import Foundation

//Lenguaje de programación SWIFT

//Esto es un comentario

//Puedes poner en practica a lo largo del archivo lo que quieras, para eso son lo playgrounds. Haz y deshaz tantas veces como quieras

/\*

Comentar más de

una línea utilizamos: /\* \*/

En este archivo debes recordar que se pueden utilizar en repetidas ocaciones, incluso

segmentos de código

\*/

//Variables y constantes

let MiApellido : String = "Minor"

//Esto es una declaración de una constante

let miNombre : String = "Andrés"

//Esto es una declaración de una variable, un valor que va a cambiar, esperamos que cambie en nuestro programa

// entiendase programa como una pequeña aplicación en nuestro celular o en una computadora

var miEdad : Int = 23

var miDireccion : String = "nombreDeCalle"

var miNumeroDeCasa : Int = 19

var miNumeroInterior : Int = 4

// Como ven las constantes se declara y se definen de la misma forma

let primeraCarrera : String = "Informática"

let miNacionalidad : String = "Mexicana"

/\*

Toda variable se compone de tres elementos. Sean constantes o mutables, que cambian su valor. Estos elementos son:

Modificador de acceso:

Es una palabra reservada que me indica cómo se va a reservar el espacio en memoria principal durante la ejecución de un programa.

Identificador:

El nombre del valor que voy a guardar, el nombre de ese espacio, cómo reconozco ese espacio en memoria. Un identificador es único e impermutable, por lo menos durante la ejecución de mi programa. Podemos tener muchas variables dentro de un programa y un identificador hace justo como su nombre lo dice, darle identidad a una variable, de esa forma al agregar o modificar esos espacios en memoria el procesador sabrá qué espacios en memoria corresponden a cuál variable

Tipo de dato:

El tipo de dato es el distintivo que determina cómo se va a guardar y cómo se va a acceder a los valores de diferentes variables, qué operaciones podemos realizar o se puede realizar con esos tipos de datos y cómo realizarlas

Operador de asignación:

Es el símbolo que indica que se va a asignar un valor

Valor literal:

El valor que quieres que contenga esa variable. El valor, literalmente, de algo.

\*/

//Modificador | identificador | : | Tipo de dato | Operador de asignación | y valor literal

var minutos : Int = 60

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* OPERACIONES ARITMÉTICAS \*\*\*\*\*\*\*\*\*\*\*+\*\*\*\*\*\*\*\*\*/

//Para realizar operaciones aritmeticas utilizamos operadores aritmeticos: + - \* / %

//Se les conoce como operadores binarios pues necesitan dos

//Suma

4 + 4

//Resta

4 - 4

//Division

4 / 2

//Multiplicación

4 \* 4

//Módulos. Calculan el residuo de una división: <Dividendo> % <Divisor>

5 % 2

4 % 2

81 % 4

// Operadores de Asignación y Tipos de Datos

var entero : Int = 20

var numeroDecimal : Double = 5.5

var otroNumeroDecimal : Float = 5.5//Es igual que el double, ocupa más espacio en memoria principal

var caracter : Character = "\u{00AE}"

var cadena : String = "Locuaz e impertinente"

var muchasCadenas : String = """

Recuerda poner

tres dobles comillas,

dar un salto de línea

y comenzar a escribir.

...

\(entero)

\(numeroDecimal)

\(cadena)

...

...

Al acabar cierra el texto dando un salto de línea final

con tres dobles comillas en la última línea

"""

//Tipo de dato booleano, es una bandera que indica si una condición es falsa o verdadera

var booleano : Bool = true//false es el otro valor.

entero += 5

entero -= 2

entero \*= 10

entero /= 4

/\*

Esto es un casteo de tipos para que podamos hacer operaciones entre diferentes valores con

con diferentes tipos de datos \*/

var suma : Double = numeroDecimal + Double(entero)

//Funciones sin parámetros

var operando1 : Double = 1.0

var operando2 : Double = 2.0

func sumar(){

let suma : Double = operando1 + operando2

print(suma)

}

func multiplicar(){

let multiplicar : Double = operando1 \* operando2

print(multiplicar)

}

sumar()

multiplicar()

//Funciones con parámetros

func sumar(\_ operando1 : Int,\_ operando2 : Int)->Int{

let suma : Int = operando1 + operando2

return suma

}

func restar(\_ operando1 : Int,\_ operando2 : Int)->Int{

let resta : Int = operando1 - operando2

return resta

}

func multplicar(\_ operando1 : Int,\_ operando2 : Int)->Int{

let multiplicacion : Int = operando1 \* operando2

return multiplicacion

}

func dividir(\_ operando1 : Int,\_ operando2 : Int)->Int{

let division : Int = operando1 / operando2

return division

}

sumar(3, 4)

restar(6, 7)

dividir(40, 5)

multplicar(50, 4)

sumar(Int(operando1),Int(operando2))

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Estructuras de control

Las estructuras de control...

Secuenciales

De seleccion

Iterativas, ciclos, bucles

\*/

booleano

entero = 30

if(booleano == true){

print("Verdadero")

}else{

print("Falso")

}

if(entero >= 10){

booleano = false

}

print(entero, booleano)

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Operadores relacionales ó de comparación

!= Diferente de

== Igual que

<

>

>=

<=

Operadores Lógicos

&& AND Debe cumplir ambas condiciones

|| OR Puede cumplir por lo menos una de las dos condiciones

! NOT, un valor opuesto o complemento en valores booleanos

NAND

NOR

XOR

NXOR

\*/

if(operando1 == operando2){

print("Son igualitos")

}

if(operando1 != operando2){

print("No son iguales")

}

if(operando1 <= operando2){

print("\(operando2) es menor que \(operando1)")

}

if(operando1 >= operando2){

print("\(operando1) es mayor que \(operando2)")

}

var edadActual : Double = 23.00

var laEdadAlaQuemeCaso : Double = 30.00

if( edadActual < laEdadAlaQuemeCaso){

print("\nAún no cumples \(Int(laEdadAlaQuemeCaso))\nNo te casas suertudo\n")

}else{

print("TeCasas!!!")

}

if !booleano{

print("La negación de \(booleano) es \(!booleano)\n\n")

}

print(muchasCadenas)