

JavaScript — obiekty i tablice obiektów

Czas: 2 × 45 min

praca indywidualna

Cel: Przetwarzanie tablicy obiektów: wyświetlanie danych w tabeli HTML, obliczanie statystyk, sortowanie po różnych polach, filtrowanie na żywo oraz użycie obiektów wbudowanych `Math` i `Date`.

Po zajęciach potrafisz:

- tworzy obiekt literalny z właściwościami i metodą oraz odczytuje i dodaje właściwości
- generuje wiersze tabeli HTML z tablicy obiektów pętlą
- sortuje tablicę obiektów po polu liczbowym i tekstowym (`localeCompare`)
- filtruje dane według tekstu wpisanego przez użytkownika bez rozróżniania wielkości liter
- losuje element tablicy (`Math.random`) i oblicza liczbę dni między datami (`Date`)

Sprzęt i materiały

- komputer z Windows 11 i przeglądarką Chrome, Edge lub Firefox
- folder `r1_24_start` z Pliki_do_zadan (`index.html` , `styl.css` , `dane.js` , `skrypt.js`)

Oprogramowanie

- **Visual Studio Code** — rozszerzenie Live Server code.visualstudio.com
- **Przeglądarka z narzędziami F12** — Console (`console.table`), Elements — podgląd wygenerowanych wierszy

Przygotowanie przed zajęciami

- Przypomnij sobie z prezentacji 1.16 obiekty literalne, tablice obiektów, `sort` z funkcją porównującą, `Math` i `Date`.
- Skopiuj folder `r1_24_start` do swojego folderu `nazwisko_imie` jako `zad24`.

Bezpieczeństwo

- Teksty wstawiaj do strony przez `textContent`, a nie `innerHTML` — to dobry nawyk, który w zadaniu 25 uzasadnimy dokładniej.
- Nie zmieniaj danych w `dane.js` — wyniki porównujesz z rozwiązaniem nauczyciela.

Przebieg zadania

1 Obiekt i dane w konsoli

Plik `dane.js` wczytuje się przed `skrypt.js` i tworzy tablicę `produkty`. Każdy element ma pola `id`, `nazwa`, `kategoria`, `cena`, `ilosc`. Przyciski i pole wyszukiwania są już podłączone do pustych funkcji na końcu `skrypt.js`.

1. Uruchom stronę przez Live Server i otwórz konsolę F12. Wpisz w swoim komentarzu imię i nazwisko.
2. O1. Utwórz obiekt `sklep` z właściwościami `nazwa`, `miasto`, `godziny` i metodą `opis()` zwracającą jeden tekst z tymi danymi (`this`). Po utworzeniu dodaj właściwość `telefon`. Wyświetl opis i telefon w akapicie `#info-sklep`.
3. O2. Wyświetl tablicę w konsoli poleceniem `console.table(produkty)`, a w `console.log` nazwę pierwszego produktu i liczbę produktów. Zrób zrzut konsoli.

2 Tabela i statystyki

Komórkę tabeli tworzysz tak (powtórz w pętli dla wszystkich pól):

```
const td = document.createElement("td");
td.textContent = p.nazwa;
tr.append(td);
```

1. O3. Uzupełnij funkcję `renderujTabele(lista)`: wyczyść `#tabela-produkty` (`tbody.textContent = ""`), a następnie dla każdego produktu dodaj wiersz z nazwą, kategorią, ceną z 2 miejscami i „zł” (komórka z klasą `cena`) oraz ilością („4 szt.” albo „brak”). Wiersz produktu z ilością 0 ma dostać klasę `brak`. Na końcu wpisz do `#znaleziono` tekst „Wyświetlono: X z 14”.
2. O4. Napisz funkcję `statystyki()`, która dopisuje do listy `#statystyki`: liczbę produktów, najtańszy i najdroższy produkt (nazwa i cena), średnią cenę, wartość magazynu (suma cena × ilość) i nazwy produktów niedostępnych (`filter` + `map` + `join`). Wywołaj ją po załadowaniu.

3 Sortowanie i wyszukiwanie

Zadeklaruj zmienną `let widoczne = produkty;` przechowującą aktualnie wyświetloną listę. Sortuj zawsze kopię (`[...widoczne]`), aby tablica `produkty` zachowała pierwotną kolejność.

1. O5. `sortujPoCenie()` — sortuje widoczne produkty rosnąco po cenie i ponownie rysuje tabelę.
2. O6. `sortujPoNazwie()` — sortuje alfabetycznie metodą `a.nazwa.localeCompare(b.nazwa, "pl")`. Sprawdź, gdzie trafił „łańcuch 8-rzędowy”, a potem sprawdź (np. w konsoli), gdzie trafiłby przy porównaniu operatorem `<`.
3. O7. `szukaj()` — pokazuje tylko produkty, których nazwa zawiera wpisany tekst, bez względu na wielkość liter (`toLowerCase`, `includes`). Sprawdź: „KASK” → 1 wynik, „ro” → 3 wyniki. Puste pole ma pokazać wszystkie produkty. Po wyszukaniu kliknij „Sortuj po cenie” — sortuje się tylko wynik wyszukiwania.

4 Math i Date

Pamiętaj: w `new Date(rok, miesiąc, dzień)` miesiące liczy się od 0 (styczeń = 0). Różnica dwóch dat to liczba milisekund.

1. O8. `losujProduktDnia()` — losuje (`Math.floor(Math.random() * n)`) jeden produkt **dostępny** (ilość > 0) i wyświetla w `#produkt-dnia` nazwę, cenę po rabacie 10% i cenę pierwotną. Kliknij kilka razy i sprawdź, czy nigdy nie wypada produkt z ilością 0.
2. O9. Napisz funkcję `odliczanie()`, która w `#odliczanie` wyświetla dzisiejszą datę (`toLocaleDateString("pl-PL")`) i liczbę dni do rajdu 26 czerwca 2027 r. (`Math.ceil`). Gdy data minęła, ma się pojawić komunikat „Rajd już się odbył”. Sprawdź tę gałąź, zmieniając tymczasowo rok na 2025.
3. Sprawdź, czy konsola nie pokazuje błędów. Zrób zrzuty: strona po załadowaniu, wynik wyszukiwania „ro” posortowany po cenie, tabela posortowana po nazwie.

W sprawozdaniu umieść

- ☐ tabela O1–O9 z fragmentami kodu i wynikami
- ☐ zrzuty: konsola z `console.table`, strona po załadowaniu, wyszukiwanie „ro” po sortowaniu, sortowanie po nazwie
- ☐ porównanie sortowania z `localeCompare` i operatorem `<` (pozycja „łańcucha”)
- ☐ folder `zad24` z wszystkimi plikami
- ☐ odpowiedzi na pytania kontrolne

Pytania kontrolne

1. Jak odczytać cenę trzeciego produktu z tablicy `produkty`? Podaj dwa sposoby zapisu dostępu do właściwości.
2. Dlaczego sortujemy kopię `[...widoczne]`, a nie tablicę `produkty`? Co by się stało z wyszukiwarką, gdyby oryginał był sortowany?
3. Co zwraca `Math.random()` i dlaczego do wylosowania indeksu potrzebne są `Math.floor` i mnożenie przez długość tablicy?
4. Dlaczego `new Date(2027, 5, 26)` oznacza czerwiec, a nie maj?

Kryteria oceny

Kryterium	Punkty
O1–O2 — obiekt z metodą i dodaną właściwością, dane w konsoli	2
O3 — tabela 14 wierszy generowana skryptem, ceny z „zł”, wiersze „brak” wyróżnione klasą	4
O4 — 6 poprawnych statystyk liczonych z danych	3
O5–O7 — sortowanie po cenie i nazwie (polskie znaki), wyszukiwarka bez względu na wielkość liter	5
O8–O9 — losowanie tylko dostępnych produktów, poprawna liczba dni i obsługa daty minionej	3
Zrzuty, brak błędów w konsoli, odpowiedzi na pytania	3
Razem	20

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