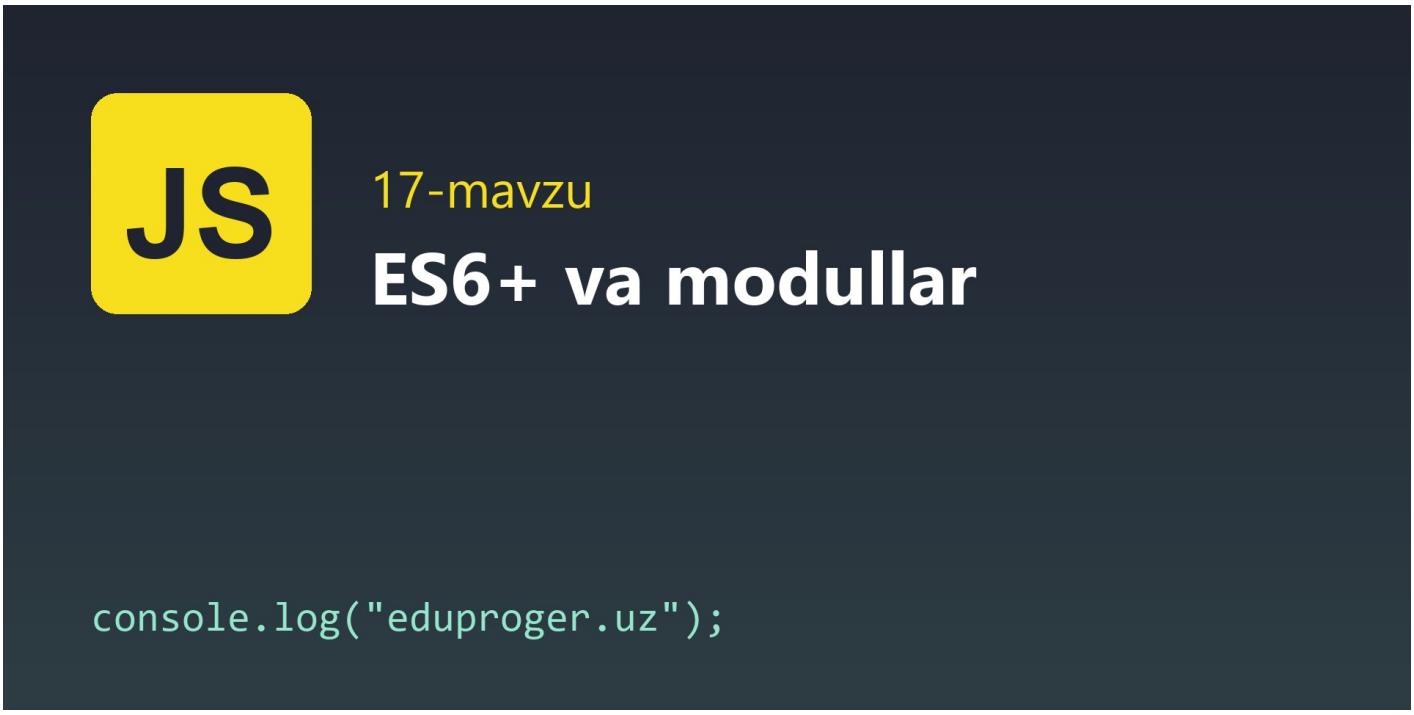
17. ES6+ va modullar

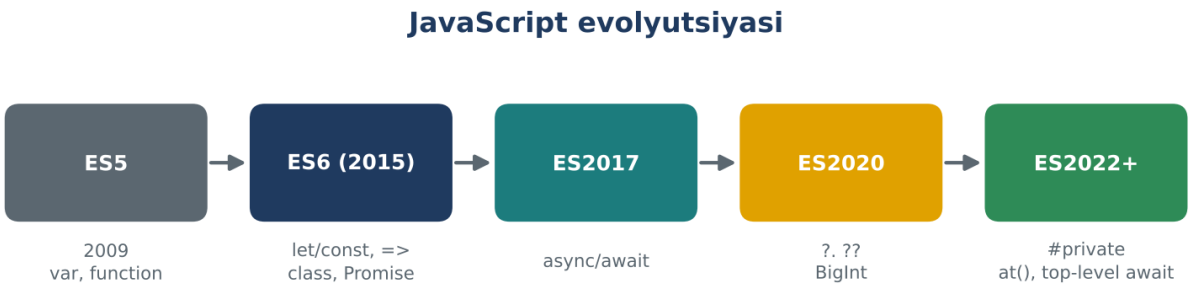


ES6 (2015) JavaScript ni tubdan yangiladi. Bu dars zamonaviy kodda **har kuni** uchraydigan imkoniyatlarni oddiy misollar bilan tushuntiradi.

> **Bu darsda nimani o'rganasiz**
>
> Shablon satrlar (template literals).
>
> Destrukturizatsiya.
>
> Spread (`...`) va rest.
>
> Standart parametrlar.
>
> `?.` va `??` operatorlari.
>
> `class` ga kirish.
>
> `import` va `export` asoslari.

1.1. ES6+ nima?

ECMAScript — JavaScript standarti. Har yili yangi versiya chiqadi. **ES6 (ES2015)** eng katta yangilanish bo'ldi: `let/const`, o'q funksiyalar, `class`, Promise, modullar. Shundan keyin har yili kichik qo'shimchalar keladi — ularni birgalikda **ES6+** deymiz.



1-rasm. JavaScript evolyutsiyasi

Siz `let`, `const`, o'q funksiyalar va `async/await` ni allaqachon bilasiz. Bu mavzuda qolgan eng muhim imkoniyatlarni o'rganamiz.

1.2. Shablon satrlar

Teskari tirnoq (`` ` ``) ichida o'zgaruvchilarni `${...}` bilan joylash va satrni bir necha qatorga yozish mumkin.

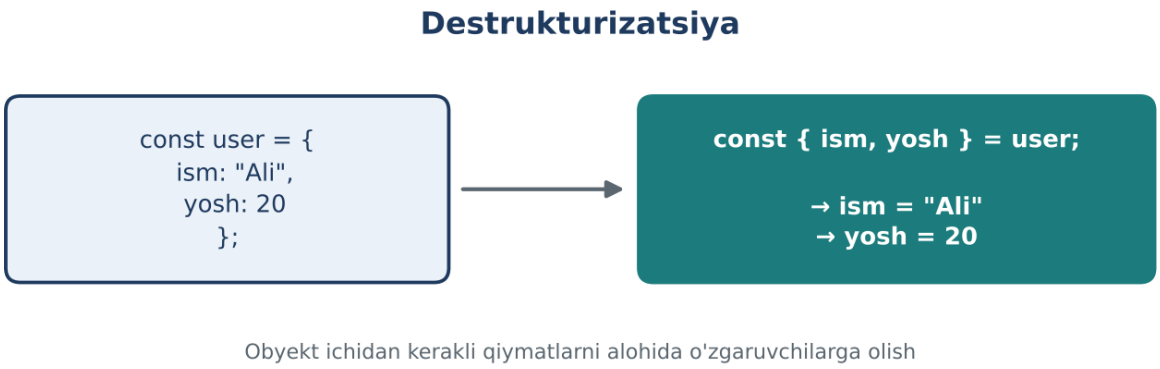
```
const ism = "Ali", yosh = 20;

// Eski usul
console.log("Ismim " + ism + ", yoshim " + yosh);

// Shablon satr
console.log(`Ismim ${ism}, yoshim ${yosh}`);
console.log(`Ertaga ${yosh + 1} yoshda bo'laman`);       // ifoda ham mumkin

// Ko'p qatorli
const xat = `Salom, ${ism}!
Bugun yaxshi kun.
Xayr.`;
console.log(xat);
```

Natija:

```
Ismim Ali, yoshim 20
Ismim Ali, yoshim 20
Ertaga 21 yoshda bo'laman
Salom, Ali!
Bugun yaxshi kun.
Xayr.
```

1.3. Destrukturizatsiya

Obyekt yoki massiv ichidan qiymatlarni **qisqa yozuvda** alohida o'zgaruvchilarga olish.



2-rasm. Destrukturizatsiya

Obyekt destrukturizatsiyasi

```
const user = { ism: "Ali", yosh: 20, shahar: "Toshkent" };

// Eski usul
const ism1 = user.ism;

// Yangi usul: kalit nomi bilan
const { ism, yosh } = user;
console.log(ism, yosh);

// Boshqa nom berish va standart qiymat
const { shahar: manzil, tel = "yo'q" } = user;
console.log(manzil, tel);
```

Natija:

```
Ali 20
Toshkent yo'q
```

Massiv destrukturizatsiyasi

```
const ranglar = ["qizil", "yashil", "ko'k"];
const [birinchi, ikkinchi] = ranglar;                // joylashuv bo'yicha
console.log(birinchi, ikkinchi);

const [, , uchinchi] = ranglar;                      // o'tkazib yuborish
console.log(uchinchi);

// Ikki qiymatni almashtirish
let a = 1, b = 2;
[a, b] = [b, a];
console.log(a, b);
```

Natija:

```
qizil yashil
ko'k
2 1
```

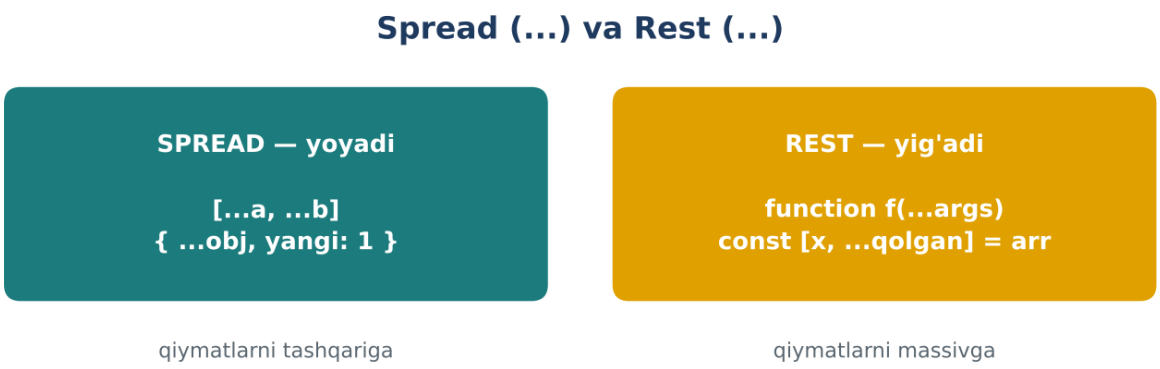
Funksiya parametrlarida

```
function salom({ ism, yosh = 18 }) {
  return `${ism}, ${yosh} yosh`;
}
console.log(salom({ ism: "Vali" }));
console.log(salom({ ism: "Ali", yosh: 25 }));
```

Natija:

```
Vali, 18 yosh
Ali, 25 yosh
```

1.4. Spread va Rest (...)



3-rasm. Spread va Rest

Uch nuqta (`...`) ikki xil ma'noda keladi: **yoyish** (spread) va **yig'ish** (rest).

```
// SPREAD: massivni yoyish
const a = [1, 2], b = [3, 4];
console.log([...a, ...b, 5]);
console.log(Math.max(...[3, 9, 4]));

// SPREAD: obyektni nusxalash va birlashtirish
const user = { ism: "Ali", yosh: 20 };
const yangi = { ...user, yosh: 21, shahar: "Buxoro" };   // yosh yangilandi
console.log(yangi);
console.log(user);                                        // asl obyekt o'zgar

// REST: qolganlarini yig'ish
function yigindi(...sonlar) {
  return sonlar.reduce((j, x) => j + x, 0);
}
console.log(yigindi(1, 2, 3, 4));

const [bosh, ...qolgan] = [10, 20, 30];
console.log(bosh, qolgan);
```

Natija:

```
[ 1, 2, 3, 4, 5 ]
9
{ ism: 'Ali', yosh: 21, shahar: 'Buxoro' }
{ ism: 'Ali', yosh: 20 }
10
10 [ 20, 30 ]
```

Eslatma

Spread faqat **yuza** (1-daraja) nusxa oladi. Ichma-ich obyektlar hali ham umumiy: ularni `structuredClone` bilan nusxalang.

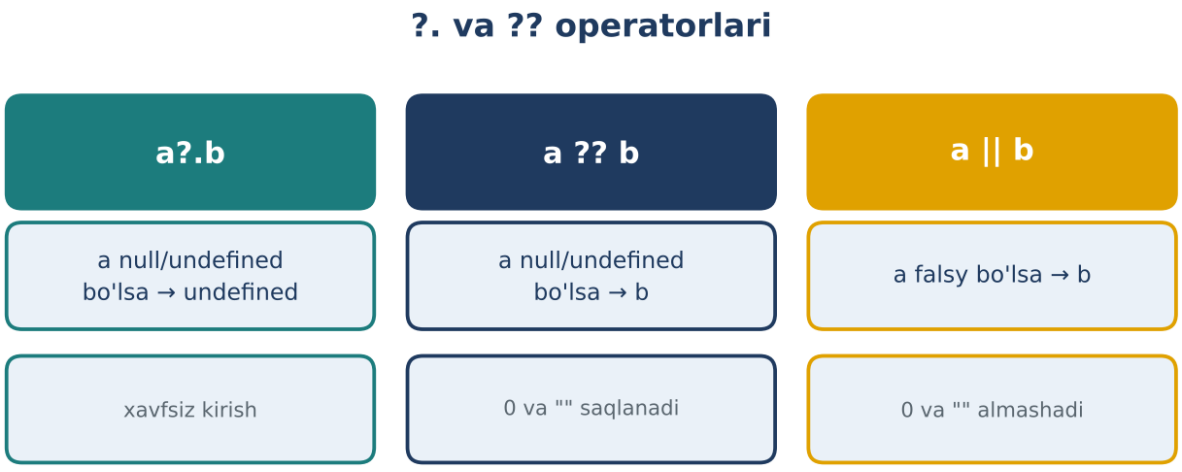
1.5. Standart parametrlar

```
function salom(ism = "mehmon", belgi = "!") {
  return `Salom, ${ism}${belgi}`;
}
console.log(salom());
console.log(salom("Ali"));
console.log(salom("Ali", "?"));
console.log(salom(undefined, "."));      // undefined — standart ishlaydi
```

Natija:

```
Salom, mehmon!
Salom, Ali!
Salom, Ali?
Salom, mehmon.
```

1.6. ?. va ?? operatorlari



4-rasm. ?. va ??

Optional chaining (`?.`) — qiymat `null/undefined` bo'lsa, xato bermay `undefined` qaytaradi. **Nullish coalescing (`??`)** — faqat `null/undefined` bo'lsa, zaxira qiymatni beradi.

```
const user = { ism: "Ali", manzil: { shahar: "Toshkent" } };

// Eski usul: tekshirib borish
console.log(user && user.manzil && user.manzil.shahar);

// ?. bilan
console.log(user.manzil?.shahar);
console.log(user.tel?.raqam);                 // tel yo'q ?
 undefined (xato emas)
console.log(user.salom?.());                  // metod yo'q ? undefined

// || va ?? farqi
const soni = 0, nom = "";
console.log(soni || 10, nom || "anonim");     // 0 va "" — falsy, almashtiril
console.log(soni ?? 10, nom ?? "anonim");     // faqat null/undefined almasht
```

Natija:

```
Toshkent
Toshkent
undefined
undefined
10 anonim
0
```

> **Qoida**
>
> "Qiymat bo'lmasa" desangiz `??` ishlating (`0`, `false`, `""` — to'g'ri qiymatlar). `||` faqat "falsy bo'lsa har qanday holatda" kerak bo'lganda.

1.7. class ga kirish

`class` — bir xil tuzilishdagi obyektlar uchun **qolip**. 9-mavzudagi konstruktor funksiyalarining qulay yozuvi.

```
class Talaba {
  constructor(ism, ball) {
    this.ism = ism;
    this.ball = ball;
  }
  baho() {
    return this.ball >= 60 ? "yaxshi" : "o'rtacha";
  }
}

const t1 = new Talaba("Ali", 75);
console.log(t1.ism, t1.baho());
console.log(t1 instanceof Talaba, typeof Talaba);   // class — aslida funksiya

// Meros olish
class Aspirant extends Talaba {
  constructor(ism, ball, mavzu) { super(ism, ball); this.mavzu = mavzu; }
  baho() { return super.baho() + " (aspirant)"; }
}
const a = new Aspirant("Vali", 50, "AI");
```

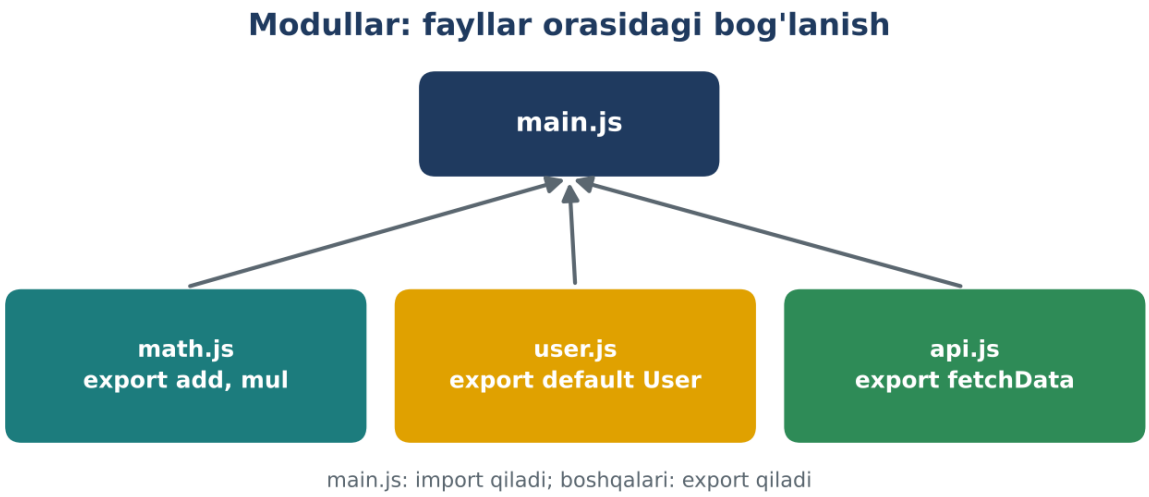
```
console.log(a.baho(), a.mavzu);
```

Natija:

```
Ali yaxshi
true function
o'rtacha (aspirant) AI
```

1.8. Modullar: import va export

Katta dasturni **bir nechta faylga** bo'lish mumkin. Har bir fayl — **modul**. Modul o'z qismlarini `export` qiladi, boshqa fayl ularni `import` qiladi.



5-rasm. Modullar

math.js

```
// math.js
export const PI = 3.14;
export function qosh(a, b) { return a + b; }
export default function ayir(a, b) { return a - b; }
```

main.js

```
// main.js
import ayir, { PI, qosh } from "./math.js";   // default: figurasiz, nomlilari

console.log(PI, qosh(2, 3), ayir(5, 2));
```

index.html

```
<!-- index.html: modulni ulash -->
<script type="module" src="main.js"></script>
```

1-jadval. import/export

Tur	Eksport	Import

Nomli (ko'p bo'lishi mumkin)	`export const a = 1;`	`import { a } from "./fayl.js";`
Default (bitta)	`export default funksiya;`	`import istalganNom from "./fayl.js";`
Nomni o'zgartirish	—	`import { a as b } from ...`
Hammasini olish	—	`import * as m from ...` → `m.a`

Eslatma

Brauzerda modullar faqat **server orqali** ochilganda ishlaydi (`file://` emas). VS Code da "Live Server" kengaytmasi yetarli. Fayl manzili **`./`** bilan boshlanishi va kengaytmasi (`.js`) yozilishi kerak.

1.9. Eng ko'p uchraydigan xatolar

2-jadval. Tipik xatolar

Xato	Sababi	To'g'ri yozuv
`import { x } from "math"`	Nisbiy yo'l `./` yo'q	`"./math.js"`
Default ni figurada olish	Default figurasiz olinadi	`import f from ...`
`type="module"` ni unutish	`import` ishlamaydi	`<script type="module">`
Spread bilan chuqur nusxa deb o'ylash	Faqat yuza nusxa	`structuredClone(obj)`
`||` bilan `0` ni yo'qotish	`0` falsy	`??`
Destrukturizatsiyada `undefined` ob'ektdan	`TypeError`	Standart `= {}` bering

```
function f({ a, b } = {}) { return [a, b]; }    // argument berilmasa ham xato
console.log(f());
try { const { x } = null; } catch (e) { console.log(e.name); }
```

Natija:

```
[ undefined, undefined ]
TypeError
```

1.10. Mini loyiha

```
// Zamonaviy sintaksis bilan ro'yxat
const talabalar = [
  { ism: "Ali", ball: 82, kurs: { raqam: 2 } },
  { ism: "Vali", ball: 55 },
  { ism: "Sardor", ball: 0, kurs: { raqam: 1 } },
```

```
];

const qatorlar = talabalar.map(({ ism, ball, kurs }) => {
  const k = kurs?.raqam ?? "noma'lum";
  return `${ism.padEnd(8)} | ${String(ball).padStart(3)} | kurs: ${k}`;
});
console.log(qatorlar.join("\n"));

const [eng, ...boshqalar] = [...talabalar].sort((a, b) => b.ball - a.ball);
console.log("Eng yuqori:", eng.ism, "| boshqalar soni:", boshqalar.length);
```

Natija:

```
Ali      |  82 | kurs: 2
Vali     |  55 | kurs: noma'lum
Sardor   |   0 | kurs: 1
Eng yuqori: Ali | boshqalar soni: 2
```

1.11. Mashqlar

1. Shablon satr bilan "Ism, yosh, shahar" kartochkasini chiqaring.
2. Obyektdan 3 ta xossani destrukturizatsiya bilan oling.
3. Massivdan birinchi va qolganlarini ajrating (`[bosh, ...qolgan]`).
4. Ikki o'zgaruvchi qiymatini almashtiring.
5. `yigindi(...sonlar)` funksiyasini yozing.
6. `a?.b?.c` bilan chuqur xossani xavfsiz o'qing.
7. `||` va `??` farqini `0` bilan ko'rsating.
8. `class Hayvon` va `class It extends Hayvon` yozing.

1.12. O'zingizni tekshiring

3-jadval. Qisqa test

№	Savol	Javob
1	Shablon satr belgisi?	Teskari tirnoq `` ` ``
2	O'zgaruvchini satrga joylash?	`${ism}`
3	`const { a } = obj` nima qiladi?	`obj.a` ni a ga oladi
4	`...` spread va rest farqi?	Yoyadi / yig'adi
5	`a?.b` nima?	a bo'sh bo'lsa xato bermay `undefined`
6	`??` qachon almashtiradi?	`null` yoki `undefined` bo'lsa
7	Meros olish kalit so'zi?	`extends`
8	Moduldan nimadir olish?	`import`

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

Xulosa

Shablon satr, destrukturizatsiya, spread/rest, `?.`/`??`, `class` va modullar — zamonaviy JavaScript ning kundalik vositalari. Ular kodni qisqa va o'qilishi oson qiladi.

Chuqurroq o'rganmoqchimisiz?

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