

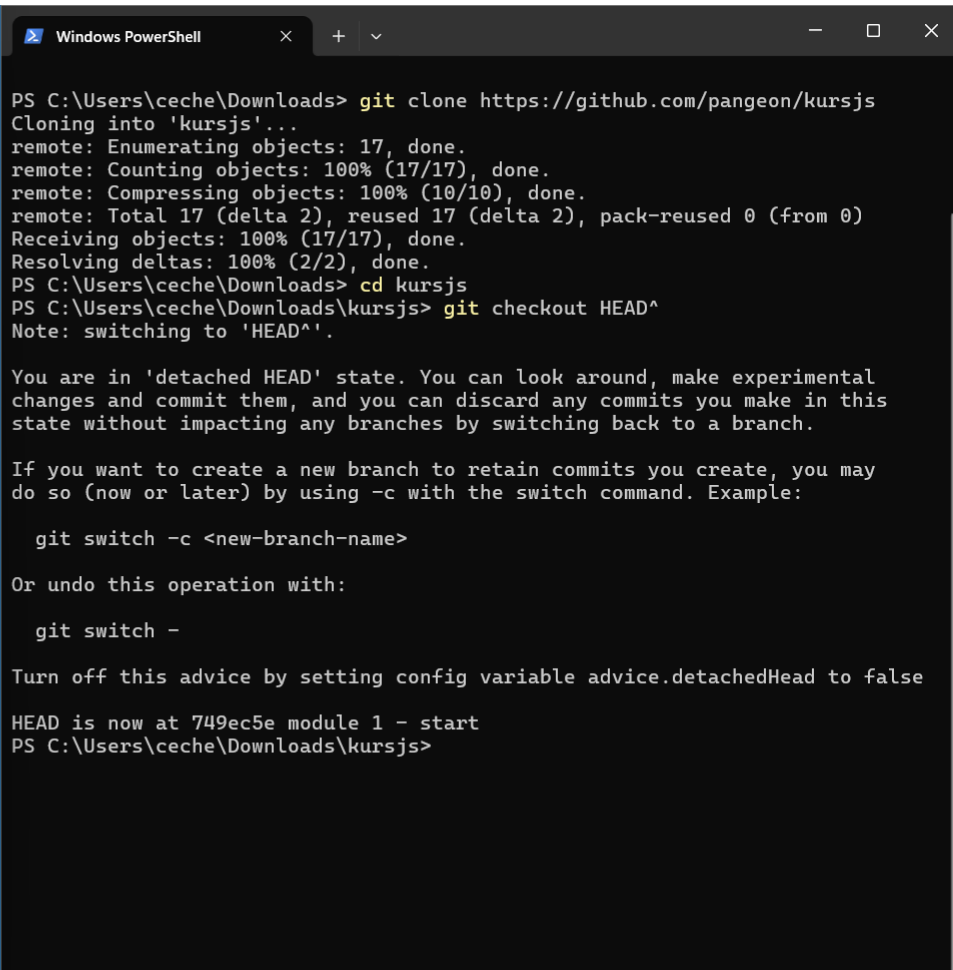
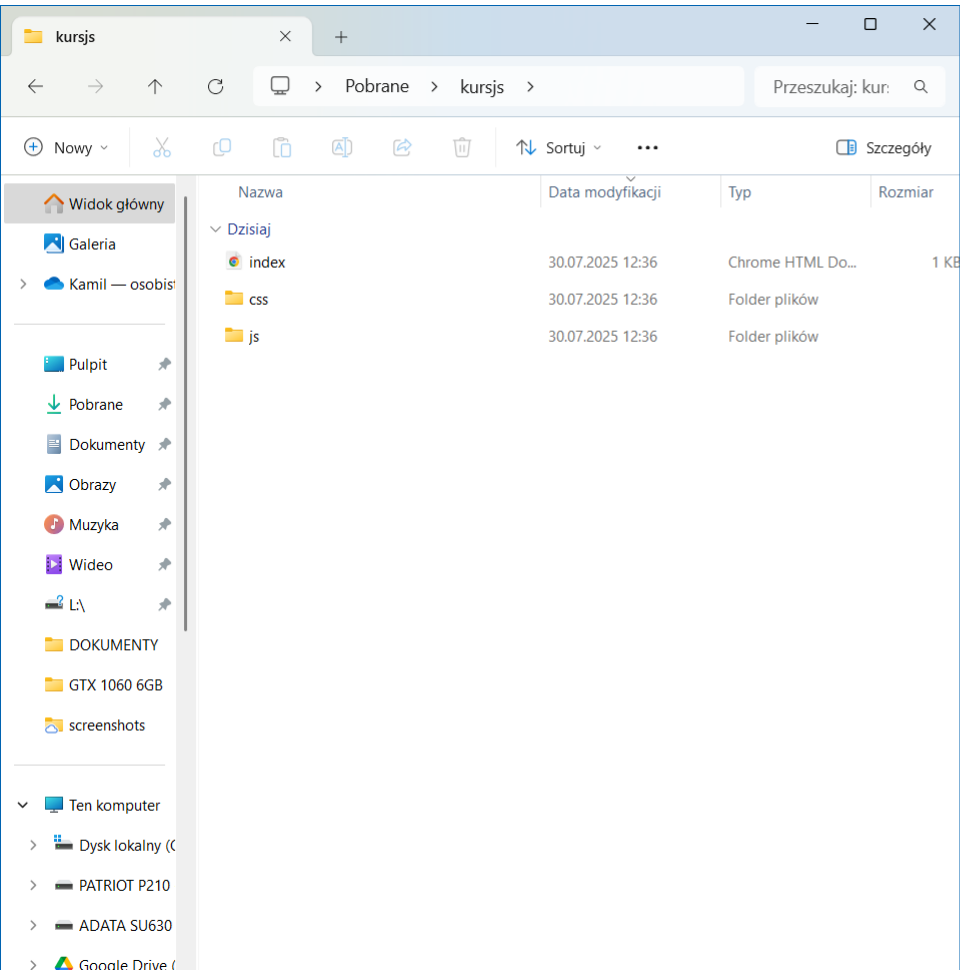
JavaScript

Techniki pracy

Kamil Cecherz

Instrukcja 1

- Zainstaluj na swoim OS, system kontroli wersji GIT: <https://git-scm.com/downloads/win>
- W dowolnej lokalizacji najlepiej bardzo dobrze Ci znanej, otwórz **Terminal**.
- Sklonuj repozytorium **kursjs** przy pomocy komendy:
git clone https://github.com/pangeon/kursjs
- Wejdź do katalogu kurs: polecenie **cd kursjs** lub skorzystaj z menadżera plików Windows.
- W **Terminalu** wpisz polecenie: **git checkout HEAD^**
- Otwórz projekt przy pomocy **VS Code**.
- Wszystko co teraz będziesz wykonywać rób z poziomu **IDE**.





Co to jest Git?

— — —

Git to program, który **zapamiętuje historię zmian w plikach**. Używa się go najczęściej do pracy nad kodem, ale można go używać z każdym typem plików.

To **magiczny notes** dla programisty, który:

- pokazuje **co się zmieniło**,
- pozwala **cofnąć się do wcześniejszej wersji**,
- umożliwia **pracę kilku osób na raz** nad tym samym projektem bez wchodzenia sobie w drogę.

Filter files...



css

style.css

index.html

js

main.js

3 files changed +22 -0 lines changed

Search within code



css/style.css

+3

@@ -0,0 +1,3 @@

```
1 + p {
2 +   color: red;
3 + }
```

index.html

+12

@@ -0,0 +1,12 @@

```
1 + <!DOCTYPE html>
2 + <html lang="pl">
3 +   <head>
4 +     <meta charset="UTF-8">
5 +     <title>Pusta strona</title>
6 +     <link rel="stylesheet" href="css/style.css">
7 +   </head>
8 +   <body>
9 +     <p>Witaj na mojej stronie!</p>
10 +    <script src="js/main.js"></script>
11 +   </body>
12 + </html>
```

js/main.js

+7

@@ -0,0 +1,7 @@

```
1 + const button = document.createElement("button");
2 + button.innerHTML = "Powitanie";
3 + document.body.appendChild(button);
4 +
5 + button.addEventListener("click", function() {
6 +   alert("Witaj na mojej stronie!");
7 + });
```

Instrukcja 2

- Podepnij do strony bibliotekę **CSS Tools: Reset CSS**: <https://meyerweb.com/eric/tools/css/reset/>
- Użyj instrukcji:
`<link rel="stylesheet" href="css/reset.css">`, które umieścisz w sekcji head strony.
- Teraz korzystasz z dwóch arkuszy styli, reset musi podpięty do strony jako pierwszy.
- Zmodyfikuj resztę zawartości pliku **index.html**, zgodnie ze slajdem nr. 9



Dlaczego to potrzebne?

— — —

Każda przeglądarka (np. Chrome, Firefox, Edge) **domyślnie styluje strony inaczej**. Na przykład:

- nagłówki (`<h1>`, `<h2>`) mogą mieć różne marginesy,
- listy (`<ul>`) mogą mieć różne wcięcia,
- formularze mogą wyglądać inaczej na każdej przeglądarce.

To powoduje **bałagan** i trudności w pisaniu spójnego wyglądu strony.

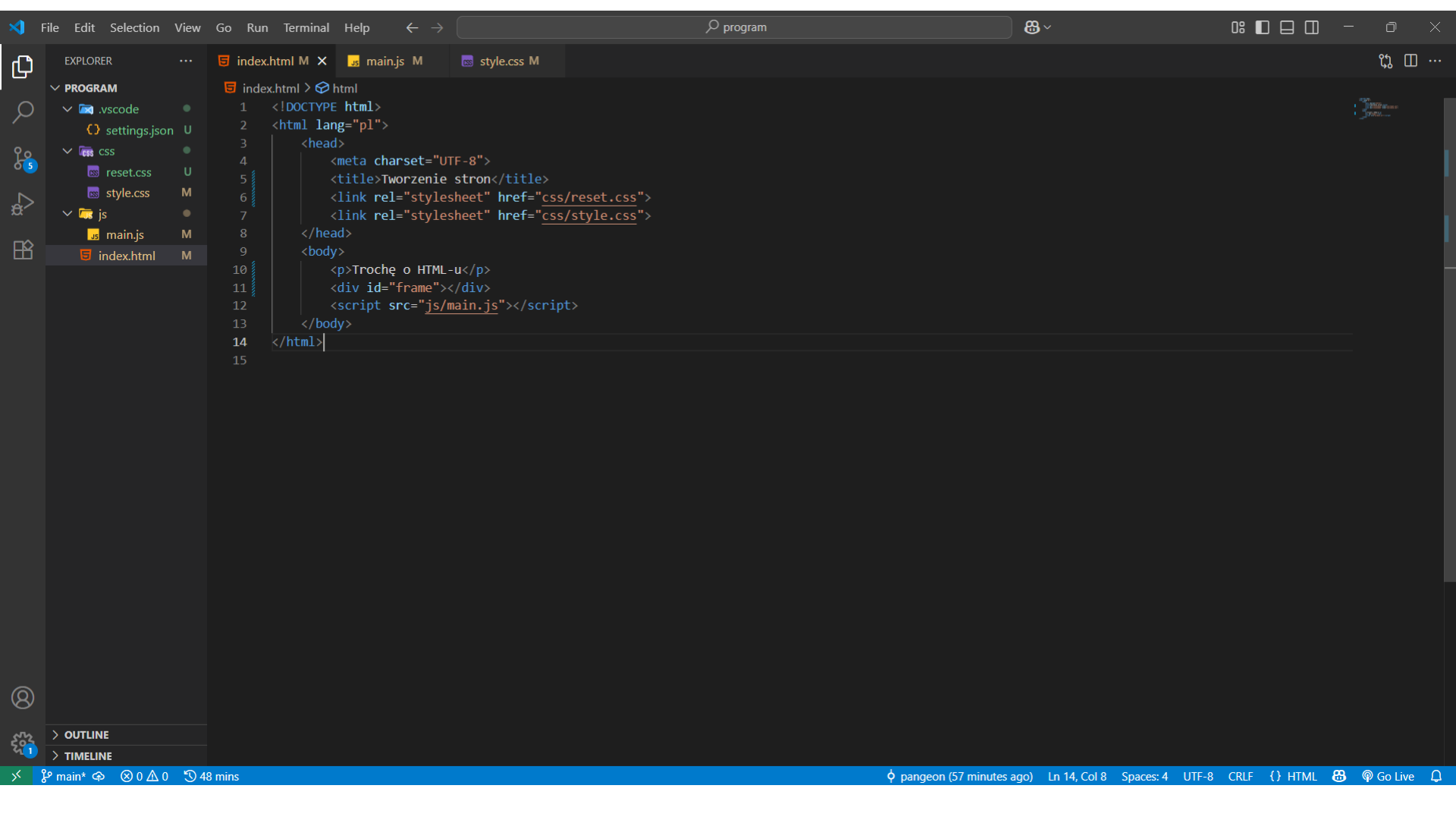


Co robi reset.css?

Reset.css mówi:

"Usuń wszystkie marginesy, odstępy, obramowania i inne style, które przeglądarka sama narzuca."

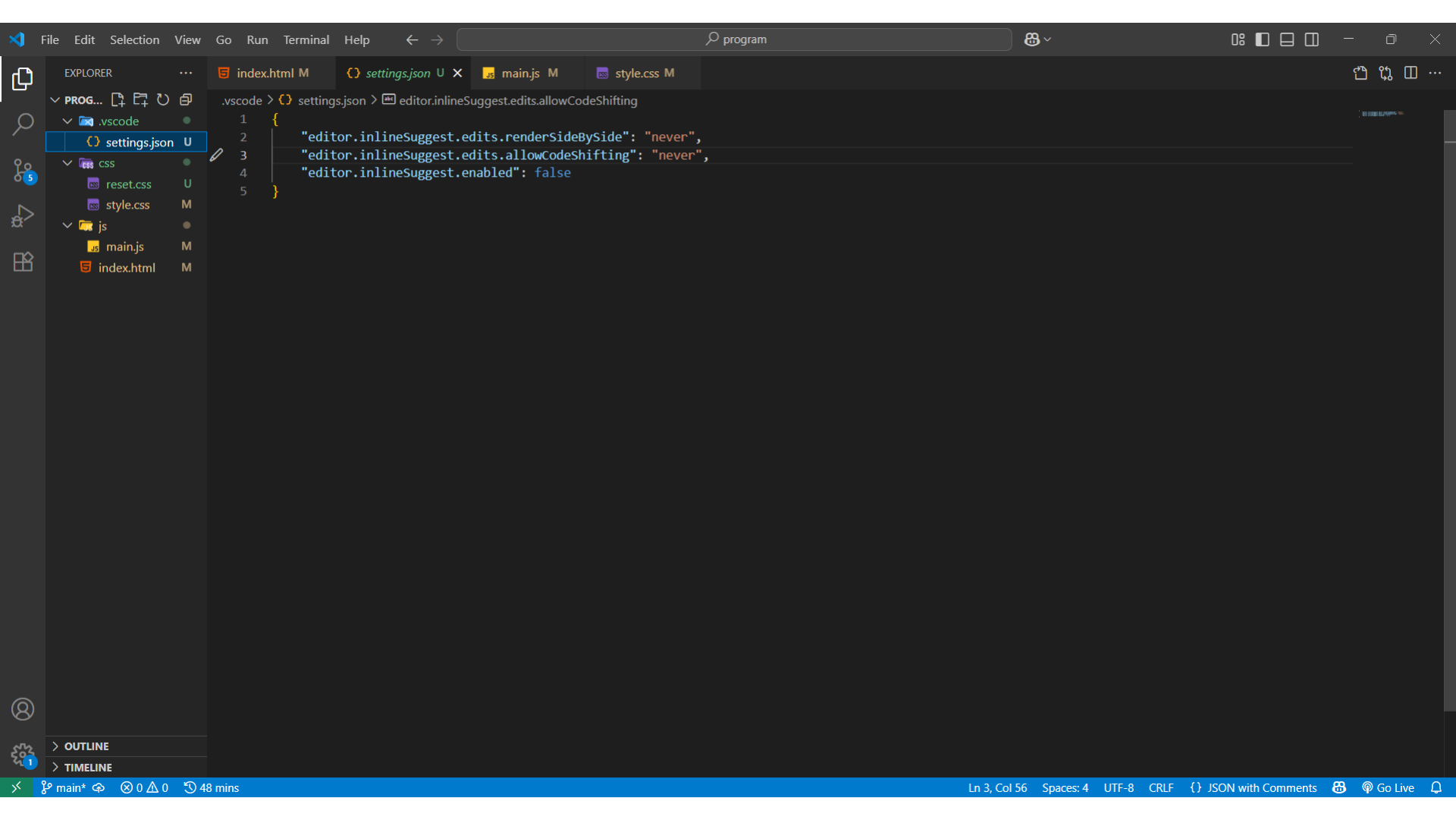
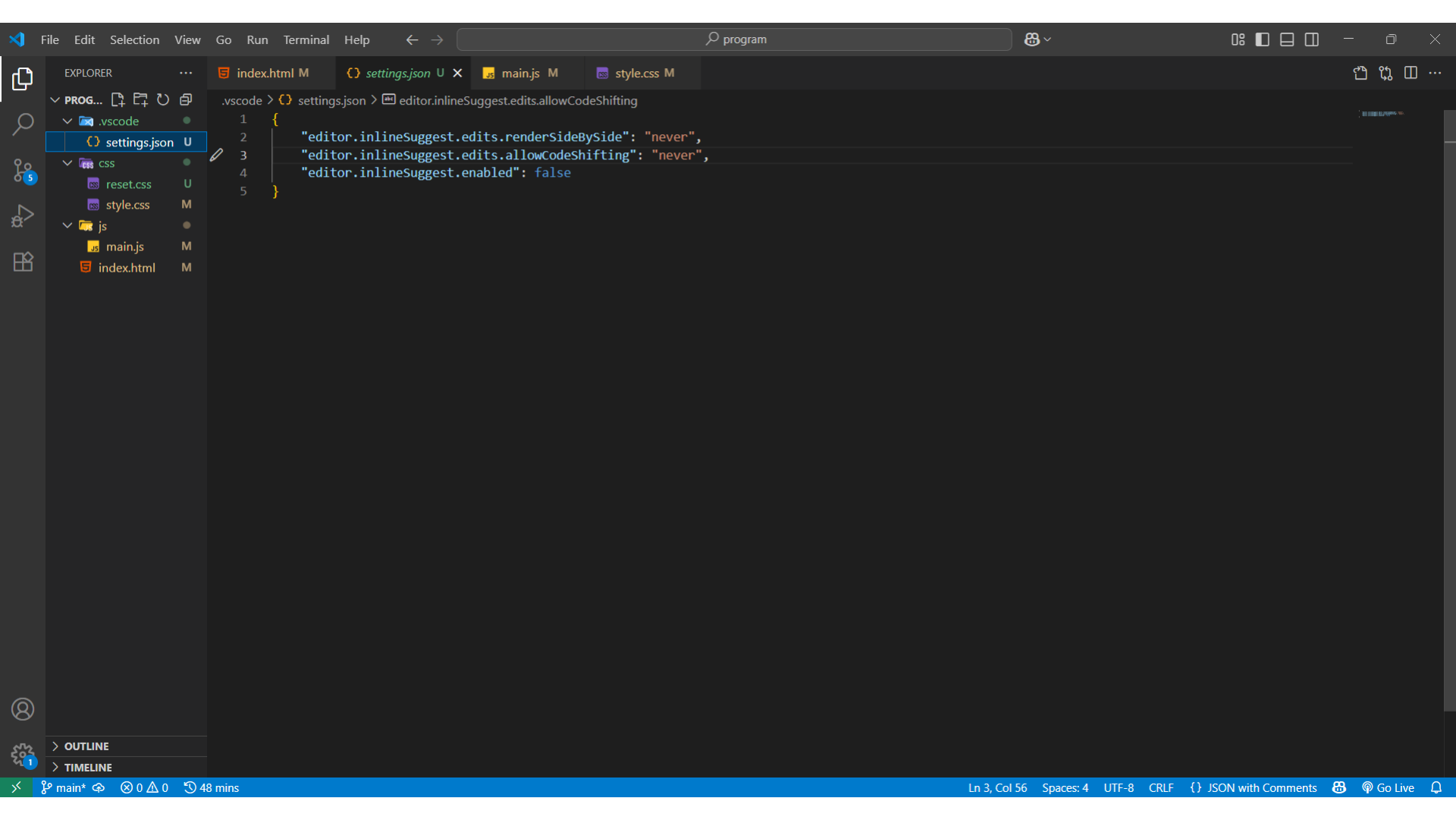
W ten sposób **wszystko wygląda tak samo** we wszystkich przeglądarkach — zupełnie "**od zera**", jak czysta kartka papieru.



Instrukcja 3

- Stwórz w swojej przestrzeni roboczej folder **.vscode**
- W folderze utwórz plik **settings.json**
- Przekopiuuj do pliku zawartość:

```
{  
    "editor.inlineSuggest.edits.renderSideBySide": "never",  
    "editor.inlineSuggest.edits.allowCodeShifting": "never",  
    "editor.inlineSuggest.enabled": false  
}
```



```
const button = document.createElement("button");  
button.innerHTML = "Rozwiń skrót";  
document.body.appendChild(button);
```

```
button.addEventListener("click", function() {  
    alert("HTML (ang. HyperText Markup Language, hipertekstowy język znaczników)");  
});
```

```
const frame = document.getElementById("frame");  
frame.innerHTML = "<p><strong>HTML</strong> pozwala opisać strukturę informacji  
zawartych wewnątrz strony internetowej," +  
    "nadając odpowiednie znaczenie semantyczne poszczególnym fragmentom tekstu -  
formując hiperłącza," +  
    "akapity, nagłówki, listy - oraz osadza w tekście dokumentu obiekty plikowe, na  
przykład multimedia," +  
    "lub elementy baz danych, na przykład interaktywne formularze danych.</p>";
```

Instrukcja 4

— — —

- Przenieś zawartość slajdu 12 pliku **main.js**
- Dokonaj dokładnej analizy każdej instrukcji.
- Sprawdź działanie strony w przeglądarce Chrome z użyciem narzędzi deweloperskich.

1.

javascript

 Kopiuuj  Edytuj

```
const button = document.createElement("button");
```

Co to robi?

— Tworzy **nowy przycisk** (`<button>`) w pamięci przeglądarki.

Jeszcze **nie widać go na stronie** — to jakby był w "prywatnej pracowni".

 Słowo `const` oznacza, że stworzysz zmienną o nazwie `button` i przypisujesz do niej ten nowy przycisk. Nie będzie można tej zmiennej nadpisać inną wartością.

2.

javascript

 Kopiuj  Edytuj

```
button.innerHTML = "Rozwiń skrót";
```

✓ Co to robi?

— Ustawia **tekst widoczny na przycisku**.

W tym wypadku:

 Przycisk będzie miał napis „**Rozwiń skrót**”.

 `innerHTML` to "środek" danego elementu HTML — w tym przypadku jego zawartość tekstowa.

3.

javascript

 Kopiuuj  Edytuj

```
document.body.appendChild(button);
```

✅ Co to robi?

— Bierze przycisk z pamięci i **wstawia go do strony**, dokładnie na jej koniec (do `body`, czyli głównej części strony).

Od tej chwili przycisk jest widoczny dla użytkownika.

Kod

```
document.createElement("button")
```

```
button.innerHTML = "Rozwiń skrót"
```

```
document.body.appendChild(button)
```

Co robi?

Tworzy nowy przycisk w pamięci

Ustawia tekst na przycisku

Dodaje przycisk do strony

```
button.addEventListener("click", function() {  
    alert("HTML (ang. HyperText Markup Language, hipertekstowy język znaczników)");  
});
```



Co robi ten kod — krok po kroku:

1. `button.addEventListener("click", ...)`

To oznacza: „**Nasłuchuj kliknięcia tego przycisku**”

– czyli:

„Jeśli ktoś kliknie w **przycisk** (stworzony wcześniej), to wykonaj **określoną czynność**”.

2. `function() { ... }`

Tutaj opisujesz **co ma się wydarzyć po kliknięciu**.

W tym przypadku: wywoływana jest funkcja (czyli zestaw instrukcji), a w niej mamy:

3. `alert("...")`

To znaczy: „**Pokaż użytkownikowi okienko z wiadomością**”

W tym przypadku:

HTML (ang. HyperText Markup Language, hipertekstowy język znaczników)

To okienko to tzw. **alert** – pojawia się na środku ekranu i trzeba je kliknąć, by zniknęło.

Część kodu

Znaczenie

```
button.addEventListener("click", ...)
```

Gdy ktoś kliknie przycisk, wykonaj coś

```
function() { ... }
```

To właśnie „coś” – opis akcji

```
alert(...)
```

Pokaż okienko z wiadomością

Co ten kod robi — wyjaśnienie krok po kroku:

◆ 1. `const frame = document.getElementById("frame");`

✓ Znaczenie:

- Szukamy na stronie elementu (np. `<div>`) o **identyfikatorze** `frame`
- Zapisujemy do zmiennej `frame`, żeby później coś z tym elementem zrobić.

🧑 Dla laika:

To jakby znaleźć **pudełko** z etykietą `"frame"` i powiedzieć:

„OK, mam je. Będę do niego wkładał zawartość.”

◆ 2. `frame.innerHTML = "...";`

✓ Znaczenie:

- Wkładamy do środka tego pudełka **nowy kod HTML**.
- Cała zawartość tego elementu zostaje **zastąpiona** tym, co podajemy po znaku `=`



◆ 3. Co wkładamy? — fragment HTML:

html

 Kopiuuj  Edytuj

`<p>`

`<strong>`HTML`</strong>` pozwala opisać strukturę informacji zawartych wewnątrz strony internetowej nadając odpowiednie znaczenie semantyczne poszczególnym fragmentom tekstu – formując hiperłącza, akapity, nagłówki, listy – oraz osadza w tekście dokumentu obiekty plikowe, na przykład multimedia lub elementy baz danych, na przykład interaktywne formularze danych.

`</p>`

 Czyli:

Tworzymy **jeden akapit** (`<p>`), w którym słowo „HTML” jest pogrubione dzięki `<strong>` .

Reszta to zwykły tekst wyjaśniający, czym jest HTML.



Efekt dla użytkownika:

Jeśli na stronie jest element np.:

html

 Kopiuj  Edytuj

```
<div id="frame"></div>
```

to po uruchomieniu powyższego kodu, ten element **zostanie wypełniony tekstem** w formacie HTML, np.:

Część kodu

Znaczenie

```
getElementById("frame")
```

Znajduje element o identyfikatorze `frame`

```
innerHTML = "...";
```

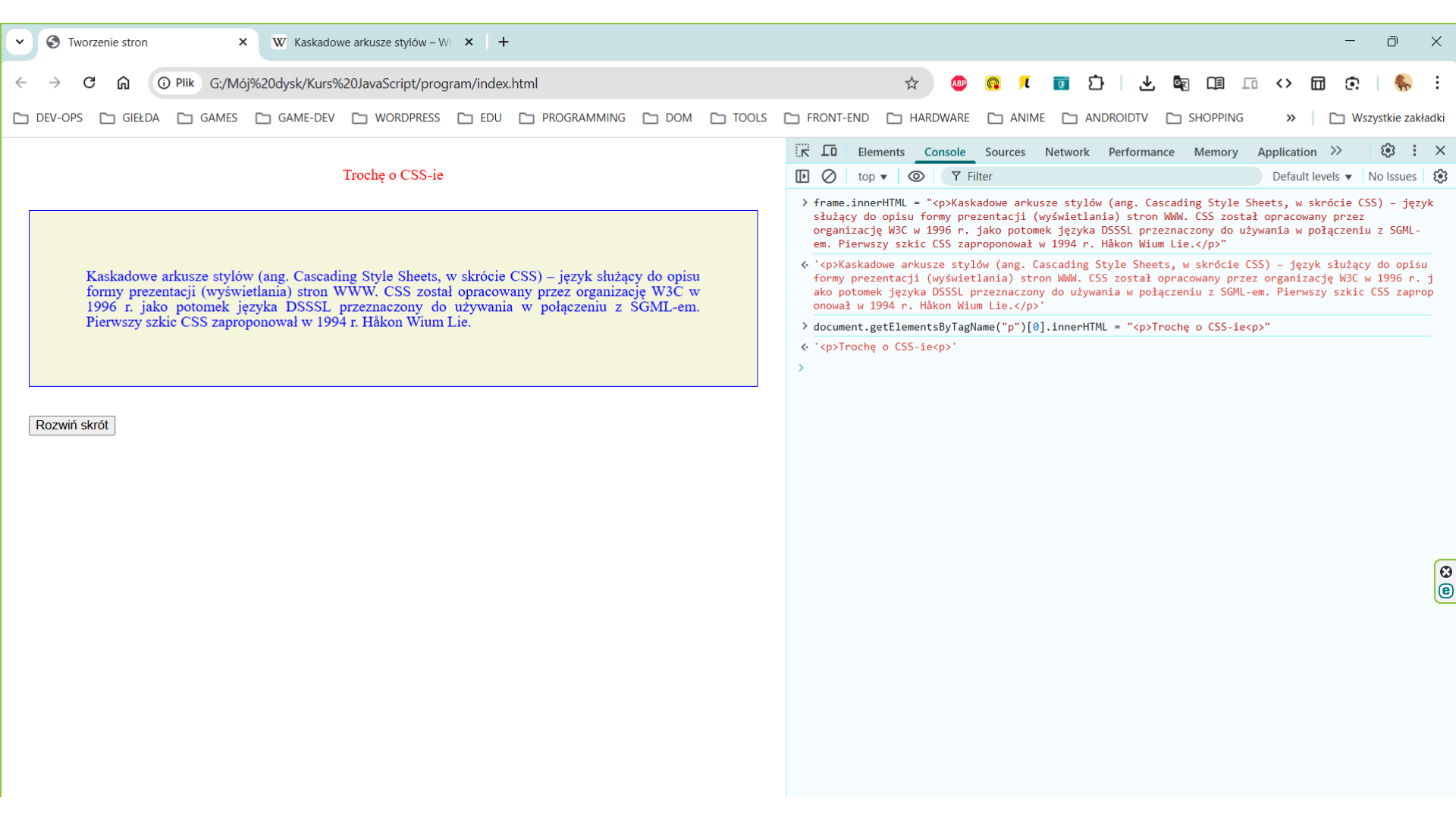
Wstawia nowy fragment HTML do tego elementu

```
<strong>...</strong>
```

Pogrubia słowo „HTML”

```
<p>...</p>
```

Tworzy akapit (paragraf) tekstu



Trochę o CSS-ie

Kaskadowe arkusze stylów (ang. Cascading Style Sheets, w skrócie CSS) – język służący do opisu formy prezentacji (wyświetlania) stron WWW. CSS został opracowany przez organizację W3C w 1996 r. jako potomek języka DSSSL przeznaczony do używania w połączeniu z SGML-em. Pierwszy szkic CSS zaproponował w 1994 r. Håkon Wium Lie.

Rozwiń skrót

Elements Console Sources Network Performance Memory Application >>

top Filter Default levels No Issues

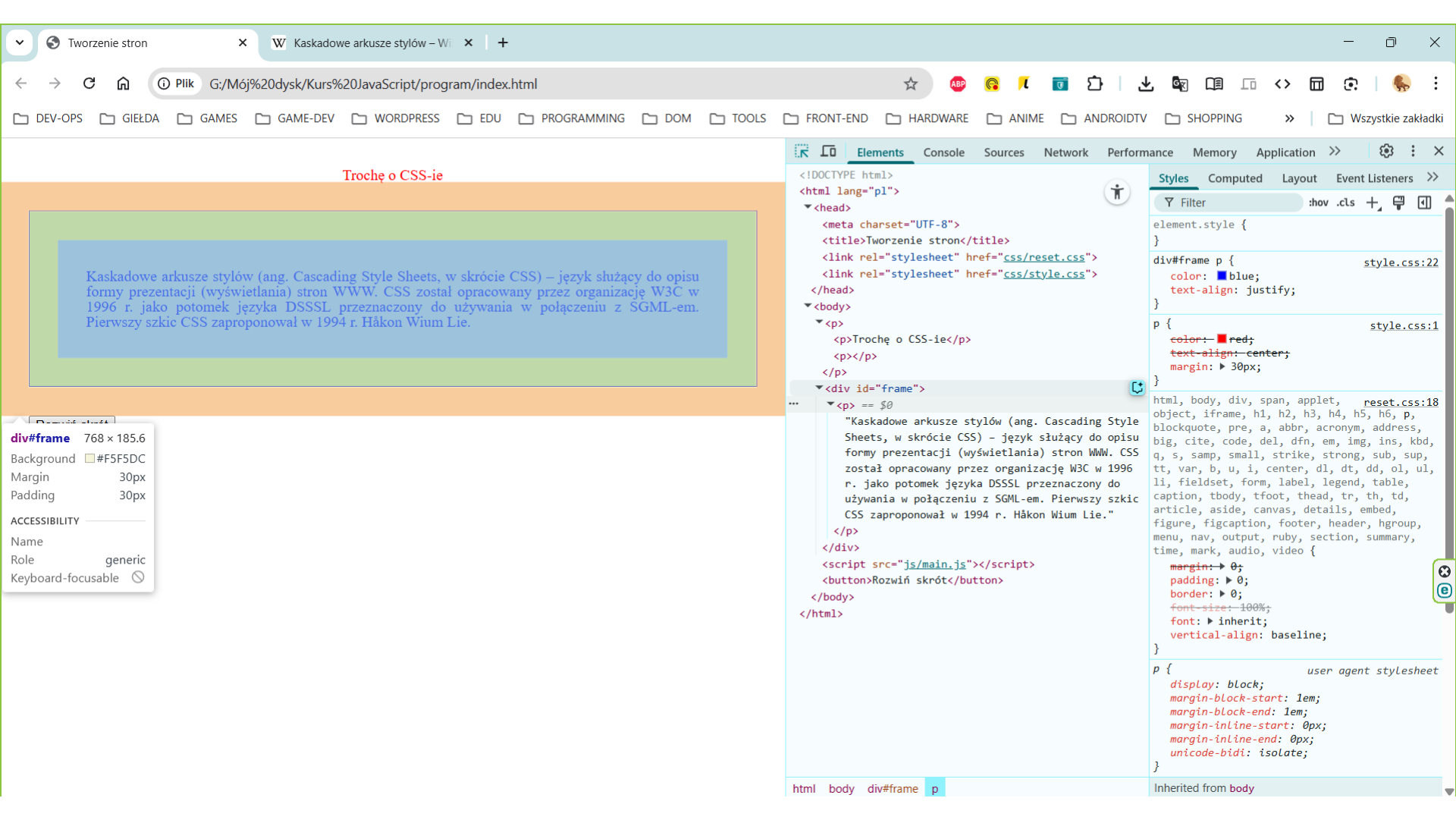
> frame.innerHTML = "<p>Kaskadowe arkusze stylów (ang. Cascading Style Sheets, w skrócie CSS) – język służący do opisu formy prezentacji (wyświetlania) stron WWW. CSS został opracowany przez organizację W3C w 1996 r. jako potomek języka DSSSL przeznaczony do używania w połączeniu z SGML-em. Pierwszy szkic CSS zaproponował w 1994 r. Håkon Wium Lie.</p>"

< 'Kaskadowe arkusze stylów (ang. Cascading Style Sheets, w skrócie CSS) – język służący do opisu formy prezentacji (wyświetlania) stron WWW. CSS został opracowany przez organizację W3C w 1996 r. j ako potomek języka DSSSL przeznaczony do używania w połączeniu z SGML-em. Pierwszy szkic CSS zaprop onował w 1994 r. Håkon Wium Lie.</p>'

> document.getElementsByTagName("p")[0].innerHTML = "<p>Trochę o CSS-ie<p>"

< 'Trochę o CSS-ie<p>'

>



Trochę o CSS-ie

Kaskadowe arkusze stylów (ang. Cascading Style Sheets, w skrócie CSS) – język służący do opisu formy prezentacji (wyświetlania) stron WWW. CSS został opracowany przez organizację W3C w 1996 r. jako potomek języka DSSSL przeznaczony do używania w połączeniu z SGML-em. Pierwszy szkic CSS zaproponował w 1994 r. Håkon Wium Lie.

div#frame 768 × 185.6
Background  #F5F5DC
Margin 30px
Padding 30px
ACCESSIBILITY
Name
Role generic
Keyboard-focusable 

Elements

Console

Sources

Network

Performance

Memory

Application

Styles

Computed

Layout

Event Listeners

<!DOCTYPE html>
<html lang="pl">
 <head>
 <meta charset="UTF-8">
 <title>Tworzenie stron</title>
 <link rel="stylesheet" href="css/reset.css">
 <link rel="stylesheet" href="css/style.css">
 </head>
 <body>
 <p>
 <p>Trochę o CSS-ie</p>
 <p></p>
 <p>
 <div id="frame">
 <p> == \$0
 "Kaskadowe arkusze stylów (ang. Cascading Style Sheets, w skrócie CSS) – język służący do opisu formy prezentacji (wyświetlania) stron WWW. CSS został opracowany przez organizację W3C w 1996 r. jako potomek języka DSSSL przeznaczony do używania w połączeniu z SGML-em. Pierwszy szkic CSS zaproponował w 1994 r. Håkon Wium Lie."
 </p>
 </div>
 <script src="js/main.js"></script>
 <button>Rozwiń skrót</button>
 </body>
 </html>
 </body>
</html>

Filter: show .cls +
element.style {
}
div#frame p { style.css:22
 color: blue;
 text-align: justify;
}
p { style.css:1
 color: red;
 text-align: center;
 margin: 30px;
}
html, body, div, span, applet, reset.css:18
object, iframe, h1, h2, h3, h4, h5, h6, p,
blockquote, pre, a, abbr, acronym, address,
big, cite, code, del, dfn, em, img, ins, kbd,
q, s, samp, small, strike, strong, sub, sup,
tt, var, b, u, i, center, dl, dt, dd, ol, ul,
li, fieldset, form, label, legend, table,
caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary,
time, mark, audio, video {
 margin: 0;
 padding: 0;
 border: 0;
 font-size: 100%;
 font: inherit;
 vertical-align: baseline;
}
p { user agent stylesheet
 display: block;
 margin-block-start: 1em;
 margin-block-end: 1em;
 margin-inline-start: 0px;
 margin-inline-end: 0px;
 unicode-bidi: isolate;
}

html body div#frame p

Inherited from body

KONIEC