
JavaScript



Kamil Cecherz
kamil.cecherz@gmail.com

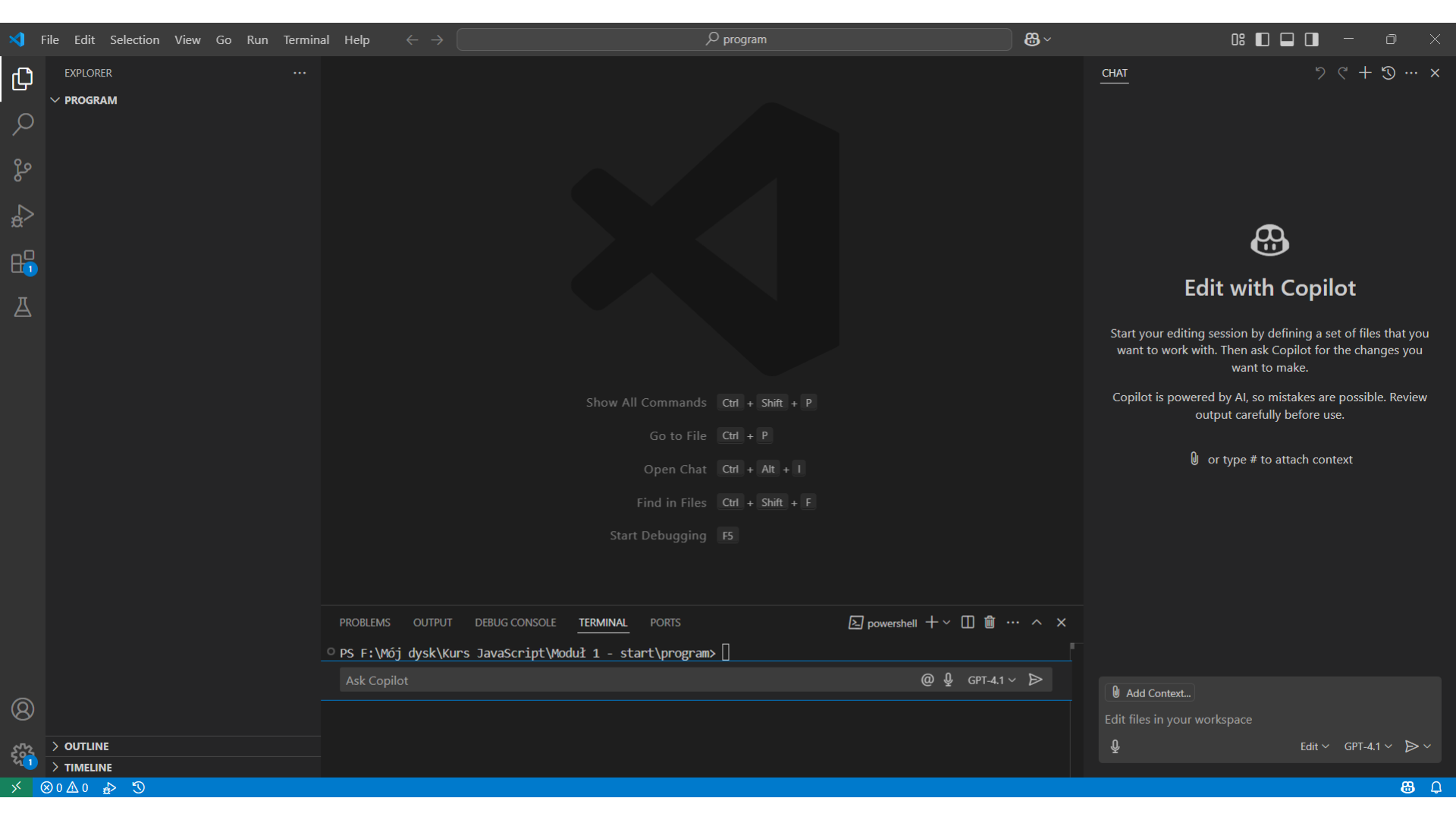
Jak zacząć tworzyć pierwsze programy?

Czego potrzebujesz?

- Najnowsza przeglądarka WWW np. **Chrome**
 - **JavaScript** jest językiem skryptowym rozszerza możliwości stron internetowych napisanych przy pomocy **HTML** i **CSS**.

Kod **JavaScript** jest wykonywany **bezpośrednio przez interpreter** (np. w przeglądarce internetowej), linia po linii, bez wcześniejszego **kompilowania do kodu maszynowego**.

- Środowisko programistyczne np. **VS Code**
 - Strony internetowe i instrukcje programów można pisać w notatniku ale jest to wygodne. Kolorowanie składni oraz wsparcie **Copilota** to nieocenioną pomoc.
-

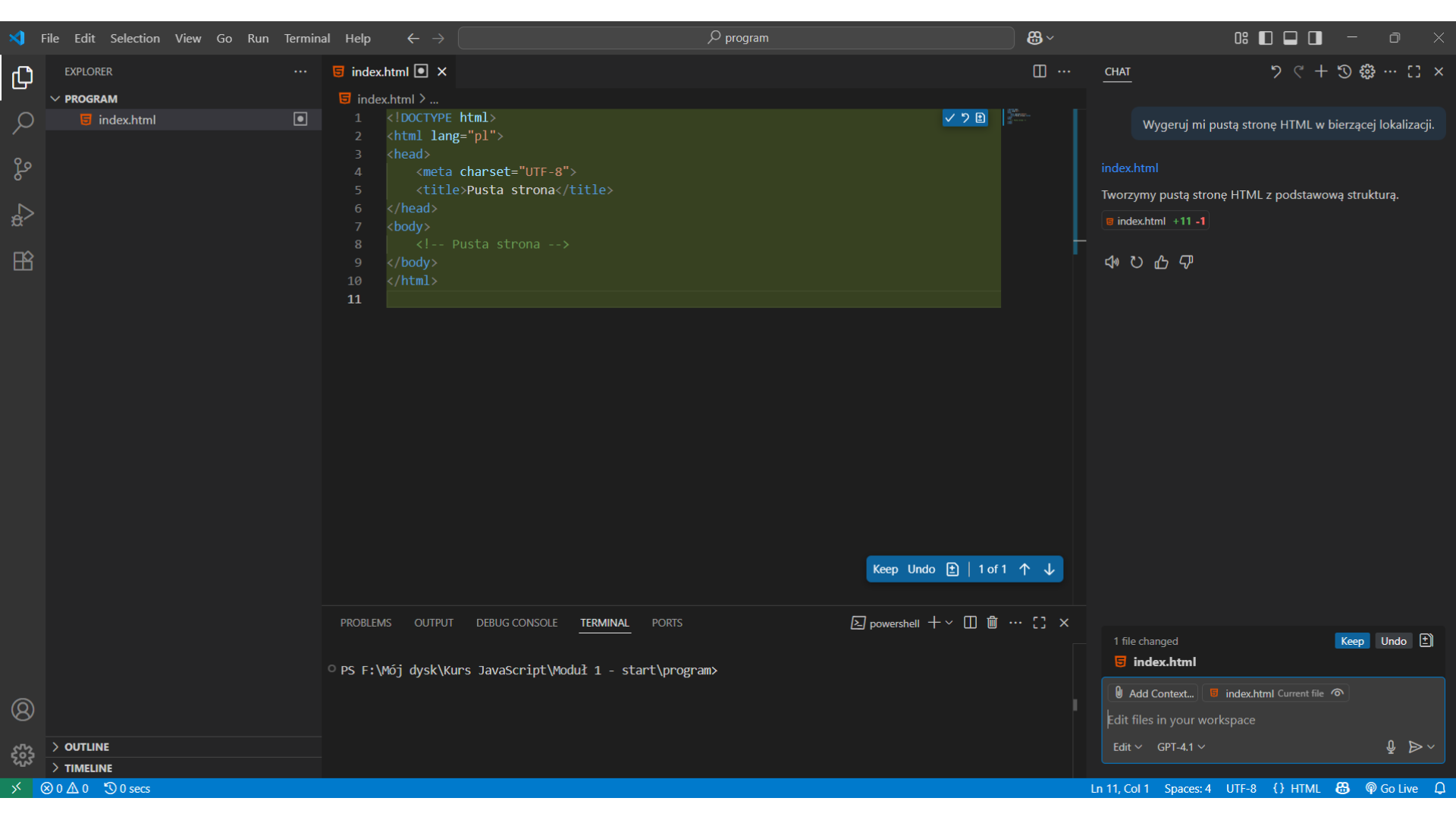


Ważne skróty klawiszowe

- **Ctrl + Shift + I** - otwiera nowe okno czata
 - **Ctrl + `** - otwiera terminal
 - **Ctrl + Shift + P** - otwiera akcje
-

Programowanie a AI

- Dzięki dostępowi do nowoczesnych narzędzi aplikacje webowe można napisać korzystając wyłącznie ze sztucznej inteligencji ale szybkie rozwiązania mają swoje wady. Kod pisany, generowany bez zrozumienia będzie trudny do naprawy.
 - Wszystko co umieszczasz w programie powinieneś znać, każdą linijkę kodu wytłumaczyć.
 - AI robi błędy trzeba być uważnym, śledzić wszystko, zadawać poprawne pytania.
-



Ćwiczenie 1

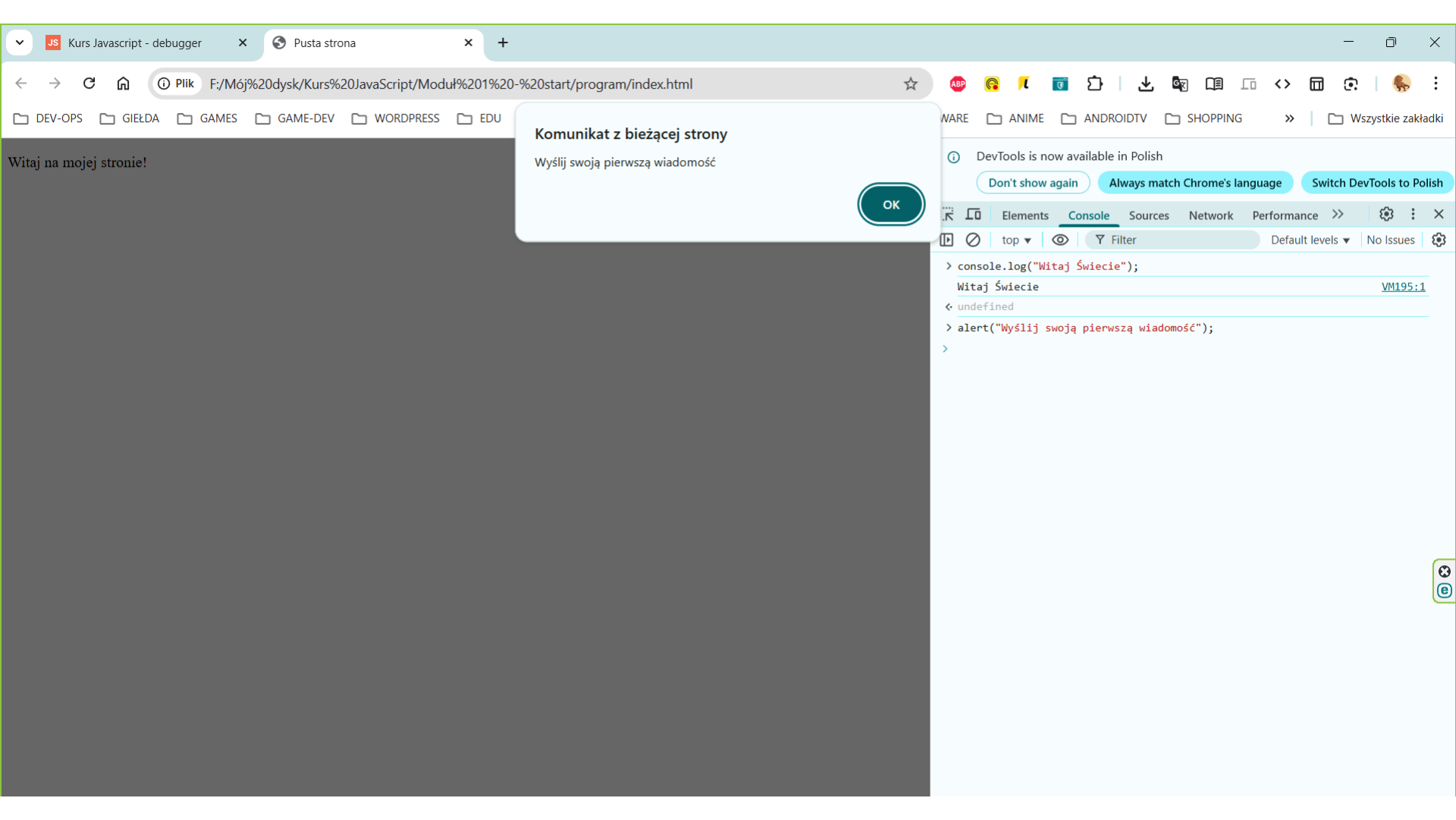
1. Utwórz pusty folder na **Pulpicie**, nadaj mu nazwę taką jaką chcesz określić swój projekt.
 2. Otwórz **VS Code** i odwołaj się do przestrzeni roboczej, którą będzie otworzony folder.
 3. Otwórz okno **Copilota i Terminal**
 4. Poproś **Copilota** o wygenerowanie pustej strony **HTML** w wybranej przestrzeni roboczej.
 5. Nadaj stronie tytuł.
 6. Wstaw na stronę **nagłówek h1** oraz **akapit p**, gdzie umieścisz tytuł oraz fragment ulubionego wiersza.
 7. Wyświetl stronę w przeglądarce.
-

Cechy języka JavaScript

- Dynamiczny, słaby system typów:
 - Nie trzeba deklarować **typu zmiennej** przed jej użyciem.
 - JavaScript może automatycznie konwertować typy danych w wyrażeniach, co może prowadzić do nieoczekiwanych wyników.
 - Nie określa się typu zmiennej przy deklaracji:
 - `let x = 5; x = "tekst";`
 - Działa bez kompilacji:
 - Kod JavaScript jest **interpretowany bezpośrednio** przez przeglądarkę lub silnik wykonawczy.
 - Środowisko działania (runtime) – głównie przeglądarka i [Node.js](#)
 - W przeglądarce odpowiada za **interakcję ze stroną (DOM), reagowanie na zdarzenia, animacje, walidację formularzy**.
 - W środowisku **Node.js** staje się językiem ogólnego przeznaczenia do tworzenia serwerów i aplikacji backendowych.
-

Konsola

- Aby zacząć zabawę z programowaniem w JavaScript wystarczy otworzyć przeglądarkę na dowolnej stronie WWW, choć dobrze by zaczynać od pustego szablonu HTML, i uruchomić narzędzia deweloperskie skrót klawiszowy: **Ctrl + Shift + I**
 - Konieczne jest przejście do zakładki **Console**, gdzie możemy w czasie rzeczywistym wykonywać bezpośrednio podane instrukcje, bez konieczności tworzenia osobnych plików.
 - Sposób ten jest dobry gdy chcemy przetestować jakąś funkcjonalność, sprawdzić działanie strony. Po przeładowaniu strony napisany skrypt znika z pamięci komputera.
-

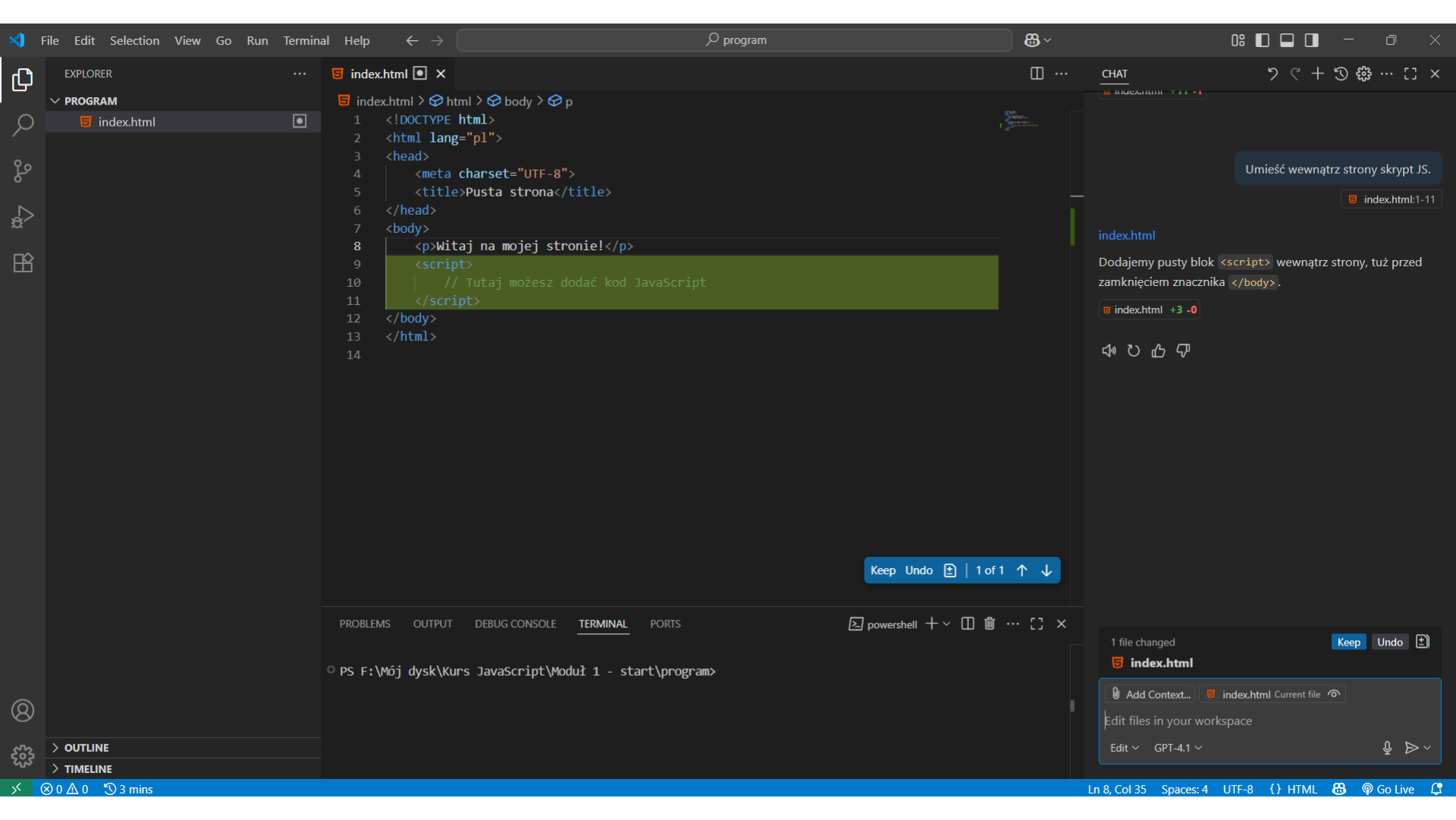


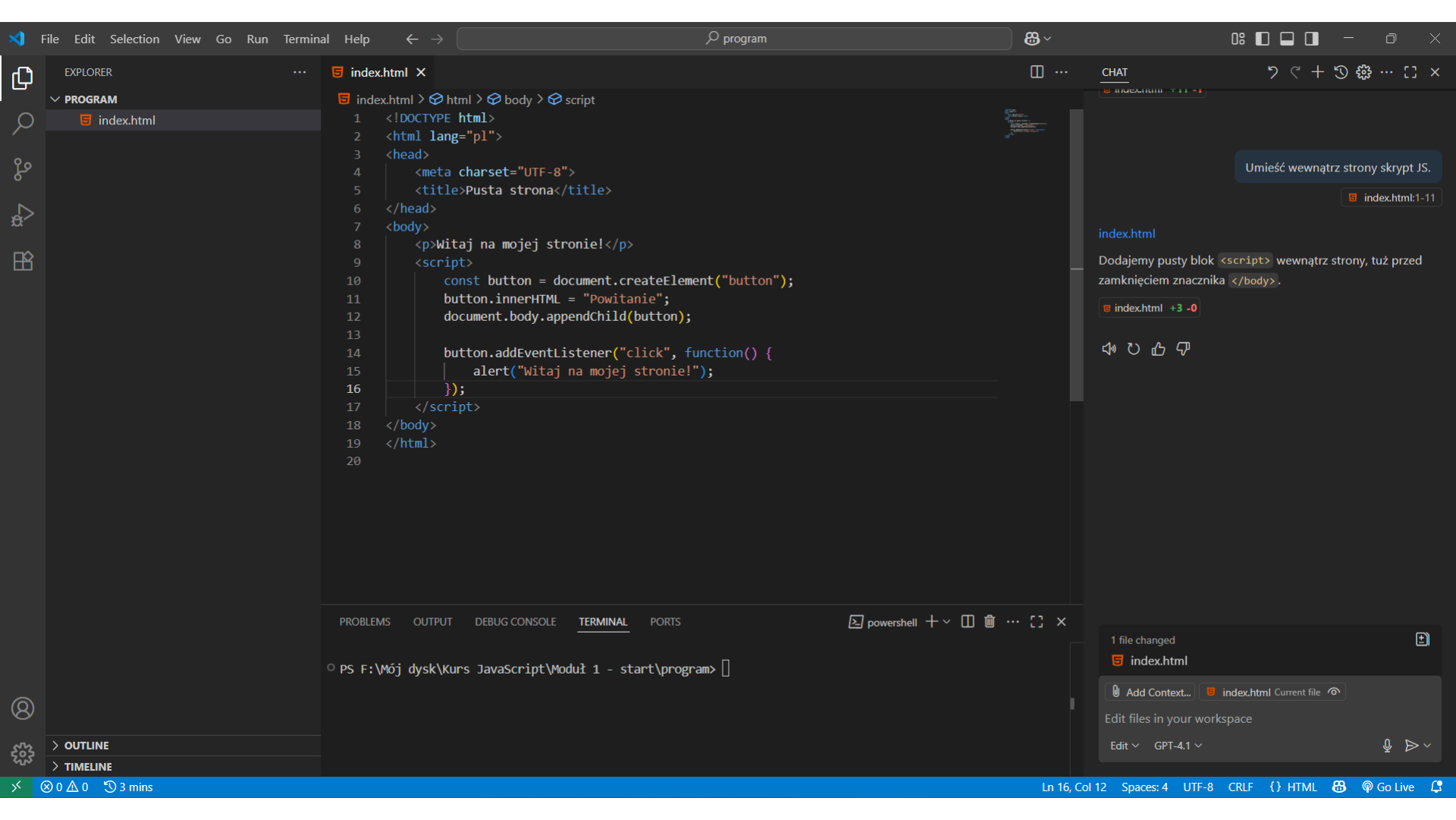
Ćwiczenie 2

1. Bazujemy na projekcie z ćwiczenia pierwszego.
 2. Otwórz narzędzia deweloperskie, zmień kolor **nagłówka h1** na czerwony oraz **akapitu p** na zielony posługując się CSS bezpośrednio w przeglądarce.
 3. Przeładuj stronę, czy widzisz by zmiany się zachowały? Czy zgodzisz się z tym, że elementy HTML na stronie nie posiadają żadnych stylów?
 4. Przejdź do zakładki **console** i wywołaj w niej instrukcje **console.log(...)** oraz **alert(...)** zdecyduj jaki tekst chcesz wyświetlić. Co się wydarzy jeśli zdecydujesz podać jako argumenty metod liczby? Do czego w praktyce możesz wykorzystać przedstawione metody?
-

Jak połączyć skrypt z dokumentem HTML?

1. Na dole wewnątrz sekcji **body** dokumentu HTML, pomiędzy znacznikami **script**.
 2. W osobnym pliku, do którego odwołujemy się w dokumencie HTML na dole w sekcji **body** za pomocą instrukcji: `<script src="js/main.js"></script>`
-





EXPLORER

PROGRAM

index.html

index.html X

index.html > html > body > script

```
1 <!DOCTYPE html>
2 <html lang="pl">
3 <head>
4   <meta charset="UTF-8">
5   <title>Pusta strona</title>
6 </head>
7 <body>
8   <p>Witaj na mojej stronie!</p>
9   <script>
10     const button = document.createElement("button");
11     button.innerHTML = "Powitanie";
12     document.body.appendChild(button);
13
14     button.addEventListener("click", function() {
15       alert("Witaj na mojej stronie!");
16     });
17   </script>
18 </body>
19 </html>
20
```

CHAT

Umieść wewnątrz strony skrypt JS.

index.html:1-11

index.html

Dodajemy pusty blok `<script>` wewnątrz strony, tuż przed zamknięciem znacznika `</body>`.

index.html +3 -0

🔊 🔄 👍 🗨

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

powershell + - 🗑 ... 🗨 X

PS F:\Mój dysk\Kurs JavaScript\Moduł 1 - start\program>

> OUTLINE

> TIMELINE

1 file changed

index.html

Add Context... index.html Current file 🔗

Edit files in your workspace

Edit GPT-4.1

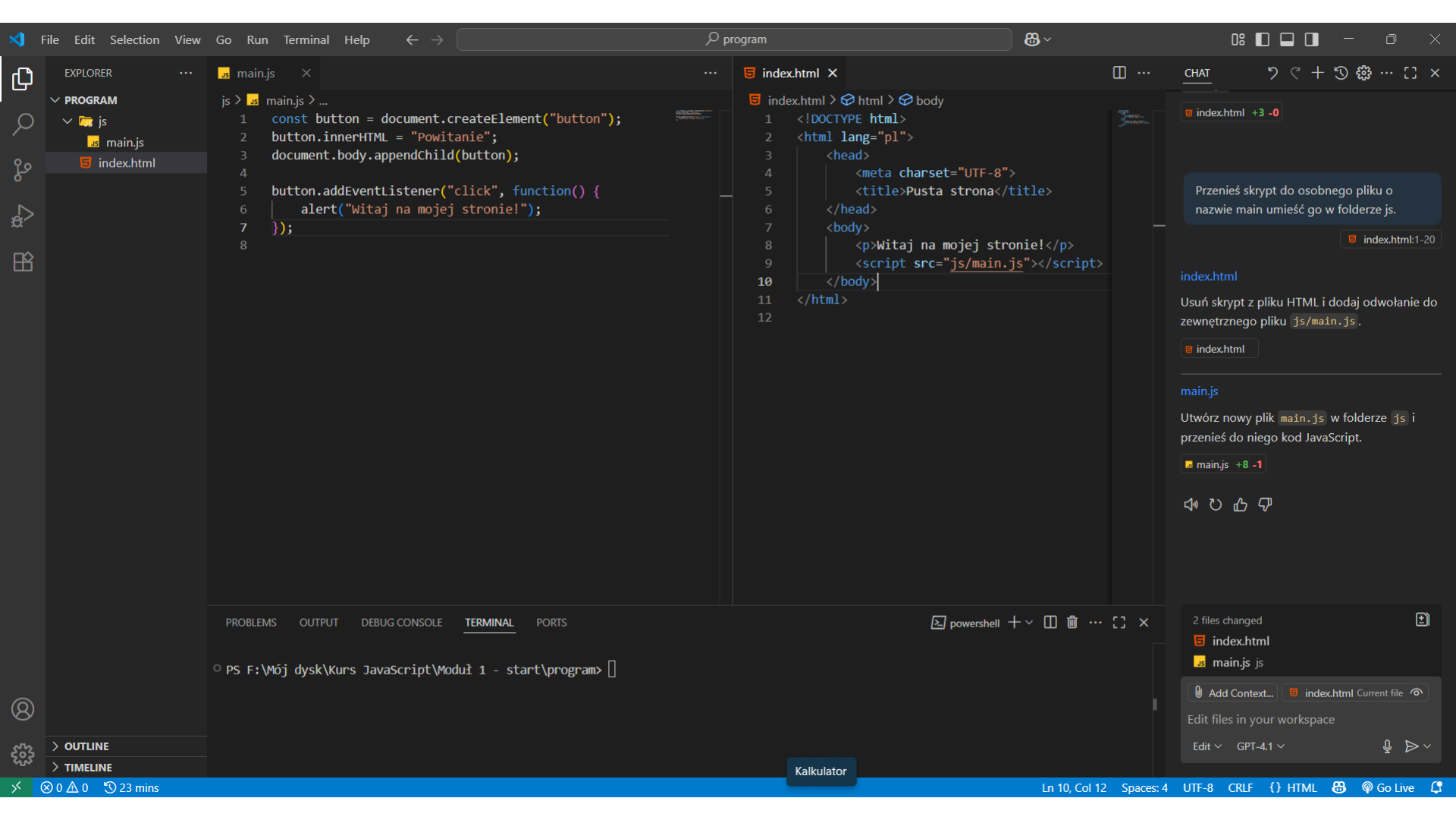
🔊 🔄 👍 🗨

Ćwiczenie 3

1. Bazujemy na projekcie z ćwiczenia pierwszego.
 2. Poproś **Copilota** o osadzenie skryptu JS wewnątrz znacznika `body`.
 3. Przepisz kod ze **slajdu 14** zmieniając wyświetlane napisy oraz nazwę stałej `button`. Nie przejmuj się jeśli na tym etapie nie rozumiesz kodu. Przykład ma prezentować możliwości języka. Możesz dodać kilka własnych instrukcji metody **`alert()`** i jak i **`console.log()`**.
 4. Uruchom stronę w przeglądarce.
 5. Otwórz narzędzia deweloperskie i sprawdź zakładkę **`sources`**. Jakie przydatne informacje możesz tam znaleźć?
-

Ćwiczenie 4

1. Bazujemy na projekcie z ćwiczenia trzeciego.
 2. Poproś **Copilota** o informacje, dowiedz się jak osadzić kod JavaScript korzystając z osobnego pliku.
 3. Stwórz folder o nazwie **js** gdzie przeniesiesz plik z kodem obecnie znajdującym się pomiędzy znacznikami **script** w sekcji **body**.
 4. Otwórz narzędzia deweloperskie i sprawdź zakładkę **sources**. Jakie przydatne informacje możesz tam znaleźć? Co się zmieniło?
-



Uwagi, do zapamiętania:

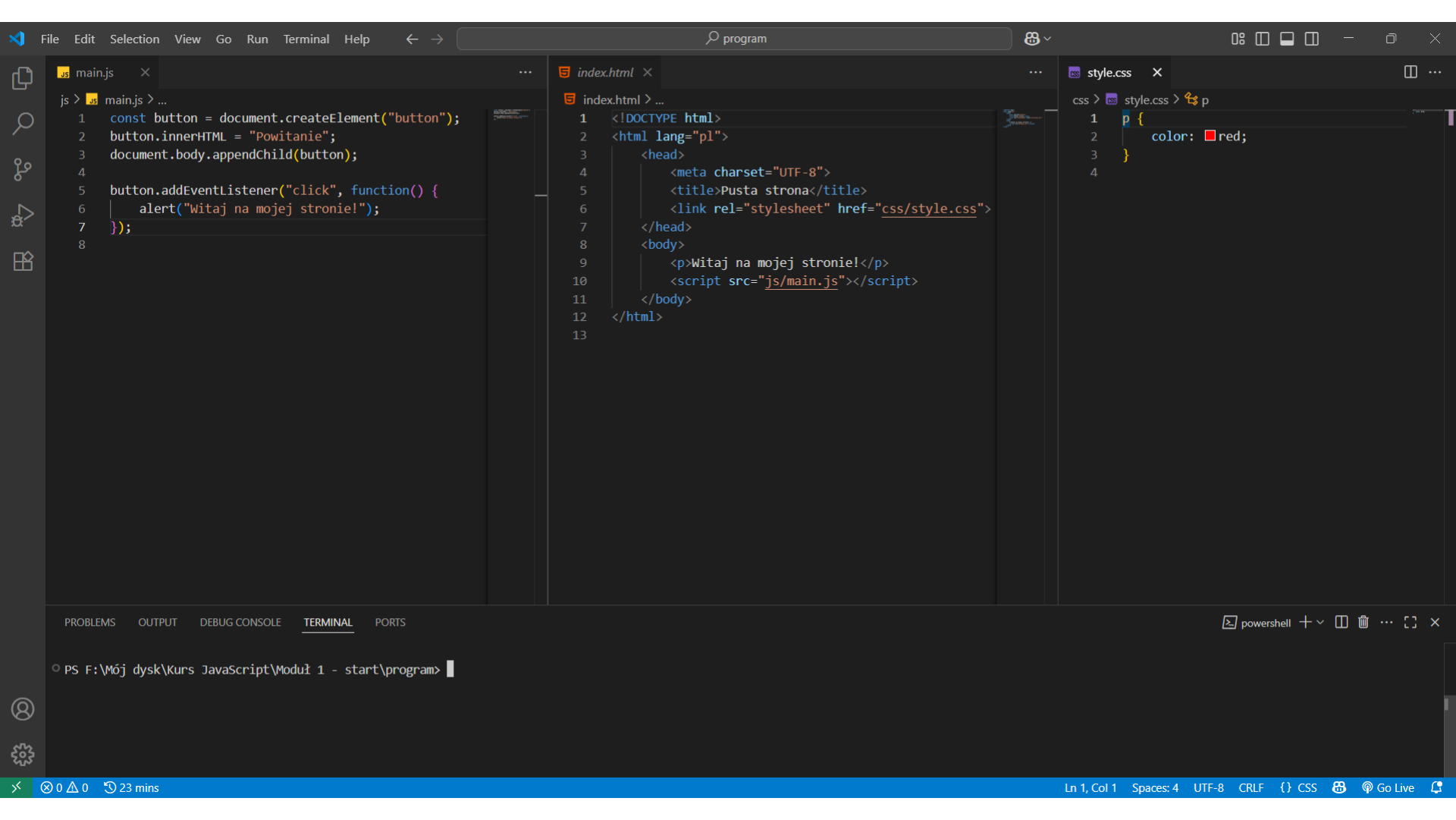
- Kod JavaScript wykonuje się po załadowaniu całego drzewa dokumentu **DOM** stąd instrukcje skryptu wykonywane są na końcu.
- Istnieje trzeci sposób osadzenia skryptu JS na stronie w sekcji **head** co wydaje się bardziej czytelne, wtedy jednak musimy skorzystać z dodatkowych instrukcji:
 - defer - opóźnia wykonanie do końca ładowania HTML-a
 - async - ładuje niezależnie i wykonuje, gdy tylko się załaduje (dobry dla skryptów niezależnych np. analytics).

```
<head>  
<title>Strona</title>  
<script src="main.js" defer></script>  
</head>
```

- Zaawansowaną techniką jest dodawanie kodu skryptu w locie, co wykracza poza ramy podstawowego kursu.
-

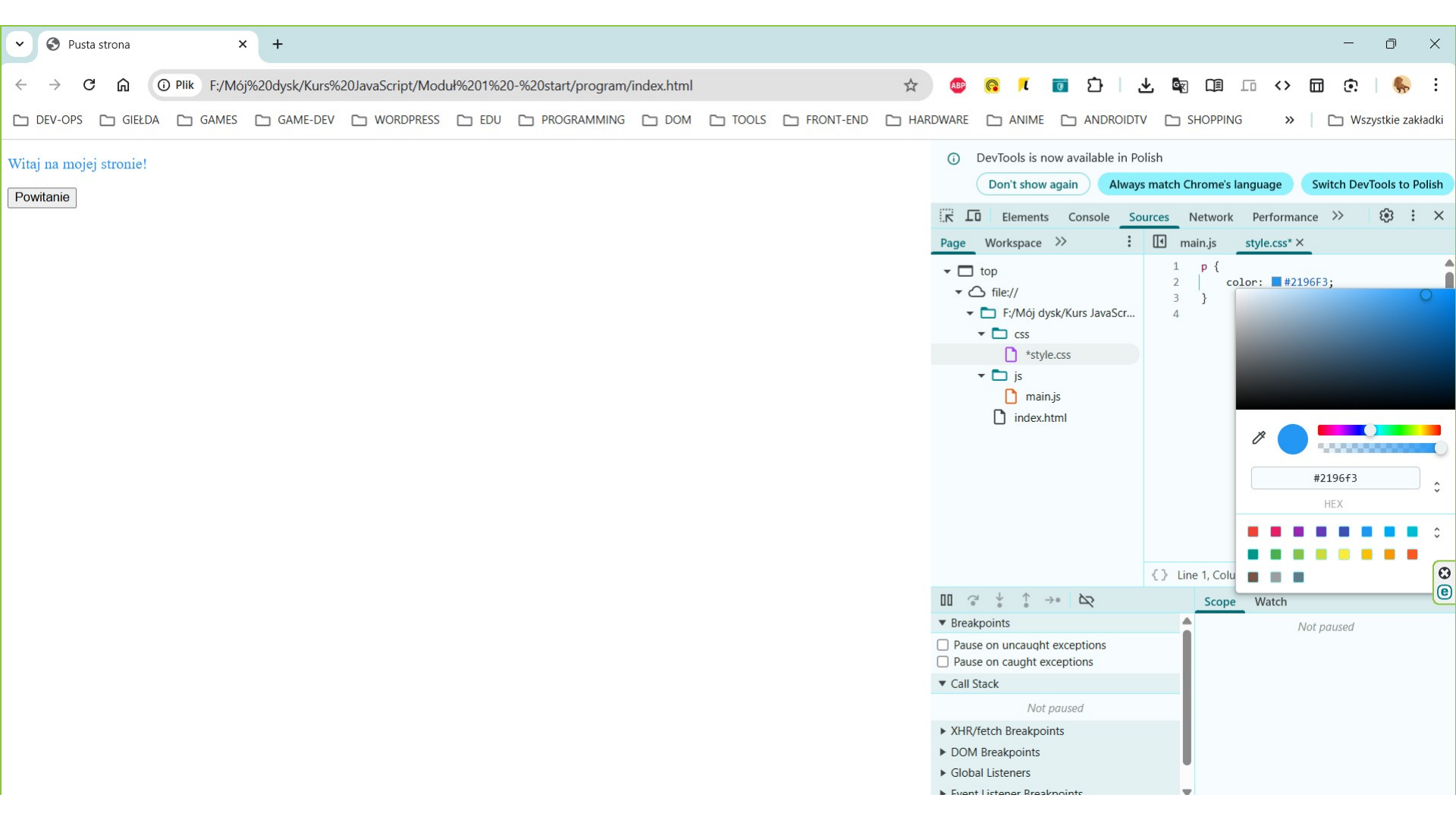
Ćwiczenie 5

1. Bazujemy na projekcie z ćwiczenia czwartego.
 2. Poproś **Copilota** o informacje, dowiedz się jak osadzić kod **CSS** korzystając z osobnego pliku.
 3. Stwórz folder o nazwie **css** gdzie utworzysz plik kaskadowych arkuszy stylu.
 4. Nadaj znacznikom **nagłówek h1** oraz **akapitu p** unikatowe identyfikatory lub klasy jak wolisz.
 5. Korzystając z zewnętrznego pliku **css** nadaj styl elementom na stronie.
 6. Korzystając z narzędzi deweloperskich sprawdź pliki strony, przeanalizuj zawartość **dokumentu HTML**.
-



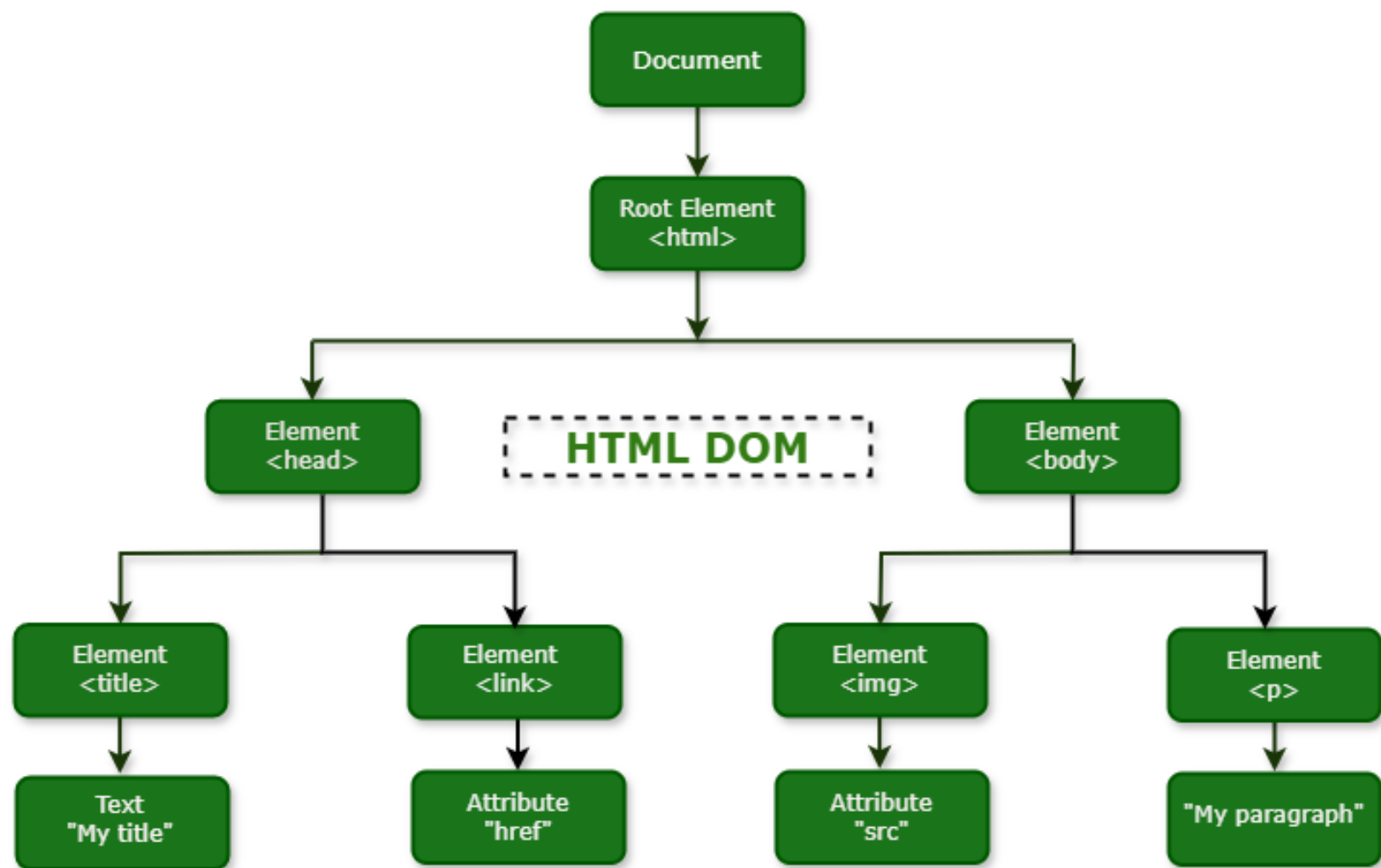
Potęga narzędzi deweloperskich

- Narzędzia dostępne z poziomu przeglądarki pozwalają na manipulację wyglądem strony w czasie rzeczywistym, podgląd jak i zmianę treści arkuszy stylów, skryptów, dokumentu html.
 - Możemy wykonać testy bez uszkodzania plików projektu, zmiany zapisywane są tylko w pamięci tymczasowej.
 - Przejrzysty interfejs zapewnia wygodę.
-



DOM - Document Object Model

- Strona WWW, konkretnie **dokument HTML** ma strukturę drzewiastą.
 - Każdy **element, znacznik HTML** znajduje na pewnym poziomie w hierarchii, można to porównać do drzewa genealogicznego. Każdy bowiem ma swojego przodka lub dzieci.
 - **JavaScript** umie odczytać taką strukturę danych (sprawdź pojęcie grafu w informatyce). Operowanie na czystym tekście nie daje możliwości manipulowania danymi.
 - Za pomocą skryptów można odczytywać, kasować jak i tworzyć nowe **węzły dokumentu HTML**.
 - Kluczowe jest zrozumienie pojęć takich jak dziedziczenie, nauczenie się czym są **klasy, pseudoklasy czy identyfikatory**.
-



KONIEC