
<Semester II>

<//Programming for Interactive Interfaces//>

Week 10

Event Listener

The `addEventListener()` method is used to attach an event handler to a HTML document. Unlike the traditional way of assigning events directly (e.g., `element.onclick = function() {}`), `addEventListener()` allows multiple handlers for the same event. It provides flexibility because one can easily remove event listeners or add them dynamically based on conditions. Unlike the traditional way of assigning events directly (e.g., `element.onclick = function() {}`), `addEventListener()` allows multiple handlers for the same event.

Syntax: `element.addEventListener(event, listener, useCapture);`

- ❑ event: event can be any valid JavaScript event. Events are used without “on” prefixes like using “click” instead of “onclick” or “mousedown” instead of “onmousedown”.

Event Listener

Syntax: `element.addEventListener(event, listener, useCapture);`

- ❑ **listener(handler function):** It can be a JavaScript function that responds to the event occurring. It can be a regular function (known as callback function, or arrow function, or anonymous function).
- ❑ **useCapture:** It is an optional parameter used to control event propagation. A boolean value is passed where “*true*” denotes the capturing phase and “*false*” denotes the bubbling phase.

event can be a parameter passed to the listener (also known as handler function). Event is an object provided by the browser automatically. It has a lot of information about the interaction that has taken place.

event object



Syntax of Event Listener

element , In which event attached

It tells event phase

```
<element>.addEventListener(<eventName>, callback, {capture:boolean})
```

It can be 'click', 'key up', 'key down' etc. events.

This function fires after the event happened.

Properties and Methods of classList

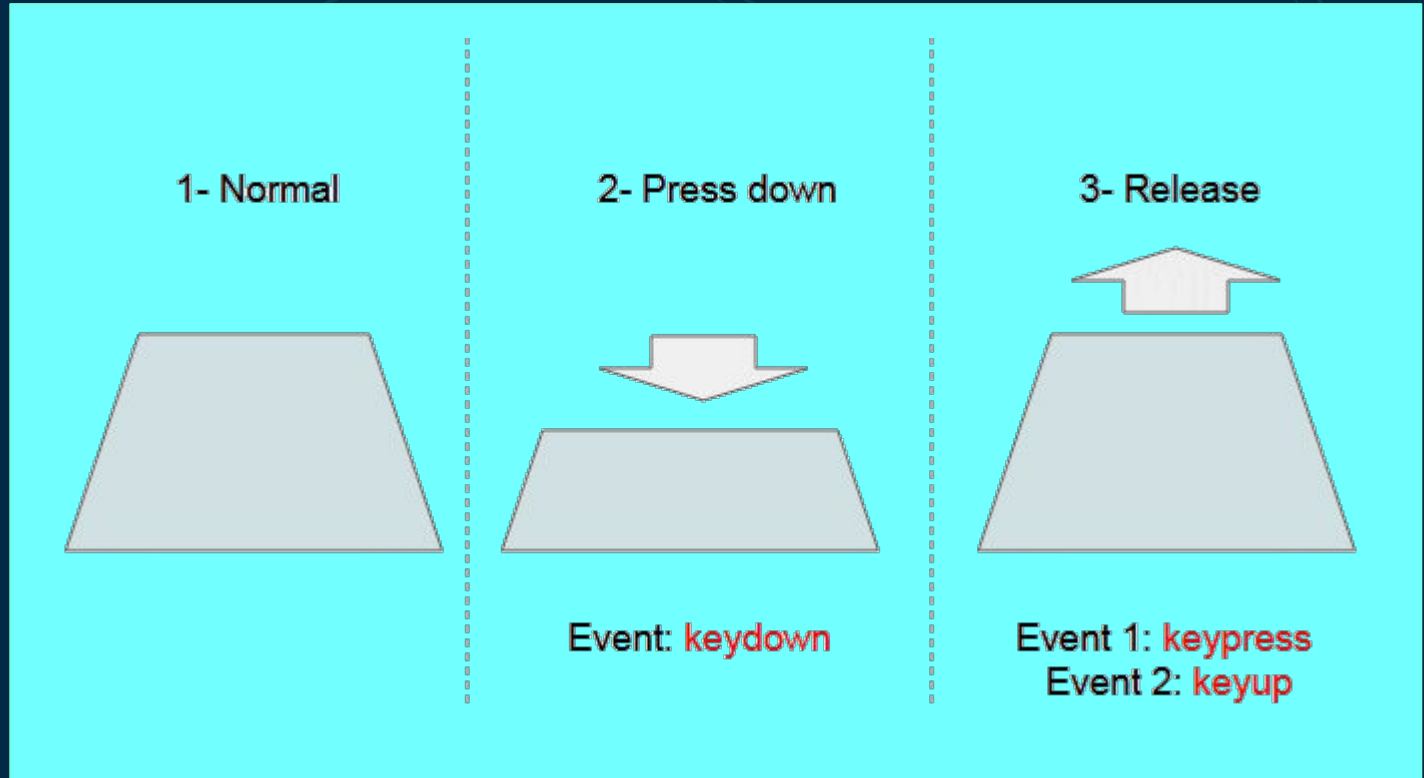
Property/Method	Description
<u>add()</u>	Adds one or more tokens to the list
<u>contains()</u>	Returns true if the list contains a class
<u>entries()</u>	Returns an Iterator with key/value pairs from the list
<u>forEach()</u>	Executes a callback function for each token in the list
<u>item()</u>	Returns the token at a specified index
<u>keys()</u>	Returns an Iterator with the keys in the list

Properties and Methods of classList

Property/Method	Description
<u>length</u>	Returns the number of tokens in the list
<u>remove()</u>	Removes one or more tokens from the list
<u>replace()</u>	Replaces a token in the list
<u>supports()</u>	Returns true if a token is one of an attribute's supported tokens
<u>toggle()</u>	Toggles between tokens in the list
<u>value</u>	Returns the token list as a string
<u>values()</u>	Returns an Iterator with the values in the list

Key-based Events in JavaScript

Note: The keypress event will not be emitted for the keys such as ALT, SHIFT, CTRL, META-KEY, ESC. If you want to detect whether those keys are being pressed, you can use the properties such as altKey, shiftKey, ctrlKey, metaKey of the event object.



Key-Based Events





Homework: Finish the HTML + CSS portion of
your LCA2 project.

[W3Schools JavaScript Tutorial](#)
[JavaScript for Web | MDN](#)

Reference



1. [HTML DOM Element classList Property and Methods \(add, remove, toggle\) - W3Schools](#)
 2. [JavaScript addEventListener\(\) | GeeksforGeeks](#)
 3. [Exploring Lesser-Known JavaScript Events for Enhanced Interactivity | Smooets](#)
 4. [“e” \(event object\) | by Wendy You | Medium](#)
 5. [Javascript KeyboardEvent Tutorial with Examples | o7planning.org](#)
 6. [Keyboard Events and Properties - W3Schools](#)
-

Fonts & colors used

This presentation has been made using the following fonts:

Blinker

(<https://fonts.google.com/specimen/Blinker>)

Inconsolata

(<https://fonts.google.com/specimen/Inconsolata>)

#ffffff

#022a46

#72ffdd

#72fff

#0d3a58