# JQuery

Prepared by Alex Sobolewski

# Hello!

## My name is Alex Sobolewski

I am working in IT as a Full Stack Software Engineer.
I am strong adherent of Clean Code and good design solutions.
Today i am here with you in order to share my practical knowledge of JQuery, which is a great tool for DOM manipulations.

# 1.
# Introduction to world of frameworks and libraries

# Introduction to world of frameworks and libs

## What is frameworks and libs?

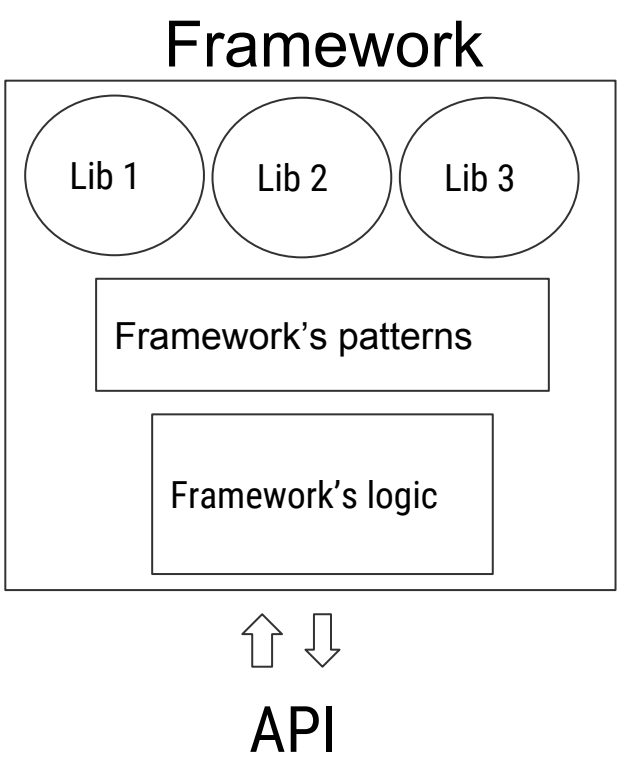
Framework - it is set of design patterns and libraries, that is created to solve strictly defined group of problems(Bootstrap).

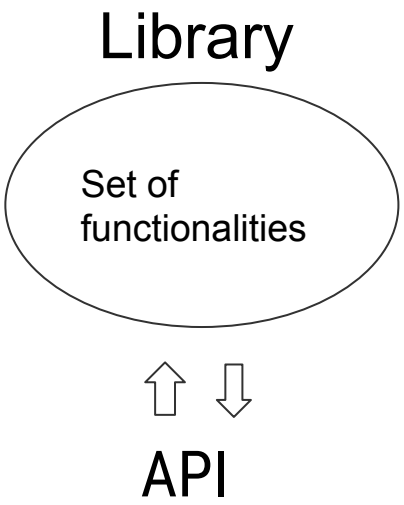
Library - it is a standalone functionality created to solve minor problem or group of minor problems(JQuery).

API - a way to communicate with framework, library of another type of functionality.

# Introduction to world of frameworks and libs

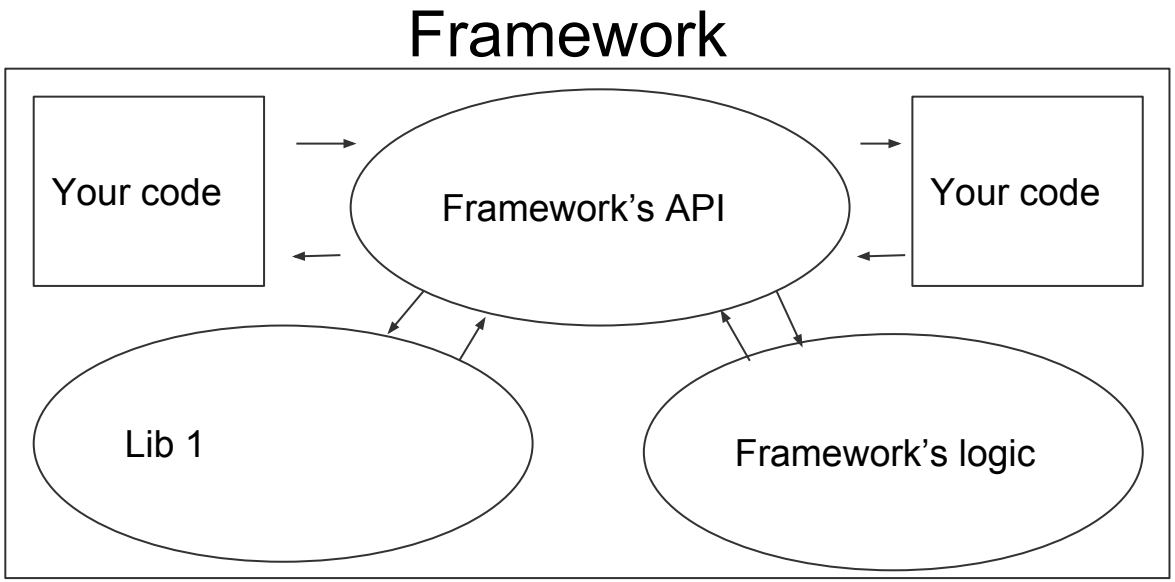
## What is frameworks and libs?



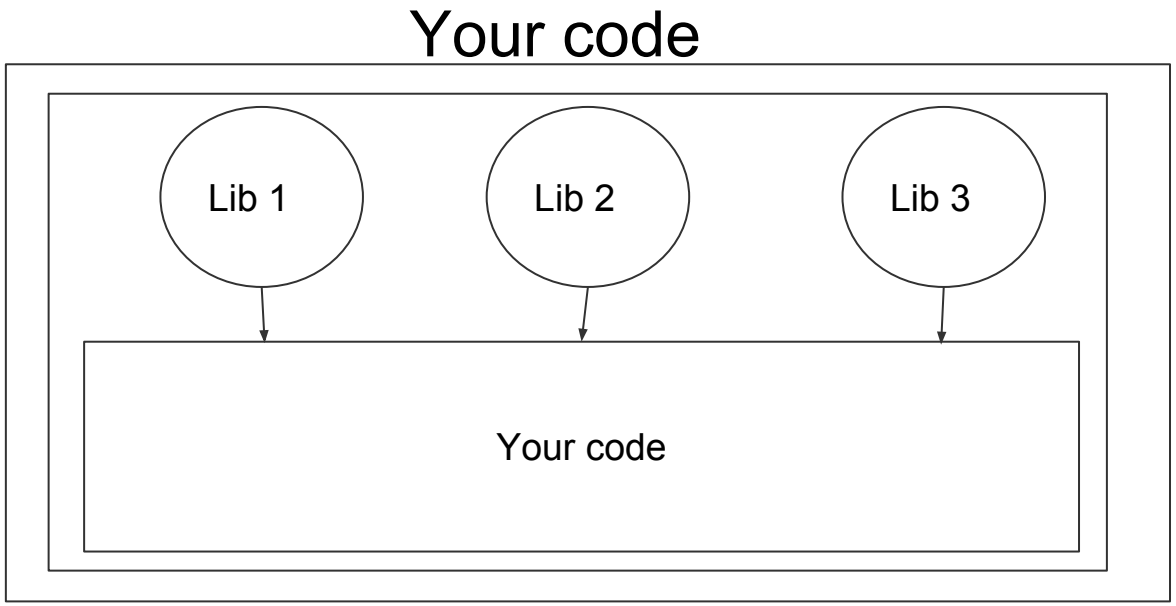
# Introduction to world of frameworks and libs

## What is frameworks and libs?



# Introduction to world of frameworks and libs

## What is frameworks and libs?



# Introduction to world of frameworks and libs

## Where it can be found?

- CSS - Bootstrap, Zurb
- SPA - Angular
- JS - JQuery, React
- Table - DataTables(JQuery used)
- Scheduler - FullCalendar(JQuery based)
- Validation - Form validation(JQuery based)
- Charts - chart.js
- Library of libraries - PrimeNG(composition of plenty other libs)

Development in modern times looks like composing already created blocks and gluing them by your business-logic code.

# Introduction to world of frameworks and libs

## Why frameworks or libs?

- Narrowed to your domain = response to our needs
- Tested = less errors
- Compatible = more ways to use
- Supported by another team = time saving
- Avoids reinventing the wheel
- Boosts development process

All things considered, frameworks and libs are the only way for creating real business solutions in a passable time interval.

# Introduction to world of frameworks and libs

## Do you need to create own frameworks and libs?

Creating of own framework is:

- Time consuming
- Requires deep understanding of your tools
- Requires plenty of time for tests and fixes
- Probably already created by someone else

In modern times, the only reasons for creating own lib or framework are:

- It will be a business product
- You work with legacy code with plenty of repeated actions and you have plenty of time and drive to make things right

# Introduction to world of frameworks and libs

## Summary

- What is framework and libs and API?
- What is the difference between them?
- Give an example of framework
- Why does it matter?
- Why it is better to not to create your own framework?

# 2.
# Introduction to JQuery

# Introduction to JQuery

## What is JQuery?

JQuery - it is a library for dom manipulations, simplification of using animations and AJAX requests.

JQuery is nothing else, that set of tested functionalities with simple API.

JQuery is a foundation for plenty of other libs and frameworks.

JQuery is on the market from 2006. Many of very popular sites use it, Twitter as example.

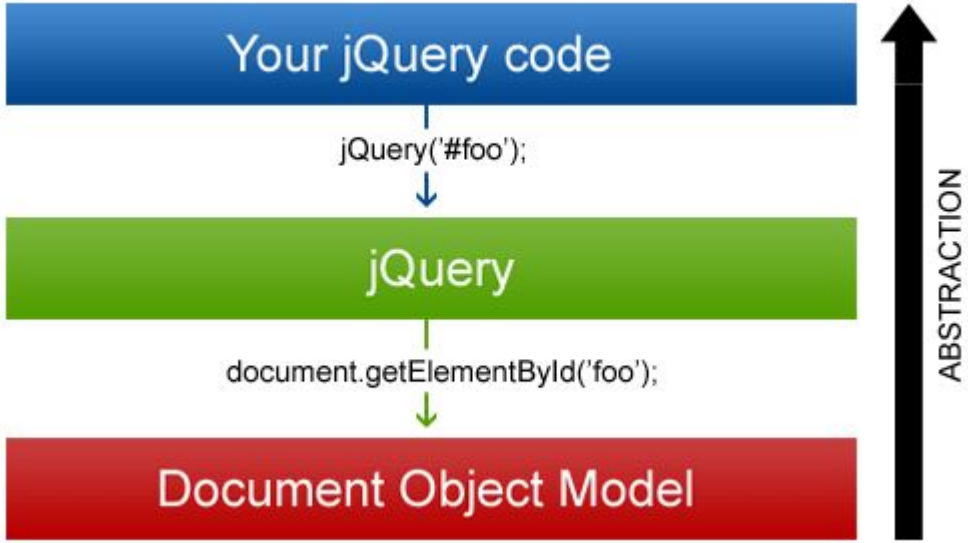
# Introduction to JQuery

## Why is JQuery?

- It has simple API
- It is well documented and has great community
- It has unified API
- It has cross browser compatibility
- It is well tested
- It offers quite rich opportunities for creating interactive and dynamic content

# Introduction to JQuery

## JQuery is just a library



# Introduction to JQuery

## You might not need JQuery

http://youmightnotneedjquery.com/

- JQuery might be redundant in your project
- You can do many things just by pure js in modern times
- It could be that JQuery is not sufficient in your project
- More dependencies == more entropy

# Introduction to JQuery

## Attaching JQuery to the site

- By CDN
- As standalone .js file

Regardless of the way, it is crucial, that JQuery must be placed **before** dependent .js code. **The order is important!**

# Introduction to JQuery

## Difference between minified and full versions

- Minified version does not contain any spaces or long function names or comments(uglify.js)
- Min version has lesser weight than full version, so that min will be downloaded faster(on phones for example)

# Introduction to JQuery

## Full version example

```
// Pass this if window is not defined yet
} )( typeof window !== "undefined" ? window : this, function( window, noGlobal ) {

// Edge <= 12 - 13+, Firefox <=18 - 45+, IE 10 - 11, Safari 5.1 - 9+, iOS 6 - 9.1
// throw exceptions when non-strict code (e.g., ASP.NET 4.5) accesses strict mode
// arguments.callee.caller (trac-13335). But as of jQuery 3.0 (2016), strict mode should be common
// enough that all such attempts are guarded in a try block.
"use strict";

var arr = [];

var document = window.document;

var getProto = Object.getPrototypeOf;

var slice = arr.slice;

var concat = arr.concat;

var push = arr.push;

var indexOf = arr.indexOf;

var class2type = {};

var toString = class2type.toString;
```

# Introduction to JQuery

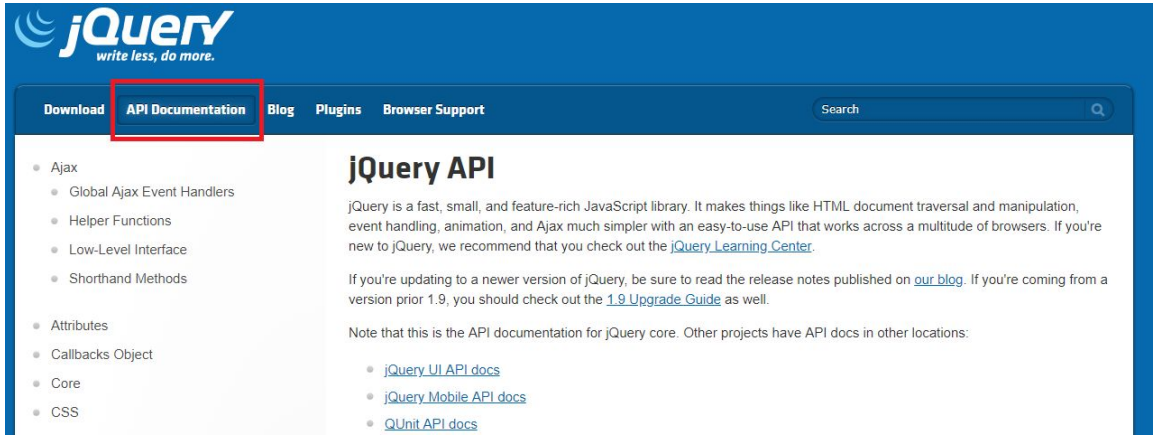
## Min version example

```
/*! jQuery v3.3.1 | (c) JS Foundation and other contributors | jquery.org/license */
!function(e,t){"use strict";"object"==typeof module&&"object"==typeof module.exports?module.expor
t(e)}:t(e)}("undefined"!=typeof window?window:this,function(e,t){"use strict";var n=[],r=e.docume
{},c=l.toString,f=l.hasOwnProperty,p=f.toString,d=p.call(Object),h={},g=function e(t){return"func
{type:!0,src:!0,noModule:!0};function m(e,t,n){var i,o=(t=t||r).createElement("script");if(o.text
e+"":"object"==typeof e||"function"==typeof e?l[c.call(e)]||"object":typeof e}var b="3.3.1",w=fur
{jquery:"3.3.1",constructor:w,length:0,toArray:function(){return o.call(this)},get:function(e){re
t.prevObject=this,t},each:function(e){return w.each(this,e)},map:function(e){return this.pushStac
this.pushStack(o.apply(this,arguments))},first:function(){return this.eq(0)},last:function(){retu
[])},end:function(){return this.prevObject||this.constructor()},push:s,sort:n.sort,splice:n.splic
(l=a,a=arguments[s]||{},s++),"object"==typeof a||g(a)||(a={}),s===u&&(a=this,s--);s<u;s++)if(null
(i=!1,o=n&&Array.isArray(n)?n:[]):o=n&&w.isPlainObject(n)?n:{},a[t]=w.extend(l,o,r)):void 0!==r&&
new Error(e)},noop:function(){},isPlainObject:function(e){var t,n;return!(!e||"[object Object]"!=
(t=i(e))||"function"==typeof(n=f.call(t,"constructor")&&t.constructor)&&p.call(n)===d)},isEmptyOb
{for(n=e.length;r<n;r++)if(!1===t.call(e[r],r,e[r]))break}else for(r in e)if(!1===t.call(e[r],r,e
null!=e&&(C(Object(e))?w.merge(n,"string"==typeof e?[e]:e):s.call(n,e)),n},inArray:function(e,t,r
e.length=i,e},grep:function(e,t,n){for(var r,i=[],o=0,a=e.length,s=!n;o<a;o++)(r=!t(e[o],o))!==s&
(i=t(e[o],o,n))&&s.push(i);else for(o in e)null!=(i=t(e[o],o,n))&&s.push(i);return a.apply([],s)]
Function Array Date RegExp Object Error Symbol".split(" "),function(e,t){l["[object "+t+"]"]=t.to
("array"===n||0===t||"number"==typeof t&&t>0&&t-1 in e)}var E=function(e){var t,n,r,i,o,a,s,u,l,c
(f=!0),0},N={}.hasOwnProperty,A=[],j=A.pop,q=A.push,L=A.push,H=A.slice,O=function(e,t){for(var n=
1},P="checked|selected|async|autofocus|autoplay|controls|defer|disabled|hidden|ismap|loop|multipl
([*^$|!~]?=)"+M+"*(?:'((?:\\\\.|[^\\\\']])*)'|\"((?:\\\\.|[^\\\\\"])*)\"|("+R+"))|)"+M+"*\\]",W=":
```

# Introduction to JQuery

## API documentation

When you work with a library or a framework, it is very important to remember that your best friend is API documentation.



infoShare academy

## *Task 1*

*Attach jQuery to the HTML page.*

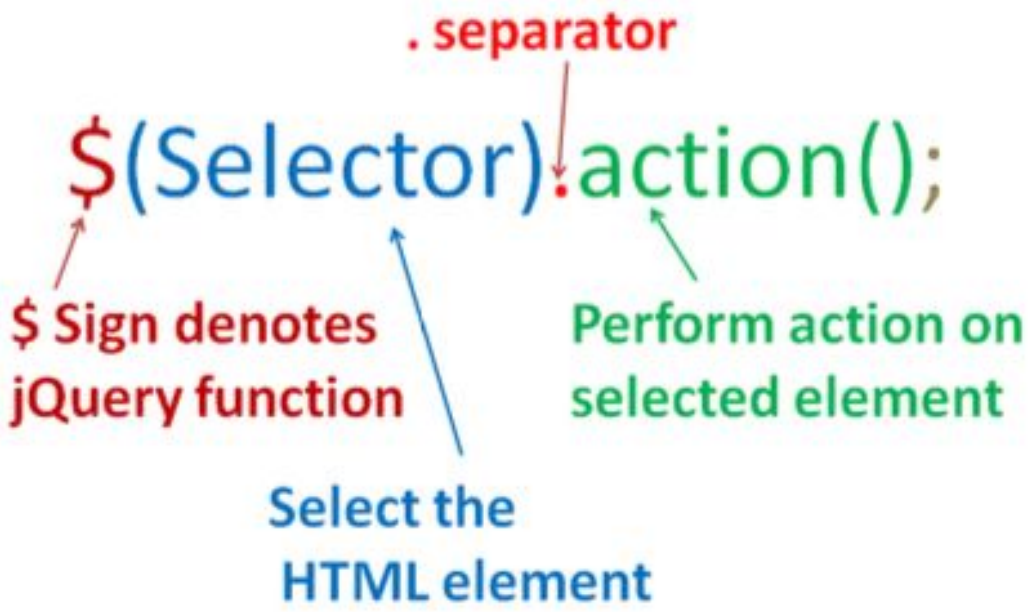
*Check JQuery in the window by window.jQuery*

# Introduction to JQuery

## Task 1 - result

# Introduction to JQuery

## Structure of JQuery command



# Introduction to JQuery

## Structure of JQuery command

- $() - it is just a function, we could replace it by callJQuery() or whatever();
- .action() - is an API function, that JQuery exposes. All API functions can be found in the documentation

```
23
24  function $$$(){
25      console.log("i am a $$$")
26  };
27
```

# Introduction to JQuery

## The difference between $ and jQuery

- The only difference is in time to type it

**$(“p”).hide() = jQuery(“p”).hide();**

Somewhere at the very beginning of JQuery lib there is something like that:

**window.$ = window.jQuery = jquery;**

# Introduction to JQuery

## The difference between $ and jQuery

JQuery initialization:
**window.$ = window.jQuery = jquery;**

It is the same as:
**a1 = a2 = a3;**

```
var a1;
var a2;
var a3 = function(){
    console.log("i am a a3")
};

a1 = a2 = a3;
```

# Introduction to JQuery

## Waiting for fully loaded DOM

Every page needs time to load all scripts, styles and other additional resources.

```
$(document).ready(function(){
	console.log("i am ready");
});
```

It is the same as:

```
$(function(){
	console.log("i am ready");
});
```

## *Task 2*

*Attach onLoad function to the HTML page. Function should console.log something after the page is loaded.*

# Introduction to JQuery
## Selector as element collection

By default, JQuery applies given command to **ALL** elements that satisfy the selector.

**$(“p”) -** selects **ALL** p tags on a page.

**$(“.user-form”) -** selects **ALL** elements with class **user-form**.

**$(“#smt”)** - select **ALL** elements with that id, though there should be only one.

# Introduction to JQuery

## Hide and show

Hide all p tag elements:
**$("p").hide();**

Show all p tag elements:
**$("p").show();**

Hide through pure js:

```
el.style.display = 'none';
```

Show through pure js:

```
el.style.display = '';
```

# Introduction to JQuery

## Fade in and fade out

Fade in all p tag elements:
**$(“p”).fadeIn();**

Fade out all p tag elements:
**$(“p”).fadeOut();**

Toggle fade all p tag elements:
**$(“p”).fadeToggle();**

# Introduction to JQuery

## Fade in and fade out

Fade in through pure js:

```
function fadeIn(el) {
  el.style.opacity = 0;

  var last = +new Date();
  var tick = function() {
    el.style.opacity = +el.style.opacity + (new Date() - last) / 400;
    last = +new Date();

    if (+el.style.opacity < 1) {
      (window.requestAnimationFrame && requestAnimationFrame(tick)) || setTimeout(tick, 16);
    }
  };

  tick();
}
fadeIn(el);
```

# Introduction to JQuery
## SlideUp and SlideDown

Slide up of all p tag elements:
**$(“p”).slideUp();**

Slide down of all p tag elements:
**$(“p”).slideDown();**

Toggle slide of all p tag elements:
**$(“p”).slideToggle();**

# Introduction to JQuery
## Good practise with selectors

Every time you use a selector JQuery performs search through the whole DOM. This actions can be quite costly if executed for hundreds of elements.

It is a good practise to save the search result and to use it in further actions:

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

$p.fadeOut();
$p.fadeIn();
```

# Introduction to JQuery
## Selectors

Apply to tag name:
**$(“p”).hide();**

Apply to a class:
**$(“.user-form”).hide();**

Apply to an id:
**$(“#confirmation-dialog”).hide();**

Apply to a element filtered by attribute:
**$(“input[name=’user-name’]”).hide();**

# Introduction to JQuery
## More CSS like selectors

Multiple tags:
**$(“p, h2”).hide();**

Filtered by a class:
**$(“div.superclass”).hide();**

Filtered by a class and narrowed to id:
**$(“div.hide-me #hidden”).hide();**

## *Task 3*

*Create p element. Write 2 functions: to hide(fadeOut) and to show(fadeIn) created p element. Call them from the console.*

# *Task 3.5*

*Change created functions in such way, that each function will be able to hide(fadeOut) or show(fadeIn) whichever element*

infoShare academy

# *Task 3.5.5*

*Create 3 div blocks. Place there a text. Give a class “red” to 2 of them. In one of div with class red create a p tag with text with id “red-text”. Create a function to fadeOut p tag only in divs with “red” class.*

# Introduction to JQuery
## Working with classes

Add class:
**$(“p”).addClass(“superclass-for-p”);**

Remove class:
**$(“p”).removeClass(“superclass-for-p”);**

Toggle class:
**$(“p”).toggleClass(“superclass-for-p”);**

Has class:
**$(“p”).hasClass(“superclass-for-p”);**

# *Task 4*

*Define a class with red text color. Create 2 functions for add the class and remove the class. Call them from the console*

infoShare academy

# Task 4.5

*Create toggleClass function which uses addClass, hasClass, removeClass and good practise of working with search results. The function checks if a class is present on an element and removes it if present and adds if absent*

# Introduction to JQuery
## Attaching styles directly

**$(selector).css()** - method for changing styles through JS.

Change single css rule:
**$("p").css("width", "10px");**

Change bunch of css rules:
**$("p").css({width: 100 + 100 + "px", color: "red", fontSize: 150 + "%"});**

# *Task 5*

*Create a function that can change color and text size for a given tag name.*
*Call the function from the console*

# 3.
# Deeper dive into JQuery

# Deeper dive into JQuery

## Working with text content

Get text of the element:

**var elementText = $(“p#p-descr”).text();**

Change text of the element:

**$(“p#p-descr”).text(“something”);**

# *Task 6*

*Create a function that can take a text from an element, change it and attach to the element*

# Deeper dive into JQuery
## Working with value

Methods **.text() and .val() are NOT** the same!

Get value of the element:
**var elementValue= $(“textarea[name=’ta-descr’]”).val();**

Change value of the element:
**$(“textarea[name=’ta-descr’]”).val(elementValue + “something”);**

## *Task 7*

*Create a function that can take a value from an element, change it and attach to the element*

# Deeper dive into JQuery

## Traversing the DOM

Traversing == Navigating through DOM.

```
$(“p#red-text”).parents();
$(“p#red-text”).parent();
$(“p#red-text”).siblings();
$(“p#red-text”).next();
$(“p#red-text”).prev();
$(“div.red”).children();
```

# Deeper dive into JQuery
## Searching in DOM

Find in children elements:
**$(“div.red”).find(“p”).fadeOut();**

Filter current element set:
**$(“div”).filter(“div.red”).fadeOut();**

So, .find() and .filter() **ARE NOT** the same.

# Deeper dive into JQuery
## Method chaining

JQuery allows us to chains its methods. It is necessary in order to create pipe of actions on a collection of selected elements:

```
$(“p”).fadeOut()
        .fadeIn();

$(”p”).parents()
        .find(“div#div-id”)
        .fadeOut()
        .fadeIn();
```

# Deeper dive into JQuery

## Method chaining

index.html main.js main.css

```
1   function jMock(selector){
2       var jMockObj = {
3           innerSelector: selector,
4
5           show: function(){
6               $(this.innerSelector).fadeIn();
7               console.log(`${this.innerSelector} is visible now`);
8
9               return this;
10          },
11
12          hide: function(){
13              $(this.innerSelector).fadeOut();
14              console.log(`${this.innerSelector} is hidden now`);
15
16              return this;
17          }
18      }
19
20
21      return jMockObj;
22  }
```

infoShare academy

# *Task 8*

*Add method toggleClass to the chained object*

infoShare academy

# Task 9

*Create a table. In the table create three rows. Each row has 3 columns. Place a p tag with id p-td in in the first column in the middle row. Now create a function to find the parent tr of the p-id and change background color to blue*

infoShare academy

## *Task 10*

*Create a function that changes background color of even rows to blue.*

*Create a function that changes background color to blue of all children of the given element*

infoShare academy

# Task 11

*Knowing id of p-id hide all columns in current row except for parent of p-id tag*

*Knowing id of p-id hide the middle column*

# Deeper dive into JQuery

## Element creation through JQuery

In order to create an element, we can use selector function:

**$(“<h1>asd</h1>”).appendTo(“div.class”);**
**$(“<h1>asd</h1>”).prependTo(“div.class”);**

# Deeper dive into JQuery

## Attaching elements to DOM

In order to append new element to DOM:
**$(element).append(“ asd ”);**

In order to prepend new element to DOM:
**$(element).prepend(“ asd ”);**

# Deeper dive into JQuery

## Replacing inner HTML

Replacing inner HTML:

**$(element).html(“ smt ”);**

# Deeper dive into JQuery

## Clearing and removing of elements

Remove an element from DOM:
**$(element).remove();**

Remove the content(all of HTML tags) of an element:
**$(element).empty();**

# Deeper dive into JQuery

## .each()

Apply a function to each element, should be used only when specific business logic must be executed:

```
$(“div.red”).each(function(){
        console.log($(this).html())
});
```

infoShare academy

# *Task 12*

*Create a ul based on array of data. Replace content in the last element for strong + custom text.*

*Create a function that can clear all created data*

# Deeper dive into JQuery

## Events and event listeners

- Event is fired when any of predefined actions happens
- Event can be attached and detached
- Events can be combined on a single element

Log text of an element when click event is fired:

```
$("p#p-desc").click(function(){
    var $this = $(this);
    console.log($this.text());
});
```

# *Task 13*

*Create an input, a textarea and a button. Create a logic that is fired when click event is fired on the button. The logic takes the input value and places it into the textarea.*

# Deeper dive into JQuery

## Input listeners

Log value of an input when change event is fired:

```
$("input[name='user-name']").change(function(){
	var $this = $(this);
	console.log($this.val());
});
```

Log value of a select when change event is fired:

```
$("select[name='user-type']").change(function(){
	var $this = $(this);
	console.log($this.val());
});
```

# *Task 14*

*Create an input, on which change is fired after inner value is changed and logged.*

*Create a select on which change is fired after selection and the value is logged.*

# Deeper dive into JQuery

## Event listeners types:

There are many types of events:

**.dbclick();**
**.hover();**
**.mousedown();**
**.mouseup();**
**.focus();**
**.blur();**
**.keypress();**

# *Task 14.5*

*Create event listeners for each listed action.*

# Deeper dive into JQuery

## Alternative syntax of event listeners

There is another way of attaching events in JQuery:

```
$("p").on("click", function(){
        …
});
```

Though it is acceptable, due to modern standards, using .click/.whatever is better approach.However, in order to auto attach a listener to **dynamically added elements** this syntax should be used.

# Deeper dive into JQuery

## Event listeners on dynamically added elements

Log the text of an clicked p, that was added dynamically, when click event is fired:

```
$("div#div-id").on("click", "p", function(){
        console.log($this.text());
});
```

infoShare academy

# *Task 15*

*Create an click event listener on all p element in a div, so that clicked p should log its text.*

*Create a function which appends another p the div. Click the p element - log should be called.*

# Deeper dive into JQuery

## Turning off all of the listeners

Sometimes it is necessary to turn off events listeners on the element/elements.

Turns off all event listeners:

**$(“p”).off();**

infoShare academy

# Task 16

*Create a select. Attach on change event listener to the select.*
*Create a button. Attach on click event listener to the button in order to turn off previous event listener from the select.*

# Deeper dive into JQuery

## Working with attributes and properties

.attr() works with classic attributes.

Reading an attr of an element:
**$(element).attr();**

Replacing an attr of an element:
**$(element).attr("name", "name-smt");**

Remove an attr from an element:
**$(element).removeAttr("name");**

.prop() works with internal meta model.

Reading an prop of an element:
**$(element).prop();**

Replacing an prop of an element:
**$(element).prop("checked", true);**

Remove an prop from an element:
**$(element).removeProp("name");**

infoShare academy

# *Task 17*

*Create some checkboxes. Create a function to uncheck all checked checkboxes and check all unchecked checkboxes*

# Deeper dive into JQuery

## Animations

Select an element and pass the css rules object to be animated:
**$(element).animate({property: “value”},  timeInMiliseconds);**

Turning opacity to half visibility during 3 secs:
**$(“div”).animate({opacity: “0.5”}, 3000);**

# *Task 16*

*“ Create a function to animate top or left property of an element by id*