

Դաս 1: JS window, DOM

Window - ն գլոբալ օբյեկտ է, որը ներկայացնում է բրաուզերի պատուհանը, որում ցուցադրվում է մեր html ֆայլը: Այն ապահովում է ինտերֆեյս բրաուզերի, գլոբալ փոփոխականների, ֆունկցիաների և այլ օբյեկտների հետ աշխատելու համար:

JS-ում **window** օբյեկտը գլոբալ օբյեկտ է: Սա նշանակում է, որ ցանկացած գլոբալ փոփոխական կամ ֆունկցիա ինքնավերաբար դառնում են իր հատկությունները:

```
var myVariable = "Hello";  
function test() {  
    return "world!"  
}  
console.log(window.myVariable, window.test());
```

Window օբյեկտը պարունակում է բազմաթիվ հատկություններ և մեթոդներ՝ բրաուզերի պատուհանը կառավարելու, օգտատիրոջ հետ փոխազդելու և այլ տիպի առաջադրանքներ իրականացնելու համար:

Հիմնական հատկություններն են՝

Window.document հղում է **document** օբյեկտին, որը ներկայացնում է ընթացիկ փաստաթղթի **DOM**-ը:

Window.location իրենից ներկայացնում է ընթացիկ էջի **URL** հասցեն: Այս մեթոդը թույլ է տալիս վերահղել օգտատիրոջը մեկ այլ **URL** հասցե:

```
window.location.href = "https://example.com";
```

Window.navigator պարունակում է բրաուզերի մասին ինֆորմացիա, օրինակ բրաուզերի անունը, վերսիան, լեզուն և այլն:

```
console.log(window.navigator.userAgent);
```

Window.screen պարունակում է էկրանի մասին ինֆորմացիա, օրինակ էկրանի բարձրությունը և լայնությունը:

```
console.log(window.screen.width, window.screen.height);
```

Window.history պարունակում է բրաուզերի պատմության հետ աշխատելու մեթոդներ: Օրինակ թույլ է տալիս տեղափոխվել առաջ կամ հետ:

```
window.history.back();
```

Մեթոդների օրինակներ`

Ցուցադրում են մոդալ պատուհաններ`

```
alert("Это предупреждение!");  
const result = confirm("Вы уверены?");  
const name = prompt("Как вас зовут?");
```

իրականացնում են բլոկում իրականացվող կոդը որոշ ժամանակ անց կամ որոշ ժամանակը մեկ`

```
setTimeout(() => console.log("Через 2 секунды"), 2000);  
setInterval(() => console.log("Каждую секунду"), 1000);
```

open(), close() բացում և փակում են բրաուզերի նոր պատուհան`

```
const newWin = window.open("https://example.com",  
"_blank");  
setTimeout(() => newWin.close(), 2000);
```

Բրաուզերի էջի չափսեր և կոորդինատներ`

innerWidth/innerHeight - բրաուզերի viewport -ի չափսերը

outerWidth/outerHeight - բրաուզերի ունջ պատուհանի չափսերը

scrollX/scrollY - պիկսելների քանակը, որոնց չափով փաստաթուղթը տեղափոխվել է հորիզոնական կամ ուղղահայաց հարթություններով:

```
console.log(`Ширина окна: ${window.innerWidth}px`);  
console.log(`Прокрутка: ${window.scrollY}px`);
```

Բրաուզերի պատուհանի իրադարձություններ`

Window օբյեկտը ունի բազմաթիվ իրադարձությունների իրականացման հնարավորություն:

Load - գործում է, երբ էջը ամբողջությամբ բեռնված է՝

```
window.addEventListener("load", () => {  
  console.log("Էջը ամբողջությամբ բեռնվել է");  
});
```

Resize - գործում է, երբ պատուհանի չափը փոփոխվել է՝

```
window.addEventListener("resize", () => {  
  console.log("Պատուհանի չափը փոփոխվել է");  
});
```

Scroll - գործում է, երբ էջը սքրոլ է արվել՝

```
window.addEventListener("scroll", () => {  
  console.log("Էջը սքրոլ է արվել");  
});
```

Բրաուզերի անվտանգությունը և սահմանափակումներ՝

Բրաուզերները օգտագործում են տվյալների միևնույն սկզբնաղբյուրի քաղաքականությունը՝ այսինքն բրաուզերի մի էջի ֆունկցիոնալով ստեղծված գործողությունը չի կարող ստանալ հասանելիություն մեկ այլ էջում սկրիպտներում հայտարարված փոփոխականներին:

JS DOM (Document Object Model)

DOM-ը ինտերֆեյս է, որը ծրագրավորողին հնարավորություն է տալիս անմիջապես փոխազդել **HTML** ֆայլերի հետ: Այս ինտերֆեյսը ներկայացնում է մեր փաստաթուղթը որպես օբյեկտների ծառ, որտեղ փաստաթղթի ցանկացած մաս (տեգեր, տեքստեր և տեգերի ատրիբուտներ) ներկայացված է որպես հանգույցներ: **DOM-ը** թույլ է տալիս ծրագրավորողներին կարդալ, փոփոխել փաստաթղթի պարունակությունը, ստրուկտուրան և դիզայնը:

```
<html>  
  <head>  
    <title>Օրինակ</title>
```

```
</head>
<body>
  <p>Ողջունյ՜</p>
</body>
</html>
```

Այս փաստաթղթում **DOM** ծառը կունենա html հանգույցը որպես հիմնական, իսկ **head, body-ն** որպես դուստր հանգույցներ, իսկ համապատասխանաբար **head, body-ն** կունենան իրենց դուստր հանգույցները համապատասխան տարրերով:

Հանգույցների տեսակները`

DOM-ի ցանկացած հանգույց ունի իր տեսակը: Ամենա հաճախ հանդիպող տեսակներ են`

Տարրեր (element nodes) - իրենից ներկայացնում են տեգեր (div, p)

Տեքստային (text nodes) - p, span ի մեջ գրված տեքստերն են

Ատրիբուտային (attribute nodes) - տեգերի ատրիբուտներն են

Մեկնաբանությունային (comment nodes) - մեկնաբանությունները

DOM -ի օգնությամբ կարող ենք դիմել ցանկացած տեգի: Դրա համար անհրաժեշտ է օգտվել document գլոբալ օբյեկտից:

Տեգին դիմելու համար պետք է այն ստանանք տեգի անունով, կլասսի անունով կամ այդիով:

` Եթե չի գտնում, վերադարձնում է null

```
console.log(document.getElementById("d1"));
```

` վերադարձնում է HTML կոլեկցիա

```
console.log(document.getElementsByClassName("d2"));
```

Tag-ով դիմելու եղանակ` վերադարձնում է HTML կոլեկցիա

```
console.log(document.getElementsByTagName("p"));
```

HTML կոլեկցիայի հետ կարող ենք աշխատել կամ **for...of** ի միջոցով կամ էլ **Array.from(elements)**

```
for (const element of elements) {  
    console.log(element);  
}  
  
const arrayElements = Array.from(elements);  
console.log(arrayElements);
```

Կարող ենք օգտվել նաև `querySelector` մեթոդով՝
Id-ով դիմելու եղանակ

```
console.log(document.querySelector("#d1"));
```

Class-ով դիմելու եղանակ

```
console.log(document.querySelector(".d2"));
```

Եթե պետք է ընտրել մեկից ավելի տեգեր, ապա՝

```
console.log(document.querySelectorAll(".d2"))
```

querySelector-ի շնորհիվ կարող ենք անմիջապես օգտագործել **foreach** մեթոդը:

Այն տեգերը, որոնք նախատեսված են օգտատիրոջից տվյալներ ստանալու համար, կարող ենք վեկավել մուտքագրված տվյալները նրա **.value** մեթոդի միջոցով:

```
function sum() {  
    // օգտագործում ենք ունար օպերատորը  
    let x = +document.querySelector("#inp1").value;  
    let y = +document.querySelector("#inp2").value;  
    let s = x + y;  
    // սվյալ օբյեկտի մեջ ձևանցվող արժեքը  
}
```

Եթե առաջանում է անհրաժեշտություն տեղի մեջ ավելացնելու նոր ինֆորմացիա, կամ փոփոխելու տեղի մեջ առկա տեքստային տվյալները, ապա պետք է օգտվենք innerHTML կամ innerTEXT մեթոդներից:

```
let s = 9
document.querySelector('#res').innerHTML = '<i>Sum = </i>' + s; // Sum = 9
document.querySelector('#res').innerText = '<i>Sum = </i>' + s; //<i>Sum = </i>9
```