

50+ jQuery vežbi sa rešenjima (od osnovnih do naprednih tema)

Priručnik za učenike

October 18, 2025

Cilj: sistematski pređi ključne koncepte jQuery biblioteke kroz kratke praktične zadatke. Svaka vežba ima ukratko opis i minimalno rešenje (HTML + jQuery).

Kako koristiti ovaj materijal

- U svim primerima pretpostavlja se da je jQuery uključen:

```
<script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>
```

- U primerima koristimo `$(document).ready(...)` ili stavljamo skriptu pri dnu `<body>`.
- Kod je minimalistički – cilj je ideja, ne stil.

Blok A — Osnove selektora, manipulacije i događaja (1–15)

Zadatak 1. Selektor po ID-ju i promena teksta

Napravi h1 sa id-em title. Klik na dugme menja tekst naslova u „Zdravo, jQuery!“.

Rešenje.

```
<h1 id="title">Naslov</h1>
<button id="btn">Promeni</button>
<script>
  $('#btn').on('click', function(){
    $('#title').text('Zdravo, jQuery!');
  });
</script>
```

Zadatak 2. Selektor po klasi i CSS

Dva paragrafa imaju klasu note. Klik na „Označi“ dodaje žutu pozadinu svima sa tom klasom. **Rešenje.**

```
<p class="note">Tekst 1</p><p class="note">Tekst 2</p>
<button id="mark">Označi</button>
<script>
  $('#mark').on('click', function(){
    $('.note').css('background', '#fff7a8');
  });
</script>
```

Zadatak 3. Složen selektor

Oboji sve neparne stavke liste (`li:odd`) u sivo. **Rešenje.**

```
<ul id="items"><li>A</li><li>B</li><li>C</li><li>D</li></ul>
<script>
  $('#items li:odd').css('background','#eee');
</script>
```

Zadatak 4. `text()` vs `html()`

Prikaži razliku: jednim klikom ubaci raw tekst „`jQuery`“, drugim klikom ubaci HTML. **Rešenje.**

```
<div id="box"></div>
<button id="asText">Kao tekst</button>
<button id="asHtml">Kao HTML</button>
<script>
  $('#asText').on('click', ()=>$('#box').text('<b>jQuery</b>'));
  $('#asHtml').on('click', ()=>$('#box').html('<b>jQuery</b>'));
</script>
```

Zadatak 5. Dodavanje i uklanjanje klasa

Toggle klasu `active` na kartici kad se klikne. **Rešenje.**

```
<div id="card">Kartica</div>
<script>
  $('#card').on('click', function(){
    $(this).toggleClass('active');
  });
</script>
<style>
  .active{border:2px solid #4a90e2; padding:6px;}
</style>
```

Zadatak 6. `val()` unos

Polje input + dugme: pritisak kopira vrednost inputa u `#out`. **Rešenje.**

```
<input id="name" placeholder="Ime"><button id="go">Prikaži</button>
<div id="out"></div>
<script>
  $('#go').on('click', ()=>$('#out').text($('#name').val()));
</script>
```

Zadatak 7. Atributi: `attr()` i `prop()`

Toggle `disabled` na dugmetu i promeni `title` atribut. **Rešenje.**

```
<button id="action" title="Klikni me">Akcija</button>
<button id="toggle">Zaključaj/Otključaj</button>
<script>
  $('#toggle').on('click', function(){
    const b = $('#action');
    b.prop('disabled', !b.prop('disabled'));
    b.attr('title', b.prop('disabled')?'Zaključano':'Klikni me');
  });
</script>
```

Zadatak 8. Umetanje elemenata: *append()*, *prepend()*

Dodaj stavke na kraj i početak liste. **Rešenje.**

```
<ul id="lst"><li>Start</li></ul>
<button id="addEnd">Dodaj kraj</button>
<button id="addStart">Dodaj početak</button>
<script>
  $('#addEnd').on('click', ()=>$('#lst').append('<li>Kraj</li>'));
  $('#addStart').on('click', ()=>$('#lst').prepend('<li>Početak</li>'));
</script>
```

Zadatak 9. *before()* i *after()*

Ubaci objašnjenje pre i posle slike. **Rešenje.**

```

<script>
  $('#pic').before('<p>Pre slike</p>').after('<p>Posle slike</p>');
</script>
```

Zadatak 10. Uklanjanje i pražnjenje: *remove()* vs *empty()*

Klik „Empty“ prazni sadržaj, „Remove“ briše ceo kontejner. **Rešenje.**

```
<div id="panel"><b>Tekst</b> i <i>deca</i></div>
<button id="btnEmpty">Empty</button>
<button id="btnRemove">Remove</button>
<script>
  $('#btnEmpty').on('click', ()=>$('#panel').empty());
  $('#btnRemove').on('click', ()=>$('#panel').remove());
</script>
```

Zadatak 11. *Chaining*

Jednim klikom: doda klasu, promeni tekst, zatim fadeOut pa fadeIn. **Rešenje.**

```
<div id="chip">Čip</div><button id="do">Anim</button>
<script>
  $('#do').on('click', function(){
    $('#chip').addClass('hit').text('Obrađeno').fadeOut(200).fadeIn(200);
  });
</script>
<style>.hit{border:1px solid #aaa;}</style>
```

Zadatak 12. *Delegacija događaja*

Klik na dinamički dodate stavke liste hvata se preko roditelja. **Rešenje.**

```
<ul id="menu"></ul>
<button id="add">Dodaj stavku</button>
<script>
  let i=1;
  $('#add').on('click', ()=>$('#menu').append('<li>Stavka '+(i++)+'</li>'));
  $('#menu').on('click', 'li', function(){ $(this).toggleClass('sel'); });
</script>
<style>.sel{background:#d0ebff;}</style>
```

Zadatak 13. *Keyboard događaji*

Enter u inputu pokreće isti handler kao klik na dugme „Traži“. **Rešenje.**

```
<input id="q" placeholder="Upit"><button id="search">Traži</button>
<div id="result"></div>
<script>
  function go(){ $('#result').text('Tražim: ' + $('#q').val()); }
  $('#search').on('click', go);
  $('#q').on('keydown', e => { if(e.key==='Enter') go(); });
</script>
```

Zadatak 14. *Prevent default*

Link „Pošalji“ ne sme da otvara URL već da prikaže poruku. **Rešenje.**

```
<a id="send" href="https://example.com">Pošalji</a>
<script>
  $('#send').on('click', function(e){
    e.preventDefault();
    alert('Slanje kroz jQuery!');
  });
</script>
```

Zadatak 15. *Stop propagation*

Klik na dugme u okviru kartice ne sme da aktivira klik kartice. **Rešenje.**

```
<div id="card2" style="padding:10px;border:1px solid #aaa;">
  Kartica <button id="inner">Dugme</button>
</div>
<script>
  $('#card2').on('click', ()=>alert('Kartica klik'));
  $('#inner').on('click', function(e){ e.stopPropagation(); alert('Dugme klik'); });
</script>
```

Blok B — DOM obilazak, forme, animacije, dimenzije (16–30)

Zadatak 16. *Traversing: parent(), children(), find()*

Označi roditelja kliknutog elementa i svu njegovu decu. **Rešenje.**

```
<div class="box"><span class="hit">Klikni</span><i>još</i></div>
<script>
  $('.hit').on('click', function(){
    const p = $(this).parent().css('border','1px solid #4a90e2');
    p.children().css('background','#eef6ff');
  });
</script>
```

Zadatak 17. *closest() i siblings()*

Klik na element boji njegov najbliži .row i sve „susedne“ stavke. **Rešenje.**

```
<ul class="row"><li>A</li><li class="x">B</li><li>C</li></ul>
<script>
  $('x').on('click', function(){
    const r = $(this).closest('.row').css('outline','1px dashed #888');
    $(this).siblings().css('opacity',0.6);
  });
</script>
```

Zadatak 18. *Forme: serialize()*

Po submit-u forme, spreči slanje podataka, reload i prikaži serijalizovane podatke. **Rešenje.**

```
<form id="f"
  action="https://httpbin.org/anything"
  method="GET"
  enctype="application/x-www-form-urlencoded"
  target="_blank">
  <input name="ime"><input name="grad"><button>OK</button></form>
<pre id="out2"></pre>
<script>
  $('#f').on('submit', function(e){
    e.preventDefault();
    $('#out2').text($(this).serialize());
  });
</script>
```

Zadatak 19. *Validacija forme (minimalno)*

Ako je polje prazno, prikaži poruku i dodaj klasu greške. **Rešenje.**

```
<input id="email" placeholder="Email"><button id="save">Sačuvaj</button>
<div id="msg"></div>
<script>
  $('#save').on('click', function(){
    const v = $('#email').val().trim();
    if(!v){ $('#email').addClass('err'); $('#msg').text('Email je obavezan'); }
    else { $('#email').removeClass('err'); $('#msg').text('OK'); }
  });
</script>
<style>.err{border:1px solid #e03131;}</style>
```

Zadatak 20. *data() API*

Sačuvaj broj klikova u data i prikaži ga. **Rešenje.**

```
<button id="count" data-n="0">Klikni me</button>
<script>
  $('#count').on('click', function(){
    let n = $(this).data('n'); n++; $(this).data('n', n).text('Klikova: '+n);
  });
</script>
```

Zadatak 21. *Animacije: hide()/show()/toggle()*

Toggle panel na klik. **Rešenje.**

```
<button id="t">Prikaži/Sakrij</button>
<div id="p" style="padding:10px;background:#fafafa;">Panel</div>
<script>
  $('#t').on('click', ()=>$('#p').toggle(200));
</script>
```

Zadatak 22. Animacije: *fadeIn/fadeOut*

Fade efekat na sliku pri hover-u (ulaz/izlaz). **Rešenje.**

```

<script>
  $('#photo').on('mouseenter', ()=>$('#photo').fadeTo(150,1))
    .on('mouseleave', ()=>$('#photo').fadeTo(150,0.6));
</script>
```

Zadatak 23. Animacije: *slideDown/slideUp*

Akoridion: klik na naslov otvara/zatvara sadržaj. **Rešenje.**

```
<h3 class="hdr">Naslov</h3>
<div class="cnt" style="display:none;">Sadržaj</div>
<script>
  $('.hdr').on('click', function(){ $(this).next('.cnt').slideToggle(200); });
</script>
```

Zadatak 24. *animate()* prilagođeno

Povećaj širinu kutije sa 100px na 250px u 400ms. **Rešenje.**

```
<div id="bx" style="width:100px;height:20px;background:#ddd;"></div>
<button id="grow">Rasti</button>
<script>
  $('#grow').on('click', ()=>$('#bx').animate({width:'250px'}, 400));
</script>
```

Zadatak 25. Dimenzije: *width/height/innerWidth/outerWidth*

Prikaži dimenzije elementa sa paddingom i borderom. **Rešenje.**

```
<div id="box3" style="width:120px;padding:10px;border:5px solid #ccc;"></div>
<pre id="dim"></pre>
<script>
  const b = $('#box3');
  $('#dim').text([
    'width=', b.width(),
    ' innerWidth=', b.innerWidth(),
    ' outerWidth=', b.outerWidth(true)
  ].join(' '));
</script>
```

Zadatak 26. Pozicija: *offset()* i *position()*

Prikaži offset stranice i relativnu poziciju u roditelju. **Rešenje.**

```

<div style="position:relative; margin-top:40px;">
  <div id="dot" style="position:absolute;left:30px;top:20px;">•</div>
</div>
<pre id="pos"></pre>
<script>
  $('#pos').text('offset='+JSON.stringify($('#dot').offset()+
    ' position='+JSON.stringify($('#dot').position()));
</script>

```

Zadatak 27. Skrol događaji

Prikaži koliko je prozor skrolovan po Y osi. **Rešenje.**

```

<div style="height:1200px"></div>
<pre id="s"></pre>
<script>
  $(window).on('scroll', ()=>$('#s').text($(window).scrollTop()));
</script>

```

Zadatak 28. Throttle sa setTimeout

Na input keyup, ažuriraj rezultat tek 300ms nakon poslednje promene. **Rešenje.**

```

<input id="deb"><div id="r"></div>
<script>
  let t=null;
  $('#deb').on('keyup', function(){
    clearTimeout(t);
    const v = $(this).val();
    t = setTimeout(()=>$('#r').text(v), 300);
  });
</script>

```

Zadatak 29. Rad sa checkbox-ovima

„Označi sve“ selektuje/deselektuje sve stavke liste. **Rešenje.**

```

<label><input type="checkbox" id="all"> Označi sve</label>
<div>
  <label><input class="it" type="checkbox"> A</label>
  <label><input class="it" type="checkbox"> B</label>
</div>
<script>
  $('#all').on('change', function(){ $('#.it').prop('checked', this.checked); });
</script>

```

Zadatak 30. Promena atributa sa funkcijom

Povećavaj data-score za +10 pri svakom kliku koristeći funkcijsku varijantu attr(). **Rešenje.**

```

<div id="sc" data-score="0">Score</div>
<script>
  $('#sc').on('click', function(){
    $(this).attr('data-score', function(_,old){ return (+old)+10; });
  });
</script>

```

Zadatak 31. Filtriranje: *filter()* i *not()*

Oboji samo stavke liste koje sadrže slovo „a“, ostale deaktiviraj. **Rešenje.**

```
<ul id="w"><li>Java</li><li>CSS</li><li>HTML</li><li>Scala</li></ul>
<script>
  $('#w li').filter(function(){ return $(this).text().toLowerCase().includes('a'); })
    .css('background','#e6fcf5');
  $('#w li').not(':contains(a)').css('opacity',0.5);
</script>
```

Zadatak 32. Kloniranje: *clone()*

Na klik dupliciraj element i ubaci ga posle originala. **Rešenje.**

```
<div id="badge">Tag</div><button id="dup">Dupliraj</button>
<script>
  $('#dup').on('click', ()=>$('#badge').clone().insertAfter('#badge'));
</script>
```

Blok C — AJAX, JSON, templating, skladištenje (31–40)

Zadatak 33. *\$.get()* osnovno

GET na statički JSON fajl i prikaži polje title. **Rešenje.**

```
// data.json: { "title":"Pozdrav", "items":[1,2,3] }
<div id="ajax1"></div>
<script>
  $.get('data.json', function(d){ $('#ajax1').text(d.title); }, 'json');
</script>
```

Zadatak 34. *\$.post()* forma

Pošalji formu kao POST i prikaži odgovor (mock/server pretpostavljen). **Rešenje.**

```
<form id="login"><input name="u"><input name="p" type="password"><button>Prijava</button></form>
<pre id="resp"></pre>
<script>
  $('#login').on('submit', function(e){
    e.preventDefault();
    $.post('/api/login', $(this).serialize(), function(r){
      $('#resp').text(JSON.stringify(r));
    }, 'json').fail(()=>$('#resp').text('Greška'));
  });
</script>
```

Zadatak 35. *\$.ajax()* sa opcijama

GET sa beforeSend, timeout i error handlerima. **Rešenje.**


```

<pre id="outA"></pre>
<script>
$.ajax({
  url: '/api/data', method: 'GET', dataType: 'json', timeout: 2000,
  beforeSend: () => $('#outA').text('Učitavam...'),
  success: d => $('#outA').text(JSON.stringify(d)),
  error: () => $('#outA').text('Greška ili timeout')
});
</script>

```

Zadatak 36. Dinamičko renderovanje liste iz JSON-a

Mapiraj niz objekata u li elemente. **Rešenje.**

```

// Pretpostavimo: var data=[{name: 'Ana'}, {name: 'Marko'}];
<ul id="listX"></ul>
<script>
  var data=[{name: 'Ana'}, {name: 'Marko'}];
  $('#listX').html(data.map(o=>`<li>${o.name}</li>`).join(''));
</script>

```

Zadatak 37. Loader/spinner tokom AJAX-a

Pokaži „Učitavam...” dok traje zahtev; sakrij kad stigne odgovor. **Rešenje.**

```

<div id="spin" style="display:none;">Učitavam...</div><div id="data"></div>
<script>
$(document).ajaxStart(()=>$('#spin').show());
$(document).ajaxStop(()=>$('#spin').hide());
$.getJSON('/api/items', d => $('#data').text(d.length+' stavki'));
</script>

```

Zadatak 38. Keširanje podataka u localStorage

Ako već postoji keš, koristi ga; inače učitaj pa sačuvaj. **Rešenje.**

```

<div id="cache"></div>
<script>
const k='itemsCache';
const cached = localStorage.getItem(k);
if(cached){
  $('#cache').text('Iz keša: '+JSON.parse(cached).length);
}else{
  $.getJSON('/api/items', d => {
    localStorage.setItem(k, JSON.stringify(d));
    $('#cache').text('Sveže: '+d.length);
  });
}
</script>

```

Zadatak 39. Form-data u JSON objekat

Pretvori `serializeArray()` rezultat u objekt pa pošalji kao JSON. **Rešenje.**

```

<form id="f2"><input name="a"><input name="b"><button>OK</button></form>
<pre id="j"></pre>
<script>
  $('#f2').on('submit', function(e){
    e.preventDefault();
    const obj={}; $(this).serializeArray().forEach(x=>obj[x.name]=x.value);
    $('#j').text(JSON.stringify(obj));
    $.ajax({url:'/api/save', method:'POST', contentType:'application/json', data:JSON.
      stringify(obj)});
  });
</script>

```

Zadatak 40. *Greške i .fail()*

Namerno zovi nepostojeći URL i prikaži poruku o grešci u crvenom. **Rešenje.**

```

<div id="err"></div>
<script>
  $.get('/nope').fail(function(xhr){
    $('#err').css('color', '#e03131').text('Greška: '+xhr.status);
  });
</script>

```

Zadatak 41. *Timeout i abort zahteva*

Pokreni zahtev i prekini ga nakon 300ms ako nije završen. **Rešenje.**

```

<pre id="ab"></pre>
<script>
  const req = $.get('/api/slow', d=>$('#ab').text('OK'));
  setTimeout(()=>{ req.abort(); $('#ab').text('Prekinuto'); }, 300);
</script>

```

Zadatak 42. *Mini templating funkcija*

Funkcija koja menja `{{name}}` u stringu na osnovu objekta; koristi je uz jQuery za prikaz. **Rešenje.**

```

<div id="tplOut"></div>
<script>
  function tpl(s, ctx){ return s.replace(/\{\{(\w+)\}\}/g, (_,k)=>ctx[k]??''); }
  const s = '<p>Zdravo, {{name}}!</p>';
  $('#tplOut').html(tpl(s, {name:'Svet'}));
</script>

```

Blok D — Plugin obrasci, performanse, pristupačnost, integracije (41–50)

Zadatak 43. *Kreiranje prostog jQuery plugina*

Napiši plugin `blink` koji treperi element N puta. **Rešenje.**

```

<div id="blinkMe">Treperim</div>
<script>
  (function($){
    $.fn.blink = function(times=3, speed=150){
      return this.each(function(){
        const $el=$(this); let i=0;
        function step(){
          $el.fadeOut(speed).fadeIn(speed, ()=>{ if(++i<times) step(); });
        }
        step();
      });
    }(jQuery);
    $('#blinkMe').blink(4,100);
  })
</script>

```

Zadatak 44. *Plugin sa opcijama i podrazumevanim vrednostima*

`$.fn.highlight` prima `color` i `duration`. **Rešenje.**

```

<p class="hl">Tekst</p>
<script>
  (function($){
    $.fn.highlight = function(opts){
      const o = $.extend({color:'#fff3bf', duration:300}, opts);
      return this.each(function(){
        const $t=$(this), old=$t.css('backgroundColor');
        $t.css('backgroundColor', o.color)
          .animate({opacity:0.9}, o.duration)
          .animate({opacity:1}, o.duration, ()=>$t.css('backgroundColor', old));
      });
    };
  })(jQuery);
  $('.hl').highlight({color:'#c0ebff', duration:200});
</script>

```

Zadatak 45. *Event namespacing*

Registruj dva handlera sa različitim „namespace“-om pa ukloni samo jedan. **Rešenje.**

```

<button id="ns">Klik</button>
<script>
  $('#ns').on('click.a', ()=>console.log('A'));
  $('#ns').on('click.b', ()=>console.log('B'));
  // Uklanjanje samo A:
  $('#ns').off('click.a');
</script>

```

Zadatak 46. *Custom događaj*

Triggeruj i slušaj sopstveni događaj `done`. **Rešenje.**

```

<div id="job"></div>
<script>
  $('#job').on('done', (e, data)=>$('#job').text('Gotovo: '+data.msg));
  setTimeout(()=>$('#job').trigger('done', {msg:'uspeh'}), 200);
</script>

```

Zadatak 47. Delegacija i performance

Umesto bind-a na 1000 dugmadi, veži jedan handler na kontejner. **Rešenje.**

```
<div id="wrap"></div><pre id="log"></pre>
<script>
  for(let i=0;i<1000;i++) $('#wrap').append('<button class="b">#'+i+'</button>');
  $('#wrap').on('click', '.b', function(){ $('#log').text($(this).text()); });
</script>
```

Zadatak 48. Pristupačnost: fokus i tastatura

Na Enter/Space aktiviraj „link-karticu“ koja nije <a>. **Rešenje.**

```
<div id="card" tabindex="0" role="button" aria-pressed="false">Otvori</div>
<script>
  $('#card').on('keydown', function(e){
    if(e.key==='Enter' || e.key===' '){ e.preventDefault(); $(this).click(); }
  }).on('click', function(){
    $(this).attr('aria-pressed', $(this).attr('aria-pressed')==='false'?true:false'
    );
  });
</script>
```

Zadatak 49. Rad sa Deferred / Promise

Kreiraj veštački async zadatak koji vraća \$.Deferred() i koristi .then(). **Rešenje.**

```
<pre id="pms"></pre>
<script>
  function asyncJob(){
    const d=$.Deferred();
    setTimeout(()=>d.resolve('OK'), 200);
    return d.promise();
  }
  asyncJob().then(v=>$('#pms').text(v));
</script>
```

Zadatak 50. Integracija sa Fetch API (van jQuery) + jQuery render

Učitaj JSON preko Fetch-a, prikaži pomoću jQuery. **Rešenje.**

```
<ul id="u"></ul>
<script>
  fetch('/api/users').then(r=>r.json()).then(d=>{
    $('#u').html(d.map(x=><li>${x.name}</li>`).join(''));
  });
</script>
```

Zadatak 51. Debounce util za input događaje

Napiši generički debounce(fn, wait) i koristi ga sa keyup. **Rešenje.**

```
<input id="search2"><div id="outX"></div>
<script>
  function debounce(fn, wait){ let t; return function(...a){ clearTimeout(t); t=
    setTimeout(()=>fn.apply(this,a), wait); }; }
  $('#search2').on('keyup', debounce(function(){
```

```

    $('#outX').text($(this).val());
  }, 300));
</script>

```

Zadatak 52. Mala SPA navigacija (hashchange)

Na promenu hash-a prikazuj odgovarajući „view“. **Rešenje.**

```

<nav>
  <a href="#home">Home</a> | <a href="#about">About</a>
</nav>
<div id="view"></div>
<script>
  function render(){
    const h=location.hash||'#home';
    $('#view').text(h=='#about'?'0 aplikaciji':'Dobrodošli');
  }
  $(window).on('hashchange', render); render();
</script>

```

Zadatak 53. Testiranje funkcionalnosti (Smoke test)

Proveri da je jQuery učitani i da je element dostupan pre bind-ovanja. **Rešenje.**

```

<div id="ok"></div>
<script>
  if(window.jQuery && $('#ok').length){ $('#ok').text('jQuery OK'); }
</script>

```

Bonus mini-vežbe (51–55) za dodatnu praksu

Zadatak 54. Dinamički select iz niza

Popuni select iz niza stringova i prikaži izbor. **Rešenje.**

```

<select id="sel"></select><div id="cho"></div>
<script>
  const arr=['Beograd','Novi Sad','Niš'];
  $('#sel').html(arr.map(x=><option>${x}</option>).join(''))
    .on('change', function(){ $('#cho').text($(this).val()); });
</script>

```

Zadatak 55. Brojač karaktera

Prikazuj preostale karaktere (maks 50) i oboji crveno kad pređe limit. **Rešenje.**

```

<textarea id="t" maxlength="50"></textarea>
<div id="c"></div>
<script>
  $('#t').on('input', function(){
    const left=50-$(this).val().length;
    $('#c').text(left);
    $('#c').css('color', left<0?'#e03131':'inherit');
  });
</script>

```

Zadatak 56. *Sortiranje (DOM reflow)*

Klik na „Sortiraj“ sortira li po alfabetu. **Rešenje.**

```
<ul id="l2"><li>banan</li><li>ana</li><li>cuk</li></ul>
<button id="sort">Sortiraj</button>
<script>
  $('#sort').on('click', function(){
    const li = $('#l2 li').get().sort((a,b)=>$(a).text().localeCompare($(b).text()));
    $('#l2').empty().append(li);
  });
</script>
```

Zadatak 57. *Drag toggle (klasa pri držanju miša)*

Dok je dugme miša pritisnuto nad elementom, dodaj klasu dragging. **Rešenje.**

```
<div id="drag" style="width:100px;height:40px;background:#ddd;">Drži</div>
<script>
  $('#drag').on('mousedown', function(){ $(this).addClass('dragging'); })
    .on('mouseup mouseleave', function(){ $(this).removeClass('dragging'); });
</script>
<style>.dragging{outline:2px dashed #4a90e2;}</style>
```

Zadatak 58. *Clipboard (kopiranje teksta)*

Klikom kopiraj tekst iz inputa u clipboard (moderna API + jQuery za vezivanje). **Rešenje.**

```
<input id="clip" value="Kopiraj me"><button id="cpy">Kopiraj</button>
<script>
  $('#cpy').on('click', async function(){
    try{ await navigator.clipboard.writeText($('#clip').val()); alert('Kopirano'); }
    catch{ alert('Neuspeh'); }
  });
</script>
```

Tematska pokrivenost

- Selektori, traversing, manipulacija DOM-om i CSS-om
- Događaji: klik, tastatura, delegacija, namespace, custom događaji
- Forme: čitanje/validacija, serialize/serializeArray, val, prop, attr
- Animacije: show/hide, fade/slide, animate
- Dimenzije i pozicioniranje: width/height, inner/outer, offset/position, scroll
- AJAX: \$.get, \$.post, \$.ajax, JSON, loader, greške, abort
- Skladištenje: localStorage
- Utility: debounce, templating, Deferred/Promise
- Plugin obrazac: \$.fn.* sa opcijama i podrazumevanim vrednostima

- Integracije: Fetch + jQuery render, mini SPA hash routing

Saveti. Vežbe su kratke – idealne za „live coding“. Način rada:

1. prvo napiši HTML kostur, zatim jQuery rešenje,
2. eksperimentiši sa selektorima i događajima,
3. dodaj **edge** slučajeve (prazni inputi, greške AJAX-a),
4. refaktoriši u male pomoćne funkcije i plugin-e.