



jQuery to popularny framework JavaScript, który umożliwia łatwe i szybkie tworzenie skryptów i interakcji z elementami HTML.

Selektory

Selektory w jQuery są podstawą do manipulowania elementami HTML.

1) Selektor uniwersalny

```
$("#*") // wybiera wszystkie elementy HTML
```

Wybiera wszystkie elementy w dokumencie.

2) Selektor elementu

```
$("element") // wybiera wszystkie elementy danego typu
```

Wybiera wszystkie elementy HTML danego typu. Jest to bezpośredni odpowiednik selektora CSS.

Przykład: `$("p")` // wybiera wszystkie elementy `<p>`

3) Selektor id

```
$("#nazwa-id") // wybiera element o ID "nazwa-id"
```

Wybiera element na podstawie jego unikalnego atrybutu id.

4) Selektor klasy

```
$(".nazwa-klasy") // wybiera wszystkie elementy, które mają klasę "nazwa-klasy"
```

Używany do wyboru wszystkich elementów z określoną klasą.

5) Selektor wielu elementów

```
$("element-1, element-2, element-3")
```

Wybiera wszystkie elementy HTML danych typów.

6) Selektor wielu klas

```
$(".nazwa-klasy-1, .nazwa-klasy-2, .nazwa-klasy-3")
```

Używany do wyboru wszystkich elementów z określoną klasą.

7) Selektor :first

```
$("element:first")
```

Wybiera pierwszy element HTML danego typu.

8) Selektor :last

```
$("#element:last")
```

Wybiera ostatni element HTML danego typu.

9) Selektor atrybutu

```
$("[nazwa-atrybutu"])
```

Wybiera wszystkie elementy posiadające określony atrybut (np. class, id)

Eventy (zdarzenia)

Zdarzenia jQuery obejmują obsługę interakcji użytkownika i innych zdarzeń na stronie internetowej.

1. Metoda bind()

Metoda `bind()` w jQuery służy do przypisania jednego lub więcej zdarzeń do elementu i podania funkcji obsługi zdarzenia.

Składnia:

```
$(element).bind(eventType, eventData, handler);
```

- `eventType`: Typ zdarzenia (np. `click`, `blur`, `change`), które chcemy obsłużyć.

Przykład:

```
$("#p").bind("click", function() {  
    alert("Kliknięto akapit!");  
});
```

2. Metoda blur()

Metoda `blur()` jest używana do obsługi zdarzenia utraty fokusu przez element, np. pole tekstowe.

Składnia:

```
$(element).blur(handler);
```

Przykład:

```
$("#input").blur(function() {  
    alert("Pole utraciło fokus.");  
});
```

3. Metoda change()

Metoda `change()` obsługuje zdarzenie zmiany wartości elementu, szczególnie dla pól formularzy, takich jak `<input>`, `<select>`, `<textarea>`.

Składnia:

```
$(element).change(handler);
```

Przykład:

```
$("select").change(function() {  
    alert("Zmieniono wybór.");  
});
```

4. Metoda click()

Metoda `click()` służy do obsługi zdarzenia kliknięcia na element.

Składnia:

```
$(element).click(handler);
```

Przykład:

```
$("button").click(function() {  
    alert("Kliknięto przycisk!");  
});
```

5. Metoda dblclick()

Metoda `dblclick()` obsługuje zdarzenie podwójnego kliknięcia na element.

Składnia:

```
$(element).dblclick(handler);
```

Przykład:

```
$("p").dblclick(function() {  
    alert("Podwójne kliknięcie!");  
});
```

6. Metoda mouseenter()

Obsługuje zdarzenie wejścia kursora myszy na wybrany element.

Składnia:

```
$(element).mouseenter(handler);
```

- **handler:** Funkcja, która zostanie wykonana, gdy mysz najedzie na element.

Przykład:

```
$("div").mouseenter(function() {  
    $(this).css("background-color", "green");  
});
```

7. Metoda mouseleave()

Obsługuje zdarzenie opuszczenia kursora myszy z wybranego elementu.

Składnia:

```
$(element).mouseleave(handler);
```

- **handler:** Funkcja, która zostanie wykonana, gdy mysz opuści element.

Przykład:

```
$("#div").mouseleave(function() {  
    $(this).css("background-color", "red");  
});
```

8. Metoda hover()

Służy do obsługi zdarzeń związanych z najechaniem kursorem myszy na element oraz jego opuszczeniem.

Składnia:

```
$(element).hover(handlerIn, handlerOut);
```

- **handlerIn:** funkcja, która zostanie wykonana, gdy mysz najedzie na element.
- **handlerOut:** funkcja, która zostanie wykonana, gdy mysz opuści element.

Przykład:

```
$("#div").hover(  
    function() {  
        $(this).css("background-color", "yellow");  
    },  
    function() {  
        $(this).css("background-color", "blue");  
    }  
);
```

Kolor zmienia się na żółty przy najechaniu myszą. Kolor zmienia się na niebieski po opuszczeniu elementu.

9. Metoda submit()

Obsługuje zdarzenie związane z przesłaniem formularza.

Składnia:

```
$(element).submit(handler);
```

- **handler:** Funkcja, która zostanie wykonana, gdy formularz zostanie przesłany.

Przykład:

```
$("#form").submit(function(event) {  
    alert("Formularz został przesłany!");  
    event.preventDefault(); // Zapobiega rzeczywistemu przesłaniu formularza  
});
```

10. Metoda scroll()

Obsługuje zdarzenie przewijania okna lub elementu, gdy użytkownik przewija stronę lub określony kontener.

Składnia:

```
$(element).scroll(handler);
```

- **handler:** Funkcja, która zostanie wykonana, gdy nastąpi przewijanie.

Przykład:

```
$(window).scroll(function() {  
    console.log("Strona jest przewijana.");  
});
```

11. Metoda focus()

Obsługuje zdarzenie ustawienia fokusu na danym elemencie,

Składnia:

```
$(element).focus(handler);
```

- **handler:** Funkcja, która zostanie wykonana, gdy element otrzyma fokus.

Przykład:

```
$("#input").focus(function() {  
    $(this).css("background-color", "lightblue");  
});
```

12. Metoda keyup()

Obsługuje zdarzenie zwolnienia klawisza na klawiaturze, czyli moment, gdy użytkownik puści wciśnięty klawisz.

Składnia:

```
$(element).keyup(handler);
```

- **handler:** Funkcja, która zostanie wykonana, gdy użytkownik zwolni klawisz.

Przykład:

```
$("#input").keyup(function() {  
    console.log("Zwolniono klawisz!");  
});
```

13. Metoda keydown()

Obsługuje zdarzenie naciśnięcia klawisza na klawiaturze. Wykonywana jest w momencie wciśnięcia dowolnego klawisza, zanim zostanie on zwolniony.

Składnia:

```
$(element).keydown(handler);
```

- **handler:** Funkcja, która zostanie wykonana, gdy użytkownik wciśnie klawisz.

Przykład:

```
$(document).keydown(function(event) {  
    console.log("Wciśnięto klawisz: " + event.key);  
});
```

14. Metoda `resize()`

Obsługuje zdarzenie zmiany rozmiaru okna przeglądarki. Wykorzystywana do dynamicznego dostosowania układu strony w zależności od rozmiaru okna.

Składnia:

```
$(window).resize(handler);
```

- **handler:** Funkcja, która zostanie wykonana, gdy zmieni się rozmiar okna.

Przykład:

```
$(window).resize(function() {  
    console.log("Zmieniono rozmiar okna.");  
});
```

15. Metoda `on()`

Metoda `on()` służy do przypisywania obsługi jednego lub wielu zdarzeń do elementu. Jest to zalecany sposób przypisywania zdarzeń (może zastępować `bind()`, `click()`, `dblclick()` i inne starsze metody).

Składnia:

```
$(element).on(eventType, selector, data, handler);
```

- **eventType:** Typ zdarzenia (np. `click`, `blur`, `change`), które chcemy obsłużyć.

Przykład:

```
$("button").on("click", function(){ alert("Kliknięto przycisk!"); });
```

16. Metoda `off()`

Metoda `off()` służy do usuwania wcześniej przypisanych zdarzeń. Można jej użyć, aby usunąć pojedyncze zdarzenie, grupę zdarzeń, wszystkie zdarzenia lub zdarzenia przypisane do określonego selektora.

Składnia:

```
$(element).off(eventType, selector, handler);
```

- **eventType:** Typ zdarzenia (np. `click`, `blur`, `change`), które chcemy obsłużyć.

Przykład:

```
$("button").off("click");
```

17. Metoda trigger()

Ręcznie wywołuje zdarzenie na elemencie, symulując interakcje użytkownika.

Składnia:

```
$(element).trigger(eventType, extraParameters);
```

Przykład:

```
$("button").trigger("click");
```

Efekty

Efekty jQuery umożliwiają tworzenie animacji i efektów wizualnych na stronach internetowych.

1. Metoda animate()

Służy do tworzenia animacji CSS na wybranych elementach. Można ją użyć do płynnej zmiany wartości wielu właściwości CSS, takich jak wysokość, szerokość, opacity (nieprzezroczystość) i inne.

Składnia:

```
$(element).animate(properties, duration, easing);
```

- **properties:** obiekt definiujący właściwości CSS do animacji.
- **duration:** czas trwania animacji w milisekundach lub słowo kluczowe ("slow", "fast").

Przykład:

```
$("div").animate({ width: "300px", opacity: 0.5 }, "slow");
```

2. Metoda fadeIn()

Metoda fadeIn() służy do płynnego pojawiania się elementu na stronie przez zwiększanie jego opacity.

Składnia:

```
$(element).fadeIn(duration);
```

- **duration:** Czas trwania animacji.

Przykład:

```
$("div").fadeIn("slow");
```

3. Metoda fadeOut()

Płynnie ukrywa element przez stopniowe zmniejszanie jego opacity.

Składnia:

```
$(element).fadeOut(duration);
```

- **duration:** Czas trwania animacji.

Przykład:

```
$("div").fadeOut("fast");
```

4. Metoda hide()

Metoda hide() ukrywa wybrany element, ustawiając jego właściwość display na "none".

Składnia:

```
$(element).hide(duration);
```

Przykład:

```
$("p").hide();
```

5. Metoda show()

Służy do płynnego pokazywania ukrytego elementu, poprzez zmianę jego display.

Składnia:

```
$(element).show(duration);
```

Przykład:

```
$("p").show("slow");
```

6. Metoda slideUp()

Służy do animacji elementu, ukrywając go w górę przez zmniejszanie jego wysokości.

Składnia:

```
$(element).slideUp(duration);
```

Przykład:

```
$("div").slideUp("fast");
```

7. Metoda toggle()

Przełącza między pokazaniem i ukryciem elementu w zależności od jego bieżącego stanu.

Składnia:

```
$(element).toggle(duration, complete);
```

Przykład:

```
$("div").toggle("slow");
```


8. Metoda `fadeToggle()`

Przełącza między płynnym pokazaniem i ukryciem elementu, w zależności od jego bieżącego stanu.

Składnia:

```
$(element).fadeToggle(duration);
```

Przykład:

```
$("div").fadeToggle("fast");
```

9. Metoda `delay()`

Wstrzymuje wykonanie następnej animacji w kolejce na określoną ilość czasu.

Składnia:

```
$(element).delay(duration);
```

- **duration:** Czas opóźnienia w milisekundach.

Przykład:

```
$("div").delay(1000).fadeIn();
```

10. Metoda `stop()`

Zatrzymuje animacje biegnące w danym momencie.

Składnia:

```
$(element).stop(clearQueue, jumpToEnd);
```

Przykład:

```
$("div").stop();
```

11. Metoda `finish()`

Metoda `finish()` w jQuery zatrzymuje wszystkie bieżące animacje na wybranym elemencie, natychmiast kończąc je i ustawiając elementy w stan końcowy animacji. Działa podobnie do metody `stop()`, ale dodatkowo czyści wszystkie animacje z kolejki i bezpośrednio kończy wszystkie aktywne animacje.

Składnia:

```
$(element).finish();
```

Przykład:

Jeśli element ma kilka animacji w kolejce, metoda `finish()` zakończy wszystkie bieżące animacje i ustawi element w stanie końcowym każdej z tych animacji:

```
$("div").slideUp(2000).fadeIn(2000);  
$("div").finish(); // Zatrzyma i zakończy wszystkie animacje natychmiast
```

HTML/CSS

Metody jQuery HTML/CSS koncentrują się na manipulowaniu zawartością HTML i stylami CSS elementów za pomocą jQuery.

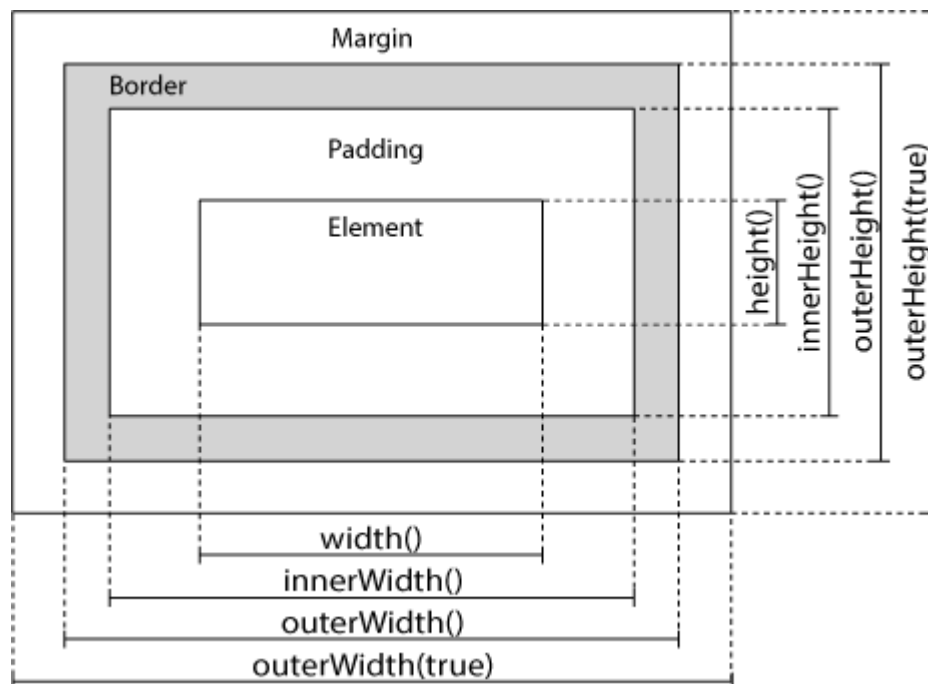
1. Metoda `css()`

Metoda `css()` Pobiera lub ustawia wartości właściwości CSS dla wybranego elementu.

- **Składnia:**
 - Pobranie: `$(element).css(propertyName);`
 - Ustawienie: `$(element).css(propertyName, value);`
- **Argumenty:**
 - `propertyName`: nazwa właściwości CSS.
 - `value`: wartość, jaką chcemy ustawić.
- **Przykład:**

```
$("p").css("color", "blue");
```

2. Metody `width()`, `height()`, `innerWidth()`, `innerHeight()`, `outerWidth()`, `outerHeight()`



`height()` pobiera lub ustawia wysokość wybranego elementu **bez marginesów, obramowań i wypełnień** (padding).

- **Składnia:**
 - Pobranie: `$(element).height();`
 - Ustawienie: `$(element).height(value);`
- **Argumenty:**
 - `value`: nowa wysokość w pikselach (lub w innym formacie jak % dla względnych szerokości).

- **Przykład:**

```
var wysokosc = $("div").height(); // Pobiera szerokość elementu

$("div").height(500); // Ustawia szerokość elementu
```

width() pobiera lub ustawia szerokość wybranego elementu **bez marginesów, obramowań i wypełnień (padding)**.

innerHeight() pobiera wysokość wybranego elementu **razem z wypełnieniami (padding), ale bez marginesów i obramowań**.

innerWidth() pobiera szerokość wybranego elementu **razem z wypełnieniami (padding), ale bez marginesów i obramowań**.

outerHeight() pobiera wysokość wybranego elementu **razem z wypełnieniami (padding) i obramowaniami (border)**. Jeśli podamy argument true, metoda zwróci wysokość **z marginesami (margin)**.

outerWidth() pobiera szerokość wybranego elementu **razem z wypełnieniami (padding) i obramowaniami (border)**. Jeśli podamy argument true, metoda zwróci wysokość **z marginesami (margin)**.

3. Metody text(), html(), val()

Metoda text()

Pobiera lub ustawia **tekst** wybranego elementu (bez interpretacji HTML). Zwraca tylko zawartość tekstową, ignorując jakiegokolwiek tagi HTML wewnątrz elementu.

- **Składnia:**

- Pobranie: `$(element).text();`
- Ustawienie: `$(element).text(textContent);`

- **Argumenty:**

- `textContent`: nowy tekst do ustawienia w elemencie (opcjonalny).

- **Przykład:**

```
// Pobiera tekst z elementu
var tekst = $("p").text();

// Ustawia nowy tekst w elemencie
$("p").text("Nowy tekst");
```

W przypadku elementu `<p>Hello World</p>`, metoda `text()` zwróci "Hello World", ignorując tag ``.

Metoda html()

Pobiera lub ustawia **HTML** wewnątrz wybranego elementu. Pozwala pobrać pełną zawartość HTML, łącznie z tagami, lub ustawić nową zawartość HTML w elemencie.

- **Składnia:**

- Pobranie: `$(element).html();`
- Ustawienie: `$(element).html(htmlContent);`

- **Argumenty:**

- `htmlContent`: nowa zawartość HTML do ustawienia (opcjonalna).

- **Przykład:**

```
// Pobiera zawartość HTML z elementu
var zawartosc = $("div").html();

// Ustawia nową zawartość HTML w elemencie
$("div").html("<p>Nowa zawartość</p>");
```

W przypadku elementu `<div><p>Hello World</p></div>`, metoda `html()` zwróci pełen HTML: `"<p>Hello World</p>"`.

Metoda `val()`

Pobiera lub ustawia **wartość** dla elementów formularzy, takich jak **input**, **textarea** lub **select**. Używana głównie do pracy z polami formularza.

- **Składnia:**

- Pobranie: `$(element).val();`
- Ustawienie: `$(element).val(value);`

- **Argumenty:**

- `value`: nowa wartość do ustawienia w elemencie formularza (opcjonalna).

- **Przykład:**

```
// Pobiera wartość z pola input
var wartosc = $("input").val();

// Ustawia nową wartość w polu input
$("input").val("Nowa wartość");
```

Dla pola `<input type="text" value="Hello">`, metoda `val()` zwróci `"Hello"`.

4. Metody `addClass()`, `removeClass()`, `toggleClass()`, `hasClass()`

Metoda `addClass()`

Dodaje klasę do wybranego elementu.

Przykład:

```
$("p").addClass("highlight");
```

Metoda `removeClass()`

Usuwa klasę z wybranego elementu.

Przykład:

```
$("p").removeClass("highlight");
```

Metoda `toggleClass()`

Przełącza daną klasę (dodaje, jeśli jej nie ma; usuwa, jeśli już jest).

Przykład:

```
$("#p").toggleClass("highlight");
```

Metoda `hasClass()`

Sprawdza, czy element ma określoną klasę. Zwraca true lub false.

Przykład:

```
$("#p").hasClass("highlight")
```

5. Metody `attr()`, `removeAttr()`**Metoda `attr()`**

Służy do pobierania lub ustawiania wartości atrybutów HTML.

Przykład:

```
// Pobranie atrybutu  
var src = $("#img").attr("src");  
  
// Ustawienie atrybutu  
$("#img").attr("src", "nowy-obrazek.jpg");
```

Metoda `removeAttr()`

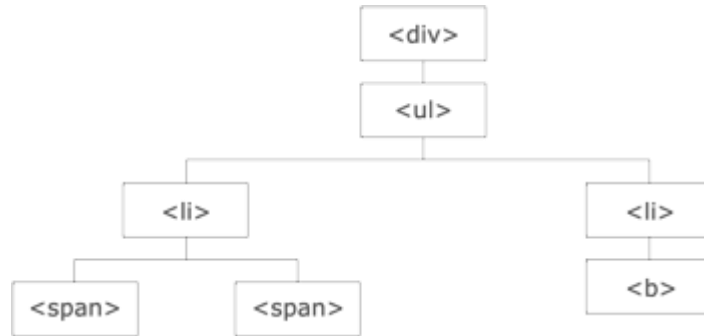
Usuwa atrybut z elementu.

Przykład:

```
$("#img").removeAttr("src");
```

jQuery Traversing

Traversowanie w jQuery oznacza poruszanie się po drzewie DOM, aby znaleźć elementy HTML na podstawie ich relacji z innymi elementami. Można przemieszczać się w górę (do przodków), w dół (do potomków) i na boki (do rodzeństwa).



Drzewo DOM (ang. Document Object Model) to struktura danych reprezentująca dokument HTML. Każdy element w dokumencie (np. tagi HTML, atrybuty, tekst) jest węzłem w drzewie, a relacje między nimi są zorganizowane w formie rodzic-dziecko.

Metoda parent()

Zwraca bezpośredniego rodzica (jeden węzeł) dla wybranego elementu.

```
$('#childElement').parent();
```

Metoda parents()

Zwraca wszystkich przodków (wszystkie węzły rodzicielskie) dla wybranego elementu w formie kolekcji.

```
$('#childElement').parents();
```

Metoda children()

Zwraca bezpośrednie dzieci (wszystkie elementy potomne) dla wybranego elementu.

```
$('#parentElement').children();
```

Metoda find()

Zwraca wszystkie elementy potomne (wszystkie węzły zagnieżdżone w danym elemencie) na dowolnym poziomie głębokości.

```
$('#parentElement').find('div');
```

Metoda siblings()

Zwraca wszystkie rodzeństwa (wszystkie elementy, które mają tego samego rodzica) dla wybranego elementu.

```
$('#element').siblings();
```

Metoda next()

Zwraca następnego bezpośredniego sąsiada (następny węzeł) dla wybranego elementu.

```
$('#element').next();
```

Metoda nextAll()

Zwraca wszystkie elementy, które następują po wybranym elemencie w tym samym poziomie DOM.

```
$('#element').nextAll();
```

Metoda nextUntil()

Zwraca wszystkie elementy, które następują po wybranym elemencie, aż do (ale nie włącznie) elementu określonego przez selektor.

```
$('#startElement').nextUntil('.stopElement');
```

Metoda prev()

Zwraca poprzedniego bezpośredniego sąsiada (poprzedni węzeł) dla wybranego elementu.

```
$('#element').prev();
```

Metoda prevAll()

Zwraca wszystkie elementy, które poprzedzają wybrany element w tym samym poziomie DOM.

```
$('#element').prevAll();
```

Metoda prevUntil()

Zwraca wszystkie elementy, które poprzedzają wybrany element, aż do (ale nie włącznie) elementu określonego przez selektor.

```
$('#endElement').prevUntil('.stopElement');
```

Metoda first()

Zwraca pierwszy element z kolekcji wybranych elementów.

```
$('.items').first();
```

Metoda last()

Zwraca ostatni element z kolekcji wybranych elementów.

```
$('.items').last();
```

Metoda eq()

Zwraca element o określonym indeksie z kolekcji wybranych elementów.

```
$('.items').eq(1);
```

Metoda filter()

Zwraca elementy, które pasują do określonego selektora z kolekcji wybranych elementów.

```
$('.items').filter('.active');
```

Metoda not()

Zwraca elementy, które **nie** pasują do określonego selektora z kolekcji wybranych elementów.

```
$('.items').not('.inactive');
```