

# Microsoft

Technology Associate



Worldwide  
Technological  
Certification.

# JavaScript Fundamentals

Examen de Certificación Microsoft MTA 98-382  
Modulo 2: Sesion 6 : Operaciones especiales con  
arreglos, funciones y objetos



# Objetivos de la Sesión



Worldwide  
Technological  
Certification.

- Desarrollar habilidades en el uso de operaciones especiales con arreglos y funciones

# 1. Operaciones especiales con arreglos

# 1. Mecanismos de inicialización

- `Let arreglo = new Array();` //arreglo vacío
- `Let arreglo = new Array(10, 20, 30);` //con valores
- `Let arreglo = new Array(10);` //10 espacios en blanco
- `Let arreglo = new Array('10');` //un espacio con el valor 10

# Mecanismos de inicialización

- Notacion literal:
- `Let arreglo = [];` //arreglo vacío
- `Let arreglo = [10, 20, 30];` //con valores
- `Let arreglo = [10];` //10 espacios en blanco
- `Let arreglo = ['10'];` //un espacio con el valor 10

# Mecanismos de inicialización

- Múltiples líneas:
- Let arreglo = [
  - “1”,
  - “2”,
  - “3”
  - ];

## 2. Iteración con arreglos

- Elemento por elemento:
- Arreglo = [ 'herbert', 'fernandez', '24'];
- `Console.log(arreglo[0]);`
- `Console.log(arreglo[1]);`
- `Console.log(arreglo[2]);`



- Usando estructura de control for:
- Arreglo = [1, 2, 3, 4, 5, 6];
- for(i = 0; i < arreglo.length; i++){
  - Console.log(arreglo[i]);
- }

- Usando estructura de control while:
- `i = 0;`
- `Arreglo = [1, 2, 3, 4, 5, 6];`
- `while(i < arreglo.length){`
  - `Console.log(arreglo[i]);`
  - `i++;`
- `}`

- Usando el metodo forEach de arreglo:
- `i = 0;`
- `Arreglo = [1, 2, 3, 4, 5, 6];`
- `Arreglo.forEach(iterar);`
- `Function iterar(item, i){`
  - `Console.log(item);`
- `}`

### 3. Ordenamiento de arreglos

- `Var paises = ["El Salvador", "Costa Rica", "Honduras", "Nicaragua", "Guatemala"];`
- `Paises.sort();` //ordena el arreglo de forma ascendente
- `Paises.reverse();` //ordena el arreglo de forma inversa

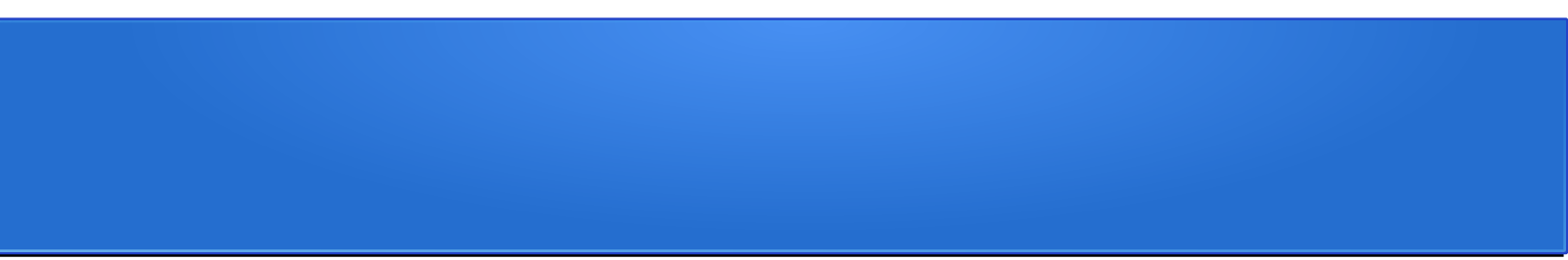
- Ordenamiento de numeros: los resultados de la funcion sort no son los mismos con numeros en lugar de cadenas
- `Var puntajes = [40, 100, 1, 5, 25, 10];`
- `Puntajes.sort();`
- Se puede lograr el ordenamiento de esta manera:
- `Puntajes.sort(function(a, b){return a -b});`

- Si se necesita un ordenamiento descendente, el resultado puede ser este:
- `Puntajes.sort(function(a, b){return b -a});`

- Función de comparación:
- Una comparación implícita permite lograr el ordenamiento de tipos de datos numericos
- La salida puede ser un valor negativo, cero o positivo, depende de los argumentos:
- `function(a, b){return a – b}`
- Ver [comparar.html](#)

- Ordenar un arreglo en forma aleatoria:
- `var puntos = [40, 100, 1, 5, 25, 10];`
- `puntos.sort(function(a, b){return 0.5 – Math.random()});`
- Asignacion: **genera un script que implemente el orden aleatorio**



- 
- No hay un método específico para obtener el mayor / menor valor en un arreglo
  - Asignacion: codifique una función que obtenga el mayor / menor valor de un arreglo
  - Otros comandos para iteracion:
  - Asignacion semanal: codifique ejemplos para uso de los métodos: `filter()`, `reduce()`, `reduceRight()`, `every()`, `some()`, `indexOf()`, `lastIndexOf()`, `findIndex()`,
  - Video semanal: arreglos multidimensionales

## 2. Funciones

# Objetivo de las funciones

- Modularizar y reutilizar código fuente
- Sintaxis:
  - `function <nombre>(<parametros>){`
    - `<cuerpo de la funcion>`
  - `}`
  - `function mostrarMensaje(){`
    - `alert('El proceso de registro de datos ha finalizado');`
  - `}`

# Tipos de variables en funciones

- 1. Variable local: existe dentro de una función y no es accesible fuera de ella
- `function mostrarMensaje() {`
- `let mensaje = "El proceso ha finalizado"; // variable local`
- `alert( mensaje );`
- `}`

# Tipos de variables en funciones

- 2. Variable externas (outer): existen fuera del ambito de una función y pueden ser accesibles:
- `let nombreUsuario = 'Jorge';`
- `function mostrarMensaje() {`
- `let mensaje = "El proceso ha finalizado" + nombreUsuario;`
- `alert( mensaje );`
- `}`

# Tipos de variables en funciones

- La variable externa a la función puede ser modificada dentro de ésta:
- `let nombreUsuario = 'Jorge';`
- `function mostrarMensaje() {`
- `NombreUsuario = 'Carlos';`
- `let mensaje = "El proceso ha finalizado" + nombreUsuario;`
- `alert( mensaje );`
- `}`

# Tipos de variables en funciones

- 3. Variables globales:
- Las variables declaradas fuera de una función -justo como las variables externas- son identificadas como globales
- Éstas son visibles desde cualquier función que pertenezca al script
- Se anima a NO UTILIZAR variables globales debido a consideraciones de seguridad
- Toda variable debe estar adscrita a una función

# Parametros

- Una función puede recibir parametros de entrada aunque no es obligatorio
- `function mostrarMensaje(proceso, usuario) {`
  - `alert('Estimado ' + usuario + 'el proceso: ' + proceso + ' ha finalizado');`
- `}`
- `mostrarMensaje('crear perfil', 'Jorge');`



# Parametros

- Para el ejemplo anterior no se utilizaron variables locales para gestionar los parametros, se aconseja:
- Definir y usar una variable local por cada parámetro
- Si no amerita el caso se aconseja definir “on the fly”
- [Ver parametros.js](#)

# Valores default

- mostrarMensaje("Carlos")
- La función espera 2 parámetros
- En Javascript si un parametro no fue enviado se asume que tiene la categoría de "undefined"
- A cada parámetro se puede asignar un valor default
- **Ver mostrarMensaje3 y mostrarMensaje4 en parametros.js**

# Asignacion

- 1. Codifique la función `definirTexto()` para que `mostrarTexto4()` pueda hacer uso de ella.
- 2. Desarrolle una aplicación web que acepte los siguientes datos de entrada: nombres, apellidos, correo electrónico, horas trabajadas, salario por hora
- Validar que todos los datos de entrada sean coherentes (correos, cifras, cadenas sin rellenar)
- Calcular: salario bruto, descuento de la renta, descuento del isss, horas extras (mayor de 160 horas equivalente al 5% sobre salario por hora)
- Mostrar descuentos y salario liquido
- Generar un numero aleatorio que identifique el cálculo