Programación

A partir de ahora, los algoritmos que desarrollemos van a ser procesados por la computadora. Estaremos trabajando de la siguiente manera:

1. Desarrollaremos algoritmos en papel (pseudocódigo), donde estaremos viendo conceptos y lógica de programación.
2. Implementaremos los algoritmos escritos anteriormente en el lenguaje de programación JavaScript.

Un **lenguaje de programación** es un lenguaje que permite a los desarrolladores escribir un conjunto de instrucciones que serán ejecutadas por una computadora.

Variables y constantes

Las variables son **lugares** donde se pueden **almacenar datos**. Para identificarlas se les da un nombre. Cuando se referencie ese nombre, se está hablando del dato que contiene.
Una variable puede almacenar **solo un dato a la vez**. Cuando se asigna un nuevo valor, el anterior de pierde.

Las **constantes** son similares a las variables. La diferencia está en que le podemos asignar un dato una **única vez**. Si luego le tratamos de asignar otro dato tendremos un error en nuestro programa.

A cada variable o constante que creemos en nuestros programas se les debe indicar el **tipo de dato** que almacenaran.

Tipos de datos primitivos

La computadora es capaz de procesar **números**, **textos**, y **booleanos**.

Números: Dentro de los tipos de datos numéricos, tenemos 2 tipos:

- **Numérico entero**: Son los números negativos y positivos exactos, es decir, sin comas, tales como:
  - 5
  - -100
  - 0
  - 12345678

- **Numérico real**: Son los números con coma, tales como:
  - 3.14
  - -10.5
  - 1.0

Operaciones: Podemos hacer varias operaciones entre los tipos de datos numéricos, tales como:

- **Sumar**, con el operador **+**. Ej: **10 + 5** (Resultado: **15**)
- **Restar**, con el operador **-**. Ej: **10 – 5** (Resultado: **5**)
- **Multiplicar**, con el operador *. Ej: **10 * 5** (Resultado: **50**)
- **Dividir**, con el operador **/**. Ej: **10 / 5** (Resultado: **2**)
- **Resto de una división**, con el operador **%**. Ej: **10%5** (Resultado: **0**)

Condiciones: Podemos preguntar si dos números son:

- **Iguales**, con el operador **==**. Ej: **10 == 5** (Resultado: **false**)
- **Diferentes**, con el operador **!=**. Ej: **10 != 5** (Resultado: **true**)
- **Mayor que**, con el operador **>**. Ej: **10 > 5** (Resultado: **true**)
- **Menor que**, con el operador **<**. Ej: **10 < 5** (Resultado: **false**)
- **Mayor o igual que**, con el operador **>=**. Ej: **5 >= 5** (Resultado: **true**)
- **Menor o igual que**, con el operador **<=**. Ej: **5 <= 5** (Resultado: **true**)

Cadena de caracteres (texto): También son conocidos como **strings**. Para crear datos de este tipo, se los debe encerrar entre comillas dobles. Ejemplos de cadenas de caracteres son:

- “Argentina”
- “10”
- “Hola mundo!”

Podemos concatenar (unir) textos con el operador **+**:

- **“Hola ” + “Mundo”** (Resultado: **“Hola Mundo”**)

También podemos obtener un carácter específico de una cadena, indicando la posición entre **[]** (corchetes). El primer carácter de un texto empieza en la posición **0**. Ejemplo:

```
miString <= “Hola mundo”
miCaracter <= miString[0]
```

En la variable miCaracter tendremos el carácter ‘H’.

Condiciones: Podemos preguntar si dos cadenas de caracteres son:

- **Iguales**, con el operador **==**. Ej: **“Hola” == “hola”** (Resultado: **false**, porque el carácter **‘H’** es diferente a **‘h’**)
- **Diferentes**, con el operador **!=**. Ej: **“Hola” != “hola”** (Resultado: **true**)

Boolean: Este es un tipo de dato muy especial, ya que solo puede tener **dos** posibles valores (verdadero o falso):

- true
- false

Y, como se imaginarán, son las **condiciones** las que nos devuelven este tipo de dato.

Estructura de un programa

Vamos a estructurar nuestros algoritmos en papel de la siguiente manera:

Programa <nombre del programa>
Constantes
    <nombre de la constante> = <valor>
    …

Variables
    <nombre de la variable>: <tipo de dato>
    …

Comenzar
    <aquí escribiremos nuestro código>
    …
Fin

Entrada y salida de datos

Podemos pedirle datos al usuario usando **leer(**<variable>**)**
Por ejemplo, si en nuestro programa teníamos declarada la variable **edad**, y queremos que el usuario ingrese su edad, lo hacemos así: **leer(edad)**
Esto nos guardara el valor ingresado por el usuario en la variable.

Para mostrar datos por pantalla usaremos **mostrar(**<texto a mostrar>**)**.
Ejemplo: **mostrar(“Hola!”)**. Esto mostrará en pantalla el texto “Hola!”.

En el **mostrar()**, también podemos concatenar texto con variables y/o números usando la **,** (coma). Por ejemplo, si tenemos una variable **nombre** y otra variable llamada **edad**, podemos mostrar el siguiente mensaje por consola: **mostrar(“Hola. Mi nombre es “, nombre, “ y tengo ”, edad, “ años”)**

Ejercicios

1) Escribe un programa que solicite al usuario dos números enteros e imprima la suma, resta, multiplicación y división de estos dos números.

```
Programa Ejercicio1
Variables
    numero1: numérico entero
    numero2: numérico entero
    suma: numérico entero
    resta: numérico entero
    producto: numérico entero
    division: numérico real

Comenzar
    mostrar(“Ingrese dos números enteros por favor:”)
    leer(numero1)
    leer(numero2)

    suma <= numero1 + numero2
    resta <= numero1 – numero2
    producto <= numero1 * numero2
```

```
    si numero2 != 0 entonces
        division <= numero1 / numero2
        mostrar(“División: ”, division)
    finsi
    mostrar(“Suma: ”, suma)
    mostrar(“Resta: ”, resta)
    mostrar(“Producto: ”, producto)
Fin
```

2) Escribir un programa que solicite al usuario tres calificaciones (números enteros) y calcule el promedio de las tres calificaciones, mostrándolo por pantalla.

3) Escribe un programa que solicite al usuario tres números enteros y determine e imprima por pantalla el número mayor y el número menor.

Les deje nuevos archivos dentro de la carpeta de Ejercicios, para que los prueben y puedan ver todo lo que vimos en este pdf, pero en javascript: https://github.com/GabrielAlthaparro/AlthaparroGabriel/tree/main/6toB

Para programar en Javascript y probar los archivos del link de arriba vayan a la siguiente url y peguen los programas aca:
https://www.programiz.com/javascript/online-compiler/