**Date**: 13/02/2013
**Procedure:** How to force the browser to reload CSS and JavaScript files
**Source: LINK**
**Permalink: LINK**
**Created by:** HeelpBook Staff
**Document Version:** 1.0

# HOW TO FORCE THE BROWSER TO RELOAD CSS AND JAVASCRIPT FILES

When editing the **CSS** and **JavaScript** of a site, there is a slight chance that the user may get the old versions, which at best will cause a lot of complains and at worst will cost you a lot of lost customers. The reason for this is that browsers cache static content like **CSS** and **JavaScript**. However, telling a browser to reload them is not that hard to do. All it takes is a little trick that will **force** our friendly browsers into thinking that the CSS and JavaScript are dynamic content.

Before we go on, let us take a look at how we specify the CSS and JavaScript to be used within **HTML**. Normally we have something in the line of:

```
<link rel="StyleSheet" type="text/css" href="/static/style.css">


<script src="/static/js/script.js" type="text/javascript"></script>
```

Where **style.css** and **script.js** are our files. Naturally CSS files are specified with the **href** attribute of the **link** tag, while JavaScript files are included using the **script** tag. Now, let's take a look how we can make our browser reload CSS and JavaScript **every time**:

```
<link rel="StyleSheet" type="text/css" href="/static/style.css?version=15">
<script src="/static/js/script.js?version=15" type="text/javascript"></script>
```

"Now what is that "version"?", you are probably thinking. Well it's **nothing**. No really, it does absolutely nothing. You can use **?favouritesite=onlinehowto.net15** if you want. It would have the absolutely same effect.
"But what does it do?", you are probably thinking. "This looks just like passing GET variables." Exactly!

The browser cache uses a pretty complicated mechanism, but there is one thing that you should know for sure **index.html?version=14** and **index.html?version=15** are two different pages for the browsers, therefore it can't rely on its cache, therefore it must get them again.

But what happens on the server side? Well those arguments <u>are ignored</u> by our server and it simply supplies **style.css** and **script.js**.

Whatever language, technology, etc. you are using, you always keep a **header** file which is included pretty much everywhere so changing the **version=<number>** part after each revision is pretty easy. You can even make it a part of your site's logic.

In conclusion, this trick may save you a lot of headaches related to cached CSS and JavaScript, and if you believe you may have some issues, we **greatly** encourage you to use it!