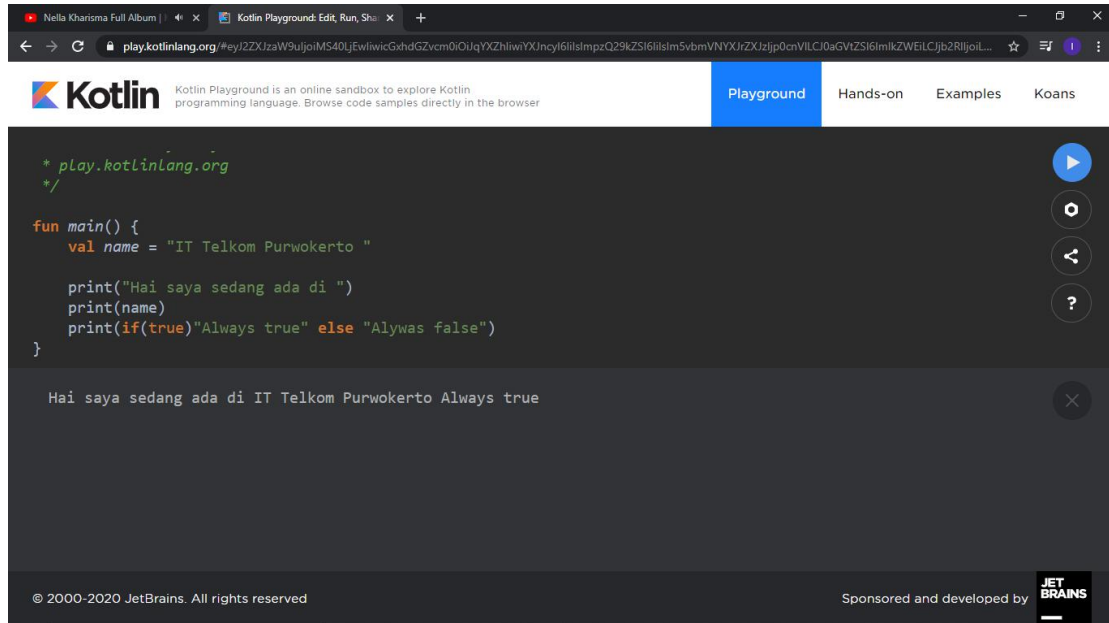


Nama : Intan Ayu Insani  
Kelas : S1IF-06-MM2  
NIM : 18102233

## 1. Hello World



The screenshot shows the Kotlin Playground interface. The code editor contains the following Kotlin code:

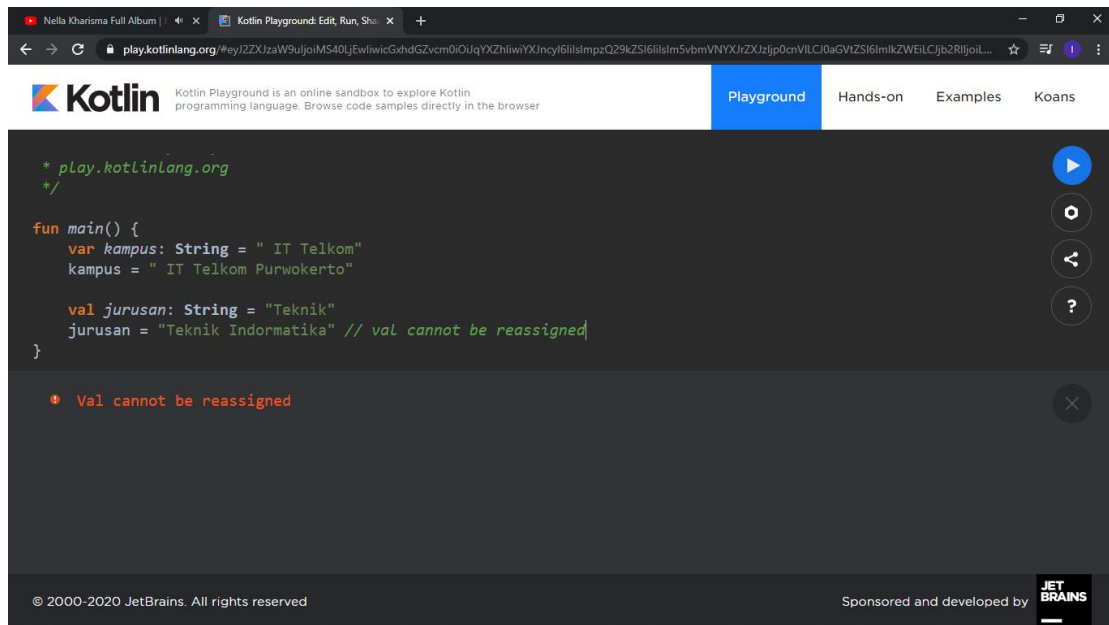
```
* play.kotlinlang.org
*/

fun main() {
    val name = "IT Telkom Purwokerto "

    print("Hai saya sedang ada di ")
    print(name)
    print(if(true)"Always true" else "Alywas false")
}
```

The output of the program is displayed below the code: "Hai saya sedang ada di IT Telkom Purwokerto Always true". The interface includes a navigation bar with "Playground", "Hands-on", "Examples", and "Koans" tabs. The footer shows "© 2000-2020 JetBrains. All rights reserved" and "Sponsored and developed by JET BRAINS".

## 2. Data Type



The screenshot shows the Kotlin Playground interface. The code editor contains the following Kotlin code:

```
* play.kotlinlang.org
*/

fun main() {
    var kampus: String = " IT Telkom"
    kampus = " IT Telkom Purwokerto"

    val jurusan: String = "Teknik"
    jurusan = "Teknik Indormatika" // val cannot be reassigned
}
```

The output area shows a red error message: "Val cannot be reassigned". The interface includes a navigation bar with "Playground", "Hands-on", "Examples", and "Koans" tabs. The footer shows "© 2000-2020 JetBrains. All rights reserved" and "Sponsored and developed by JET BRAINS".

Kotlin Playground: Edit, Run, Share

playkotlinlang.org/#eyJ2ZXJzaW9uIjoI... Kotlin Playground is an online sandbox to explore Kotlin programming language. Browse code samples directly in the browser

Playground Hands-on Examples Koans

```
fun main() {  
    val kataAwal = "Saya"  
    val kataAkhir = "Hebat"  
    val nilaiA: Int = 10  
    val nilaiB = 20  
  
    print(nilaiA + nilaiB)  
    print("\n")  
    print(kataAwal + kataAkhir)  
}
```

30  
SayaHebat

© 2000-2020 JetBrains. All rights reserved Sponsored and developed by JET BRAINS

### 3. Characters

Kotlin Playground: Edit, Run, Share

playkotlinlang.org/#eyJ2ZXJzaW9uIjoI... Kotlin Playground is an online sandbox to explore Kotlin programming language. Browse code samples directly in the browser

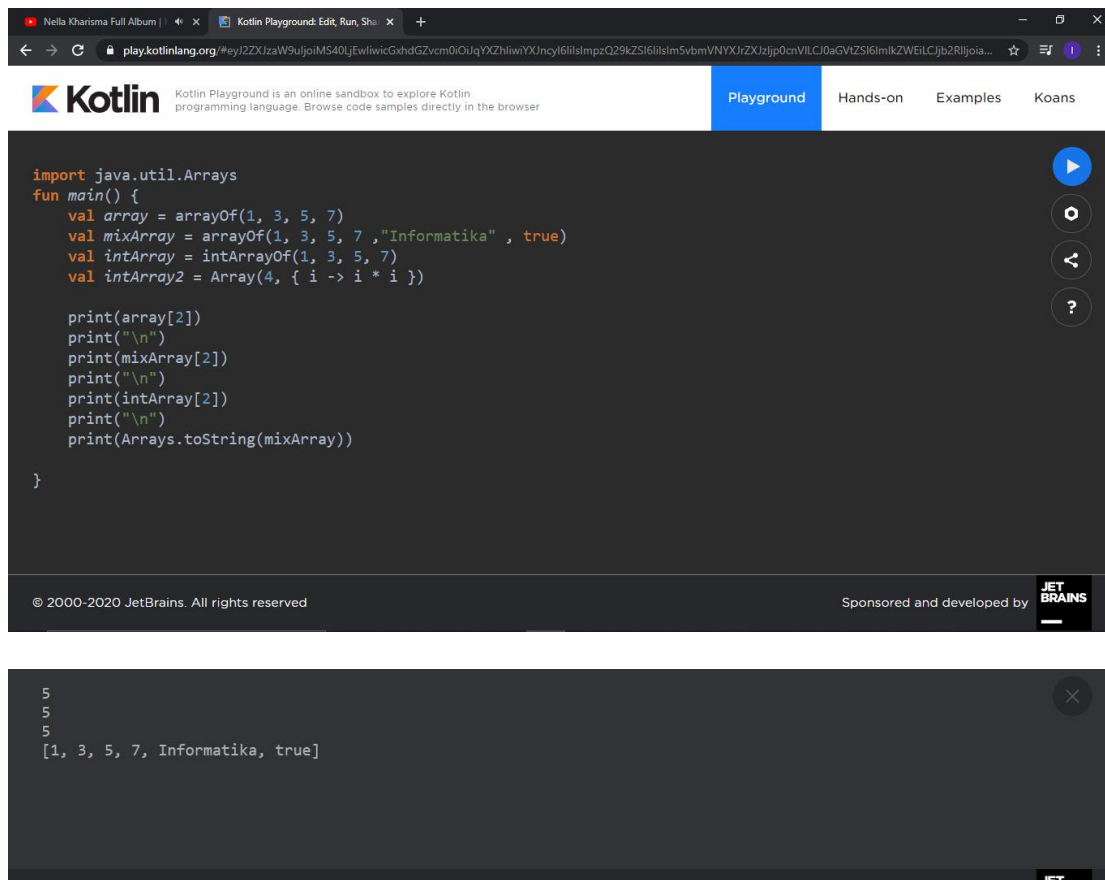
Playground Hands-on Examples Koans

```
fun main() {  
    var vocal = 'A'  
    vocal++  
    println("Vocal " + vocal++)  
    println("Vocal " + vocal++)  
    println("Vocal " + vocal++)  
    println("Vocal " + vocal--)  
    println("Vocal " + vocal--)  
    println("Vocal " + vocal--)  
    println("Vocal " + vocal--)  
}
```

Vocal B  
Vocal C  
Vocal D  
Vocal E  
Vocal D  
Vocal C  
Vocal B

© 2000-2020 JetBrains. All rights reserved Sponsored and developed by JET BRAINS

## 4. Array



The screenshot shows the Kotlin Playground interface. The code editor contains the following Kotlin code:

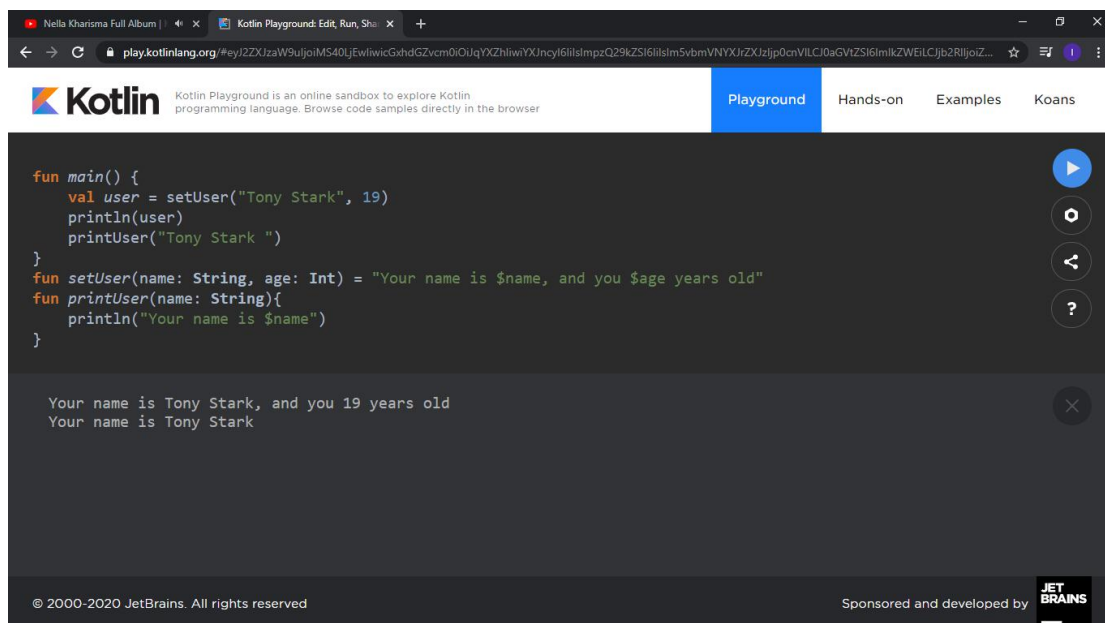
```
import java.util.Arrays
fun main() {
    val array = arrayOf(1, 3, 5, 7)
    val mixArray = arrayOf(1, 3, 5, 7, "Informatika", true)
    val intArray = intArrayOf(1, 3, 5, 7)
    val intArray2 = Array(4, { i -> i * i })

    print(array[2])
    print("\n")
    print(mixArray[2])
    print("\n")
    print(intArray[2])
    print("\n")
    print(Arrays.toString(mixArray))
}
```

The output console shows the results of the code execution:

```
5
5
5
[1, 3, 5, 7, Informatika, true]
```

## 5. Functions



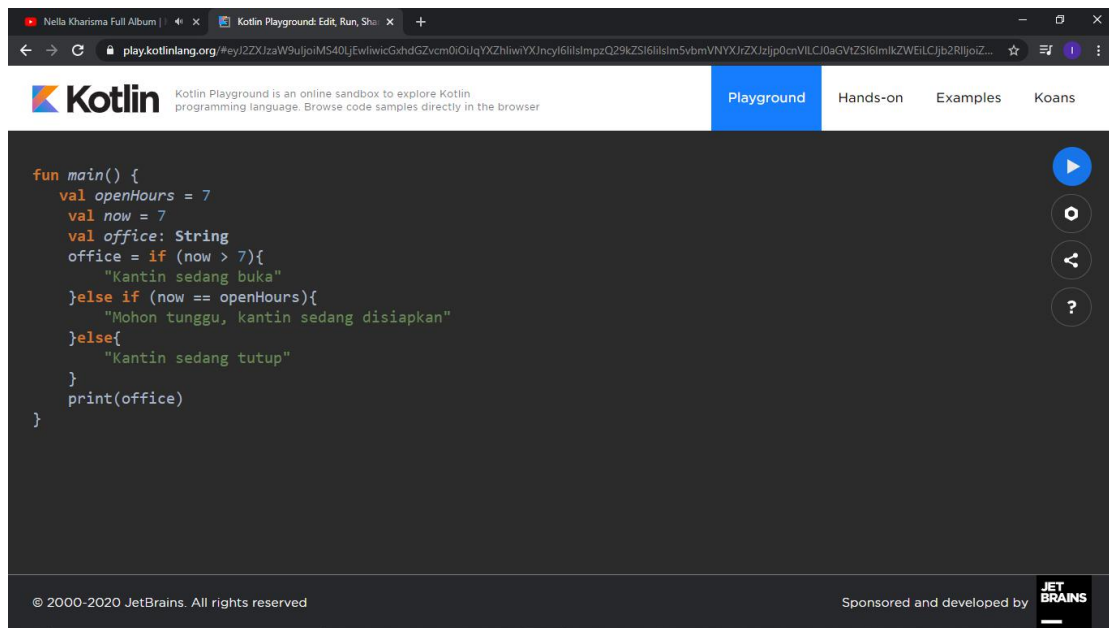
The screenshot shows the Kotlin Playground interface. The code editor contains the following Kotlin code:

```
fun main() {
    val user = setUser("Tony Stark", 19)
    println(user)
    printUser("Tony Stark ")
}
fun setUser(name: String, age: Int) = "Your name is $name, and you $age years old"
fun printUser(name: String){
    println("Your name is $name")
}
```

The output console shows the results of the code execution:

```
Your name is Tony Stark, and you 19 years old
Your name is Tony Stark
```

## 6. If Expressions

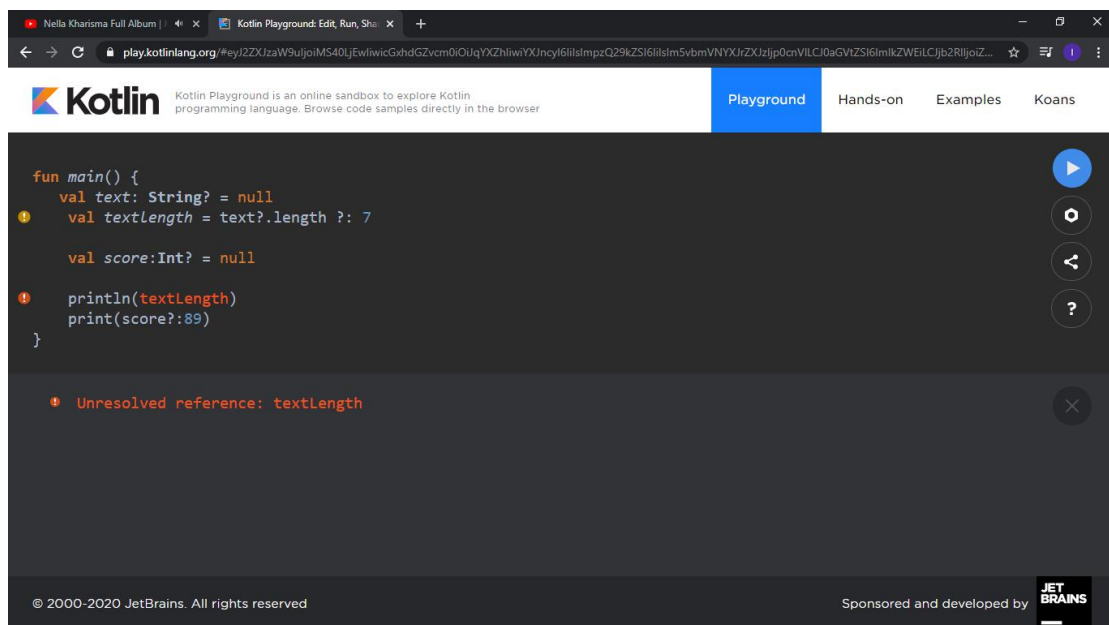


The screenshot shows the Kotlin Playground interface. The code editor contains the following Kotlin code:

```
fun main() {  
    val openHours = 7  
    val now = 7  
    val office: String  
    office = if (now > 7){  
        "Kantin sedang buka"  
    }else if (now == openHours){  
        "Mohon tunggu, kantin sedang disiapkan"  
    }else{  
        "Kantin sedang tutup"  
    }  
    print(office)  
}
```

The output area displays the result: "Mohon tunggu, kantin sedang disiapkan". The interface includes a top navigation bar with "Kotlin" logo and "Playground", "Hands-on", "Examples", and "Koans" tabs. The bottom footer shows "© 2000-2020 JetBrains. All rights reserved" and "Sponsored and developed by JET BRAINS".

## 7. Elvis Operation



The screenshot shows the Kotlin Playground interface. The code editor contains the following Kotlin code:

```
fun main() {  
    val text: String? = null  
    val textLength = text?.length ?: 7  
  
    val score: Int? = null  
  
    println(textLength)  
    print(score?:89)  
}
```

The output area displays an error message: "Unresolved reference: textLength". The interface includes a top navigation bar with "Kotlin" logo and "Playground", "Hands-on", "Examples", and "Koans" tabs. The bottom footer shows "© 2000-2020 JetBrains. All rights reserved" and "Sponsored and developed by JET BRAINS".

## 8. String Template

The screenshot shows the Kotlin Playground interface. The code editor contains the following Kotlin code:

```
fun main() {  
    val name = "Tony Stark"  
    val old = 3  
    val hour = 7  
    println("My name is $name, im $old years old")  
    println("Office ${if (hour > 7) "already close" else "is open"}")  
}
```

The output area displays the results of the program execution:

```
My name is Tony Stark, im 3 years old  
Office is open
```

The bottom of the interface shows the copyright notice "© 2000-2020 JetBrains. All rights reserved" and the JetBrains logo.

## 9. When Expressions

The screenshot shows the Kotlin Playground interface. The code editor contains the following Kotlin code:

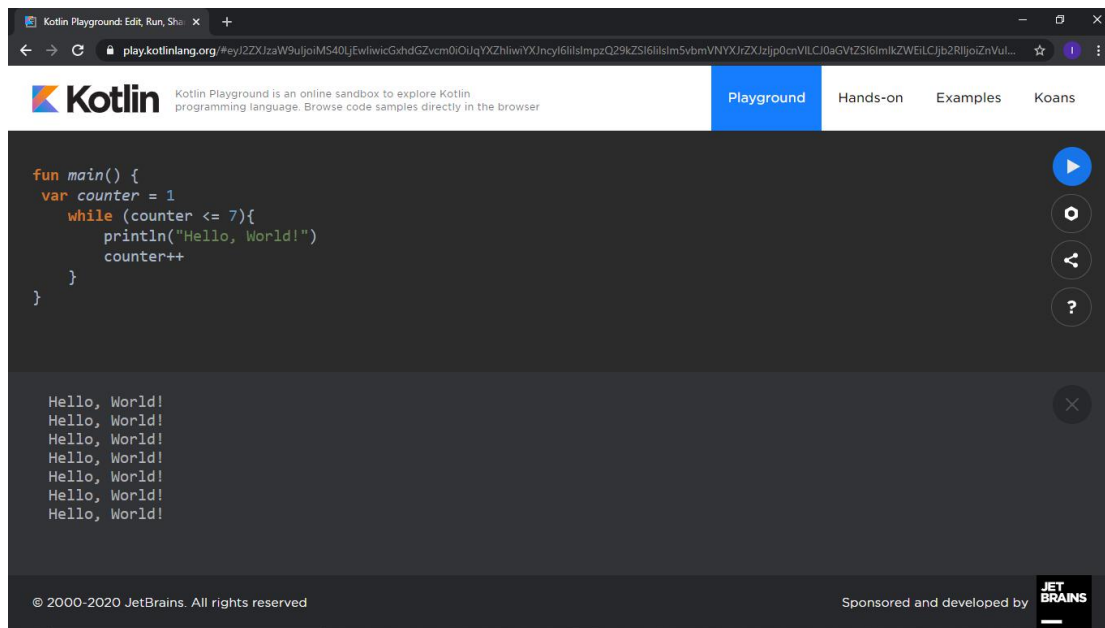
```
fun main() {  
    val value = 20  
    when(value){  
        6 -> println("value is 6")  
        7 -> println("value is 7")  
        8 -> println("value is 8")  
        else -> println("value cannot be reached")  
    }  
    val value2 = 7  
    val ranges = 10..50  
  
    when(value2){  
        in ranges -> println("value is in the range")  
        !in ranges -> println("values is outside the range")  
        else -> println("value underfined")  
    }  
}
```

The output area displays the results of the program execution:

```
value cannot be reached  
values is outside the range
```

The bottom of the interface shows the copyright notice "© 2000-2020 JetBrains. All rights reserved" and the JetBrains logo.

## 10. While Expressions



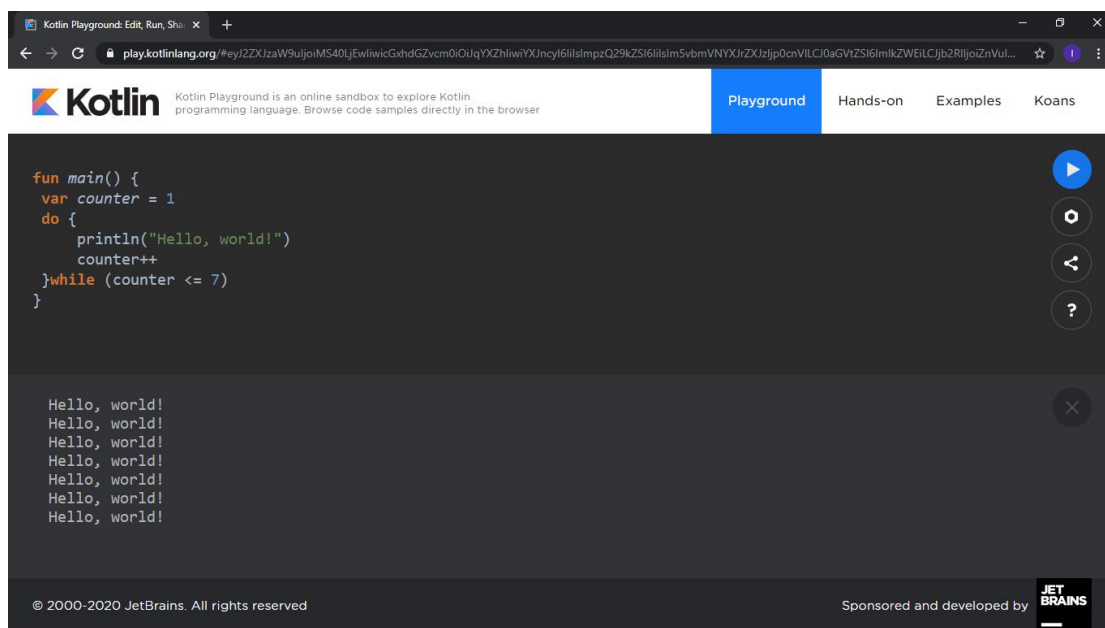
The screenshot shows the Kotlin Playground interface. The code editor contains the following Kotlin code:

```
fun main() {  
    var counter = 1  
    while (counter <= 7){  
        println("Hello, World!")  
        counter++  
    }  
}
```

The output area displays the result of the program execution:

```
Hello, World!  
Hello, World!  
Hello, World!  
Hello, World!  
Hello, World!  
Hello, World!  
Hello, World!
```

The footer of the playground shows the copyright notice: © 2000-2020 JetBrains. All rights reserved. It also mentions "Sponsored and developed by JET BRAINS".



The screenshot shows the Kotlin Playground interface. The code editor contains the following Kotlin code:

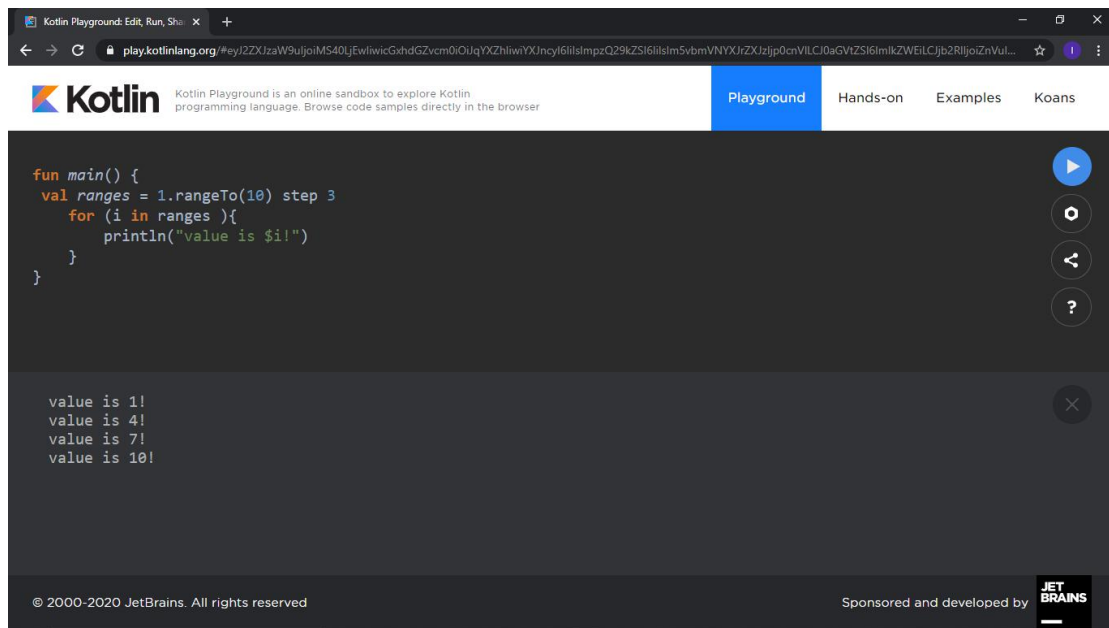
```
fun main() {  
    var counter = 1  
    do {  
        println("Hello, world!")  
        counter++  
    }while (counter <= 7)  
}
```

The output area displays the result of the program execution:

```
Hello, world!  
Hello, world!  
Hello, world!  
Hello, world!  
Hello, world!  
Hello, world!  
Hello, world!
```

The footer of the playground shows the copyright notice: © 2000-2020 JetBrains. All rights reserved. It also mentions "Sponsored and developed by JET BRAINS".

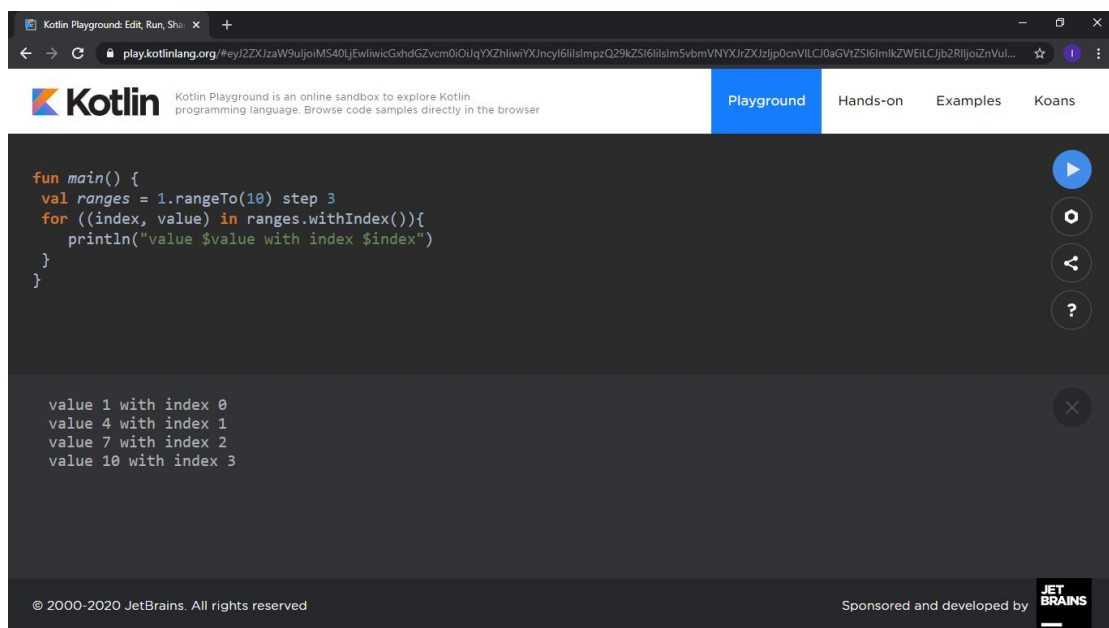
## 11. For Loop



The screenshot shows the Kotlin Playground interface. The code editor contains a simple for loop that iterates over the range 1 to 10 with a step of 3. The output shows the values 1, 4, 7, and 10.

```
fun main() {  
    val ranges = 1.rangeTo(10) step 3  
    for (i in ranges) {  
        println("value is $i!")  
    }  
}
```

value is 1!  
value is 4!  
value is 7!  
value is 10!



The screenshot shows the Kotlin Playground interface. The code editor contains a for loop that iterates over the range 1 to 10 with a step of 3, using the `withIndex()` method to access both the index and the value. The output shows the value and index for each iteration.

```
fun main() {  
    val ranges = 1.rangeTo(10) step 3  
    for ((index, value) in ranges.withIndex()) {  
        println("value $value with index $index")  
    }  
}
```

value 1 with index 0  
value 4 with index 1  
value 7 with index 2  
value 10 with index 3