

Demo 1 - Retinafy can be used to check if the user is running on a retina display.

The plugin will return true or false based on what display you have. I'm also using \$.retinafy(false) to prevent the plugin from retinafying the entire page in this instance, for the sake of the demo.

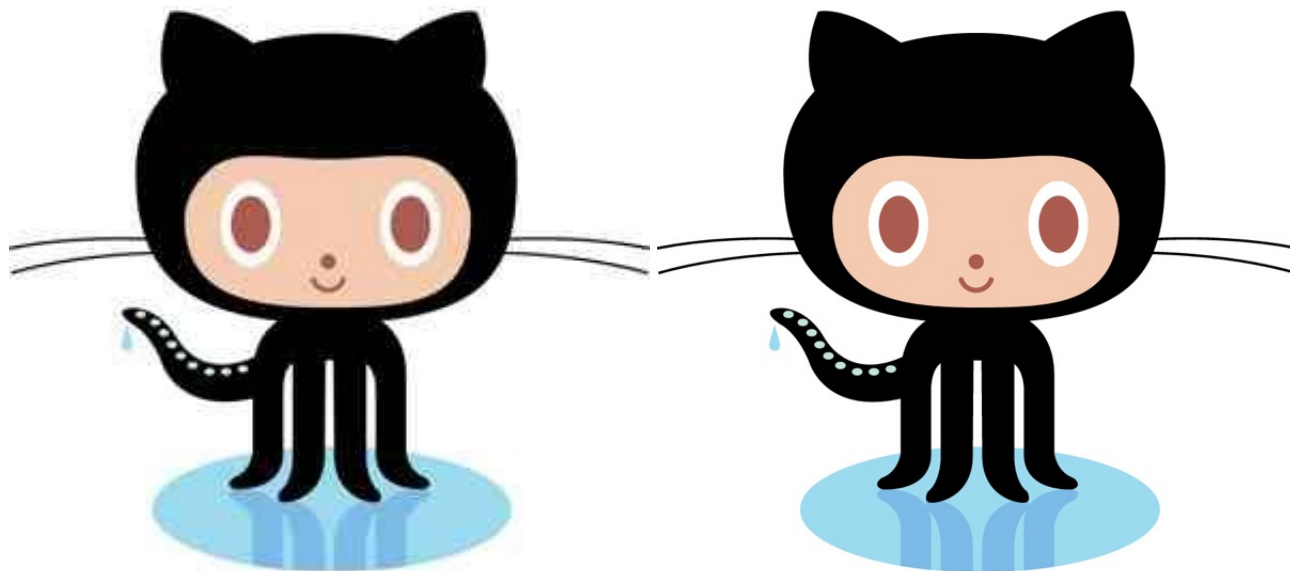
```
$('#detect input').click(function(){  
    if ($('#html').retinafy(null)) {  
        alert(' You have a retina display! ' )  
    } else {  
        alert(' You DONT have a retina display. :( ' )  
    }  
});
```

Do I have a retina display?

Demo 2 - Retinafy a section of the document.

If I have two images, I can tell the plugin to retinafy one of them.

```
$('#img.two').retinafy();
```



Demo 3 - Retinafy while hovering over part of the document.

This shows of the jQuery portion of Retinafy, which allows for easy retinification after specific events.

```
$(window).load(function(){
    $('img.three').hover(function(){
        $(this).retinafy();
    },function(){
        src = $(this).attr('src')
        $(this).attr('src',src.replace(/@2x/, ''))
    });
});
```



Demo 4 - Retinafy an element's background image.

Retinafy images set with the CSS background-image property or its shorthand kin.

```
$('#backgrounds div').retinafy()
```

Add to to the includes list.

Due to restrictions of servers, Retinafy can't always detect if files exist across different domains. This is why Retinafy has an include list, to allow all files that match any part of a specific url. Keep in mind that since the inclusion list is the second variable, you must set the first to true in order to add to it. The following code allows all files with 'octocat' in the path or file name to be included, even if the server is distant and can't be reached.

```
$('#inclusion img').retinafy(true,'octocat')
```

