

Progressive Enhancement

Bridging the gap between CSS2 and CSS3

Zohar Arad, April 2010

zohar@zohararad.com | <http://www.zohararad.com>

Agenda

Current situation - Problems with implementing cross-platform web-designs

Progressive Enhancement - Solving cross-platform design implementation

CSS3 - Common desired features

Bridging the gap between CSS2 and CSS3

Current Situation



**Not all browsers
were created equal
in the eyes of God**



Current Situation

There are at least 4 major rendering engines out there:
Trident, WebKit, Gecko, Presto

Browser version releases and adoption are slow

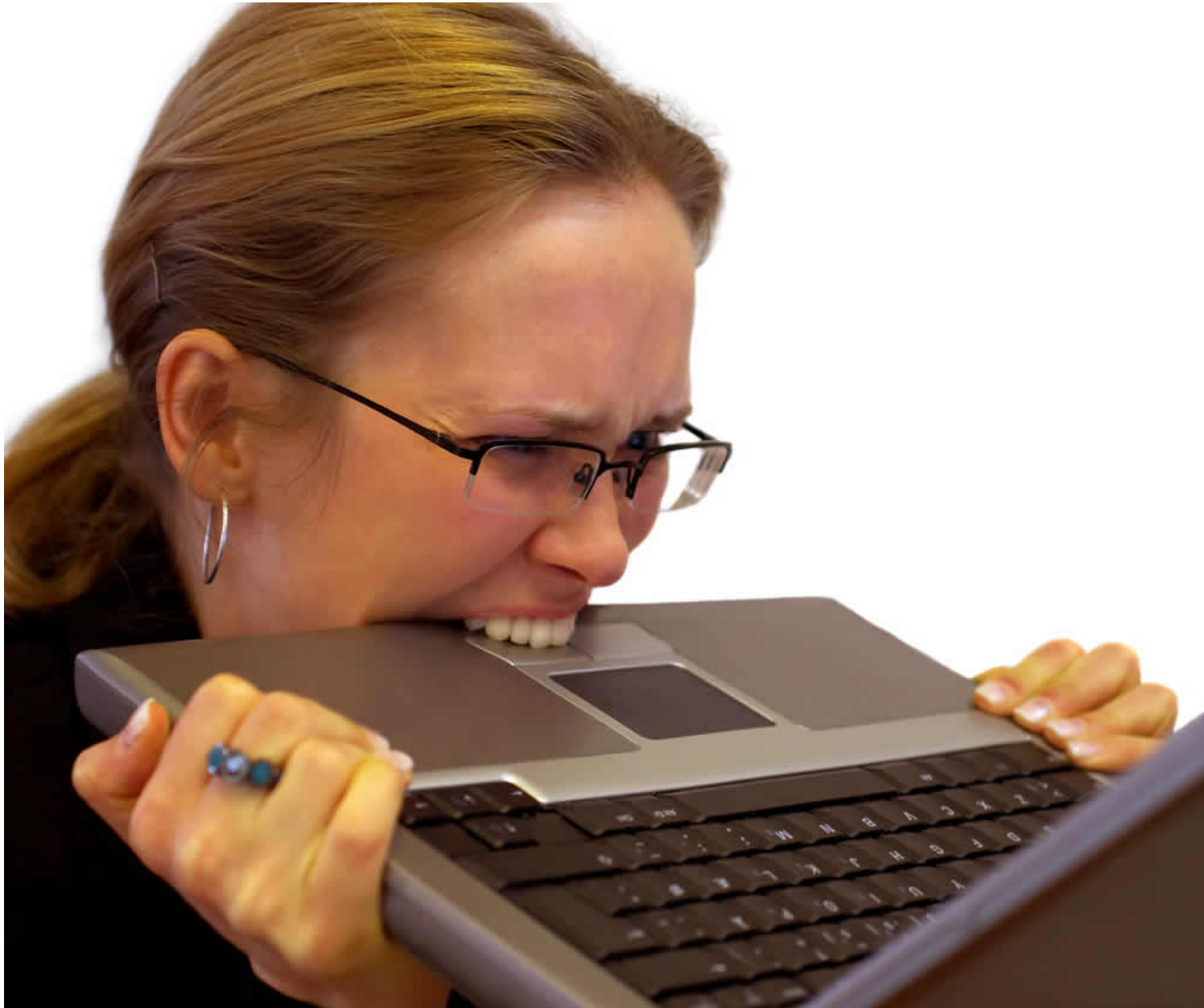
Two camps situation - IE vs. the rest

CSS2 is great, but alas some support is still missing

CSS3 is even better but full support of all major
browsers is still lacking

We deserve better-looking websites!

What is a developer to do?



Enter
"Progressive Enhancement"

Progressive Enhancement

Web 1.0 era introduced the "**Bleeding Edge**" philosophy:

We do not compromise on design

Our site has to look the same on all supported browsers

We support only browsers that support 100% of our design

We ignore everyone else

We can't sleep well at night because we're shallow, narrow-minded people

Progressive Enhancement

Web 2.0 introduced the "**Progressive Enhancement**" philosophy:

We support all platform to the best of our ability - The Web is for everyone!

We compromise on design in favour of wider support - Our design will not look the same on all platform

We implement a basic design (and functionality) that will work across the board

Progressive Enhancement

We serve enhanced (bleeding-edge) experience only to browsers that can support it

We sleep well at night knowing we stayed true to the real purpose of the Web

CSS3 - Common desired features

- Rounded corners
- Custom fonts
- Box and text shadows
- Multiple-backgrounds
- Transparencies (opacity, rgba, hsla)
- Powerful selectors

And much much more.....

Bridging the gap



Bridging the gap

Rounding the corners:

Firefox: `-moz-border-radius`

Safari: `-webkit-border-radius`

Opera / Chrome: `border-radius`

KHTML: `-khtml-border-radius`

Internet Explorer ???

Bridging the gap

Rounding the corners in Internet Explorer:

HTML:

```
<div class="rounded">
```

```
<!--[if !IE]
```

```
<u class="t1">&nbsp;</u>
```

```
<u class="tr">&nbsp;</u>
```

```
<u class="b1">&nbsp;</u>
```

```
<u class="br">&nbsp;</u>
```

```
<![endif]-->
```

```
</div>
```

Bridging the gap

Rounding the corners in Internet Explorer:

CSS:

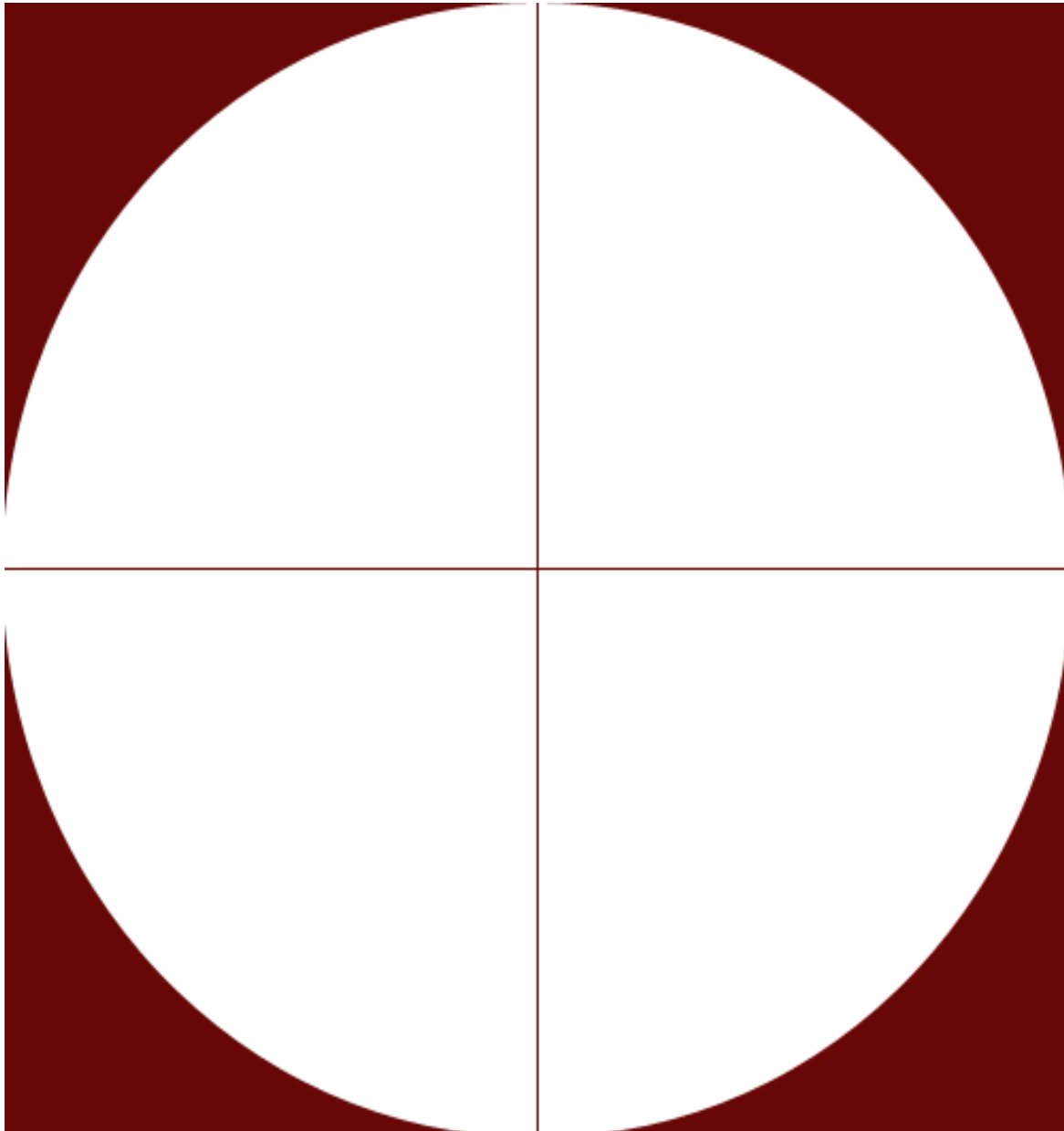
```
.rounded {  
    position: relative;  
    background: #dedede;  
}  
  
.tl, .tr, .bl, .br {  
    position: absolute;  
    background-image: url('/images/corners.png');  
    background-repeat: no-repeat;  
    width: 5px;  
    height: 5px;  
}
```

Bridging the gap

Rounding the corners in Internet Explorer:

```
.tl{  
    top:0; left:0;  
    background-position: top left;  
}  
.tr{  
    top:0; right:0;  
    background-position: top right;  
}  
.bl{  
    bottom:0; left:0;  
    background-position: bottom left;  
}  
.br{  
    bottom:0; right:0;  
    background-position: bottom right;  
}
```

Bridging the gap



Bridging the gap

Adding shadows

Firefox: `-moz-box-shadow`

Safari/Chrome: `-webkit-box-shadow`

Opera: `box-shadow`

KHTML: `-khtml-box-shadow`

Internet Explorer ???

Bridging the gap

Adding box shadows in Internet Explorer

We need to "trick" the user into thinking our box actually has something underneath that (sort of) looks like a shadow

We're limited to square boxes (round corners are possible but more complicated)

We cannot have gradually fading shadow unless we know the box's dimensions in advance

We can either use a semi-transparent background image or a simple underlay

Bridging the gap

Adding box shadows in Internet Explorer

HTML:

```
<div class="shadow">  
  <p class="shadowed">  
    Some content we want to shadow  
  </p>  
</div>
```

Bridging the gap

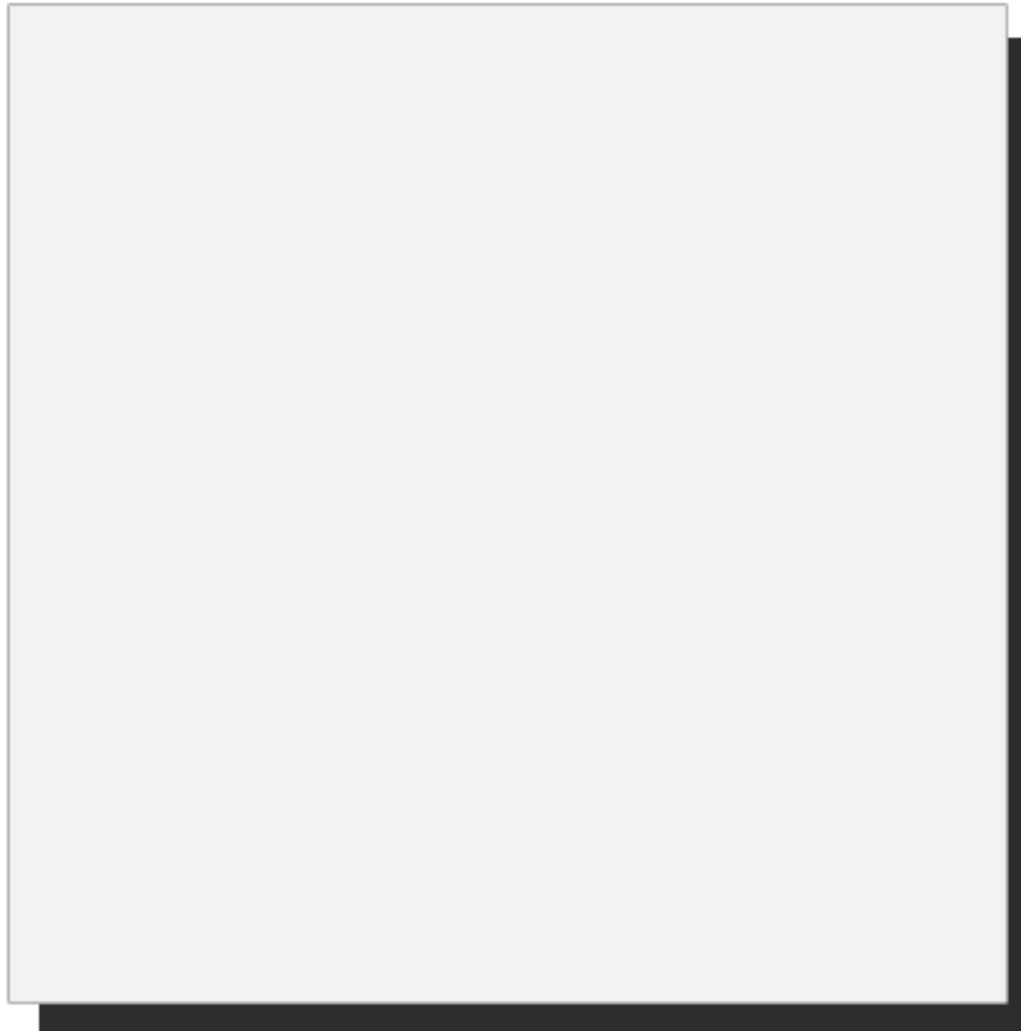
Adding box shadows in Internet Explorer

CSS:

```
.shadow {  
    position:relative;  
    background:#111;  
}  
.shadowed {  
    position:relative;  
    top:-1px;  
    left:-1px;  
    background:#fff;  
}
```

Bridging the gap

Adding box shadows in Internet Explorer



Bridging the gap

There is a faster, shorter, more effective way to deal with box shadows on Internet Explorer

DO NOTHING

Bridging the gap

Custom fonts

Custom fonts have been around for over a decade
They are originally part of CSS2

They are supported by IE since version 4
IE will only support Embedded Open-Type fonts
Other major browsers (latest versions) support
OpenType and TrueType

Bridging the gap

Custom fonts

```
/* all non-IE browsers */
```

```
@font-face {  
    font-family: "SomeFont";  
    src: url(fonts/myfont.ttf) format("truetype");  
}
```

```
/* IE */
```

```
@font-face {  
    font-family: "SomeFont-IE";  
    src: url(fonts/myfont.eot);  
}
```

Bridging the gap

Custom fonts

```
body {  
  font-family: "SomeFont-IE", "SomeFont", sans-serif;  
}
```

Bridging the gap

Powerful selectors

CSS3 gives us some very powerful selectors

Some may argue that CSS2 selectors are good enough for nearly everything

One good example is [CSS Ninja](#)'s custom checkbox / radios using CSS3 pseudo-selectors

Bridging the gap

CSS3 Please

The site CSS3please gives a perfect example of cross-browser CSS3 implementation that gracefully degrades in unsupported browsers.

see css3please.com

Final notes

□ Progressive enhancement goes hand-in-hand with graceful degradation

Avoid hacks and client-side browser detection.

Use either server-side browser detection or conditional comments to mark your `<body>` with an ID for the rendering engine

DO NOT RELY ON JAVASCRIPT - Use Javascript to enhance a fully functional page!

Final notes

Talk to your designers, convince them or fire them

Compromise on form not on function

Remember the minorities - IE6, Camino, Konqueror,
Opera

Think of mobile browsers

Avoid unjustifiable code

Thank You

Zohar Arad,

zohar@zohararad.com | <http://www.zohararad.com>