

jQuery: the fn pattern applied to web apps

I generally follow a well-defined jQuery and JavaScript pattern or schema every time I develop a new web app. When I developed my first web app my code was clunky (but it worked). Soon I noticed a very interesting pattern used by many jQuery-based plugins. This pattern keep the auxiliary routines separated from the core of the code.

Since jQuery's plugins use the fn object, we can use the same approach when developing a web app:

```
var APP = {  
  
    Elements: {  
        // common HTML/DOM elements  
    },  
  
    Utils: {  
        // helper methods  
    },  
  
    fn: {  
        // The Core  
    },  
  
    init: function() {  
  
        // Initialization method  
    }  
};
```

Using this code is rather simple:

```
$(function() {  
    APP.init();  
});
```

If your core object contains only methods, you can use the `init()` method sequentially, as follows:

```
init: function() {  
  
    for(var method in this.fn) {  
  
        if(typeof this.fn[method] ===  
'function') {  
  
            this.fn[method]();  
  
        }  
  
    }  
  
}
```

If you suspect that some of your core methods contain logical or syntax errors, you can run them one by one using the same initialization method and logging everything to the JavaScript console.

Another interesting aspect is that your helper methods are available also outside the core of your app. This is not a bad idea because when after a couple of months your client asks you a modification on the fly, you can use these methods on some specific sections to get the desired result.

On the contrary, you can keep them private if you decide to not use them outside the main namespace of your app.