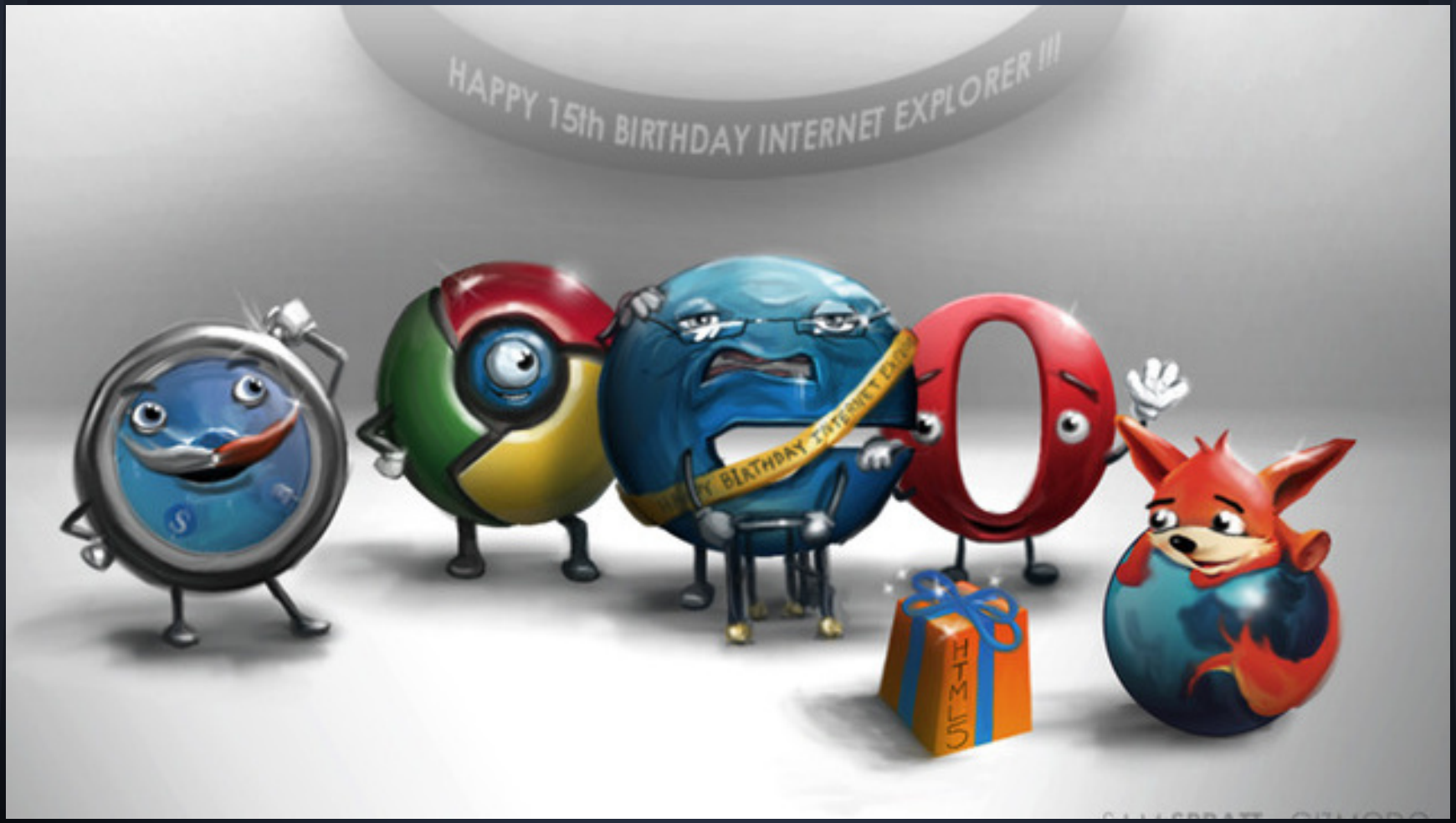
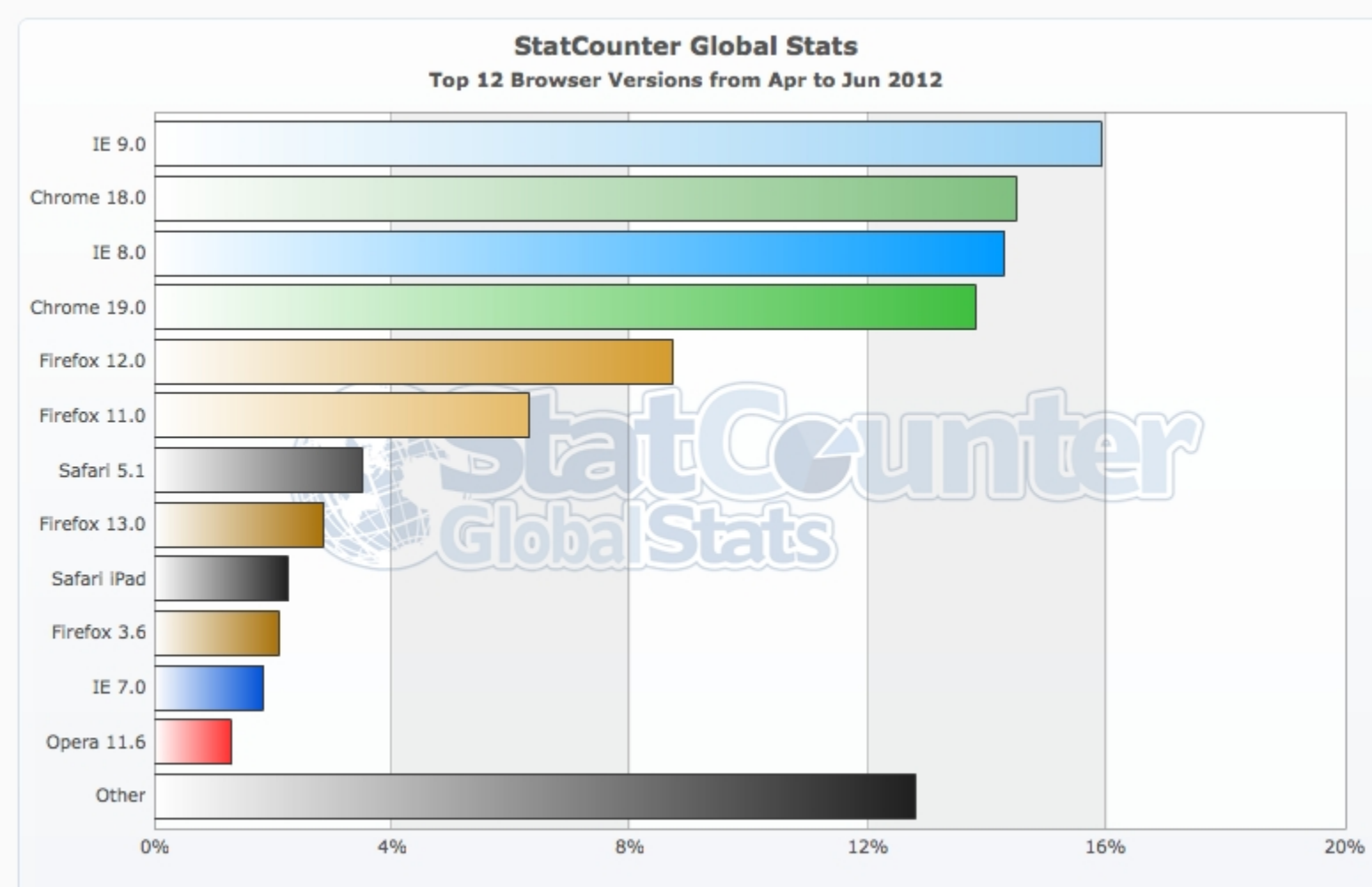
Developing for IE

Roy McElmurry

About IE



Why we care at all...



History of IE

- IE6: August 27, 2001
- IE7: October 18, 2006
- IE8: March 19, 2009
- IE9: March 14, 2011

- Google supports IE7+
- Facebook supports IE8+
- Amazon supports IE7+

Testing IE Compatibility

This is best described with pseudo-code

```
if (you_use_windows) {
   if (need_to_support <= IE6 ||
         !you_have(IE9)) {
      use IETester
   } else {
      use IE9 developer tools
   }
} else {
   throw exception("good job though")
}
```

Using IE Developer Tools

System Overview

The **Browser Mode** sets the default for how web content is handled. Developers and Users change this through:

- **F12 Developer Tools**
- **Compatibility View button**

IE tells the site its version through the **User Agent (UA) String** as defined by the Browser Mode.

The site responds with the **Document Mode** it wants to be rendered in by using the appropriate:

- **Doctype**
- **X-UA-Compatible** value

The Browser Mode determines the default Document Mode.

The page declares what features to use through:

- **Feature detection**
- **Behavior detection**
- **Conditional Comments**

IE Issues

select default value (demo)

```
<script>
    window.onload = function() {
        document.getElementById('btn').onclick =
            function() {
                var select =
                    document.getElementsByTagName('select')[0];
                alert("Value: " + select.value);
        };
    }
</script>
<select>
    <option>100</option>
    <option>200</option>
</select>
<button id="btn">Show Value</button>
```

Workaround: None, be nice to IE, it's a little slow unless you define the `value` attribute explicitly.

list-style-type (demo)

```
<!DOCTYPE html>
<html>
   <body>
      <ol style="list-style-type: lower-latin;">
         <li>Stuff</li>
         <li>Stuff</li>
         <li>Stuff</li>
      </ol>
   </body>
</html>
```

Workaround: lower-latin is lower-alpha all the way until IE8. Same with upper-latin.

favicon (demo: gif, png)

```
<!DOCTYPE html>
<html>
   <head>
      <link href="favicon.ico"
            rel="shortcut icon" />
   </head>
   <body>
      <p>Derp</p>
   </body>
</html>
```

Workaround: You must use a favicon.ico file instead (even in IE9!)

const (demo)

```
<!DOCTYPE html>
<html>
    <head>
        <script>
            const PI = 3.14;
            var radius = 10;
            alert(2 * PI * radius);
        </script>
    </head>
    <body></body>
</html>
```

Workaround: None, just use a `var` with an all caps name as the convention. (even in IE9!)

document.getElementById (demo)

```
<script>
   window.onload = function() {
      var element = document.getElementById('fake');
      if (element) {
         alert(element.nodeName);
      } else {
         alert("not found");
      }
   }
</script>
<input type="input" name="fake" />
```

getElementById Workarounds

- Use getElementsByTagName, iterate over results and check for the id you wanted

```
var elems = document.getElementsByTagName("p");
var element= null;
for (var i = 0; i < elems.length; i++) {
      if (elems[i].getAttribute("id") == "description") {
            element = elems[i];
            break;
      }
}
```

- Redefine the getElementById function manually to something that works

element.addEventListener (demo)

- addEventListener simply does not exist

```
var elem = document.getElementById("foo");
if (elem.addEventListener) {
    elem.addEventListener("click", clickHandler);
} else if (elem.attachEvent) {
    elem.attachEvent("click", clickHandler);
} else {
    elem.onclick = clickhandler;
}
```

- IE9- uses a different API, you have to check whether the function exists before calling it

event handler parameter (demo)

- Normally event info is passed to the handler, but IE8- sets a global instead

```
document.getElementById("foo").onclick = function(evt) {
    if (evt) {
        alert("event parameter exists: " + evt);
    } else if (window.event) {
        alert("event global exists: " + window.event);
    }
}
```

- You can fix this with another hack, the || operator

event.target (demo)

- The target of an event is the element that triggered the event

```
document.getElementById("foo").onclick = clickhandler;
function clickHandler(evt) {
    evt = (evt || window.event);
    alert("target: " + evt.target + ", scrElement: " + evt.srcElement);
}
```

- IE8- chooses to use the *target* property for this instead

event.which (demo)

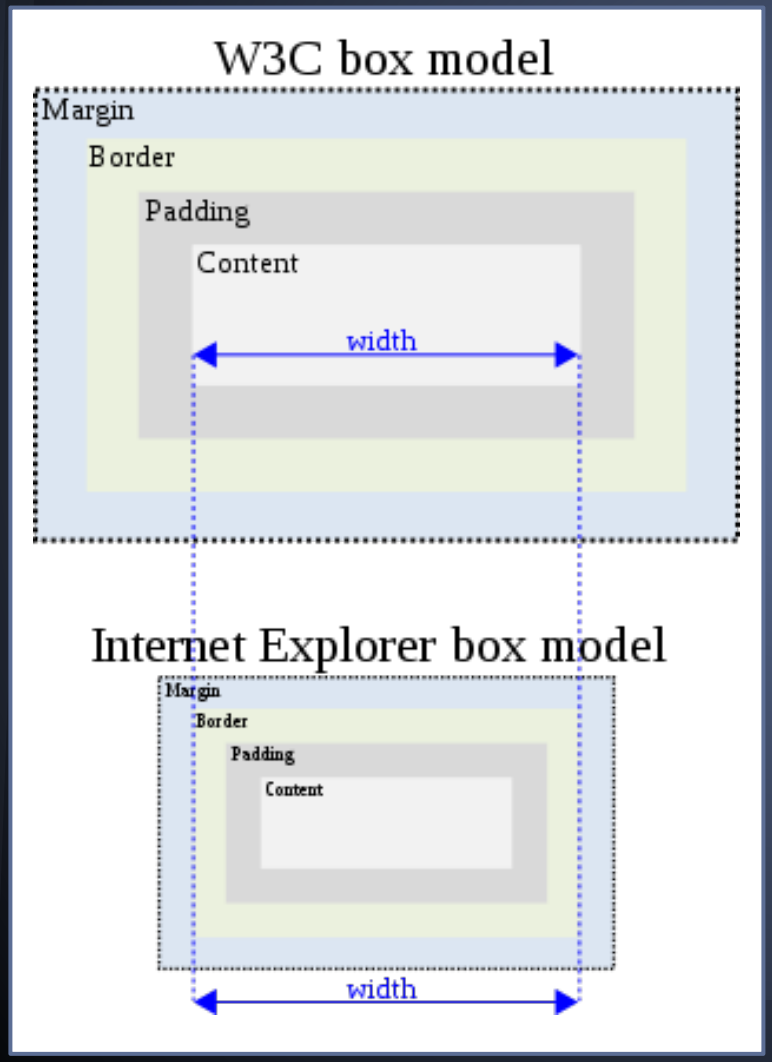
- The *which* field of an event can tell you which key or mouse button was pressed

```
document.getElementById("foo").onkeypress = clickhandler;
function clickHandler(evt) {
    evt = (evt || window.event);
    alert("event.which: " + evt.which +
        ", event.keyCode: " + evt.keyCode);
}
```

- IE8- chooses to use the *keyCode* property for this instead

IE CSS is broken in many ways

IE6 and below Box Model



Luckily this is no longer the case in IE7+,

yet...

box model bug 1 (demo)

```
<style>
    div {
        width: 600px;
        margin: auto;
        border: 1px solid red;
    }
    p {
        width: 325px;
    }
    #p1 {
        float: left;
        border: 1px solid green;
    }
    #p2 {
        float: right;
        border: 1px solid green;
    }
</style>
```

box model bug 2 (demo)

```
<style>
    div {
        width: 100px;
        height: 100px;
    }
    #red {
        background-color: red;
        overflow: hidden;
    }
    #green {
        background-color: green;
        top: 100px;
        left: 100px;
        position: relative;
    }
</style>
```

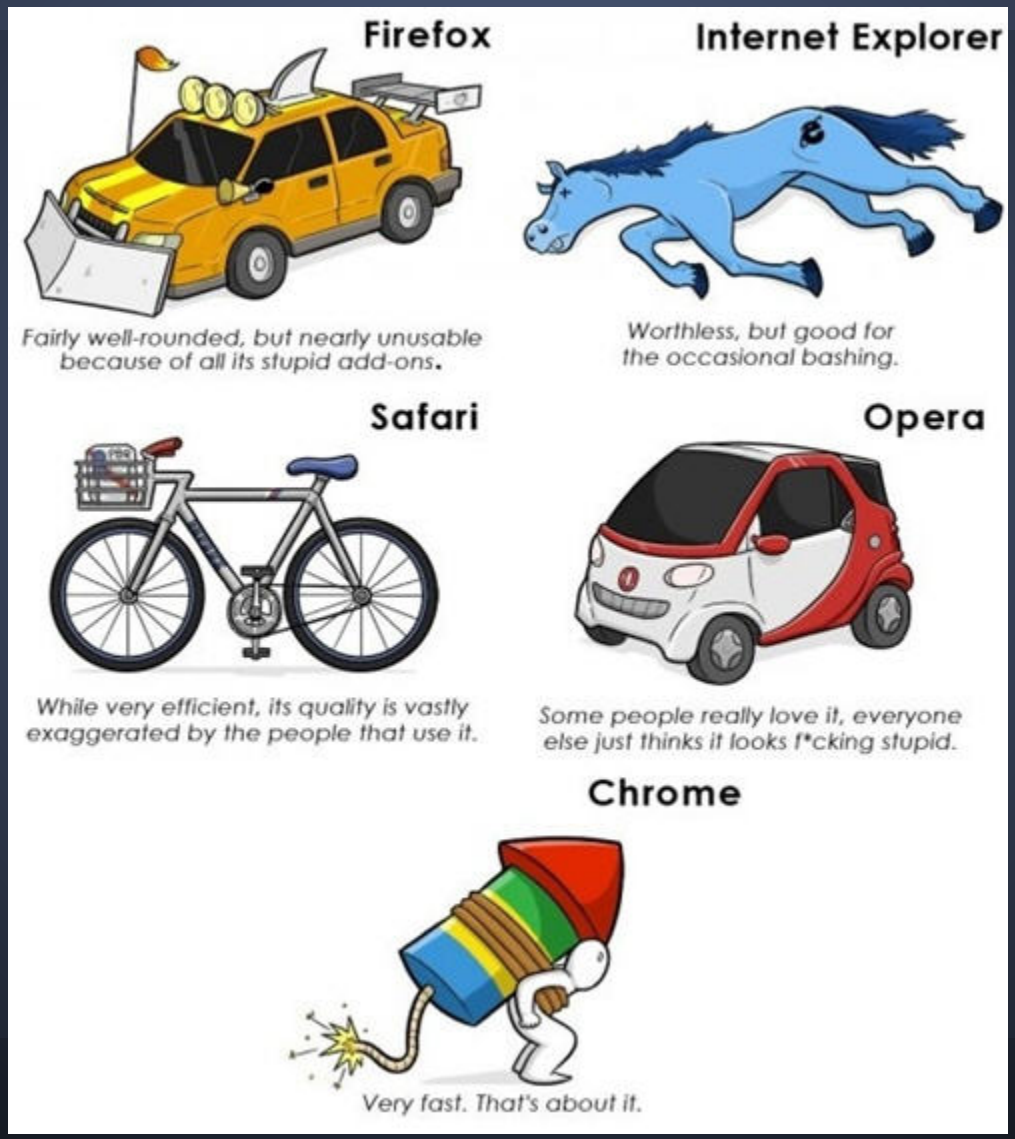
box model bug 3 (demo)

```
<style>
    div {
        width: 50%;
        height: 100px;
        background-color: green;
    }
    #floater {
        float: right;
    }
</style>
```

WTF, come on IE!!! (demo)

```
<style>
    #container { width: 400px; }
    p { margin: 64px; }
    #container a {
        border: 1px solid green;
        float: left;
        width: 100px;
    }
</style>
<div id="container">
    <p>
        <a href="#">A</a>
        <a href="#">B</a>
        <a href="#">C</a>
    </p>
</div>
```

Techniques for developing for IE



Use bugs to fix bugs (demo)

- Use CSS selectors that only IE will recognize
  - IE6-: `* html {}`
  - IE7: `*:first-child+html {}`
  - IE7+: `html>body {}`
  - More recent than IE7: `html>/**/body {}`
- Use CSS styles that only IE7- will recognize
  - `*height` (in general `*style-name`)
  - `!ie` instead of `!important`

CSS Conditional Comments (demo)

This is the preferred way to develop for IE client-side since it maintains valid CSS and HTML. Only IE will interpret these comments.

```
<head>
   <link rel="stylesheet" type="text/css"
      href="styles.css" />
   <!--[if lte IE 7]>
   <link rel="stylesheet" type="text/css"
      href="ie7hacks.css" />
   <![endif]-->
</head>
```

User-Agent Request Header

Every web request (GET or POST) comes with many headers including User-Agent. The server can return different things based on the User-Agent header of the request.

User-Agent:

- `Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0)`
- `Mozilla/5.0 (Windows NT 6.1) AppleWebKit/535.19 (KHTML, like Gecko) Chrome/18.0.1025.168 Safari/535.19`
- `Mozilla/5.0 (Windows NT 6.1; WOW64; rv:15.0) Gecko/20120427 Firefox/15.0a1`

Summary - IE makes our job hard