


## tagName property - why all CAPS?

You may have wondered why we had to use uppercase when referencing our tag name, like this:

```
function showOrHide(e) {  
  let target = e.target;  
  if(target.tagName == "LI") {  
    // ... etc  
  }  
}
```



why is this all in CAPS?


It is confusing because normally you don't need to use caps.

If we want to target a simple div tag for example like this:

```
<div>  
  <!-- You can target this div using lowercase -->  
</div>
```

Its pretty easy, we could just write simple code with lowercase letters:

```
let targetDiv = document.getElementsByTagName('div');
```



we don't have to  
use uppercase  
here. WHY?

## Cutting to the chase

The reason for the difference is HOW we are accessing our tag names.

1. In the first instance, we are accessing a DOM property called `tagName`. We access this `tagName` from the target element that fired the event. Importantly, it is a property on an element and its given to us by the DOM.

2. In the second example, we accessed the DOM `getElementsByTagName()` method, which does not require uppercase tag names.

**Bottom line:** `tagName` and `getElementsByTagName()` are 2 completely different properties / methods. The HTML DOM ALWAYS returns the `tagName` of an HTML element in uppercase form, regardless of the case in the source HTML document. The convention of uppercase tag names probably stems from legacy, when HTML was previously developed based on SGML, and element types were declared in uppercase.

