



Windows® Internet
Explorer 9

TechEd Reprise

TechEd 2010 IE9 for Developers session + α



マイクロソフト ディベロップメント 株式会社
プログラム マネージャ
五寶 匡郎

Internet Explorer 9 おさらい

What's Internet Explorer 9?

- 次世代 Web アプリケーション プラットフォーム

高速 - パフォーマンス向上

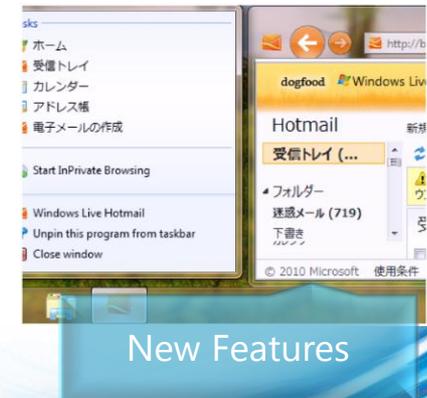
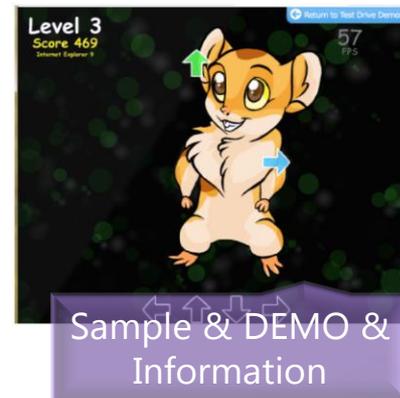
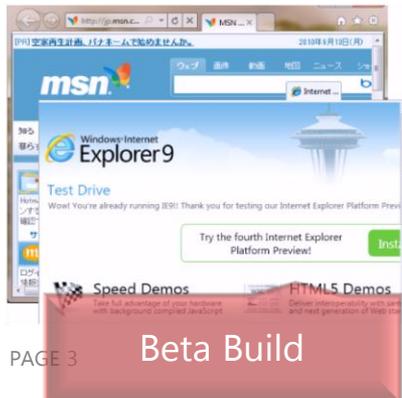
相互運用性の向上

信頼性の向上

洗練された User Experience



- 提供されるもの



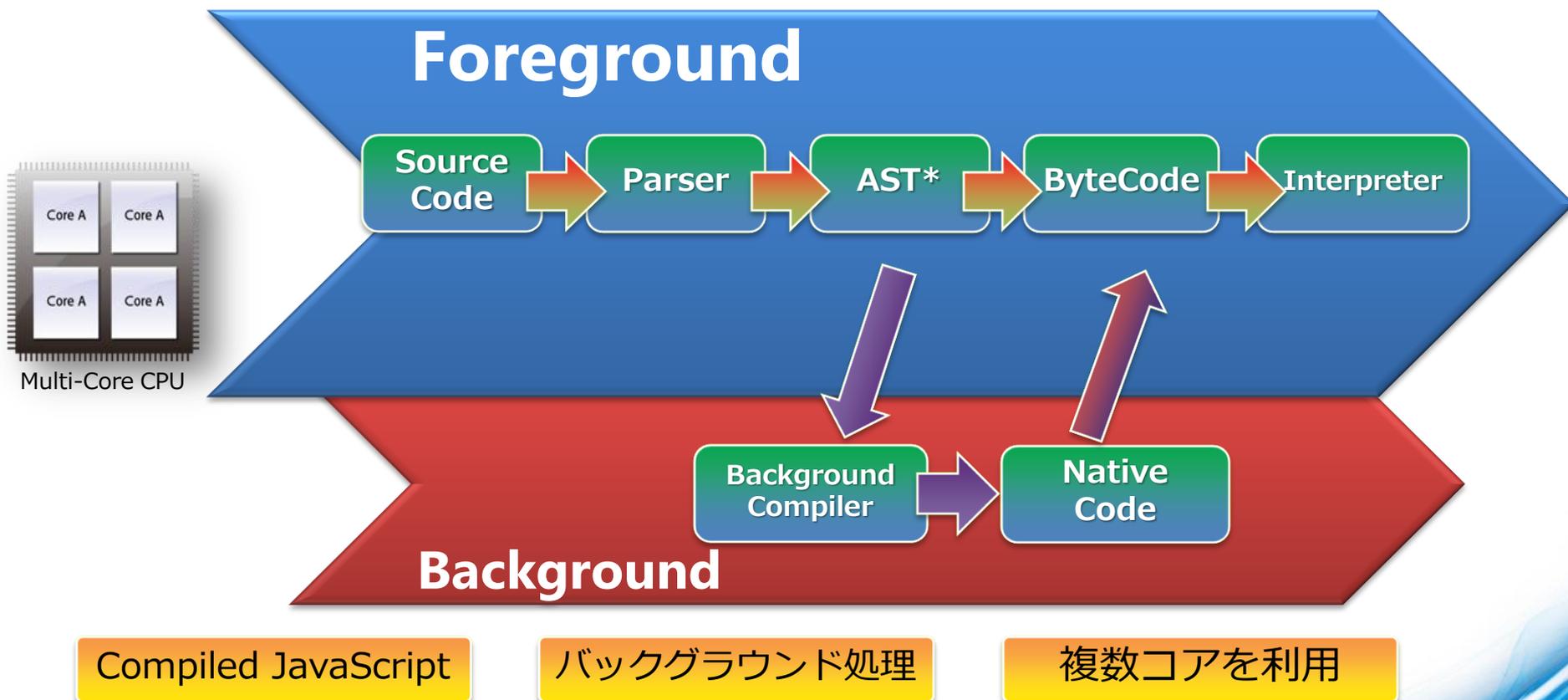
パフォーマンスの向上

**New JavaScript Engine “Chakra”
Hardware-Accelerated**

新しい JavaScript エンジン “Chakra”

Background コンパイラ

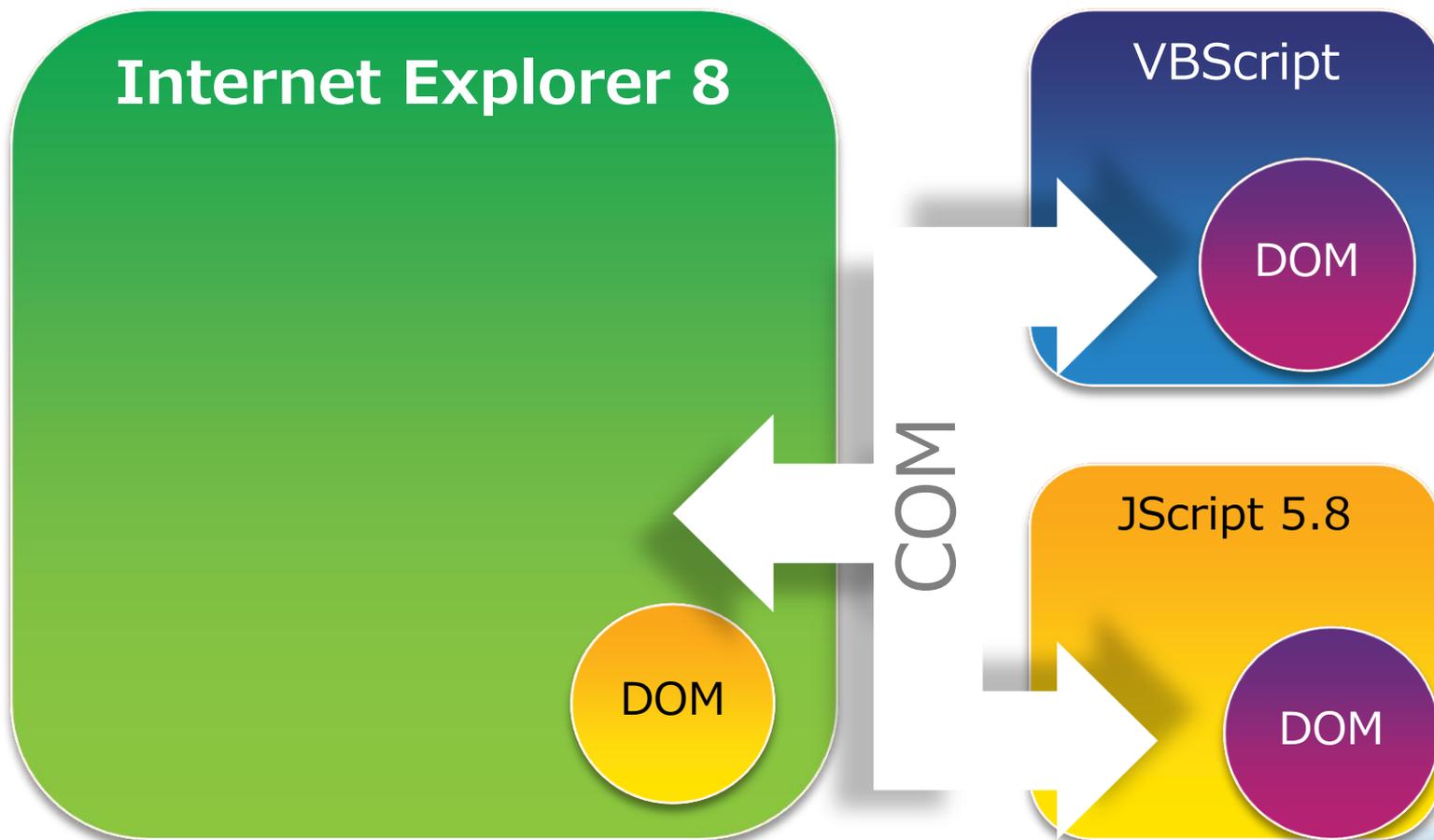
- Multi-Core CPU 処理に最適化



新しい JavaScript エンジン “Chakra”

JavaScript エンジンの Native 実装

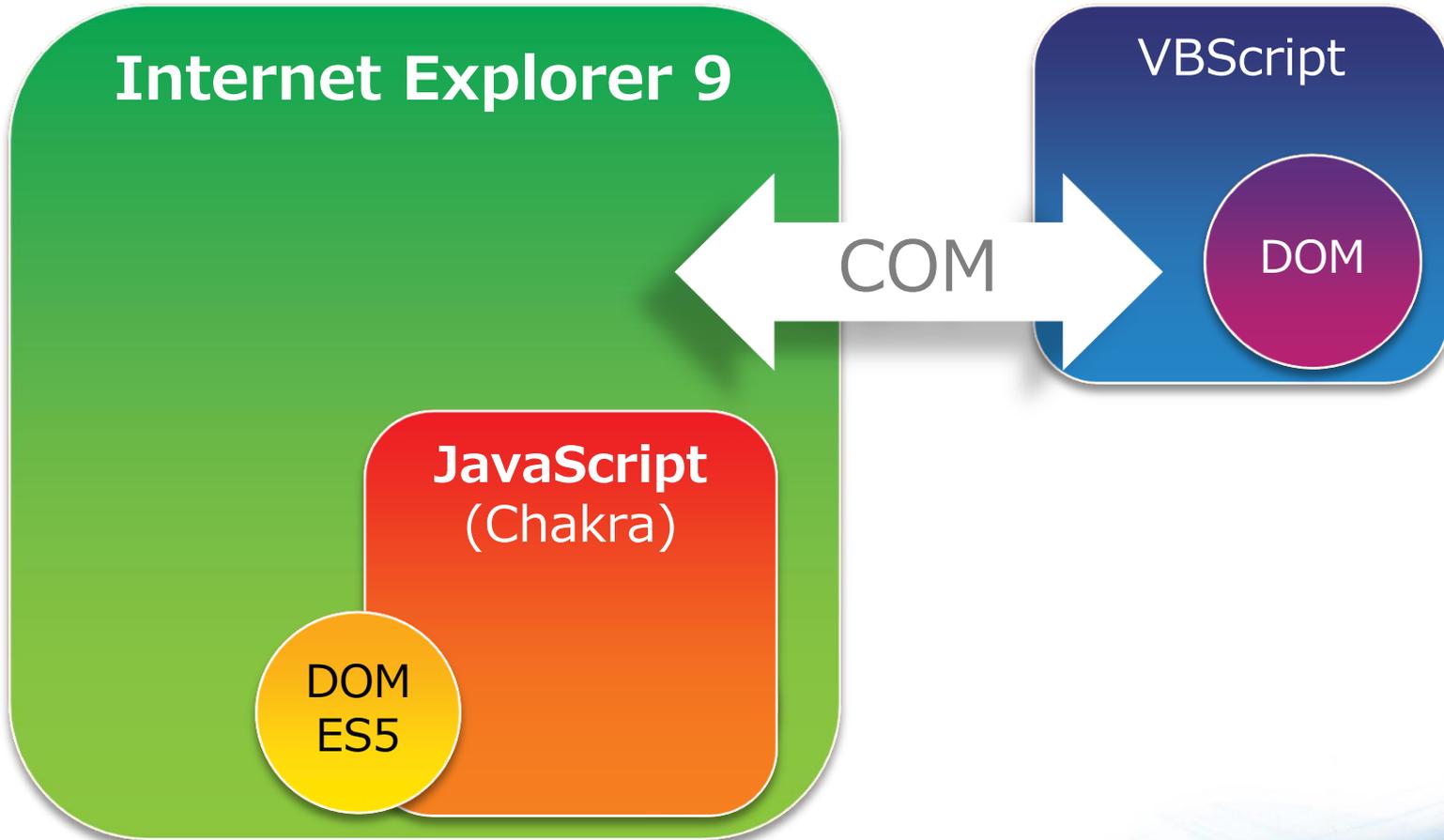
- Document Object Model (DOM) と JavaScript の最適化



新しい JavaScript エンジン “Chakra”

JavaScript エンジンの Native 実装

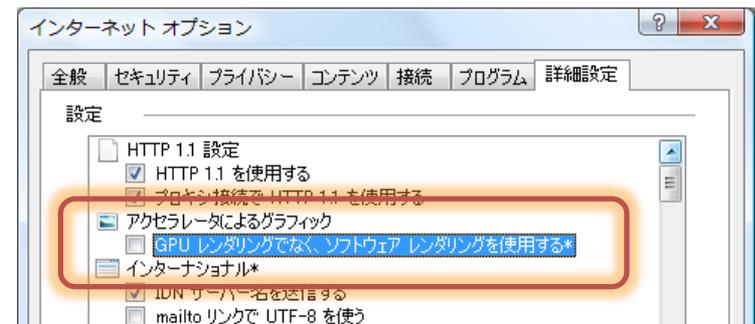
- Document Object Model (DOM) と JavaScript の最適化



Hardware-Accelerated グラフィックスとテキストレンダリング

GPU の活用

- DirectX API を利用 – Direct2D & DirectWrite
- “GPU-Powered HTML5”
 - HTML5 Video & Audio のハードウェア再生支援機能の利用
 - HTML5 Canvas や SVG の描画、CSS3 の透過処理
 - WOFF のレンダリング
 - Color Profile 処理
- ハードウェア処理が追い付けない場合は、ソフトウェア処理に
 - GDI 処理ではなく、DirectX のソフトウェア処理



GPU アクセラレーションの切り替え

- DirectX のソフトウェア処理に切り替えるレジストリ

```
[HKEY_CURRENT_USER¥Software¥Microsoft¥Internet  
Explorer¥Main¥FeatureControl¥FEATURE_FORCE_D2D_SOFTWARE]  
"iepreview.exe"=dword:00000001
```

```
[HKEY_CURRENT_USER¥Software¥Microsoft¥Internet  
Explorer¥Main¥FeatureControl¥FEATURE_FORCE_D3D_WARP_DEVICE]  
"iepreview.exe"=dword:00000001
```

* ハードウェア処理に戻すときは、dword 値を 0 にする

さらに改善された点

TechEd 2010 Plus

- **ネットワーク キャッシュ アルゴリズムの改善**
 - とても長い Long-Life キャッシュヘッダーの扱い
 - IE9 では Cache-Control: max-age が 2の63乗 second までサポートされているが、Freshness のインターバルは 2の31乗までに切り捨てられます。
 - Vary レスポンス ヘッダーの改善
 - IE9 では Vary: Accept-Encoding と Vary: Host の指示が含まれるレスポンスに対して、Web サーバー側に再検証を要求しません。今まで通り、Vary: User-Agent も無視します。
 - リダイレクト キャッシュ
 - RFC 2616 に記載されている HTTP リダイレクト レスポンスをキャッシュします。
 - HTTP キャッシュ の改善
 - 不必要なホスト間にまたがる HTTPSリクエストは Conditional リクエストとみなし、変更のないコンテンツに対してはサーバーへは HTTP / 304 Not Modified を返せずだけになる。
 - 戻る・進む ボタン動作の最適化
 - RFC2616 に準拠。無駄なキャッシュの確認作業を軽減。ユーザーが戻る・進むボタンをクリックした時のキャッシュの Freshness は基本チェックしない。

	IE8	IE9	Improvement
	Request Count: 21	Request Count: 1	Request Count: -20 (-95%)
Back/Forward Navigation	Bytes Sent: 12,475	Bytes Sent: 325	Bytes Sent: -12,150 (-97.4%)
	Bytes Recv: 216,580	Bytes Recv: 144,617	Bytes Recv:-71,963 (-33.3%)

- よく使われる API 類のパフォーマンス UP
 - document.getElementById などの処理を高速化

Web 標準対応

HTML5

CSS3

DOM L2&3

SVG

ECMA Script 5th Edition

WOFF

Color Profile & ICC v2&3

HTML5

Web 標準対応

- 文書構造の解析そのものから HTML5 準拠
 - シンプルな DOCTYPE 宣言を利用
 - application / xhtml+xml のサポート
 - Inline SVG のサポート
- Internet Explorer 9 の HTML5 対応
 - HTML5 Video & Audio elements
 - HTML5 Canvas element
 - HTML5 Selection APIs
 - XHTML 対応
 - Inline SVG 対応
- Internet Explorer 8 の HTML5 対応
 - DOM Storage, Ajax Navigation, Cross-document Messaging など

```
<! DOCTYPE html>
```

HTML5

<video> & <audio>

- 外部プラグイン なしで実装可能

```
<video id="myVideo" src="video.mp4" autoplay controls>  
</video>
```

- 再生コントロールがデフォルトで実装
- 動画や音声のデコードを GPU で処理



サポートフォーマット
video: MPEG-4/H.264 ~60fps
audio: MP3 と AAC

HTML5

Canvas

- Canvas 要素は、図などのフォーマットではなく、グラフィックスを描画する領域を示す
- 指定した範囲内で、図形などの線画、画像などの 2D Graphics を自由に描画

Canvas の座標系



HTML5

Canvas の実装

- グラフィックスの描画が可能な領域を示す

```
<canvas id="mycanvas" width="100" height="200">  
</canvas>
```

- JavaScript で Canvas に描画するための、2d 描画コンテキストを取得する

```
var canvas = document.getElementById ("mycanvas");  
var context = canvas.getContext("2d");
```

HTML5

Canvas アニメーション

- Frame-based: オブジェクト
 - Frame の Update 毎に同じ数だけ動く
 - 動作させるマシンに依存
 - 実装がシンプル
- Time-based: オブジェクト
 - 時間毎に同じピクセル数だけ動く
 - マシン依存がない



Cascading Style Sheets, Level 3 (CSS3)

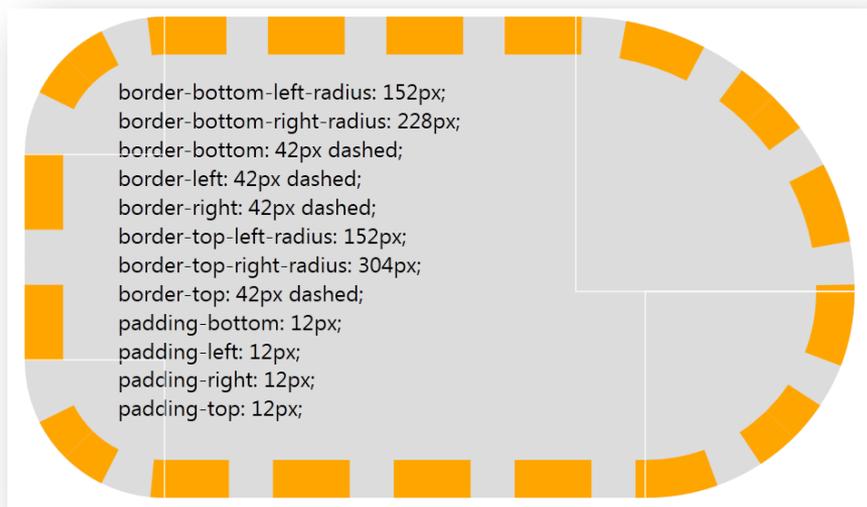
IE9 でサポートした CSS3 コンポーネント

- CSS3 Backgrounds & Borders Module
 - background-XXX や border-radius プロパティなどのサポート
- CSS3 Color Module
 - RBBA、HSL と HSLA カラー モデル、opacity プロパティ、currentColor キーワード、Beta から transparent キーワードもサポート
- CSS3 Fonts Module
 - 既存の CSS フォントも CSS3 Fonts Module に準拠。WOFF, 未処理フォント
- CSS3 Media Queries Module
 - スタイル シートから正確なデバイスの機能の設定を特定するためのメソッドを指定
- CSS3 Namespace Module
 - 規定の名前空間、プレフィックスを作成
 - SVG 要素を対象とする名前空間も宣言できる
- CSS3 Selectors Module
 - 一覧はこちら -> <http://msdn.microsoft.com/en-us/library/cc351024.aspx>
- CSS3 Values and Units Module
 - 新しい値と単位のサポート
 - attr() 関数 と calc() 関数のサポート
- CSSOM View Module
 - レイアウト ボックスの位置、ビューポートの幅、および要素のスクロールなどを含む、ドキュメントとそのコンテンツの視覚的なプロパティを扱うための API

Cascading Style Sheets, Level 3 (CSS3)

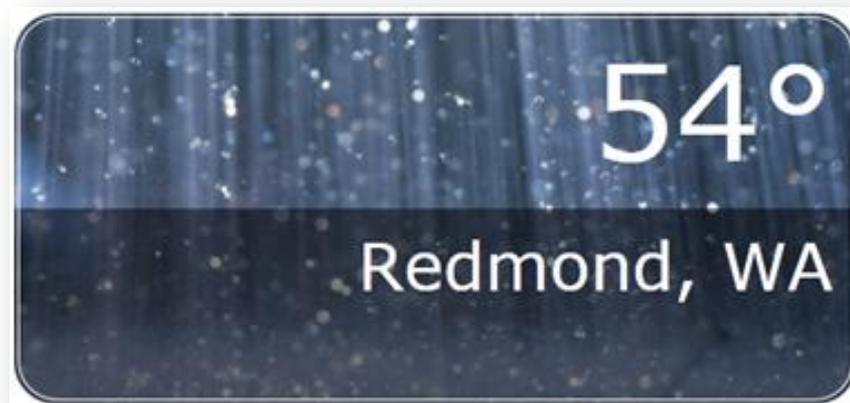
Border-radius や透過処理

- 要望が多かったモジュールやプロパティを実装
- Web デザイナーの創造の可能性が広がる



Border-radius プロパティのサポート

透過処理のコントロール



Cascading Style Sheets, Level 3 (CSS3)

CSS3 Color Module

- RGBA, HSL, HSLA color models

```
Background-color: rgba(赤, 緑, 青, alpha)
```

*Alpha: 0.0 (透明) から 1.0 (不透明) の値

- opacity プロパティ

```
<div style="background:navy; opacity:0.2;"></div>  
<div style="background:navy; opacity:40%;"></div>
```

*Opacity: 0% (透明) から 100% (不透明) の値も可

Cascading Style Sheets, Level 3 (CSS3)

CSS3 Media Queries

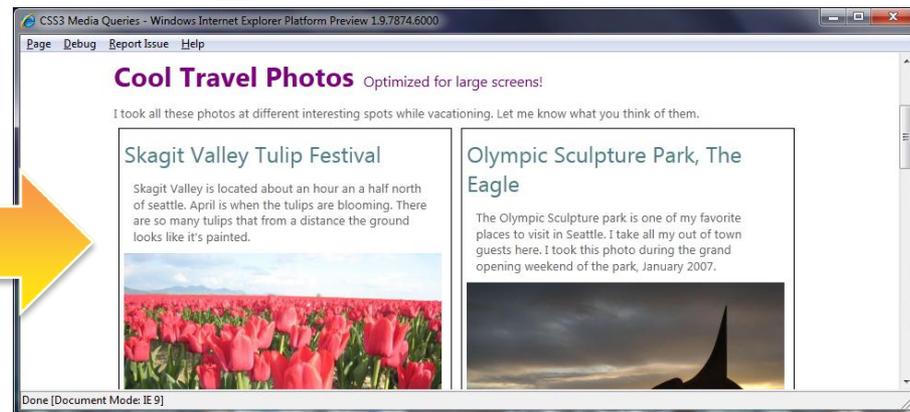
- 表示するデバイスの解像度によって、表示内容のデザインを切り替える
- サポートされるメディア関連プロパティ
 - width、height、device-width、device-height、orientation、aspect-ratio、device-aspect-ratio、color、color-index、monochrome、resolution



Mobile 向け



Netbook 向け



大画面向け

Document Object Model (DOM)

IE9 で強化された点

- DOM のエントリーポイントが “Chakra” JavaScript エンジン内へ
- “Chakra” が拡張する DOM オブジェクトと API
 - DOM オブジェクトを拡張可能 (Object.preventExtensions から実行)
 - DOM API を変更可能 (DOM API のプロパティを変更、またはプロパティ全体を削除することで実行)
 - DOM オブジェクトの JSON サポート
 - 継承によるネイティブ JavaScript オブジェクト (hasOwnProperty、toString など)
 - 継承によるネイティブ JavaScript 関数 (call、apply、bind)
- W3C DOM で定義されている機能への対応
 - W3C DOM 例外: DOM 例外は、DOM API の異常終了の結果として投げられる、新しい種類のエラー オブジェクトです。これらの例外のエラー コードは、例外自体に定義されている定数にマップされます。
 - W3C DOM “const” プロパティ (Node.ELEMENT_NODE など): const プロパティは、その他の DOM API から戻される、数多くの一般的な数値の戻り値に名前を与える単純なフィールドです。たとえば、nodeType DOM API によって戻される数値を定数と照らし合わせることで、より人間が分かりやすいコードを作成できます: `if (myElement.nodeType == Node.ELEMENT_NODE)`

Document Object Model (DOM)

Beta 版で強化された点

- DOMParser

- 文字列を解析して XML ドキュメントにする

```
var parser = new DOMParser();  
var doc = parser.parseFromString("<test>Some XML</test>",  
"text/xml");
```

- XMLSerializer

- DOM ノードを XML 文字列にシリアル化する

```
var serializer = new XMLSerializer();  
var xmlString = serializer.serializeToString(doc);
```

Document Object Model (DOM)

Event リスナーについて

- `addEventListener()` メソッドのサポート
 - 従来の `attachEvent` モデルからの解放
 - `addEventListener`, `removeEventListener`, `createEvent`, `dispatchEvent` が利用可能

```
addEventListener(type, listener, useCapture)
```

- `useCapture` でキャプチャ/ターゲット フェーズでのイベントの補足が可能
- `type` には DOM Level 0 で使われていた "on" プレフィックスは必要ない
 - 例 `onclick` -> `click`、`onmouseover` -> `mouseover`

Document Object Model (DOM)

Level 2 & 3 のサポート

- DOM L2 Events
 - MouseEvent (mouseenter/mouseleave)
- DOM L3 Events
 - KeyboardEvent (keydown/keypress/keyup)
 - Composition events
(compositionstart, compositionupdate,
compositionend)

Document Object Model (DOM)

DOM ホワイトスペースの扱い

- スペース、タブ、改行もテキスト ノード
- ホワイトスペースの存在に影響されない手法で要素を特定しましょう

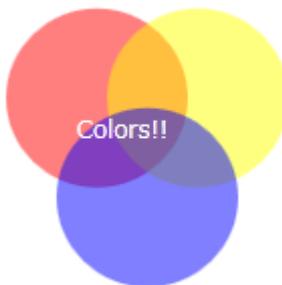
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    Text - Empty Text Node
    <title>
      Text - Internet Explorer 9: Platform Demos
    Text - Empty Text Node
    <meta http-equiv="X-UA-Compatible" content="IE=9"/>
    Text - Empty Text Node
    <!-- Thanks for checking out this site. The purpose of these demos is to
    Text - Empty Text Node
    <link href="includes/css/testdrive.css" rel="Stylesheet" type="text/css"/>
    Text - Empty Text Node
    <!-- Loading this script here because it adds styles based on the browser an
    Text - Empty Text Node
    <script src="includes/script/testdrive.js" type="text/javascript"></script>
```

Scalable Vector Graphics (SVG)

IE9 の SVG 対応

- W3C SVG 1.1 2nd Edition をベースに実装
- 中身は XML で、JavaScript などから簡単にアクセス可能
- Inline HTML、Inline XHTML に対応
- CSS で指定する画像 (background など)
- <object>, <embed>, <iframe>, で実装可能

```
<svg width="200" height="100">  
  <circle cx="50" cy="50" r="45" fill-opacity=".5" fill="red"/>  
  <circle cx="100" cy="50" r="45" fill-opacity=".5" fill="yellow"/>  
  <circle cx="75" cy="100" r="45" fill-opacity=".5" fill="blue"/>  
  <text x="40" y="70" fill="white">Colors!!</text>  
</svg>
```

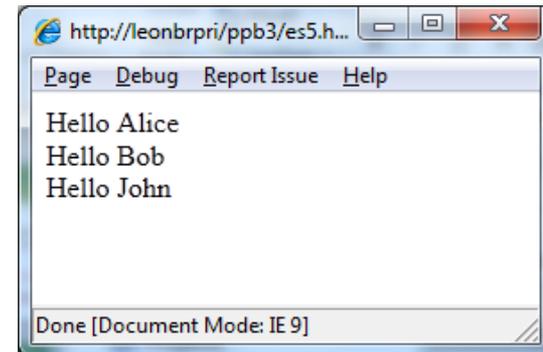


ECMA Script 5th Edition

- IE9 Standard Document モードで利用
 - DOM とダイレクトにコミュニケーション
 - DOM との連携が ES5 に最適化
- IE9 で追加された新しい ES5 機能
 - 新しい配列用のメソッドの実装
 - オブジェクト モデルの拡張
 - その他の Computational メソッドと機能

ES5 配列 Method のサンプル

```
<body>
<script language="javascript" type="text/javascript">
  var myArray = ["Alice", "Bob", "John"];
  var myFunc = function(value, index, myArray){
    document.write("Hello" + value + "<br/>");
  }
  var result=myArray.forEach(myFunc);
</script>
</body>
```



Web Open Font Format (WOFF)

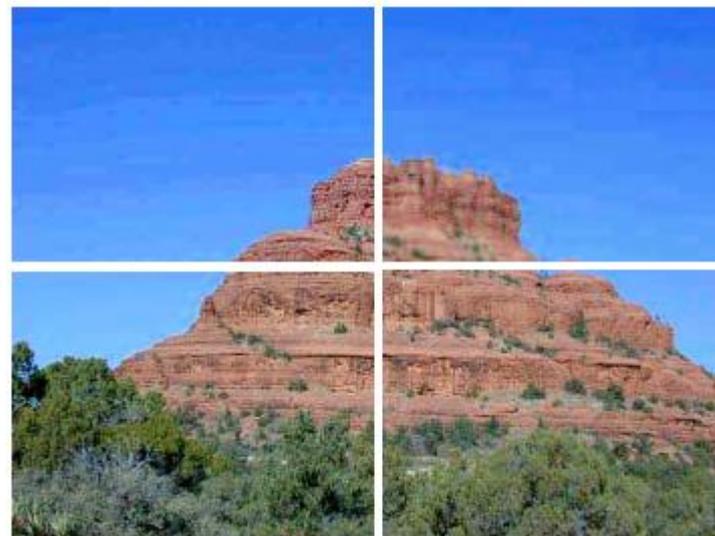
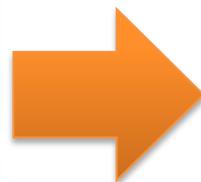
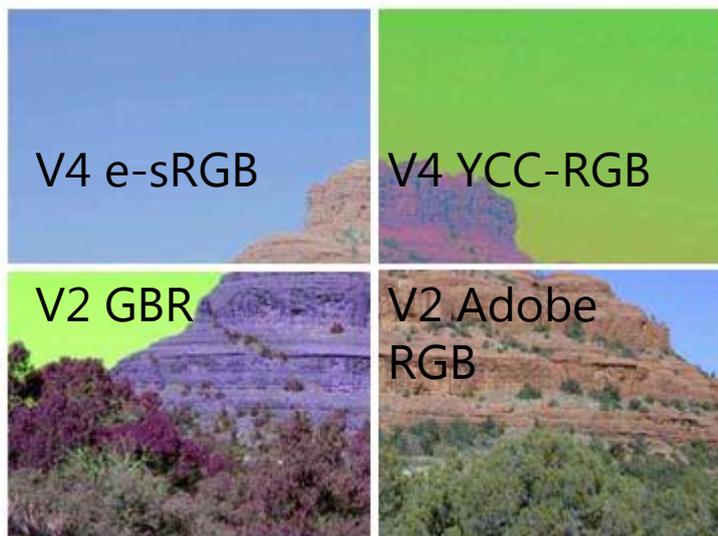
- Web 専用のフォントフォーマットをサポート
- CSS3 @font-face 宣言で利用
 - 従来からサポートしていた EOT や Raw TrueType に続いてのサポート

```
<style type="text/css">
  @font-face {
    font-family:MyFontName;
    src: url('FontFile.woff');
  }
</style>
<div style="font: 24pt MyFontName, sans-serif;">
  This will render using MyFontName in FontFile.woff
</div>
```

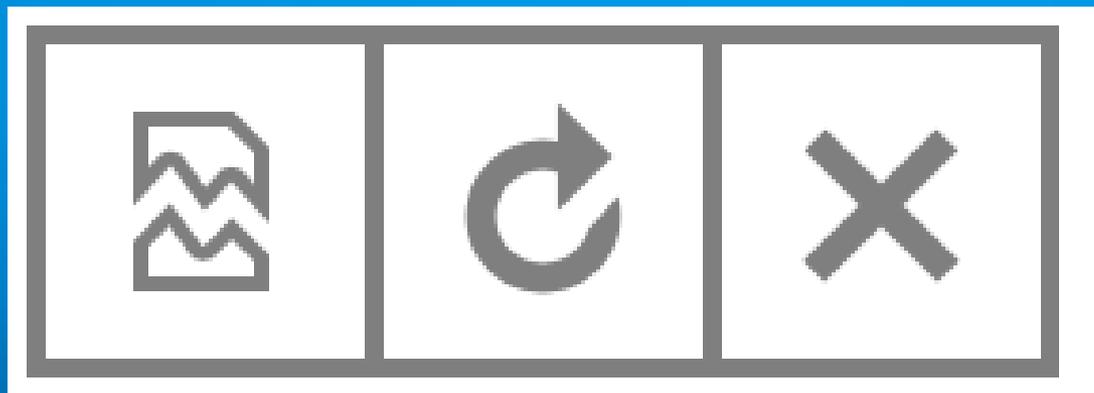
Color Profile & ICC v2&3

Web 標準対応

- イメージのデコードは、Windows Imaging Component (WIC) を利用
- 新しくサポート
 - International Color Consortium (ICC) v2 と v4 カラープロファイル
 - TIFF
 - JPEG XR 画像



互換性と対応



ブラウザー モードとドキュメント モード

互換性と対応

- ブラウザー モード
 - User Agent String で指定される
 - 開発者向けに 4 つのモードを用意
 - (F12 Developer Tools で変更可能)
 - IE9 利用者は互換ボタンで IE9 互換表示に切替
 - 互換モード: UA string、version vector ならびに document mode が IE7 として動作する。
- ドキュメント モード
 - 新しい IE9 Standard Mode が追加
 - Trident が自動的に DOCTYPE などでモードを判断する

Meta タグと HTTP ヘッダー

互換性と対応

Meta タグ	<code><meta http-equiv="X-UA-Compatible" content="IE=9"></code>
HTTP ヘッダー	名前: X-UA-Compatible 値: IE=EmulateIE9
優先順位	DOCTYPE < HTTP レスポンスヘッダー < meta タグ

■ IE9 で追加された値

content = "IE=_____"	概要
EmulateIE9	doctype 宣言を使い、ドキュメント モードを決定 <ul style="list-style-type: none">• doctype または Quirks のどちらの記述もない場合は、Quirks モード• その他すべての doctype では、IE9 Standard モード
IE9	Doctype に関係なく、強制的に IE9 Standard モード

User-Agent String

互換性と対応

- デフォルトでは他のモダンブラウザ同様の短い String

```
Mozilla/5.0 (compatible; MSIE 9.0; Windows  
NT 6.1; Trident/5.0)
```

- IE9 互換モードの User-Agent
 - IE7 Standard モードの扱いとなる
 - Trident のバージョンで IE9 と判断
 - 拡張部分の UA は、navigator.userAgent で取得できる

```
Mozilla/4.0 (compatible; MSIE 7.0; Windows  
NT 6.1; Trident/5.0)
```

互換モードにおける変更点・注意点

- Web ページが IE7 互換を求める場合
 - EmulateIE を並べて記述することで、IE8, IE9 両方に対応

```
<meta http-equiv="X-UA-Compatible"  
content="IE=EmulateIE7; IE=EmulateIE9">
```

- iframe の扱い
 - トップレベルのページが IE9 Standard モードの場合、すべての iframe ページは IE9 Standard モードで表示される
 - ただし、Quirks モードを除く

Same Markup & Cross-Browser 対応

多くの異なる種類のブラウザ

```
graph TD; A[多くの異なる種類のブラウザ] --> B[多くの異なるバージョン]; B --> C[新しいバージョンの頻繁な Update];
```

多くの異なるバージョン

新しいバージョンの頻繁な Update

Cross-Browser 対応

Detection 方法の遷移

```
if( condition )  
{  
    // 優先するコード  
  
}  
else  
{  
    // 補助的なコード  
  
}
```

Cross-Browser 対応

バージョンで分岐

```
if( navigator.userAgent.indexOf( 'MSIE' ) != -1 )
{
    // ブラウザー X 用に書かれたコード
}
else
{
    // ブラウザー Y 用に書かれたコード
}
```

Cross-Browser 対応

オブジェクトの有無で分岐

```
if( document.all )
{
    // ブラウザー X 用に書かれたコード
}
else
{
    // ブラウザー Y 用に書かれたコード
}
```

Cross-Browser 対応

Feature で分岐

```
if( window.addEventListener )
{
    // addEventListener をサポートした
    // ブラウザー用のコード
}
else
{
    // addEventListener をサポートしていない
    // ブラウザー用のコード
}
```

Same Markup のコア ガイドライン

- 推奨
 - 機能の検出: 機能を利用する前にブラウザがサポートしているか?
 - 振る舞いの検出: 代替案を適用前に既知の問題をテストする
- 非推奨
 - 特定のブラウザの検出: 過去のやり方
 - 関係のない機能を想定しない: 実際に使用する機能かどうか?

```
getComputedStyle: function(el, property) {  
  if (window[GET_COMPUTED_STYLE]) {  
    return el[OWNER_DOCUMENT][DEFAULT_VIEW]  
      [GET_COMPUTED_STYLE](el, null)[property];
```

```
  } else if (el[CURRENT_STYLE]) {  
    return Y.Dom.IE_ComputedStyle.get  
      (el, property);  
  }  
},
```

Standards Mode

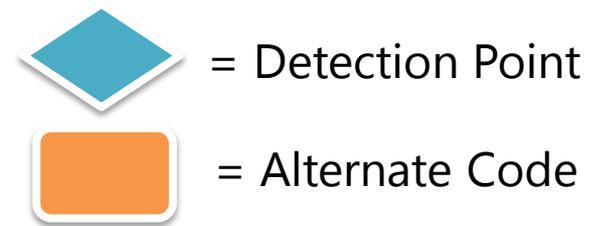


Legacy IE modes

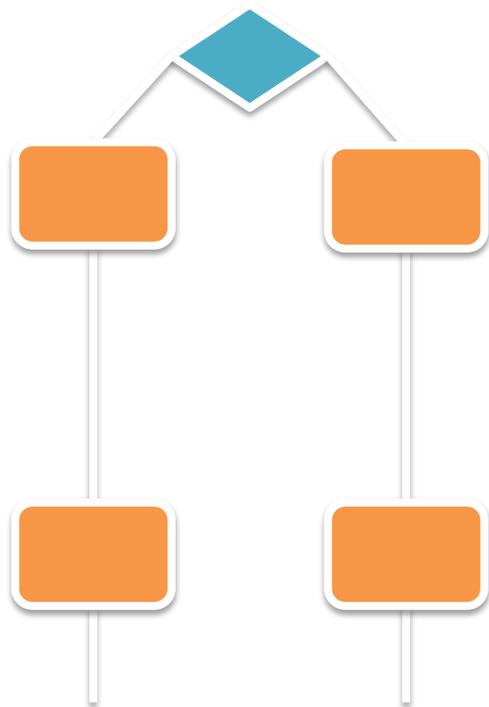


Code Path の比較

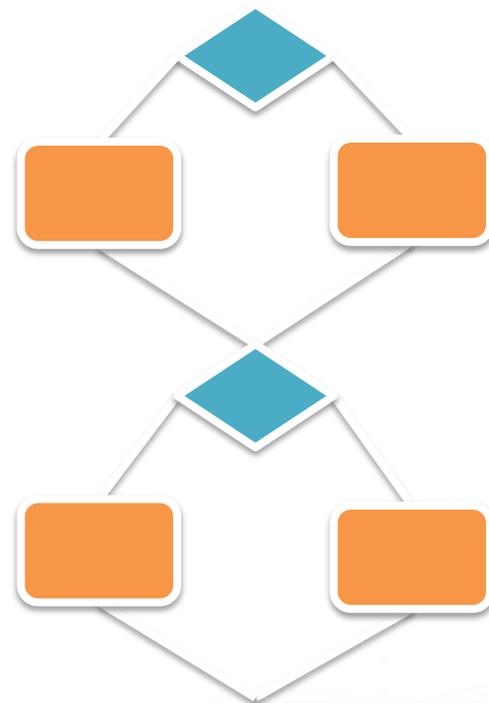
コードをどのように分岐させるか？



バージョンでの分岐



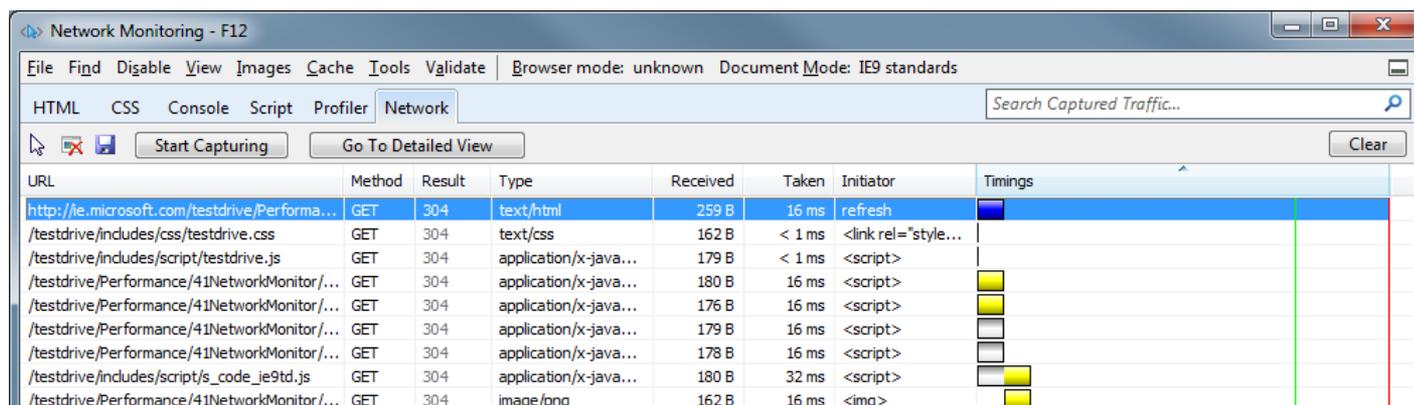
機能毎の分岐



Built in Tools

F12 開発者ツール

- Web 開発者向けのデバッガー
 - IE8 から Integrate されて、F12 キーで起動
- Visual なインタフェースを提供
- 素早い評価・検証を実現
 - 変更した点がすぐに反映される
- **IE 9 での新機能**
 - Network 対応 (Fiddler のサブセットのような)
 - JavaScript のプロファイリング
 - 高速動作



信頼性の向上

XSS Filter の向上
ASLR の活用

信頼性の向上

Internet Explorer 8 からさらに向上

- アドオン パフォーマンス アドバイザー
 - IE8 のアドオンの管理からさらに進化
- 自動クラッシュ回復機能の向上
 - Tear-off タブ、Windows Aero Snap などへの対応
 - アドオン パフォーマンス アドバイザーとの連携
- XSS Filter
 - IE8 の時は、実世界に存在する問題 (Cross Site Scripting 攻撃) に対応
 - IE9 では、IE8 リリースの後に見つかった XSS Filter の弱点や脅威を修正
- メモリ保護機能 (DEP/NX) の進化
 - ASLR サポートの向上
- SmartScreen フィルター機能と連携したダウンロード マネージャー
- MIME Sniffing 設定の追加
- アクセシビリティにおけるシステム サウンドの再生
- 印刷設定の “背景の色とイメージを印刷する” のページ設定画面への統合
- 混在したコンテンツを持つセキュアでないイメージファイルのブロック

Microsoft®
tech.ed
Japan | 2010

AUGUST 25-27, 2010 | PACIFICO, YOKOHAMA

TechEd Reprise まとめ



Internet Explorer 9 の特長まとめ

Web を Windows ネイティブアプリケーションのように



高速

PC のハードウェア
リソースを最大活用

- グラフィックとテキストの表示を GPU で高速化
- 新しい JavaScript エンジン



洗練

主役はブラウザーではなく
Web サイト

- Web を際立たせる UI
- Windows 7 に最適化
- 新しいタブブラウジング
- ワンボックス
- 操作を邪魔しない通知



信頼

安全性、安定性、プライバシーの向上

- SmartScreen による保護を備えたダウンロードマネージャー
- ハングアップ耐性
- アドオンのパフォーマンスモニタリングツール



相互運用

同一のマークアップの
実現

- HTML 5 をはじめとする次世代 Web 標準規格に準拠
- W3C のワーキンググループに積極的に参加
- 数多くのテストキットを提供



Thank you.

© 2010 Microsoft Corporation. All rights reserved. Microsoft, Windows, Internet Explorer and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation.
MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.