

# Microsoft

Technology Associate



Worldwide  
Technological  
Certification.

# JavaScript Fundamentals

Examen de Certificación Microsoft MTA 98-382  
Modulo 1: Sesión 1 : Variables, DataTypes,  
Funciones



# Objetivos de la Sesión



Worldwide  
Technological  
Certification.

- Comprender la propuesta de JavaScript frente a la gestión de variables, tipos de datos y funciones

# PRESENTACION

- Instructor
- participantes

# 1. Qué es JavaScript?

- Puedes buscar multiples definiciones en libros o en la web



## 2. Java == Javascript?



**Java**

A programming language used to write applications for computers and other devices



**JavaScript**

A scripting language used to interact with content in a web browser and more recently in other places

### 3. El ecosistema JavaScript



# ES

- JavaScript conforma el estandar ECMAScript el cual es una compilación de propuestas para funcionamiento de lenguajes de programación en navegadores
- ECMAScript 2015: ultima versión del estandar y conocida como ECMAScript 6 (ES6)
- ES7 se encuentra en fase de desarrollo



# jQuery

- Librería basada en JavaScript
- La sintaxis está basada en la propuesta CSS, orientada a User Interface (UI)
- Simplificación de JavaScript
- Ha quedado desfasada (recomiendo no usarla)

# JavaScript Frameworks

- Angular.JS, React, Vue.JS y otros: son frameworks frontend utilizados para simplificar el desarrollo de aplicaciones web que generalmente ejecutan operaciones de avanzada

# Node.js

- Framework orientado a operaciones backend
- Habilidad en alta demanda

## 4. Herramientas para desarrollo con JavaScript

- No se requiere compilación del código
- No se necesita instalación de ningún paquete para tener el core de JavaScript
- Únicamente se necesita un editor de texto y un navegador
- Hay agregados específicos para apoyar labores de depuración

## 5. Cómo seleccionar el editor de texto

- Distinción de funciones y statements que pertenecen a JS
- Preferiblemente que tenga funciones de pistas de lenguaje: Code Hinting o Intellisense
- Editores de código: Atom, VSCode, Sublime Text
- IDE: NetBeans, Aptana Studio, Eclipse
- Para estas sesiones de trabajo utilizaremos ATOM (atom.io)

## 6. Qué browsers puedo usar

- Chrome
- Firefox
- Edge
- Opera
- Brave

## 7. Practica

- Instalación de Atom
- Instalación del agregado Atom Live Server

## 8. Consola del navegador

```
>> alert("I love dogs")
```

```
← undefined
```

```
>> alert(5+8)
```

```
← undefined
```

```
>> var fecha = new Date()  
    alert("la fecha actual es: "+fecha)
```

```
← undefined
```

```
>> console.log("la fecha actual es: "+ fecha)
```

```
la fecha actual es: Fri Aug 21 2020 20:48:58 GMT-0600 (Central Standard Time)
```

```
← undefined
```



## 9. Inyeccion de HTML

- `document.body.innerHTML = "<h1>La fecha actual es "+fecha+"</h1>"`
- `document.body.innerHTML = "<h1>La fecha actual es "+fecha.getDate() + "/" + (fecha.getMonth()+1) + "/" + fecha.getFullYear()+"</h1>"`

## 10. In line JavaScript

- Si se necesita ejecutar código js antes de la carga del documento HTML, éste debe escribirse entre la etiqueta HEAD del archivo
- Caso contrario, puede escribirse después de la etiqueta BODY
- DEMO

# 11. External Script

- Normalmente cuando las operaciones que se requieren con JS son muchas, es aconsejable diseñar un archivo exclusivo para éste
- Aunque no es una mala práctica usar inline js, se recomienda que sea cuando se necesita efectuar una actividad no repetitiva ni crucial
- DEMO

## 12. Asignación

- Investigar sobre el impacto que tiene en el rendimiento de un sitio web renderizar contenido js
- Contrastar con la propuesta de HTTP/2