

# Variables y Propiedades

## Declaración de una variable

Para declarar una variable se debe utilizar la palabra clave 'var' seguida del nombre de la variable, su tipo de dato y valor. El valor inicial es opcional.

```
var num:Int = 10
```

Las variables pueden cambiar de valor en cualquier momento, ya sea por el flujo del programa o por una asignación directa.

```
num = 20    // La variable ahora vale 20
```

Si se desea una variable que no cambie su valor, es decir una constante, se debe utilizar la palabra clave **let** seguido de su nombre, tipo de dato y valor. El valor inicial (y permanente) no es opcional.

```
let num:Int = 10    // Este valor no puede cambiar
```

Es posible declarar una variable y asignarle un valor sin indicar el tipo de dato. Si no se especifica un tipo de dato, Swift inferirá y asignará el tipo de manera automática.

```
let ten = 10 // num is an Int  
let pi = 3.14 // pi is a Double  
let floatPi: Float = 3.14 // floatPi is a Float
```

Los nombres de las variables y constantes no están limitados a caracteres alfanuméricos, pueden contener la mayoría de caracteres *Unicode* aunque existen algunas excepciones. Los nombres no pueden contener espacios, símbolos matemáticos, flechas, símbolos *Unicode* privados y no pueden comenzar con un número.

```
var π: Double = 3.14159  
var 🍏: String = "Apples"
```

## Tipos de datos

Los tipos de datos son una parte muy importante de cualquier lenguaje de programación.

### Enteros

Se utiliza para representar números sin parte fraccionaria. En Swift se pueden usar enteros con signo o sin signo de 8, 16, 32 y 64 bits. Se debe utilizar la palabra clave **Int** al declarar una variable.

Tipo	Tamaño	Rango
Int8	8 bits	[-128, 128]
Int16	16 bits	$[-2^{15}, 2^{15}-1]$
Int32	32 bits	$[-2^{31}, 2^{31}-1]$
Int64	64 bits	$[-2^{63}, 2^{63}-1]$
UInt32	Depende la plataforma	$[0, 2^{32}]$
UInt64	Depende la plataforma	$[0, 2^{64}]$

## Ejemplo

```
// Swift program to demonstrate integer datatype

// Creating signed integer data types
var digit1: Int = 10

// Display the number
print("Signed integer data type is", digit1)

// Creating unsigned integer data type
var digit2: UInt = 23

// Display the number
print("Unsigned integer data type is", digit2)
```

## String

Cadena de caracteres, se declara con la palabra clave **String**.

```
// Swift program to demonstrate string datatype

// Creating string data type
var inputdata1: String = "GeeksforGeeks"

// Display the value
print("String data type is", inputdata1)

// Creating string data type
var inputdata2: String
inputdata2 = "GFG"

// Display the value
print("String data type is", inputdata2)
```

## Float

Para representar números no enteros utilizando 32 bits de memoria. El tipo de dato permite números de hasta 6 decimales, el rango de números que es posible almacenar son  $[1.2E^{-38}, 3.45E^{38}]$ . Se utiliza la palabra clave ``+Float`` en la declaración.

```
// Swift program to demonstrate float datatype

// Creating float data type
var inputdata1: Float = 3.0545

// Display the value
print("Float data type is", inputdata1)

// Creating double data type
var inputdata2: Float = 0.978623

// Display the value
print("Float data type is", inputdata2)
```

## Double

Números no enteros de doble precisión, se emplean 64 bits de memoria. Se pueden representar números de hasta 15 decimales. El rango de números posibles es  $[2.3E^{-308}, 1.7E^{308}]$ . Se utiliza la palabra clave ``+Double`` en la declaración.

```
// Swift program to demonstrate double datatype

// Creating double data type
var inputdata1: Double = 23.098545

// Display the value
print("Double data type is", inputdata1)

// Creating double data type
var inputdata2: Double = 1.9786677532

// Display the value
print("Double data type is", inputdata2)
```

## Booleano

Valores lógicos `true` y `false`. Se utiliza la palabra clave `Bool` en la declaración.

```
// Swift program to demonstrate boolean datatype

// Creating boolean data type
```

```
var inputdata1: Bool = true

// Display the value
print("Boolean data type is", inputdata1)

// Creating boolean data type
var inputdata2: Bool = false

// Display the value
print("Boolean data type is", inputdata2)
```

## Character

Se utiliza para almacenar un solo caracter. Se utiliza la palabra clave **Character** en la declaración.

```
// Swift program to demonstrate character datatype

// Creating character data type
var inputdata1: Character = "G"

// Display the value
print("Character data type is", inputdata1)

// Creating character data type
var inputdata2: Character = "E"

// Display the value
print("Character data type is", inputdata2)
```

[Anterior](#) | [Inicio](#) | [Siguiete](#)