

A yellow square containing the letters 'JS' in a bold, black, sans-serif font.

JS

# SCSA JavaScript Courses

---

1) DOM (EVENTS)

# Event Objects

---

The target property of the Event interface is a reference to the object onto which the event was dispatched. It is different from Event.currentTarget when the event handler is called during the bubbling or capturing phase of the event.

```
let btn = document.querySelector('#btn');  
  
btn.addEventListener('click', function(event) {  
  console.log(event.type);  
});
```

roperty / Method	Description
bubbles	true if the event bubbles
cancelable	true if the default behavior of the event can be canceled
currentTarget	the current element on which the event is firing
defaultPrevented	return true if the preventDefault() has been called.
detail	more informatio nabout the event
eventPhase	1 for capturing phase, 2 for target, 3 for bubbling
preventDefault()	cancel the default behavior for the event. This method is only effective if the cancelable property is true
stopPropagation()	cancel any further event capturing or bubbling. This method only can be used if the bubbles property is true.
target	the target element of the event
type	the type of event that was fired

# Dom Traversing

---

```
let navigation=document.getElementById('navigation');  
console.log(navigation.parentElement)  
console.log(navigation.children)  
console.log(navigation.firstChild)  
console.log(navigation.lastElementChild)  
console.log(navigation.firstChild.nextElementSibling )  
console.log(navigation.firstChild.previousElementSibling )
```

# preventDefault()

---

To prevent the default behavior of an event, you use the `preventDefault()` method. For example, when you click a link, the browser navigates you to the URL specified in the `href` attribute:

```
let link = document.querySelector('a');
link.addEventListener('click',function(event) {

console.log('clicked'); event.preventDefault();

});
```

# Target

---

The target property of the Event interface is a reference to the object onto which the event was dispatched. It is different from Event.currentTarget when the event handler is called during the bubbling or capturing phase of the event.

```
let link = document.querySelector('a');
link.addEventListener('click',function(event) {

console.log('clicked'); event.target();

});

let link = document.querySelector('a');
link.addEventListener('click',function(event) {

console.log('clicked'); event.currentTarget();

});
```

# Append Vs Prepend

---

Using the append() method to append an element;

```
<li>
  <a href="#" class="delete">
    First
    //prepend
  </a>

</li>
<li>
  <a href="#" class="delete">Second</a>

</li>
```

Using the append() method to append text to an element

```
<li>
  <a href="#" class="delete">
    //append
    First
  </a>

</li>
<li>
  <a href="#" class="delete">Second</a>

</li>
```

Using the append() method to append an element;

```
const li=document.querySelectorAll('li');
li.forEach(function(btn){
    btn.addEventListener('click', function(e){
        const change=e.target;
        let p= document.createElement('p');
        change.append(p)

    })
})
```

```
<li>
◦ <a href="#" class="delete">
    First
◦ </a>
```



# HomeWork

---

- 1) უკვე არსებული სლაიდერი დავალაგოთ ფუნქციებით..
- 2) გვაქვს ორი ღილაკი. და გვაქვს ორი ბლოკი „აღწერა 1“ „აღწერა 2“; ერთ ღილაკზე დაჭერის შედეგად გამოვიდეს პირველი აღწერა. ხოლო მეორე ღილაკზე დაჭერის შედეგად გამოვიდეს მეორე აღწერა;
- 3) გვაქვს ლისტები. თითოეულ ლისტზე დაჭერის შედეგად მოხდეს ამ ლისტზე ხაზის გადასმა. იყოს წითელი ხაზი და გვერდით დაეწეროს წაშლა