

PRESENTACIÓN

Desarrollo de Aplicaciones Móviles

AL FINALIZAR EL CURSO...

- Podrás desarrollar una aplicación móvil para iOS con una complejidad intermedia/alta.

EVALUACIÓN

- 70 % Practicas
- 30% Proyecto

¿QUÉ HERRAMIENTAS VAMOS A USAR?

- Lenguaje de programación Swift
- Playground
- IDE de desarrollo Xcode



LENGUAJE SWIFT

- Swift es un lenguaje de programación creado por Apple para desarrollar apps de iOS, Mac, Apple TV y Apple Watch.

Primeros pasos con Swift.



El libro *The Swift Programming Language*

Descarga *The Swift Programming Language* del Apple Books Store gratis. Descubre cómo Swift hace que programar sea más sencillo, flexible y divertido.

[Descarga el libro gratis >](#)



Xcode

Xcode es la app de Mac que se utiliza para crear las demás apps para Mac, así como las apps de iOS, pues tiene todas las herramientas que necesitas para desarrollar apps sorprendentes. Además, puedes descargarla gratis del Mac App Store.

[Descarga Xcode del Mac App Store >](#)



Sitio web para desarrolladores

Obtén información técnica y una descripción detallada de Swift. Consulta el blog de desarrolladores y mantente al día con las últimas noticias. Y accede a recursos gratis, como guías, videos y ejemplos de códigos.

[Más información >](#)

Hasta

2.6 veces

más rápido que
Objective-C

Hasta

8.4 veces

más rápido que
Python 2.7

10,000 números enteros encontrados
en un gráfico mediante un algoritmo
de búsqueda en profundidad*

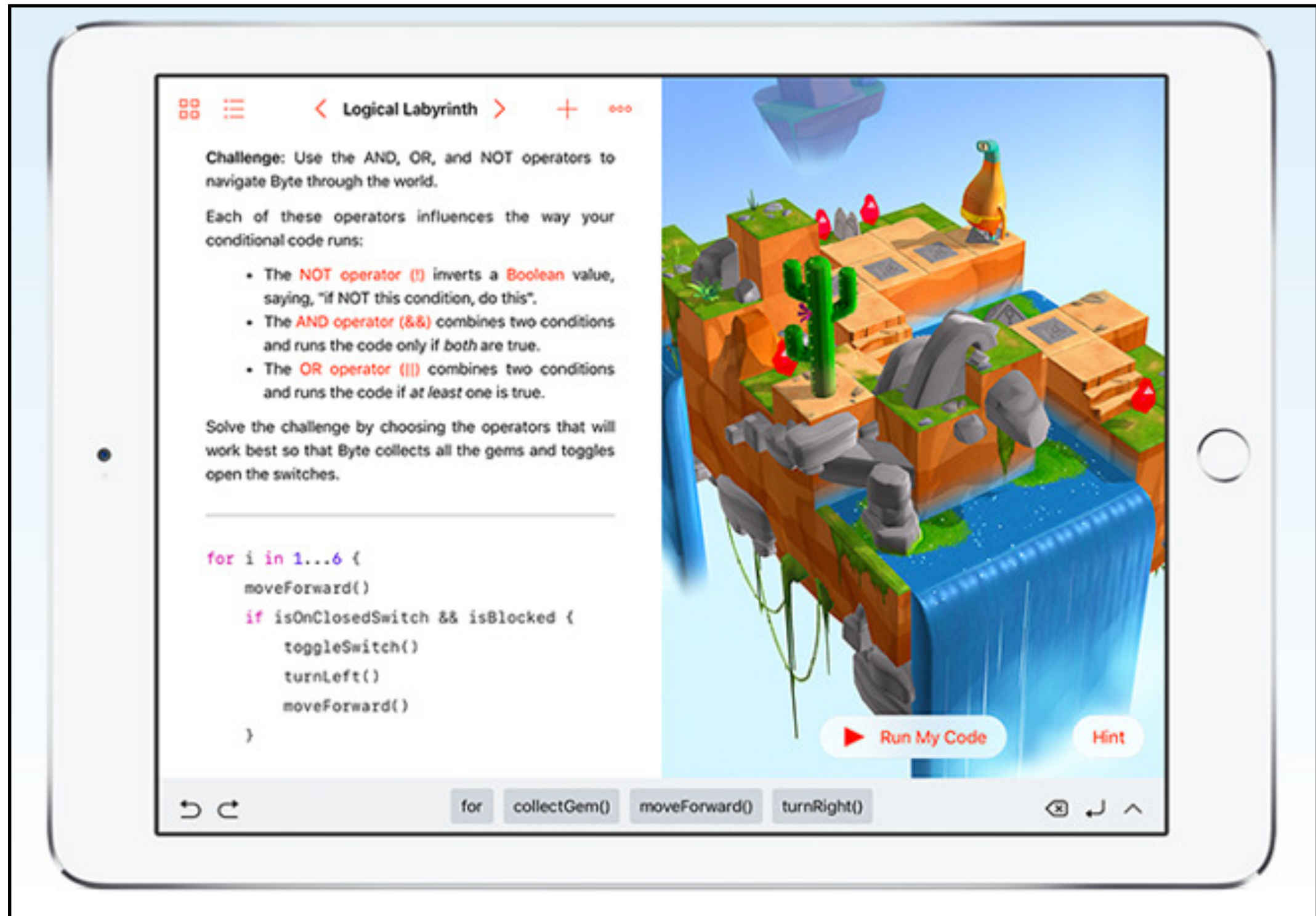


*Cuando algo no puede evolucionar más en un nivel, tal vez su
mejora esté en mejorar sus bases.*

LENGUAJE SWIFT

- **Swift- última versión estable: 5.1.3 (Septiembre 2019)**
 - Disponer de una computadora Mac con sistema operativo OS X 11, con el software de desarrollo Xcode, versión 10.2
 - Sistema operativo Linux Ubuntu, se recomienda usar el editor de texto enfocado en desarrollo Atom, de la web GitHub (gratis).
 - Swift community. <https://swift.org>
 - Online Swift Playground —
<http://online.swiftplayground.run>

► Aplicación para iPad: Swift Playgrounds.



PYPL . – [HTTPS://PYPL.GITHUB.IO/PYPL.HTML](https://pypl.github.io/pypl.html)

.....

PYPL PopularitY of Programming Language

Worldwide, Aug 2019 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	28.73 %	+4.5 %
2		Java	20.0 %	-2.1 %
3		Javascript	8.35 %	-0.1 %
4		C#	7.43 %	-0.5 %
5		PHP	6.83 %	-1.0 %
6		C/C++	5.87 %	-0.3 %
7		R	3.92 %	-0.2 %
8		Objective-C	2.7 %	-0.6 %
9		Swift	2.41 %	-0.3 %
10		Matlab	1.87 %	-0.3 %
11	↑	TypeScript	1.76 %	+0.2 %
12	↓	Ruby	1.44 %	-0.2 %

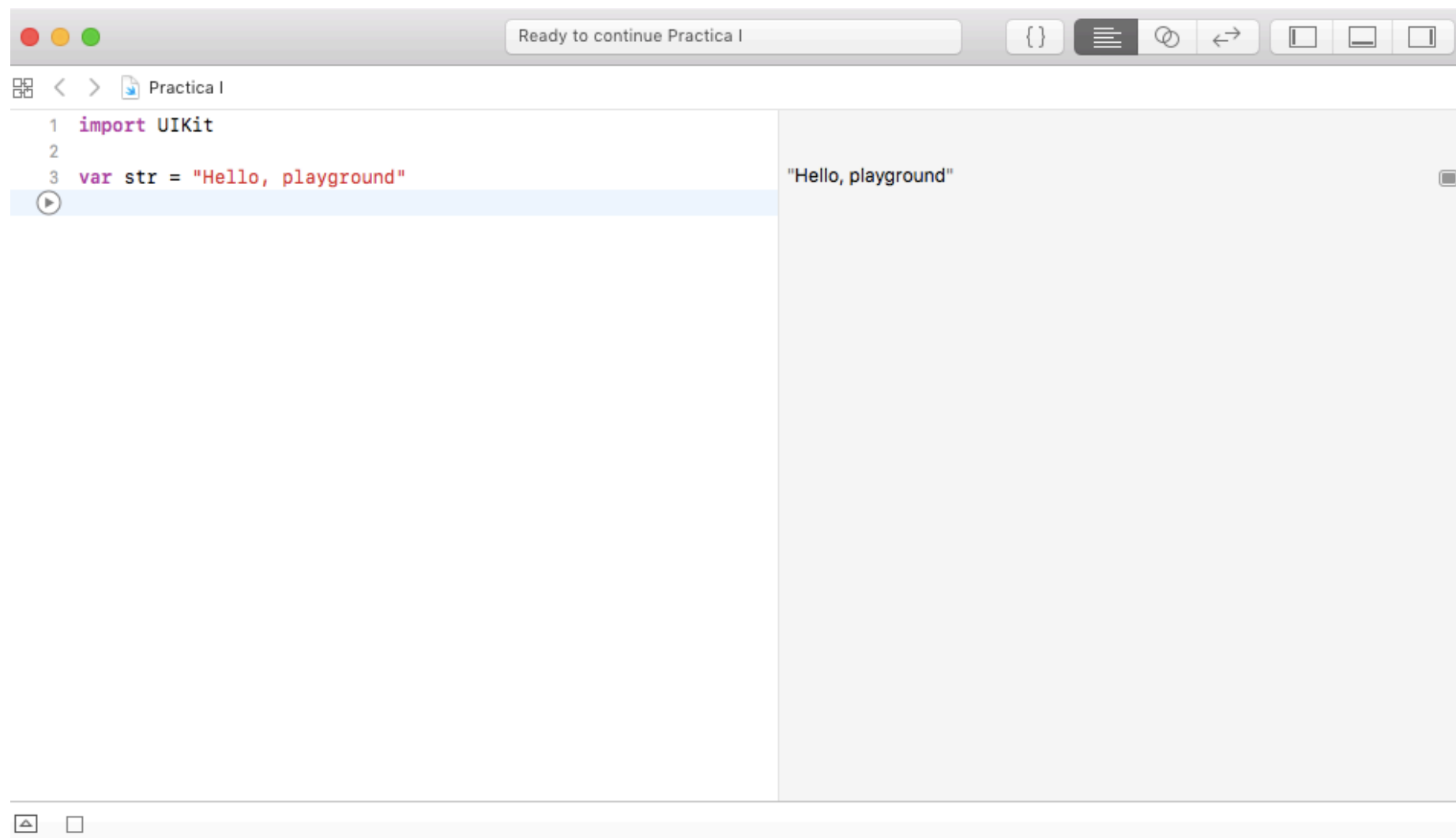
The PYPL PopularitY of Programming Language Index is created by analyzing how often language tutorials are searched on Google.

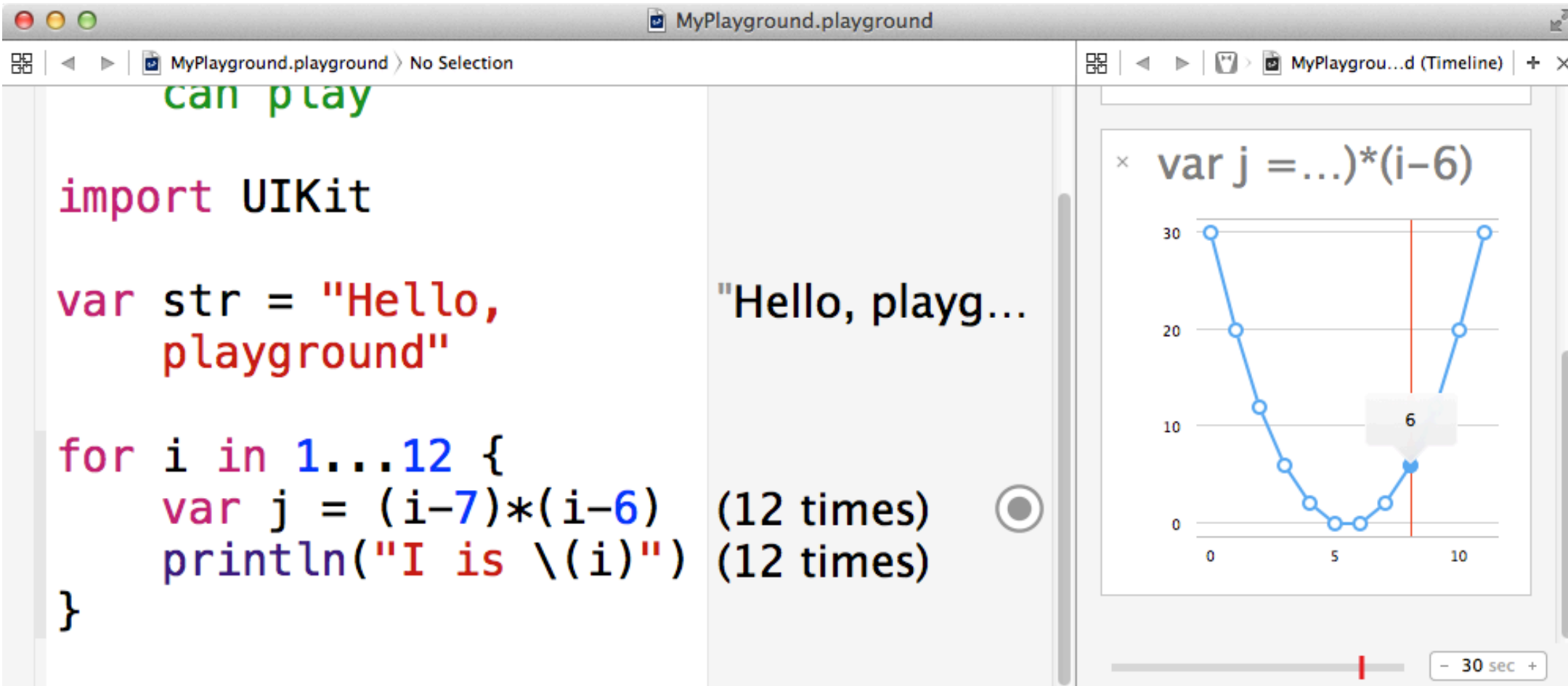
The more a language tutorial is searched, the more popular the language is assumed to be. It is a leading indicator. The raw data comes from Google Trends.

If you believe in collective wisdom, the PYPL Popularity of Programming Language index can help you decide which language to study, or which one to use in a new software project.

PLAYGROUND

- Apple creó para Swift, una herramienta que va evaluando, compilando el resultado y mostrando posibles mensajes de error o advertencias en tiempo real. Es útil para construir prototipos, un concepto o una prueba.

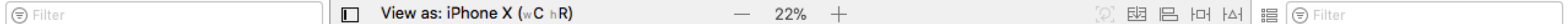




IDE – XCODE

- Entorno de desarrollo integrado de Apple Inc., se suministra gratuitamente con Mac Os.
- Incluye la colección de compiladores GNU, se puede compilar código en C, C++, Swift, Objective-C, Objective C++.
- Última versión de Xcode 10.14 (Mayo 2019), Xcode 11 Beta.





LENGUAJE SWIFT

► Primera regla:

Olvida el ;

CONSTANTES Y VARIABLES

```
var a = 7.7
var b = 42
var c = 20
b = 90
b = b + c
c = a + b

let d = 10
d = 12
```

➤ Comentando código:

```
// Comentario en una línea  
let ab = 10  
let ac = 20 // Comentario intra-línea
```

➤ Imprimiendo cosas:

```
var numero = 10  
var otro_numero = 20  
var cadena = "Swift"  
  
print(numero)  
print(otro_numero)  
print(numero, otro_numero, cadena)
```

```
print("\(cadena) es un \(numero)")  
print("Le damos un \(numero) a \(cadena)")
```

VARIABLES Y CONSTANTES

- Se usa la palabra reservada `let` para declarar constantes y `var` para declarar variables.

```
let name = "Mrs. Pop Star"  
let birthDate = "January 1, 1991"  
let homeTown = "Nashville"  
let firstHitSong = "Party Tonight"  
  
let name = "Mr. Athlete"  
let heightInFeet = 6.4  
let rookieYear = 1997  
let nickname = "Sparky"
```

ASIGNACIÓN IMPLÍCITA Y EXPLÍCITA

- No es necesario especificar el tipo de variable o constante declarado.

```
var a: Double = 7.7
var b: Int = 42
var c = 20
var d = "Hola"
var e: String = " Swift"

var f = d + e
```

CONVERSIÓN DE TIPOS

```
let label = "The width is "  
let width = 94  
let widthLabel = label + String(width)
```

```
let va = 20.0  
let vb = Float(20.0)
```


INCLUSIÓN DE VALORES EN LOS STRINGS

```
let apples = 3
let oranges = 5
let appleSummary = "I have \$(apples) apples."
let fruitSummary = "I have \$(apples + oranges) pieces of
                    fruit."
```

IF - ELSE

```
let x = 2
if x < 2 {
  print("Menor de 2")
} else {
  print("2 o mayor")
}
```

```
let y = 3
let z = 4

if x < 2 {
  print("Menor de 2")
} else if y > 3 {
  print("Mayor de 3")
} else if z == 4 {
  print("Es 4")
}
```

```
let temperatura = 22
if temperatura > 40 {
  print("¡Qué calor!")
} else if temperatura > 30 {
  print("Empieza a hacer calor")
} else if temperatura > 20 {
  print("Fresquito moderado")
} else if temperatura > 10 {
  print("Hace frío moderado")
} else {
  print("Frío tal cual")
}
```

SWITCH

```
let nota = 8
switch nota {
case 0: print("Absolutamente deficiente")
case 1, 2: print("Muy deficiente")
case 3, 4: print("Suspenso")
case 5: print("Suficiente")
case 6: print("Bien")
case 7, 8: print("Notable")
case 9: print("Sobresaliente")
case 10: print("Matrícula")
default: print("Nota no evaluable")
}
```

```
let temperatura = 24
switch temperatura {
case 31...40:
    print("¡Qué calor!")
case 21...30:
    print("Empieza a hacer calor")
case 11...20:
    print("Fresquito moderado")
default:
    print("Frío tal cual")
}
```

CICLO WHILE Y REPEAT WHILE

```
var n = 2
while n < 100{
    n = n * 2
}
print(n)
```

```
var m = 2
repeat{
    m = m * 2
}while m < 100
print(m)
```

CICLO FOR IN

```
1 let names = ["Anna", "Alex", "Brian", "Jack"]
2 for name in names {
3     print("Hello, \(name)!")
4 }
5 // Hello, Anna!
6 // Hello, Alex!
7 // Hello, Brian!
8 // Hello, Jack!
```

```
1 let numberOfLegs = ["spider": 8, "ant": 6, "cat": 4]
2 for (animalName, legCount) in numberOfLegs {
3     print("\(animalName)s have \(legCount) legs")
4 }
5 // ants have 6 legs
6 // cats have 4 legs
7 // spiders have 8 legs
```

CICLO FOR IN

```
1  for index in 1...5 {
2      print("\(index) times 5 is \(index * 5)")
3  }
4  // 1 times 5 is 5
5  // 2 times 5 is 10
6  // 3 times 5 is 15
7  // 4 times 5 is 20
8  // 5 times 5 is 25
```

```
1  let base = 3
2  let power = 10
3  var answer = 1
4  for _ in 1...power {
5      answer *= base
6  }
7  print("\(base) to the power of \(power) is \(answer)")
8  // Prints "3 to the power of 10 is 59049"
```



```
1 let minutes = 60
2 for tickMark in 0..
```

```
1 let minuteInterval = 5
2 for tickMark in stride(from: 0, to: minutes, by:
3     minuteInterval) {
4     // render the tick mark every 5 minutes (0, 5, 10, 15
5         ... 45, 50, 55)
6 }
```

Closed ranges are also available, by using `stride(from:through:by:)` instead:

```
1 let hours = 12
2 let hourInterval = 3
3 for tickMark in stride(from: 3, through: hours, by:
4     hourInterval) {
5     // render the tick mark every 3 hours (3, 6, 9, 12)
6 }
```

```
let individualScores = [75, 43, 103, 87, 12]
var teamScore = 0
for score in individualScores {
    if score > 50 {
        teamScore += 3
    } else {
        teamScore += 1
    }
}
println(teamScore)
```

TAREA 1:

- Investigar la sintaxis para declarar funciones en Swift
- Crea una función que devuelva la suma y multiplicación de los 10 números enteros almacenados en un arreglo.
- En una tienda efectúan un descuento a los clientes dependiendo del monto de la compra. El descuento se efectúa con base en los siguientes criterios:
 - Si el monto es menor a 500 no hay descuento.
 - Si el monto está entre \$500 y \$1,000 hay 5% de descuento.
 - Si el monto está entre \$1,000 y \$7,000 hay 11% de descuento.
 - Si el monto está entre \$7,000 y \$15,000 hay 18% de descuento.
 - Si el monto es mayor a \$15,000 hay un 25% de descuento.
 - Imprimir en pantalla el descuento y el costo total de la compra.