{desafío}
latam_

Introducción a Kotlin _

Parte I

Kotlin

Kotlin

- Lenguaje creado el año 2010 por JetBrains
- 2017 es oficialmente soportado por Android.
- 2019 anunciado por Google como lenguaje principal para Android.
- Trabaja sobre la JVM de Java
- Soporta Kotlin Native, Android, Java, JavaScript

2017: Kotlin officially supported for Android
2019: Android is Kotlin first
🙌💙💜❤️💛

Instalación

Requisitos mínimos para la instalación

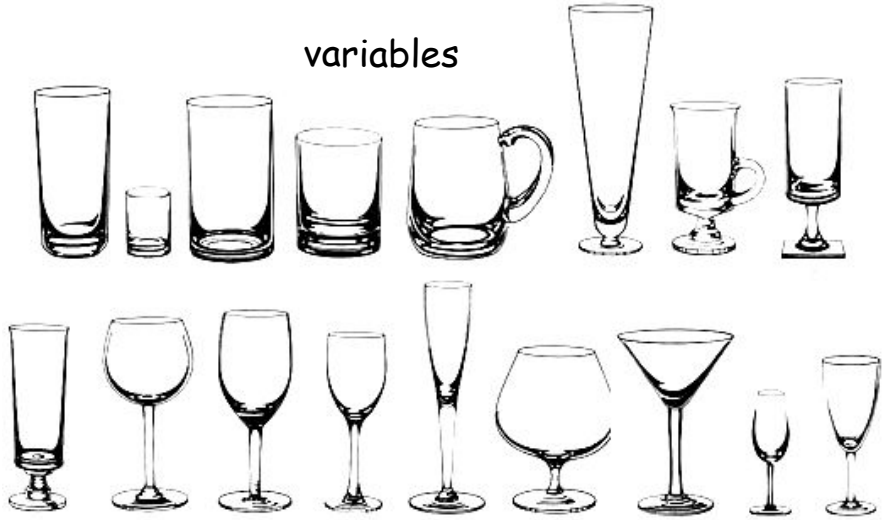
System requirements

Requirement	Minimum	Recommended
RAM	2 GB of free RAM	8 GB of total system RAM
Disk space	1.5 GB and another 1 GB for caches	SSD drive with at least 5 GB of free space
Monitor resolution	1024x768	1920×1080
Operating system	Microsoft Windows 7 SP1 or later	Latest 64-bit version

Variables

Variables en Kotlin

Cuándo piensen en **variables** imaginate que estas son vasos, existen distintos tipos de vasos, de distinta forma y distinto color, pero todas tienen algo en común, un vaso contiene algo.



Declaración de Variables

Para poder declarar una variable en Kotlin, necesitamos saber 3 cosas:

- Tipo de variable (“var” o “val”)
- Nombre de la variable
- Tipo de valor o Clase(Numérico, Cadena de Caracteres, Lógico, etc.)

```
fun main() {
  val numero : Int
  var nombreVariable : TipoValor
}
```

Declaración de Variables Inteligentes

En Kotlin podemos omitir la declaración del tipo de valor, y esto lo hacemos inicializando la variable con un valor determinado.

```
fun main() {
  val nombre = “Goro Daimon”
  val edad = 22
}
```

Diferencias entre un val y var

Las variables inmutables se declaran como “**val**” y contienen un valor que “No” puede ser modificado.
Las variables mutables se declaran como “**var**” y estas variables “Si” pueden ser modificadas.

```
fun main() {
  val numeroFijo = 10
  var numeroVariable = 10
  numeroFijo = 12
}
```

String Template en Kotlin

Una de las mejoras importantes que trae Kotlin como lenguaje de programación, es la facilidad de trabajar con variables String, a esto se les conoce como "**String Template**".

Cuando estemos utilizando una variable String podemos utilizar el signo "**$**" para concatenar valores.

```
fun main() {
   val x = "valor string"
   val template = "Concatenando **** $x **** valores"
   println(template)
}
```

String Template en Kotlin

También podemos llamar a una función del lenguaje o de una variable utilizando el signo "**$**" seguido del uso de llaves "**{ }**". Dentro de estas llaves podemos hacer la llamada a la función que necesitemos.

```
fun main() {
    val x = "valor string"
    val template = "Tiene ${x.count()} caracteres"
    println(template)
}
```

Variables Básicas en Kotlin

Variables Enteras

En Kotlin existen 4 tipos de variables básicas para trabajar con números, y estos son: **Byte**, **Short**, **Int** y **Long.**

Tipo	Tamaño	Rango de valor
Byte	8 bits	-128 a 127
Short	16 bits	-32768 a 32767
Int	32 bits	-2147483648 a 2147483647
Long	64 bits	-9223372036854775808 a 9223372036854775807

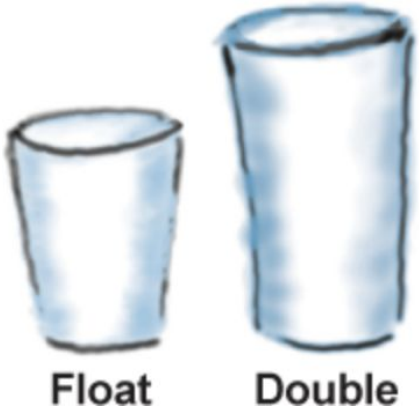
Variables Enteras

En cada clase Byte, Short, Int y Long existe una constante llamada MAX_VALUE y MIN_VALUE.

```
fun main() {
    val byteMaximo = Byte.MAX_VALUE
    val byteMinimo = Byte.MIN_VALUE
    val shortMinimo = Short.MIN_VALUE
    val shortMinimo = Short.MIN_VALUE
    val intMinimo = Int.MIN_VALUE
    val intMinimo = Int.MIN_VALUE
    val longMinimo = Long.MIN_VALUE
    val longMinimo = Long.MIN_VALUE
}
```

Variables de coma flotante

En Kotlin existen 2 tipos de variables básicas para trabajar con valores de coma flotante estos son Float y Double.




```kotlin
fun main() {
    val x = 19.5f
    var y : Double = 20.5
}
```

Variables booleanas

Las variables booleanas almacenan sólo dos valores, “verdadero” o “falso” .

```
fun main() {
   val x = true
   val y : Boolean
   y = false
}
```

Cadenas de caracteres

En Kotlin existen dos tipos básicos para trabajar con cadenas de caracteres, String y Char.

```
fun main() {
   val x = 'a'
   val y = "Cadena de caracteres"
   var vocal : Char
   val nombre : String
   vocal = 'u'
   nombre = "Chizuru Kagura"
}
```

Tipos de Funciones

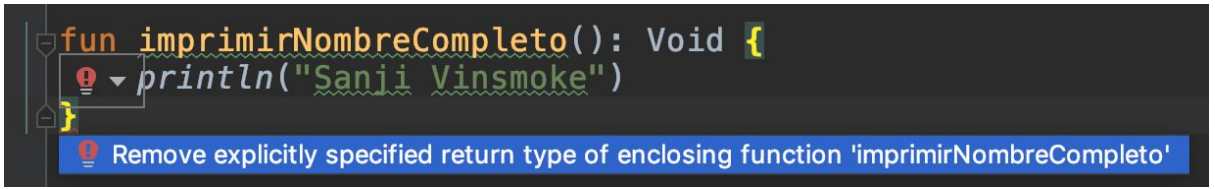
Funciones en Kotlin

En Kotlin, solo existen los métodos con tipo de retorno y a estos métodos se les llama funciones.
Se declaran de la siguiente forma:

```
fun getNombreCompleto(): String{
   return "Monkey D. Luffy"
}
```

Funciones void en Kotlin

En Kotlin no existen la palabra reservada “**void**”, sólo existe la clase “**Void**”, pero si le indicamos a una función que retornará un objeto tipo “Void” nos llevaremos una sorpresa.



Me dice que remueva explícitamente el “Void” especificado como tipo de retorno.

Clase Unit en Kotlin

La clase **Unit** es el equivalente en Kotlin a la variable primitiva "**void**" de Java.

```
fun imprimirNombreCompleto(): Unit {
   println("Sanji Vinsmoke")
}
```

Si declaramos explícitamente que nuestra función retorna un tipo "Unit", el mismo lenguaje te sugiere que no lo agregues.

Funciones con parámetros

La forma de declarar parámetros en una función Kotlin es la siguiente:

```
fun imprimirNombreCompleto(nombres: String, apellidos: String, edad: Int){
   println("Nombres: $nombres")
   println("Apellidos: $apellidos")
   print("Edad: $edad")
}
```

Función Main en Kotlin

Una de las funciones más importantes a nivel de programación es el método main.

El método main es tan importante, que si desarrollamos una aplicación y no creamos el método main, no podemos ejecutar nuestra aplicación.

En Kotlin a los métodos les llamamos funciones, y la función main no es la excepción.

```
fun main(){
   println("Hola Mundo, soy la función Main y no soy estática.")
}
```

{desafío} latam_

Academia de talentos digitales

www.desafiolatam.com