**TALLER IMPORT, EXPORT Y OPERADORES TERNARIOS**

**Ejercicio 1: Determinar si un número está en un rango específico**

**Enunciado**:  
Crea una función llamada enRango que reciba un número y determine si está en el rango entre 10 y 50 (inclusive). La función debe retornar:

* "Está en el rango" si el número está entre 10 y 50 (inclusive).
* "Fuera del rango" si el número no está en ese rango.

**Uso de módulos**:

* Exporta la función enRango en un archivo utilidades.js.
* En un archivo main.js, importa la función enRango y pruébala con diferentes números.

**Ejercicio 2: Calcular el descuento en un producto según el precio**

**Enunciado**:  
Escribe una función llamada calcularDescuento que reciba el precio de un producto y calcule el descuento aplicable de acuerdo a la siguiente lógica:

* Si el precio es mayor a 1000, el descuento es del 20%.
* Si el precio es entre 500 y 1000, el descuento es del 10%.
* Si el precio es menor a 500, no se aplica descuento.

La función debe retornar el precio final después de aplicar el descuento.

**Uso de módulos**:

* Exporta la función calcularDescuento en un archivo utilidades.js.
* En un archivo main.js, importa la función calcularDescuento y pruébala con diferentes precios.

**Ejercicio 3: Categorizar el IMC de una persona**

**Enunciado**:  
Escribe una función llamada categoriaIMC que reciba el índice de masa corporal (IMC) de una persona y determine su categoría:

* "Bajo peso" si el IMC es menor a 18.5.
* "Normal" si el IMC está entre 18.5 y 24.9.
* "Sobrepeso" si el IMC está entre 25 y 29.9.
* "Obesidad" si el IMC es 30 o mayor.

**Uso de módulos**:

* Exporta la función categoriaIMC en un archivo utilidades.js.
* En un archivo main.js, importa la función categoriaIMC y pruébala con diferentes valores de IMC.

**Ejercicio 4: Clasificar el nivel de riesgo de un usuario según su edad y si tiene enfermedades previas**

**Enunciado**:  
Escribe una función llamada nivelRiesgo que reciba dos parámetros: la edad de una persona y un valor booleano (true o false) que indique si la persona tiene enfermedades previas. La función debe retornar:

* "Alto riesgo" si la persona tiene más de 60 años o tiene enfermedades previas.
* "Riesgo moderado" si la persona tiene entre 40 y 60 años y tiene enfermedades previas.
* "Bajo riesgo" en cualquier otro caso.

**Uso de módulos**:

* Exporta la función nivelRiesgo en un archivo utilidades.js.
* En un archivo main.js, importa la función nivelRiesgo y pruébala con diferentes combinaciones de edad y enfermedades.

**Ejercicio 5: Verificar si un año es bisiesto**

**Enunciado**:  
Escribe una función llamada esBisiesto que reciba un año y determine si es bisiesto. Un año es bisiesto si:

* Es divisible por 4, pero no es divisible por 100, a menos que también sea divisible por 400.

La función debe retornar:

* "Es bisiesto" si el año es bisiesto.
* "No es bisiesto" si el año no lo es.

**Uso de módulos**:

* Exporta la función esBisiesto en un archivo utilidades.js.
* En un archivo main.js, importa la función esBisiesto y pruébala con diferentes años.

**Ejercicio 6: Verificar si una persona es elegible para un préstamo bancario**

**Enunciado**:  
Escribe una función llamada esElegibleParaPrestamo que reciba dos parámetros: el salario anual de una persona y su puntaje de crédito. La función debe retornar:

* "Elegible para préstamo" si el salario es mayor a 30000 y el puntaje de crédito es mayor a 650.
* "No elegible para préstamo" en cualquier otro caso.

**Uso de módulos**:

* Exporta la función esElegibleParaPrestamo en un archivo utilidades.js.
* En un archivo main.js, importa la función esElegibleParaPrestamo y pruébala con diferentes valores de salario y puntaje de crédito.