# "Privileges: A Terminal-Based Game"

## Abschlussarbeit im Rahmen des Moduls: Grundlagen der Programmierung (M2)

Betreut durch den Dozenten: Gordon Lucas
und die Tutor*innen: Ahmad Alkhalaf, Jana Jansen,
Justus Finke, Lukas Altmann, Noah Diemel, Pavel Keßler und Talha Taskaya

Syntax Institut

Boris Niehaus Syntax Institute 20.6.2023

In Module 2, 'Fundamentals of Programming', our focus was on understanding the core concepts and techniques that empower us to write efficient and functional programs.

The course was a mix of theoretical lectures in the morning with several daily checkpoints, coupled with practical exercises assigned each day, all designed to strengthen our grasp of programming fundamentals.

WEEK 1: Introduction, Variables, Data Types:

Day 01: IDE: Introduction to the Integrated Development Environment for Kotlin programming.

Day 02: Variables: Understanding the declaration, initialization, and use of variables in Kotlin.

Day 03: Mathematical Operations and Type Conversion: Learning about the conversion of data types.

04: Practice Friday: Consolidation of the week's learnings with practical exercises.

WEEK 2: Lists & Maps:

Day 06: Introduction to Lists

Day 07: List Functions: Diving deeper into the methods and operations that can be performed with lists.

Day 08: List Functions 2

Day 09: Maps: Introduction to map data structures and key-value pairs.

Day 10: Practice Friday

WEEK 3: Functions:

Day 11: Introduction to Functions: Understanding how to declare and call functions in Kotlin.

Day 12: Return Values: Learning how functions return values

Day 13: Parameters: Grasping the concept of function parameters and how to pass data to functions using parameters.

Day 14: Packages and Library Functions: Understanding the concept of packages and library functions

Day 15: Card Game: Implementing a card game using the concepts learned particularly functions.

WEEK 4: Branching:

Day 16: Branching

Day 17: Operators

Day 18: Try Catch

Day 19: Packages and Library Functions: Further exploration of packages and library functions.

WEEK 5: Loops:

Day 22: While and Do-While Loops: Understanding and practicing the concepts of while and do-while loops.

Day 23: For Loops and Ranges: Exploring the usage of for loops and ranges.

Day 24: Break and Continue: Learning how to use break and continue statements within loops.

Day 25: Practice Friday: Applying the knowledge gained during the week through practical exercises.

Week 6: Object-Oriented Programming:

Day 26: Classes: Introduction to classes and their role in object-oriented programming.

Day 27: Constructors: Understanding constructors and their usage in creating objects.

Day 28: Inheritance: Exploring the concept of inheritance and its benefits in code reuse.

Day 29: Encapsulation: Learning about encapsulation and how it helps in data hiding and abstraction.

Day 30: Practice Friday: Applying the concepts learned throughout the week through practical exercises.

## About the Game
## My M2-Capstone Project

"Privileges" is a terminal-based game that I chose to develop as a culmination of the learnings from Module 2: 'Fundamentals of Programming'. This game is designed to bring societal inequalities and the uneven distribution of opportunities into stark focus.

The choice to create this game was an intentional one. It allowed for the practical application of key programming concepts we tackled throughout the course, like loops, classes, and functions, in an interactive and engaging context.

The game presents players with a series of questions, either based on their real-life experiences or from the perspective of a pre-defined character. This gameplay not only showcases the power of conditional statements and user input but also emphasizes empathy, introspection, and solidarity.

Throughout the game, a visual representation of the player's journey is generated, providing an evaluation and encouragement for reflection. This integration of visual elements demonstrates the practical usage of loops and functions to dynamically modify the game.

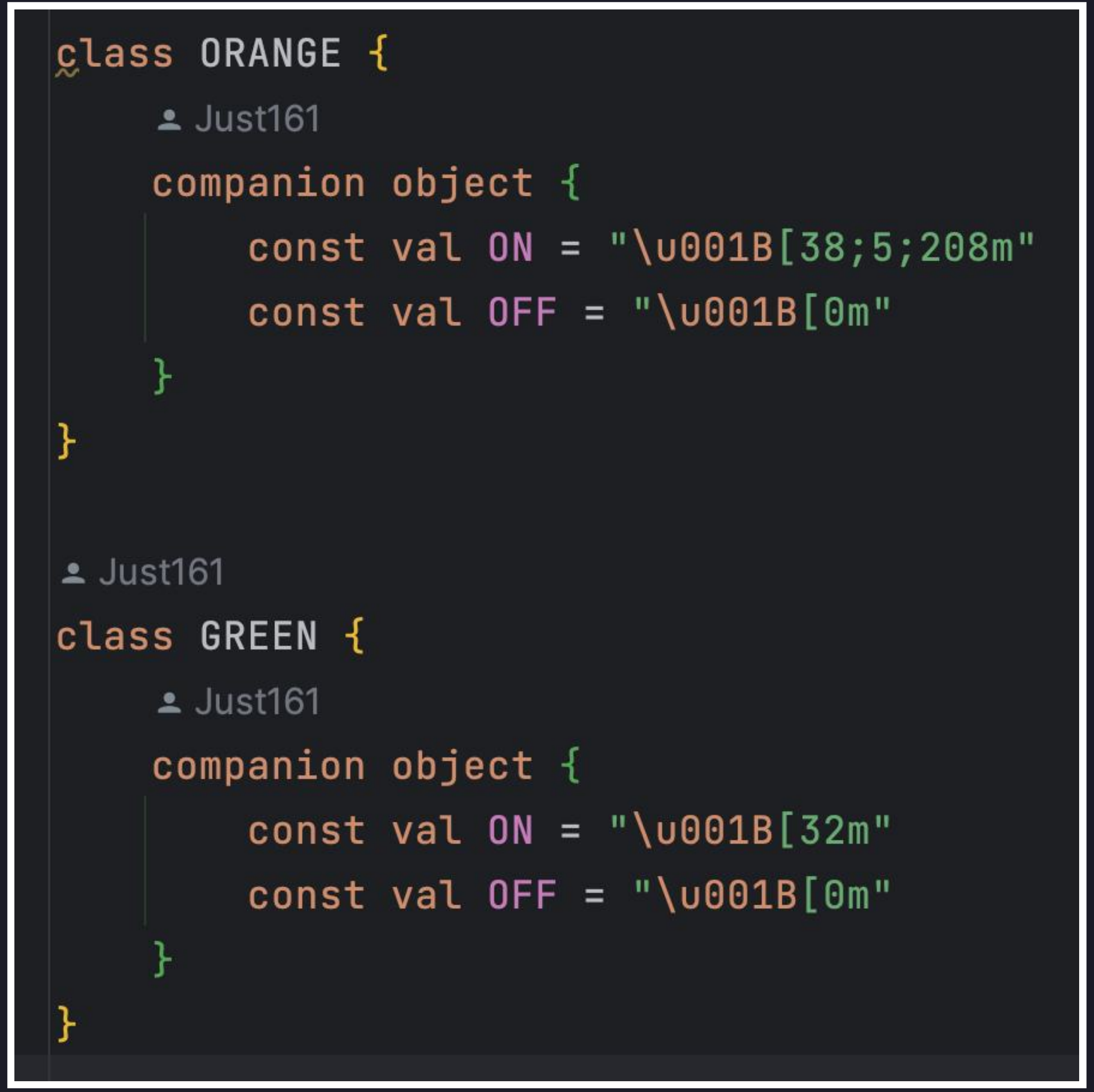
Overall, the development of "Privileges" allowed me to fuse programming fundamentals with a meaningful, relevant, and engaging theme.

# STYLEGUIDE

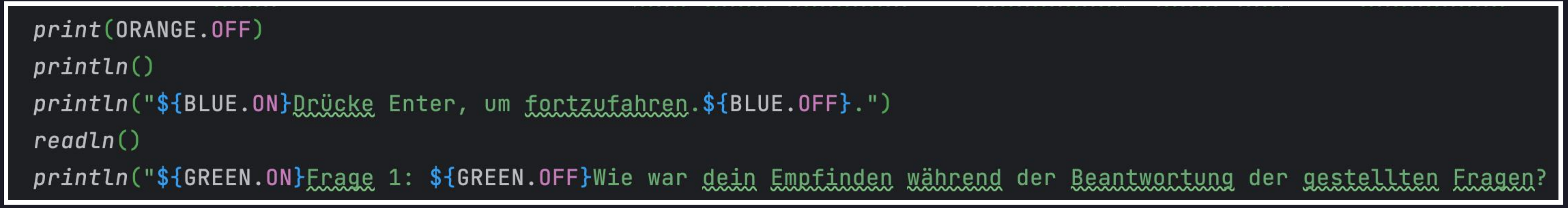
"Privileges: A Terminal-Based Game"

To display colors in the terminal for this game, I implemented ANSI escape codes in Kotlin, creating companion objects for each color such as orange, green, red, and more, which allowed me to modify the text output color dynamically during gameplay.

| | |
|---|---|
| ● | class ORANGE "\u001B[38;5;208m" |
| ● | class BLUE "\u001B[34m" |
| ● | class RED "\u001B[31m" |
| ● | class YELLOW "\u001B[33m" |
| ● | class GREEN "\u001B[32m" |



```
class ORANGE {
    Just161
    companion object {
        const val ON = "\u001B[38;5;208m"
        const val OFF = "\u001B[0m"
    }
}

Just161
class GREEN {
    Just161
    companion object {
        const val ON = "\u001B[32m"
        const val OFF = "\u001B[0m"
    }
}
```



```
print(ORANGE.OFF)
println()
println("${BLUE.ON}Drücke Enter, um fortzufahren.${BLUE.OFF}.")
readln()
println("${GREEN.ON}Frage 1: ${GREEN.OFF}Wie war dein Empfinden während der Beantwortung der gestellten Fragen?
```

## Game Installation
## Game Flow

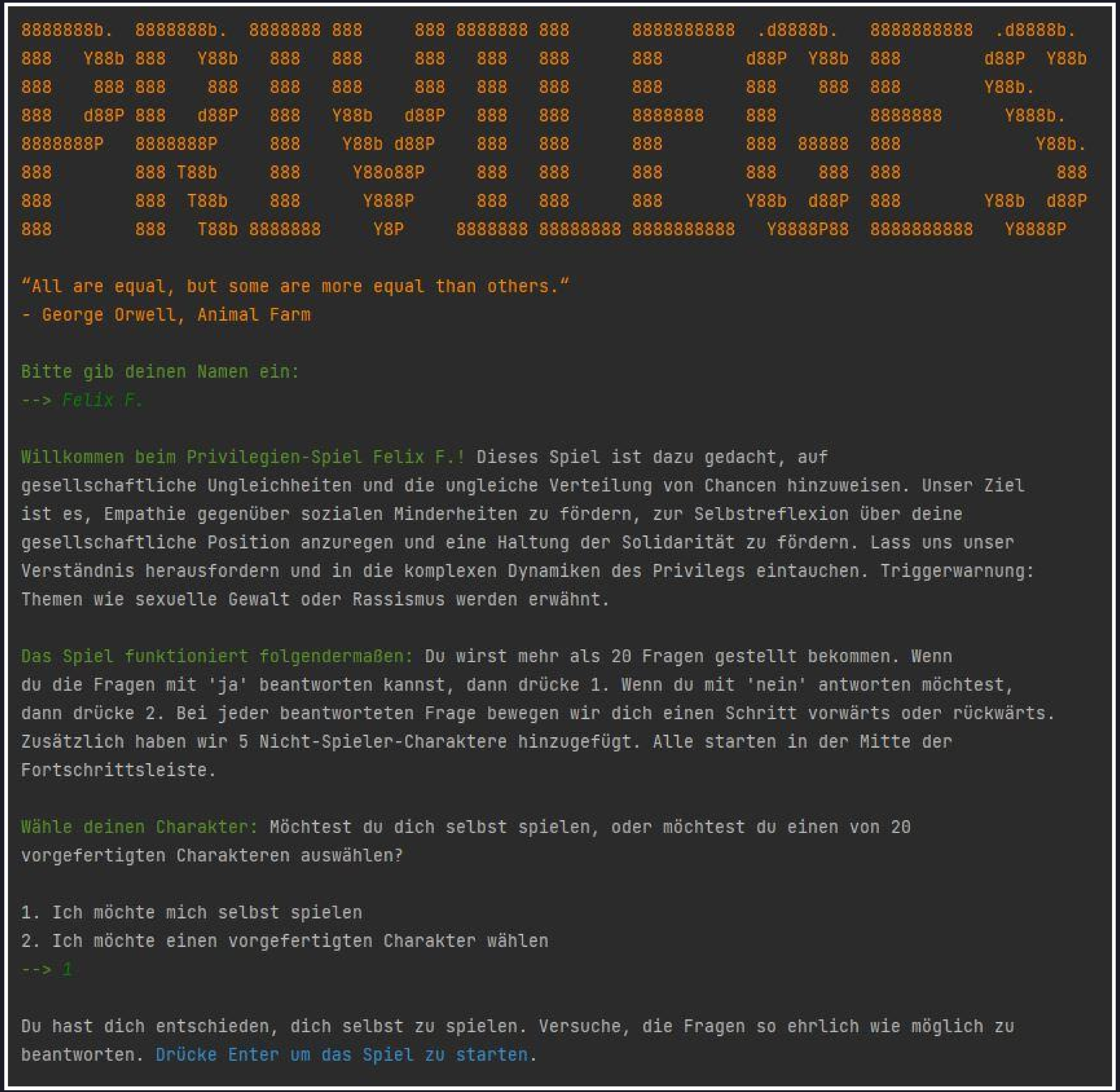
To start, you need to locate and run the main.kt file in the src/main/ directory of the project. This action triggers the game's initial setup and prepares it for play. The beauty of a terminal-based game is its minimal requirements - no elaborate system prerequisites or intricate installation procedures are needed.

Privileges' follows a structured game flow to guide players through different life scenarios. After choosing the language for gameplay (currently only available in German), players can decide whether to play as themselves or adopt the role of a pre-defined character.

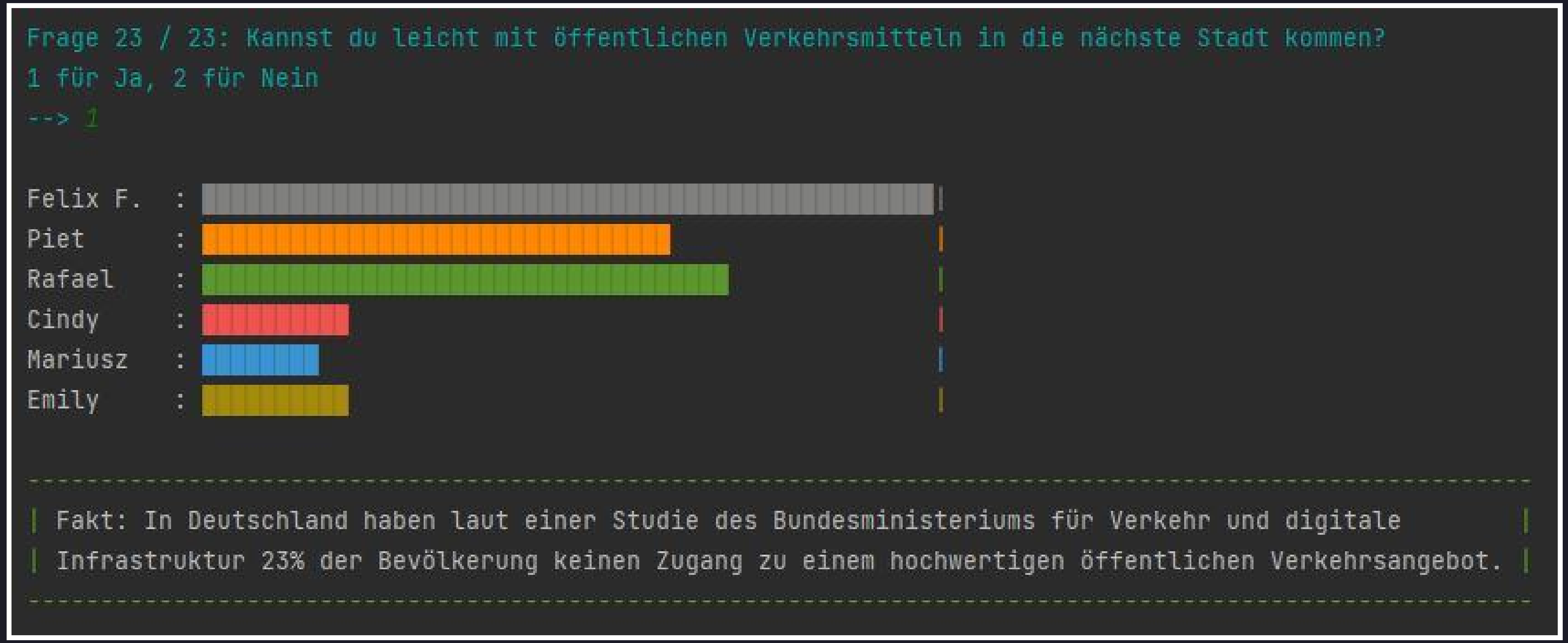
During gameplay, players face a series of questions related to various life situations. Each answer, 'yes' or 'no', influences the game's progress. Answering 'yes' moves the player one step forward, while 'no' pushes them a step back. This movement reflects the advantages or drawbacks in one's life based on their privileges.

By the end of the game, players receive an evaluation based on their responses. This final stage encourages introspection and reflection on the impact of societal privileges or lack thereof. Each gameplay experience is unique, representing the diverse realities that exist in our global society.

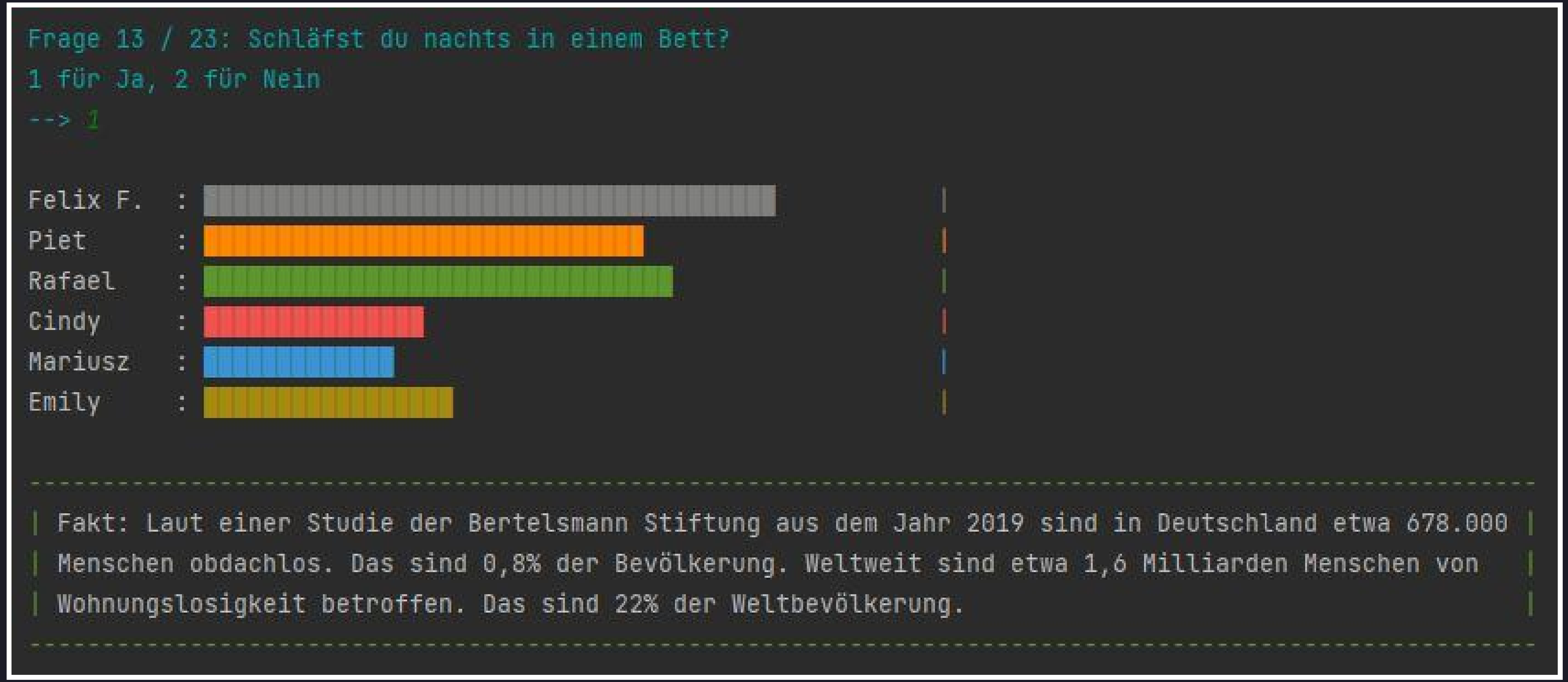
# Screenshots



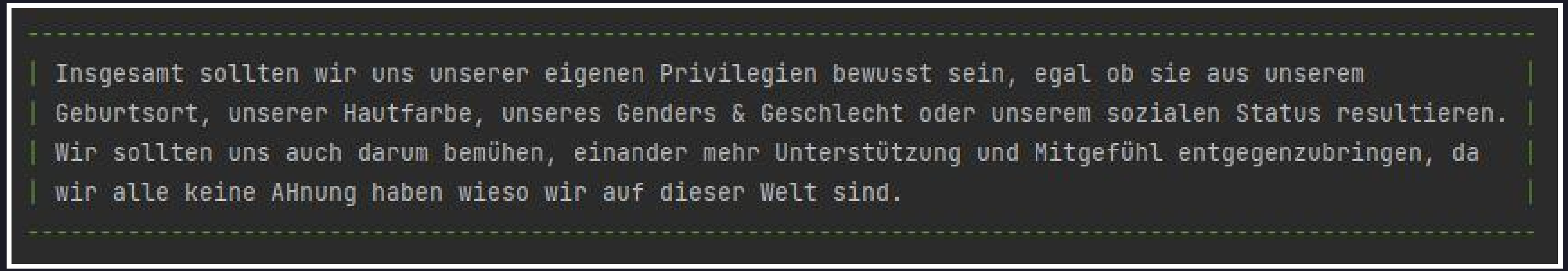
```
8888888b.  8888888b.  8888888 888     888 8888888 888      8888888888  .d8888b.   8888888888  .d8888b.
888   Y88b 888   Y88b   888   888     888   888   888      888        d88P  Y88b  888        d88P  Y88b
888    888 888    888   888   888     888   888   888      888        888    888  888        Y88b.
888   d88P 888   d88P   888   Y88b   d88P   888   888      8888888    888         8888888     Y888b.
8888888P   8888888P     888    Y88b d88P    888   888      888        888  88888  888            Y88b.
888        888 T88b     888     Y88o88P     888   888      888        888    888  888              888
888        888  T88b    888      Y888P      888   888      888        Y88b  d88P  888        Y88b  d88P
888        888   T88b 8888888     Y8P     8888888 88888888 8888888888  Y8888P88   8888888888  Y8888P

“All are equal, but some are more equal than others.“
- George Orwell, Animal Farm

Bitte gib deinen Namen ein:
--> Felix F.

Willkommen beim Privilegien-Spiel Felix F.! Dieses Spiel ist dazu gedacht, auf
gesellschaftliche Ungleichheiten und die ungleiche Verteilung von Chancen hinzuweisen. Unser Ziel
ist es, Empathie gegenüber sozialen Minderheiten zu fördern, zur Selbstreflexion über deine
gesellschaftliche Position anzuregen und eine Haltung der Solidarität zu fördern. Lass uns unser
Verständnis herausfordern und in die komplexen Dynamiken des Privilegs eintauchen. Triggerwarnung:
Themen wie sexuelle Gewalt oder Rassismus werden erwähnt.

Das Spiel funktioniert folgendermaßen: Du wirst mehr als 20 Fragen gestellt bekommen. Wenn
du die Fragen mit 'ja' beantworten kannst, dann drücke 1. Wenn du mit 'nein' antworten möchtest,
dann drücke 2. Bei jeder beantworteten Frage bewegen wir dich einen Schritt vorwärts oder rückwärts.
Zusätzlich haben wir 5 Nicht-Spieler-Charaktere hinzugefügt. Alle starten in der Mitte der
Fortschrittsleiste.

Wähle deinen Charakter: Möchtest du dich selbst spielen, oder möchtest du einen von 20
vorgefertigten Charakteren auswählen?

1. Ich möchte mich selbst spielen
2. Ich möchte einen vorgefertigten Charakter wählen
--> 1

Du hast dich entschieden, dich selbst zu spielen. Versuche, die Fragen so ehrlich wie möglich zu
beantworten. Drücke Enter um das Spiel zu starten.
```



```
Frage 23 / 23: Kannst du leicht mit öffentlichen Verkehrsmitteln in die nächste Stadt kommen?
1 für Ja, 2 für Nein
--> 1

Felix F.  :                                                     |
Piet      :                                                     |
Rafael    :                                                     |
Cindy     :                                                     |
Mariusz   :                                                     |
Emily     :                                                     |

----------------------------------------------------------------------------------------------------------
| Fakt: In Deutschland haben laut einer Studie des Bundesministeriums für Verkehr und digitale            |
| Infrastruktur 23% der Bevölkerung keinen Zugang zu einem hochwertigen öffentlichen Verkehrsangebot.    |
----------------------------------------------------------------------------------------------------------
```



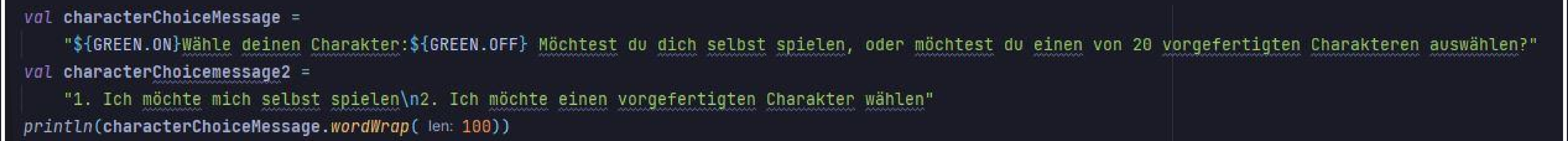
```
Frage 13 / 23: Schläfst du nachts in einem Bett?
1 für Ja, 2 für Nein
--> 1

Felix F.  :                                                     |
Piet      :                                                     |
Rafael    :                                                     |
Cindy     :                                                     |
Mariusz   :                                                     |
Emily     :                                                     |

----------------------------------------------------------------------------------------------------------
| Fakt: Laut einer Studie der Bertelsmann Stiftung aus dem Jahr 2019 sind in Deutschland etwa 678.000    |
| Menschen obdachlos. Das sind 0,8% der Bevölkerung. Weltweit sind etwa 1,6 Milliarden Menschen von      |
| Wohnungslosigkeit betroffen. Das sind 22% der Weltbevölkerung.                                         |
----------------------------------------------------------------------------------------------------------
```



```
----------------------------------------------------------------------------------------------------------
| Insgesamt sollten wir uns unserer eigenen Privilegien bewusst sein, egal ob sie aus unserem             |
| Geburtsort, unserer Hautfarbe, unseres Genders & Geschlecht oder unserem sozialen Status resultieren.  |
| Wir sollten uns auch darum bemühen, einander mehr Unterstützung und Mitgefühl entgegenzubringen, da    |
| wir alle keine AHnung haben wieso wir auf dieser Welt sind.                                            |
----------------------------------------------------------------------------------------------------------
```

# Code Snippets Examples

Week 1:
Variables
basic Stuff



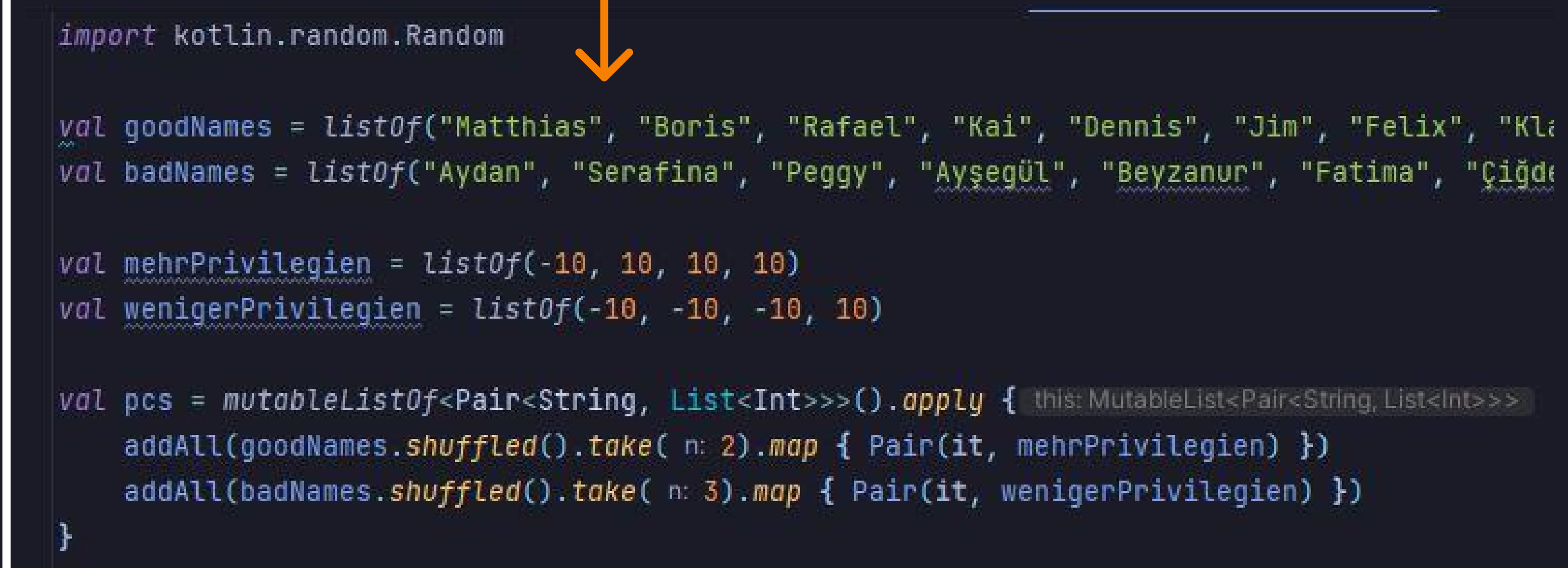
```kotlin
val characterChoiceMessage =
    "${GREEN.ON}Wähle deinen Charakter:${GREEN.OFF} Möchtest du dich selbst spielen, oder möchtest du einen von 20 vorgefertigten Charakteren auswählen?"
val characterChoicemessage2 =
    "1. Ich möchte mich selbst spielen\n2. Ich möchte einen vorgefertigten Charakter wählen"
println(characterChoiceMessage.wordWrap( len: 100))
```

Week 3:
Functions &
Parameters

1. "This code snippet includes the use of ANSI escape sequences, which allow for the manipulation of text colors within the console output."
2. "The .wordWrap function is used here to automatically break lines of text after 100 characters, ensuring that the output fits comfortably within the console width."

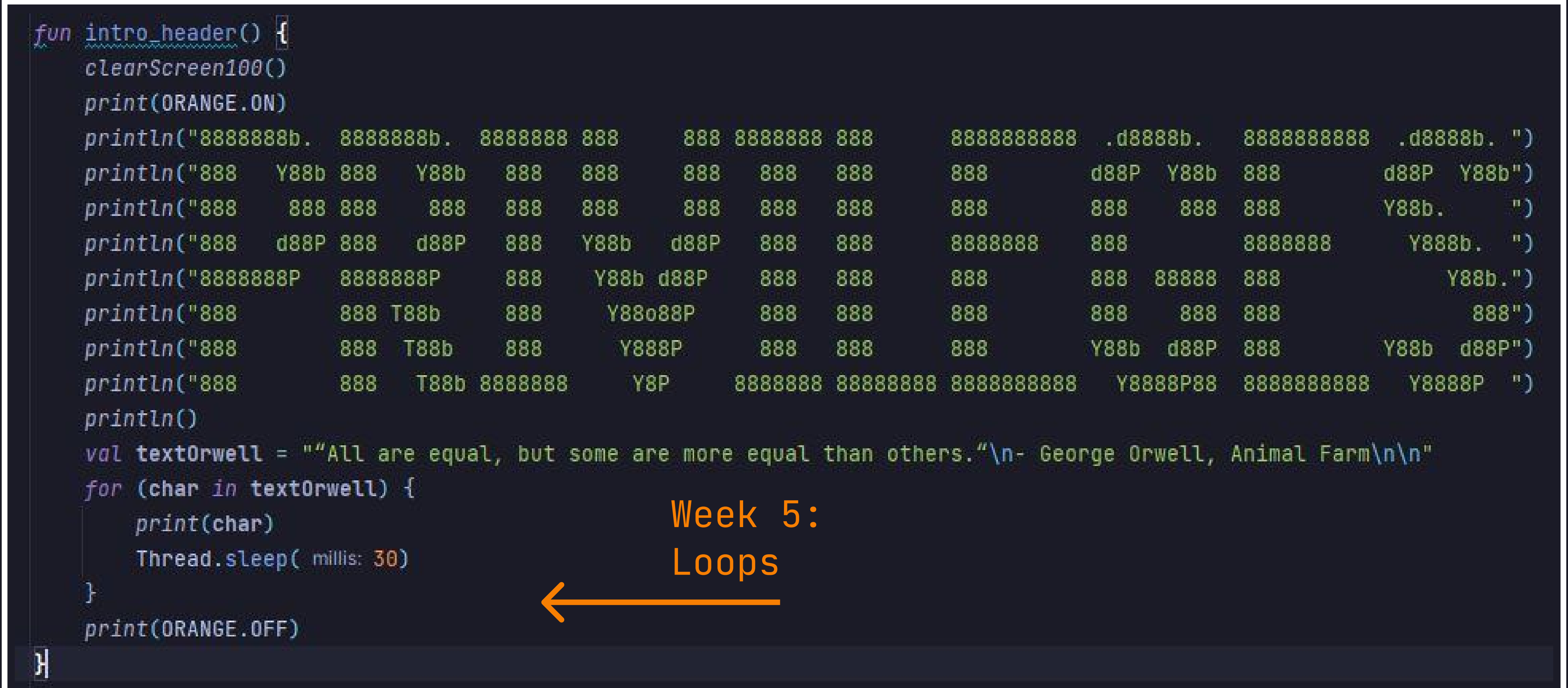
Week 3:
Packages

Week 2: Lists
and List-Functions



```kotlin
import kotlin.random.Random

val goodNames = listOf("Matthias", "Boris", "Rafael", "Kai", "Dennis", "Jim", "Felix", "Kla
val badNames = listOf("Aydan", "Serafina", "Peggy", "Ayşegül", "Beyzanur", "Fatima", "Çiğde

val mehrPrivilegien = listOf(-10, 10, 10, 10)
val wenigerPrivilegien = listOf(-10, -10, -10, 10)

val pcs = mutableListOf<Pair<String, List<Int>>>().apply { this: MutableList<Pair<String, List<Int>>>
    addAll(goodNames.shuffled().take( n: 2).map { Pair(it, mehrPrivilegien) })
    addAll(badNames.shuffled().take( n: 3).map { Pair(it, wenigerPrivilegien) })
}
```

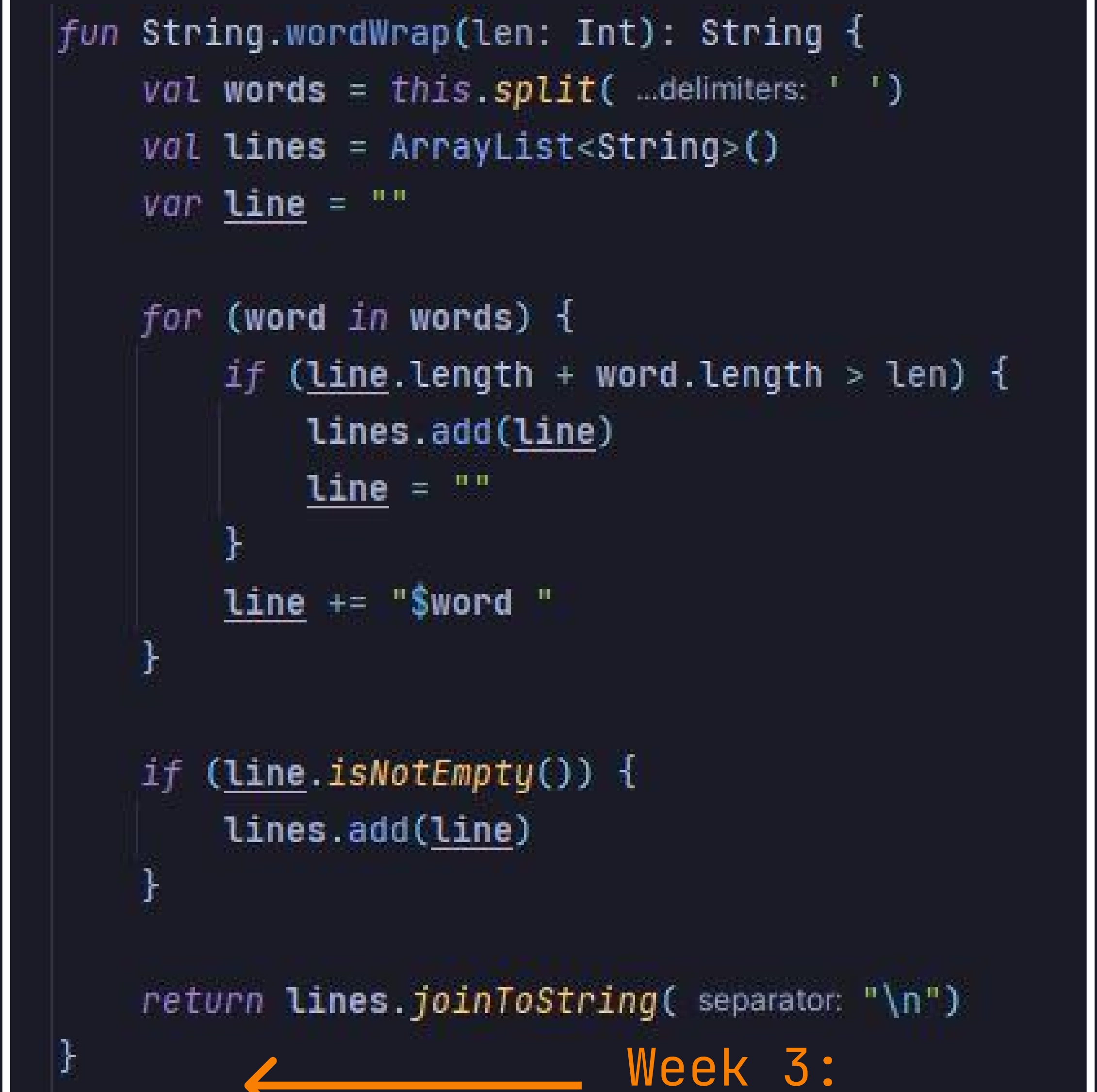
I declared two lists of names, goodNames and badNames. Next, two lists mehrPrivilegien and wenigerPrivilegien are defined, which represent the level of privileges associated with a certain name.



```kotlin
fun intro_header() {
    clearScreen100()
    print(ORANGE.ON)
    println("8888888b.  8888888b.  8888888 888     888 8888888 888      8888888888  .d8888b.   8888888888  .d8888b. ")
    println("888   Y88b 888   Y88b   888   888     888   888   888      888        d88P  Y88b  888        d88P  Y88b")
    println("888    888 888    888   888   888     888   888   888      888        888    888  888        Y88b.     ")
    println("888   d88P 888   d88P   888   Y88b   d88P   888   888      8888888    888         8888888     Y888b.  ")
    println("8888888P   8888888P     888    Y88b d88P    888   888      888        888  88888  888            Y88b.")
    println("888        888 T88b     888     Y88o88P     888   888      888        888    888  888              888")
    println("888        888  T88b    888      Y888P      888   888      888        Y88b  d88P  888        Y88b  d88P")
    println("888        888   T88b 8888888     Y8P     8888888 88888888 8888888888  Y8888P88   8888888888  Y8888P  ")
    println()
    val textOrwell = "“All are equal, but some are more equal than others.“\n- George Orwell, Animal Farm\n\n"
    for (char in textOrwell) {
        print(char)
        Thread.sleep( millis: 30)
    }
    print(ORANGE.OFF)
}
```

Week 5:
Loops

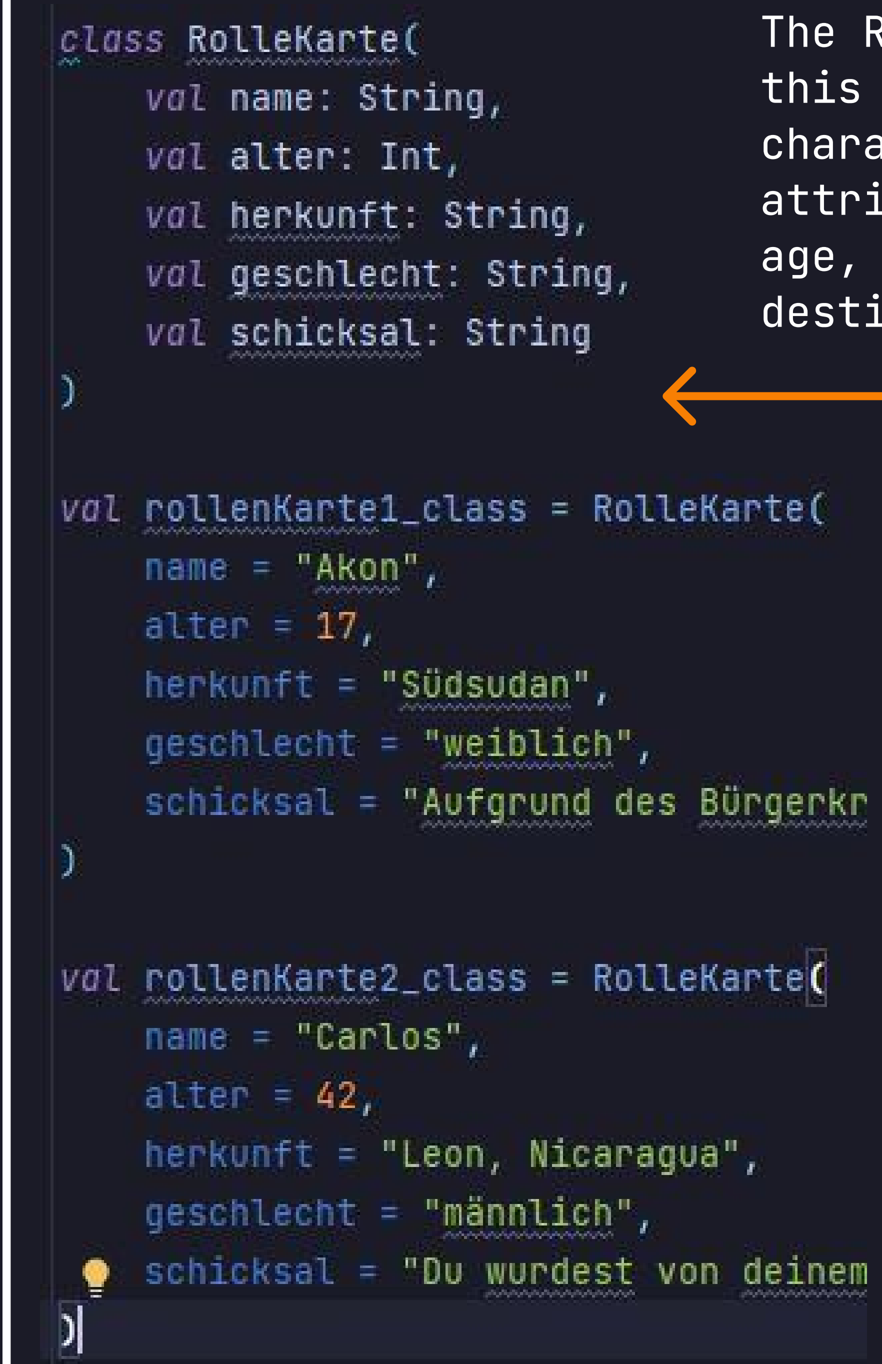
The intro_header function serves two primary purposes. Firstly, it clears the screen and displays an ASCII art header text in orange color by using ANSI escape codes. Secondly, it presents a quotation from George Orwell's 'Animal Farm', printing out each character with a small delay to create a typing effect.



```kotlin
fun String.wordWrap(len: Int): String {
    val words = this.split( ...delimiters: ' ')
    val lines = ArrayList<String>()
    var line = ""

    for (word in words) {
        if (line.length + word.length > len) {
            lines.add(line)
            line = ""
        }
        line += "$word "
    }

    if (line.isNotEmpty()) {
        lines.add(line)
    }

    return lines.joinToString( separator: "\n")
}
```

Week 3:
Return Values

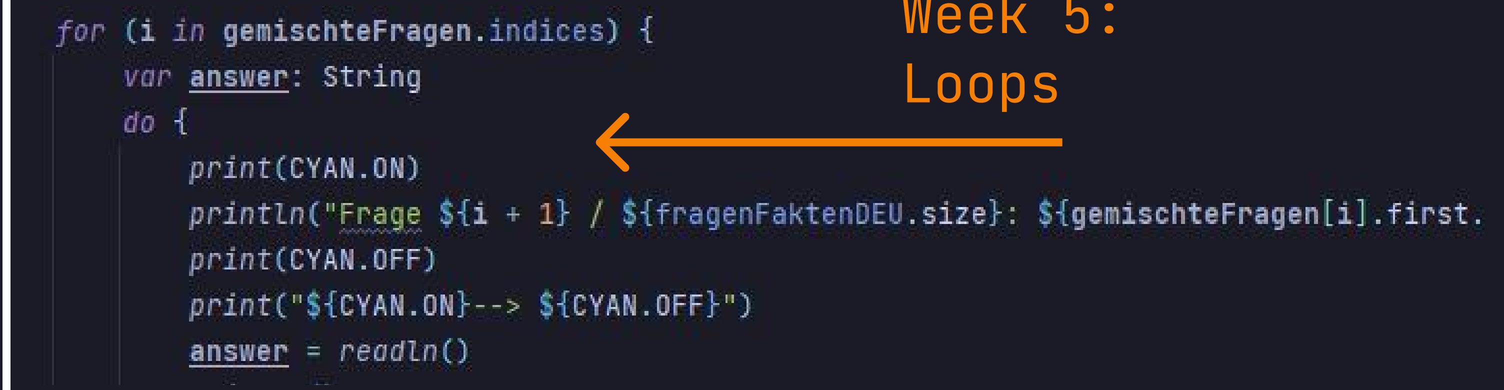
This is the function mentioned above. It wraps the text into a new line when it exceeds the limit ‘len’.



```kotlin
class RolleKarte(
    val name: String,
    val alter: Int,
    val herkunft: String,
    val geschlecht: String,
    val schicksal: String
)

val rollenKarte1_class = RolleKarte(
    name = "Akon",
    alter = 17,
    herkunft = "Südsudan",
    geschlecht = "weiblich",
    schicksal = "Aufgrund des Bürgerkr
)

val rollenKarte2_class = RolleKarte(
    name = "Carlos",
    alter = 42,
    herkunft = "Leon, Nicaragua",
    geschlecht = "männlich",
    schicksal = "Du wurdest von deinem
)
```

The RolleKarte class in this code represents a character card with attributes such as name, age, origin, gender, and destiny.

Week 6:
CLASSES



```kotlin
for (i in gemischteFragen.indices) {
    var answer: String
    do {
        print(CYAN.ON)
        println("Frage ${i + 1} / ${fragenFaktenDEU.size}: ${gemischteFragen[i].first.
        print(CYAN.OFF)
        print("${CYAN.ON}--> ${CYAN.OFF}")
        answer = readln()
```

Week 5:
Loops

This loop iterates over the indices of gemischteFragen, a shuffled list of game questions. It prompts the user to answer each question, displayed with cyan color, by inputting either '1' for 'yes' or '2' for 'no'. This prompt is repeated until a valid input is received.

## Customization and Further Development
## License and Contact Information

Users are encouraged to modify the game to fit different contexts, extend its linguistic reach, or even enhance its gameplay mechanics. The game's questions can be adjusted to suit various societal scenarios, and support for more languages can be added to increase its accessibility. Of course the code is well documented.

LICENSE & CONTRIBUTING: The game is licensed under the MIT license, granting users the freedom to use, modify, and distribute it. If you have improvements, new features, or language supports to propose, feel free to submit a pull request or open an issue on my GitHub page. For any inquiries or suggestions related to 'Privileges', you can reach out through my GitHub project page or visit https://www.borisniehaus.de for alternative contact methods. I'm always keen to hear your thoughts and ideas.

SOURCES: Privileges' has been developed by drawing insights from various sources, such as 'EINEN SCHRITT NACH VORN' by handicap-international.de and 'abgehängt - Ein Privilegienspiel' by bne-sachsen.de. These resources have greatly influenced the game's design and mechanics, providing valuable perspectives on privilege and societal dynamics.

Privileges is an idea conceived by Boris Niehaus. It was developed as part of the Mobile App Development course at the Syntax Institute - Copyright 2023.

Contact: bo.niehaus@gmail.com
Web: https://borisniehaus.de
Github: https://github.com/just1984