

# Warsztaty: Strona z Paralaksą + Quiz + Gra w Zgadywanie Słów

---

opracował: Rafał Taraszka

Wstęp (5–10 minut)

linki

[Prezentacja ZST](#)

[Quiz](#)

[Strona Parallax](#)

[Word guessing game](#)

Cel: Uczestnicy:

- Poznają podstawy tworzenia stron z wykorzystaniem HTML, CSS i JavaScript.
- Zrozumieją, jak działa efekt paralaksy (ang. parallax) na stronach.
- Stworzą własny quiz i grę słowną.

Co to jest HTML, CSS i JavaScript?

- HTML: struktura strony.
- CSS: stylowanie, układ, animacje.
- JavaScript: interaktywność, logika aplikacji (tu: quiz i gra).

Pokaz końcowego efektu:

- Pokaż docelową stronę z parallax, quizem i grą słowną (np. u siebie na komputerze/rzutniku).

Przygotowanie plików i struktury projektu (5 minut)

1. Załóż folder projektu, np. warsztaty.

2. Wewnątrz utwórz trzy pliki:

- index.html – kod HTML.
- style.css – style CSS.
- script.js – skrypt JavaScript.

Uwaga: Dla uproszczenia cały kod JS umieścimy w jednym pliku. Przy większych projektach można stosować podział, np. osobno quiz.js i game.js.

## Podstawowa struktura HTML (10 minut)

Poniższy kod wklej (lub przepisz) do pliku index.html:

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="UTF-8">
  <title>Warsztaty: Paralaksa + Quiz + Word Guessing</title>
  <link rel="stylesheet" href="style.css">
</head>
<body>

  <!-- Sekcja nawigacyjna (opcjonalnie) -->
  <header>
    <nav>
      <ul>
        <li><a href="#parallax-section">Paralaksa</a></li>
        <li><a href="#quiz-section">Quiz</a></li>
        <li><a href="#game-section">Gra</a></li>
      </ul>
    </nav>
  </header>

  <!-- Sekcja z efektem paralaksy -->
  <section id="parallax-section" class="parallax">
    <div class="parallax-content">
      <h1>Witamy na naszych warsztatach!</h1>
      <p>Przeviń w dół, aby przejść do quizu i gry</p>
    </div>
  </section>

  <!-- Sekcja z quizem -->
  <section id="quiz-section" class="quiz-container">
    <h2>Quiz</h2>
    <div id="quiz"></div>
    <button id="nextButton">Następne pytanie</button>
  </section>

  <!-- Sekcja z grą w zgadywanie słów -->
  <section id="game-section" class="game-container">
    <h2>Gra w zgadywanie słów</h2>
    <div class="game-content">
      <p id="wordHint"></p>
      <p id="incorrectLetters"></p>
      <div id="guessArea"></div>
      <input type="text" id="letterInput" maxlength="1" placeholder="Wpisz literę">
      <button id="resetButton">Nowe słowo</button>
    </div>
  </section>
```

```
<!-- Podłączenie pliku JavaScript -->
<script src="script.js"></script>
</body>
</html>
```

## Zadanie Nawigacja:

`<header>` / `<nav>`: przykładowe menu nawigacyjne.

Sekcja parallax: `id="parallax-section"` + klasa `.parallax` (do stylowania).

Sekcja quizu: `id="quiz-section"`.

Sekcja gry: `id="game-section"`.

`<script src="script.js"></script>` na końcu: dołączamy kod JS.

```
<nav>
  <ul>
    <li><a href="#parallax-section">Paralaksa</a></li>
    <li><a href="#quiz-section">Quiz</a></li>
    <li><a href="#game-section">Gra</a></li>
  </ul>
</nav>
```

## Krótki opis:

- Kod odpowiada za **menu nawigacyjne** widoczne na stronie.
- Każdy link (`<a>`) w liście (`<ul>`) odwołuje się do innej sekcji za pomocą `href="#nazwa-sekcji"`.
- Kliknięcie w link powoduje przewinięcie strony do elementu z odpowiadającym `id` (np. `id="parallax-section"`).
- Upewnij się, że sekcje w HTML mają identyczne `id` jak w `href`.

## Dlaczego `href` ma taką nazwę i jak przekierowuje na konkretną część strony?

W nawigacji używamy linków typu `href="#nazwa-sekcji"`. Gdy klikniesz taki link, przeglądarka szuka elementu z atrybutem `id="nazwa-sekcji"` i przewija stronę właśnie do tego miejsca. Dlatego musisz nadać sekcji identyfikator (`id="quiz-section"`, `id="parallax-section"` itp.), a w linku (`<a>`) umieścić nazwę odpowiadającą temu `id` (np. `href="#quiz-section"`). Dzięki temu klikanie w menu prowadzi dokładnie do pasującej sekcji na stronie.

## Zadanie: Dodanie dwóch kolejnych sekcji paralaksowych

### Opis zadania

W tym ćwiczeniu uczniowie mają za zadanie wyszukać w internecie własne obrazy i wykorzystać je jako tło w **dwóch nowych sekcjach** paralaksowych.

Dzięki temu strona zyska dodatkowe „warstwy” przewijania z efektem `background-attachment: fixed;`.

---

## Kod do uzupełnienia

HTML (fragment do dodania w `index.html`)

```
<!-- Druga sekcja paralaksy -->
<section id="parallax-section2" class="parallax">
  <div class="parallax-content">
    <h2>Kolejna Sekcja Paralaksy</h2>
    <p>Wstaw tutaj własny tekst.</p>
  </div>
</section>

<!-- Trzecia sekcja paralaksy -->
<section id="parallax-section3" class="parallax">
  <div class="parallax-content">
    <h2>Jeszcze Jedna Sekcja</h2>
    <p>Wstaw tutaj własny tekst.</p>
  </div>
</section>
```

## Stylowanie i efekt paralaksy (15 minut)

Kod do pliku `style.css`:

```
/* Reset stylów domyślnych (opcjonalnie) */
* {
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}

/* Podstawowe stylowanie body */
body {
  font-family: Arial, sans-serif;
  line-height: 1.6;
  color: #333;
}

/* Stylizacja nawigacji */
header {
  background: #fff;
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
header nav ul {
```

```
    display: flex;
    list-style: none;
    padding: 10px;
}
header nav li {
    margin-right: 20px;
}
header nav a {
    text-decoration: none;
    color: #333;
    font-weight: bold;
}

/* Sekcja paralaksy */
.parallax {
    /* Tło, które będzie się poruszało wolniej niż okno przeglądarki */

    background: url('https://picsum.photos/1920/1080') no-repeat center center; /*
    background: url('./images/wyspa.jpg') no-repeat center center;

    background-size: cover;
    min-height: 100vh; /* wysokość okna */
    background-attachment: fixed; /* klucz do efektu paralaksy */
    background-position: center;
    background-repeat: no-repeat;
    display: flex;
    align-items: center;
    justify-content: center;
}
.parallax-content {
    background: rgba(255,255,255,0.7);
    padding: 20px;
    border-radius: 5px;
    text-align: center;
}

/* Sekcja quizu */
.quiz-container {
    padding: 40px;
    background: #f3f3f3;
}
.quiz-container h2 {
    margin-bottom: 20px;
}
#quiz {
    margin-bottom: 20px;
}

/* Sekcja gry */
.game-container {
    padding: 40px;
    background: #fff;
}
```

```
.game-container h2 {  
  margin-bottom: 20px;  
}  
.game-content {  
  margin-top: 10px;  
}  
#letterInput {  
  width: 40px;  
  text-align: center;  
  margin-right: 10px;  
}
```

### Omówienie:

- `background-attachment: fixed;` odpowiada za efekt paralaksy (tło „zatrzymuje się” przy przewijaniu).
- Dopasuj zdjęcie (np. z [picsum.photos](https://picsum.photos)) do własnych potrzeb.
- Reszta to ogólne style: nawigacja, quiz i gra.

---

## Logika quizu (JavaScript) (20–25 minut)

W pliku `script.js` wklej najpierw logikę quizu:

```
// ----- QUIZ LOGIKA ----- //
```

```
// Przykładowe pytania i odpowiedzi  
const questions = [  
  {  
    question: "Jaki język służy do stylowania strony?",  
    answers: ["HTML", "CSS", "JavaScript", "Python"],  
    correctAnswer: 1 // indeks "CSS"  
  },  
  {  
    question: "Który znacznik używamy, by osadzić skrypt JavaScript w HTML?",  
    answers: ["<js>", "<script>", "<javascript>", "<code>"],  
    correctAnswer: 1  
  },  
  {  
    question: "Jaka jest poprawna metoda tworzenia efektu paralaksy?",  
    answers: [  
      "background-attachment: move;",  
      "background-attachment: fixed;",  
      "background-fix: parallax;",  
      "background-parallax: true;"  
    ],  
    correctAnswer: 1  
  }  
];
```

```
let currentQuestionIndex = 0;

// Wyświetlanie pytania
function showQuestion() {
  const quizContainer = document.getElementById("quiz");
  quizContainer.innerHTML = "";

  const questionObj = questions[currentQuestionIndex];

  // Tworzymy element pytania
  const questionElement = document.createElement("h3");
  questionElement.textContent = questionObj.question;
  quizContainer.appendChild(questionElement);

  // Tworzenie przycisków z odpowiedziami
  questionObj.answers.forEach((answer, index) => {
    const answerButton = document.createElement("button");
    answerButton.textContent = answer;
    answerButton.onclick = () => checkAnswer(index);
    quizContainer.appendChild(answerButton);
  });
}

// Sprawdzanie poprawności odpowiedzi
function checkAnswer(selectedAnswer) {
  const questionObj = questions[currentQuestionIndex];
  if (selectedAnswer === questionObj.correctAnswer) {
    alert("Dobra odpowiedź!");
  } else {
    alert("Niestety, to niepoprawna odpowiedź.");
  }
}

// Obsługa przycisku 'Następne pytanie'
const nextButton = document.getElementById("nextButton");
nextButton.addEventListener("click", () => {
  currentQuestionIndex++;
  if (currentQuestionIndex >= questions.length) {
    currentQuestionIndex = 0;
    alert("Koniec pytań! Zaczynamy od nowa.");
  }
  showQuestion();
});

// Wyświetlamy pierwsze pytanie na start
showQuestion();

// ----- KONIEC QUIZU ----- //
```

Omówienie:

- `questions`: tablica zawiera pytania, odpowiedzi i indeks poprawnej odpowiedzi.
  - `showQuestion()`: generuje zawartość quizu w elemencie `#quiz`.
  - `checkAnswer()`: porównuje wybór użytkownika z poprawną odpowiedzią i wyświetla stosowny komunikat.
  - `nextButton`: klika się w niego, by przejść do kolejnego pytania.
- 

## Logika gry w zgadywanie słów (JavaScript) (20–25 minut)

Na końcu pliku `script.js` wklej następujący kod:

```
// ----- GRA W ZGADYWANIE SŁÓW ----- //
```

```
// Lista przykładowych słów z podpowiedziami
const wordList = [
  { word: "javascript", hint: "Język skryptowy do tworzenia interakcji na stronach" },
  { word: "parallax", hint: "Efekt tła przesuwającego się wolniej niż pierwszy plan" },
  { word: "quiz", hint: "Forma interaktywnego testu" }
];

let selectedWord = "";
let selectedHint = "";
let incorrectLetters = [];
let correctLetters = [];

const wordHint = document.getElementById("wordHint");
const incorrectLettersElement = document.getElementById("incorrectLetters");
const guessArea = document.getElementById("guessArea");
const letterInput = document.getElementById("letterInput");
const resetButton = document.getElementById("resetButton");

// Funkcja do losowania słowa
function randomWord() {
  const randomItem = wordList[Math.floor(Math.random() * wordList.length)];
  selectedWord = randomItem.word;
  selectedHint = randomItem.hint;
}

// Wyświetlanie widocznego stanu zgadywanego słowa
function displayWord() {
  guessArea.innerHTML = "";
  for (let i = 0; i < selectedWord.length; i++) {
    const span = document.createElement("span");
    // Jeśli litera została odgadnięta, pokazujemy ją, w przeciwnym razie '_'
    span.textContent = correctLetters.includes(selectedWord[i]) ? selectedWord[i] : "_";
    guessArea.appendChild(span);
  }
}
```



```
wordHint.textContent = "Podpowiedź: " + selectedHint;
incorrectLettersElement.textContent = "Niepoprawne litery: " +
incorrectLetters.join(", ");
}

// Obsługa wprowadzania liter
letterInput.addEventListener("input", () => {
  let letter = letterInput.value.toLowerCase();
  letterInput.value = ""; // czyścimy pole po wpisaniu litery

  // Sprawdzamy, czy wpisano tylko pojedynczą literę
  if (!letter.match(/^[a-zA-Z]$/)) {
    alert("Proszę wpisać tylko jedną literę (a-z)!");
    return;
  }

  if (selectedWord.includes(letter)) {
    if (!correctLetters.includes(letter)) {
      correctLetters.push(letter);
    }
  } else {
    if (!incorrectLetters.includes(letter)) {
      incorrectLetters.push(letter);
    }
  }

  displayWord();
  checkGameStatus();
});

// Sprawdzanie stanu gry
function checkGameStatus() {
  // Sprawdzamy wygraną
  let currentGuess = "";
  for (let i = 0; i < selectedWord.length; i++) {
    currentGuess += correctLetters.includes(selectedWord[i]) ? selectedWord[i] :
    "_";
  }
  if (currentGuess === selectedWord) {
    alert("Gratulacje! Odgadłeś słowo: " + selectedWord);
    startGame();
  }

  // Przykładowy warunek porażki
  if (incorrectLetters.length >= 6) {
    alert("Przekroczono limit błędów! Słowo to: " + selectedWord);
    startGame();
  }
}

// Rozpoczęcie nowej gry
function startGame() {
  incorrectLetters = [];
  correctLetters = [];
```

```
    randomWord();
    displayWord();
}

// Obsługa przycisku "Nowe słowo"
resetButton.addEventListener("click", startGame);

// Inicjalizacja gry na starcie
startGame();
```

## Omówienie:

- `wordList`: tablica słów i podpowiedzi.
  - `randomWord()`: losuje nowe słowo i jego podpowiedź.
  - `displayWord()`: aktualizuje interfejs – pokazuje odgadnięte litery lub \_.
  - `letterInput` (zdarzenie input): użytkownik wprowadza pojedynczą literę, a skrypt weryfikuje jej wystąpienie w słowie.
  - `checkGameStatus()`: sprawdza, czy użytkownik odgadł słowo lub przekroczył liczbę dozwolonych błędów (6).
  - `startGame()`: resetuje stan i uruchamia nową rozgrywkę.
- 

## Podsumowanie, testowanie i modyfikacje (10 minut)

- Uruchom w przeglądarce plik index.html.
- Sprawdź:
  - Efekt paralaksy (przewijanie strony).
  - Quiz: czy pytania i odpowiedzi działają prawidłowo.
  - Gra w zgadywanie słów: czy słowo faktycznie można odgadnąć.
- Zachęć uczestników do modyfikacji:
  - Zmiana pytań w quizie.
  - Dodanie większej liczby słów w grze.
  - Modyfikacja liczby dozwolonych błędów.
  - Upiększanie strony w CSS (zmiana kolorów, czcionek, animacji).
  - Zamiast okienek `alert()`, można wyświetlać powiadomienia w dedykowanych sekcjach HTML.