**Գլուխ 1**

**Վեբ հավելվածի մշակման համար օգտագործված տեխնոլոգիաների նկարագրությունները**

1. **Տվյալների բազա**

**Ընդհանուր տեղեկություններ**

Ժամանակակից աշխարհում մեծածավալ ինֆորմացիայի մշակումը  դժվար է պատկերացնել առանց համակարգիչների։

Ավանդական «թղթյա» տվյալների բազաներն ունեն մի շարք թերություններ, օրինակ՝

* պահանջվում են ֆիզիկական մեծ տարածքներ,
* անհրաժեշտ տվյալները գտնելու համար պահանջվում է բավականաչափ երկար ժամանակ,
* դարակները և կարգացանկերը կարգավորված վիճակում պահպանելու հա-մար պետք է ծավալուն աշխատանք կատարել և մեծ ջանքեր գործադրել։

Տվյալների էլեկտրոնային բազաները թույլ են տալիս լուծել բոլոր այդպիսի խնդիրները։Այդ պատճառով անցյալ դարի 60-ական  թվականներից ինֆորմատիկայի բնագավառում շրջանառության մեջ  մտավ “տվյալների բազա” տերմինը: Տվյալների բա-զան կարելի է դիտարկել որպես կոմպյուտերային մշակում ենթադրող ինֆորմացի-ոն  բազմություն: Սովորաբար այդ տերմինով նշանակում են այնպիսի տվյալների բազ-մություն, որոնք ղեկավարվում են, այսպես կոչված, տվյալների բազաների ղեկավար-ման համակարգի կողմից (այսուհետ՝  ՏԲՂՀ (DBMS) կամ պարզապես՝ տվյալների բա-զաների համակարգ): Նշենք, որ ՏԲՂՀ -ն համակարգչային ծրագիր է, որը թույլ է տալիս՝

ա) հատուկ լեզվի օգնությամμ, որը կանվանենք տվյալների սահմանման լեզու, կառուցել նոր տվյալների բազաներ (ՏԲ) և դրանց սխեմաները (տվյալների տրամաբանական կառուցվածքը),

բ) համապատասխան լեզվի միջոցով, որը կանվանենք հարցումների լեզու, ՏԲ-ում ընտրել և փոփոխել որոշակի ինֆորմացիա,

գ) երկար ժամանակ պահպանել մեծածավալ ինֆորմացիա՝ ապահովելով դրա անվտանգությունը (բացառելով ՏԲ-ի որևէ չարտոնագրված օգտագործում),

դ)  մեկից ավելի օգտագործողներին միաժամանակ դիմել ՏԲ, ընդ որում՝ նրանցից ցանկացածի հարցումը չպետք է ազդի մեկ ուրիշի հարցման վրա (հարցումները պետք է իրացվեն այնպես, ինչպես եթե կատարվեին առանձին առանձին):

Սկզբնական շրջանում ՏԲՂՀ-ները մեծածավալ և թանկարժեք ծրագրային ապահովումներ էին ու կարող էին կիրառվել միայն հզոր համակարգիչների միջոցով: Դա բացատրվում էր մոտ մեկ գիգաբայթ տվյալներ պահելու անհրաժեշտությամբ, ինչն այսօր կարող է տեղավորվել ընդամենը մեկ սկավառակում: Այսպիսով հնարավորու-թյուն կա ՏԲ համակարգն օգտագործել անհատական համակարգիչների միջոցով:

Վերջին տարիներին լայն տարածում են գտել, այսպես կոչված, տվյալնե-րի ռելացիոն մոդելն ու դրա հենքի վրա մշակված ռելացիոն ՏԲՂՀ-ն (RDBMS), որոնց առաջին մատակարարողն էր IBM կորպորացիան: [7,8]

**Տվյալների ռելացիոն բազա**

[Ռելյացիոն](https://hy.wikipedia.org/w/index.php?title=%D5%8F%D5%BE%D5%B5%D5%A1%D5%AC%D5%B6%D5%A5%D6%80%D5%AB_%D5%BC%D5%A5%D5%AC%D5%B5%D5%A1%D6%81%D5%AB%D5%B8%D5%B6_%D5%A2%D5%A1%D5%A6%D5%A1&action=edit&redlink=1) է կոչվում այն տվյալների բազան, որը բաղկացած է տրամաբանորեն փոխկապակցված բաղադրիչներից, որոնք կոչվում են աղյուսակներ։

Աղյուսակը բազաներում տվյալների պահպանման հիմնական միավորն է։ Ռել-յացիոն բազաներում ղեկավարման հատուկ համակարգի միջոցով ապահովվում են տվյալների հետ տարբեր գործառնությունների կատարման հնարավորություններ։ Այդ համակարգերը կոչվում են տվյալների բազայի կառավարման ռելյացիոն համակարգեր - ՏԲԿՌՀ։ ՏԲԿՌՀ-ների առավել խոշոր արտադրողներն են Oracle, Microsoft, Informix, Sybase և IBM ֆիրմաները։

Ռելյացիոն բազաներում աղյուսակների տրամաբանական փոխկապակցվությու-նը (ինչպես սովորաբար ասում են՝ աղյուսակները գտնվում են հարաբերությունների (relations) մեջ և այդ պատճառով կոչվում են աղյուսակ-հարաբերություններ) թույլ է տալիս մեկ հարցման միջոցով գտնել բոլոր անհրաժեշտ տվյալները, որոնք կարող են պարունակվել մի քանի աղյուսակներում։ [7, 8, 9, 10]

**SQL**

SQL-ը կամ հարցումների կառուցվածքավորված լեզուն ստանդարտ լեզու է, որը նշանակված է տվյալների ռելյացիոն բազաների հետ փոխգործակցելու համար։ Դրա նախատիպը մշակվել էր IBM-ում, դոկտոր Կոդդի (E.F.Codd) “Ռելյացիոն մոդելի օգտագործումը տվյալների բաժանված բանկերում” հոդվածի հիման վրա, յոթանասունական թվականներին։ SQL-ի առաջին ծրագրային արտադրանքը՝ Oracle ՏՌԲԿՀ-ն (տվյալների ռելյացիոն բազայի կառավարման համակարգ), ստեղծվեց 1979 թ. Relational Software, Inc. կորպորացիայի (հետագայում այն վերաանվանվեց Oracle Corporation) կողմից։ Ծրագրային արտադրանքի շուկայում այդ ՏՌԲԿՀ-ն մեծ հաջողությունը ստիպեց մյուս խոշոր ֆիրմաներին (Sybase, Informix, Microsoft) թողարկել SQL-ի վրա հիմնված ՏՌԲԿՀ-ների սեփական տարբերակները։ Քանի որ SQL-ը այն ժամանակ նորույթ էր, և դեռ ստեղծված չէին ստանդարտներ, ապա յուրաքանչյուր ֆիրմա արտադրում էր ՏՌԲԿՀ SQL-ի սեփական իրացումով։ Միայն 1989 թ. ANSI-ն (American National Standards Institute) մշակեց SQL-ի համընդհանուր ստանդարտը։ Սակայն այն հաշվի չէր առնում SQL-ի տեխնիկական կառուցվածքը, և անցումը լեզվի մեկ իրացումից մյուսին բավականաչափ բարդ էր։ 1992 թ. ստեղծվեց ստանդարտ, որը ընդգրկում է SQL-ի հնարավորին չափ շատ ընդլայնումները։ Այն անվանում են SQL92 կամ SQL2։ Նշենք, որ SQL2-ի մեծ ծավալի պատճառով ՏՌԲԿՀ-ները, որոնք իրացնում են լրիվ ստանդարտը, բավականաչափ բարդ են և աշխատատար։[9, 10, 11, 15]

**SQL-ի հրամանների տեսակները**

SQL-ն ունի հրամանների վեց հիմնական խմբեր՝

1. DDL (Data Definition Language) - տվյալների բազայի կառուցվածքների սահմանում։
2. DML (Data Manipulation Language) - տվյալների մանիպուլյացիա։
3. DQL (Data Query Language) – տվյալների ընտրում։
4. DCL (Data Control Language) – տվյալների վերահսկման լեզու։
5. DAC (Dedicated Administrator Connection) - Տվյալների կառավարման հրահանգներ։
6. TCC - Տրանզակցիաների վերահսկման հրամաններ։

DDL-ը թույլ է տալիս տվյալների բազայի օգտագործողին ստեղծել եւ վերակառու-ցել տվյալների բազայի օբյեկտները, ինչպիսիք են աղյուսակների ստեղծումը կամ վերացումը: Այդ հրամաններն են՝

* CREATE TABLE
* ALTER TABLE
* DROP TABLE
* CREATE INDEX
* ALTER INDEX
* DROP INDEX
* CREATE VIEW
* DROP VIEW

DML-ը օգտագործվում է ռելացիոն տվյալների բազաների տվյալների կառավար-ման համար։ Այդ հրամաններն են՝

* [INSERT](https://ru.wikipedia.org/wiki/Insert_(SQL))
* UPDATE
* [DELETE](https://ru.wikipedia.org/wiki/Delete_(SQL))

DQL-ը, թեև բաղկացած է միայն մեկ հրամանից, հանդիսանում է SQL- ի առավել օգտագործվող տվյալների բազայի օգտագործողների կողմից: Այդ հրամաններն են՝

* [select](https://ru.wikipedia.org/wiki/Select_(SQL))։

DCL-ը թույլատրում է վերահսկել տվյալների բազայում առկա տվյալների հասա-նելիությունը: DCL հրամանները սովորաբար օգտագործվում են օգտագործողների մուտքի հետ կապված օբյեկտներ ստեղծելու համար, ինչպես նաեւ վերահսկում են օգտվողների շրջանում արտոնությունների բաշխման համար: Այդ հրամաններն են՝

* ALTER PASSWORD
* GRANT
* REVOKE
* CREATE SYNONYM

DAC-ը թույլ է տալիս օգտվողին կատարել աուդիտներ եւ իրականացնել վերլու-ծություններ տվյալների բազայում գործողությունների վերաբերյալ: Նրանք կարող են օգ-տագործվել նաեւ համակարգի վերլուծության համար: Այդ հրամաններն են՝

* START AUDIT
* STOP AUDIT

6. Տրանզակցիաների կառավարման հրամաններն են՝

* commit,
* rollback,
* savepoint,
* set transaction։

SQL լեզվի իրացումների մեծամասնությունը թույլ է տալիս գրանցել հրամանների օպերատորները ինչպես մեծատառերով, օրինակ՝ CREATE TABLE, այնպես էլ փոքրատառերով՝ create table։[9, 10, 11,15]

**MySQL**

MySQL-ը [բաց կոդով](https://hy.wikipedia.org/w/index.php?title=%D4%B2%D5%A1%D6%81_%D5%AF%D5%B8%D5%A4&action=edit&redlink=1) [տվյալների բազաների կառավարման համակարգ](https://hy.wikipedia.org/w/index.php?title=%D5%8F%D5%BE%D5%B5%D5%A1%D5%AC%D5%B6%D5%A5%D6%80%D5%AB_%D5%A2%D5%A1%D5%A6%D5%A1%D5%B6%D5%A5%D6%80%D5%AB_%D5%AF%D5%A1%D5%BC%D5%A1%D5%BE%D5%A1%D6%80%D5%B4%D5%A1%D5%B6_%D5%B0%D5%A1%D5%B4%D5%A1%D5%AF%D5%A1%D6%80%D5%A3&action=edit&redlink=1)։ MySQL-ի մշակումն ու աջակցումն իրականացվում է [Oracle](https://hy.wikipedia.org/w/index.php?title=Oracle&action=edit&redlink=1) կորպորացիայի կողմից։ Վերջինս MySQL-ի նկատմամաբ իրավունքները ստացել է [Sun Microsystems](https://hy.wikipedia.org/w/index.php?title=Sun_Microsystems&action=edit&redlink=1) ընկերության գնումից հետո, որն էլ, նախքան այդ, իր հերթին գնել էր [MySQL-ը AB](https://hy.wikipedia.org/w/index.php?title=MySQL_AB&action=edit&redlink=1) ընկերությունից։ MySQL-ն տարածվում է [GNU GPL](https://hy.wikipedia.org/w/index.php?title=GNU_GPL&action=edit&redlink=1), ինչպես նաև սեփական կոմերցիոն արտոնագրի ներքո, չնայած մշակողներն նոր ֆունկցիաներ են ավելացնում արտոնագրված օգտագործողների պատվերներով։ MySQL-ը հիմնականում օգտագործվում է փոքր և միջին ծանրաբեռնվա-ծության ծրագրերի համար։ Այն ներառված է մի շարք վեբ-ծրագրավորման համար նախատեսված ծրագրերի փաթեթների մեջ, ինչպիսիք են [WAMP](https://hy.wikipedia.org/w/index.php?title=WAMP&action=edit&redlink=1)-ը, [AppServ](https://hy.wikipedia.org/w/index.php?title=AppServ&action=edit&redlink=1)-ը, [LAMP](https://hy.wikipedia.org/w/index.php?title=LAMP&action=edit&redlink=1)-ը, [XAMPP](https://hy.wikipedia.org/w/index.php?title=XAMPP&action=edit&redlink=1)-ը և [Open Server](https://hy.wikipedia.org/w/index.php?title=Open_Server&action=edit&redlink=1)-ը։

MySQL բավականին ճկուն է։ Այն աջակցում է աղյուսակների այնպիսի տեսակ-ներ, ինչպիսիք են [MyISAM](https://hy.wikipedia.org/w/index.php?title=MyISAM&action=edit&redlink=1)-ը կամ [InnoDB](https://hy.wikipedia.org/w/index.php?title=InnoDB&action=edit&redlink=1)-ն։ Բացի այդ, այն տեղադրվում է հա-տուկ EXAMPLE անվանմամբ աղյուսակի տեսակով, որն ցուցադրում է նոր աղյուսակի տիպեր ստեղծելու սկզբունքները։ GPL արտոնագրի շնորհիվ, MySQL-ն անընդհատ հարստանում է նոր աղյուսակի տեսակներով։

* 1. **HTML**

**Պատմություն**

HTML([անգլ.](https://hy.wikipedia.org/wiki/%D4%B1%D5%B6%D5%A3%D5%AC%D5%A5%D6%80%D5%A5%D5%B6)՝ HyperText Markup Language, գիպերտեքստերի նշագրման լեզու)-ն ստեղծվել է բրիտանացի գիտնական [Թիմ Բերներս-Լիի](https://hy.wikipedia.org/w/index.php?title=%D4%B9%D5%AB%D5%B4_%D4%B2%D5%A5%D6%80%D5%B6%D5%A5%D6%80%D5%BD-%D4%BC%D5%AB&action=edit&redlink=1) կողմից [1991](https://hy.wikipedia.org/wiki/1991)-[1992](https://hy.wikipedia.org/wiki/1992) թթ.։ Այն ծագել է SGML նշման լեզվից և հանդիսանում է նշագրման լեզու, որն օգտագործվում է [վեբ-կայքերի](https://hy.wikipedia.org/wiki/%D5%8E%D5%A5%D5%A2-%D5%A7%D5%BB) մեծամասնության նշագրման համար։ HTML լեզուն վերամշակվում է դիտարկ-չի կողմից և ցուցադրվում է մարդու կողմից հեշտ ընթեռնելի փաստաթղթի տեսքով։ HTML-ի օգնությամբ հեշտությամբ կարելի է ստեղծել համեմատաբար պարզ, սակայն գեղեցիկ ձևավորված փասթաթուղթ։ Ի հավելումն փասթաթղթի կառուցվածքի պարզեցմանը, HTML-ում աջակցվում էր հիպերտեքստը։ Մուլտիմեդիայի հնարավորությունները ավելացվեցվել են ավելի ուշ։ Ի սկզբանե HTML լեզուն ստեղծվել է որպես փասթաթղթերի կառուցման և ձևավորման միջոց՝ անկախ օգտագործվող արտածման միջոցից (էկրան, տպիչ և այլն) ցուցադրելու միջոց։ Նախնական HTML ձևաչափով տեքստը պետք է առանց ոճային և կառուծվածքային ձևափոխման արտածվեր սարքավորման վրա՝ տարբեր տեխնիկական համալրմամբ։ Սակայն HTML-ի ժամանակակից կիրառումը հեռու է սկզբնական նշանակությունից։

1990-ական թվականի կեսերին առաջ եկավ հետևյալ երևույթը. դիտարկիչների հիմնական արտադրող ընկերությունները՝ [Նեթսքեյփ](https://hy.wikipedia.org/w/index.php?title=%D5%86%D5%A5%D5%A9%D5%BD%D6%84%D5%A5%D5%B5%D6%83&action=edit&redlink=1) և [Մայքրոսոֆթ](https://hy.wikipedia.org/wiki/%D5%84%D5%A1%D5%B5%D6%84%D6%80%D5%B8%D5%BD%D5%B8%D6%86%D5%A9), սկսեցին ներառել HTML-ը տարրերի սեպական հավաքածուներով։ Տարբեր դիտարկիչների անհամատե-ղելիության պատճառով ստեղծվեց խառնաշփոթ, առավելապես բարդություններ էին առաջանում նույն փասթաթուղթը տարբեր դիտարկիչներում նույն տեսքով ստանալու հարցում։ Վեբ մասնագետները ստիպված էին միևնույն էջը մի քանի տարբերակով ստեղծել։ 2000-ականների սկզբին, մի քանի տարով այդ խնդիրը փոքր ինչ կորցրեց իր հրատապությունը՝ կապված Ինտերնետ Էքսպլորերի տարածման և զարգացնամ հետ։ ‌

Այժմ այս խնդիրը լուծվում է [W3C](https://ru.wikipedia.org/wiki/W3C) ստանդարտի միջոցով, որտեղ մշակվում են համաշխարհային սարդոստայնի ստանդարտները։[12, 17]

**HTML փաստաթղթի կառուցվածքը**

HTML լեզվով գրված ցանկացած փաստաթուղթ իրենից ներկայացնում է իրար մեջ ներդրված տարրերի հավաքածու ([անգլ.](https://hy.wikipedia.org/wiki/%D4%B1%D5%B6%D5%A3%D5%AC%D5%A5%D6%80%D5%A5%D5%B6)՝ *HTML element*), ընդ որում բոլոր տարրերի սկիզբը և վերջը նշվում է հատուկ նշիչներով՝ պիտակներով ([անգլ.](https://hy.wikipedia.org/wiki/%D4%B1%D5%B6%D5%A3%D5%AC%D5%A5%D6%80%D5%A5%D5%B6)՝ *tag*)։ Տարրերը կարող են լինել դատարկ, այսինքն՝ չպարունակեն ոչ մի տեքստ կամ տվյալ, օրինակ. <br/>  պիտակը, որը ջարդում է տողը, հաջորդող տեքստը սկսելով նոր տողից։ Այս դեպքում սովորաբար չի նշվում փակող պիտակ։ Բացի այդ, տարրերը կարող են պարունակել հատկանիշներ ([անգլ.](https://hy.wikipedia.org/wiki/%D4%B1%D5%B6%D5%A3%D5%AC%D5%A5%D6%80%D5%A5%D5%B6)՝ HTML ատրիբուտներ), որոնք որոշում են նրանց որոշակի հատկանիշներ, օրնակ. տառաչափը span էլեմենտի համար։ Ատրիբուտները նշվում են բացող պիտակի մեջ։ Օրինակ՝

<span color="blue"> Այս օրինակում ատրիբուտ է հանդիսանում color-ը </span>

Բացի էլեմենտներից, HTML փաստաթղթում օգտագործվում են նաև հատուկ նշաններ, որոնք նշվում են &անուն ձևով, օրինակ. &copy-ն կարտածի հեղինակային իրավունքի նշանը՝ ©։

Յուրաքանչյուր HTML փաստաթուղթ, որ համապատասխանում է HTML ստան-դարտի որևէ տարբերակի, պետք է սկսվի <!DOCTYPE…>-ով, որն հայտարարում է HTML-ի տարբերակը և ունի այս տեսքը՝

Տարբերակ 4․01՝

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/ strict.dtd">

Տարբերակ 5․0՝

<!DOCTYPE html>

Եթե այս տողը նշված չէ, ապա դիտարչկում փաստաթղթի ճիշտ արտածման հասնել դժվար կլինի։ Այնուհետև պետք է գրվեն <html><head><body> պիտակները։ <head> և <body> տարրերը կարող են լինել դատարկ կամ ընդհանրապես չգրվեն։

* 1. **CSS**

**Նկարագրություն**

CSS-ը օգտագործվում է վեբ-էջեր նախագծողների կողմից գույները, տառատե-սակները, գտնվելու վայրը և այլ նկարագրություններ տալու համար։ CSS-ի հիմնական նպատակը հանդիսանում է այն, որ դրա օգնությամբ առանձնացվում են HTML էջի պարունակությունը և նրա նկարագրությունը, դրանով իսկ էջին հաղորդվում է մատչելիություն, ճկունություն և նրա նկարագրությունը կառավարելու հնարավո-րություն, ինչպես նաև նվազեցնում է հրամանների կրկնվելը և բարդացնելը։ Բացի այդ CSS-ը թույլ է տալիս տվյալ փաստաթուղթը ներկայացնել տարբեր ոճերով։ CSS հատկու-թյուններ ունեն հետևյալ տեսքը՝

սելեկտոր, սելեկտոր, ․․․ {

հատկություն: արժեք;

հատկություն: արժեք;

հատկություն: արժեք;

․․․․․․․․․

}

**CSS-ը վեբ փաստաթղթին միացնելու եղանակները**

CSS-ի կանոնները գրված են CSS լեզվով եւ գտնվում են ոճերի աղյուսկում, այսին-քըն, ոճերի աղյուսկում պարունակվում են CSS կանոններ: Այս ոճերի աղյուսակները կարող են տեղադրվել ինչպես վեբ փաստաթղթում, այնպես էլ CSS ձեւաչափով առանձին ֆայլերով: Իրականում, CSS ձեւաչափը պարզ տեքստային ֆայլ է:.css ֆայլը ոչ այլ ինչ է, քան CSS կանոնների եւ մեկնաբանությունների ցանկ։ CSS-ը կարող է ներառվել վեբ-փաստաթղթում չորս եղանակով՝

* Ոճերի աղյուսկը գրվում է առանձին .css ֆայլը, ապա այն վեբ-փաստաթղթին միացվում է <head><head/> էլեմենտում <link> էլեմենտի միջոցով, նրա href հատ-կանիշին փոխանցելով .css ֆայլի հասցեն։

<!DOCTYPE html>

<html>

<head>

.....

<link rel="stylesheet" href="style.css">

</head>

<body>

.....

</body>

</html>

* Ոճերի աղյուսկը գրվում է առանձին .css ֆայլը, ապա այն վեբ-փաստաթղթին միացվում է <head></head> էլեմենտում <style></style> էլեմենտի միջոցով, նրա @import դիրեկտիվի url-ին փոխանցելով .css ֆայլի հասցեն։

<!DOCTYPE html>

<**html**>

<**head**>

.....

<**style** media="all">

@**import** **url**(**style**.**css**);

</**style**>

</**head**>

<**body**>

.....

</**body** >

</**html**>

* Ոճերի աղյուսկը գրվում է վեբ-փաստաթղթի մեջ՝ <head></head> էլեմենտի <style> </style> էլեմենտի մարմնում։

<!DOCTYPE html>

<**html**>

<**head**>

.....

<**style**>

**body** {

**color**: **red**;

}

</**style**>

</**head**>

<**body**>

.....

</**body**>

</**html**>

* Ոճերի աղյուսկը գրվում է վեբ-փաստաթղթի մեջ՝ էլեմենտների style հատկանիշի արժեքի տեսքով։

<!DOCTYPE>

<**html**>

<**head**>

.....

</**head**>

<**body**>

<**p** style="font-size: 20px; color: green; font-family: arial, helvetica, sans-serif">

.....

</**p**>

</**body**>

</**html**>

1. **ՋավաՍկրիպտ**

**Նկարագրություն**

ՋավաՍկրիպտը ինտերպրետացվող սկրիպտային լեզու է։ Աջակցում է օբյեկտ կողմնորո-շված, հրամայական եւ ֆունկցիոնալ ոճերը: Հանդիսանում է ECMAScript լեզվի իրականացում։ Այն համացանցում ամենահանրահայտ սկրիպտային լեզուն է և աշխատում է բոլոր հանրահայտ [բրաուզերներում](https://hy.wikipedia.org/wiki/%D4%B2%D6%80%D5%A1%D5%B8%D6%82%D5%A6%D5%A5%D6%80), որոնց թվում են [Ինտերնետ Էքսպլո-րեր](https://hy.wikipedia.org/wiki/%D4%BB%D5%B6%D5%BF%D5%A5%D6%80%D5%B6%D5%A5%D5%BF_%D4%B7%D6%84%D5%BD%D5%BA%D5%AC%D5%B8%D6%80%D5%A5%D6%80), Մոզիլլա Ֆաիրֆոքս, Գուգլ Խրոմ, Օպերա և [Սաֆարի](https://hy.wikipedia.org/wiki/Safari)։ Այս լեզուն առաջին անգամ կիրառվել է Netscape Communications-ի կողմից Netscape Navigator 2 beta (1995) բրաուզե-րում։ ՋավաՍկրիպտ լեզուն ստեղծվել է [Բրենդան Այկի](https://hy.wikipedia.org/wiki/%D4%B2%D6%80%D5%A5%D5%B6%D5%A4%D5%A1%D5%B6_%D4%B1%D5%B5%D5%AF) կողմից [1995](https://hy.wikipedia.org/wiki/1995)թ.-ին։

ՋավաՍկրիպտը լայն կիրառություն է ստացել բրաուզերներում՝ հաճախորդի կողմում, քանի այն դինամիկություն է հաղորդում վեբ փաստաթղթին, իսկ 2009թ․-ից սկսած, երբ ստեղծվեց node.js ծրագրային հարթակը՝ Գուգլի կողմից ստեղծված v8 շարժիչի հիմքի վրա, ջավաՍկրիպտը լայն կիրառություն ստացավ նաև սերվերի կողմում։

Հիմնական ճարտարապետական առանձնահատկություններն են դինամիկ տի-պայնացումը, թույլ տիպայացումը, ավտոմատ հիշողության կառավարումը, նախատի-պային ծրագրավորումը, ֆունկցիաները որպես առաջին կարգի օբյեկտներ:

ՋավաՍկրիպտի վրա ազդել են բազմաթիվ լեզուներ, սակայն լեզուն ստեղծելիս գլխավոր նպատակն էր այն նմանեցնել Java-ին, բայց միևնույն ժամանակ այն պետք է հեշտ լիներ ոչ ծրագրավորողների համար:

**ՋավաՍկրիպտի կառուցվածքը և հնարավորությունները**

ՋավաՍկրիպտի կառուցվածքը կարելի է բաժանել հետեևյալ 3 մասերի՝

1. միջուկ կամ [ECMAScript](https://hy.wikipedia.org/w/index.php?title=ECMAScript&action=edit&redlink=1)
2. բրաուզերի օբյեկտային մոդել
3. փաստաթղթի օբյեկտային մոդել

ECMAScript-ը չի հանդիսանում բրաուզերային լեզուն եւ չի սահմանում ինֆորմա-ցիայի մուտքի և ելքի մեթոդներ: Այն հանդիսանում է ջավաՍկրիպտ լեզվի ստանդարտը։ ECMAScript ստանդարտը նկարագրում է տվյալների տիպերը, հրամանները, բանալի և պահեստավորված բառերը, օպերատորները, օբյեկտները, կանոնավոր արտահայտու-թյունները և այլն։

Բրաուզերի օբյեկտային մոդելը հանդիսանում է միջուկի և փաստաթղթի օբյեկ-տային մոդելի միջև շերտ․ Բրաուզերի օբյեկտային մոդելի հիմնական նպատակն է՝ բրաուզերի պատուհանների կառավարումը և նրանց փոխազդեցությունների ապահո-վումը։ Բրաուզերի յուրաքանչյուր պատուհան ստեղծվում է window օբյեկտից։ Բացի պատուհանների կառավարումից, բրաուզերի օբյեկտային մոդելը ապահովում է՝

* համակարգային երկխոսություններ,
* բացված էջի հասցեի կառավարումը,
* բրաուզերի մասին տեղեկատվության կառավարում,
* մոնիտորի պարամետրերի մասին տեղեկությունները կառավարում,
* կայքերի դիտումների հիշողության կառավարում,
* HTTP cookie-ի հետ աշխատելու հնարավորություն։

Փաստաթղթի օբյեկտային մոդելը թույլ է տալիս ծրագրին մուտք գործել HTML, XHTML եւ XML փաստաթղթեր, ինչպես նաեւ փոփոխել այդ փաստաթղթերի բովանդա-կությունը, կառուցվածքը և նման փաստաթղթեր ձևավորում։

**Վեբ էջերում ներդրումը**

ՋավաՍկրիպտը կարելի է գրել անմիջապես վեբ փաստաթղթի մեջ՝ կոդը ներառե-լով <script> բացվող և </script> փակվող թեգերի մեջ։ JavaScript-ը կարելի է նաև գրել *.*js

ընդլայնմամբ առանձին ֆայլում։

Եթե ջավաՍկրիպտ կոդը գրված է անմիջապես վեբ փաստաթղթի մեջ, ապա <scri-pt></script> թեգը խորհուրդ է տրվում գրել <head></head> տեգի մեջ՝

<!DOCTYPE html>

<html>

<head>

<title>Օրինակ</title>

<script>

alert("Hello World");

</script>

</head>

<body>

</body>

</html>

Տվյալ կոդը ակտիվացնելիս կբացվի փոքր պատուհան Hello World հաղորդագրությամբ։

Բացի այս տարբերակից ջավաՍկրիպտը կարելի է գրել հենց վեբ փաստաթղթում, առանց <script></script> թեգի։ Օրինակ HTML փաստաթղում այն հնարավոր է գրվել ան-միջապես էլեմենտի վրա՝

<a href="delete.php" onclick="return confirm('համոզված ե՞ք');">

ջնջել

</a>

Սակայն այս տարբերակը համարվում է վատ փորձ և խորհուրդ չի տրվում այն կիրառել։

Եվ վերջին ու ամենաշատ կիրառելի տարբերակը է, երբ ջավաՍկրիպտը գրվում է առանձին .js ընդլայնում ունեցող փաստաթղթում։ .js ընդլայնում ունեցող փաստաթղու-թը վեբ փաստաթղթին միացվում է <script></script> թեգի src հատկանիշին փոխանցելով փաստաթղթի հասցեն՝

<!DOCTYPE html>

<html>

<head>

<title>Օրինակ</title>

<script src=”script.js”>

alert("Hello World");

</script>

</head>

<body>

</body>

</html>