High performance Widgets: Combine your JavaScripts and CSS in external Files

Introduction

This article is the number two of the series, **High performance Widgets**, and continues what we started in High Performance Widgets: CSS Sprites

JavaScript and CSS are indispensable in web pages, and front-end engineers all around the world study the best way to choose how to display them. The objective of this article is to show, why it's important to apply them for WRT Widgets and Mobile Websites.

This one is a little less pratice and more theory.

How display my JavaScript and CSS

There are three main ways to display your JavaScript or CSS.

- Include it in the head of your XHTML, like this:

```
<!-- @autor: Felipe Rodrigues -->
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title></title>

    <script src="jquery-1.3.2.min.js" type="text/javascript"></script>

    <style type="text/css">
        #container h1 {
            text-indent: -999px;
            width: 170px;
            height: 50px;
            float: left;
            margin: 20px 0 0 20px;
        }

        #container #menu {
            width: 140px;
            height: 55px;
            float: left;
            margin: 20px 0 0 30px;
        }

    </style>

    <script type=text/javascript>
        $(function(){

            $("#btn-url").click(function(){
                $("#url").show();
                $("#read").hide();
                $("#btn-url a").addClass("hover");
                $("#btn-read a").removeClass("hover");

            });

            $("#btn-read").click(function(){
```


(C) Copyright Nokia 2014. All rights reserved.

```
                    $("#url").hide();
                    $("#read").show();

                    $("#btn-read a").addClass("hover");
                    $("#btn-url a").removeClass("hover");
                });
            });
        </script>

    </head>
```

- Using external files

```
    <!-- @autor: Felipe Rodrigues -->
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

        <title></title>

            <script src="jquery-1.3.2.min.js" type="text/javascript"></script>


            <link href="css/screen.css" rel="stylesheet" type="text/css"
media="screen" />

    </head>
```

- Using in-line code into the html tag

```
    <!-- @autor: Felipe Rodrigues -->
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

        <title></title>

            <script src="jquery-1.3.2.min.js" type="text/javascript"></script>


            <link href="css/screen.css" rel="stylesheet" type="text/css"
media="screen" />

    </head>
<body>
<p align=center font=Arial> This line is center aligned and with font as arial </p>
// inline CSS example
<input type=text name=txtname id=txtname onfocus="javascript:alert(this.value);" />
// inline javascript
</body
```

The basic difference among all of above 3 codes is their priority to execute. The inline code is at first priority, the saperate style defined with <style> tag is on second priority and the external css file is at last priority. So when your code is on hybrid style one must be careful to see the priority of implimentation.

(C) Copyright Nokia 2014. All rights reserved.

Which is the best way to do it?

In general the embed or inline solutions are better, because there's no need to make any other file. But, the external files are a more organized and faster solution if you are going to load two or more screens, because the external files are cached, while the inline have to be downloaded everytime the XHTML document is requested.

The main care you need to take, is not modularize too much your code, breaking it in several different files. That will make you lose a lot on your widget performance. As we saw before, which new file you use, is a new HTTP Request made. Reducing the number of components reduces the number of HTTP requests required to render.

What this article suggests is that you combine all your archives of the same type together an modularize it internally.

For Example:

Instead of using different files for different view modes, screens, or even parts of your widget or site, use just one for the StyleSheets and other for the JavaScripts.

Easy to maintain easy to re-use

To make your code more maintainable and easy to re-use, is better if you keep it with a clean markup, commented and indented, making it easier for you, or any other person that may need to see it, to find what you want.

Put the CSS in the head

According to the Performance team of Yahoo! research, StyleSheets in the head of the document, gives the user the impression that the page are loading faster. That happens because of the progressive render.

In this order, the browser shows the Styles faster, because it will be one of the first things it will requests. That will give the user a visual feedback, which will improve a lot in the User Experience of your widget.

JavaScript at the bottom

While loading scripts, nothing more is downloaded, cause the browsers cannot load more than two components in parallel, so putting them on the bottom, will make them be the last thing requested, making the UX even better.

References

The references used in this article are:

- Best Practices for Speeding Up Your Web Site
- High performance Web Sites, O'Reilly Media, 2007, Steve Souders

See also

- High performance Widgets: CSS Sprites
- High performance Widgets: Optimize your JavaScript

(C) Copyright Nokia 2014. All rights reserved.

 (C) Copyright Nokia 2014. All rights reserved.