# 3.1

# **ANS:**

**Repeat 4 times Index: 4, element: undefined**

# **Explanation:**

1. **In this case var i global scope and overwrote every time so after for loop complete value of i is 4. Value of i increments and check condition if not satisfied then exit loop. Last stored value of var i is 4**
2. **SetTimeout() non-blocking call so for loop execute first and meanwhile setTimeout() in browser callback queue and event loop. This is asynchronous process.**

1. **After 3 second setTimeout() execute and console.log statement print Index: 4, element: undefined because value of i is 4 (for loop i last value)**
2. **setTimeout() access for loop(in this case parent) var i=4**

# Solution 1 - IIFE

**var** arr = ['1','2','3','4']  
**for**(**var** i=0; i<arr.length; i++){  
 (**function**(i){  
 setTimeout(**function**(){  
 console.log('Index:' + i + ', element:' + arr[i]);  
 },3000)  
 })(i)  
}

# Solution 2 – Each function create unique scope.

**var** arr = ['1','2','3','4']  
**for**(**var** i=0; i<arr.length; i++){  
  
 **function** uniqueFnScope(){  
 **var** j = i;  
 setTimeout(**function**(){  
 console.log('Index:' + j + ', element:' + arr[i]);  
 },3000)  
 }  
 }  
 uniqueFnScope();  
}

# Solution 3 – ES6 let block scope

**var** arr = ['1','2','3','4']  
**for**(**let** i=0; i<arr.length; i++) {  
 setTimeout(**function** () {  
 console.log('Index:' + i + ', element:' + arr[i]);  
 }, 3000)  
}