

# JavaScript — pierwsze skrypty

Czas: 2 × 45 min

praca indywidualna

**Cel:** Pierwsze programy w JavaScriptcie: konsola przeglądarki, zmienne i typy, operatory (w tym `==` i `===`), instrukcje warunkowe i pętle, wyświetlanie wyniku na stronie oraz szukanie błędów z punktem przerwania.

## Po zajęciach potrafisz:

- dołącza skrypt z pliku i wypisuje wyniki w konsoli F12
- deklaruje zmienne `let/const` i rozpoznaje typy operatorem `typeof`
- przewiduje wynik operatorów `+`, `-`, `%`, `==` i `===` dla liczb i tekstów
- stosuje `if/else if` oraz pętle `for` i `while` (także zagnieżdżone)
- znajduje błąd w skrypcie, zatrzymując go punktem przerwania i obserwując wartości zmiennych

## Sprzęt i materiały

- komputer z Windows 11 i przeglądarką Chrome lub Edge
- folder `r1_21_start` z Pliki\_do\_zadan (`index.html`, `style.css`, `skrypt.js`, `bledy.js`)

## Oprogramowanie

- **Visual Studio Code z Live Server** — po zapisie strona odświeża się sama [code.visualstudio.com](https://code.visualstudio.com)
- **Konsola i debugger przeglądarki** — F12 → Console oraz Sources (punkty przerwania) [developer.chrome.com/docs/devtools/javascript](https://developer.chrome.com/docs/devtools/javascript)

## Przygotowanie przed zajęciami

- Skopiuj folder `r1_21_start` do folderu `nazwisko_imie` jako `zad21`.
- Przypomnij sobie z wykładu 1.15: `let` i `const`, typy, `==` i `===`, `if`, `for`, `while`.

## Bezpieczeństwo

- Nie wklejaj do konsoli przeglądarki kodu z nieznanych źródeł (np. „wklej to, a dostaniesz darmowe punkty”) — może przejąć Twoje konto na tej stronie. W tym zadaniu wpisuj tylko własny kod.
- W pętli `while` zawsze sprawdź, czy warunek kiedyś stanie się fałszywy — nieskończona pętla zawiesza kartę (zamknij ją wtedy przez Menedżera zadań przeglądarki: Shift+Esc).

## Przebieg zadania

### 1 Konsola i zmienne

Uruchom `index.html` przez Live Server i otwórz konsolę (F12 → Console). Każdą część kodu w `skrypt.js` poprzedź komentarzem z numerem polecenia.

- J1. Sprawdź, że w konsoli widać „Skrypt działa!”. W `head` znajdź znacznik `script` z atrybutem `defer` — usuń na chwilę `defer` z `bledy.js`, odśwież i zapisz komunikat błędu z konsoli. Przywróć `defer`.
- J2. Zadeklaruj stałą `imie` (tekst), zmienną `wiek` (liczba), stałą `czyUczen` (wartość logiczna) i zmienną bez wartości. Wypisz każdą z jej typem: `console.log(wiek, typeof wiek)`.
- J3. Zwiększ `wiek` o 1 i wypisz zdanie „Za rok Ola będzie mieć 16 lat.” szablonem tekstowym (odwrotne apostrofy i `${...}`). Spróbuj zmienić wartość stałej `imie` — zapisz komunikat błędu i usuń tę linię.
- J4. Zanim uruchomisz, zapisz w sprawozdaniu przewidywany wynik, potem sprawdź w konsoli: `"5" + 3`, `"5" - 3`, `"5" * "2"`, `5 == "5"`, `5 === "5"`, `7 % 3`, `0.1 + 0.2`.

### 2 Kalkulator BMI w konsoli

BMI = masa [kg] / wzrost<sup>2</sup> [m]. Przedziały: poniżej 18,5 — niedowaga; od 18,5 do mniej niż 25 — waga prawidłowa; od 25 do mniej niż 30 — nadwaga; od 30 — otyłość.

- J5. Zapisz w stałych masę 62 kg i wzrost 1,70 m (w kodzie kropka: `1.70`). Oblicz BMI.
- J6. Instrukcją `if ... else if ... else` dobierz opis według przedziałów. Wypisz wynik z jednym miejscem po przecinku: `bmi.toFixed(1)`.
- J7. Przetestuj skrypt dla 3 innych zestawów danych tak, aby otrzymać trzy różne opisy. Zapisz dane i wyniki w tabeli sprawozdania. Do której kategorii trafi BMI równe dokładnie 25?

### 3 Pętle

- J8. Pętlą `for` wypisz w jednym wierszu konsoli liczby parzyste od 2 do 20 (krok 2, bez instrukcji `if`).
- J9. Pętlą `while` policz, ile razy trzeba podwoić liczbę 1, aby przekroczyła 1000. Wypisz liczbę kroków i końcową wartość.
- J10. Dwoma zagnieżdżonymi pętlami `for` wypisz w konsoli tabliczkę mnożenia 10 × 10 — każdy wiersz w osobnej linii, liczby wyrównane do 4 znaków (`String(liczba).padStart(4)`).
- J11. Zbuduj tę samą tabliczkę jako tabelę HTML w zmiennej tekstowej (wiersz nagłówek 1–10, pierwsza kolumna jako `th`) i wstaw ją do `div#tabliczka` przez `innerHTML`. Komórki na przekątnej (`w === k`) mają klasę `kwadrat`.

### 4 Debugowanie skryptu z błędami

Skrypt `bledy.js` ma policzyć średnią 7 ocen (6 z tablicy i jedna dodatkowa), ale strona pokazuje „Średnia: NaN”. Poprawna średnia to 4,14.

- J12. W F12 → Sources otwórz `bledy.js` i kliknij numer linii z `suma = suma + oceny[i];`, aby ustawić punkt przerwania. Odśwież stronę. Przyciskiem „Step over” (F10) przejdź kilka obrotów pętli i zapisz w tabeli wartości `i`, `oceny[i]` i `suma`. W którym obrocie pojawia się `undefined` i `NaN`?
- J13. Popraw warunki pętli. Odśwież — jaki wynik pokazuje teraz strona? Ponownie zatrzymaj skrypt (punkt przerwania na linii z `dodatkowa`) i sprawdź w panelu Scope typ zmiennej `suma` po dodaniu oceny dodatkowej. Popraw drugi błąd (`Number(...)`).
- J14. Strona ma pokazywać „Średnia: 4.14”. Zrób zrzuty: konsola z wynikami J2–J10, strona z tabliczką i średnią, debugger zatrzymany w pętli.

## W sprawozdaniu umieść

- ☐ tabela przewidywań i wyników operatorów (J4)
- ☐ tabela testów BMI (J7)
- ☐ tabela wartości zmiennych z debugera (J12) i opis obu błędów z poprawkami
- ☐ rzuty: konsola, strona z tabliczką i średnią, debugger z punktem przerwania
- ☐ pliki skrypt.js i błedy.js
- ☐ odpowiedzi na pytania kontrolne

## Pytania kontrolne

1. Czym różni się let od const? Kiedy użyjesz każdego z nich?
2. Dlaczego "5" + 3 daje "53", a "5" - 3 daje 2? Którego porównania (== czy ===) warto używać i dlaczego?
3. Co robi atrybut defer i co się dzieje bez niego, gdy skrypt w head szuka elementu na stronie?
4. Jak punkt przerwania pomaga znaleźć błąd szybciej niż dopisywanie console.log?

## Kryteria oceny

| Kryterium                                                                             | Punkty    |
|---------------------------------------------------------------------------------------|-----------|
| Zmienne, typy, szablon tekstowy, błąd zmiany stałej, komunikat bez defer (J1–J3)      | 3         |
| Tabela przewidywań operatorów z wyjaśnieniem różnic (J4)                              | 2         |
| Kalkulator BMI z poprawnymi przedziałami i 4 testami (J5–J7)                          | 4         |
| Pętle: parzyste, podwajanie, tabliczka w konsoli (J8–J10)                             | 3         |
| Tabliczka jako tabela na stronie z wyróżnioną przekątną (J11)                         | 2         |
| Debugowanie: tabela wartości, oba błędy znalezione i poprawione, wynik 4.14 (J12–J14) | 4         |
| Odpowiedzi na pytania kontrolne                                                       | 2         |
| <b>Razem</b>                                                                          | <b>20</b> |

Sprawozdanie przygotuj na wzorze dołączonym do zadania (plik .docx o tym samym numerze).