

JavaScript — funkcje i tablice

Czas: 2 × 45 min

praca indywidualna

Cel: Pisanie własnych funkcji z parametrami i wartością zwracaną oraz przetwarzanie tablic liczb: średnia, minimum i maksimum, sortowanie liczbowe, filter i map.

Po zajęciach potrafisz:

- deklaruje funkcje klasyczne i strzałkowe z parametrami, wartością domyślną i instrukcją return
- wywołuje jedną funkcję wewnątrz drugiej i rozumie zasięg zmiennych lokalnych
- oblicza średnią, minimum i maksimum elementów tablicy
- sortuje liczby poprawnie funkcją porównującą i wyjaśnia, dlaczego `sort()` bez niej daje zły wynik
- stosuje metody `push`, `pop`, `forEach`, `filter` i `map`

Sprzęt i materiały

- komputer z Windows 11 i przeglądarką Chrome, Edge lub Firefox
- folder `r1_23_start` z Pliki_do_zadan (`index.html`, `styl.css`, `skrypt.js`)

Oprogramowanie

- **Visual Studio Code** — rozszerzenie Live Server (podgląd odświeżany po zapisie) code.visualstudio.com
- **Przeglądarka z narzędziami F12** — zakładka Console — komunikaty błędów i testy wyrażen

Przygotowanie przed zajęciami

- Przypomnij sobie z prezentacji 1.16 deklarację funkcji, funkcję strzałkową i metody tablic.
- Skopiuj folder `r1_23_start` do swojego folderu `nazwisko_imie` i zmień jego nazwę na `zad23`.

Bezpieczeństwo

- Pracuj na kopii plików startowych w swoim folderze; nie zmieniaj plików w Pliki_do_zadan.

Przebieg zadania

1 Przygotowanie

Skrypt dołączono w sekcji head z atrybutem `defer`, więc wykona się po wczytaniu całej strony. Funkcja `dodajDoListy(idListy, tekst)` jest gotowa — używaj jej do wyświetlania każdego wyniku.

```
dodajDoListy("wyniki-funkcje", "F1. Pole prostokąta 4 × 7.5 = " + poleProstokata(4, 7.5));
```

1. Otwórz folder `zad23` w VS Code (Plik → Otwórz folder) i uruchom `index.html` przez Live Server (Go Live na pasku stanu).
2. Wpisz w `index.html` i w komentarzu na początku `skrypt.js` swoje imię i nazwisko. Otwórz konsolę F12 i sprawdź, że na stronie widać napis „Skrypt działa!”. Potem usuń ten przykład.
3. Każdy wynik wyświetlaj z numerem polecenia (F1, B3, C2...), tak jak w przykładzie powyżej. Po każdym poleceniu zapisz plik i sprawdź konsolę — czerwony komunikat zatrzymuje cały dalszy skrypt.

2 Część A. Funkcje liczące (lista wyniki-funkcje)

1. F1. Napisz funkcję `poleProstokata(a, b)` zwracającą pole prostokąta. Wyświetl pole dla boków 4 i 7.5.
2. F2. Napisz funkcję `poleKola(r)` korzystającą z `Math.PI`. Wyświetl pole koła o promieniu 3 zaokrąglone do 2 miejsc metodą `toFixed(2)`.
3. F3. Zapisz jako funkcję strzałkową `cenaPoRabacie(cena, procent)`. Wyświetl cenę towaru za 249.99 zł po rabacie 15% (2 miejsca po przecinku).
4. F4. Napisz funkcję `brutto(netto, stawka = 23)` z parametrem domyślnym. Wyświetl cenę brutto dla 100 zł netto przy wywołaniu z jednym argumentem oraz ze stawką 8%.
5. F5. Napisz funkcję `kwotaVat(netto, stawka = 23)`, która **wywołuje** `brutto()` i zwraca różnicę brutto – netto. Wyświetl VAT od 350 zł netto.
6. F6. W funkcji `test()` zadeklaruj `const wewnatrz = "..."` i zwróć tę stałą. Wyświetl wynik `test()`, a następnie poza funkcją dopisz `console.log(wewnatrz);`. Zanotuj komunikat z konsoli, sprawdź, czy wyświetliły się dalsze wyniki, i zamień tę linię na komentarz.

3 Część B. Tablica ocen (lista wyniki-oceny)

Pracuj na tablicy `oceny` z pliku startowego. Nie wpisuj wyników ręcznie — mają być liczone przez skrypt, tak by po zmianie tablicy zmieniły się same.

1. B1. Wyświetl wszystkie oceny oddzielone przecinkami (`join`) i ich liczbę (`length`).
2. B2. Napisz funkcję `srednia(tab)`, która pętlą `for...of` sumuje elementy i zwraca średnią. Wyświetl średnią ocen z 2 miejscami po przecinku.
3. B3. Wyświetl najniższą i najwyższą ocenę (`Math.min`, `Math.max` z operatorem `...` albo pętlą).
4. B4. Wyświetl oceny posortowane rosnąco, a pod spodem tablicę oryginalną. Oryginał ma zostać bez zmian — sortuj kopię `[...oceny]`.
5. B5. Metodą `filter` utwórz tablicę ocen co najmniej dobrych (≥ 4). Wyświetl ją i liczbę jej elementów.
6. B6. Napisz funkcję `slovnice(o)` z instrukcją `switch` (6 $\rightarrow$ „celujący” ... 1 $\rightarrow$ „niedostateczny”) i metodą `map` wyświetl wszystkie oceny słownie.
7. B7. Dodaj na koniec tablicy ocenę 6 (`push`) i wyświetl nową liczbę ocen i średnią, korzystając z funkcji z B2. Następnie usuń ostatni element (`pop`) i wyświetl zwróconą wartość.

4 Część C. Punkty z testu (lista wyniki-punkty)

Tablica `punkty` zawiera wyniki testu, w którym można było zdobyć 120 punktów.

1. C1. Wyświetl wynik `[...punkty].sort()` bez argumentów, potem posortowane rosnąco funkcją porównującą `(a, b) => a - b`, a na końcu malejąco. Porównaj pierwszy wiersz z drugim.
2. C2. Metodą `map` utwórz tablicę `procenty` (`punkty / 120 * 100`, zaokrąglone `Math.round`) i wyświetl ją ze znakiem %.
3. C3. Metodą `filter` policz, ilu uczniów zaliczyło test (co najmniej 50%). Wyświetl np. „Zaliczyło: 5 z 9”.
4. C4. Metodą `forEach` z dwoma parametrami (`p, i`) wyświetl każdy wynik w osobnym wierszu: „Uczeń 1: 87 pkt (73%)”.
5. Zapisz pliki, zrób zrzut całej strony oraz zrzut konsoli z komunikatem z F6. Sprawdź, czy w konsoli nie ma żadnych błędów.

W sprawozdaniu umieść

- ☐ tabela F1–F6, B1–B7 i C1–C4 z fragmentem kodu i wynikiem wyświetlonym na stronie
- ☐ zrzut całej strony z wynikami i zrzut konsoli z komunikatem z F6
- ☐ wyjaśnienie różnicy między `sort()` a `sort((a, b) => a - b)` na przykładzie z C1

☐ folder `zad23` z plikami `index.html`, `styl.css`, `skrypt.js`

☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Czym różni się funkcja, która wyświetla wynik, od funkcji, która go zwraca (`return`)? Która jest bardziej uniwersalna i dlaczego?
2. Co się stanie, gdy wywołasz `brutto(100)` bez drugiego argumentu, a co, gdyby funkcja nie miała wartości domyślnej?
3. Dlaczego `[87, 9, 100].sort()` zwraca `[100, 87, 9]`? Jak działa funkcja porównująca `(a, b) => a - b`?
4. Czym różnią się metody `map` i `filter` pod względem długości tablicy wynikowej?

Kryteria oceny

Kryterium	Punkty
F1-F5 — poprawne funkcje z parametrami i return, toFixed, funkcja strzałkowa, wartość domyślna	5
F6 — test zasięgu z zanotowanym komunikatem błędu	1
B1-B7 — wyniki liczone przez skrypt, oryginał tablicy niezmieniony po sortowaniu	6
C1-C4 — sortowanie liczbowe, map, filter, forEach z indeksem	4
Kod czytelny (komentarze z numerami poleceń, wcięcia), brak błędów w konsoli, rzuty	2
Odpowiedzi na pytania kontrolne	2
Razem	20

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