

Ejemplo 1

Comparando dos números

Solicitar dos números, y luego la comparación deseada (>, <, =, !=) mayor que, menor que, igual o desigual.

Retornar el resultado de la comparación

Paso 1: Análisis y Requerimientos

Este problema es sencillo, debemos de obtener dos números, y luego un operador. así que tenemos que guardar esos tres valores, luego dependerá del tercer valor la ejecución del resto del programa

Paso 2: Algoritmo

Pedir el primer número

Guardar dicho número

Pedir el segundo número

Guardar el segundo número

Pedir el operador

Verificar el operador

 si es (>) mayor que, retornar el resultado de (numero1 > numero2)

 si es (<) menor que, retornar el resultado de (numero1 < numero2)

 si es (=) igual, retornar el resultado de (numero1 = numero2)

 si es (!=) desigual, retornar el resultado de (numero1 != numero2)

Paso 3: PseudoCódigo

en **numero1** guardar **prompt**("Ingrese un número")

en **numero2** guardar **prompt**("Ingrese otro número")

en **operador** guardar **prompt**("Seleccione una operación (>,<=,!=)")

si **operador == >**

 retornar **numero1 > numero2**

si **operador == <**

 retornar **numero1 < numero2**

si **operador == =**

 retornar **numero1 == numero2**

si **operador == !=**

retornar **numero1 != numero2**

Paso 4: Código

```
var numero1 = prompt("ingrese un numero")
var numero2 = prompt("ingrese otro numero")
var operador = prompt("ingrese la operacion deseada (>,<=?,!=)")
if(operador == ">")
    console.log(numero1+" es mayor que "+numero2+": "+(numero1 >
numero2))
else if(operador == "<")
    console.log(numero1+" es menor que "+numero2+": "+(numero1 <
numero2))
else if(operador == "=")
    console.log(numero1+" es igual que "+numero2+": "+(numero1 ==
numero2))
else if(operador == "!=")
    console.log(numero1+" es desigual que "+numero2+": "+(numero1 !=
numero2))
```

Ejemplo 2

Obtener los primeros N números primos

Paso 1: Análisis y Requerimientos

Este es un problema de lógica, donde debemos de poder calcular números primos y mientras encontramos cada uno, ir contando para parar la ejecución del programa cuando encontremos los primeros N números.

Para solucionar este problema debemos de, primero, saber encontrar números primos, para hacer esto, debemos de conocer la definición de un número primo

“Un número primo es aquel que solamente tiene dos divisores exactos, 1 y el mismo”

Sabiendo esto, para determinar si un número es primo, lo que tenemos que hacer es averiguar si tiene algún divisor exacto.

Una vez sepamos hacer esto, es cuestión de saber cuando parar la ejecución del programa

Paso 2: Algoritmo

Pedir la cantidad de números primos

Guardar la cantidad N(será nuestra condición de salida)

Empezar a buscar números primos

Comprobamos números hasta encontrar N números

Para esto debemos de buscar si el número tiene algún divisor exacto

Comprobamos divisores desde 2 hasta nuestro número actual

Si encontramos divisores entonces no es primo

Si *no* encontramos, entonces es primo, y lo guardamos

Paso 3: PseudoCódigo

crear **arregloPrimos** arreglo en el que guardaremos los números encontrados

en **numeroPrimos** guardar **prompt**("ingrese la cantidad de números primos que desea ver")

comprobar números iniciando en el 2 y todos los superiores, con un **ciclo infinito**

comprobar los **divisores** desde el 2 hasta el **número actual**

si existe un **divisor exacto**, no es primo, comprobamos el siguiente número

si no existe, agregarlo a **arregloPrimos**

comprobar la **condición de salida**

si se encontraron **numeroPrimos** números, salir

si no, continuar

desplegar el arreglo

Paso 4: Código

```
var arregloPrimos = []
var numeroPrimos = prompt("ingrese la cantidad de numeros primos que desea ver")
var esPrimo = true

for(var i = 2; ; i++){
  for(var j = 2; j<i; j++){
    if(i%j === 0){
      esPrimo = false
      break;
    }
  }
  if(esPrimo === true){
```

```

    arregloPrimos[arregloPrimos.length] = i
  }
  esPrimo = true
  if(arregloPrimos.length == numeroPrimos){
    break;
  }
}

console.log(arregloPrimos)

```

***Nota:** aquí usamos un ciclo 'for', que no tiene una condición de salida, esto se debe a que no sabemos cuando va a parar el ciclo. Por eso tenemos la comprobación al final que nos indica cuando hemos encontrado los números requeridos*

Ejemplo 3

Encontrar todos los divisores de los números entre N y M

Paso 1: Análisis y Requerimientos

Este es un ejercicio más simple, en el tendremos un rango de números, entonces tenemos que obtener esos dos números.

Luego, obtener los divisores es simple, hay que comprobar los números más pequeños que el actual, y ver el resultado de la división

Paso 2: Algoritmo

Pedir el primer número

en **rangoInferior** guardar el primer número

Pedir el segundo número

en **rangoSuperior** guardar el segundo número

para todos los números entre **rangoInferior** y **rangoSuperior**

dividir entre todos los números desde **2** hasta el **número actual**

cuando llegamos al número actual, desplegamos todos los divisores

Paso 3: PseudoCódigo

en **rangoInferior** guardar **prompt**("Ingrese el rango inferior")

en **rangoSuperior** guardad **prompt**("Ingrese el rango superior")

en **ciclo** desde **rangoInferior** hasta **rangoSuperior**

comprobar si algún número es divisor exacto de nuestro contador

si lo es lo guardamos

al comprobar todos los números, desplegamos los divisores que encontramos

Paso 4: Código

```
var rangoInferior = prompt("Ingrese el rango inferior")
var rangoSuperior = prompt("Ingrese el rango superior")

for (var i = rangoInferior; i<=rangoSuperior; i++){
    var divisores = []
    for(var j = 1; j<=i;j++){
        if(i%j === 0){
            divisores[divisores.length] = j
        }
    }
    console.log("los divisores de "+ i +" son: "+divisores)
}
```

Ejemplo 4

Tenemos dos arreglos llenos de números, solicitar un número, y ver si podemos obtener ese número como una sumatoria usando un número de cada arreglo.

Paso 1: Análisis y Requerimientos

Este problema requiere un poco más de análisis, primero nos dicen que tenemos dos arreglos de números ej.

a1 = [1,2,3,5,7,11,13,17,23]

a2 = [1,2,3,5,8,13,21,34,55]

luego tendremos que solicitar un número al usuario, y con ese número tenemos que ver si podemos armar una sumatoria de la forma

$a1[i] + a2[j] = \text{número_solicitado}$

tomando un número de cada arreglo, armar una sumatoria para obtener el número solicitado

Paso 2: Algoritmo

Solicitar un número

Para todos los números del primer arreglo

Comprobar todos los números del segundo arreglo

Tratar de armar la sumatoria

Si encontramos la sumatoria retornar

Paso 3: PseudoCódigo

en **numero_solicitado** guardar **prompt**("que número desea sumar?")

ciclo para todos los números del **arreglo1**

comprobar **si existe** $\text{numero_solicitado} = \text{arreglo1}[i] + \text{arreglo2}[j]$

para todos los números del **arreglo2**

Paso 4: Código

```
var arreglo1 = [1,2,3,5,7,11,13,17,23]
var arreglo2 = [1,2,3,5,8,13,21,34,55]
var numero_solicitado = prompt("que número desea sumar?")
var suma = "";
for(var i = 0; i<arreglo1.length; i++){
  for(var j = 0; j<arreglo2.length;j++){
    if(numero_solicitado == arreglo1[i] + arreglo2[j]){
      suma = ["el numero: "+arreglo1[i],"del arreglo1 en la posicion: "+i,"el numero: "+arreglo2[j], "en la posicion: "+j]
      break;
    }
  }
  if(suma.length > 1){
    break;
  }
}
console.log(suma)
```

Extra: Resolviendo el problema con funciones.

Este mismo ejercicio se puede resolver usando una función. La utilidad de usar funciones es que nos permite tener código más modular. y más fácil de entender.

En js podemos declarar una función, asignando a una variable de la siguiente forma

```
var funcion = function(parametro1, parametro2, ...){  
    //Código a ejecutar cuando se llama a la función  
}
```

La palabra **function**, indica que se va a crear una función

Entre paréntesis van **parámetros opcionales**, las funciones se pueden declarar sin parámetros, pero también podemos pasarle valores. como es el caso con el **prompt** podemos mostrar un **prompt** vacío, pero también podemos hacer que muestre un mensaje, e incluso que muestre un valor predeterminado.

Cuando definamos lo que hace nuestra función podemos invocarla de la siguiente forma

```
funcion(parametro1, parametro2, ...)
```

El ejemplo, lo podemos resolver así

```
var arreglo1 = [1,2,3,5,7,11,13,17,23]  
var arreglo2 = [1,2,3,5,8,13,21,34,55]  
var numero_solicitado = prompt("que número desea sumar?")  
var suma = function(numero_solicitado){  
    for(var i = 0; i<arreglo1.length; i++){  
        for(var j = 0; j<arreglo2.length;j++){  
            if(numero_solicitado == arreglo1[i] + arreglo2[j]){  
                var sumatoria = ["el numero: "+arreglo1[i],"del  
arreglo1 en la posicion: "+i,"el numero: "+arreglo2[j], "en la  
posicion: "+j]  
                return sumatoria  
            }  
        }  
    }  
    return "No se puede hacer la suma con los valores de los  
arreglos"  
}  
console.log(suma(numero_solicitado))
```