

## JavaScript: biblioteki, JSON i AJAX (ściega)

### Dołączenie biblioteki

```
<script src="jquery-3.7.1.min.js"></script>
<script src="skrypt.js"></script>
```

Biblioteka PRZED własnym skryptem, oba przed `</body>`. Zła kolejność daje błąd `$ is not defined`. Człon `min` to wersja bez spacji i komentarzy.

### jQuery: selektory i metody

| Zapis                                              | Co robi                                       | To samo w czystym JavaScriptcie                               |
|----------------------------------------------------|-----------------------------------------------|---------------------------------------------------------------|
| <code>\$("#cena")</code>                           | element o identyfikatorze <code>cena</code>   | <code>document.getElementById("cena")</code>                  |
| <code>\$(".produkt")</code>                        | wszystkie elementy klasy <code>produkt</code> | <code>document.querySelectorAll(".produkt")</code>            |
| <code>\$( "p" )</code>                             | wszystkie znaczniki <code>&lt;p&gt;</code>    | <code>document.getElementsByTagName("p")</code>               |
| <code>\$(document).ready(f)</code>                 | wykonaj <code>f</code> po zbudowaniu drzewa   | <code>document.addEventListener("DOMContentLoaded", f)</code> |
| <code>\$("#a").text("x")</code>                    | wstawia tekst                                 | <code>el.textContent = "x"</code>                             |
| <code>\$("#a").html("&lt;b&gt;x&lt;/b&gt;")</code> | wstawia kod HTML                              | <code>el.innerHTML = "&lt;b&gt;x&lt;/b&gt;"</code>            |
| <code>\$("#a").val()</code>                        | czyta wartość pola                            | <code>el.value</code>                                         |
| <code>\$("#a").css("color", "red")</code>          | ustawia styl                                  | <code>el.style.color = "red"</code>                           |
| <code>\$("#a").hide()</code>                       | ukrywa element                                | <code>el.style.display = "none"</code>                        |
| <code>\$("#a").show()</code>                       | pokazuje element                              | <code>el.style.display = "block"</code>                       |
| <code>\$("#a").click(f)</code>                     | podpina kliknięcie                            | <code>el.addEventListener("click", f)</code>                  |

Bez argumentu metoda czyta, z argumentem zapisuje. Metody łączy się w łańcuch, bo każda oddaje tę samą kolekcję.

```
$("#licz").click(function () {
  const ilosc = Number($("#ilosc").val());
  $("#suma").text(ilosc * 49).css("color", "green");
});
```

### Nazwy do rozpoznania

| Nazwa            | Co to                               | Język      |
|------------------|-------------------------------------|------------|
| jQuery           | biblioteka, skraca pracę z DOM      | JavaScript |
| React            | biblioteka interfejsu z komponentów | JavaScript |
| Angular, Vue     | framework aplikacji                 | JavaScript |
| Symfony, Laravel | framework                           | PHP        |
| Bootstrap        | biblioteka gotowych stylów          | CSS        |
| Node.js          | środowisko poza przeglądarką        | JavaScript |

### JSON i jego funkcje

```
{ "nazwa": "Mysz", "cena": 59, "dostepny": true, "tagi": ["mysz"] }
```

Klucz zawsze w cudzysłowie podwójnym, wartość to napis, liczba, `true`, `false`, `null`, tablica albo obiekt. Bez funkcji, bez komentarzy, bez przecinka po ostatniej parze.

```
const produkt = JSON.parse('{ "nazwa": "Mysz", "cena": 59 }');
console.log(produkt.nazwa + " " + JSON.stringify(produkt));
```

Wynik: `Mysz {"nazwa":"Mysz","cena":59}`. `JSON.parse` idzie z napisu na obiekt, `JSON.stringify` odwrotnie.

### AJAX

Asynchronous JavaScript and XML: pobranie danych z serwera w tle, bez przeładowania strony. Dane jadą dziś w JSON, nie w XML.

```
fetch("produkty.json")
  .then(function (o) {
    return o.json();
  })
  .then(function (dane) {
    document.getElementById("nazwa").textContent = dane.nazwa;
  });
```

Odpowiedź przychodzi później, więc dalszy ciąg siedzi w `.then`, nie w kolejnym wierszu.

### Pułapki

- `.val()` i `.value` oddają napis: `"2" + 3` daje 23, liczbę robi `Number` albo `parseInt`.
- Biblioteka czytana po skrypcie znaczy `$ is not defined`.
- `.text()` pokazuje znaczniki jako znaki, `.html()` je wykonuje.
- W JSON apostrof zamiast cudzysłowu i klucz bez cudzysłowu to błąd składni.
- AJAX to technika wymiany danych, nie język programowania.
- Framework dla PHP to Symfony albo Laravel, nie Angular ani Spring.

**Egzamin praktyczny:** biblioteki i JSON stoją na liście innych zadań; plik `jQuery` leży wtedy na dysku stanowiska, a skrypt pobiera wartość pola przez `.val()`, sprawdza ją instrukcją `if` i wpisuje wynik przez `.text()` albo `.html()`.