PRACTICA 1

var sCade: String

sCade = "Hola Mundo"

//sCade += "xD"

var sCade2 = "Hola mundo2"

let miConstante: Int=7

print(sCade)

print(sCade2)

print("El valor de la constante es: \(miConstante)")

print(miConstante)

var sCade3, sCade4, sCade5: String

var iVal = 100

var dValDoble = 5.5

print("La suma es = \(Double(iVal)+dValDoble)")

var sNom = "JOSE"

print("Mi Nombre es: \(sNom)")

var sCon = "Hola"

//sCon = sCon + "" +"mundo"

sCon += " "

sCon += "mundo"

sCon += " "

sCon += "cruel"

print(sCon)

PRACTICA 2 - WRAPPED(?) / UNWRAPPED(!)

var sCade: String?

sCade = nil

sCade = "HOLA MUNDO"

print(sCade!)

var iVal: Int?

iVal = nil

var sCade2: String!

sCade2 = nil

sCade2 = "QUIBO"

print(sCade2)

var aArreCade = ["HOLA" , "MUNDO", "CRUEL"]

print(aArreCade[0])

print(aArreCade[1])

print(aArreCade[2])

var aArreNume = Array(repeating: 10, count: 5)

print(aArreNume[3])

for iNum in aArreNume {

print(iNum)

}

for iNum2 in 0...4{

print(aArreNume[iNum2])

}

for iNum2 in 0..<aArreNume.count{

print(aArreNume[iNum2])

}

PRACTICA 3 - Ciclos

//var aArrayCade: [String]

var aArrayCade = ["HOLA","MUNDO","CRUEL"]

print(aArrayCade[0])

print(aArrayCade[1])

print(aArrayCade[2])

var aArreNume = Array (repeating: 10, count: 5 )

print(aArreNume[3])

for iNum in aArreNume{

print(iNum)

}

for iNum2 in 0...4{

print(aArreNume[iNum2])

}

for iNum2 in 0..<aArreNume.count{

print(aArreNume[iNum2])

}

PRACTICA 4

func mandaMensaje(){

print("Hola Mundo")

}

mandaMensaje()

func devuelveValor() -> Int{

return 10

}

let iVal = devuelveValor()

print(iVal)

func mandaMensaje2(mensaje: String){

print(mensaje)

}

mandaMensaje2(mensaje: "Hola mundo cruel")

func mandaMensaje3(\_ mensaje: String){

print(mensaje)

}

mandaMensaje3("Hola mundo curel 3")

func mandaMensaje4(introduceTuMensaje mensaje: String){

print(mensaje)

}

mandaMensaje4(introduceTuMensaje: "USO DE ALIAS")

PRACTICA 5 – Clases y Herencia

class MiClase{

init(){

print("CONSTRUCTOR SIN ARGUMENTOS")

}

func mandarMensaje(\_ mensaje: String){

print(mensaje)

}

}

let mcObj = MiClase()

mcObj.mandarMensaje("Holis mundo")

class MiOtraClase: MiClase{

}

let mocObjeto = MiOtraClase()

mocObjeto.mandarMensaje("OTRO HOLA MUNDO")

class MiOtraOtraClase: MiClase{

override init(){

print("OTRO INITIALIZER")

super.init()

}

}

let moocObjeto = MiOtraOtraClase()

PRACTICA 6 – Protocol > Interfaces

//protocol --> interfaces

protocol hacerAlgo {

func mandarMensaje()

}

class ImplementarProtocolo: hacerAlgo{

init(){

print("SIN ARGUMENTOS")

}

init(mensaje: String){

print(mensaje)

}

init(mensaje: String, veces: Int){

for \_ in 0...veces{

print(mensaje)

}

}

func mandarMensaje(){

print("HOLA MUNDO")

}

}

let ipObj = ImplementarProtocolo(mensaje: "HOLA", veces: 5)

ipObj.mandarMensaje()