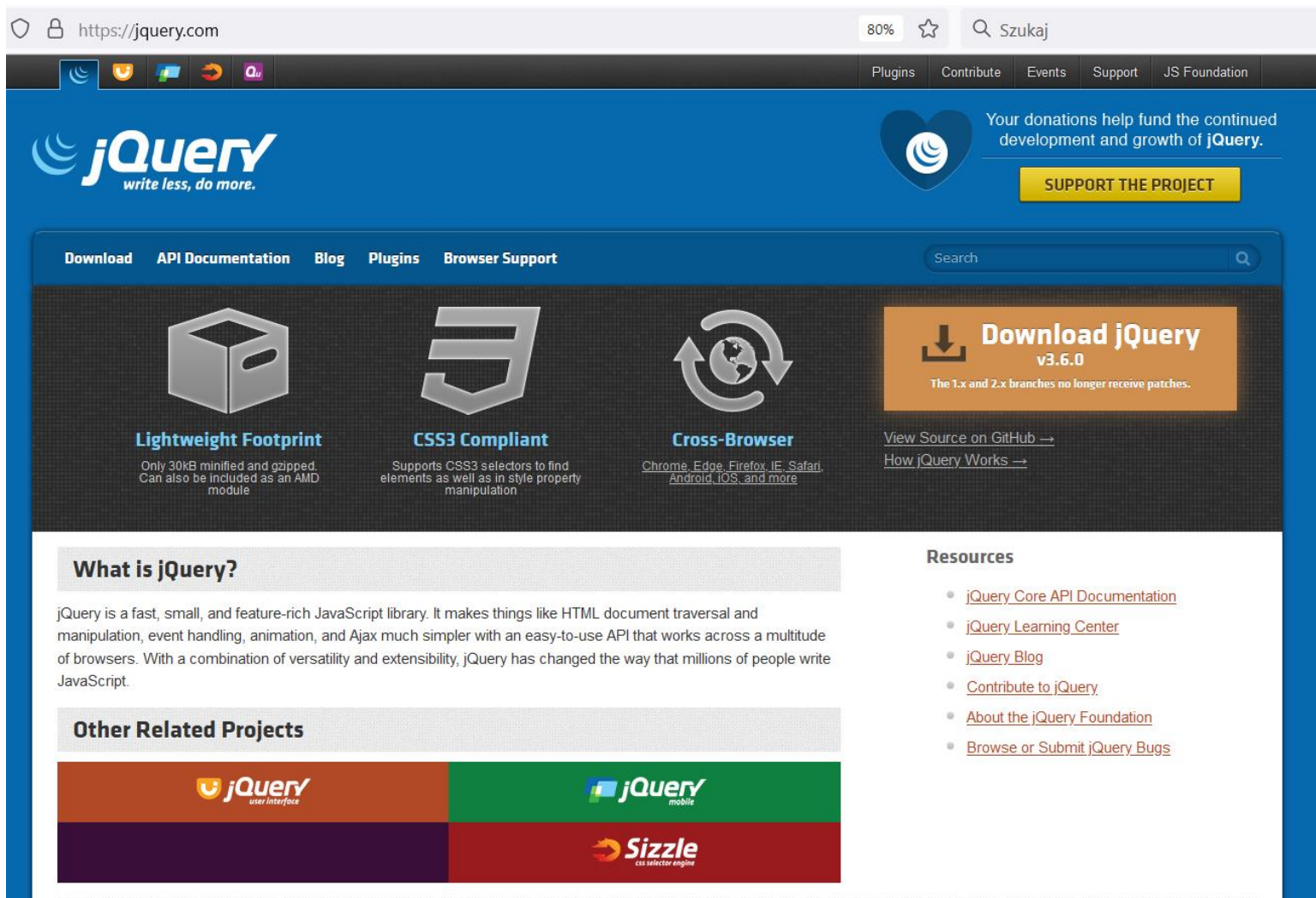


jQuery

jQuery

jQuery jest biblioteką napisaną w języku JavaScript, powstała w celu ułatwienia stosowania tego języka na stronie

write less, do more



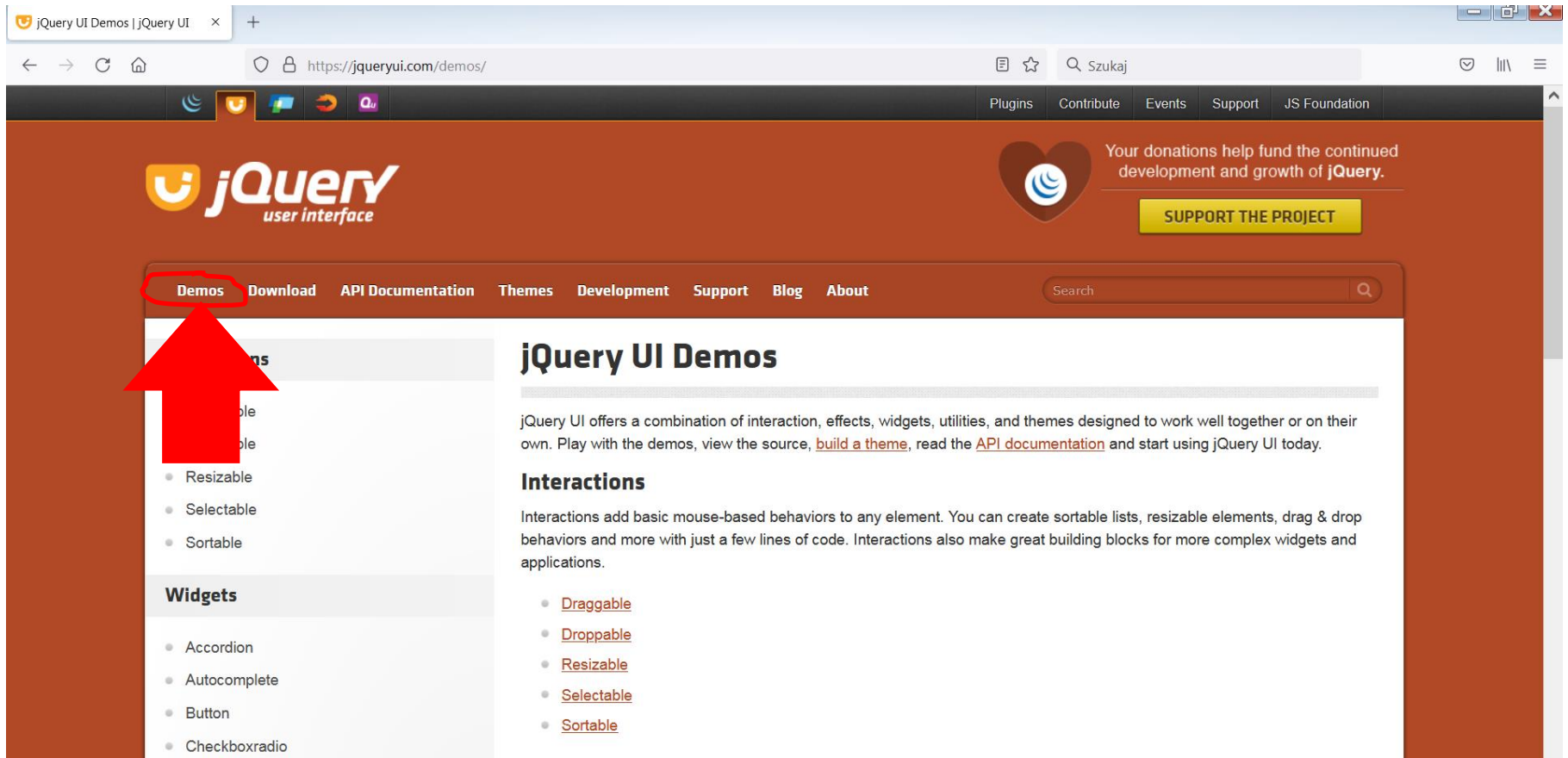
The screenshot shows the jQuery website homepage. At the top, there's a navigation bar with links for Plugins, Contribute, Events, Support, and JS Foundation. The main header features the jQuery logo with the tagline "write less, do more." and a call to action "SUPPORT THE PROJECT". Below the header, there's a navigation menu with links for Download, API Documentation, Blog, Plugins, and Browser Support. The main content area is divided into three columns: "Lightweight Footprint" (30KB minified and gzipped), "CSS3 Compliant" (supports CSS3 selectors), and "Cross-Browser" (supports Chrome, Edge, Firefox, IE, Safari, Android, iOS, etc.). A prominent "Download jQuery v3.6.0" button is on the right. Below this, there's a "What is jQuery?" section, a "Resources" section with links to documentation, learning center, blog, and bug reports, and an "Other Related Projects" section featuring jQuery user interface, jQuery mobile, and Sizzle CSS selector engine.

[w3schools tutorial](#)

[Khan Academy tutorial](#)

<https://jquery.com/>

jQuery demo



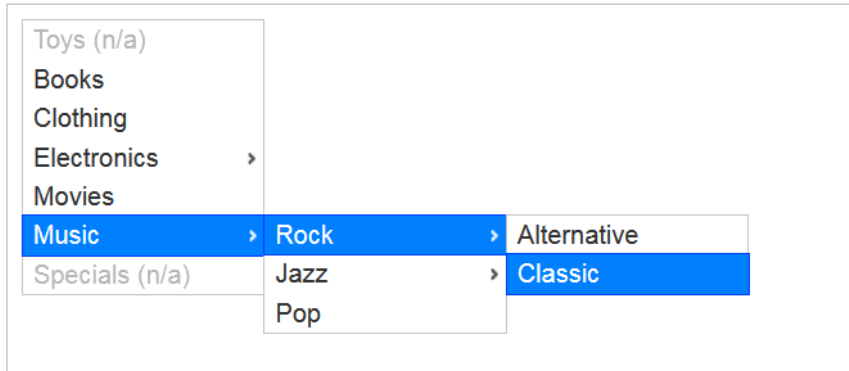
przegląd możliwości biblioteki

<https://jqueryui.com/>

jQuery demo

Menu

Themeable menu with mouse and keyboard interactions for navigation.

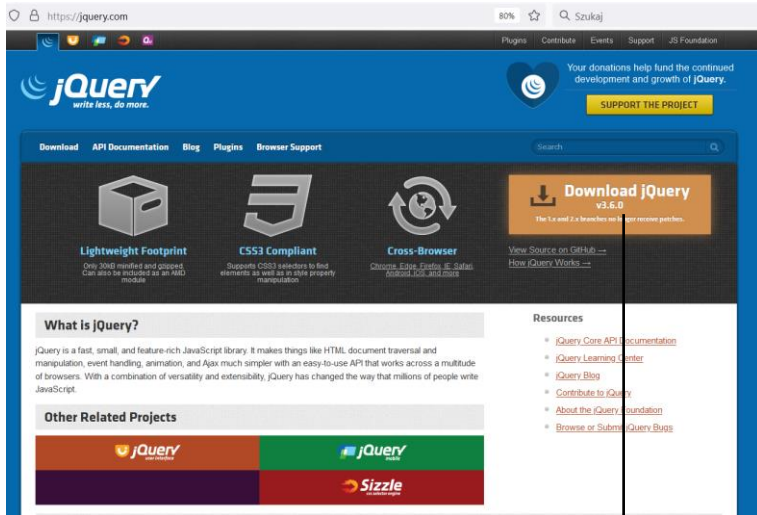


można obejrzeć źródło skryptu

 [view source](#)

```
1 <!doctype html>
2 <html lang="en">
3 <head>
4   <meta charset="utf-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <title>jQuery UI Menu - Default functionality</title>
7   <link rel="stylesheet" href="//code.jquery.com/ui/1.13.0/themes/base/jquery-ui.css">
8   <link rel="stylesheet" href="/resources/demos/style.css">
9   <script src="https://code.jquery.com/jquery-3.6.0.js"></script>
10  <script src="https://code.jquery.com/ui/1.13.0/jquery-ui.js"></script>
11  <script>
12    $( function() {
13      $( "#menu" ).menu();
14    } );
15  </script>
16  <style>
17    .ui-menu { width: 150px; }
18  </style>
19 </head>
20 <body>
```

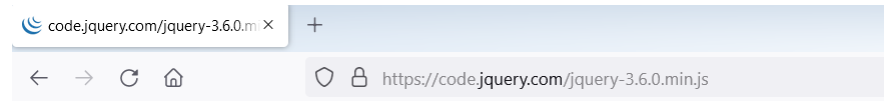
jQuery download



<https://jquery.com/>

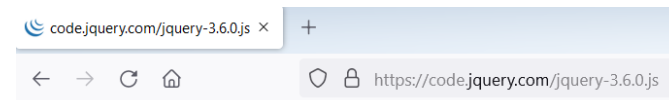
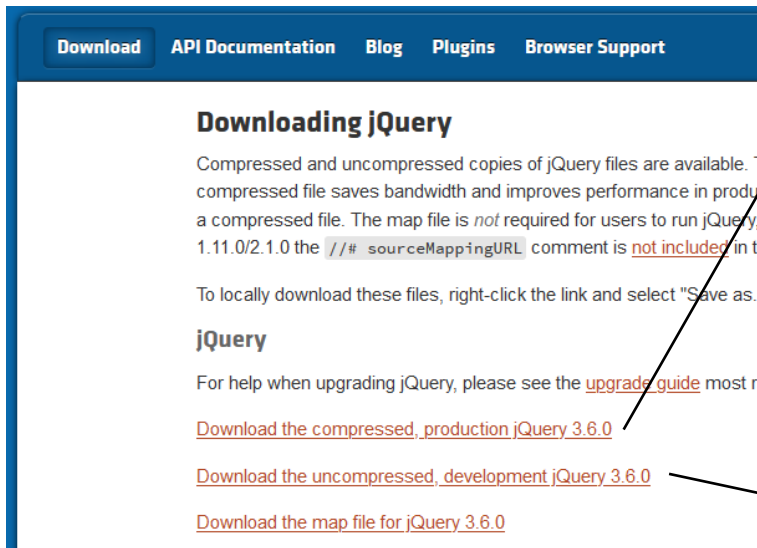
wersja produkcyjna
mniejszy plik, brak znaków białych i komentarzy

jquery-3.6.0.min.js



```
/*! jQuery v3.6.0 | (c) OpenJS Foundation and other contributors | jquery.org/license */
!function(e,t){...}

```



```
/*!
 * jQuery JavaScript Library v3.6.0
 * https://jquery.com/
 *
 * Includes Sizzle.js
 * https://sizzlejs.com/
 *
 * Copyright OpenJS Foundation and other contributors
 * Released under the MIT license
 * https://jquery.org/license
 *
 * Date: 2021-03-02T17:08Z
 */
( function( global, factory ) {

```

jquery-3.6.0.js

wersja deweloperska
większy plik, czytelniejsza zawartość

jQuery API



[Download](#) [API Documentation](#) [Blog](#) [Plugins](#) [Browser Support](#)

Search

- Ajax
 - Global Ajax Event Handlers
 - Helper Functions
 - Low-Level Interface
 - Shorthand Methods
- Attributes
- Callbacks Object
- Core
- CSS
- Data
- Deferred Object
- Deprecated
 - Deprecated 1.3
 - Deprecated 1.7
 - Deprecated 1.8
 - Deprecated 1.9
 - Deprecated 1.10
 - Deprecated 3.0
 - Deprecated 3.2

jQuery API

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. If you're new to jQuery, we recommend that you check out the [jQuery Learning Center](#).

If you're updating to a newer version of jQuery, be sure to read the release notes published on [our blog](#). If you're coming from a version prior 1.9, you should check out the [1.9 Upgrade Guide](#) as well.

Note that this is the API documentation for jQuery core. Other projects have API docs in other locations:

- [jQuery UI API docs](#)
- [jQuery Mobile API docs](#)
- [QUnit API docs](#)

.add()

Traversing > Miscellaneous Traversing

Create a new jQuery object with elements added to the set of matched elements.

.addBack()

Traversing > Miscellaneous Traversing

Add the previous set of elements on the stack to the current set, optionally filtered by a selector.

.addClass()

Attributes | Manipulation > Class Attribute | CSS

Adds the specified class(es) to each element in the set of matched elements.

dokumentacja biblioteki

<https://api.jquery.com/>

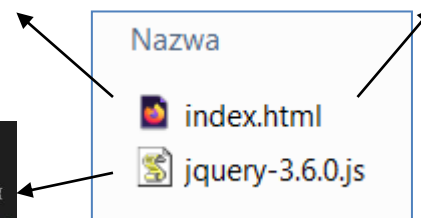
Umiejscowienie skryptu

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
  <script src="jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function(){
      $("p").click(function(){
        $(this).hide();
      });
    });
  </script>
</head>
<body>
  <p>jQuery</p>
</body>
</html>
```

lub

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
</head>
<body>
  <p>jQuery</p>
  <script src="jquery-3.6.0.js"></script>
  <script>
    $(document).ready(function(){
      $("p").click(function(){
        $(this).hide();
      });
    });
  </script>
</body>
</html>
```

```
( function( global, factory ) {
  "use strict";
  if ( typeof module === "object" && typeof module.exports === "object" ) {
    // For CommonJS and CommonJS-like environments where a proper "window"
    // is present, execute the factory and get jQuery.
    // For environments that do not have a "window" with a "document"
    // (such as Node.js), expose a factory as module.exports.
    // This accentuates the need for the creation of a real "window".
    // e.g. var jQuery = require("jquery")(window);
    // See ticket #14549 for more info.
    module.exports = global.document ?
      factory( global, true ) :
```



folder

Umieszczenie skryptu w zewnętrznym pliku

skrypt umieszczony w sekcji HEAD

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
  <script src="jquery-3.6.0.js"></script>
  <script src="znikanie.js"></script>
</head>
<body>
  <p>jQuery</p>
</body>
</html>
```

skrypt umieszczony w sekcji BODY

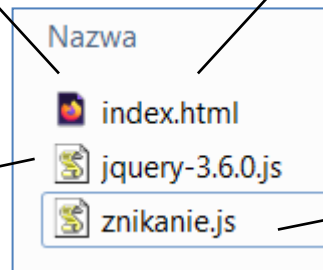
```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
</head>
<body>
  <p>jQuery</p>
  <script src="jquery-3.6.0.js"></script>
  <script src="znikanie.js"></script>
</body>
</html>
```

lub

preferowany sposób

```
( function( global, factory ) {
  "use strict";

  if ( typeof module === "object" && typeof module.exports === "object" ) {
    // For CommonJS and CommonJS-like environments where a proper "window"
    // is present, execute the factory and get jQuery.
    // For environments that do not have a "window" with a "document"
    // (such as Node.js), expose a factory as module.exports.
    // This accentuates the need for the creation of a real "window".
    // e.g. var jQuery = require("jquery")(window);
    // See ticket #14549 for more info.
    module.exports = global.document ?
      factory( global, true ) :
```



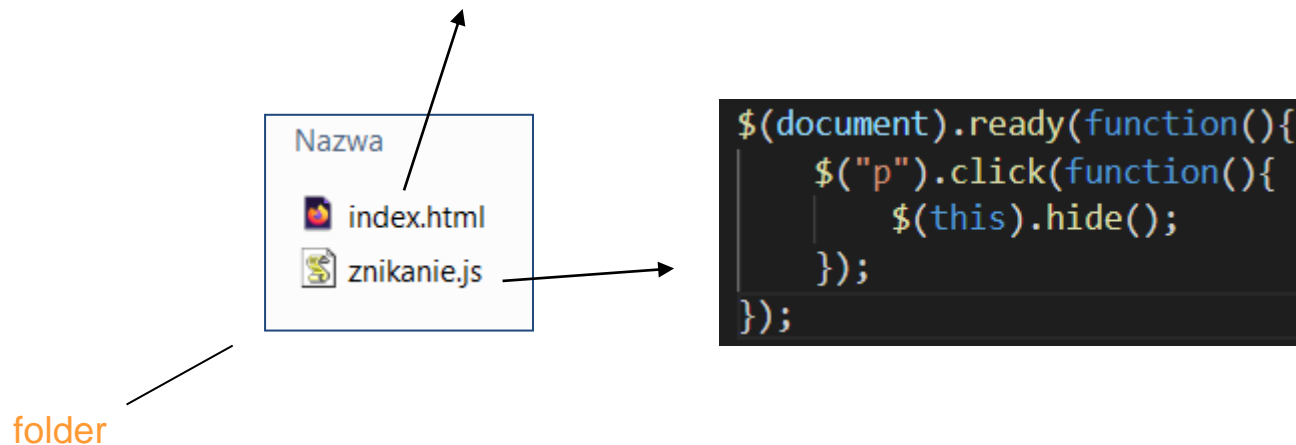
```
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
```

folder

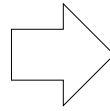
Użycie serwerów CDN (Content Delivery Network)

adres serwera CDN

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>jQuery</title>
</head>
<body>
  <p>jQuery</p>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="znikanie.js"></script>
</body>
</html>
```



działanie skryptu „znikanie.js”



jQuery



click

składnia

składnia jQuery jest dostosowana do wybierania elementów HTML (selector) i wykonywania na nich pewnych działań (action).

[jQuery API](#)

`$(selector).action()`

identyfikator
zmiennej

wybrany element HTML
(np. przez selektor stylu
CSS)

akcja do wykonania
na wybranym elemencie

```
$(this).hide() // chowa bieżący element.  
$("p").show() // pokazuje wszystkie elementy <p>  
$(".menu").show() // pokazuje elementy należące do class="menu"  
$("#left").hide() // chowa elementy z id="left".
```

przykłady
prawidłowej składni

Źródło:

https://www.w3schools.com/jquery/jquery_syntax.asp

Zdarzenie (event) Document Ready

metodę jQuery umieszcza się w ciele zdarzenia Document Ready, aby wykonała się po pełnym załadowaniu strony (strona musi być gotowa, aby skrypt jQuery zaczął działać)

```
$(document).ready(function(){
    $("p").click(function(){
        $(this).hide();
    });
});
```

metoda jQuery

gdy strona w pełni się załaduje skrypt jest gotowy do działania (kliknięcie na element <p> powoduje jego zniknięcie ze strony)

Zdarzenie (event) Document Ready

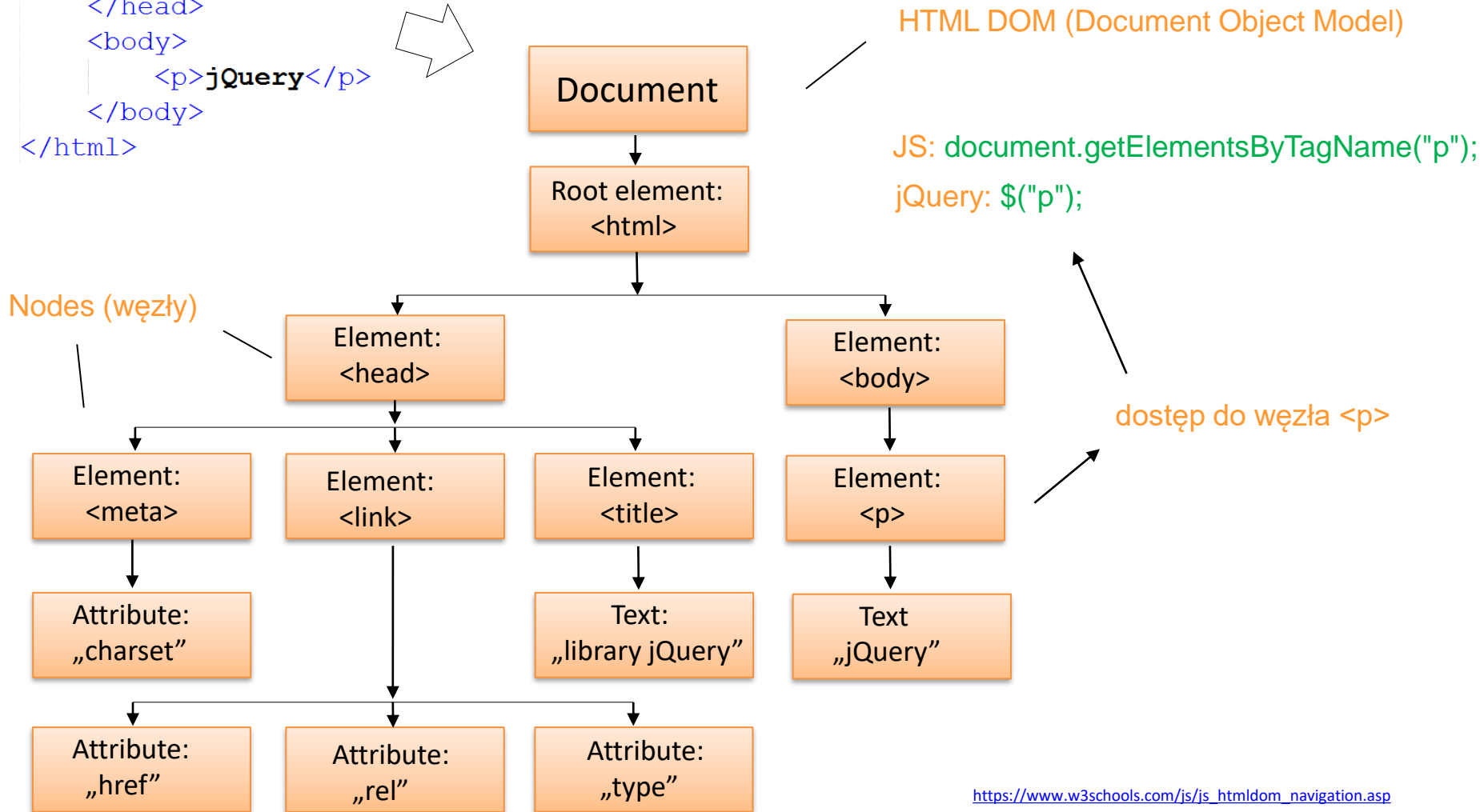
krótszy zapis

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

```
$(function(){  
});
```

Selektory – wybór elementu HTML

```
<html>
  <head>
    <meta charset="utf-8">
    <link href="style.css" rel="stylesheet" type="text/css">
    <title>library jQuery</title>
  </head>
  <body>
    <p>jQuery</p>
  </body>
</html>
```



Selektory – wybór elementu HTML

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link href="style.css" rel="stylesheet" type="text/css">
    <title>library jQuery</title>
  </head>
  <body>
    <p>jQuery</p>
    <script>
      const myElement = document.getElementsByTagName("p");
      myElement[0].textContent = "JavaScript";
    </script>
  </body>
</html>
```

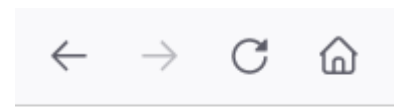
zamiana treści znacznika <p> za pomocą JS



JavaScript

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8">
    <link href="style.css" rel="stylesheet" type="text/css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
    <title>library jQuery</title>
  </head>
  <body>
    <p>jQuery</p>
    <script>
      $(document).ready(function() {
        var myElement = $("p");
        myElement.text("jQuery jest OK");
      });
    </script>
  </body>
</html>
```

zamiana treści znacznika <p> za pomocą jQuery



jQuery jest OK

Selektory – wybór elementu HTML

<code>\$("*")</code>	wszystkie elementy
<code>\$(this)</code>	bieżący element
<code>\$("#stopka")</code>	element z id="stopka"
<code>\$(".header")</code>	wszystkie elementy należące do klasy class="header"
<code>\$(".p.czerwone")</code>	wszystkie znaczniki <p> należące do klasy class="czerwone"
<code>\$("p:first")</code>	pierwszy znacznik <p> na stronie
<code>\$("ol li:first")</code>	pierwszy element listy pierwszego znacznika na stronie
<code>\$("ol li:first-child")</code>	pierwszy element listy każdego znacznika na stronie
<code>\$("[src]")</code>	wszystkie elementy z atrybutem "src"
<code>\$("a[target='_blank']")</code>	wszystkie linki <a> z atrybutem target="_blank"
<code>\$("a[target!='_blank']")</code>	wszystkie linki <a> z atrybutem target różnym od "_blank"
<code>\$(":button")</code>	wszystkie elementy <button> oraz elementy <input> z typem type="button"
<code>\$("tr:even")</code>	parzyste elementy <tr>
<code>\$("tr:odd")</code>	nieparzyste elementy <tr>

[tester selektorów](#)

Selektory – wybór elementu HTML

użycie wyrażeń regularnych

```
$("a").attr("href", "https://jquery.com/"); //wybiera każdy link na stronie i zamienia adres na jquery
$('[href="http://www.onet.pl"]').attr("href", "https://jquery.com/"); // wybiera adres onetu i zamienia na adres jquery
$('[href!="http://www.onet.pl"]').attr("href", "https://jquery.com/"); // każdy inny adres niż onetu zamienia na adres jquery
$('[href^="http://www.o"]').attr("href", "https://jquery.com/"); // wybiera adresy rozpoczynające się na http://www.o"
// (czyli onetu) i zamienia na adres jquery
$('[href$="pl"]').attr("href", "https://jquery.com/"); // wybiera adresy kończące się na "pl" (czyli onetu i wp)
// i zamienia na adres jquery
$('[href*="oo"]').attr("href", "https://jquery.com/"); // wybiera adresy zawierające ciąg tekstowy "oo"
// (czyli google) i zamienia na adres jquery
$('[title~="polecam"]').text("ulubione linki"); // wybiera elementy z atrybutem title, w którym na liście słów
// znajduje się słowo "polecam" i zamienia tekst
// (ulubione adresy -> ulubione linki)
```

```
<p title="linki, które polecam">ulubione adresy:</p>
<a href="http://www.onet.pl">onet</a><br>
<a href="http://www.wp.pl">wp</a><br>
<a href="http://www.google.com">google</a>
```

Selektory – wybór elementu HTML

Podmieniaacz linków

www.google.pl

www.onet.pl

www.wp.pl

Zmień linki



Nowe linki

jquery.com

angular.io

www.nowoczesnaszkola.edu.pl

Zmień linki

css - zmiana atrybutu elementu

zmiana koloru i rodziny
czcionek na całej stronie

```
<script>
$(document).ready(function(){
    $("body").css("color", "red");
    $("body").css("font-family", "Arial, Helvetica, sans-serif");
});
</script>
```

lub

```
<script>
$(document).ready(function(){
    $("body").css({"color": "red",
    "font-family": "Arial, Helvetica, sans-serif"});
});
</script>
</head>
```

css – odczytanie wartości atrybutu elementu

odczytanie koloru tła prostokąta
po kliknięciu na nim

```
$(document).ready(function(){
  $("#cube").on("click", function() {
    const bgColor = $(this).css("background-color");
    $("#info").html("kolor obiektu: " + bgColor);
  });

  $("#btn").click(function(){
    $("#cube").css("backgroundColor", "#00ff00");
  });
});
```



zmiana koloru

kolor obiektu: rgb(255, 0, 0)

Selektory – wybór elementu HTML

zmiana koloru linku,
usunięcie podkreślenia
oraz zmiana atrybutu href

```
<script>
$(document).ready(function(){
    $("a").css({"color": "orange",
               "text-decoration": "none"});
    $("a").attr("href", "http://www.google.pl");
});
</script>
```

zmiana pierwszego
na stronie

```
<ul>
  <li>Janek</li>
  <li>Ania</li>
  <li>Włodek</li>
  <li>Zygmunt</li>
  <li>Basia</li>
</ul>
```

```
<script>
$(document).ready(function(){
    $( "li" ).first().css( "background-color", "yellow" );
});
</script>
```

```
<ul>
  <li>Janek</li>
  <li>Ania</li>
  <li>Włodek</li>
  <li>Zygmunt</li>
  <li>Basia</li>
</ul>
```

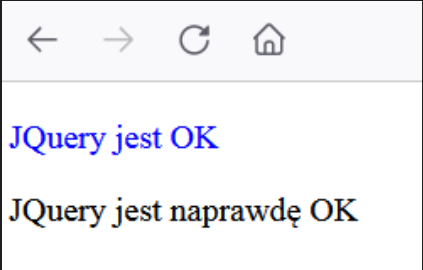
- Janek
- Ania
- Włodek
- Zygmunt
- Basia

- Janek
- Ania
- Włodek
- Zygmunt
- Basia

addClass, removeClass

```
<!DOCTYPE html>
<html>
<head>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <style>
    .czerwony {
      color: red;
    }
    .niebieski {
      color: blue;
    }
    .zielony {
      color: green;
    }
  </style>
</head>
<body>
  <p class="czerwony">jQuery jest OK</p>
  <p class="zielony">jQuery jest naprawdę OK</p>
  <script>
    $(document).ready(function(){
      $(".czerwony").addClass('niebieski');
      $(".zielony").removeClass('zielony');
    });
  </script>
</body>
</html>
```

wykorzystanie metod:
addClass(),
removeClass()



addClass, removeClass

element bez klasy

jquery

add class red

add class green

remove red class

remove green class

element po dodaniu klasy red

jquery

add class red

add class green

remove red class

remove green class


Selektory – wybór elementu HTML

<code>.first()</code>	wybiera pierwszy element
<code>.last()</code>	wybiera ostatni element
<code>.odd()</code>	wybiera nieparzyste elementy, elementy numerowane są od zera
<code>.even()</code>	wybiera parzyste elementy, elementy numerowane są od zera
<code>.eq(index)</code>	wybiera element o podanym indeksie
<code>.not(selektor)</code>	wybiera wszystkie elementy z wyjątkiem selektora
<code>.find(selektor)</code>	szuka selektora w danym elemencie
<code>.filter(kryterium)</code>	wybiera z listy elementów te, które pasują do kryterium

<https://api.jquery.com/>

filter


```
<script>
$(document).ready(function(){
  $( "li" ).filter( "#kolega" ).css("color", "red" );
});
</script>
```



metoda filter() wybiera elementy wg podanego selektora – elementy, dla których id = kolega

```
<ul>
  <li id="kolega">Janek</li>
  <li>Ania</li>
  <li id="kolega">Włodek</li>
  <li id="kolega">Zygmunt</li>
  <li>Basia</li>
</ul>

<ul>
  <li>Janek</li>
  <li>Ania</li>
  <li>Włodek</li>
  <li>Zygmunt</li>
  <li>Basia</li>
</ul>
```




- Janek
- Ania
- Włodek
- Zygmunt
- Basia


- Janek
- Ania
- Włodek
- Zygmunt
- Basia

not

```
<script>
$(document).ready(function(){
  $( "li" ).not( "#kolega" ).css("color", "red" );
});
</script>
```




metoda not() wybiera
wszystkie znaczniki
oprócz tych z id=„kolega”



```
<ul>
  <li id="kolega">Janek</li>
  <li>Ania</li>
  <li id="kolega">Włodek</li>
  <li id="kolega">Zygmunt</li>
  <li>Basia</li>
</ul>

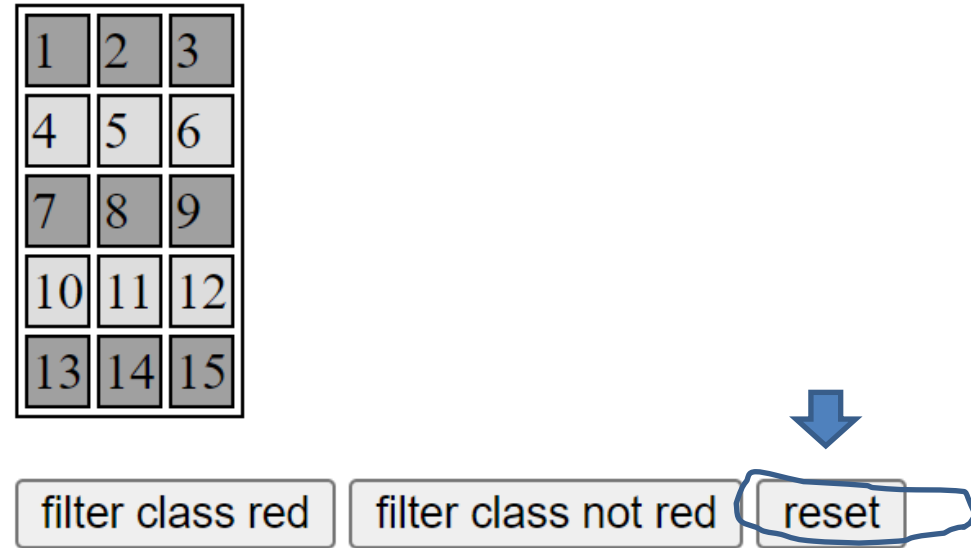
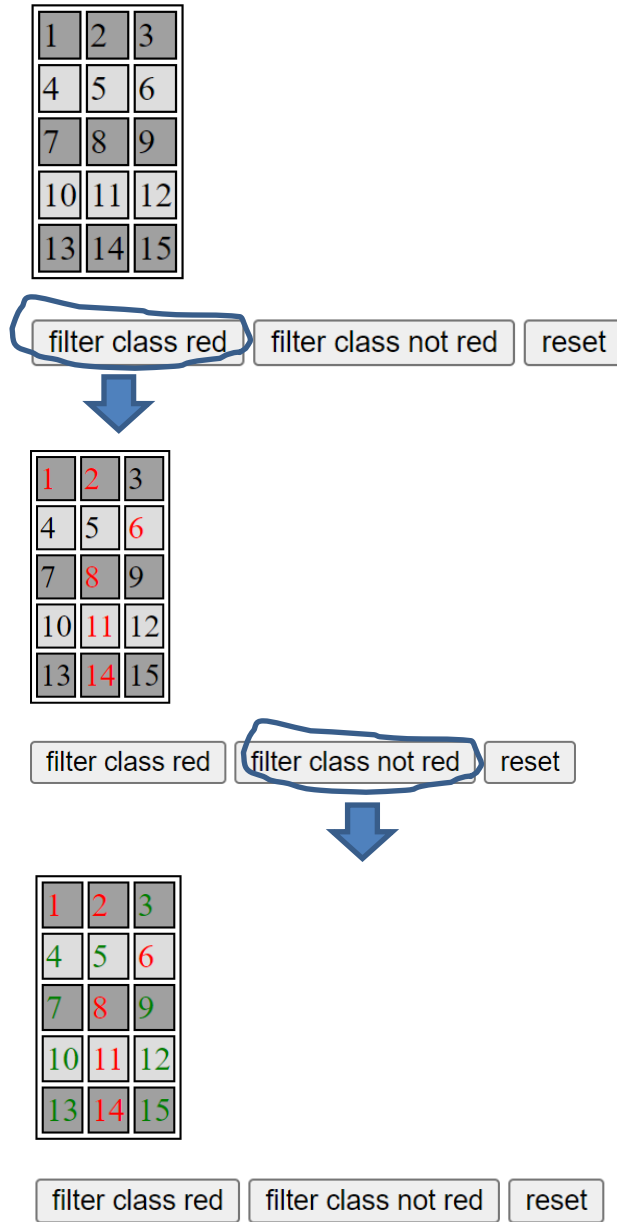
<ul>
  <li>Janek</li>
  <li>Ania</li>
  <li>Włodek</li>
  <li>Zygmunt</li>
  <li>Basia</li>
</ul>
```



- Janek
- Ania
- Włodek
- Zygmunt
- Basia

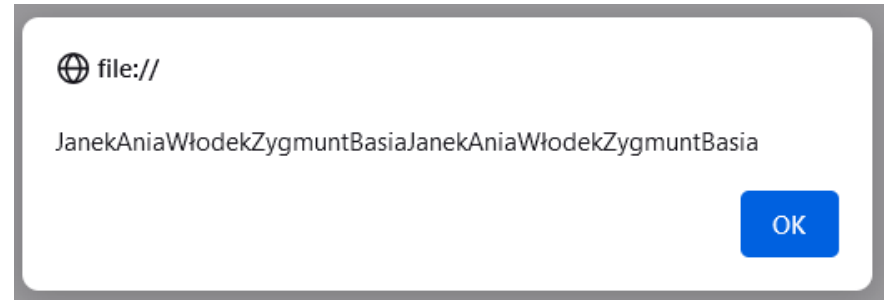
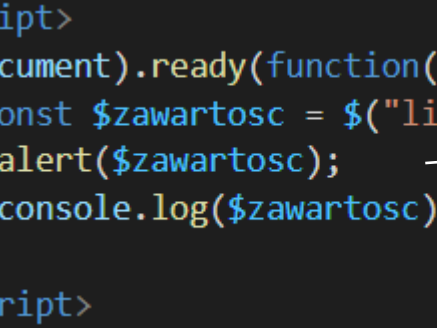
- Janek
- Ania
- Włodek
- Zygmunt
- Basia

filter, not



text()

```
<script>
$(document).ready(function(){
  const $zawartosc = $("li").text();
  alert($zawartosc);
  console.log($zawartosc);
});
</script>
```



```
<ul>
  <li id="kolega">Janek</li>
  <li>Ania</li>
  <li id="kolega">Włodek</li>
  <li id="kolega">Zygmunt</li>
  <li>Basia</li>
</ul>

<ul>
  <li>Janek</li>
  <li>Ania</li>
  <li>Włodek</li>
  <li>Zygmunt</li>
  <li>Basia</li>
</ul>
```

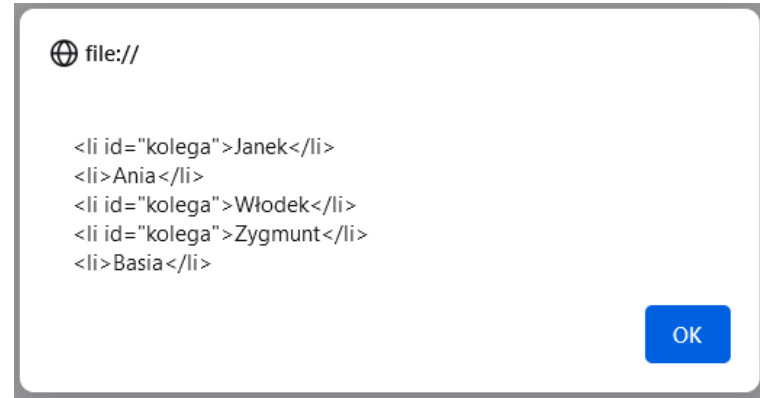
JanekAniaWłodekZygmuntBasiaJanekAniaWłodekZygmuntBasia

metoda text() wybiera
(lub ustawia) zawartość
tekstową elementu –
wszystkich na stronie

- Janek
 - Ania
 - Włodek
 - Zygmunt
 - Basia
-
- Janek
 - Ania
 - Włodek
 - Zygmunt
 - Basia

html()

```
<script>
$(document).ready(function(){
  const $zawartosc = $("ul").html();
  alert($zawartosc);
  console.log($zawartosc);
});
</script>
```



```
<ul>
  <li id="kolega">Janek</li>
  <li>Ania</li>
  <li id="kolega">Włodek</li>
  <li id="kolega">Zygmunt</li>
  <li>Basia</li>
</ul>

<ul>
  <li>Janek</li>
  <li>Ania</li>
  <li>Włodek</li>
  <li>Zygmunt</li>
  <li>Basia</li>
</ul>
```

```
<li id="kolega">Janek</li>
<li>Ania</li>
<li id="kolega">Włodek</li>
<li id="kolega">Zygmunt</li>
<li>Basia</li>
```

- Janek
- Ania
- Włodek
- Zygmunt
- Basia

- Janek
- Ania
- Włodek
- Zygmunt
- Basia

metoda html() wybiera kod html elementu – pierwszego na stronie z elementami potomnymi

text(), html()

1	2	3
4	5	6
7	8	9
10	11	12
13	14	15

get text()

get html()



Komunikat z bieżącej strony

1
2
3

4
5
6
7

OK

Komunikat z bieżącej strony

```
<td class="red"> 1 </td>  
<td class="red"> 2 </td>  
<td> 3 </td>
```

OK

text(„tekst”), html(„kod html”)

metoda html() ustawia kod html wszystkich elementów div na stronie

```
$("#div").html("<h1>Nowy, Lepszy Tytuł</h1>");
```

https://www.w3schools.com/jquery/html_html.asp

metoda text() ustawia zawartość tekstową wszystkich elementów <p> na stronie

```
$("#p").text("Nowy, Lepszy tekst");
```

https://www.w3schools.com/jquery/html_text.asp

before,after, prepend,append

<znacznik> zawartość </znacznik>

The diagram shows the text "<znacznik> zawartość </znacznik>". Below this text, there are four arrows pointing upwards. The first arrow points to the opening tag "<znacznik>". The second arrow points to the space between the opening tag and the content "zawartość". The third arrow points to the space between the content and the closing tag "</znacznik>". The fourth arrow points to the closing tag "</znacznik>".

metoda **before**(treść)
wstawia treść (kod HTML) do wybranego elementu przed znacznikiem otwierającym.


metoda **prepend**(treść)
wstawia treść (kod HTML) do wybranego elementu za znacznikiem otwierającym.

metoda **append**(treść)
wstawia treść (kod HTML) do wybranego elementu przed znacznikiem zamykającym.

metoda **after**(treść)
wstawia treść (kod HTML) do wybranego elementu po znaczniku zamykającym.

before,after, prepend,append

```
<script>
$(document).ready(function(){
    $("ul").append("<li>Zenek</li>");
});
</script>
```



metoda append() wstawia do wybranego elementu (przed znacznikiem zamykającym) podaną zawartość.

```
<ul>
  <li id="kolega">Janek</li>
  <li>Ania</li>
  <li id="kolega">Włodek</li>
  <li id="kolega">Zygmunt</li>
  <li>Basia</li>
</ul>

<ul>
  <li>Janek</li>
  <li>Ania</li>
  <li>Włodek</li>
  <li>Zygmunt</li>
  <li>Basia</li>
</ul>
```

- Janek
- Ania
- Włodek
- Zygmunt
- Basia
- Zenek

- Janek
- Ania
- Włodek
- Zygmunt
- Basia
- Zenek

before,after, prepend,append

```
<script>
$(document).ready(function(){
  $("ul").prepend("<li>Zenek</li>");
  $("ul").before("<p>Oto moi koledzy i koleżanki:</p>");
  $("ul").after("<p>Bardzo lubię moich kolegów i koleżanki.</p>");
});
</script>
```



wykorzystanie
metod:
prepend(),
before(),
after()

```
<ul>
  <li id="kolega">Janek</li>
  <li>Ania</li>
  <li id="kolega">Włodek</li>
  <li id="kolega">Zygmunt</li>
  <li>Basia</li>
</ul>

<ul>
  <li>Janek</li>
  <li>Ania</li>
  <li>Włodek</li>
  <li>Zygmunt</li>
  <li>Basia</li>
</ul>
```

Oto moi koledzy i koleżanki:

- Zenek
- Janek
- Ania
- Włodek
- Zygmunt
- Basia

Bardzo lubię moich kolegów i koleżanki.

Oto moi koledzy i koleżanki:

- Zenek
- Janek
- Ania
- Włodek
- Zygmunt
- Basia

Bardzo lubię moich kolegów i koleżanki.



before,after, prepend,append

before

Przed tabelą

prepend

początek	początek	początek
1	2	3
4	5	6
7	8	9
10	11	12
13	14	15
koniec	koniec	koniec

append

Po tabeli

after

before after prepend append reset

Selektory – wybór elementu HTML

```
<body>
  <p >jQuery jest OK</p>
  <p >jQuery jest naprawdę OK</p>
  <button>each()</button>
  <script>
    $('button').click(function(){
      $("p").each(function(){
        alert($(this).text());
      });
    });
  </script>
</body>
```

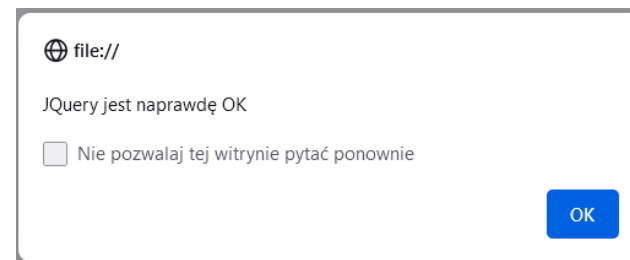
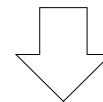
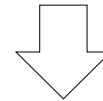
metoda each() wykonuje alert na wszystkich elementach <p> na stronie



jQuery jest OK

jQuery jest naprawdę OK

each()



Zdarzenia

Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

click	dblclick	mouseenter
mouseleave	mousedown	mouseup
hover	on	:)

focus (mouse,tab)	login
blur (loses focus)	password

```
$("#one").click(function(){  
    $("#one").css("background-color", "red");  
});
```


Form Events – blur, focus

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#login").focus(function(){
    $("p").text("wpisz login");
  });
  $("#login").blur(function(){
    $("p").text("Logowanie");
  });
  $("#pass").focus(function(){
    $("p").text("wpisz hasło");
  });
  $("#pass").blur(function(){
    $("p").text("Logowanie");
  });
});
</script>
</head>
<body>
<p>Logowanie</p>
<input id="login" type="text">login<br>
<input id="pass" type="password">password
</body>
</html>
```

Zdarzenie focus
zachodzi gdy pole
tekstowe id=login
jest wybrane

wtedy zamieniamy
komunikat w
paragrafie

Logowanie

wpisz login

wpisz hasło

Form Events – blur, focus

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#login").focus(function(){
    $("p").text("wpisz login");
  });
  $("#login").blur(function(){
    $("p").text("Logowanie");
  });
  $("#pass").focus(function(){
    $("p").text("wpisz hasło");
  });
  $("#pass").blur(function(){
    $("p").text("Logowanie");
  });
});
</script>
</head>
<body>
<p>Logowanie</p>
<input id="login" type="text">login<br>
<input id="pass" type="password">password
</body>
</html>
```

Zdarzenie blur
zachodzi gdy pole
tekstowe id=login
przestaje być
wybrane

wtedy zamieniamy
komunikat w
paragrafie

Logowanie

wpisz login

wpisz hasło

Efekty

Komputer1

hide(2000)

toggle

Komputer2

fadeIn

fadeIn("slow")

fadeIn(2000)

reset

jQuery Effects

jQuery Hide/Show

jQuery Fade

jQuery Slide

jQuery Animate

jQuery stop()

jQuery Callback

jQuery Chaining

[jQuery effects](#)

Animacja

1	2	3
4	5	6
7	8	9

powiększ

pomniejsz

1	2	3
4	5	6
7	8	9

powiększ

pomniejsz

```
$(document).ready(function(){  
  
    $("#powieksz").click(function(){  
        $("#table").animate({  
            width:"+=100px", height:"+=100px", marginLeft:"+=100px"  
        }, 2000);  
    });  
    $("#pomniejsz").click(function(){  
        $("#table").animate({  
            width:"-100px", height:"-100px", marginLeft:"-100px"  
        }, 2000);  
    });  
});
```

formularz

Inf.04. test

1. Którym słowem kluczowym w języku C++ należy posłużyć się, aby przypisać alternatywną nazwę dla istniejącego typu danych?

- union
- typedef
- enum
- switch

2. Podaj imię twórcy języka C++

imię

2. Skrót HTML rozwija się do:

Hiper Text Mark Language ▾

2. Które typy w C++ są zmiennoprzecinkowe?

char int float double

Komunikat z bieżącej strony

Twój wybór:

union

Adam

HiperTextMarkLanguage

char

Komunikat z bieżącej strony

nok

formularz

```
<div id="container">
  <div id="q1">
    <h3>1. Którym słowem kluczowym w języku C++ należy posłużyć się,
    <br>aby przypisać alternatywną nazwę dla istniejącego typu danych?</h3>
    <input type="radio" name="slovoKluczowe" value="union">union<br>
    <input type="radio" name="slovoKluczowe" value="typedef">typedef<br>
    <input type="radio" name="slovoKluczowe" value="enum">enum<br>
    <input type="radio" name="slovoKluczowe" value="switch">switch<br>
  </div>
  <div id="q2">
    <h3>2. Podaj imię twórcy języka C++</h3>
    <input id="imie" type="text">imie<br>
  </div>
  <div id="q3">
    <h3>2. Skrót HTML rozwija się do:</h3>
    <select name="szkola">
      <option value="HiperTextMarkLanguage">Hiper Text Mark Language</option>
      <option value="HightTextMarkupLanguage">Hight Text Markup Language</option>
      <option value="HyperTextMarkupLanguage">HyperText Markup Language</option>
      <option value="HistoryTextMarkLang">History Text Mark Lang</option>
      <option value="inne">inne rozwinięcie</option>
    </select>
  </div>
  <div id="q4">
    <h3>2. Które typy w C++ są zmiennoprzecinlowe?</h3>
    char<input type="checkbox" name="variable" value="char">
    int<input type="checkbox" name="variable" value="int">
    float<input type="checkbox" name="variable" value="float">
    double<input type="checkbox" name="variable" value="double">
  </div>
  <hr>
  <button id="check">check</button>
</div>
```

formularz

```
$(document).ready(function(){
  $("#check").click(function(){

    answer1 = $('input[name=slowoKluczowe]:checked').val();
    answer2 = $("#imie").val();
    answer3 = $('select[name=szkola]').val();
    answer4 = "";
    $.each($('input[name="variable"]:checked'),function(){
      answer4 = answer4 + $(this).val() + " ";
    });
    alert("Twój wybór:\n" + answer1 + "\n" + answer2 + "\n" + answer3 + "\n" + answer4);
    if(answer1 == "typedef" && answer2 == "Bjarne" && answer3 == "HyperTextMarkupLanguage" && answer4 == "float double "){
      alert("ok");
    }else{
      alert("nok");
    }
  });
});
```

pobieramy odpowiedzi

w 4 odpowiedzi sklejamy zaznaczone checkboxy

sprawdzamy odpowiedź i wyświetlamy stosowny alert

walidacja formularza



Form validation with jQuery

HOME

CONTRIBUTE

DOCUMENTATION

REFERENCE DOCUMENTATION

jQuery Validation Plugin

<https://jqueryvalidation.org/>

skrypt do walidacji

walidacja formularza

Ankieta

imię:

This field is required.

nazwisko:

This field is required.

email:

This field is required.

telefon:

krótko o sobie:

This field is required.

wyslij dane na serwer

resetuj ankietę

plugin waliduje poszczególne pola

walidacja formularza

```
<!DOCTYPE html>
<html>
<head>
  <meta charset='utf-8'>
  <title>Walidacja formularza</title>
  <link rel='stylesheet' type='text/css' media='screen' href='main.css'>
</head>
<body>
  <h1>Ankieta</h1>
<hr>
<form id="ankieta" action="" method="post">
  <label>imię:</label><input type="text" name="imię" minlength="2" class="required"><br>
  <label>nazwisko:</label><input type="text" name="nazwisko" minlength="2" class="required"><br>
  <label>email:</label><input type="text" name="email" class="required email" ><br>
  <label>telefon:</label><input type="text" name="telefon" minlength="9" class="number" ><br>

  <h3>krótko o sobie:</h3>
  <textarea name="o_sobie" rows=5 cols=40 minlength="5" class="required"></textarea>

  <hr>
  <input type="submit" value="wyslij dane na serwer">
  <input type="reset" value="resetuj ankietę">
</form>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script src="jquery.validate.js"></script>
<script>
  $("#ankieta").validate();
</script>

</body>
</html>
```

dołączenie skryptów walidacji

walidacja formularza

```
<label>imię:</label><input type="text" name="imię" minlength="2" class="required"><br>
```

imię:

This field is required.

```
<label>nazwisko:</label><input type="text" name="nazwisko" minlength="2" class="required"><br>
```

nazwisko:

Please enter at least 2 characters.

```
<label>email:</label><input type="text" name="email" class="required email" ><br>
```

email:

Please enter a valid email address.

```
<label>telefon:</label><input type="text" name="telefon" minlength="9" class="number" ><br>
```

telefon:

Please enter a valid number.

galeria

Galeria zdjęć z Pienin

Pieniny 1



galeria

```
<!DOCTYPE html>
<html>
<head>
  <meta charset='utf-8'>
  <meta http-equiv='X-UA-Compatible' content='IE=edge'>
  <title>galeria</title>
  <link rel='stylesheet' type='text/css' href='main.css'>
</head>
<body>
  <h1>Galeria zdjęć z Pienin</h1>
  <p></p>
  <p class="content">
    <a href="pictures/zdj1.jpg" title="Pieniny 1"></a>
    <a href="pictures/zdj2.jpg" title="Pieniny 2"></a>
    <a href="pictures/zdj3.jpg" title="Pieniny 3"></a>
    <a href="pictures/zdj4.jpg" title="Pieniny 4"></a>
    <a href="pictures/zdj5.jpg" title="Pieniny 5"></a>
  </p>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
  <script src="main.js"></script>
</body>
</html>
```

jQuery podmienia atrybuty src i alt pokazowego zdjęcia

umiejscowienie pokazowego zdjęcia

galeria

Galeria zdjęć z Pienin

Pieniny 1



tytuł pokazowego zdjęcia

po kliknięciu na miniaturę atrybuty linku do dużego zdjęcia href i title przypisywane są do zmiennych, które z kolei są przypisywane do atrybutów src oraz alt pokazowego zdjęcia.

dodanie znaczników `` w celu wyświetlenia tytułu pokazowego zdjęcia, który jest pobrany z atrybutu `title` dużego zdjęcia

```
$(function(){  
  $("h1").append("<br><br><em></em>");  
  $(".content a").click(function(){  
    let zdjDuzeHref=$(this).attr("href");  
    let zdjDuzeTytul=$(this).attr("title");  
    $("#zdjDuze").attr({src: zdjDuzeHref, alt: zdjDuzeTytul});  
    $("h1 em").html(zdjDuzeTytul);  
    return false;  
  });  
});
```