Змістовий модуль 4. Основи веб-програмування мовою JavaScript (+jQuery)

Тема 12: Кнопки, події, функції

На веб-сторінці можуть відбуватись різноманітні **події**: сам факт завантаження сторінки у браузері, клацання кнопок чи інших елементів сторінки, наведення вказівника мишки чи натиснення клавіш на клавіатурі - все це приклади подій. Вказавши **обробник для події** у вигляді **функції** мовою JavaScript, можна забезпечити гнучку інтерактивність та динамічне оновлення вмісту сторінки залежно від дій користувача.

Для окремих елементів сторінки додаються вказівки обробки стандартних подій, наприклад клацання кнопки. Обробником такої події може бути функція, описана у скрипті на цій же сторінці чи у прикріпленому файлі. Обробник події може звертатись до об'єктної моделі сторінки, модифікуючи її вміст та оформлення.

Наприклад:

Кнопка в документі html:

<button onclick="**fishka**()"> Фішка </button>

Код функції JS:

function **fishka**() {

document.getElementById("**demo**").innerHTML = "Фішка";

}

При клацанні на кнопку Фішка в елементі **demo** вміст змінюється на **Фішка**.

Розглянемо простий приклад.

На сторінці подано питання і 2 кнопки, котрі називаються **Підказка** та **Відповідь**. При клацанні кнопки **Підказка** має бути показаний текст підказки на поставлене питання; при повторному клацанні підказка знову приховується. При наведенні вказівника мишки на кнопку **Відповідь** користувачеві запропонують ввести пароль. Якщо введений пароль правильний, то буде показана відповідь. Подібну взаємодію з користувачем допоможе забезпечити програмний код мовою JavaScript.

Завдання та приклади краще виконувати в сервісі **https://jsbin.com/**

Кнопка з підказкою

Спочатку розберемо кнопку з підказкою. Об'єкт кнопка створюється тегом **<button> </button>**, всередині якого вписуємо текст напису кнопки. **Атрибутом** тегу може бути **функція**, котра буде виконана в результаті настання деякої **події**, наприклад, клацанні по цій кнопці.

<button **onclick='show()'**>Підказка</button>

Наприклад, вкажемо, що після клацання по кнопці має бути виконана функція **show()**, яку й опишемо у розділі **<script></script>** заголовка нашої сторінки.

Розберемо дію цієї функції: якщо підказку приховано, то її потрібно показати, інакше її потрібно приховати.

Підказка – це блок вмісту, позначений, наприклад ідентифікатором **hint**.

** Текст підказки **

Прихованому стану блоку відповідає атрибут **display**, що дорівнює **none**. Якщо блок показано, атрибут **display** дорівнює порожнім лапкам "". Виходячи із цього, запишемо код мовою JavaScript:

```
function show() {
		if (document.getElementById("hint").style.display == "none") {
			document.getElementById("hint").style.display = "";
		} else {
			document.getElementById("hint").style.display = "none";
		}
	}
```

Приклад: https://jsbin.com/lopevubasa/1/edit?html,js,output

Тепер розберемось із самим текстом підказки. Нам не потрібно його одразу показувати на сторінці – тоді кожен бажаючий зможе побачити підказки і правильні відповіді, відкривши початковий код сторінки. Сформуємо підказку за допомогою функції **skarb()**, яка буде виконуватись після завантаження основного вмісту веб-сторінки, тобто до тегу body додаємо обробку події **onload** :

<body onload="skarb()">

Ця функція створить текстовий напис підказки та додасть його у документ у вигляді заголовка другого рівня, приховавши його в момент додавання.

```
function skarb() {
		var hint_element=document.createElement("div");
		document.body.appendChild(hint_element);
		hint_element.innerHTML = "Це тільки підказка. Правильна відповідь за паролем!";
		hint_element.setAttribute("id", "hint");
		hint_element.style.display = "none";
	}
```

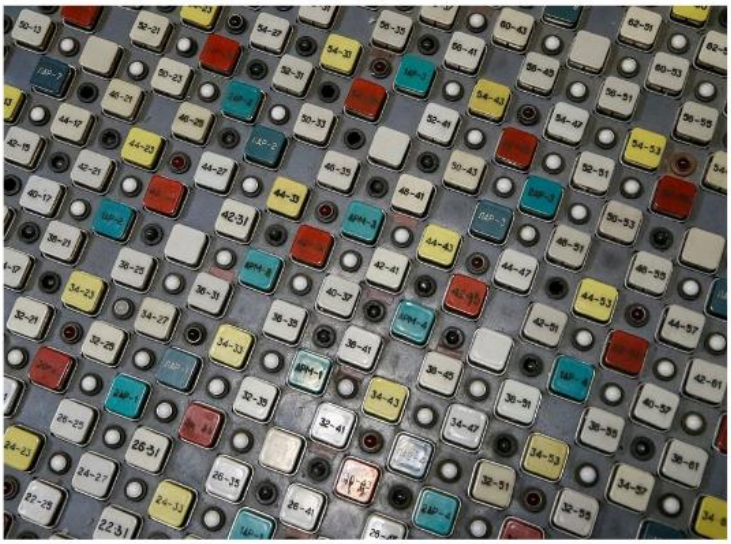
Команди цієї функції можна пояснити так:

Створюємо елемент вмісту – div
Додаємо створений елемент до основного тексту веб-сторінки
Текст створеного елементу рівний напису в лапках
Надаємо створеному елементу ідентифікатор hint (саме до цього ідентифікатора застосовується функція show())
Вказуємо властивість display = "none" для приховання створеного елемента на сторінці.
Зберігаючи сторінку, випробуємо дію створеного коду.

Приклад: https://jsbin.com/yatifirada/1/edit?html,js,output

Багатофункціональна кнопка

У сучасних інтерфейсах одна й та ж кнопка може відповідати за різні функції. Така можливість з'явилась після винаходу в ЦЕРНі панелі, на якій кнопка змінювала функцію, залежно від поточної ситуації.



Мовою JavaScript це реалізується так: змінна-покажчик має початкове значення 0. Коли кнопку клацнули, то значення цієї змінної збільшуємо на 1. Якщо значення парне, то показуємо кнопку 1 (і приховуємо кнопку 2), якщо значення непарне, то навпаки - показуємо кнопку 2 і приховуємо кнопку 1. Поміркуйте, як модифікувати наведений код для такого функціоналу:

```
var k=0;
 function show(){
 document.getElementById("demo").innerHTML="hello";
 if (k%2==0) {document.getElementById("demo").style.display = "none";}
 else {document.getElementById("demo").style.display = "";}
 k=k+1;
 }
```

Поміркуйте, чи можна використати тут одну й ту ж кнопку, а не оперувати двома різними?

Кнопка з паролем

Додамо функцію обробки події наведення вказівника миші на кнопку **Відповідь**. Нагадаємо, при цьому має з'являтись запит паролю, і правильна відповідь показуватиметься лише якщо введено правильний пароль.

Якщо елемент був прихований, то потрібно виконати запит паролю. Якщо введений пароль правильний, то потрібно показати прихований текст, інакше повідомити, що пароль помилковий.

Якщо пароль правильний, то підказку показуємо. Якщо неправильний, то пишемо, що Пароль неправильний.

```
< p >Текст запитання: Скільки буде 3+3?< /p >
 < button onclick="fishka()" > Підглянути відповідь < / button >
 < p id="vidp" style="display:none" >Це буде 6< /p >

    function fishka(){
    var enteredpass = prompt("Вкажіть пароль");
    if (enteredpass == "123456")
        { document.getElementById("vidp").style.display = ""; }
    else { alert("Пароль неправильний");}
    }
```

Приклад: https://jsfiddle.net/oksanapas/xnjq9ayk/3/

Додатково: якщо текст не був прихований, то його слід приховати.

```
function show_with_pass() {
        if (document.getElementById("vidp").style.display == "none") {
                var enteredpass = prompt("Вкажіть пароль");
                if (enteredpass == "123456") {
                        document.getElementById("vidp").style.display = "";
                } else {
                        alert("пароль неправильний");
                }
        } else {
                document.getElementById("vidp").style.display = "none";
        }
}
```

Приклад: https://jsbin.com/yuniganefo/1/edit?html,js,output

Ускладнення паролю

Правильним паролем може бути не просто певна послідовність символів (у наведеному прикладі це **123456**), але й, наприклад, поточна дата (число). Додамо на початок функції створення об'єкту типу Дата:

var d= new Date();

А введений користувачем текст будемо порівнювати не із заданим паролем, а поточною датою:

if (enteredpass == d.getDate())

Цікаве: спробуйте дослідити роботу методів **getMonth()** та **getYear()**.

Код JavaScript можна розмістити у зовнішньому файлі, підключаючи його до поточного документа. Таким чином, у самому веб-документі залишиться лише таке наповнення:

<h1>Питання</h1>

<button onclick="show()">Підказка</button>

<button onmouseover="show_with_pass()">Відповідь</button>

яке не містить ані текстів підказки чи відповіді, а також не містить паролю, який відкриває відповідь. Більше того, паролю у явному вигляді немає і у тексті скрипту **script.js**

ВПРАВИ ТА ПРАКТИЧНІ ЗАВДАННЯ ДО ВИКОНАННЯ

1. Вправа «Пригадай число»:

Показується загадане число (велике). Користувач має ввести це число. Для ускладнення виводяться відволікаючі повідомлення.

Приклад реалізації: https://www.csfieldguide.org.nz/en/interactives/number-memory/

Підказка. Пауза між повідомленнями не обов'язкова, якщо скористатись виведенням в alert. Але якщо хочеться поекспериментувати з паузами з виведенням повідомлення в innerHTML можна застосувати функцію:

```
setTimeout(myFunction, 5000);
function myFunction() {
document.getElementById("demo").innerHTML="1"
```

```
setTimeout(myFunction2, 2000);
}
function myFunction2() {
document.getElementById("demo").innerHTML="2"
}
```

Відомості та рекомендації щодо виконання даного завдання – див. Лекцію 12.

2. Практичне завдання «Підказка-Відповідь»:

Створіть інтерактивний документ з кнопками **Підказка** та **Відповідь**. При клацанні кнопки **Підказка** має бути показаний текст підказки на поставлене питання; при повторному клацанні підказка знову приховується. При наведенні вказівника мишки на кнопку **Відповідь** користувачеві запропонують ввести пароль. Якщо введений пароль правильний, то буде показана відповідь.

Текст html-документа:

Питання: 2+2=?

<button onclick="fishka1()" id="p1"> Підказка1 </button>

<button onclick="fishka2()" style="display:none" id="p2"> Підказка2 </button>

<button onclick="fishka3()" style="display:none" id="p3"> Відповідь</button>

** **

В функції потрібно змінювати вміст елемента demo, а також змінювати видимість елементів p1 та p2.

Завдання можна виконувати в сервісі https://jsbin.com/?html,js,output . Результат роботи можна зберегти **Share** та здати посилання на виконану роботу.

Відомості та рекомендації щодо виконання даного завдання – див. Лекцію 12.