

Coding mit KI – Wenwen Marin

21.10.2024 – 31.10.2024:

Dragon Ball Statische Website + Dynamische Website Entwicklung

1. Originale intention kommt aus...

Derzeit arbeite ich als Nachhilfelehrerin für Mathematik und habe festgestellt, dass viele Schülerinnen wenig Motivation zeigen oder Schwierigkeiten haben, sich beim Mathematiklernen zu konzentrieren.

Ich habe lange darüber nachgedacht, wie ich ihnen helfen kann. Möglicherweise könnte eine Trainingswebsite im Comic-Stil eine Lösung sein, um Mathematikübungen für die Schülerinnen so spannend zu gestalten wie ein Computerspiel.

2. Ideenentwicklung & erste Rahmenstruktur der Website



Zuerst habe ich Copilot Chat mitgeteilt, dass ich eine Website im Dragon-Ball-Stil erstellen möchte. Sobald die Seite geladen wird, soll im Hintergrund Musik abgespielt werden, und die Figur 'Kid Goku' erscheint. Er fliegt von der oberen rechten Ecke zur Mitte der rechten Bildschirmseite und stoppt dann unten in der Mitte. Nachdem er angehalten hat, beginnt er sich zu bewegen, als würde er auf dem Wasser surfen, und sagt: 'Hello, DragonBall fan, let us train together!'.

Danach habe ich drei Dateien erstellt: 'index.html' für die Grundstruktur der Website, 'style.css' für das Layout und 'script.js' zur Steuerung der Animationen und Interaktionen.

Zu diesem Zweck habe ich nach Dragon-Ball-Musik auf YouTube gesucht und einen Online-Editor verwendet, um einen Teil davon zuzuschneiden und im MP3-Format als Hintergrundmusik zu speichern. Außerdem habe ich einige Dragon-Ball-Bilder von einer anderen Website heruntergeladen, um sie als Comic-Material zu verwenden.

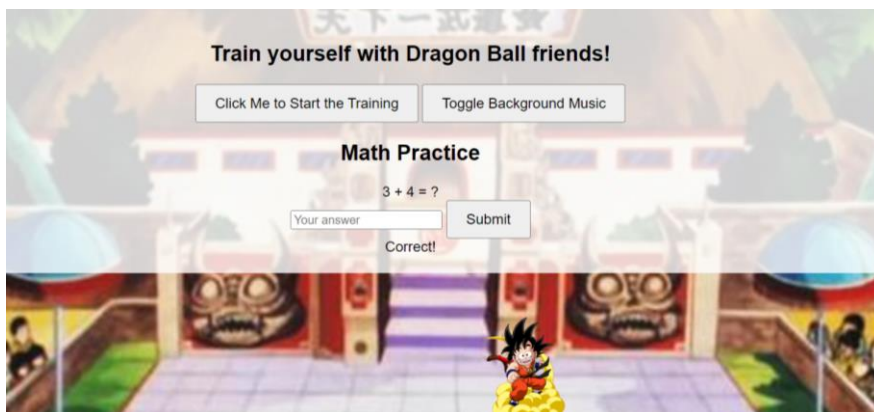
Daraufhin stieß ich auf mein erstes Problem: Die Hintergrundmusik wurde nicht automatisch abgespielt, als die Seite geladen wurde, obwohl ich im Code die Einstellungen 'autoplay' und 'loop' korrekt implementiert hatte. Ich wusste nicht, warum das so war, also stellte ich die Frage an ChatGPT.

```

<!-- Autoplay and loop attributes for continuous music playback -->
<audio id="background_music" src="background_music.mp3" autoplay loop></audio>
<script src="script.js"></script>
```

Ich habe von ChatGPT gelernt, dass Chromium-basierte Browser das automatische Abspielen von Medieninhalten in den meisten Fällen nicht erlauben. Einige Browser gestatten die Autoplay-Funktion nur dann, wenn der Ton stummgeschaltet ist, da automatisches Abspielen oft als eine negative Benutzererfahrung (UX) angesehen wird.

Anschließend habe ich einen Button mit der Aufschrift 'Click Me to Start the Training' hinzugefügt, um die Hintergrundmusik abzuspielen und Mathematikaufgaben anzuzeigen. Außerdem habe ich einen Button 'Toggle Background Music' integriert, mit dem die Musik ein- oder ausgeschaltet werden kann.

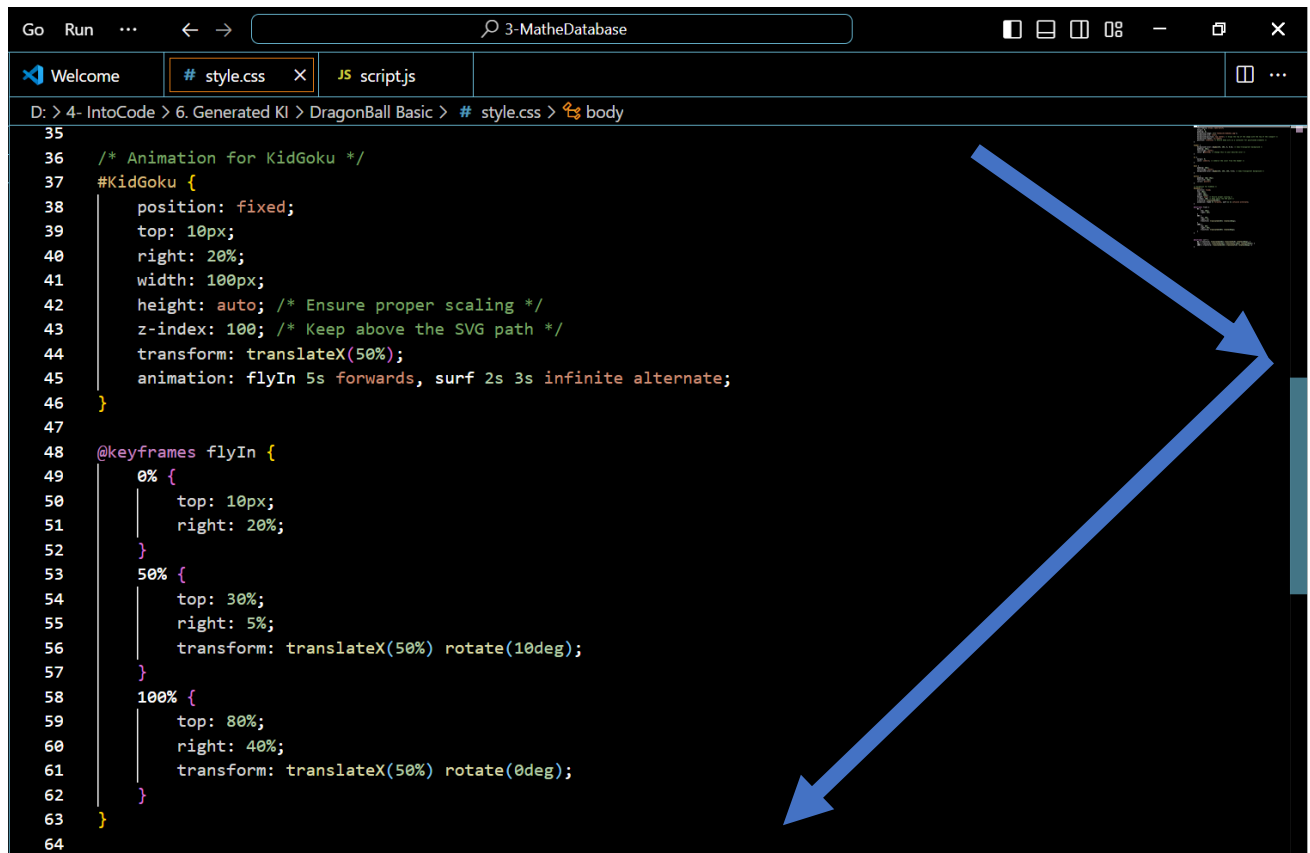


Die Mathematikfragen werden automatisch von dieser Funktion generiert. Es handelt sich um einfache Fragen, die sich gut eignen, um die Funktionen dieser Webseite zu testen. Wenn der Benutzer die richtige Antwort eingibt, erscheint der Feedback-Text 'Correct!'. Andernfalls wird 'Incorrect, try again.' angezeigt.

Bis jetzt habe ich JavaScript noch nicht gründlich gelernt; ich weiß nur, dass man damit bestimmte Funktionen umsetzen kann. Allerdings kann ich selbst keinen JavaScript-Code schreiben und weiß auch nicht, wie man eine einfache Mathematikaufgabe generiert. Daher war ich sehr überrascht, dass Copilot meine Absicht verstanden und den Code dafür eigenständig geschrieben hat.

```
Go Run ... 3-MatheDatabase
Welcome # style.css JS script.js X index.html
D:\> 4- IntoCode > 6. Generated KI > DragonBall Basic > JS script.js > ...
29 function startTraining() {
38 }
39
40 // Generate a new math problem
41 let correctAnswer;
42 function generateProblem() {
43   const num1 = Math.floor(Math.random() * 10) + 1;
44   const num2 = Math.floor(Math.random() * 10) + 1;
45   correctAnswer = num1 + num2;
46   document.getElementById('problem').innerText = `${num1} + ${num2} = ?`;
47 }
48
49 // Check if the user's answer is correct
50 function checkAnswer() {
51   const userAnswer = parseInt(document.getElementById('answer').value);
52   const feedback = document.getElementById('feedback');
53
54   // Provide feedback and generate a new problem
55   if (userAnswer === correctAnswer) {
56     feedback.innerText = 'Correct!';
57   } else {
58     feedback.innerText = 'Incorrect, try again.';
59   }
60   document.getElementById('answer').value = ''; // Clear the input after checking
61   generateProblem(); // Generate a new problem after checking the answer
62 }
63
```

Ich überlege, die Flugbahn von Kid Goku anzupassen. Momentan fliegt er in einem Zickzack-Muster, das sehr steif und abrupt wirkt. Ich möchte, dass er in einer sanften Kurve fliegt und sanft am Boden landet. Egal, wie oft ich die Frage anders formulierte, Copilot gab mir immer eine ähnliche Antwort. Bei diesem Problem konnte es mir daher nicht weiterhelfen.



```
35
36  /* Animation for KidGoku */
37  #KidGoku {
38    position: fixed;
39    top: 10px;
40    right: 20%;
41    width: 100px;
42    height: auto; /* Ensure proper scaling */
43    z-index: 100; /* Keep above the SVG path */
44    transform: translateX(50%);
45    animation: flyIn 5s forwards, surf 2s 3s infinite alternate;
46  }
47
48  @keyframes flyIn {
49    0% {
50      top: 10px;
51      right: 20%;
52    }
53    50% {
54      top: 30%;
55      right: 5%;
56      transform: translateX(50%) rotate(10deg);
57    }
58    100% {
59      top: 80%;
60      right: 40%;
61      transform: translateX(50%) rotate(0deg);
62    }
63  }
64
```

Am nächsten Tag erklärte mir Herr Charbonnier, dass ich eine Bézier-Kurve als Flugbahn verwenden kann. Dazu muss ich im Code lediglich vier Punkte angeben: den Startpunkt (P0), den Kontrollpunkt 1 (P1), den Kontrollpunkt 2 (P2) und den Endpunkt (P3). So wird Kid Goku so fliegen, wie ich es mir vorgestellt habe.

Später wurde mir klar, dass KI-Assistenten sehr präzise Anweisungen benötigen. Wenn ich zum Beispiel eine klare Anweisung wie ‚Zeichne eine Bézier-Kurve‘ gebe, ist das weitaus genauer und hilfreicher als eine vage Anweisung wie ‚Bewege dich entlang der Kurve‘.

Deshalb brauchen wir oft dennoch einen menschlichen Mentor, da menschliches Verständnis oft einfühlsamer ist und ein besseres Gespür für unsere Absichten hat.

Bézier curve

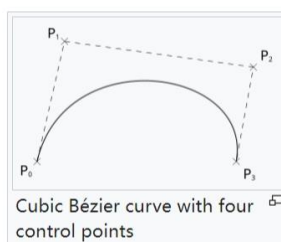
44 languages

Article Talk

Read Edit View history Tools

From Wikipedia, the free encyclopedia

A **Bézier curve** (/ˈbɛz.i.eɪ/ *BEH-zee-ay*)^[1] is a **parametric curve** used in **computer graphics** and related fields.^[2] A set of discrete "control points" defines a smooth, continuous curve by means of a formula. Usually the curve is intended to approximate a real-world shape that otherwise has no mathematical representation or whose representation is unknown or too complicated. The Bézier curve is named after **French** engineer **Pierre**



Allerdings hatte ich bei der Anwendung der Bézier-Kurve große Schwierigkeiten. Copilot verstand meine Absichten nicht vollständig, und der Code, den er mir gab, ließ Kid Goku chaotisch fliegen. Egal, wie ich die Parameter änderte oder andere Fragen stellte, das Ergebnis war nicht besser als der ursprüngliche gerade Flug. Schließlich führte dies sogar dazu, dass die gesamte Webseite keine Reaktion mehr zeigte. Ich war äußerst frustriert.

Zum Glück gelang es mir, die ursprüngliche Version der Webseite mit dem geradlinigen Flug und den funktionierenden Features wiederherzustellen. Aus dieser Erfahrung habe ich eine wichtige Lektion gelernt: Zukünftig werde ich jeden funktionierenden Code-Stand als Meilenstein speichern. Neue Ideen werde ich in einem separaten Ordner ausprobieren und erst bei Erfolg schrittweise in die Hauptdatei integrieren.

3. Bézier-Kurve als Flugbahn verwenden und gleichzeitig die Kurve anzeigen

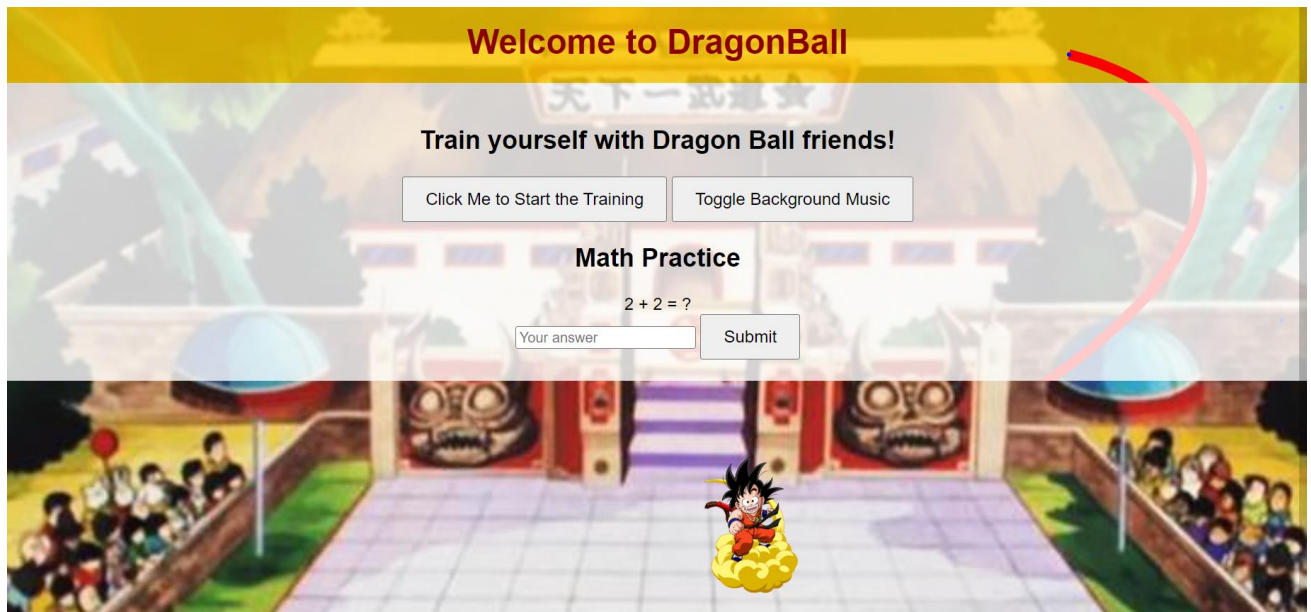


```
Go Run ... < > 3-MatheDatabase
Welcome # style.css JS script.js index.html X
D: > 4- IntoCode > 6. Generated KI > 1-Flying > <> index.html > html > body
2 <html lang="en">
9 <body>
15 <svg>
16 <path id="flying-path" d="M 1000 50 C 1200 100, 1200 300, 700 500"/>
17 <!-- Start point -->
18 <circle cx="1000" cy="50" r="2" fill="blue"/>
19 <!-- Control point 1 -->
20 <circle cx="1200" cy="100" r="2" fill="blue"/>
21 <!-- Control point 2 -->
22 <circle cx="1200" cy="300" r="2" fill="blue"/>
23 <!-- End point -->
24 <circle cx="700" cy="500" r="2" fill="blue"/>
25 </svg>
26
27 
28 <script src="script.js"></script>
29 </body>
30 </html>
```

Copilot hat mir eine `<svg>` Code geschrieben. ChatGPT hat mir gut erklärt, dass das HTML-Element `<svg>` ein Container für SVG-Grafiken ist.

SVG bietet verschiedene Methoden zum Zeichnen von Pfaden, Kreisen und vielem mehr.

Anschließend habe ich die beiden Inhalten 'Kurvenflug' und 'Flugbahn zeichnen' in die Dateien der Basic-Version integriert. Die Webseite enthält nun die Funktionen zur Steuerung der Hintergrundmusik, den Kurvenflug von Kid Goku mit gezeichneter Flugbahn sowie einfache Mathematikaufgaben – dies ist jetzt **Version 2**.



In diesem Prozess bin ich auf ein Problem gestoßen: Der Button zur Steuerung der Musik funktionierte plötzlich nicht mehr. Nach verschiedenen Rückfragen bei der KI stellte sich heraus, dass die Schicht mit der gezeichneten Flugbahn den Button überdeckte. Dabei habe ich ein neues Konzept gelernt: den „z-index“.

Die z-index-Eigenschaft legt die Stapelreihenfolge eines Elements fest.

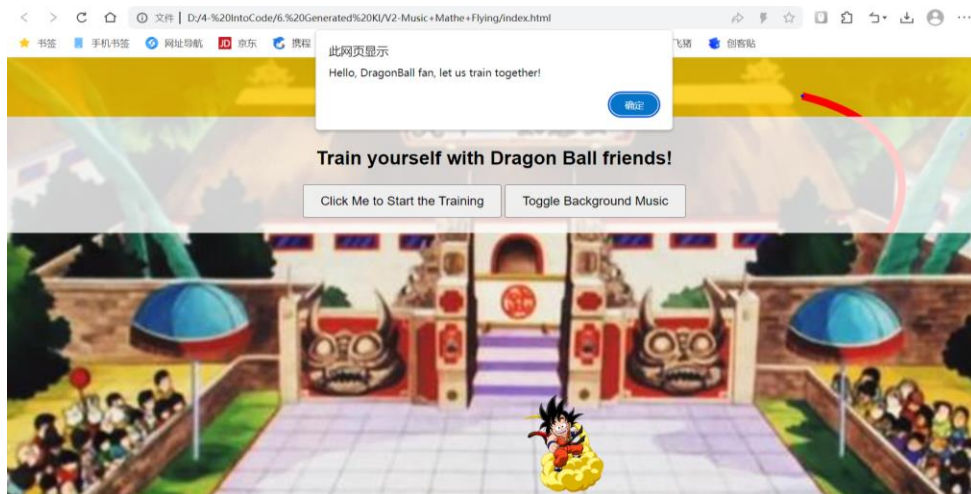
Ein Element mit einer höheren Stapelreihenfolge wird immer vor einem Element mit einer niedrigeren Stapelreihenfolge angezeigt.

Daraufhin habe ich die z-index-Werte der verschiedenen Ebenen angepasst: Das Hintergrundbild erhielt den niedrigsten z-index von 1, gefolgt von der Flugbahn mit z-index 10, dann die Ebene mit den Buttons und Mathematikaufgaben ('main') mit z-index 20, und Kid Goku liegt mit z-index 21 ganz oben.

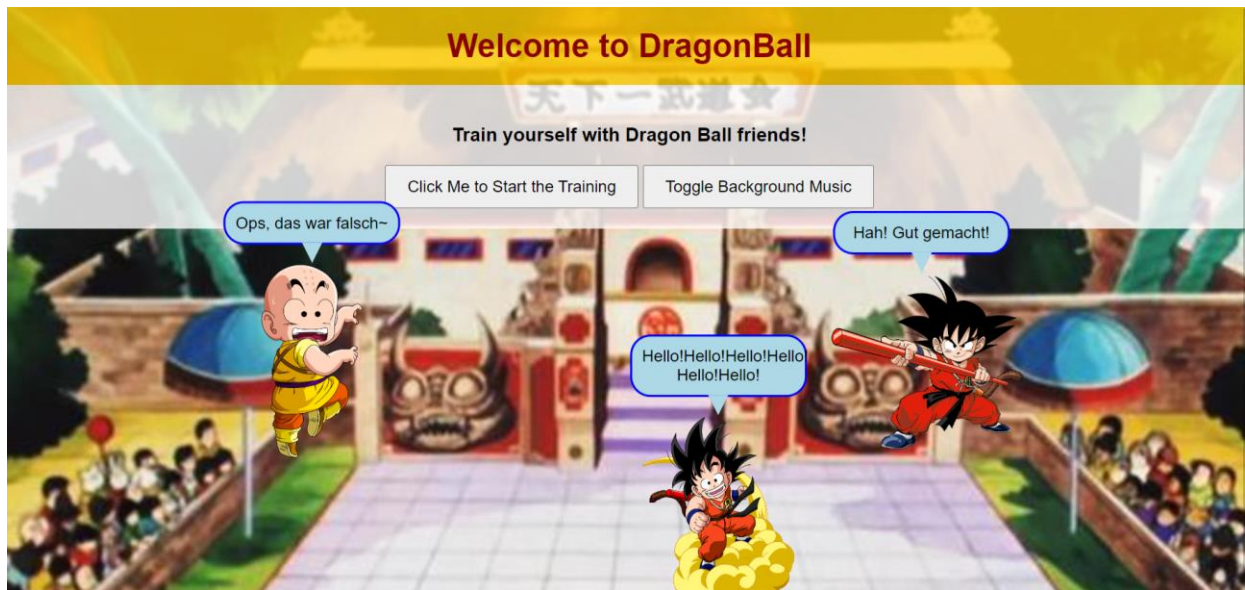

```
D: > 4- IntoCode > 6. Generated KI > V2-Music+Mathe+Flying > # style.css > body

1 body {
2   font-family: Arial, sans-serif;
3   margin: 0;
4   padding: 0;
5   background-image: url('tenkaichi-budokai.jpg');
6   background-size: cover;
7   background-position: top center; /* Align the top of the image with the top of the viewport */
8   background-repeat: no-repeat;
9   position: relative; /* Ensure body acts as a container for positioned elements */
10  z-index: 1; /* Lowest z-index for the background image */
11 }
12
25 main {
26   padding: 20px;
27   text-align: center;
28   background-color: rgba(255, 255, 255, 0.8); /* Semi-transparent background */
29   position: relative; /* Ensure main content is above the SVG */
30   z-index: 20; /* Higher z-index to ensure buttons are clickable */
31 }
32
33
40 /* Animation for KidGoku */
41 #KidGoku {
42   position: absolute; /* Changed to absolute for better control during animation */
43   width: 100px;
44   height: auto; /* Ensure proper scaling */
45   z-index: 21; /* Keep above the SVG path(z=10) and the main(z=20) */
46   transform: translate(-50%, -50%); /* Center the image on the path */
47 }
48
49
66 svg {
67   position: absolute;
68   top: 0;
69   left: 0;
70   width: 100%;
71   height: 100%;
72   z-index: 10; /* Ensure the SVG is above the background */
73 }
74 }
```

An diesem Moment bin ich noch mit einer Kleinigkeit unzufrieden: Nachdem Kid Goku seinen Flug beendet, erscheint die Begrüßung aktuell als 'Alert'-Fenster. Diese Methode wirkt etwas abrupt und unpassend. Stattdessen möchte ich die Begrüßung in Form einer Sprechblase anzeigen.



4. Sprechblase anzeigen



Dieser Schritt ist relativ einfach: Es genügt, den Stil der Sprechblase mit CSS festzulegen und zusätzlich zwei Bilder von Dragon-Ball-Charakteren herunterzuladen.

5. Version 3 = Music + Mathe + Fliegen + Sprechblase + Stimme

Als ich die Sprechblase in Version 2 integriert habe, fand ich es etwas eintönig, nur den Text anzuzeigen. **Es wäre viel besser, wenn die Figur die Begrüßung direkt sprechen könnte. Da fiel mir ein, dass im Kursmaterial die KI-gestützte Sprachsynthese-Website 'play.ht' erwähnt wurde.**

Allerdings kann auf dieser Website nur englische Sprachausgabe synthetisiert werden, und die deutsche Aussprache klingt nicht gut. Daher habe ich den Vorschlag von ChatGPT angenommen und eine andere KI-gestützte Sprachausgabe-Website namens 'FakeYou' verwendet.

FakeYou bietet eine Auswahl an verschiedenen Charakterstimmen. Einige Stimmen aus 'Dragon Ball' sind bereits verfügbar. Die synthetisierte Stimme klingt jetzt realistisch und unterhaltsam und wirkt nicht mehr so ernst und förmlich.

In Version 3:

Wenn die Website geöffnet wird, fliegt Kid Goku sanft entlang einer Kurve zum unteren Bildschirmrand und beginnt dann, auf und ab zu surfen. Wenn man ihn anklickt, hört man Kid Gokus Stimme, die sagt: 'Hello Dragon Ball fan, Let's train together.' Beim Klicken des 'Click Me to Start the Training'-Buttons startet die Dragon-Ball-Hintergrundmusik und eine Mathematikaufgabe wird angezeigt. Falls die Musik zu laut ist, kann man sie über den 'Toggle Background Music'-Button stummschalten. Bei einer richtigen Antwort erscheint ein trainierender kleiner Kid Goku und sagt: 'Very Good, Well Done!' Bei einer falschen Antwort erscheint ein besorgter kleiner Krillin und sagt: 'Oops, that was wrong.'



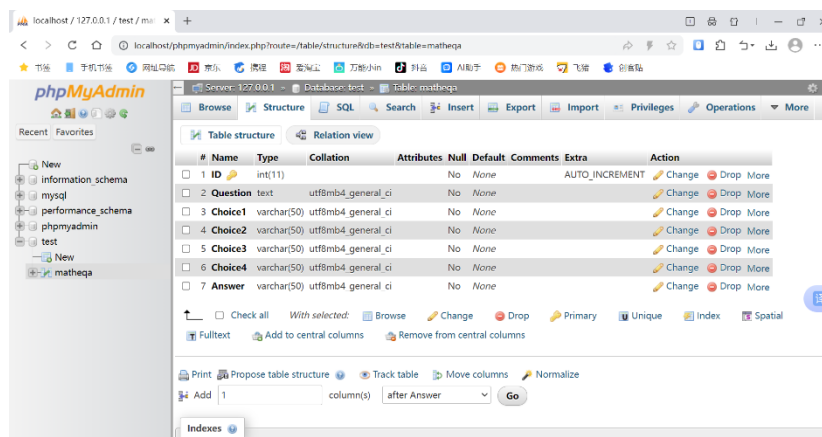
Tatsächlich bin ich jetzt sehr zufrieden. Es gibt jedoch eine kleine Sache, die ich gerne noch verbessern würde: Derzeit ist die gesamte Website statisch, aber ich möchte eine dynamische Website erstellen, um auch mein Wissen in PHP und SQL anzuwenden. Dazu müsste ich eine Datenbank einrichten, um verschiedene Mathematikaufgaben zu speichern. Die Aufgaben könnten dann dynamisch und zufällig ausgewählt und auf der Website angezeigt werden. Anschließend könnte der Benutzer eine Antwort auswählen, die automatisch auf Richtigkeit überprüft wird.

6. Eine Datenbank einrichten und Mathematikaufgaben eingeben

Dieser Schritt war äußerst schwierig. **Unter der Anleitung von Copilot und ChatGPT habe ich viele Methoden ausprobiert. Obwohl der Code jeweils gut aussah, scheiterten alle Versuche. Mein Wissen reichte nicht aus, und ich wusste nicht, warum es nicht funktionierte. Während dieses Prozesses konnte ich Copilot zudem versehentlich nicht mehr verwenden. Später fand ich vom Herrn Charbonnier heraus, dass dies daran lag, dass ich den 'Workspace' gewechselt hatte.**

Jetzt beschreibe ich nur die Methode, die schließlich erfolgreich ist:

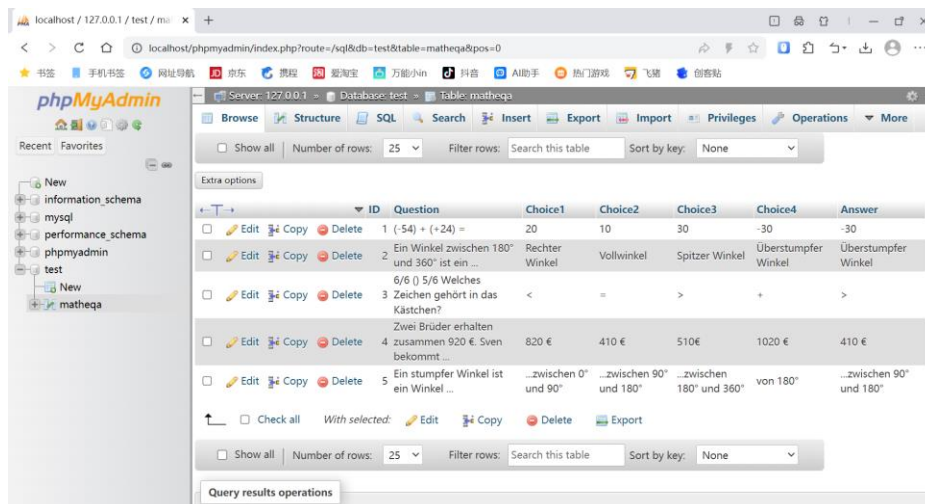
- 1) Ich habe einige Mathematikaufgaben von einer anderen Website kopiert und die Aufgabenstellung, vier Antwortmöglichkeiten sowie die richtige Antwort in einer Datei namens 'Problems-7th class.txt' gespeichert.
- 2) Öffne das XAMPP Control Panel, starte Apache und MySQL, und klicke auf 'Admin'. Erstelle in der Datenbank 'test' eine Tabelle namens 'matheqa'. Die Spaltenüberschriften der Tabelle lauten: ID, Question, Choice1, Choice2, Choice3, Choice4 und Answer.



- 3) Nutze Copilot, um ein Python-Skript zu schreiben, das automatisch die Mathematikaufgaben aus der Datei „Problems-7th class.txt“ in die „matheqa“-Tabelle einfügt.

```
10 mysql_cursor = mysql_conn.cursor()
11
12
13
14 # Read the content of the text file
15 with open('Problems-7th class.txt', 'r', encoding='utf-8') as file:
16     lines = file.readlines()
17
18 # Insert the data into the MySQL matheqa table
19 def insert_matheqa(Question, Choice1, Choice2, Choice3, Choice4, Answer):
20     mysql_cursor.execute('''
21         INSERT INTO matheqa (Question, Choice1, Choice2, Choice3, Choice4, Answer)
22         VALUES (%s, %s, %s, %s, %s, %s)
23     ''', (Question, Choice1, Choice2, Choice3, Choice4, Answer))
24
25 for i in range(0, len(lines), 6):
26     Question = lines[i].strip()
27     Choice1 = lines[i+1].strip()
28     Choice2 = lines[i+2].strip()
29     Choice3 = lines[i+3].strip()
30     Choice4 = lines[i+4].strip()
31     Answer = lines[i+5].strip()
32
33     insert_matheqa(Question, Choice1, Choice2, Choice3, Choice4, Answer)
34
35 # Commit the transaction
36 mysql_conn.commit()
37
38 # Close the connections
39 mysql_conn.close()
```

Ich selbst kann auch in Python programmieren, aber Copilot schreibt den Code tatsächlich schneller und meist fehlerfrei. Dennoch bin ich der Meinung, dass wir Programmierkenntnisse gründlich erlernen sollten, da wir uns nicht vollständig auf KI-Assistenten verlassen können. Wir sollten in der Lage sein, den von ihnen generierten Code zu verstehen und ihn entsprechend unseren eigenen Anforderungen anzupassen



- 4) Erstelle im Verzeichnis „C:\xampp\htdocs\php“ eine neue Datei namens „Mathe.php“, um die folgenden Schritte zu testen: Verbinde mit der Datenbank, wähle eine zufällige Frage aus der Tabelle „matheqa“ aus und zeige die Frage sowie die Antwortmöglichkeiten und die richtige Antwort auf der Webseite an.



7. Die finale Version: Music + Fliegen + Sprechblase + Stimme + Dynamische Mathe Frage

Der letzte Schritt besteht darin, die Funktion zu integrieren, die zufällig mathematische Aufgaben aus der Datenbank auswählt und im Bereich 'math-practice section' anzeigt. Dadurch wird die statische Webseite in eine dynamische Webseite umgewandelt.

```

9 <body>
13 <main>
15 <button id="start-training" onclick="startTraining()">Click Me to Start the Training</button> <!--
16 <button onclick="toggleMusic()">Toggle Background Music</button> <!-- New button to toggle music --
17
18 <!-- Math Practice Section -->
19 <section id="math-practice" style="display: none;">
20 <h2>Math Practice</h2>
21 <?php include "Mathe.php";?>
22 <!-- Form for the random question -->
23 <form method="post" action="">
24 <!-- Displaying question and answer options fetched from PHP -->
25 <div id="question">
26 <p><?php echo $row['ID'] . ". " . htmlspecialchars($row['Question']); ?></p>
27 <input type="hidden" name="question_id" value="<?php echo $row['ID']; ?>">
28 <input type="radio" id="choice1" name="choice" value="<?php echo htmlspecialchars($row['Choice1
29 <input type="radio" id="choice2" name="choice" value="<?php echo htmlspecialchars($row['Choice2
30 <input type="radio" id="choice3" name="choice" value="<?php echo htmlspecialchars($row['Choice3
31 <input type="radio" id="choice4" name="choice" value="<?php echo htmlspecialchars($row['Choice4
32 </div>
33 <!-- Pass the correct answer to the JavaScript function on button click -->
34 <button type="button" onclick="checkAnswer('<?php echo $row['Answer']; ?>')">Submit</button>
35 <button type="button" onclick="loadNewQuestion()">New Question</button>
36 </form>

```

`<?php include "Mathe.php" ;?>`

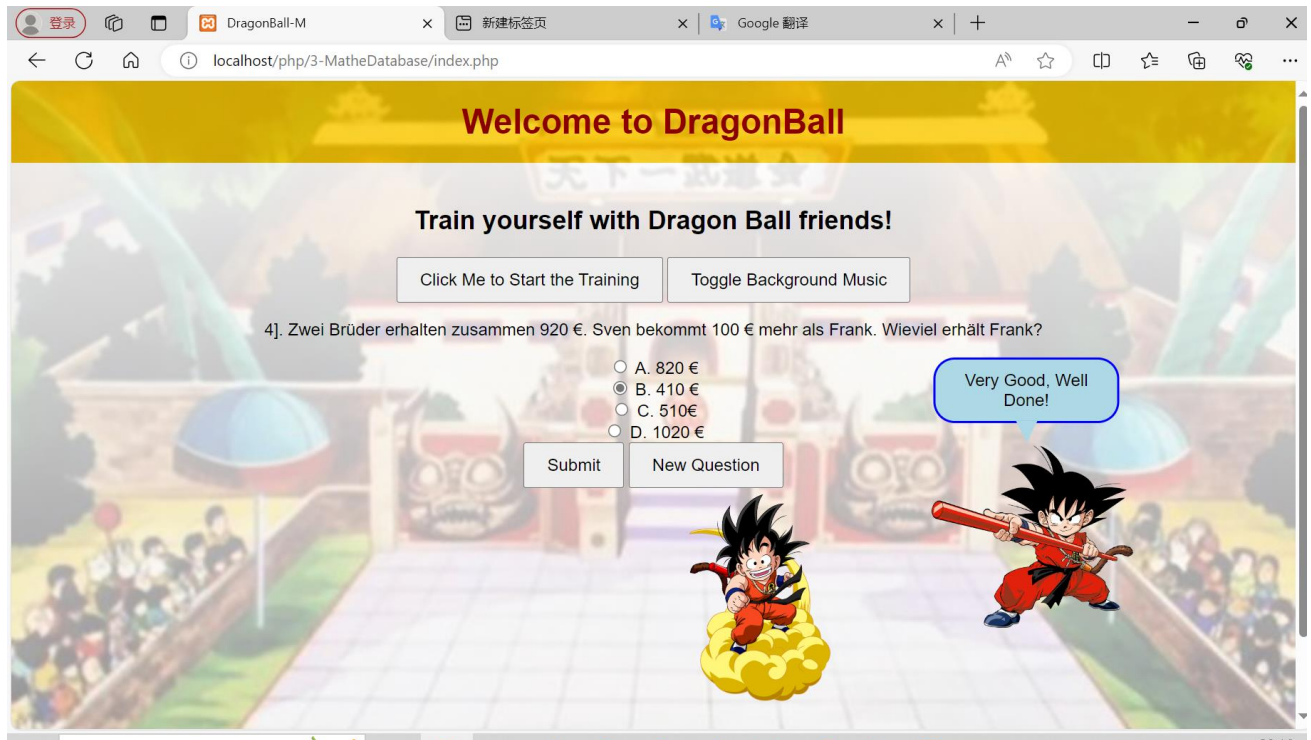
Die Datei 'Mathe.php' wird in den Bereich 'math-practice section' integriert.

```

67 }
68 //This function will use fetch to call Mathe.php and update the "math-practice" section without reloading t
69 function loadNewQuestion() {
70   fetch('fetch_question.php')
71     .then(response => response.text())
72     .then(data => {
73       document.getElementById('math-practice').innerHTML = data;
74     })
75     .catch(error => console.error('Error fetching new question:', error));
76 }
77
78 // Function to check if the selected answer is correct
79 function checkAnswer(correctAnswer) {
80   // Get the selected choice
81   const choices = document.getElementsByName('choice');
82   let userAnswer = null;
83   for (const choice of choices) {
84     if (choice.checked) {
85       userAnswer = choice.value;
86       break;
87     }
88   }
89 }

```

Diese Funktion wird fetch verwenden, um "Mathe.php" aufzurufen und den Bereich 'math-practice section ' zu aktualisieren, ohne die Seite neu zu laden. Um zu verhindern, dass Kid Goku jedes Mal neu fliegen muss, wenn ein neues Problem generiert wird. Ich hatte diese Idee zur Verbesserung, und Copilot half mir dabei, sie umzusetzen.



Zusammenfassung meiner Erfahrungen mit KI-gestützter Programmierung:

In meinem Programmierprojekt habe ich intensiv mit KI-Tools wie GitHub Copilot und ChatGPT gearbeitet und dabei wertvolle Erkenntnisse gewonnen. Trotz meiner grundlegenden Programmierkenntnisse in Python hatte ich in anderen Bereichen, wie JavaScript, nur wenig Erfahrung. Die KI-Tools erwiesen sich hier als äußerst nützlich: Sie halfen mir nicht nur, komplexere Aufgaben umzusetzen, sondern unterstützten mich auch dabei, neue Funktionen effizient und präzise zu integrieren. Besonders beeindruckt hat mich, wie Copilot meine Anweisungen oft korrekt interpretieren und eigenständig vollständigen Code generieren konnte, den ich direkt in mein Projekt einbauen konnte.

Trotz dieser Unterstützung wurde mir jedoch klar, dass KI-Assistenten genaue und umsetzbare Anweisungen benötigen, um optimale Ergebnisse zu liefern. Ich erkannte die Bedeutung präziser Anweisungen – spezifizierte Fragen wie „Zeichne eine Bézier-Kurve“ führten zu besseren Ergebnissen als vage Formulierungen. Auch zeigte sich, dass Copilot nicht jede Fragestellung lösen konnte und oft ähnliche Antworten lieferte, selbst wenn ich die Anfragen variierte.

Ein weiterer Lernpunkt war das Gleichgewicht zwischen Effizienz und Verständnis. Die

Geschwindigkeit und Genauigkeit, mit der Copilot Code schreiben kann, ist beeindruckend. Dennoch halte ich es für entscheidend, weiterhin eine fundierte Programmierausbildung zu verfolgen. So kann ich die von der KI generierten Codes verstehen, bewerten und gezielt anpassen.

Insgesamt hat die Zusammenarbeit mit den KI-Tools meine Arbeitsweise verbessert und mir wertvolle Einsichten in die Potenziale und Grenzen von KI in der Programmierung gegeben. Gleichzeitig betonte diese Erfahrung die Wichtigkeit menschlicher Mentoren, deren Verständnis für komplexe Zusammenhänge und Absichten oft hilfreicher ist. Der richtige Einsatz von KI-Tools und ein solides Fundament in Programmierkenntnissen erscheinen mir daher als idealer Weg, um langfristig erfolgreich zu programmieren.

Fazit:

- Die finale Version ist ein nahezu perfektes Ergebnis. Jede Funktion wird gemäß den Anforderungen präzise ausgeführt. Dies wäre nicht ohne die Unterstützung von Copilot, ChatGPT und Herrn Charbonnier möglich gewesen.
- Beim Einsatz von KI-Tools ist es wichtig, präzise und sinnvoll umsetzbare Fragen zu stellen. Andernfalls kann es passieren, dass die KI ungeeignete oder verwirrende Antworten liefert.
- KI-Tools sind sehr leistungsfähig, weisen jedoch auch Mängel auf, weshalb wir ihnen nicht vollständig vertrauen oder blind folgen sollten. Es ist wichtig, die Vorschläge kritisch zu prüfen und vor der Integration in das größere Projekt zunächst selbst zu testen, ob der Code unseren Erwartungen entspricht. Nach erfolgreichen kleineren Tests sollten diese in das Gesamtsystem integriert werden. Zudem ist es ratsam, regelmäßig Sicherungskopien anzufertigen, um zu vermeiden, dass das gesamte Projekt verloren geht.
- Risiko: Beim Einsatz von KI-Assistenten können mehrere potenzielle Unsicherheiten auftreten, insbesondere im Bereich der Datensicherheit, der Zuverlässigkeit und des Datenschutzes. Auch wenn mein derzeitiges einfaches Website-Projekt wahrscheinlich nicht das Ziel von Hackerangriffen wird, nehmen potenzielle Sicherheitsrisiken mit dem Ausbau der Seite und der Hinzufügung neuer Funktionen und Einstellungen exponentiell zu. Daher ist es für uns Programmierer wichtig, stets wachsam und vorsichtig zu bleiben und unser Wissen über Sicherheit kontinuierlich zu vertiefen. Andernfalls laufen wir Gefahr, Fehler zu machen, ohne es überhaupt zu bemerken.