

Professional Skills
Agile Fundamentals
Jira
Git
Databases Introduction
Java Beginner
Maven
Testing (Foundation)
Java Intermediate
HTML
CSS
JavaScript
<div><div></div>What is JavaScript</div>
<div><div></div>Getting started with JS</div>
<div><div></div>Variables</div>
<div><div></div>Data types</div>
<div><div></div>ASI</div>
<div><div></div>Strict mode</div>
<div><div></div>Iteration</div>
<div><div></div>Conditionals with Truthy / Falsey</div>
<div><div></div>Objects, Arrays + JSON</div>
<div><div></div>Structuring JS Code</div>
<div><div></div>Destructuring</div>
<div><div></div>Scope</div>
<div><div></div>Functions, function expressions and arrow functions</div>
<div><div></div>The ECMAScript 6 Specification</div>
<div><div></div>OOP in JavaScript</div>
<div><div></div>Best Practices</div>
<div><div></div>Closures</div>
<div><div></div>Callbacks and Promises</div>
<div><div></div>Cookies</div>
<div><div></div>Hoisting</div>
<div><div></div>Prototypes</div>
<div><div></div>Query Parameters</div>
<div><div></div>Higher Order Functions</div>

# ASI

## Contents

- [Overview](#)
- [Tutorial](#)
- [Exercises](#)

## Overview

ASI is Automatic Semi-Colon Insertion and can be a fiddly aspect of Javascript. Code termination is optional in JavaScript meaning that semi-colons aren't required at the end of each statement. Best code practise states that we must insert semi-colons where necessary because if not, they are automatically inserted for you; we will explore how this might not work in our favour...

## Tutorial

In JavaScript a semi-colon is automatically inserted when:

- Two lines are seperated by a line terminator
- Two statements are seperated by a closing brace (}')'
- Line terminator follows a BREAK, RETURN, THROW or CONTINUE

Although in principle, ASI is trying to do us a favour; most times it leads to bugs in our code which can make debugging difficult if you don't know what to look for.

Entered Code	Interpreted as	Correct syntax
<pre>return 2*a+1;</pre>	<pre>return; 2a*1;</pre>	<pre>return 2a*1;</pre>
<pre>let foo bar; console.log(foo); console.log(bar);</pre>	<pre>let foo; boo; console.log(foo); console.log(bar);</pre>	<pre>let foo, bar; console.log(foo); console.log(bar);</pre>
<pre>let a b =3</pre>	<pre>let a; b=3;</pre>	<pre>let a,b=3;</pre>

**Note:** Writing code without semi-colons is acceptable as long as you understand the concept of ASI and its process. As a general rule of thumb, it is good practice to terminate your code with a semi-colon - ;

## Exercises

- Declare a few variables / code blocks and terminate the code appropriately.
- Remove the semi-colons from the variables / code-blocks that you've written and see if there is an affect on the output.
- Have a look at the below code and try to spot where there might be an error:

- Web Storage
- DOM Manipulation
- Handling Events and Timed Events
- Asynchronous Programming
- HTTP-Requests
- XMLHttpRequests
- Fetch API

## Spring Boot

## Selenium

## Sonarqube

## Advanced Testing (Theory)

## Cucumber

## MongoDB

Express

## NodeJS

## React

## Express-Testing

## Networking

## Security

## Cloud Fundamentals

## AWS Foundations

## AWS Intermediate

## Linux

## DevOps

## Jenkins Introduction

## Jenkins Pipeline

## Markdown

## IDE Cheatsheet

```
function sayHello() {  
  return  
  {  
    "Someone says hello";  
  }  
}
```

► Solution