

Zadanie: Naprawa stylów CSS w quizie

opracował: Rafał Taraszka

Opis zadania

Poniżej znajduje się kod CSS z **celowo wprowadzonymi błędami i nieprawidłowymi ustawieniami**.

Twoim zadaniem jest odnaleźć i **naprawić style**, aby quiz wyglądał estetycznie, był czytelny i zgodny z zamysłem interfejsu z lekcji.

Dodatkowo przy każdej grupie stylów znajdziesz **krótkie wyjaśnienie, do czego służą te właściwości CSS**.

Instrukcje krok po kroku

1. Reset stylów

```
.quiz-box {  
  background: #fff;  
  padding: 0px;           /* ✗ Popraw na np. 30px */  
  border-radius: 0px;     /* ✗ Zaokrąglij np. 8px */  
  box-shadow: 0 0 10px rgba(0, 0, 0, 0.9); /* ! Zbyt ciemny cień */  
  max-width: 600px;  
  width: 100%;  
  text-align: right;      /* ✗ Popraw na center lub left */  
}
```

```
.quiz-box h2 {  
  font-size: 8px; /* ✗ Zbyt mały tekst */  
  margin-bottom: 20px;  
}
```

```
.answer-buttons li {  
  margin: 0px 0; /* ✗ Dodaj odstęp między przyciskami */  
}  
  
.answer-buttons button {  
  width: 100%;  
  padding: 12px 20px;  
  font-size: 16px;  
  border: 2px solid #007bff;  
  border-radius: 5px;  
  background-color: white;  
  color: #007bff;  
  cursor: pointer;
```

```
transition: 0.3s;  
}
```

```
.score {  
  margin-top: 15px;  
  font-weight: bold;  
  color: #e3e3e3; /* ✗ Zbyt jasny – słabo widoczny na jasnym tle */  
}
```

Podsumowanie

Znalezione błędy:

- Zbyt małe fonty (h2, text-align: right)
- Brak marginesów między odpowiedziami
- Zbyt ciemny cień
- Zbyt jasny tekst wyniku
- Brak pełnych stylów w .parallax dla quizu i gry

☒ Zadanie dla Ciebie:

- Znajdź te fragmenty w kodzie CSS.
- Popraw style tak, aby quiz wyglądał estetycznie i był czytelny.
- Uruchom stronę w przeglądarce i sprawdź efekt!

Zadanie: Rozbudowa quizu – co możesz zmienić?

Poniżej znajdziesz propozycje, jak samodzielnie modyfikować i rozwijać quiz, który stworzyliśmy wcześniej. Zaczynij od prostych zmian, a potem spróbuj dodać nowe funkcje!

1. Zmiana i dodanie pytań

W pliku JavaScript znajdź tablicę `questions` i:

- Zmień treść istniejących pytań i odpowiedzi.
- Dodaj własne pytania w tym samym formacie:

```
{  
  question: "Tutaj wpisz swoje pytanie",  
  answers: [  
    { text: "Odpowiedź 1", correct: false },
```

```
{ text: "Odpowiedź 2", correct: true },  
{ text: "Odpowiedź 3", correct: false },  
{ text: "Odpowiedź 4", correct: false }  
]  
}
```

Dodanie timera (licznika czasu) do quizu

1. 🧩 Dodaj element HTML do wyświetlania czasu

W pliku `index.html`, w obrębie sekcji quizu (`.quiz-box`), dodaj element na timer:

```
<p id="timer" class="timer">Czas: 10s</p>
```

2. 🎨 Dodaj styl CSS dla timera (opcjonalnie)

W pliku `style.css`:

```
.timer {  
  font-weight: bold;  
  color: #dc3545;  
  margin-bottom: 10px;  
  font-size: 18px;  
  text-align: center;  
}
```

3. 🧠 Dodaj logikę timera w JavaScript

W pliku `script.js`:

🔑 Krok 1 – Zmienne globalne:

Dodaj na górze, obok `score`, `currentQuestionIndex` itp.:

```
let timeLeft = 10;  
let timer;
```

🔑 Krok 2 – Funkcja `startTimer()`

Dodaj ją nad funkcją `showQuestion()` lub w dowolnym miejscu w pliku:

```
function startTimer() {
  clearInterval(timer); // zatrzymaj poprzedni timer (jeśli jakiś był)
  timeLeft = 10;
  document.getElementById("timer").innerText = `Czas: ${timeLeft}s`;

  timer = setInterval(() => {
    timeLeft--;
    document.getElementById("timer").innerText = `Czas: ${timeLeft}s`;
    if (timeLeft <= 0) {
      clearInterval(timer);
      nextButton.click(); // automatycznie przejdź dalej
    }
  }, 1000);
}
```

🔑 Krok 3 – Uruchom timer przy każdym nowym pytaniu

W funkcji showQuestion(), po questionElement.innerText = ... dodaj:

```
startTimer();
```

🔑 Krok 4 – Zatrzymaj timer po udzieleniu odpowiedzi

W funkcji selectAnswer(e) dodaj na początku:

```
clearInterval(timer);
```

Dzięki temu, jeśli użytkownik odpowie, odliczanie się zatrzyma.

☑ Finalny efekt

Licznik pokazuje Czas: 10s przy każdym pytaniu.

Jeśli uczeń nie kliknie odpowiedzi w ciągu 10 sekund — quiz automatycznie przejdzie do kolejnego pytania.

Jeśli uczeń kliknie wcześniej — licznik zatrzymuje się.

Instrukcja: Naprawa przycisku „Zagraj ponownie”

🔍 Problem

Po zakończeniu quizu wyświetla się przycisk "**Zagraj ponownie**", ale jego kliknięcie nic nie robi – quiz nie uruchamia się od nowa.

Dlaczego? Ponieważ po zmianie tekstu przycisku nie został przypisany ponownie kod, który uruchamia quiz od początku.

✓ Rozwiązanie

Krok 1: Zmodyfikuj funkcję `nextButton.addEventListener(...)`

Zamień istniejącą funkcję:

```
nextButton.addEventListener("click", () => {
  currentQuestionIndex++;
  if (currentQuestionIndex < questions.length) {
    showQuestion();
  } else {
    showScore();
  }
});
```

Na wersję z obsługą trybu restartu quizu:

```
nextButton.addEventListener("click", () => {
  if (nextButton.innerText === "Zagraj ponownie") {
    startQuiz();
  } else {
    currentQuestionIndex++;
    if (currentQuestionIndex < questions.length) {
      showQuestion();
    } else {
      showScore();
    }
  }
});
```

🔄 Co się zmieniło?

Sprawdzamy, czy tekst przycisku to "Zagraj ponownie".

Jeśli tak – wywołujemy ponownie `startQuiz()`.

Dzięki temu quiz resetuje się i zaczyna od nowa.

💡 Dodatkowo – reset timera na nowym starcie

W funkcji `startQuiz()` dodaj też:

```
clearInterval(timer);
document.getElementById("timer").innerText = "";
```

Aby wyczyścić stary timer zanim zacznie się nowy quiz.

✦ Efekt końcowy

Quiz działa normalnie.

Po ostatnim pytaniu pokazuje się wynik i przycisk „Zagraj ponownie”.

Po kliknięciu tego przycisku quiz się resetuje i zaczyna od początku.