

**1**

Introducción y Kotlin Esencial



|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Alumno(s):*** | ***Nina Mayta Milward Fernando*** | | | | | | ***Nota*** |  | |
| ***Grupo:*** | ***D*** | | ***Ciclo: 4*** | | | | | | |
| ***Criterio de Evaluación*** | | ***Excelente (4pts)*** | | ***Bueno (3pts)*** | ***Requiere mejora (2pts)*** | ***No acept. (0pts)*** | | | ***Puntaje Logrado*** |
| Desarrolla adecuadamente los ejercicios teóricos | |  | |  |  |  | | | ***3*** |
| Desarrolla adecuadamente los ejercicios prácticos | |  | |  |  |  | | | ***6*** |
| Realiza una correcta instalación de Android Studio | |  | |  |  |  | | | ***3*** |
| Realiza observaciones y conclusiones que aporten un opinión crítica y técnica | |  | |  |  |  | | | ***3*** |
| Es puntual y redacta el informe adecuadamente sin copias de otros autores | |  | |  |  |  | | | ***2*** |
| Evidencia avance en laboratorio | |  | |  |  |  | | | ***3*** |

**Laboratorio 01: Introducción y Kotlin Esencial**

1. **Objetivo**

* Comprender los diversos tipos de variables en Kotlin y cómo declararlas.
* Utilizar la función println para mostrar información en la consola en Kotlin.
* Crear funciones en Kotlin con argumentos y valores de retorno.
* Utilizar estructuras de control para tomar decisiones.

1. **Seguridad**

* No ingresar con líquidos, ni comida al aula de Laboratorio.
* Al culminar la sesión de laboratorio apagar correctamente la computadora y la pantalla, y ordenar las sillas utilizadas.

1. **Equipos y Materiales**

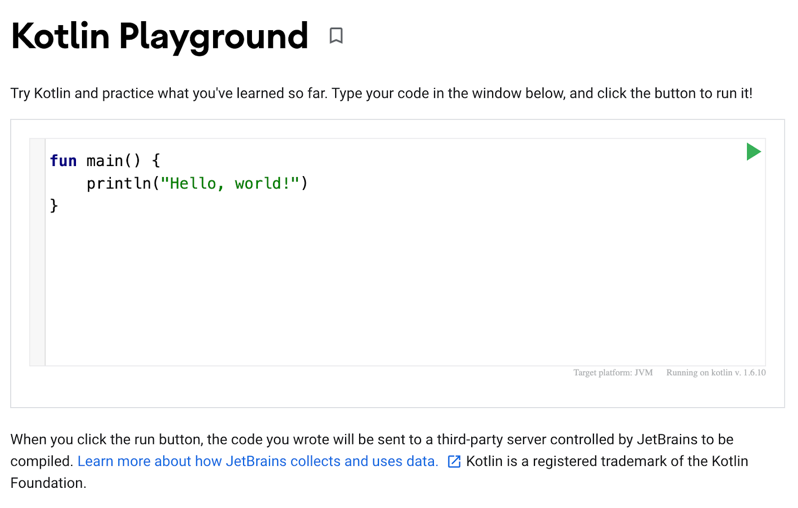
* Sistema Operativo Windows 10 o superior con conexión a la red del laboratorio
* Instalador de Android Studio

**DESARROLLO**

### Hola Mundo en Kotlin

En un navegador web de tu computadora, abre [Playground de Kotlin](https://developer.android.com/training/kotlinplayground?hl=es-419).

Deberías ver una página web similar a esta imagen:



Ya existe un código predeterminado propagado en el editor de código. Estas tres líneas de código conforman un programa simple:

|  |  |
| --- | --- |
| 1  2  3 | **fun** main() {  println("Hello, world!")  } |

Haz clic en Botón Ejecutar RUN para ejecutar el programa.

Cuando haces clic en el botón Ejecutar, suceden muchas cosas. El objetivo del código en el lenguaje de programación Kotlin es que los humanos lo entiendan, de modo que las personas puedan leer, escribir y colaborar más fácilmente en programas de Kotlin. Sin embargo, la computadora no comprende este idioma de inmediato.

Necesitas algo llamado el *compilador de Kotlin*, que toma el código Kotlin que escribiste, lo analiza línea por línea y lo traduce a algo que la computadora puede comprender. Este proceso se denomina *compilación* de código.

Si tu código se compila correctamente, el programa se ejecutará. Cuando la computadora ejecuta tu programa.

Muestra el resultado del código ejecutado:

|  |
| --- |
|  |

### Cuerpo de la función

El *cuerpo de la función* contiene las instrucciones necesarias para lograr el propósito de la función. Para buscar el cuerpo de la función, busca las líneas de código encerradas entre llaves de apertura y cierre.



## Explicación de un programa sencillo

Observa el programa sencillo que viste antes en el codelab.

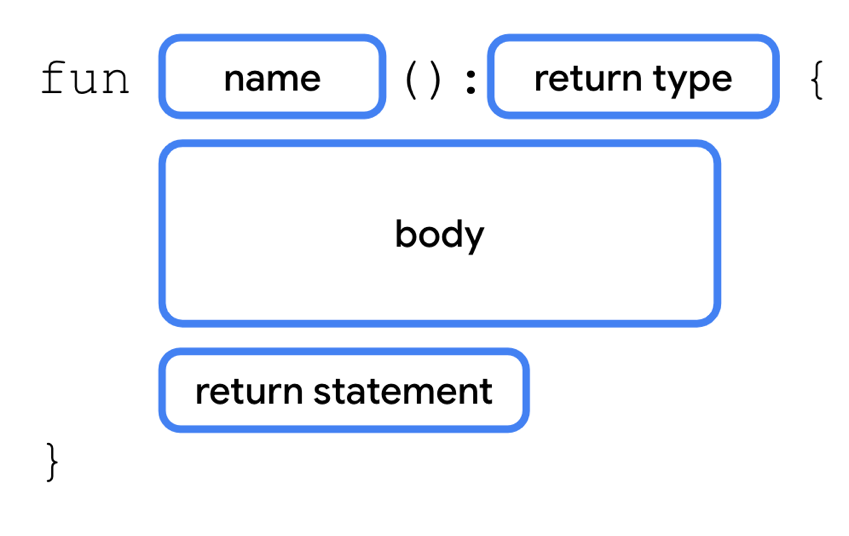


El programa contiene una función: la función principal. "main" es un nombre de función especial en Kotlin. Cuando escribes tu código en el Playground de Kotlin, debes escribirlo dentro de la función main() o llamarlo desde la función main().

A continuación, se incluyen algunas de las recomendaciones de la guía de estilo para lo que aprendiste en Kotlin hasta el momento:

* Los nombres de las funciones deben seguir la convención de mayúsculas y minúsculas, y deben ser verbos o frases verbales.
* Cada sentencia debe estar en su propia línea.
* La llave de apertura debe aparecer al final de la línea donde comienza la función.
* Debe haber un espacio antes de la llave de apertura.

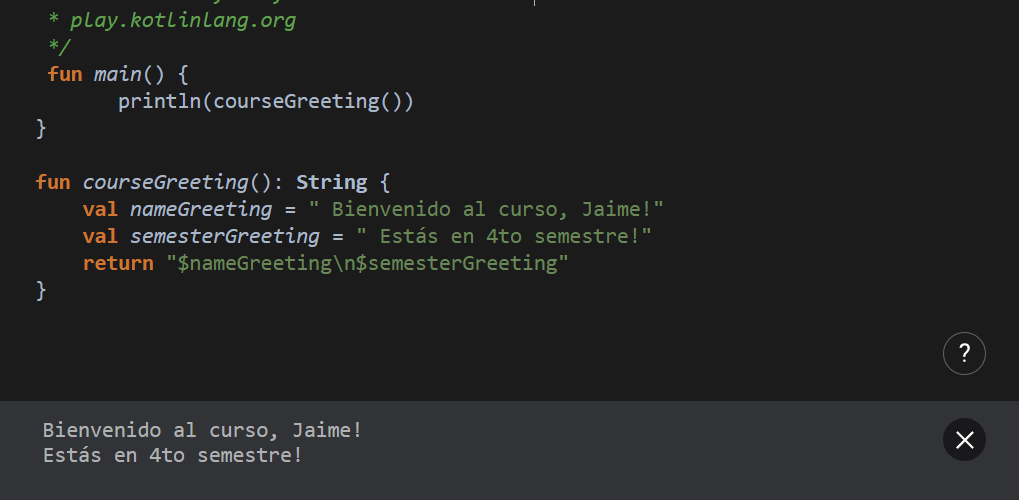
La sintaxis para declarar una función con un tipo de datos que se devuelve es la siguiente.



De forma predeterminada, si no se especifica un tipo de datos que se muestra, el predeterminado es Unit. Unit significa que la función no muestra ningún valor.

A fin de demostrar cómo una función puede mostrar un valor, deberás realizar la función courseGreeting() para que muestre una string, en lugar de simplemente imprimir el resultado.

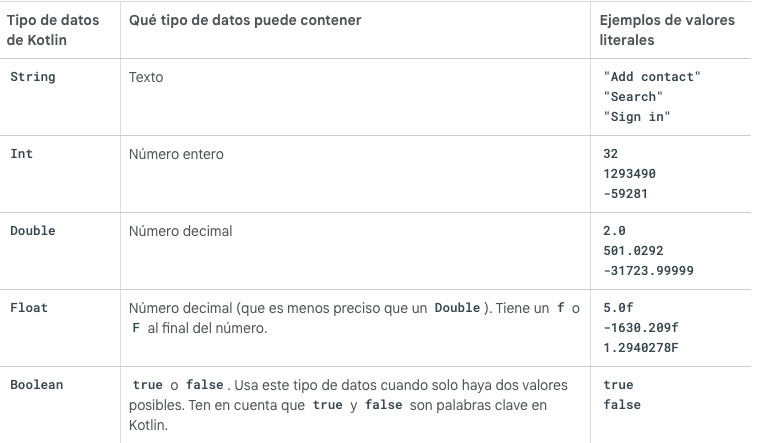
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **fun** main() {  courseGreeting()  }  **fun** courseGreeting(): String {  **val** nameGreeting = " Bienvenido al curso, Jaime!"  **val** semesterGreeting = " Estás en 4to semestre!"  **return** "$nameGreeting\n$semesterGreeting"  } |



Busca en la documentación y explica cómo agregar un parámetro y varios parametros a la función courseGreeting():

|  |
| --- |
| Para agregar un parámetro a la función courseGreeting(), simplemente se debe incluir el nombre y el tipo de dato del parámetro entre los paréntesis de la función. Por ejemplo, si quisiera añadir un parámetro studentName de tipo String, lo haría de esta manera: fun courseGreeting(studentName: String): String.  Y si quisiera agregar varios parámetros, solo tendría que separar cada uno con una coma. Por ejemplo, si deseo añadir un parámetro adicional llamado semester de tipo Int, la firma de la función se vería así: fun courseGreeting(studentName: String, semester: Int): String.  ¡Con estos cambios, sería capaz de personalizar aún más mi función courseGreeting() al incluir información específica del estudiante, como su nombre y el semestre en el que se encuentra! ¡Espero que esta explicación te haya sido de ayuda! ¡Si tienes más preguntas o necesitas más ayuda, no dudes en decírmelo! |

## Tipos de datos

****

Si necesitas actualizar el valor de una variable, declara la variable con la palabra clave de Kotlin var, en lugar de val.

* Palabra clave val: Úsala cuando esperes que el valor de la variable no cambie.
* Palabra clave var: Úsala cuando esperes que el valor de la variable pueda cambiar.

Ingresa este código en el Playground de Kotlin. ¿Puedes entender lo que sucede en cada línea de código?

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **fun** main() {  **val** trip1: Double = 3.20  **val** trip2: Double = 4.10  **val** trip3: Double = 1.72  **val** totalTripLength: Double = 0.0  println("$totalTripLength miles left to destination")  } |

|  |
| --- |
| En la función se han creado cuatro variables de tipo Double con diferentes valores para representar las distancias de varios viajes. Posteriormente, se imprime el valor de la cuarta variable, totalTripLength, que se inicializa en 0.0 |

## Ejercicios

Basandote en la [documentación oficial de kotlin](https://kotlinlang.org/docs/home.html), resuelve los siguientes ejercicios:

1. ¿Puedes escribir una función main() que imprima estos mensajes en cuatro líneas separadas?

Use the val keyword when the value doesnt change.

Use the var keyword when the value can change.

When you define a function, define the parameters that can be passed to it.

When you call a function, you pass arguments for the parameters.

|  |
| --- |
| Version Espanol |

1. Este programa imprime un mensaje que le notifica al usuario que recibió un mensaje de chat de un amigo:

**fun** main() {

println("New chat message from a friend'}

}

1. ¿Puedes determinar la causa raíz de los errores de compilación de este programa y corregirlos?
2. ¿El código usa los símbolos apropiados para indicar la apertura y el cierre de la cadena y el argumento de la función?

|  |
| --- |
| El error en el programa fue una comilla incorrecta en el mensaje que se quería imprimir. La solución fue corregir esa comilla para que la cadena de texto esté bien formada.  En cuanto a los símbolos apropiados, era importante usar comillas dobles (") para indicar el inicio y fin de la cadena de texto, y asegurarse de que el argumento de la función println() esté dentro de paréntesis.  Con la corrección, el código quedaría así de manera más simple y correcta: |

**Pista:** Puedes usar Kotlin Playground a fin de ejecutar el código y ver los errores de compilación.

Después de corregir los errores, el programa debe compilarse sin problemas y, luego, imprimir este resultado:

New chat message from a friend

1. En este ejercicio, escribirás un programa que realice operaciones matemáticas básicas y, luego, imprima el resultado.

### Paso 1

La función main() contiene un error de compilación:

**fun** main() {

**val** firstNumber = 10

**val** secondNumber = 5

println("$firstNumber + $secondNumber = $result")

}

¿Puedes corregir el error de modo que el programa imprima este resultado?

10 + 5 = 15

|  |
| --- |
| Se necesita asignar la suma de firstNumber y secondNumber a una nueva variable result antes de imprimirlo. Aquí está el código corregido: |

### Paso 2

El código funciona, pero la lógica para sumar dos números se encuentra dentro de la variable de resultado, lo que hace que el código sea menos flexible a la hora de volver a usarlo. En su lugar, puedes extraer la operación de suma en una función add() para que el código se pueda volver a usar. Para ello, actualiza el código con el que se muestra a continuación. Observa que el código ahora presenta una nueva variable val, llamada thirdNumber, e imprime el resultado de esta variable nueva con firstNumber.

**fun** main() {

**val** firstNumber = 10

**val** secondNumber = 5

**val** thirdNumber = 8

**val** result = add(firstNumber, secondNumber)

**val** anotherResult = add(firstNumber, thirdNumber)

println("$firstNumber + $secondNumber = $result")

println("$firstNumber + $thirdNumber = $anotherResult")

}

*// Define add() function below this line*

¿Puedes definir la función add() de modo que el programa imprima este resultado?

10 + 5 = 15

10 + 8 = 18

|  |
| --- |
|  |

### Paso 3

Ahora tienes una función reutilizable capaz de sumar dos números.

* ¿Puedes implementar la función subtract() de la misma manera en que implementaste la función add()? Modifica la función main() también para usar la función subtract(), de modo que puedas verificar que funcione como se espera.

|  |
| --- |
|  |

**Pista:** Piensa en la diferencia entre la suma, la resta y otras operaciones matemáticas. Comienza a trabajar en el código de solución a partir de allí.

1. En este programa, se muestra el clima de diferentes ciudades. Incluye el nombre de la ciudad, las temperaturas máxima y mínima del día, y las probabilidades de lluvia.

**fun** main() {

println("City: Ankara")

println("Low temperature: 27, High temperature: 31")

println("Chance of rain: 82%")

println()

println("City: Tokyo")

println("Low temperature: 32, High temperature: 36")

println("Chance of rain: 10%")

println()

println("City: Cape Town")

println("Low temperature: 59, High temperature: 64")

println("Chance of rain: 2%")

println()

println("City: Guatemala City")

println("Low temperature: 50, High temperature: 55")

println("Chance of rain: 7%")

println()

}

Hay muchas similitudes en el código que imprime el clima de cada ciudad. Por ejemplo, hay frases que se repiten varias veces, como "City:" y "Low temperature:". Los códigos similares y repetidos crean el riesgo de que se produzcan errores en tu programa. Puede que hayas escrito mal una de las ciudades o que hayas olvidado uno de los detalles del clima.

1. ¿Puedes crear una función que imprima los detalles del clima de una sola ciudad para reducir la repetición en la función main() y, luego, hacer lo mismo en las ciudades restantes?
2. ¿Puedes actualizar la función main() a fin de llamar a la función que creaste para cada ciudad y pasar los detalles apropiados del clima como argumentos?

|  |
| --- |
|  |

## Instalación de Android Studio

En base a tu sistema operativo, realiza la instalación de android studio.

[Windows](https://developer.android.com/codelabs/basic-android-kotlin-compose-install-android-studio?hl=es-419#2)

[macOS](https://developer.android.com/codelabs/basic-android-kotlin-compose-install-android-studio?hl=es-419#4)

[Linux](https://developer.android.com/codelabs/basic-android-kotlin-compose-install-android-studio?hl=es-419#6)

Documenta el proceso:

|  |
| --- |
| INSTALACION DEL ANDROID ESTUDIO EN UN SISTEMA WINDOWS  PASO 1  BUSCAR EN LA WEB    PASO 2  COMPLETAR EL TUTORIAL    PASO 3  DESCARGAR  PASO 4  ACEPTAR LAS CONDICIONES OBLIGATORIAS    PASO 5  INSTALAR EL PAQUETE DESCARGADO          INSTALAR CONFIGURACION    FIN |

OBSERVACIONES (5 mínimo):

(Las observaciones son las notas aclaratorias, objeciones y problemas que se pudo presentar en el desarrollo del laboratorio)

 La instalación de Android Studio varía según el sistema operativo.

 Se requiere mucho espacio en disco para Android Studio.

 Es necesaria una conexión a internet estable durante la instalación.

 Los equipos deben cumplir los requisitos mínimos para Android Studio.

 Se deben configurar las opciones del entorno de desarrollo tras la instalación.

CONCLUSIONES (5 mínimo):

(Las conclusiones son una opinión personal sobre tu trabajo, explicar cómo resolviste las dudas o problemas presentados en el laboratorio. Además de aportar una opinión critica de lo realizado)

 Un tutorial estructurado facilita la instalación de Android Studio.

 Los ejercicios permiten aprender los conceptos básicos de Kotlin.

 Crear funciones para el código repetitivo mejora la eficiencia.

 La corrección de errores de compilación es esencial para la calidad del código.

 Los ejercicios muestran la versatilidad de Kotlin en diferentes escenarios.