6. Funksiyalar



Funksiya — dasturlashning eng muhim tushunchasi. U **bir marta yozilib, ko'p marta ishlatiladigan kod bo'lagi**. Bu dars funksiyani noldan, hayotiy misollar bilan tushuntiradi.

> **Bu darsda nimani o'rganasiz**
>
> Funksiya nima va nima uchun kerak.
>
> Funksiyani e'lon qilish va chaqirish.
>
> Parametr va argument farqi.
>
> `return` bilan qiymat qaytarish.
>
> Standart qiymat berish.
>
> Funksiya ichidagi (lokal) o'zgaruvchilar.
>
> Arrow funksiya bilan tanishuv.

1.1. Funksiya nima?

Oshxonadagi **blender**ni tasavvur qiling: siz ichiga mevalarni solasiz (**kirish**), tugmani bosasiz va tayyor sharbat olasiz (**chiqish**). Blenderning ichida nima sodir bo'lishi sizni

qiziqtirmaydi. Faqat nima solsangiz, nima chiqishini bilasiz.

Funksiya ham xuddi shunday: ichiga **ma'lumot** beriladi, u **ish bajaradi** va **natija** qaytaradi.



1-rasm. Funksiya — kirish va chiqishi bor mashina

Nega funksiya kerak? Uch sabab:

- **Takrorlanmaslik.** Bir kodni 10 joyda yozish o'rniga bir marta yozib, 10 marta chaqirasiz.
- **Tartib.** Katta dastur kichik, nomi bor bo'laklarga bo'linadi.
- **O'zgartirish oson.** Xato topilsa, faqat bitta joyda tuzatasiz.

Siz funksiyalarni allaqachon ishlatgansiz: `console.log(...)`, `alert(...)`, `Number(...)` — hammasi funksiya. Endi o'zimizniki yozamiz.

1.2. Funksiyani e'lon qilish va chaqirish

```
// 1) E'lon qilish (yaratish)
function salomBer() {
  console.log("Assalomu alaykum!");
}

// 2) Chaqirish (ishga tushirish)
salomBer();
salomBer();
salomBer();
```

Natija:

```
Assalomu alaykum!
Assalomu alaykum!
Assalomu alaykum!
```

Ikki alohida qadam bor. **E'lon qilish** — funksiyani yaratish: u hali hech narsa qilmaydi. **Chaqirish** — nomini qavs bilan yozish: shundagina ishlaydi.

Eng ko'p uchraydigan xato

`salomBer` deb yozish — funksiyani **chaqirmaydi**. Chaqirish uchun qavs shart: `salomBer()`. Qavssiz yozuv faqat funksiyaning o'ziga murojaat.

1.3. Parametrlar va argumentlar

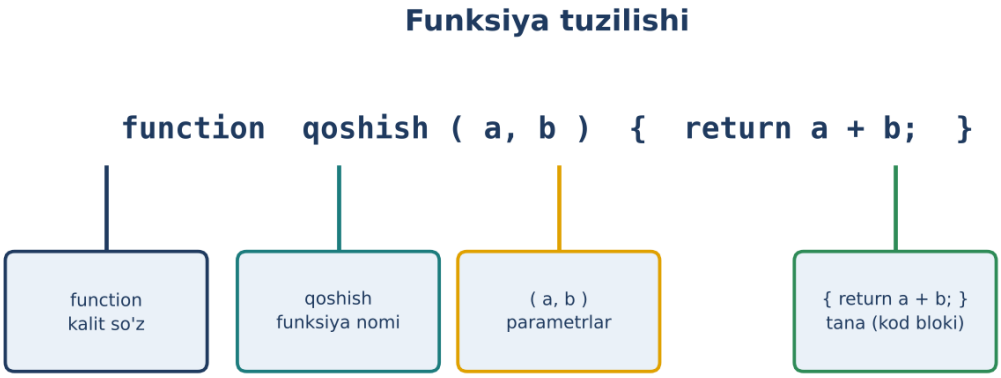
Funksiya har safar bir xil ishni emas, **berilgan ma'lumotga qarab** ish qilishi uchun parametrlar ishlatiladi.

```
function salomBer(ism) {          // ism — parametr
  console.log(`Salom, ${ism}!`);
}

salomBer("Ali");                  // "Ali" — argument
salomBer("Vali");
salomBer("Zarina");
```

Natija:

```
Salom, Ali!
Salom, Vali!
Salom, Zarina!
```



2-rasm. Funksiya tuzilishi

1-jadval. Parametr va argument

Atama	Qayerda	Ma'nosi	Misol
Parametr	E'londa, qavs ichida	Funksiya kutayotgan ma'lumot **nomi** (bo'sh quti)	`function f(ism)` dagi `ism`
Argument	Chaqirishda, qavs ichida	Haqiqatda berilgan **qiymat**	`f("Ali")` dagi `"Ali"`

Bir nechta parametr vergul bilan ajratiladi. Argumentlar **tartib bo'yicha** mos tushadi:

```
function tanishtir(ism, yosh, shahar) {
  console.log(`${ism}, ${yosh} yosh, ${shahar}dan`);
}

tanishtir("Ali", 20, "Toshkent");
tanishtir("Zarina", 22, "Samarqand");
```

Natija:

```
Ali, 20 yosh, Toshkentdan
Zarina, 22 yosh, Samarqanddan
```

1.4. return: natija qaytarish

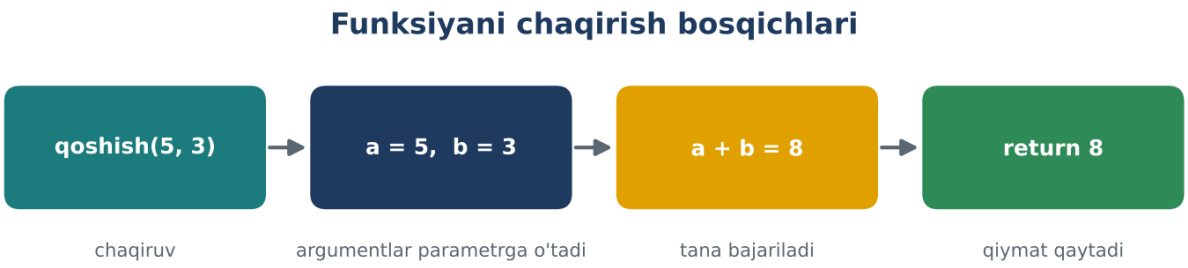
Hozirgacha funksiya natijani **ekranga chiqarardi**. Lekin ko'pincha natijani **boshqa joyda ishlatish** kerak. Buning uchun `return` bor: u qiymatni funksiyadan "tashqariga beradi".

```
function qoshish(a, b) {
  return a + b;
}

const natija = qoshish(5, 3);   // 8 qaytdi va natija ga tushdi
console.log(natija);
console.log(qoshish(10, 20));
console.log(qoshish(1, 2) * 10);   // natijani yana hisoblashda ishlatdik
```

Natija:

```
8
30
30
```



3-rasm. Funksiya chaqirilganda nima bo'ladi

console.log va return farqi

2-jadval. console.log va return

	`console.log(x)`	`return x`
Nima qiladi	Ekranga yozadi (inson ko'rishi uchun)	Qiymatni funksiyadan tashqariga beradi (dastur ishlatishi uchun)

	console.log(x)	**return x**
Natijani saqlash	Mumkin emas	Mumkin: `const r = f()`
Misol	Chop etish	Hisob-kitob natijasi

```
function kvadratChop(n) {
  console.log(n * n);        // faqat chop etadi
}
function kvadrat(n) {
  return n * n;              // qiymat qaytaradi
}

const a = kvadratChop(4);
const b = kvadrat(4);
console.log("a =", a);       // undefined! hech narsa qaytarilmadi
console.log("b =", b);
```

Natija:

```
16
a = undefined
b = 16
```

> **Muhim qoida**
>
> `return` ga yetilganda funksiya **darrov tugaydi**. `return` dan keyingi qatorlar bajarilmaydi. `return` yozilmasa, funksiya `undefined` qaytaradi.

```
function tekshir(son) {
  if (son > 0) {
    return "musbat";
  }
  return "musbat emas";
  console.log("bu qator hech qachon ishlamaydi");
}
console.log(tekshir(5));
console.log(tekshir(-2));
```

Natija:

```
musbat
musbat emas
```

1.5. Standart qiymat

Agar argument berilmasa, parametr `undefined` bo'ladi. Buning oldini olish uchun **standart qiymat** beriladi.

```
function salom(ism = "mehmon") {
  console.log(`Salom, ${ism}!`);
}

salom("Ali");
```

```
salom();           // argument yo'q, standart qiymat ishlaydi
```

Natija:

```
Salom, Ali!
Salom, mehmon!
```

1.6. Funksiya ichidagi o'zgaruvchilar

Funksiya ichida e'lon qilingan o'zgaruvchi faqat **shu funksiyaning ichida** mavjud. Funksiya tugagach, u yo'qoladi. Bu **lokal o'zgaruvchi** deyiladi.

```
function hisobla() {
  const ichki = 100;           // faqat shu yerda mavjud
  console.log("ichkarida:", ichki);
}

hisobla();

try {
  console.log(ichki);          // tashqaridan ko'rinmaydi
} catch (e) {
  console.log("Xato:", e.message);
}
```

Natija:

```
ichkarida: 100
Xato: ichki is not defined
```

> **Foydasi**
>
> Lokal o'zgaruvchilar tufayli turli funksiyalarda bir xil nomdagi o'zgaruvchilarni to'qnashuvsiz ishlatish mumkin. Funksiya "yopiq xona" kabi.

1.7. Arrow funksiya (tanishuv)

Zamonaviy JavaScriptda funksiyani qisqaroq yozish usuli bor: **arrow (o'q) funksiya**. U => belgisi bilan yoziladi.

3-jadval. Ikki yozuv usuli

Oddiy funksiya	Arrow funksiya
`function qoshish(a, b) { return a + b; }`	`const qoshish = (a, b) => a + b;`
`function ikkiBaravar(n) { return n * 2; }`	`const ikkiBaravar = n => n * 2;`
`function salom() { console.log("Hi"); }`	`const salom = () => console.log("Hi");`

```
const qoshish = (a, b) => a + b;
const ikkiBaravar = (n) => n * 2;

console.log(qoshish(3, 4));
console.log(ikkiBaravar(21));
```

Natija:

```
7
42
```

> **Eslab qoling**
>
> Hozircha oddiy `function` bilan yozishni mashq qiling. Arrow funksiyani keyingi darajalarda va ES6 mavzusida chuqur o'rganamiz.

1.8. Kod izi

```
function ikkiBaravar(n) {
  return n * 2;
}

const x = ikkiBaravar(5);
const y = ikkiBaravar(x);
console.log(y);
```

4-jadval. Kod izi

Qadam	Nima bo'ladi	Natija
1	`ikkiBaravar(5)` chaqirildi: `n = 5`, `5 * 2` qaytdi	`x = 10`
2	`ikkiBaravar(x)` chaqirildi: `n = 10`, `10 * 2` qaytdi	`y = 20`
3	`console.log(y)`	`20` chiqadi

1.9. Mini loyihalar

funksiyalar.js

```
// 1) Kvadrat yuzi
function kvadratYuzi(tomon) {
  return tomon * tomon;
}
console.log(kvadratYuzi(7));

// 2) Yoshni hisoblash
function yoshHisobla(tugilganYil, hozirgiYil = 2026) {
  return hozirgiYil - tugilganYil;
}
console.log(yoshHisobla(2000));

// 3) Juftmi?
function juftMi(son) {
```

```
  return son % 2 === 0;
}
console.log(juftMi(8), juftMi(7));

// 4) Eng kattasi
function kattasi(a, b) {
  return a > b ? a : b;
}
console.log(kattasi(15, 9));

// 5) Harorat o'zgartirish
function selsiydanFarengeytga(c) {
  return (c * 9) / 5 + 32;
}
console.log(selsiydanFarengeytga(100));
```

Natija:

```
49
26
true false
15
212
```

Qo'shimcha vazifa

`doiraYuzi(radius)` funksiyasini yozing (`3.14 * radius * radius`). Uni radiusi 1, 5, 10 bo'lgan doiralar uchun chaqiring.

1.10. Eng ko'p uchraydigan xatolar

5-jadval. Tipik xatolar

Xato	Sababi	To'g'ri yozuv
`salom;` yozib funksiyani ishlatmoqchi bo'lish	Qavs yo'q, funksiya chaqirilmadi	`salom();`
`console.log` qilib, natijani o'zgaruvchiga olmoqchi bo'lish	`console.log` qiymat qaytarmaydi (`undefined`)	`return` ishlating
`return` dan keyin kod yozish	U hech qachon bajarilmaydi	Kodni `return` dan oldin yozing
Funksiya ichidagi o'zgaruvchini tashqarida ishlatish	Lokal o'zgaruvchi tashqarida ko'rinmaydi	Natijani `return` bilan chiqaring
Argument tartibini adashtirish	Qiymatlar tartib bo'yicha mos tushadi	Parametrlar tartibiga rioya qiling
Bir nomli ikkita funksiya yozish	Keyingisi oldingisini bosib ketadi	Har funksiyaga alohida nom

1.11. Mashqlar

1. Ismingizni chiqaradigan `ismim()` funksiyasini yozing va uch marta chaqiring.
2. `kub(n)` funksiyasini yozing (n ning kubi: `n * n * n`). `kub(3)` natijasi nima?
3. Ikki sonning ayirmasini qaytaruvchi funksiya yozing.
4. `salom(ism, til)` funksiyasi yozing: til `"uz"` bo'lsa "Salom", `"en"` bo'lsa "Hello" desin.
5. Uchburchak yuzi funksiyasi: `uchburchakYuzi(asos, balandlik)`.
6. Sonning musbat, manfiy yoki nol ekanligini matn sifatida qaytaruvchi funksiya.
7. `qoshish(2, 3)` va `qoshish(2)` natijasini oldindan ayting (ikkinchisi nega `NaN`?). Standart qiymat bilan tuzating.
8. Massiv uzunligini emas, matn uzunligini qaytaruvchi `uzunlik(matn)` funksiyasini yozing (`.length`).

Javoblar

2) 27. 7) `qoshish(2)` da `b = undefined`, `2 + undefined = NaN`. `function qoshish(a, b = 0)` bilan tuzatiladi.

1.12. O'zingizni tekshiring

6-jadval. Qisqa test

№	Savol	Javob
1	Funksiya nima?	Qayta ishlatiladigan nomli kod bo'lagi
2	Funksiya qanday chaqiriladi?	Nomi va qavs bilan: `f()`
3	Parametr va argument farqi?	Parametr — e'londagi nom, argument — chaqirishdagi qiymat
4	`return` nima qiladi?	Qiymatni qaytarib, funksiyani tugatadi
5	`return` yo'q bo'lsa nima qaytadi?	`undefined`
6	Lokal o'zgaruvchi qayerda ko'rinadi?	Faqat funksiya ichida
7	Standart qiymat qanday beriladi?	`function f(a = 5)`
8	Arrow funksiya belgisi?	`=>`

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

1.13. Kichik lug'at

7-jadval. Lug'at

Atama	Sodda ma'nosi
Funksiya	Nomi bor, qayta ishlatiladigan kod bo'lagi
E'lon qilish	Funksiyani yaratish
Chaqirish	Funksiyani ishga tushirish: `f()`
Parametr	Funksiya kutadigan ma'lumot nomi
Argument	Chaqirishda berilgan haqiqiy qiymat
Return	Natijani qaytarish
Lokal o'zgaruvchi	Faqat funksiya ichida mavjud o'zgaruvchi

Xulosa

Funksiya bir marta yoziladi, ko'p marta chaqiriladi. Parametrlar kirish, `return` chiqish vazifasini bajaradi. Chaqirish uchun qavs shart. Lokal o'zgaruvchilar faqat funksiya ichida ko'rinadi.

Chuqurroq o'rganmoqchimisiz?

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