Hell is other browsers - *Sartre*

The Open Web goes Mobile

Peter-Paul Koch (ppk)
http://quirksmode.org
http://twitter.com/ppk
Google, 24 April 2009

The mobile web

Four problems with making a website work well on a phone:

- Small memory
- Small display
- Flaky browsers
- Flaky connections

The mobile web

Four problems with making a website phone-compatible:

- Small memory

Performance tests are necessary.

And I'll leave it at that.

The mobile web

Four problems with making a website phone-compatible:
- Small memory
- Small display

This problem has been recognized years ago, and people are thinking about it.

Small display

CSS: solve it with media queries

```
@media all and (max-width: 300px) {
    div#container {
        // special styles for small displays
    }
}
```

Supported by Opera, iPhone, Bolt and Iris.

Small display

JS: solve it with offsetWidth

```
if (document.body.offsetWidth < 300) {
    // special scripts for small displays
}
```

offsetWidth and offsetHeight seem well supported (except on Blackberry).

The mobile web

Four problems with making a website phone-compatible:

- Small memory
- Small display
- Flaky browsers
- Flaky connections

Mobile browsers

Thanks to Vodafone's generous support I'm now able to deliver a preliminary report on the State of the Mobile Browsers.

Mobile browsers

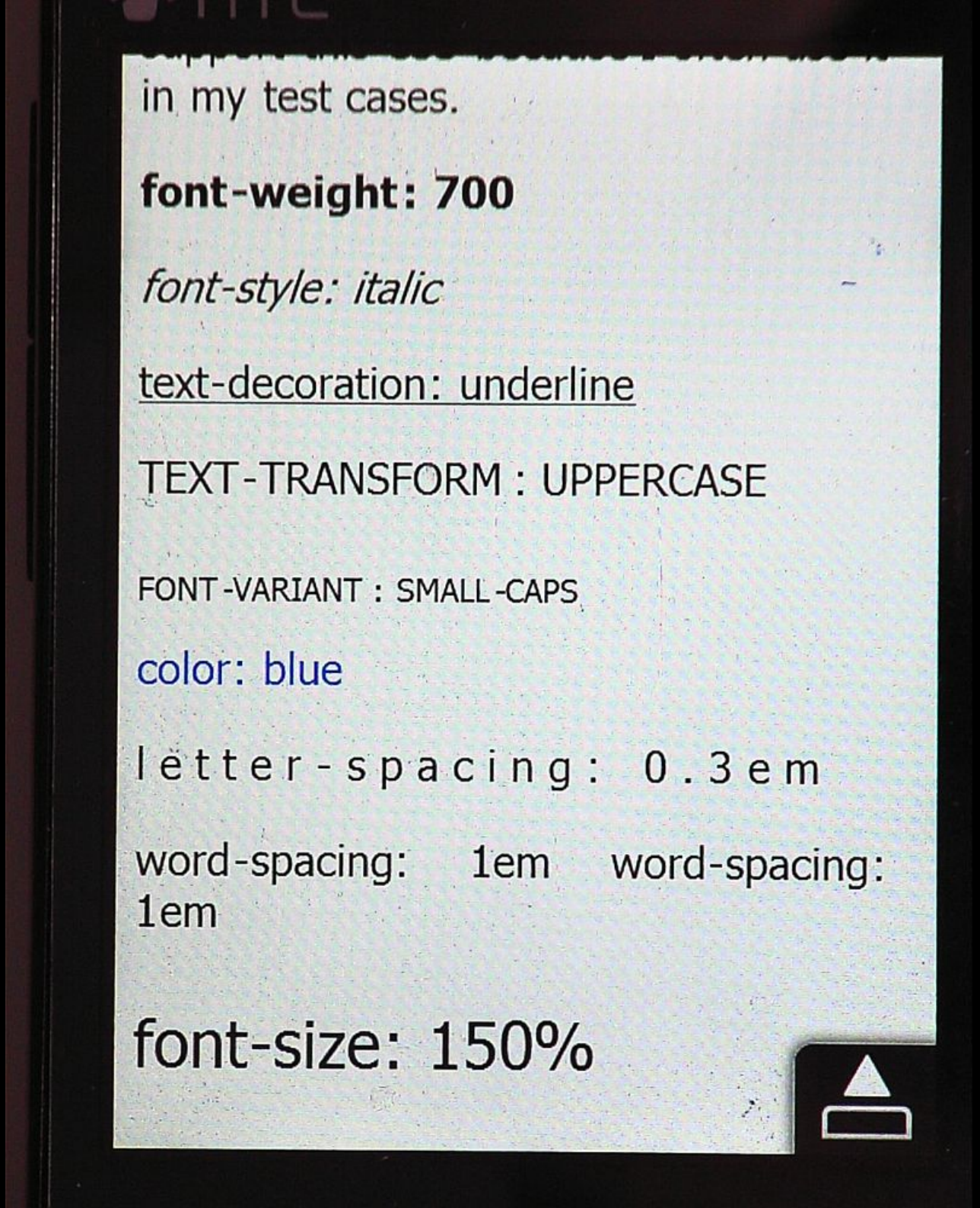
- Android WebKit
- Opera Mobile
- NetFront
- Safari
- Opera Mini
- Blackberry
- S60 WebKit
- IE Mobile
- Iris, Bolt, Skyfire, Obigo, OpenWeb, Nokia S40, Palm Blazer, Fennec, Teashark etc. etc.

You may groan now.

Mobile browsers

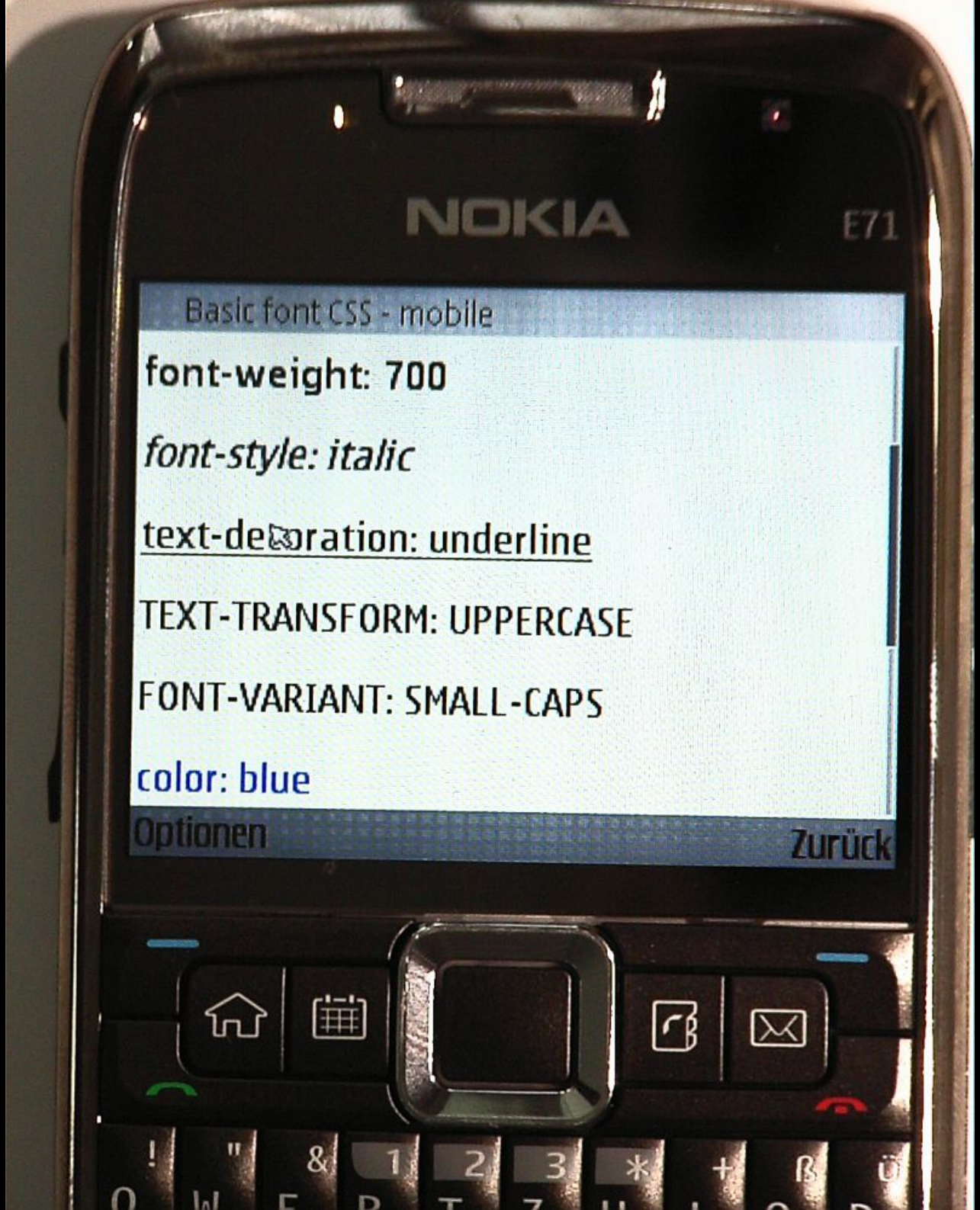
All these browsers have their own problems with advanced CSS and JavaScript.

Worse, you have to test really basic stuff, too
such as *font-style: italic*



Opera Mobile 9.5 on HTC Diamond Touch

Supported but only in desktop mode



vodafone

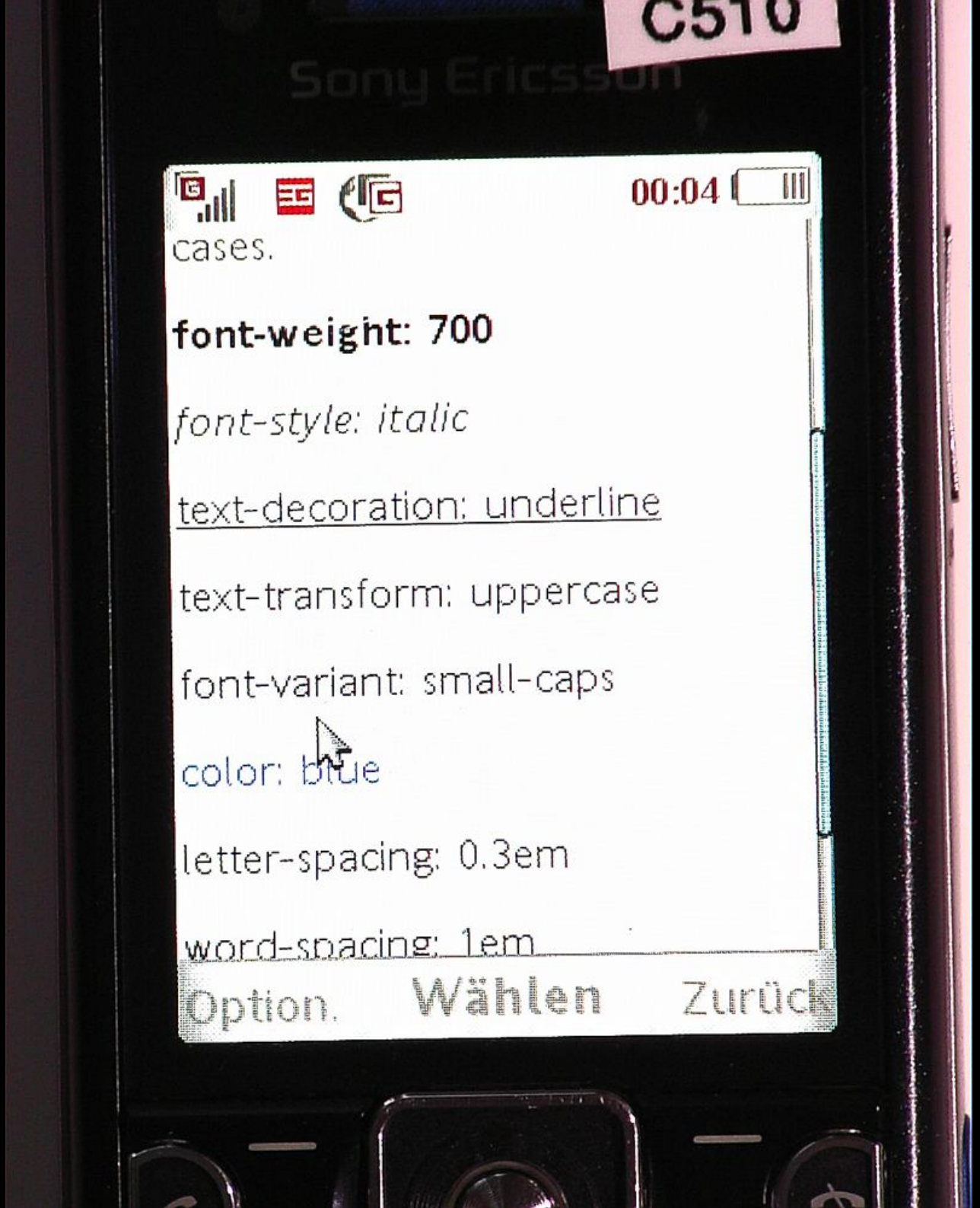
S60 WebKit on Nokia E71

Supported, but error in font-variant

vodafone

Opera Mini 4.2 on Nokia E71

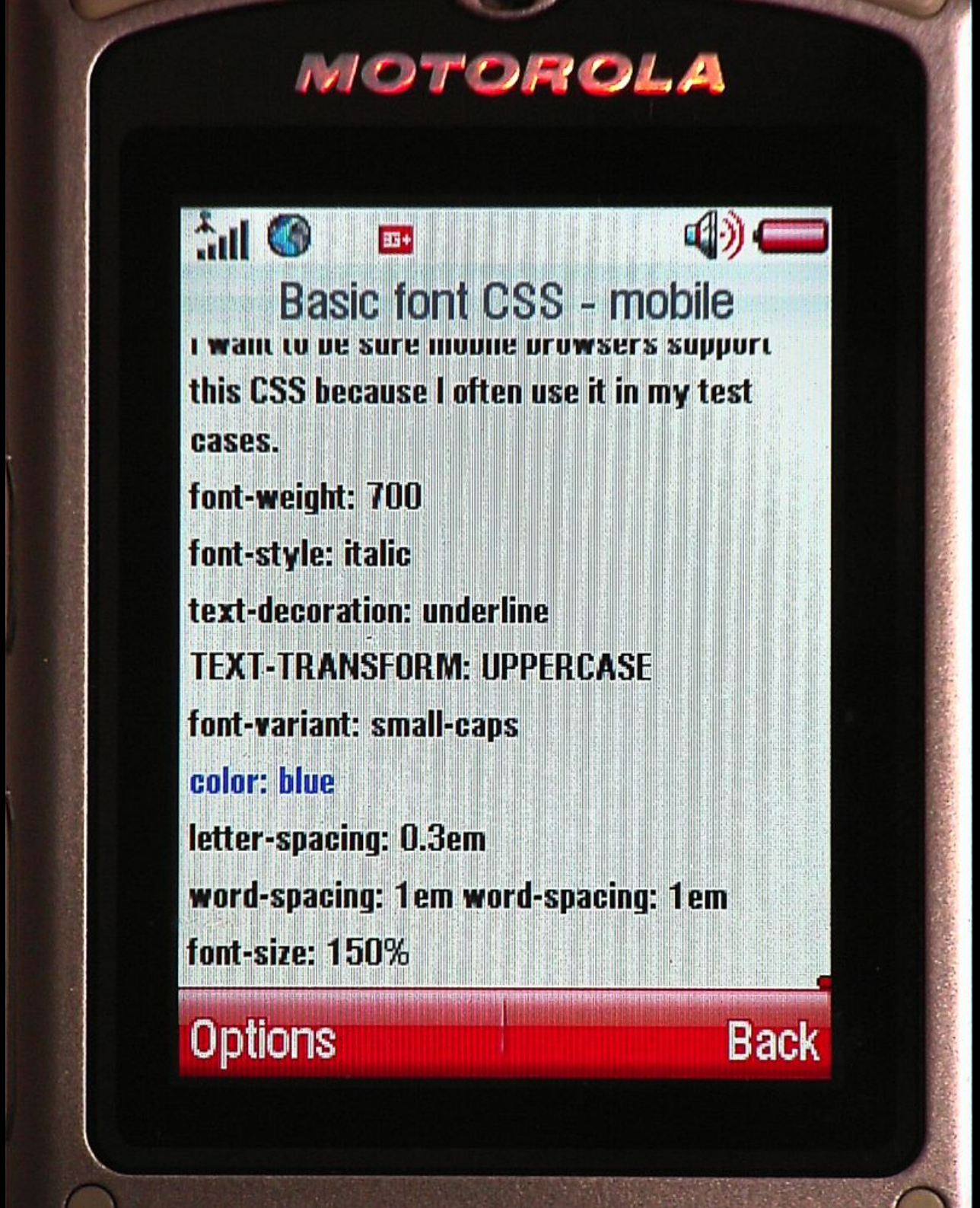
Supported except for letter-spacing



vodafone

NetFront on Sony Ericsson C510

Basics supported



vodafone

Opera Mobile 8.00 on Motorola V3xx

Only color and text-transform supported

Mobile browsers

The current top level mobile browsers are:

- Android WebKit
- Safari
- Opera Mobile

Mobile browsers

Top level:

Android WebKit, Safari, Opera Mobile

Mid level:

- S60 WebKit
- Blackberry
- Opera Mini

Mobile browsers

Top level:
Android WebKit, Safari, Opera Mobile

Mid level:
S60 WebKit, Blackberry, Opera Mini

Bottom level:
- NetFront
- IE Mobile (old)

Mobile browsers

Top level:

Android WebKit, Safari, Opera Mobile

Mid level:

S60 WebKit, Blackberry, Opera Mini

Bottom level:

NetFront, IE Mobile (old)

Other default browsers (old):

- OpenWeb, Nokia S40, Palm Blazer

Mobile browsers

Top level:

Android WebKit, Safari, Opera Mobile

Mid level:

S60 WebKit, Blackberry, Opera Mini

Bottom level:

NetFront, IE Mobile (old)

Other default browsers (old):

OpenWeb, Nokia S40, Palm Blazer

Other browsers (non-default):

- Iris, Bolt, Skyfire, Obigo, Fennec, Teashark etc. etc.

The mobile web

Four problems with making a website phone-compatible:

- Small memory
- Small display
- Flaky browsers
- Flaky connections

The mobile web

Four problems with making a website phone-compatible:

- Small memory
- Small display
- Flaky browsers
- Flaky connections

Flaky connections

If the guy next to you is downloading a few movies
your network connection will slow down regardless of how good it's supposed to be.

I don't see this problem disappearing any time soon.

Flaky connections

This is a serious problem for the mobile web,
especially when your site uses 200K of custom JavaScript plus a few libraries.

They have to be downloaded every time the user visits your site
and caching isn't always reliable.

Flaky connections

Solution:
Put the core files on your mobile phone

so that you only need to download the data.

Flaky connections

W3C Widgets offer this solution:
- Local applications
- HTML/CSS/JavaScript
- Run in a browser (any browser)
- Can handle Ajax requests

Flaky connections

The same approach is taken by all app systems:

- iPhone apps
- Windows Mobile apps
- Blackberry apps
- Android apps
- etc.

Flaky connections

The same approach is taken by all app systems:

- iPhone apps (proprietary)
- Windows Mobile apps (proprietary)
- Blackberry apps (proprietary)
- Android apps (proprietary)
- etc. (probably proprietary)

Open standards

If a company wants to put data on mobile phones, it can

- create a website (which may be slow to load)
- or create 4 or more separate applications (which is certainly expensive)

Open standards

If a company wants to put data on mobile phones, it can

~~- create a website (which may be slow to load)~~

~~- or create 4 or more separate applications (which is certainly expensive)~~

- use W3C Widgets

W3C Widgets

Widgets are better than websites because they download only the data; and not the core files.

Widgets are better than app systems because you don't have to write 4, 5, or 10 of them. Just the one is enough.

W3C Widgets

Eventually, I'll be able to share a
widget with a friend via Bluetooth,
even if I use an Android
and he uses a Nokia S60
or a HTC Windows Mobile
or a Blackberry

and It Just Works

W3C Widgets

Wouldn't that be
totally
astoundingly
absolutely
inconceivably
interoperable?

W3C Widgets

And hundreds of thousands of web developers *already know* how to create widgets.

It's just HTML/CSS/JavaScript, after all.

W3C Widgets

- Create 1 HTML page with as much CSS, JavaScript, and images you need.
- Add an icon and a config.xml
- Zip the lot
- Change extension to .wgt
- It Just Works.

W3C Widgets

Widgets will open the web faster and for more people than any other system.

If people can easily create them and share them with their friends and They Just Work why do we need anything else?

Application systems

An app system may remain more suited for some forms of applications:

- animation-heavy games
- secure applications
- more ... ?

Application systems

Besides, native app systems can foster innovation, too, and eventually W3C Widgets will profit from that.

Proprietary systems are fine
as long as you *also* support the standard.

W3C Widgets

It Just Works.

But not quite yet, unfortunately.

That's one of the reasons I'm here today.

W3C Widgets

It Just Works

in the Vodafone Widget Manager for S60 phones.

W3C Widgets

It Just Works
S60

in the Opera/T-Mobile Widget Manager for (probably) Windows Mobile phones.

W3C Widgets

It Just Works
S60
Windows Mobile

in the Nokia Widget Runtime on S60
(as long as you add an info.plist file)

W3C Widgets

It Just Works
S60 (2x)
Windows Mobile

Otherwise, though, there's no support.

Yet.

W3C Widgets

It Just Works.
S60 (2x)
Windows Mobile
Google Android?

W3C Widgets

It Just Works
in Google Android?

I came here to ask if it's possible Google Android will support W3C Widgets
not *instead of*
but *in addition to* its own app system.

W3C Widgets

We need:

- a browser (Android WebKit will do perfectly fine, thanks)
- a way of associating .wgt files with this browser OR an installation mechanism
- JavaScript device APIs

JavaScript Device APIs

are APIs that grant access to phone functionality

- camera
- contact list
- text messages
- etc.

JavaScript Device APIs

are necessary for a true mobile experience.

W3C widgets should be able to tie into phone functionality.

JavaScript Device APIs

- BONDI specification
  (not yet implemented)
- Phonegap library
  (Android, Blackberry, iPhone)
- Opera/T-Mobile widget manager
  (Windows Mobile)

JavaScript Device APIs
Security

If I receive a widget from someone
and it uses device APIs
how do I know it's not going to try to
steal my contact list?

JavaScript Device APIs
Security

This problem will probably be solved by signed widgets and security levels.

On the lowest security levels, phone users will be prompted for every device API call the widget wants to perform. Higher levels do it automatically.

JavaScript Device APIs
Security

Still, this problem will remain pretty serious and more research is necessary.

Google can certainly help us figure out a good answer.

W3C Widgets
Security

JavaScript's same-source policy is not implemented in widgets,
because they have to be able to request data from any source.

This, too, requires more thought.

W3C Widgets

Pros
- Open standards
- Countless people can already create them
- Interoperability on a massive scale
- They'll open the Web more quickly than any other system

W3C Widgets

Cons
- Other systems may remain better suited for certain applications
- As yet moderately supported
- Security issues

W3C Widgets

The pros heavily outweigh the cons.

So let's get to work.

Thank you
for your attention

Questions?

Ask away.

Or ask me on Twitter
http://twitter.com/**ppk**
or on my site
http://quirksmode.org