

# jQuery: useful tips to work with images

There are a couple of things to bear in mind when working with images in jQuery. The first thing (and the most important one) is that images are replaced elements. A replaced element has its actual content stored outside the current document. In the case of images, the actual content must be fetched by a browser through a GET request to the URL provided by the `src` attribute. This implies another important thing: images, just as other media elements, often delay the final loading of a web page. In other words, the DOM `DOMContentLoaded` event is not fired until all images have finished loading. Let's see other important details.

## Forcing images to reload

Consider the following snippet:

```
$(img).attr('src', $(img)
.attr('src') + '?' + Math.random() );
```

The above code forces a browser to reload a given image by appending to its URL a random string. This code works because browsers consider a different URL as a new GET request to be run.

## Preloading images

Consider the following code:

```
function preload() {

    var urls = ['img/1.jpg', 'img/2.jpg',
'img/3.jpg'],
```

```

        len = urls.length,
        i,
        d = document;

    for(i = 0; i < len; i++) {

        var img = d.createElement('img');
        var url = urls[i];
        img.src = url;

    }
}

$(window).load(function() {
    preload();
});

```

The above function simply preloads images by looping through an array of image URLs. We use core DOM methods to create image elements in the browser's memory because this kind of methods are considerably faster than jQuery's methods.

## Image loading

If your images take a while before finishing loading, you can use the `load()` event to generate some visual effects indicating that your content is still loading:

```

$('img').load(function() {
    $('#spinner').show();
});

```

If there's an error with image loading, you can use the `error()` event to handle this problem:

```
$('#img').each(function() {  
  
    $(this).error(function() {  
  
        $(this).replaceWith('');  
  
    });  
  
});
```

You can also keep track of how many images are actually loaded:

```
var totalimages = $('#img').length;  
var loadedimages = 0;  
  
$('#img').load(function() {  
    ++loadedimages;  
    if(loadedimages == totalimages){  
        // All images are loaded  
    }  
});
```