

JavaScript Performance Considerations



- **Performance has a higher priority on the client**
- **Less/No control of client machine configuration**
- **JavaScript is interpreted each time the page loads**
- **Too much being pushed to the client side**
- **Browsers being pushed to their limits**

Steps To Improve Performance



- 1. Write less code**
- 2. Minimize perceived load times**
- 3. Increase perceived responsiveness**
- 4. Cache back-end responses**
- 5. Play to the browsers strengths**
- 6. Be as specific as possible in DOM selection**
- 7. Render initial page as statically as possible**
- 8. Beware of aliases**
- 9. Profile often**

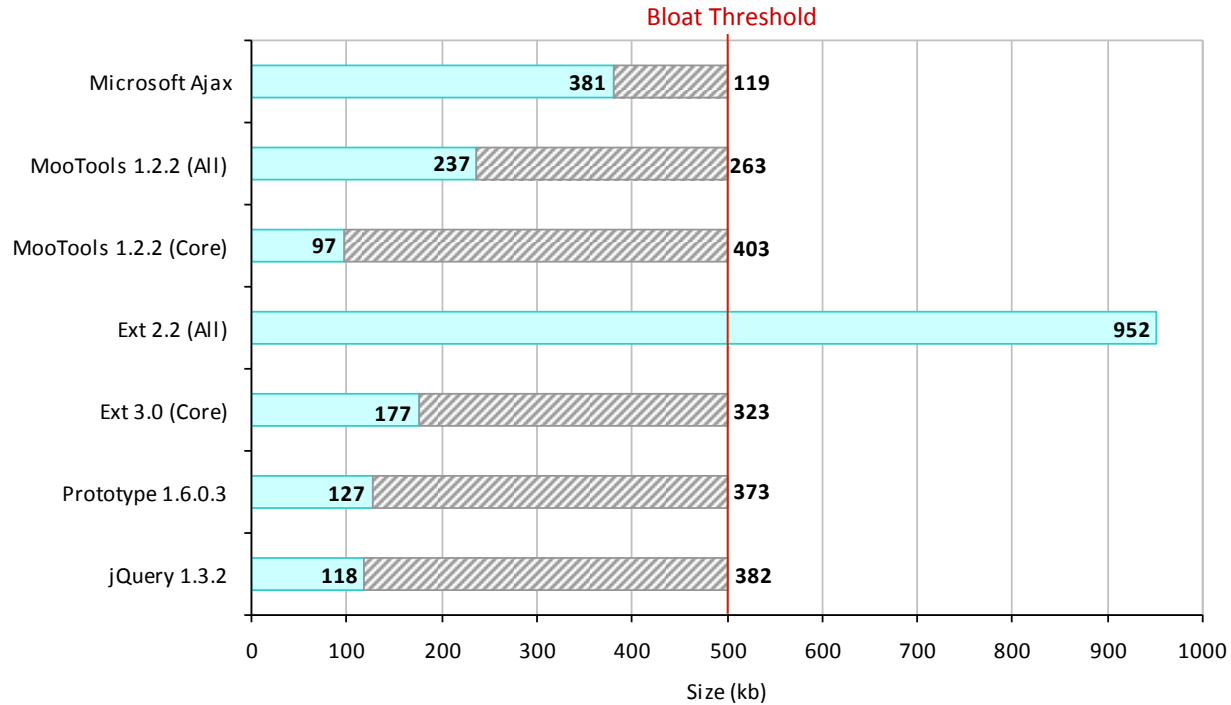
Write Less Code



- More code means more to download and execute
- Initial parsing of JavaScript is a major bottleneck
- Don't rely on browser caching to excuse code bloat
- Strip debugging code and OO boilerplate
- Minimize dependency on 3rd Party Frameworks

< 500kb of JavaScript (uncompressed)

Write Less Code



Balance 3rd party offerings with custom code

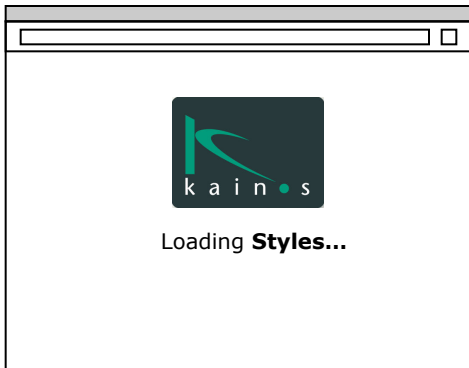
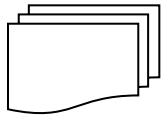
Minimize Perceived Load Times



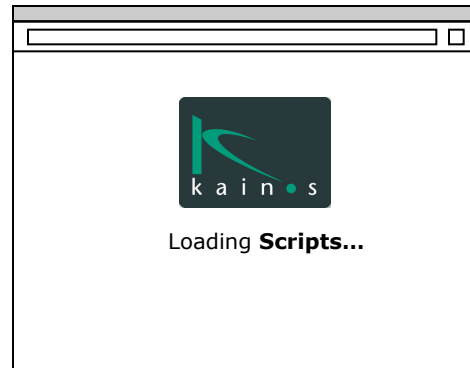
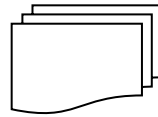
- **Minimize JavaScript and CSS sent from server**
- **Place CSS at the top of the page**
- **Place JavaScript at the bottom of the page**
- **Complex screens should be masked while loading**
- **Offer constant feedback**
- **Load lazily or “on demand”**

Minimize Perceived Load Times

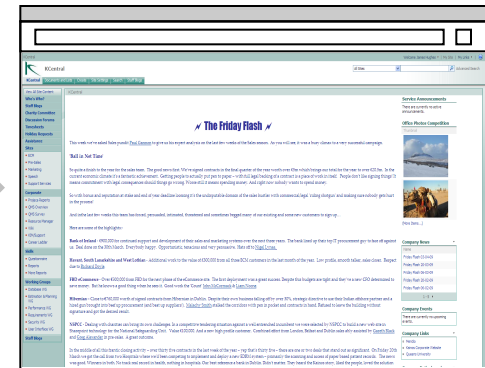
Initial styles loaded



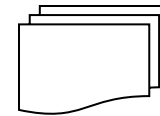
Initial scripts loaded



Loading screen removed



Further scripts loaded on demand



Minimize Perceived Load Times

```
<html>
  <head></head>
  <body>
    <div id="msg">Loading 1st Library</div>

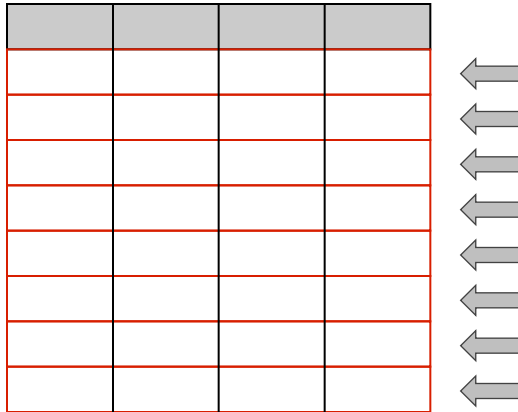
    <script src="firstLibrary.js"></script>
    <script>
      var msg = document.getElementById( 'msg' );
      msg.innerHTML = "Loading Second Library";
    </script>
    <script src="secondLibrary.js"></script>
    <script>
      msg.parentNode.removeChild(msg);
    </script>
  </body>
</html>
```

Increase Perceived Responsiveness



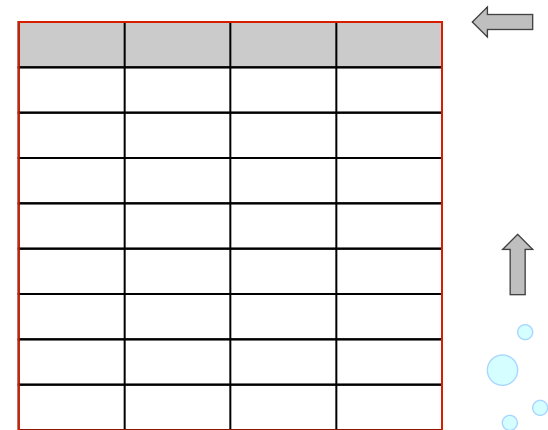
- **Do as little work as possible**
- **Use `setTimeout()` or Chunking to allow UI updates**
- **Use “sooner” events**
- **Bind directly to inline events instead of event listeners**
- **Delegate multi element events**

Increase Perceived Responsiveness



Per element

```
tr1.onclick = clickHandler;  
tr2.onclick = clickHandler;  
...  
...  
trN.onclick = clickHandler;
```



Delegation

```
table.onclick = function(e){  
  if(e.target.nodeName == "tr"){  
    clickHandler(e);  
  }  
}
```

Cache Back End Responses



- **Data requests should go through a Data Manager**
- **Cache frequent static responses**
- **Manage cache using HEAD requests**
- **Use range caches for large data sets (autocomplete)**
- **Balance local updates with refetching**

Cache Back End Responses

Type your county (for the demo):

Ne

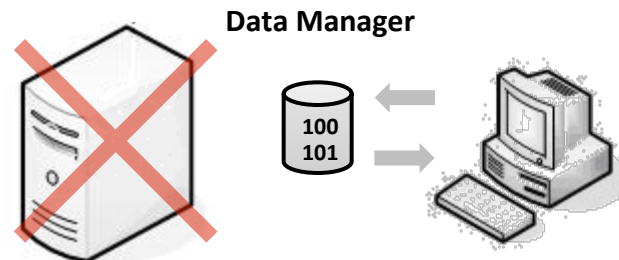
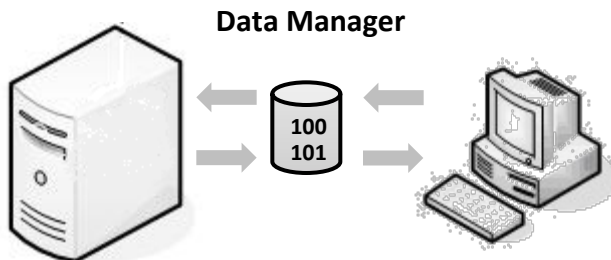
- Nepal
- Netherlands
- Netherlands Antilles
- New Caledonia
- New Zealand



Type your county (for the demo):

Neth

- Netherlands
- Netherlands Antilles



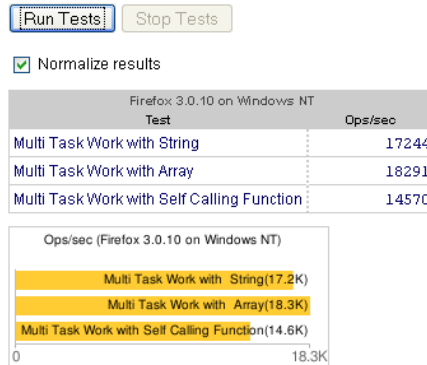
Play To The Browsers Strengths



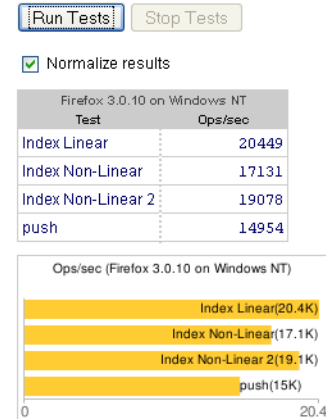
- **Avoid DOM manipulation; use innerHTML where appropriate**
- **Reduce dynamic CSS definitions in JavaScript**
- **Avoid reflow**
- **Do DOM work off-DOM and insert at end**
- **Be aware of the best performing techniques**
- **Use JSON over XML where possible**

Play To The Browsers Strengths

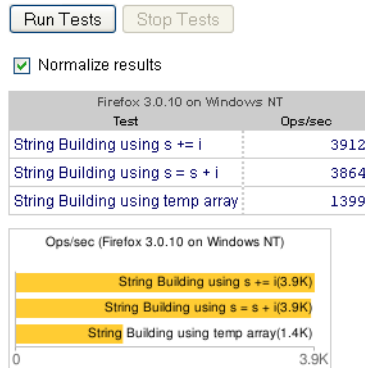
Apply Multiple Functions to Objects



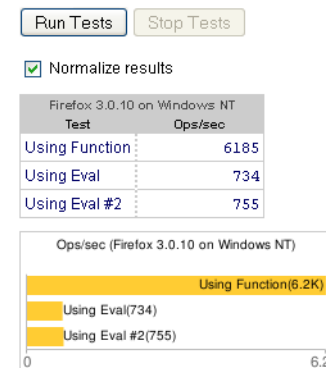
Array Expansion



String Building



Function Constructor vs Eval



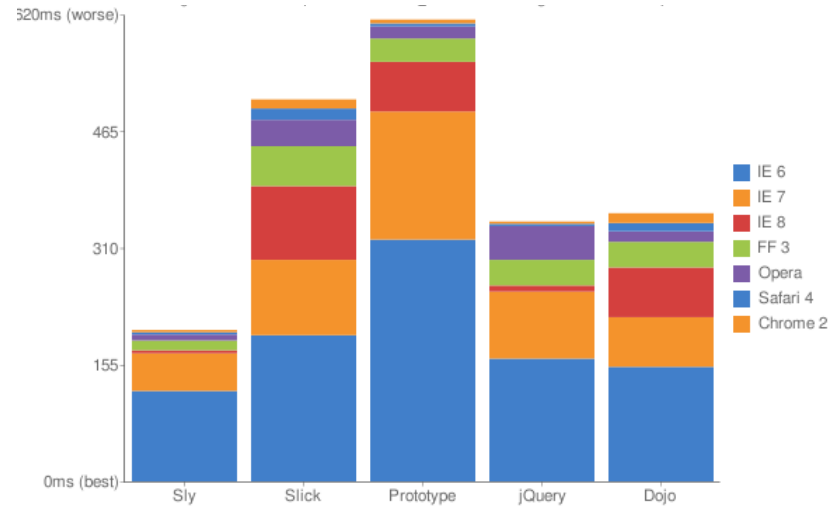
Be Specific



- Be specific as possible when working with the DOM
- Use ID's getElementByID where possible
- Use getElementsByTagName and selectorQueryAll if available
- Use a fast 3rd party selector engine for complex queries – Sizzle, Peppy, LLamaLabs Selector, Sly, base2, YUI Selector

But be aware of browser quirks!

Be Specific



Render Initial Page Statically



- **Don't be tempted to start with a blank page and use JavaScript to build page**
- **Put as much static content on the page as possible**
- **Reduces number of server round trips**
- **Set initial page state statically (visibility etc.)**
- **Embed smaller data as JSON or XML**

Beware of Aliases



```
var $ = document.getElementById;
function checkEl(id){
    if($(id) && ! $(id).checked){
        $(id).checked = true;
    }
}

checkEl("someslist");
```

- **Results in 3 calls to DOM methods**
- **Common mistake when using 3rd party libraries such as Prototype and jQuery**

Profile Often



- **Profile often; use Firebug, IE8 Developer Tools, Safari & Chrome Developer Tools, Opera's Dragonfly, Timestamp diffs or just comment out code to identify bottlenecks**
- **Measure in a consistent environment, restart browser, clear cache etc.**
- **Take averages of runs to normalise results**
- **Define responsiveness as early as possible**

Conclusion



- **Do as much on the server as possible**
- **Avoid pushing needless tasks to the client**
- **Prioritise performance on the client**
- **Review and profile often**

Know your enemy!

Questions?

