Kurs Front-End Developer

Biblioteka jQuery

BIBLIOTEKA JQUERY

`jQuery` to lekka `biblioteka JavaScript`, która znacznie upraszcza programowanie.

Zamiast wielu lini w czystym JavaScript zastępujemy kod czasami nawet w jednej linii kodu w jQuery

jQuery upraszcza również np. wywołania AJAX i manipulację DOM.

Porównanie składni `jQuery` z czystym `JavaScript` na wybranych przykładach można znaleźć pod adresem:

http://youmightnotneedjquery.com/

DODANIE BIBLIOTEKI DO PROJEKTU

Aby korzystać z biblioteki jQuery w projekcie należy najpierw dołączyć ją do niego.

Można to zrobić na dwa sposoby:

- ściągnąć plik z biblioteką i dodać go lokalnie;
- odwołać się do wersji znajdującej się w sieci za pomocą adresu url.

DODANIE BIBLIOTEKI DO PROJEKTU

Ze strony http://jquery.com/download/ można ściągnąć plik z biblioteką na dysk do katalogu, w którym znajduje się projekt i dołączyć go tak jak inne pliki z rozszerzeniem .js.

```
<script type="text/javascript" src="jquery-3.1.1.min.js"></script>
```

jquery wrzucamy w HTMLu przed skryptem JS

DODANIE BIBLIOTEKI DO PROJEKTU

Najłatwiej przejść na stronę https://cdnjs.com/, wpisać jQuery i po wyszukaniu skopiować adres url.

```
<script type="text/javascript" src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
```

DOCUMENT READY (I-***)

jQuery działa na strukturze DOM. Aby mieć pewność, że dokument został w całości załadowany należy skorzystać z `metody document.ready().`

```
$( document ).ready( function() {

    // kod do wykonania
} );
```

lub w skróconej wersji

```
$( function() {

    // kod do wykonania
} );
```

ZNAK $ I SKŁADNIA ŁAŃCUCHOWA (I-***)

JQuery ma bardzo zwięzłą konstrukcję skryptów, do której wykorzystuje składnię łańcuchową.

Każdy taki łańcuch rozpoczyna się znakiem $, a poszczególne polecenia są łączone w łańcuchy za pomocą kropki.

```
$( '#first' )
      .css( { background : 'red' } )
      .hide( 'slow' )
      .show( 3000 );
```

ZNAK $ I SKŁADNIA ŁAŃCUCHOWA (I-***)

`Znak dolara($)` jest skrótowym wywołaniem funkcji jQuery.

Zwraca obiekt jQuery dziedziczący po obiekcie DOM'u rozszerzający jego funkcje o możliwości jQuery. Funkcja ta przyjmuje za parametr string lub obiekt DOM'u.

```
var jakisElement = $( "#idElementu" );      // identyczny efekt co: jQuery( "#idElementu" )
var jakisElement2 = $( ".nazwaKlasy" );     // identyczny efekt co: jQuery( ".nazwaKlasy" )
var jakisElement3 = document.getElementById( "idElementu" ); //czysty JS
```

SELEKTORY JQUERY (2-***)

`Selektory jQuery` służą do wskazania pojedynczego elementu lub kolekcji elementów w dokumencie HTML.

Za pomocą selektorów jQuery można wskazać element po jego nazwie, ID, klasie, typie atrybutów, wartości atrybutów i wiele więcej.

Wszystkie selektory w jQuery rozpoczynają się od znaku dolara i nawiasów: `$()` .

SELEKTORY JQUERY (2-***)

`Selektor elementu` wybiera elementy na podstawie `nazwy elementu`.

```
var allP = $( "p" );
```

`Selektor ID` wybiera elementy na podstawie atrybutu `ID elementu`. Aby znaleźć element o określonym id, należy przed id elementu HTML umieścić znak hash (#).

```
var idElement = $( "#firstSection" );
```

SELEKTORY JQUERY – SELEKTOR KLASY (2-***)

`Selektor klasy` wybiera elementy o podanej nazwie klasy. Aby znaleźć elementy o określonej klasie, należy przed nazwą klasy elementu HTML umieścić znak kropki (.).

```
var allGreen = $( ".green" );
```

SELEKTORY JQUERY – INNE SELEKTORY (2-***)

```
var allElements = $( "*" );                    // wszystkie elementy na stronie
var thisElement = $( this );                   // bieżący element HTML
var elementClass = $( "p.green" );             // wszystkie elementy p z class="green"
var elementFirst = $( "p:first" );             // pierwszy element p
var elementsAttr = $( "[href]" );              // wszystkie elementy z atrybutem href
var elementsTitle = $( "[title='Image']" );    // wszystkie elementy z atrybutem title='Image'
```

Więcej selektorów:

https://api.jquery.com/category/selectors/

EDYCJA HTML I CSS (3-***)

Dzięki jQuery można w łatwy sposób `manipulować HTML i CSS`.

Można zmieniać zawartość dowolnego elementu, dodawać w wybranym przez nas miejscu fragment kodu HTML, kopiować elementy i dowolnie manipulować obiektami na stronie.

EDYCJA HTML I CSS (3-***)

Metoda `text()`:

```
$( selektor ).text();    // zwraca zawartość tekstową ze wszystkich dopasowanych selektorów
$( selektor ).text( "tekst" );// ustawia tekst we wszystkich dopasowanych selektorach
```

np.

```
$( "p" ).text( "Hello world!" );
```

EDYCJA HTML I CSS (3-***)

Metoda `html()`:

```
$( selektor ).html();                  // zwraca zawartość z pierwszego dopasowanego selektora
$( selektor ).html( "kodHTML" );  // ustawia kod HTML we wszystkich dopasowanych selektorach
```

np.

```
$( "p" ).html( "Hello <b>world</b>!" );
```

EDYCJA HTML I CSS (3-***)

Metody dodające nową treść:

`append()` - wstaw zawartość na końcu wybranego selektora
`prepend()` - wstaw zawartość na początku wybranego selektora
`after()` - wstaw zawartość za wybranym selektorem
`before()` - wstaw zawartość przed wybranym selektorem

np.

```
$( "p.green" ).append( "Hello world!" );
```

EDYCJA HTML I CSS (3-***)

Metody usuwające treść i elementy:

`remove()` - usuwa wybrany element i elementy znajdujące się wewnątrz tego elementu
`empty()` - usuwa elementy znajdujące się wewnątrz wybranego elementu

np.

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

EDYCJA HTML I CSS (3-***)

Metoda `css()`:

```
$( selektor ).css( "wlasnoscCSS" );       // zwraca określoną własność CSS z pierwszego dopasowanego selektora
$( selektor ).css( "wlasnoscCSS", "wartosc" ); // ustawia własność CSS we wszystkich dopasowanych selektorach
```

np.

```
$( "p" ).css( "color" ); // zwróci kolor tekstu pierwszego elementu p

$( "p" ).css( "color", "red" );  // ustawi kolor tekstu wszystkim elementom p
```

EDYCJA HTML I CSS (3-***)

Metoda `val()`:

```
$( selektor ).val();      // zwraca wartość z pierwszego dopasowanego pola formularza
$( selektor ).val( "tekst" ); // ustawia wartość do wszystkich dopasowanych pól formularza
```

np.

```
$( "input#name" ).val();
```

```
$( "input#email" ).val( "name@gmail.com" );
```

EDYCJA HTML I CSS (3-***)

Metody dodające i usuwające klasy:

```
addClass();     // dodaje jedną lub więcej klas do wybranego elementu
removeClass();     // usuwa jedną lub więcej klas z wybranego elementu
```

np.

```
$( "div" ).addClass( "important" );

$( "p" ).removeClass( "green", "important" );
```

EDYCJA HTML I CSS (3-***)

Więcej metod do manipulacji HTML i CSS:

https://api.jquery.com/category/manipulation/

PORUSZANIE SIĘ PO DOM (4-***)

Biblioteka jQuery oferuje również metody pozwalające się z łatwością poruszać po drzewie HTML.

Do takich metod należą `metoda each()` oraz `metoda find()`.

PORUSZANIE SIĘ PO DOM (4-***)

Metoda `find()` zwraca elementy potomne wybranego elementu. Metoda przechodzi całe drzewo HTML od góry do dołu i wyszukuje odpowiednich elementów.

Metoda posiada wymagany parametr, który mówi jakich potomków danego elementu wyszukać.

Ogólna konstrukcja:

```
$( selector ).find( filter );
```

```
$( "div" ).find( ".first" ).css( { "color": "red", "border-color": "red" } );
```

Znajdź potomków elementu `<div>` o `klasie first` i nadaj im kolor tekstu i kolor obramowania czerwony.

PORUSZANIE SIĘ PO DOM (4-***)

Metoda `each()` określa funkcję, która ma być uruchomiona dla wszystkich znalezionych elementów.

Funkcja wymaga parametru w postaci funkcji, która ma być uruchomiona dla znalezionych elementów.

Ogólna konstrukcja:

```
$( selector ).each( function() {
    // kod do wykonania
} );
```

PORUSZANIE SIĘ PO DOM (4-***)

```
$( "button" ).click( function() {
    $( "li" ).each( function() {
        console.log( $( this ).text() );
    } );
} );
```

Po kliknięciu na element `<button>` uruchamiana jest funkcja dla wszystkich elementów `<li>`, która wypisuje w konsoli tekst znajdujący się w elementach `<li>`.

PORUSZANIE SIĘ PO DOM (4-***)

Więcej metod do manipulacji HTML i CSS:

https://api.jquery.com/category/traversing/

ZDARZENIA JQUERY (5-***)

Większość zdarzeń obsługiwanych przez JavaScript DOM ma równoważną metodę jQuery.

Aby skorzystać ze zdarzenia w jQuery wystarczy przypiąć zdarzenie do pobranego obiektu korzystając z metody będącej nazwą danego zdarzenia.

ZDARZENIA JQUERY (5-***)

```
$( 'a.guzik' ).click( function() {
    console.log( 'Guzik został naciśnięty.' );
} );
```

lub

```
$( 'a.guzik' ).on( 'click', function() {
    console.log( 'Guzik został naciśnięty.' );
} );
```

ZDARZENIA JQUERY (5-***)

Podpinanie kilku zdarzeń:

```
$( 'a.guzik' ).on( {
      'click':   function() {

            // kod do wykonania
      },
      'mouseover':      function() {

            // kod do wykonania
      }
} );
```

ZDARZENIA JQUERY (5-***)

Wybrane zdarzenia jQuery:

`click()` - gdy użytkownik kliknie na wskazany element
`dblclick()` - gdy użytkownik dwukrotnie kliknie na wskazany element
`mouseenter()` - gdy wskaźnik myszy wejdzie na wskazany element
`mouseleave()` - gdy wskaźnik myszy opuści wskazany element
`resize()` - gdy zmieniany jest rozmiar wskazanego elementu
`load()` - gdy wybrany element jest ładowany
`scroll()` - gdy użytkownik skroluje wskazany element
`submit()` - gdy formularz jest wysyłany

ZDARZENIA JQUERY (5-***)

Więcej zdarzeń jQuery:

https://api.jquery.com/category/events/

ANIMACJE JQUERY (6-***)

Za pomocą jQuery można wykonywać `animacje na elementach.`

Można korzystać z metod zdefiniowanych w jQuery, ale również można tworzyć własne animacje.

ANIMACJE JQUERY (6-***)

Za pomocą metod `hide()` i `show()` można ukrywać i pokazywać elementy. Metoda `hide()` służy do ukrywania elementów, natomiast metoda `show()` do ich pokazywania.

Obie metody mogą przyjmować dwa parametry. Pierwszym z nich jest prędkość ukrywania/pojawiania się elementu.

Drugim jest funkcja, która jest wykonywana gdy metody się zakończyły.

ANIMACJE JQUERY (6-***)

```
$( "#hide" ).click( function() {
    $( "p" ).hide( "slow" );
} );
```

```
$( "#show" ).click( function() {
    $( "p" ).show( "fast" );
} );
```

ANIMACJE JQUERY (6-***)

Za pomocą metody `fadeIn()` i `fadeOut()` można płynnie pokazywać i ukrywać elementy na stronie. Metody te przyjmują takie same parametry ja metody `show()` i `hide()`.

```
$( "#hide" ).click( function() {
    $( "p" ).fadeOut( "slow" );
} );

$( "#show" ).click( function() {
    $( "p" ).fadeIn( "fast" );
} );
```

ANIMACJE JQUERY (6-***)

Za pomocą metody `slideUp()` i `slideDown()` można płynnie zwijać elementy w górę i rozwijać w dół. Metody te przyjmują takie same parametry ja metody wymienione wcześniej.

```
$( "#hide" ).click( function() {
    $( "p" ).slideDown( "slow" );
} );

$( "#show" ).click( function() {
    $( "p" ).slideUp( "fast" );
} );
```

ANIMACJE JQUERY (6-***)

Więcej efektów animacji:

https://api.jquery.com/category/effects/

ANIMACJE JQUERY (6-***)

W jQuery można tworzyć niestandardowe animacje.

Ogólna konstrukcja:

```
$( selector ).animate( { params }, speed, callback );
```

`params` - właściwości CSS, które mają być animowane
`speed` - szybkość animacji
`callback` - funkcja która wykona się po zakończeniu animacji

ANIMACJE JQUERY (6-***)

```
$( "button" ).click( function() {
        $( "div" ).animate( { left: '250px' } );
} );

$( "button" ).click( function() {
        $( "div" ).animate( {
                left: '250px',
                opacity: '0.5',
                height: '150px'
        } );
} );

$( "button" ).click( function() {
         $( "div" ).animate( { left: '100px' }, "slow" );
         $( "div" ).animate( { fontSize: '3em' }, "slow" );
} );
```

WARSZTATY – Formularz zamawiania Pizzy

Stwórz formularz zamówienia Pizzy z następującymi polami:

- Imie
- Nazwisko
- Ulica
- Nr domu/mieszkania
- Kod Pocztowy
- Miasto
- Pizza (do wyboru 5 rodzajów Pizzy – każda w innej cenie) - pole to ma ustawioną domyślną wartość "-- Wybierz Pizzę --". Po wyborze zamawianej pizzy, powinna wyświetlić się jej cena (np. jako tekst w paragrafie)
- 2 pola wyboru sosów (pomidorowy i czosnkowy) (najlepiej jako checkbox'y)
- Zgoda na przetwarzanie danych – checkbox (musi być zgoda na relizacje zamówienia)

Za pomocą jQuery zrób walidację formularza. Walidator ma sprawdzać:

- czy wszystkie pola zostały wypełnione? Jeśli nie, to zwracać odpowiedni komunikat
- czy została wybrana Pizza przy zamówieniu? Jeśli nie, to zwracany odpowiedni komunikat

Na końcu walidator ma stworzyć obiekt zamówienia np. w formacie JSON i wyświetlić go w konsoli

Akademia 108

https://akademia108.pl