

# FAQs

Friday, March 31, 2017

6:50 PM

# R&D, Findings

Friday, March 31, 2017 7:00 PM

# Extracting DOM - Murali

Friday, March 31, 2017 6:11 PM

We can use the following code to extract DOM of the application during the run

```
var val=return [].map.call(document.getElementsByTagName("body"),function(node) { return node.outerHTML||node.innerHTML||node.innerText||"";}).join("");
```

So within protractor code will be

```
Browser.executeScript(function(){  
    var val=document.getElementsByTagName("body")  
    return [].map.call(val,function(node)  
    { return node.outerHTML||node.innerHTML||node.innerText||"";}).join("");  
}).then(function(<identifier>){  
    // <identifier> will hold the DOM in form of string  
});
```

# Xpath evaluation using Javascript-Murali

Friday, March 31, 2017 6:17 PM

We can evaluate an xpath using the following code on the JavaScript. Sometime this can help to add events to particular element within application

```
document.evaluate(<xpath>,document,null,XPathResult.FIRST_ORDERED_NODE_TYPE,null).singleNodeValue
```

Stackoverflow comment: <http://stackoverflow.com/questions/36209193/clicking-on-specific-div-class-using-javascript/36209332#36209332>

# Javascript Functions, Hoisting and Scoping - Murali

Friday, March 31, 2017 6:09 PM

1. New functions can be created on runtime using the following code

Var <identifier>=new Function(<para1>,<para2>,<execution statement>);

2. Function Declaration:

function <name>(){} is called function declaration where as

3. Function expression:

var a=function(){} is function expression.

4. Function declarations and function variables are always moved ('hoisted') to the top of their JavaScript scope by the JavaScript interpreter(This is called Hoisting). Function declaration is created on pre-execution stage thus we can declare and call anywhere. Function expression is created only when code flow reaches thus it needs to be created before called.
5. Javascript has functional level scoping. That is if a var is declared outside the block and tried to redeclare inside the block with same name it will rewrite the original value. This is because JS can only declare a new variable scope on function level. Let's say you want to create a new variable with same name use self-invoking funtions.

```
function foo() {  
    var x = 1;  
    if (x) {  
        (function () {  
            var x = 2;  
            // some other code  
       })();  
    }  
    // x is still 1.  
}
```

Further details: <http://www.adequatelygood.com/JavaScript-Scoping-and-Hoisting.html>