jQuery plugins: how to write and document your CSS styles

Believe it or not, one of the most frequently asked questions related to jQuery plugins concerns the customization of their layout. Developers tend to focus on the plugin's code without worrying too much about the default CSS styles attached to their plugins. Since the very first language I did learn is CSS, here are some best practices which I think you may find useful.

Avoid the css() method

jQuery allows us to define CSS styles via the `css()` method. It's a very powerful method but its major drawback is that all your styles are hardcoded within your plugin. It's very difficult for an end-user to modify such styles.

You should use this method sparingly and only when it's absolutely required (e.g. when you have to perform some dynamic calculations).

Be generic

Your CSS styles should provide only the formatting required by your plugin to work properly. In short, you should write only the following styles:

- Floating and positioning
- Box model properties
- Visibility properties
- Colors and backgrounds

Instead, avoid the following:

- Font properties
- Text properties

Bear in mind that your plugin will be included in a web site whose default layout is totally unknown to you.

Be responsive

You should avoid fixed measures and lengths and use instead relative and fluid lengths. For example, for a Twitter plugin avoid this:

```
.twitter-timeline {
        width: 300px; /* Avoid ! */
        border: 1px solid #ccc;
}
```

What happens if the widget is inserted in a column that is narrower than 300 pixels? You should avoid rules that can lead to unwanted side effects.

Document your styles

Using comments to document your CSS styles is a must, but you should also provide a specific documentation to allow users to modify your stylesheet without having to scan your code.

> **`.twitter-timeline`** – The main wrapper of the Twitter widget. It comes with the following CSS rules:
>
> 1. `border: 1px solid #ccc;` – A solid grey border of 1 pixel.

It may seem excessive, but if you think really carefully on how many support requests we receive every day about trivial things, you'll probably end up with reconsidering your general approach to plugin's documentation when it comes to CSS styles.