

# Mobile markup best practices

## Use vendor prefixes

When you need to support old versions of mobile browsers like Android 2.3 (or less) and Mobile Safari on iOS 4 (or less) you need to use vendor prefixes.

To solve this problem you can use [prefix free](#) library (only 2 Kb for minified version).

If you need to check which vendor prefixes you need to use, you can try [Can I Use](#) service.

```
-webkit-box-shadow: 0 0 10px blue; // old Mobile Safari
box-shadow: 0 0 10px blue;
```

## Reset css

Some mobile browsers use own CSS styles to control the layout of DOM elements. For example, Safari add rounded corners for all `input[type="button"]` elements on the page.

To fix it you need to use some solution:

```
input, textarea {
  -webkit-appearance: none;
  -webkit-border-radius: 0;
}
```

To fix similar problems recommended to use Mobile css reset. You can download it [here](#)

## Viewport

Viewport is a special meta HTML tag which used to control layout on mobile browsers. It allows you to install zooming settings on mobile devices.

List with all viewport properties you can see [here](#)

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

## Mobile first technique

This technique has been proposed by [Ethan Marcotte](#) in his famous book 'Responsive Web Design'.

This technique relieves mobile traffic loading.

## Autocorrect, autocapitalize, autocomplete

Mobile and desktop browser vendors do their best to help us by providing autocomplete, autocorrect, and autocapitalize features.

But this features often prevent us in web development when we working with authorization forms etc.

You can handle this problem with including of some HTML attributes. How to do it you can read [here](#)

```
<input autocomplete="off" autocorrect="off" autocapitalize="off" spellcheck="false" />
<textarea autocomplete="off" autocorrect="off" autocapitalize="off"
spellcheck="false"></textarea>
```

## Special meta tags in Safari Mobile

Apple developers give us specific meta tags for work with mobile Safari from each device.

For configurate it for your needs you can change it from HTML code. See it [here](#)

```
<meta name="apple-mobile-web-app-status-bar-style" content="black">
<meta name="apple-mobile-web-app-capable" content="yes">
```

## Cache manifest in HTML5

The cache manifest in HTML5 is a software storage feature which provides the ability to access a web application even without a network connection.

You can read about it [here](#)

```
<!DOCTYPE HTML>
<html manifest="demo.appcache">
  <body>
    The content of the document.....
  </body>
</html>
```

## Safari overflow:hidden problem

There may be times when you need to hide the overflow (scrollbars) being visible on a page. You can specify to have all scrollbars hidden or the x and y scroll bars separately.

For instance, if you were to specify for the body to hide the horizontal scrollbars you would want to have the following css in your stylesheet:

```
body {
  position:relative;
  overflow-x:hidden;
}
```

## Webkit tap color

This bug is very common in [SPA](#) and [PhoneGap](#) applications.

This is the problem with highlight color after touching on Android devices in browser.

To fix it you need to do actions like [here](#)

```
.somebox {  
  background: url(/img/html5-badge-128.png) no-repeat;  
  -webkit-tap-highlight-color: rgba(0,0,0,0);  
  -webkit-tap-highlight-color: transparent; /* For some Androids */  
}
```