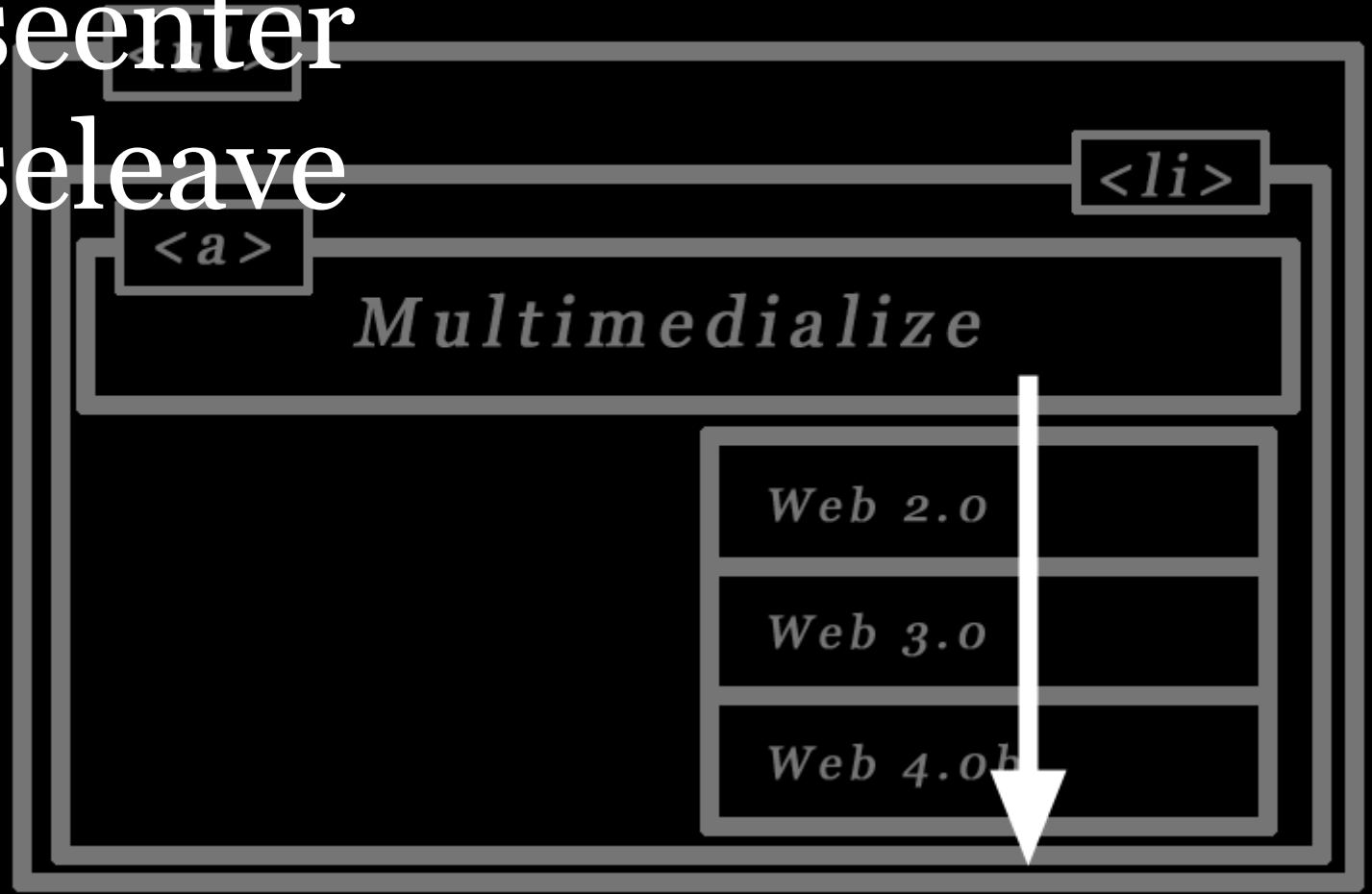
# Dropdown menu

`ul.onmouseenter`

`ul.onmouseleave`

...

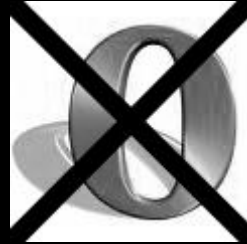
that's it



# Dropdown menu

works better with mouseenter and mouseleave.

IE-only.



# Compatibility patterns

#5

This is a proprietary extension.

A way of saying “We think this is a good idea”

If other browser vendors agree, it might become part of the standard

# When FF gets it wrong



# When FF gets it wrong

(or at least ignores common sense)

There are a few methods and properties that are not defined in the spec but are nonetheless so totally useful that all browsers support them.

Except for Firefox.

# children

```
<body>  
  <blockquote>  
    Web developers! Web developers! Web  
    developers!  
  </blockquote>  
  <cite>Steve Ballmer</cite>  
</body>
```

# children

Five children

```
document.body.childNodes[]
```

-> 5 elements

But we're usually interested only in the children that are elements.



# children

Five children, two of which are elements

`document.body.children[]`

-> 2 elements

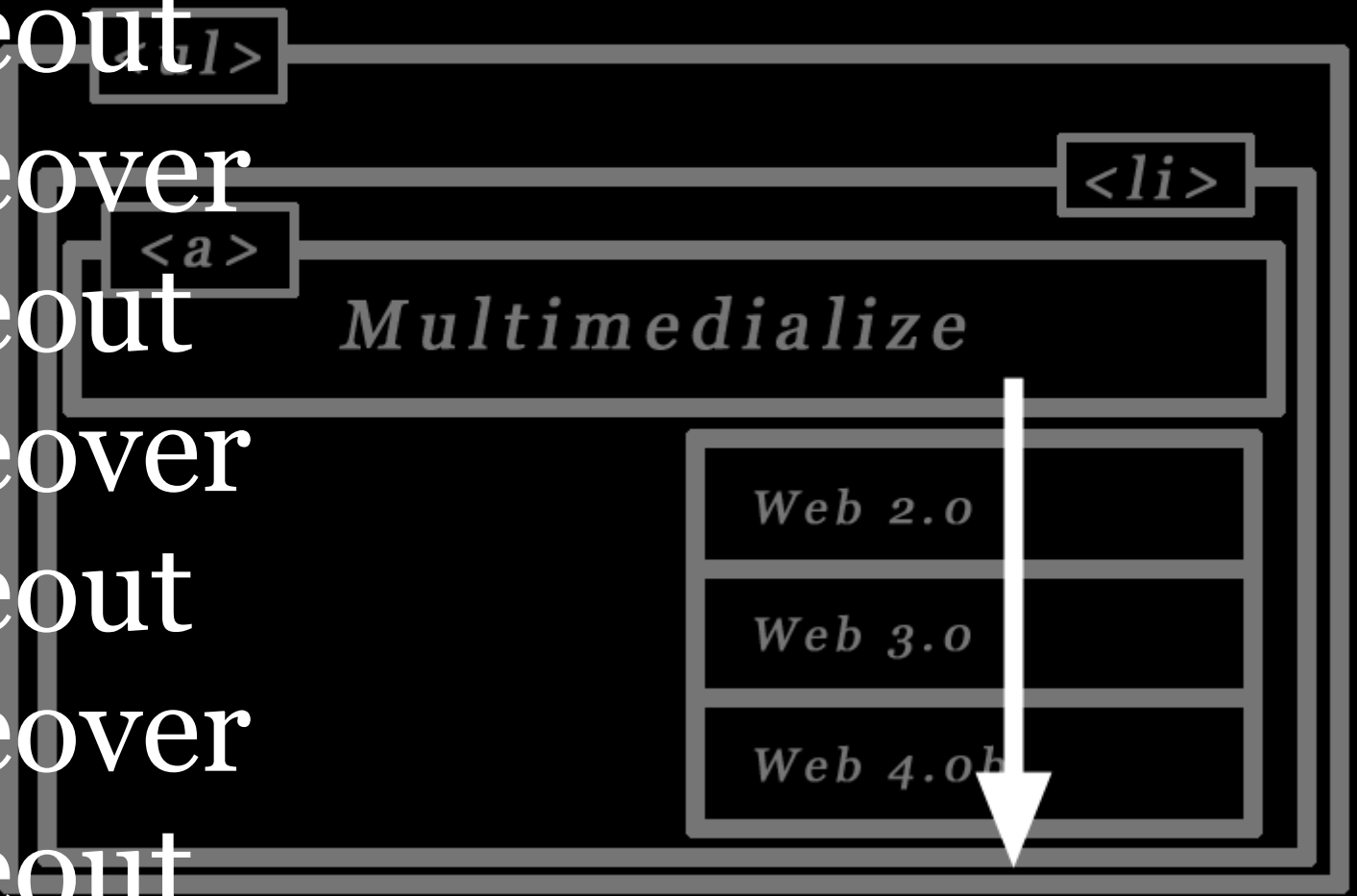
The blockquote and the cite.

Just what we need



# contains

a.onmouseout  
a.onmouseover  
a.onmouseout  
a.onmouseover  
a.onmouseout  
a.onmouseover  
a.onmouseout



contains

“Does the currently opened layer contain the element the user moused *to*?”

If No, the layer should close.

# contains

“Does the currently opened layer contain the element the user moused *to*?”

```
if (!current.contains(related)) {  
    current.close();  
}
```

# contains

“Does the currently opened layer contain the element the user moused *to*?”

```
if (!current.contains(related)) {  
    current.close();  
}
```



# Compatibility patterns

#5b

These are also proprietary extensions.

If they're a really good idea, all browsers should support them.  
(And FF 3.1 is.)

# Compatibility patterns

IE is not the only browser to refuse to support useful stuff for years on end.

Browser incompatibilities will remain a fact of life for the near future.

That means we need to know which browsers visit our sites.



# Browser detects

m  
(you e  
ou?)



# Browser detects

- 1) To gather stats about browser use on your site.
- 2) To take application decisions.

# Browser detects

1) To gather stats about browser use on your site.  
GOOD.

2) To take application decisions.

# Browser detects

1) To gather stats about browser use on your site.

2) To take application decisions.

EVIL

(most of the time)

# The problem

Once upon a time there were Mosaic and Netscape.

Now Netscape could do nifty stuff.

- cookies
- `<center>`

# The problem

Once upon a time there were Mosaic and Netscape.

Both sent a `userAgent` HTTP header back to the server.

(JavaScript: `navigator.userAgent`)

# The problem

```
<% if userAgent.contains('Mozilla') %>
  <H1>
    <CENTER>
      Hello, <% cookie.name %>!
    </H1>
  </CENTER>
<% else %>
  <H1>
    You're really UNCOOL, you know?
  </H1>
<% endif %>
```

# The problem

Then came IE

It wanted to end up on the right side of those browser detects.

```
Mozilla/3.0 (compatible; MSIE  
yadda yadda yadda)
```



# The problem

Then came IE

It wanted to end up on the right side of those browser detects.

```
Mozilla/3.0 (compatible; MSIE  
yadda yadda yadda)
```

It disguised itself as Netscape.

By the time the Browser Wars ended



By the time the the Browser Wars ended

many web developers had decided to support only IE.

In retrospect, a sensible decision, given its market share and development ease.

By the time the the Browser Wars  
ended

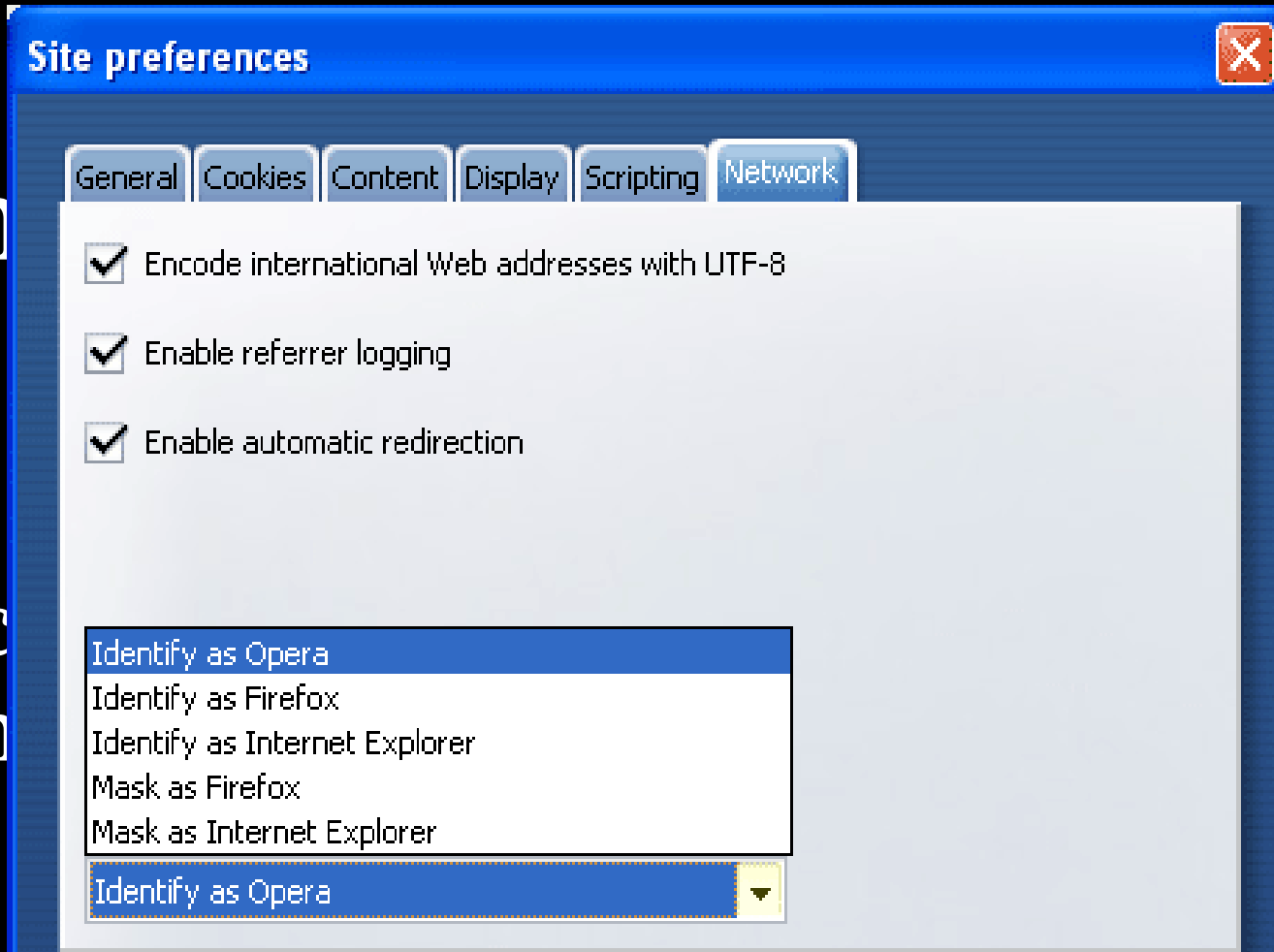
many web developers had decided  
to support only IE.

```
if (userAgent.contains('MSIE')) {  
  <H1>Welcome</H1>  
} else {  
  redirect('wrongbrowser.html')  
}
```

# But Opera could handle most of those sites

So when

It hid  
(More c  
differen



ect a

## Identify as Opera:

Opera/9.26 (Windows NT 5.1; U; en)

## Identify as IE:

Mozilla/4.0 (compatible; MSIE 6.0;  
Windows NT 5.1; en) Opera 9.26

## Mask as IE:

Mozilla/4.0 (compatible; MSIE 6.0;  
Windows NT 5.1; en)

## Identify as Opera:

Opera/9.26 (Windows NT 5.1; U; en)

## Identify as IE:

Mozilla/4.0 (compatible; MSIE 6.0;  
Windows NT 5.1; en) Opera 9.26

## Mask as IE:

Mozilla/4.0 (compatible; MSIE 6.0;  
Windows NT 5.1; en) ...

# Browser detects

Browsers commonly lie about their identity

in order to bypass browser detects written by clueless web developers

That's OUR fault.



# Browser detects

Browsers commonly lie about  
their identity

`navigator.userAgent` is

unreliable

in order to bypass browser detects  
written by clueless web developers

That's OUR fault.

# Doing it right

1) Use a property meant for identification

such as `window.opera`

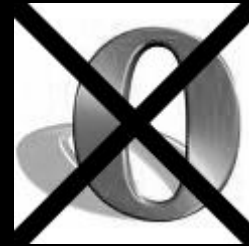
or IE conditional comments



# Doing it right

- 1) Use a property meant for identification
- 2) Use `navigator.vendor`

It does not lie.  
Yet.



# Doing it right

- 1) Use a property meant for identification
- 2) Use `navigator.vendor`
- 3) Use `navigator.userAgent`  
and first detect the minor browsers  
Resolve FF/IE only when you're  
certain that it's no other browser

# Doing it right

- 1) Use a property meant for identification
- 2) Use `navigator.vendor`
- 3) Use `navigator.userAgent`

# Doing it right

- 1) Use a property meant for identification
- 2) Use `navigator.vendor`
- 3) Use `navigator.userAgent`

The first two steps can only be done in JavaScript.

# Doing it right

- 1) Use a property meant for identification
- 2) Use `navigator.vendor`

A JavaScript browser detect is more trustworthy than a server side one.

# What's next?

IE, Firefox, Safari, Opera  
Now Chrome

A shake-up is inevitable.



# Browser Wars II, 2008-?



# A world divided?

- Deliberate incompatibilities
- Ideological in nature
- Influence of web devs: zero

# A world divided?

- ~~- Deliberate incompatibilities~~
- Ideological in nature
- Influence of web devs: zero

# A world divided?

- ~~Deliberate incompatibilities~~
- Ideological in nature
- Influence of web devs: some

# A world divided?

- ~~Deliberate incompatibilities~~
- Ideological in nature
- Influence of web devs: some

# Ideology



Microsoft is Evil.

Yawn.

Go to war again?

To defeat  
the Evil  
Empire?



Go to war again?

Or  
to gain status  
and nice  
medals?





Or maybe...  
not go to war at all ... ?



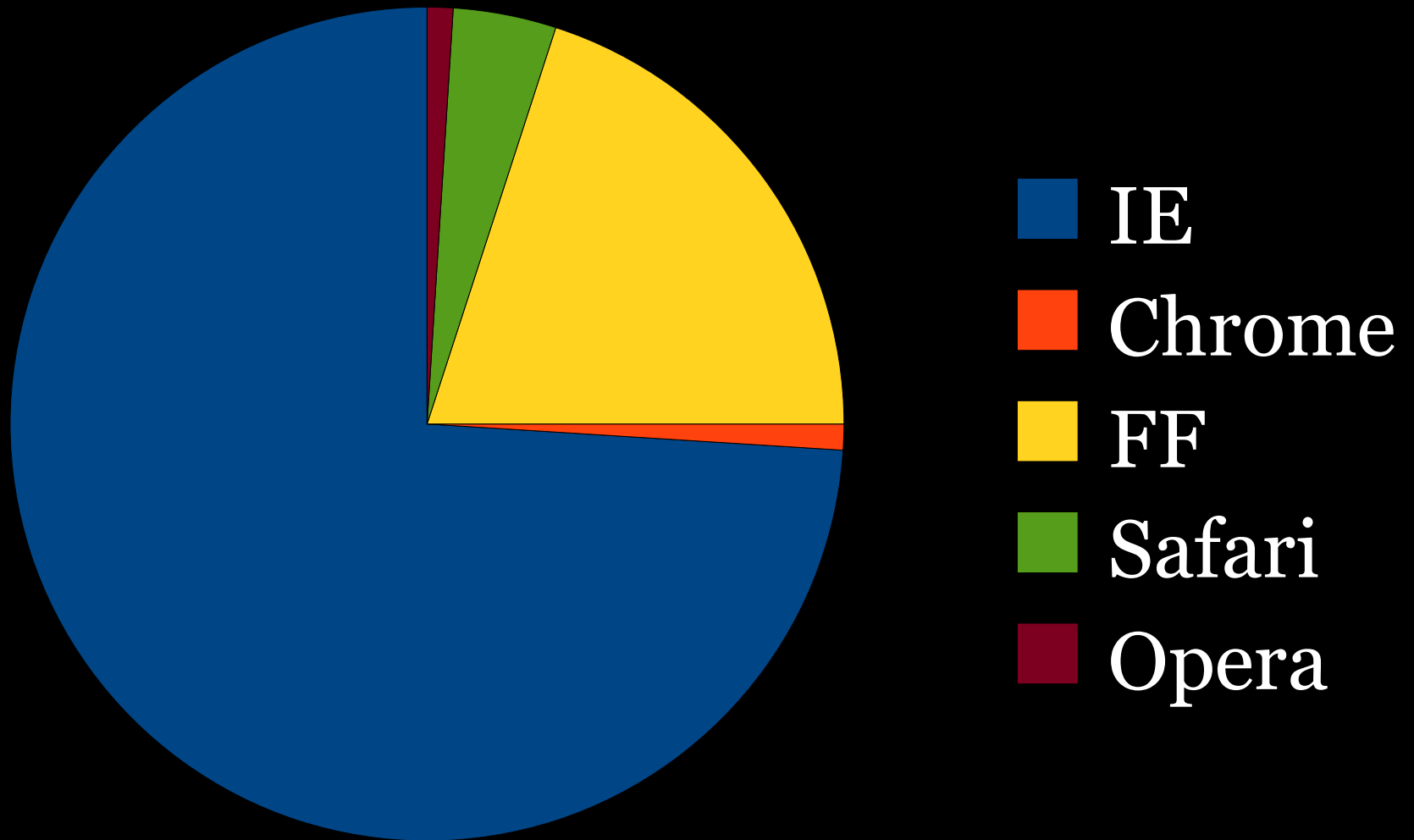
# Go to war again?

The only reason would be to influence changes in the browser market.

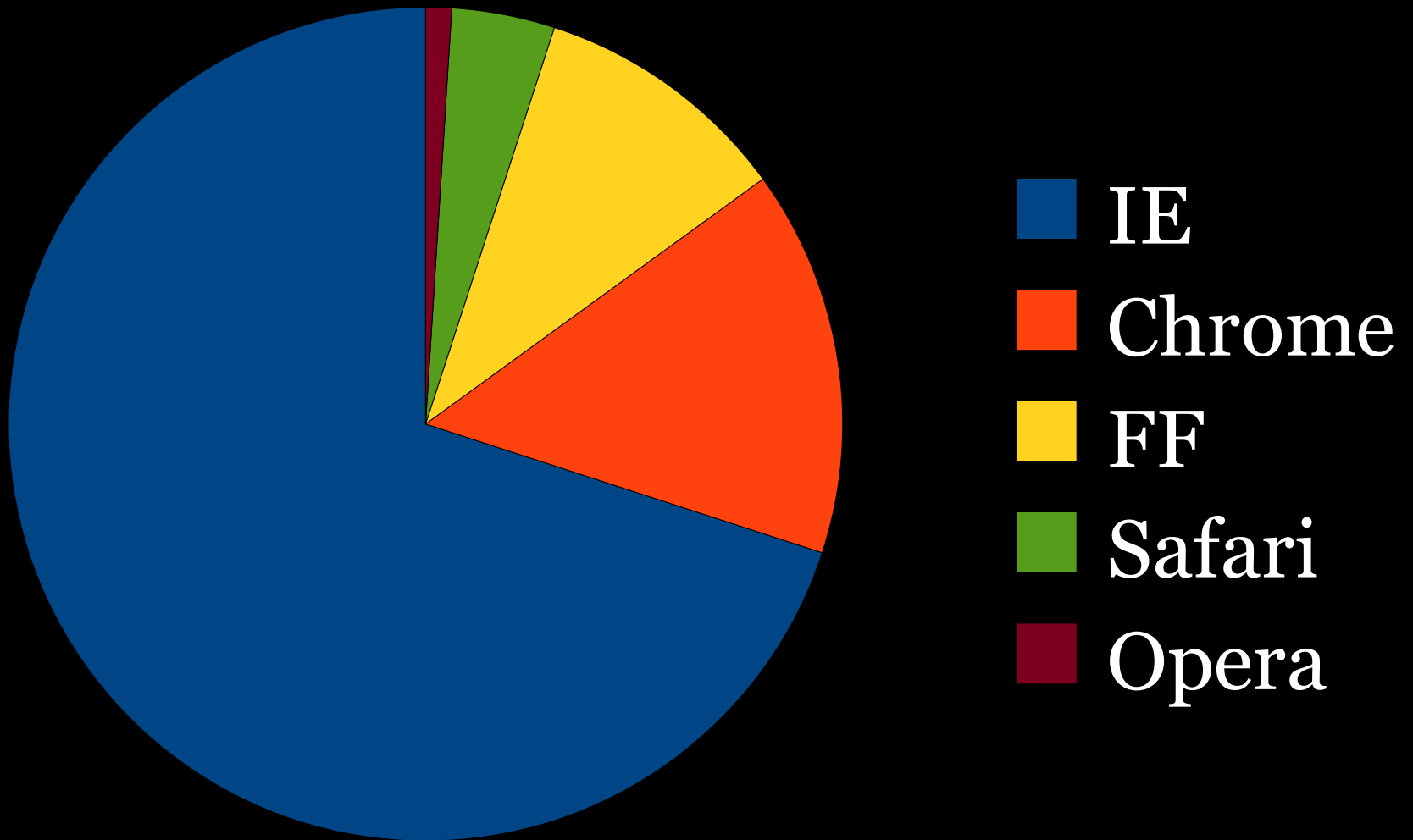
But can we?

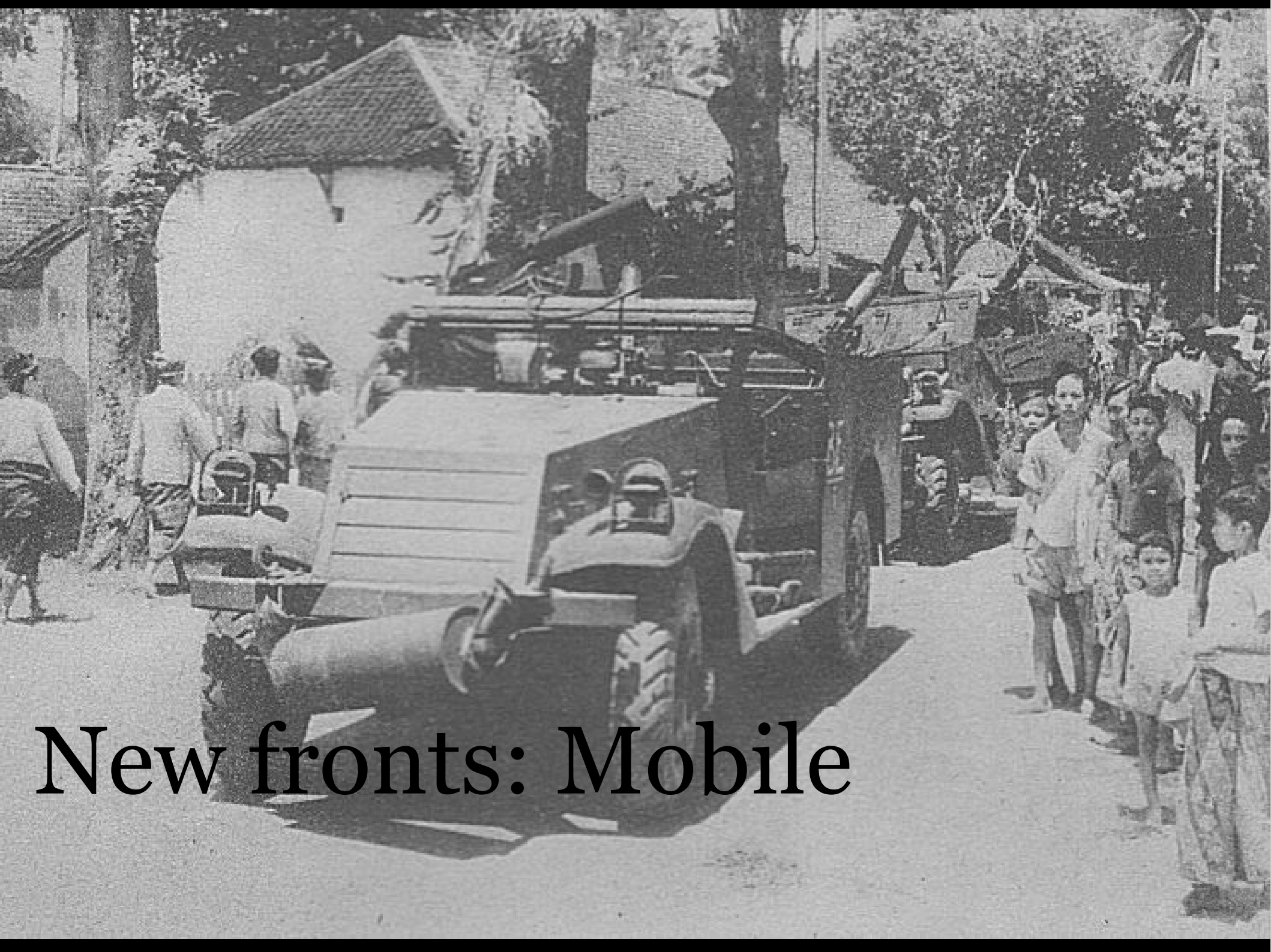
And if we can, should we?

# The current browser market



# The future browser market?





New fronts: Mobile

# New fronts: Mobile

A dark, grayscale photograph of a busy street scene in a developing area. In the foreground, a large truck is parked on the left side of the road. To the right, a group of people, including children, are standing and looking towards the camera. The background shows more buildings and trees, suggesting a densely populated urban or semi-urban environment. The overall tone is somber and documentary.

iPhone

Opera Mini

Opera Mobile

Windows Mobile

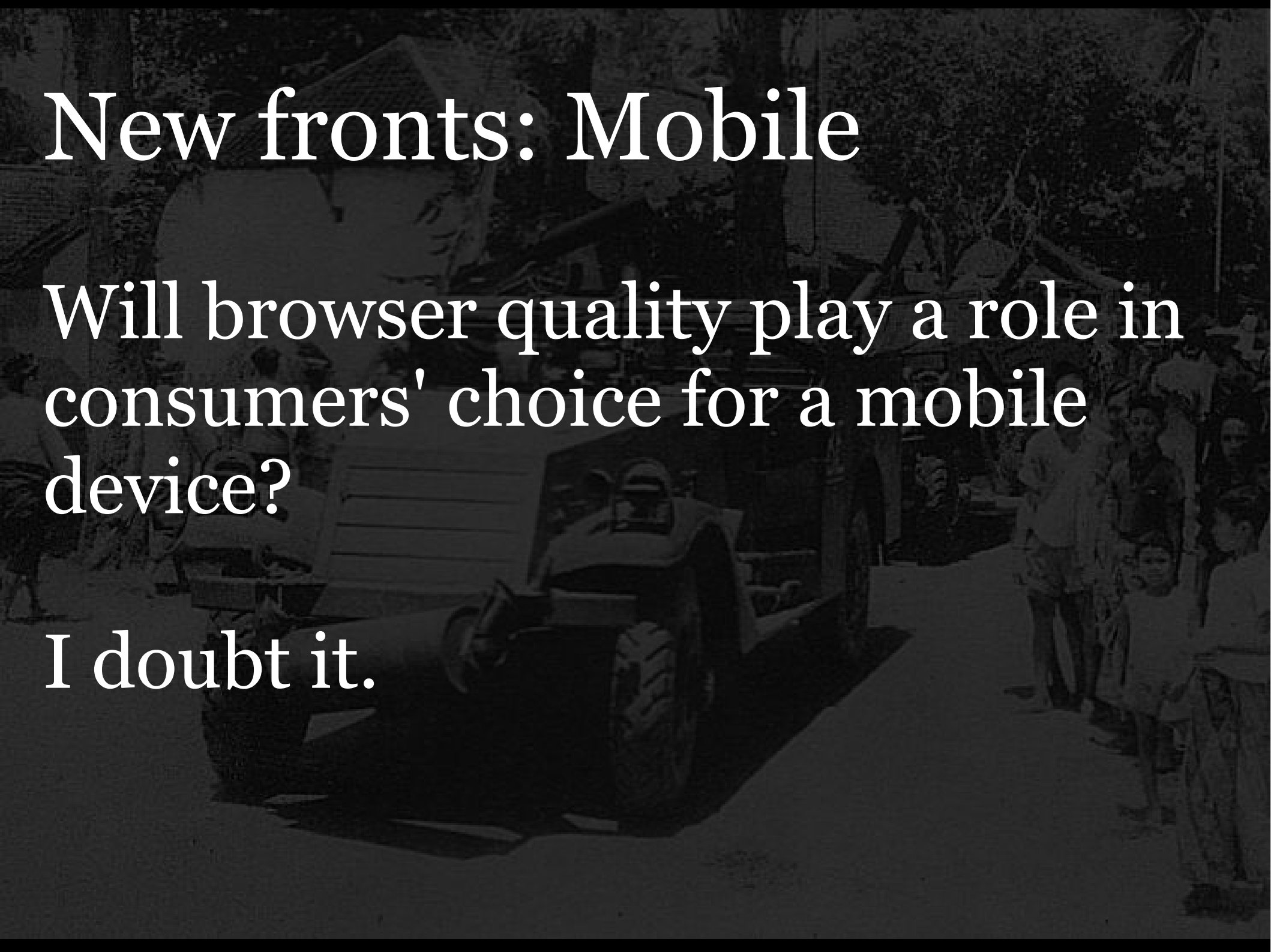
Google Android

...

# New fronts: Mobile

Will browser quality play a role in consumers' choice for a mobile device?

I doubt it.



A quote from  
Nate Koechley

“Why do we need  
more than one  
browser?”

Seriously, though.





“Why do we need  
more than one  
browser?”

Our answer:  
Competition, which  
fosters innovation.



Competition, which fosters innovation.

Is that good enough?

Good enough to force countless web developers to spend thousands of hours on solving browser incompatibilities?

“Why do we need  
more than one  
browser?”

If you have an answer, let me know.

# To wrap it up

Browsers will continue to move towards each other in terms of compatibility.

The browser market is going to change rapidly over the next year or so.

Maybe there'll be another browser war. I hope not, though.

Thank you

Questions?







