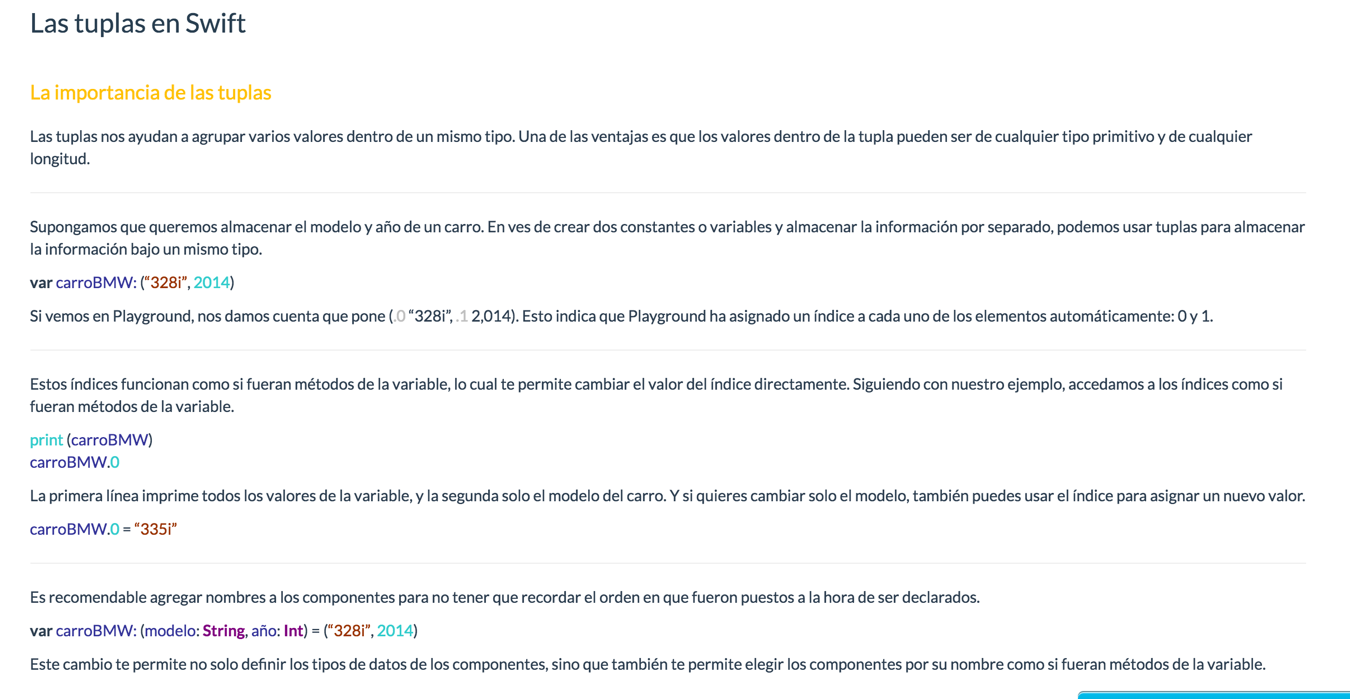
Curso Swift





Strings y sus funciones

Repasemos rápidamente algunas de las diferentes funciones que podemos utilizar para manipular **Strings** en Swift.

**Concatenar**

Como hemos aprendido, podemos utilizar el operador de la suma para combinar o concatenar dos Strings.

**// Dos literales de String**  
let valor = “gato ”  
let valor2 = “y perro”  
  
**// Concatenar dos Strings**  
let valor3 = valor + valor2  
print(valor3)  
  
**Resultado**  
gato y perro

**Agregar**

Una cadena que se declara como variable puede ser reasignada.

**// Esta cadena puede cambiar**  
var valor = “gato ”  
  
**// Agregar información a la variable String**  
valor += “…perro”  
print(valor)  
  
**Resultado**  
gato…perro

**Cadena vacía, isEmpty**

Para crear una cadena vacía, utilizamos **String()** para inicializarla. Y usamos **isEmpty** (devuelve los valores *true* o *false*) para comprobar si esta vacía.

 var cadena = String()  
  
**// Esta cadena está vacía**  
if cadena.isEmpty {

print(“No hay caracteres en la cadena”)

}  
  
**// Ahora isEmpty va a retornar falso**  
cadena = “America”

if !cadena.isEmpty {

print(“No está vacía”)

}  
  
**Resultado**  
No hay caracteres en la cadena  
No está vacía

**Contar caracteres**

Para obtener el número de caracteres en una cadena usamos **.count**. Todos los caracteres, incluyendo los espacios y las puntuaciones serán contadas por esta propiedad.

**// Esta cadena tiene 6 caracteres**  
var valor = “ho la\_”  
var largo = valor.characters.count  
print(largo)  
  
**Resultado**  
6

**Interpolación de cadenas**

La interpolación se utiliza para insertar variables, como enteros o incluso otras cadenas, en un lugar determinado de la cadena original.

let nombre = “Swift”  
let version = 2  
  
**// Crear una cadena con los valores de las variables**  
let resultado = “Programa: \(nombre), Version: \(version)”  
print(resultado)  
  
**Resultado**  
Programa: Swift, Version: 2

**Subíndice**

Para poder acceder un carácter en una cadena, utilizamos **startIndex** y **endIndex**, y usamos rangos para poder obtener subcadenas con el **advancedBy()**.

En el siguiente ejemplo vamos a crear un nuevo rango a partir de la cadena original y vamos a llamar el método **advancedBy()** usando el **startIndex** y el **endIndex**.

**// Esta es la cadena original**  
       let valor = “Aprendiendo Swift, el nuevo programa de Apple.”  
  
**// Obtener el rango**  
       let resultado = Range<String.Index>(start: valor.startIndex.advancedBy(12), end: valor.endIndex)  
  
**// Acceder la cadena usando el rango**  
let subcadena = valor[resultado]  
print(subcadena)  
  
**Resultado**  
Swift, el nuevo programa de Apple.

También  podemos llamar directamente el método substringWithRange para obtener la subcadena

**// Acceder la cadena usando el rango**  
let subcadena2 = valor.substringWithRange(resultado)  
print(subcadena2)

