Microsoft

Technology Associate

JavaScript Fundamentals

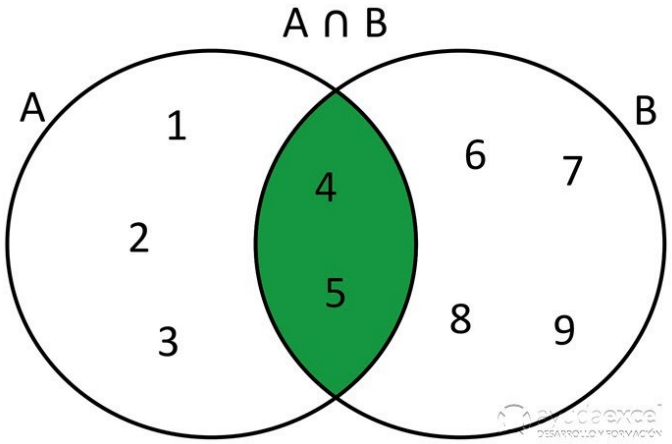
Examen de Certificación Microsoft MTA 98-382
Modulo 1: Sesion 3 : Condicionales, conversión y formateo

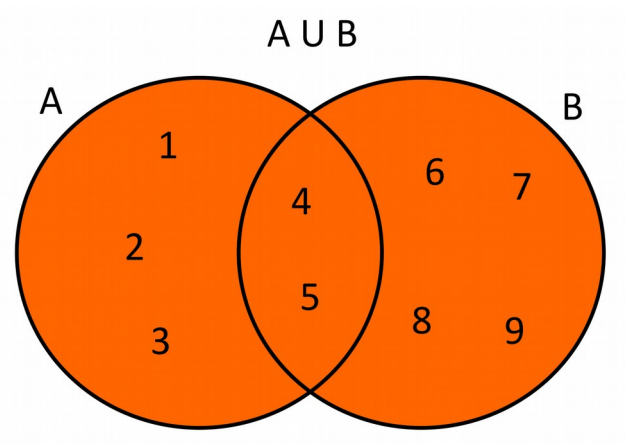
Objetivos de la Sesión

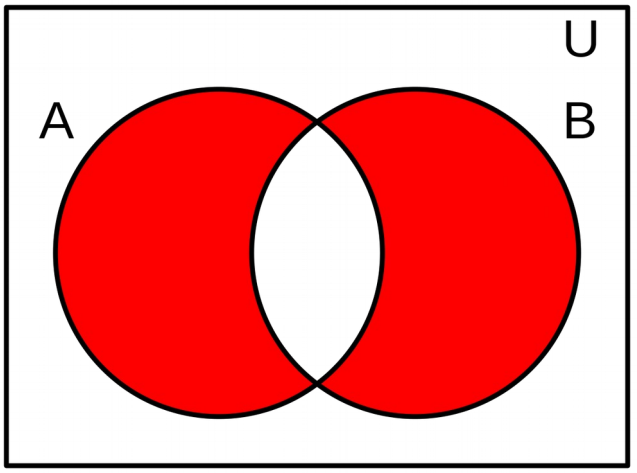
- Desarrollar habilidades en el uso de condicionales, formateo y conversión de datos de entrada

1. Uso de condicionales anidados

- Operadores lógicos:
  - AND : &&
  - OR: ||
  - XOR: no existe un operador en JS







1. Uso de condicionales anidados

- Ejemplo de AND:
  - If (a == b && c ==d) { }
- Ejemplo de OR:
  - If (a == b || c ==d) { }
- Ejemplo de XOR:
  - If (a == b || c ==d) && ((a == b) !=(c == d)) { }

1. Operador ternario y condicionales anidados

- Diseño lineal de condicional en una sóla linea:
  - a == b ? console.log(“verdadero”) : console.log(“falso”)
- Condicionales anidados:
- En ocasiones se requiere la evaluación secuencial de una condición, el condicional if puede tener esta característica:
  - if(condicion){ }
  - else if (condicion){ }
  - else{ }

1. Asignaciones

- Resuelva las siguiente asignaciones:
- Ejercicio 1: codificar un programa que identifique el mayor de 3 numeros
- Ejercicio 2: Se requiere leer el año de nacimiento de una persona, a partir del resultado debe informarse el rango de edades al que pertenece:
  - 0-10: primera infancia
  - 11-17: adolescente
  - 18-40: adulto – joven
  - 41-59: adulto
  - 60 a más: adulto mayor
- Ejercicio 3: en base a un numero establecido por el usuario, determine si es par o impar

2. Conversiones entre data types

- En Javascript ciertos operadores y funciones automáticamente convierten a un tipo de datos que se ajuste a la salida de datos
- La función alert convierte cualquier valor a String
- Operadores aritméticos convierten valores a números
- En ciertas ocasiones la conversión debe ser explícita

Ejemplo 1: boolean a String

- var variableBooleana = true;
- alert(typeof variableBoolean);
- variableBooleana =String(variableBooleana);
- alert(typeof variableBooleana);

Ejemplo 2: conversión numérica

- alert(“20” / “10”); //conversión automática
- Conversión explítica:
  - var variableNumerica = “123”;
  - alert(typeof variableNumerica);
  - alert(typeof variableNumerica);
- Conversión tipo NaN:
  - var edad = Number(“Esta es una cadena”);
  - alert(edad);

Ejemplo 2: conversión numérica

- Reglas de conversión numérica:
  - Undefined → NaN
  - Null → 0
  - True y false → 1 y 0

Ejemplo 2: conversión numérica

- alert(Number(“ 755 “));
- alert(Number(“123z”));
- alert(Number(true));
- alert(Number(false));

Ejemplo 3: conversión boolean

- alert(Boolean(1));
- alert(Boolean(0));
- alert(Boolean(“hola”)); //una cadena no vacia es true
- alert(Boolean(“”));
- alert(Boolean(“0”)); //una cadena no vacia es true
- alert(Boolean(“ ”)); //una cadena no vacia es true

2. Asignaciones

- Diseñe un programa que permita obtener un dato de entrada y permita convertirlo a sus equivalentes
- Escriba la versión 2.0 del programa anterior que sea compatible con el estándar ES6

3. Formateo de numeros

- Respecto a representación de cantidades numericas, distintos lenguajes de programación tienen funciones especiales que facilite la salida de datos en formatos de:
  - Miles
  - Millares
  - Decimales
  - Símbolos de monedas
  - Notación científica
- En Javascript hay distintas maneras de formatear números

3.1 Expresiones regulares

- Son operaciones especiales que se utilizan para crear un patrón (plantilla) que será utilizada para generar salida de datos
- En JS suele combinarse una expresión regular por medio de la función replace()

3.1 Expresiones regulares

- Patrón (expresión regular): (?=(\d{3})+(?!\d))
- Valor de reemplazo: $1,
- Explicación: la expresión regular ubicará un patrón de el dígito más a la izquierda y lo va a relacionar con los siguientes 3 dígitos, si hay una coincidencia al primer dígito identificado se agregará el símbolo de $ más el separado de miles (,)
- DEMO

3.1 Expresiones regulares

- En proximas sesiones retomaremos el formateo de salida de datos por medio de expresiones regulares junto con otras técnicas

4. Buenas practicas: como inicializar variables

- Se aconseja inicializar variables cuando se declaran, de esta manera definimos en nuestro código la etapa para declaración de variables y aún más importante evitar “variables indefinidas”
- Ejemplo que NO DEBE SEGUIRSE;
  - var x;
  - var y

4. Buenas practicas: como inicializar variables

- Ejemplos que se aconseja seguir:
  - var primerNombre = "",
  - var apellidosCliente= "",
  - var precioContado = 0,
  - var descuentoEspecial = 0,
  - var precioVenta = 0,
  - var calificacionesFinales = [],
  - var objetoX = {};

4. Buenas practicas: como inicializar variables

- Tipos de datos primitivos en JS: number, string, boolean
- Tipos de datos extendidos en JS: objects
- En la consola del navegador evalúe el siguiente código:
  - Ejemplo 1:
  - var x = "Jose";
  - var y = new String("Jose");
  - (x === y)
  - Ejemplo 2:
  - var x = new String("Maria");
  - var y = new String("Maria");
  - (x == y)