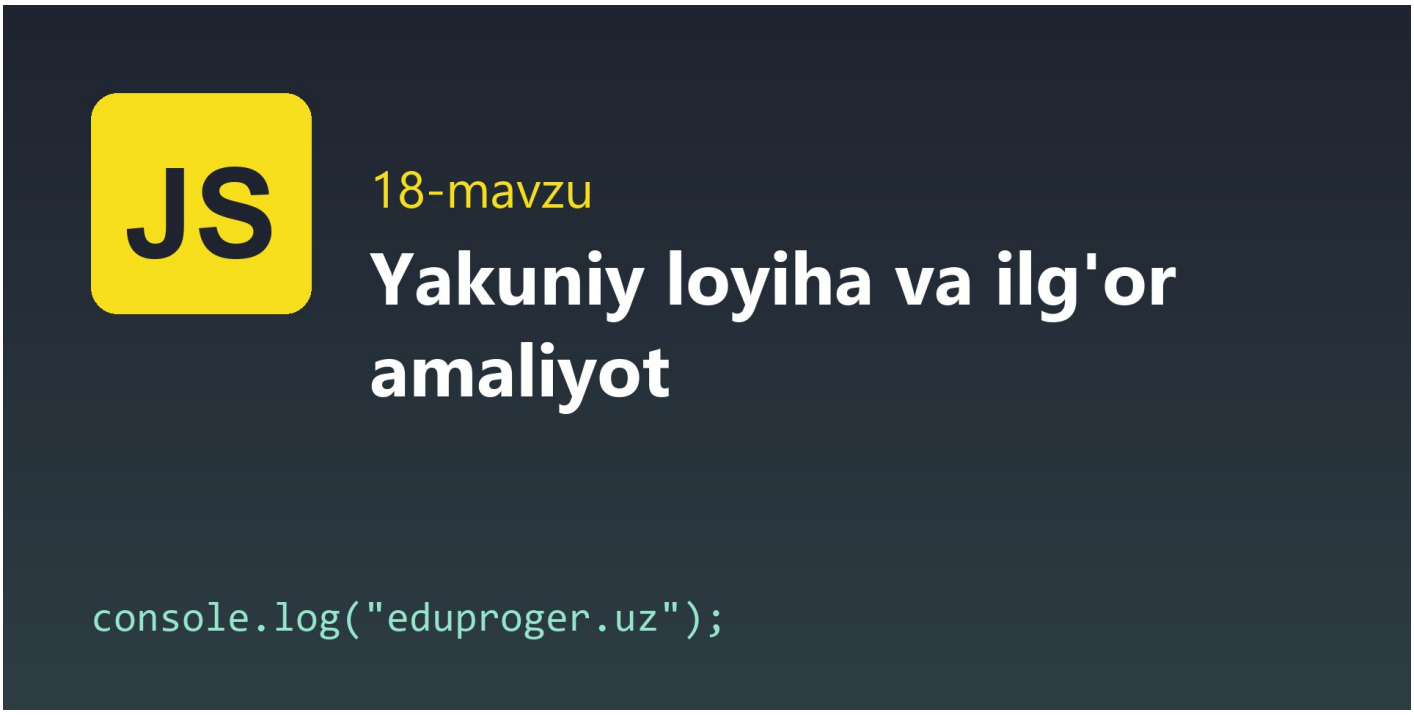
18. Yakuniy loyiha va ilg'or amaliyot

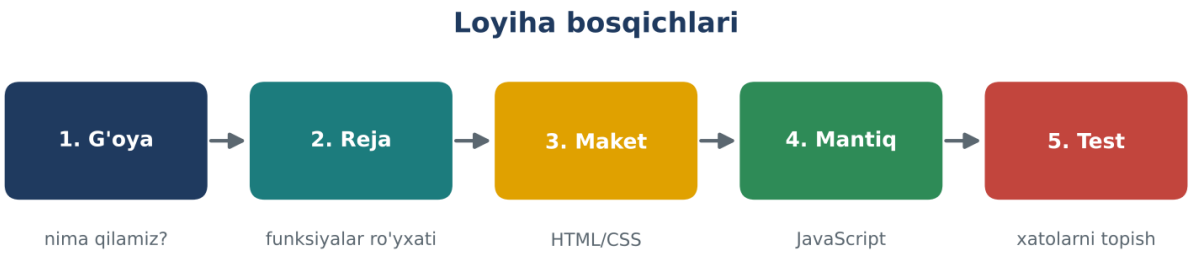


Siz 17 ta mavzuni o'rgandingiz. Endi hammasini **birlashtiramiz**: birinchi to'liq loyihamiz — **vazifalar ro'yxati (To-Do)** ilovasi. Bu dars loyihani rejalashtirish va uni qadam-baqadam yozishni ko'rsatadi.

> **Bu darsda nimani o'rganasiz**
>
> Loyihani bosqichlarga bo'lish.
>
> To-Do ilovasi: massiv + DOM + hodisalar.
>
> Ma'lumotni `localStorage` ga saqlash.
>
> Toza kod yozish odatlari.
>
> Xatoni tez topish usullari.

1.1. Loyiha qanday boshlanadi?

Yangi boshlovchilarning eng katta xatosi — **darrov kod yozishga** tushish. Ustalar avval o'ylaydi. Loyiha beshta bosqichdan iborat:



1-rasm. Loyiha bosqichlari

1-jadval. Loyiha rejasi

Bosqich	Savol	To-Do ilovasi uchun javob
1. G'oya	Nima qilamiz?	Vazifalar ro'yxati
2. Reja	Qanday funksiyalar kerak?	Qo'shish, belgilash, o'chirish, hisoblash
3. Maket	Qanday ko'rinadi?	Kiritish maydoni, tugma, ro'yxat
4. Mantiq	Qanday ishlaydi?	Massiv + DOM + hodisalar
5. Test	Xato yo'qmi?	Bo'sh matn, qayta yuklash, o'chirish

Katta ishni kichiklarga bo'ling

"To-Do ilovasi" katta ish. "Massivga vazifa qo'shish" — kichik ish. Har safar **bitta kichik qadam** yozing va ishlashini tekshiring.

1.2. 1-qadam: ma'lumot (holat)

Ilovaning hamma ma'lumoti **bitta joyda** — massivda turadi. Bu **holat (state)** deyiladi. Har bir vazifa — obyekt.

1_qadam.js

```
let vazifalar = [
  { id: 1, matn: "JavaScript o'rganish", bajarildi: false },
  { id: 2, matn: "Loyiha yozish", bajarildi: false },
];

function qosh(matn) {
  const toza = matn.trim();
  if (!toza) return false;                              // bo'sh matnni rad etam
  vazifalar.push({ id: Date.now() + vazifalar.length, matn: toza, bajarildi: f
  return true;
}
function belgila(id) {
  const v = vazifalar.find((x) => x.id === id);
  if (v) v.bajarildi = !v.bajarildi;
}
function ochir(id) {
  vazifalar = vazifalar.filter((x) => x.id !== id);
```

```
}

console.log(qosh("  Kitob o'qish  "), qosh("   "));
belgila(1);
ochir(2);
console.log(vazifalar.map((v) => (v.bajarildi ? "[x] " : "[ ] ") + v.matn));
```

Natija:

```
true false
[ "[x] JavaScript o'rganish", "[ ] Kitob o'qish" ]
```

1.3. 2-qadam: ekranga chiqarish

Holat tayyor. Endi uni DOM ga chizadigan `chiz()` funksiyasi kerak (12-mavzu). Qoida: **holat o'zgarsa — `chiz()` ni qayta chaqiramiz**.

2_qadam.js

```
const vazifalar = [{ id: 1, matn: "Sut olish", bajarildi: true }, { id: 2, mat

const ro = document.querySelector("#ro");
const soni = document.querySelector("#soni");

function chiz() {
  ro.innerHTML = "";                                    // eskisini tozalaymiz
  vazifalar.forEach((v) => {
    const li = document.createElement("li");
    li.textContent = v.matn;                            // textContent — xavfsi
    li.dataset.id = v.id;
    if (v.bajarildi) li.classList.add("tayyor");
    ro.append(li);
  });
  const qoldi = vazifalar.filter((v) => !v.bajarildi).length;
  soni.textContent = `Qolgan: ${qoldi} / ${vazifalar.length}`;
}
chiz();
console.log(document.body.innerHTML);
```

Natija:

```
<p id="soni">Qolgan: 1 / 2</p><ul id="ro"><li data-id="1" class="tayyor">Sut o
```

1.4. 3-qadam: hodisalar

Foydalanuvchi harakatlari: forma yuborish (qo'shish), vazifani bosish (belgilash), o'chirish tugmasi. **Event delegation** (13-mavzu) bilan bitta tinglovchi hamma `li` larga yetadi.

3_qadam.js

```
let vazifalar = [];
const forma = document.querySelector("#forma");
```

```
const kirit = document.querySelector("#kirit");
const ro = document.querySelector("#ro");

function chiz() {
  ro.innerHTML = "";
  vazifalar.forEach((v) => {
    const li = document.createElement("li");
    li.dataset.id = v.id;
    li.className = v.bajarildi ? "tayyor" : "";
    li.innerHTML = '<span></span> <button class="o">×</button>';
    li.querySelector("span").textContent = v.matn;
    ro.append(li);
  });
}

forma.addEventListener("submit", (e) => {
  e.preventDefault();
  const matn = kirit.value.trim();
  if (!matn) return;
  vazifalar.push({ id: vazifalar.length + 1, matn, bajarildi: false });
  kirit.value = "";
  chiz();
});

ro.addEventListener("click", (e) => {             // event delegation
  const li = e.target.closest("li");
  if (!li) return;
  const id = Number(li.dataset.id);
  if (e.target.matches(".o")) vazifalar = vazifalar.filter((v) => v.id !== id)
  else { const v = vazifalar.find((x) => x.id === id); v.bajarildi = !v.bajari
  chiz();
});

// --- Foydalanuvchini taqlid qilamiz ---
for (const matn of ["Non olish", "Dars qilish", "   "]) {
  kirit.value = matn;
  forma.dispatchEvent(new Event("submit", { cancelable: true }));
}
console.log("Qo'shildi:", vazifalar.map((v) => v.matn));
ro.querySelector("span").click();                    // 1-vazifani bosish
ro.querySelectorAll(".o")[1].click();                // 2-vazifani o'chirish
console.log(ro.innerHTML);
```

Natija:

```
Qo'shildi: [ 'Non olish', 'Dars qilish' ]
<li data-id="1" class="tayyor"><span>Non olish</span> <button class="o">×</but
```

1.5. 4-qadam: localStorage bilan saqlash

Sahifa yangilansa ma'lumot yo'qolmasligi uchun holatni `localStorage` ga yozamiz (16-mavzu: faqat matn saqlanadi, shuning uchun `JSON`).

4_qadam.js

```
const KALIT = "todo-v1";
```

```
function saqla(v) { localStorage.setItem(KALIT, JSON.stringify(v)); }
function yukla() {
  try { return JSON.parse(localStorage.getItem(KALIT)) ?? []; }
  catch { return []; }                                  // buzilgan ma'lumotda
}

saqla([{ id: 1, matn: "Saqlanadigan vazifa", bajarildi: false }]);
console.log("Saqlangan matn:", localStorage.getItem(KALIT));
console.log("Qayta yuklash:", yukla());

localStorage.setItem(KALIT, "{buzuq");
console.log("Buzilgan ma'lumot:", yukla());          // xato bermaydi
```

Natija:

```
Saqlangan matn: [{"id":1,"matn":"Saqlanadigan vazifa","bajarildi":false}]
Qayta yuklash: [ { id: 1, matn: 'Saqlanadigan vazifa', bajarildi: false } ]
Buzilgan ma'lumot: []
```

> **Eslatma**
>
> `localStorage` ni brauzerda ishlating — u foydalanuvchining o'z kompyuterida saqlanadi. Parol va maxfiy ma'lumotni **saqlamang**.

1.6. Loyiha tuzilmasi

Loyiha papka tuzilmasi

```
todo-app/
  index.html
  style.css
  src/
    main.js    — kirish nuqtasi
    store.js   — holat va mantiq
    ui.js      — DOM chizish
    storage.js — localStorage
```

2-rasm. Papka tuzilmasi

Kod ko'payganda hamma narsani bitta faylga yozish noqulay. Vazifalar bo'yicha bo'linadi: **mantiq** (`store.js`), **ekran** (`ui.js`), **saqlash** (`storage.js`), **kirish nuqtasi** (`main.js`). Modullar (17-mavzu) shu uchun kerak.

1.7. Toza kod odatlari

2-jadval. Toza kod

Yomon	Yaxshi	Nega?
`let x = 5; let y = 10;` (ma'nosiz nomlar)	`let yosh = 5; let narx = 10;`	Nomdan ma'no tushunilsin
100 qatorli bitta funksiya	Har biri bitta ishni qiluvchi 10 qatorli funksiyalar	O'qish va tuzatish oson
`if (x == 5)`	`if (x === 5)`	Qat'iy tenglik kutilmagan o'zgarishlarni yo'qotadi
`var`	`const` / `let`	Scope tushunarli
Sehrli sonlar: `if (n > 18)`	`const BALOGAT_YOSHI = 18;`	Ma'no nom bilan
Izohsiz murakkab joy	Qisqa tushuntirish izohi	Keyin o'zingiz tushunasiz

1.8. Xatoni qanday topamiz?

1. Xato xabarini **o'qing** (qaysi fayl, qaysi qator).
2. `console.log` bilan o'zgaruvchilarni tekshiring.
3. Brauzer **Sources** panelida `debugger` / to'xtash nuqtasi qo'ying (14-mavzu).
4. Kodni **kichraytiring**: faqat xatoli qismni qoldiring.
5. Qidiring: xato matnini Google ga yozing.

1.9. Eng ko'p uchraydigan xatolar

3-jadval. Loyiha xatolari

Xato	Belgisi	Yechim
Holatni o'zgartirib `chiz()` ni chaqirmaslik	Ekran yangilanmaydi	O'zgarishdan keyin `chiz()`
`innerHTML` ga foydalanuvchi matnini qo'yish	Xavfsizlik xavfi	`textContent`
Bo'sh matnni tekshirmaslik	Bo'sh vazifalar	`trim()` va `if (!matn)`
`id` ni `Number` ga aylantirmaslik	`dataset.id` — matn, topilmaydi	`Number(li.dataset.id)`
`localStorage` ni o'qimaslik	Yangilanganda hamma narsa yo'qoladi	Boshida `yukla()`

1.9.1. To'liq To-Do ilovasi (hammasi birga)

Endi 4 qadamni bitta dasturga birlashtiramiz: holat, `chiz`, hodisalar va saqlash. Bu — brauzerda ishlaydigan **to'liq** ilova yadrosi. Quyidagi demo foydalanuvchi harakatlarini taqlid qiladi, shuning uchun natijalar haqiqiy:

Shu ilovani qanday takomillashtirish mumkin?

4-jadval. Takomillashtirish g'oyalari

G'oya	Qaysi bilim kerak	Qiyinlik
Filtr: hammasi / faol / bajarilgan	Massiv `filter`, holat	★
Tahrirlash (ikki marta bosish)	Hodisalar, `contentEditable` yoki `input`	★★
Sanalar va muddat	`Date` (10-mavzu), saralash	★★
Serverga ulash	`fetch`, `async/await` (15–16)	★★★
Drag & drop tartiblash	Hodisalar (13-mavzu)	★★★

Quyidagi demo shu ilovaning to'liq ishlaydigan yadrosi:

toliq_todo.js

```
const KALIT = "todo-yakuniy";
let vazifalar = JSON.parse(localStorage.getItem(KALIT) ?? "[]");
const forma = document.querySelector("#forma"), kirit = document.querySelector
const ro = document.querySelector("#ro"), soni = document.querySelector("#soni

function saqla() { localStorage.setItem(KALIT, JSON.stringify(vazifalar)); }

function chiz() {
  ro.innerHTML = "";
  for (const v of vazifalar) {
    const li = document.createElement("li");
    li.dataset.id = v.id;
    li.className = v.bajarildi ? "tayyor" : "";
    li.innerHTML = '<span></span><button class="o">×</button>';
    li.querySelector("span").textContent = v.matn;
    ro.append(li);
  }
  const qoldi = vazifalar.filter((v) => !v.bajarildi).length;
  soni.textContent = `${qoldi} ta qoldi`;
  saqla();                                                  // har o'zgarishda saq
}

forma.addEventListener("submit", (e) => {
  e.preventDefault();
  const matn = kirit.value.trim();
  if (!matn) return;
  vazifalar.push({ id: Date.now() + vazifalar.length, matn, bajarildi: false }
  kirit.value = "";
  chiz();
});
ro.addEventListener("click", (e) => {
  const li = e.target.closest("li"); if (!li) return;
  const id = Number(li.dataset.id);
  if (e.target.matches(".o")) vazifalar = vazifalar.filter((v) => v.id !== id)
```

```
  else vazifalar = vazifalar.map((v) => (v.id === id ? { ...v, bajarildi: !v.b
  chiz();
});
chiz();

// --- Foydalanuvchi harakatlari (taqlid) ---
const yoz = (m) => { kirit.value = m; forma.dispatchEvent(new Event("submit",
yoz("Sut olish"); yoz("Dars qilish"); yoz("<b>xavfli</b>");
ro.querySelector("span").click();
console.log(soni.textContent);
console.log(ro.children[2].querySelector("span").innerHTML, "?
 HTML emas, matn");
console.log("Saqlangan:", JSON.parse(localStorage.getItem(KALIT)).map((v) => v
?" : "")));
```

Natija:

```
2 ta qoldi
&lt;b&gt;xavfli&lt;/b&gt; ? HTML emas, matn
Saqlangan: [ 'Sut olish ?', 'Dars qilish', '<b>xavfli</b>' ]
```

1.9.2. Muammo bilan ishlash: bosqichma-bosqich

Kodni yozgach u darrov ishlamasligi **normal**. Hatto tajribali dasturchilar ham vaqtlarining yarmini xato qidirishga sarflaydi. Muhimi — tartibli yondashuv. Quyidagi misolda xato bor: uni ketma-ket tahlil qilamiz.

xato_tahlili.js

```
const vazifalar = [{ id: 1, matn: "A", bajarildi: false }, { id: 2, matn: "B",

// Xatoli funksiya: id berilganda belgilashi kerak, lekin ishlamayapti
function belgilaXato(royxat, id) {
  const v = royxat.find((x) => x.id === id);
  v.bajarildi = true;
}

// DOM dan id matn bo'lib keladi (dataset.id)
const domdanId = "2";
try { belgilaXato(vazifalar, domdanId); }
catch (e) { console.log("1) Xato:", e.name + " — " + e.message); }

// 2) Sababini topish: turlarni tekshiramiz
console.log("2) typeof domdanId:", typeof domdanId, "| typeof vazifa.id:", typ
console.log("   2 === '2' ?", 2 === "2");

// 3) Tuzatish: Number() bilan o'giramiz va topilmasa xabar beramiz
function belgila(royxat, id) {
  const v = royxat.find((x) => x.id === Number(id));
  if (!v) return false;
  v.bajarildi = true;
  return true;
}
console.log("3) Natija:", belgila(vazifalar, domdanId), belgila(vazifalar, "99
console.log(vazifalar);
```

Natija:

```
1) Xato: TypeError — Cannot set properties of undefined (setting 'bajarildi')
2) typeof domdanId: string | typeof vazifa.id: number
   2 === '2' ? false
3) Natija: true false
[
  { id: 1, matn: 'A', bajarildi: false },
  { id: 2, matn: 'B', bajarildi: true }
]
```

Xulosa

Xato → xabarni o'qish → sababni topish (`console.log`/`typeof`) → tuzatish → **shunga o'xshash holatlarni** ham himoyalash (topilmasa nima bo'ladi?).

1.10. Mashqlar

1. To-Do ilovasining 4 qadamini birlashtirib ishlaydigan sahifa yarating.
2. Bajarilgan vazifalarni soni bilan ko'rsating ("2 / 5 bajarildi").
3. "Hammasini o'chirish" tugmasini qo'shing.
4. Vazifa matnini ikki marta bosib tahrirlash imkonini bering.
5. Vazifalarga muhimlik (yulduzcha) belgisini qo'shing.
6. Qorong'i/yorug' rejim tugmasini yozing va `localStorage` da saqlang.
7. Hisoblagich (counter) ilovasi: +, −, nolga qaytarish.
8. Kod uchun fayl tuzilmasini qog'ozda rejalashtiring.

1.11. O'zingizni tekshiring

5-jadval. Qisqa test

№	Savol	Javob
1	Holat (state) nima?	Ilovaning joriy ma'lumoti
2	Holat o'zgarsa nima qilamiz?	Ekranni qayta chizamiz
3	Event delegation nima?	Ota elementga bitta tinglovchi qo'yish
4	`dataset.id` turi?	Matn (`string`)
5	`localStorage` nima saqlaydi?	Faqat matn
6	Foydalanuvchi matnini ekranga?	`textContent` bilan
7	`===` nega yaxshi?	Qat'iy: turni ham solishtiradi
8	Xatoni topish birinchi qadami?	Xato xabarini o'qish

6 va undan ko'p to'g'ri javob — **2-darajaga** tayyorsiz.

Xulosa

Loyiha: reja → maket → mantiq → test. Ma'lumot bitta joyda (holat), `chiz()` uni ekranga chiqaradi, hodisalar holatni o'zgartiradi, `localStorage` saqlaydi.

Chuqurroq o'rganmoqchimisiz?

Bu mavzuning o'rta va murakkab darajalari JavaScript adaptiv kursida — daraja testi sizga mos materialni o'zi tanlaydi, oxirida sertifikat.