**МІНЕСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Національний технічний університет**

**«Харківський політехнічний інститут»**

Навчально науковий інститут комп’ютерних наук та інформаційних технологій

Кафедра Інформатики та інтелектуальної власності

**ЗВІТ**

**до лабораторної роботи 7 з дисципліни**

**«Основи WEB-технологій»**

Студент Яковлев І. А.

Викладач Шуба І. В.

м. Харків

Лабораторна робота 7

Завдання 1

Використовуючи методи getElementById, getElementsByClassName,

getElementsByName, getQuerySelector getElementsByTagName, стандартного об'єкту DOM моделi document та форму з останнього завдання третьої лабораторної роботи чи її прикрашений варіант з завдань інших робіт

згідно до свого варіанту, встановити на кнопку Відіслати скрипт, який би

робив в окремому блоці розгорнутий звіт за інформацією, яка була занесена

до форми, тобто навести зміст інформації з полів та інших компонентів

форми.

Опис директив:

getElementById - Повертає посилання елемент по його ідентифікатору (ID); ідентифікатор є рядком, який може бути використаний для ідентифікації елемента; вона може бути визначена за допомогою атрибута id HTML або зі скрипта.

"use strict";

let checkBoxes = ["comp", "sport", "art", "sci"];

let ids = ["name", "age", "sex", "inter", "opin"];

function display(){

document.getElementById("name").append(document.getElementById("inName").value);

document.getElementById("age").append(document.getElementById("inAge").options[document.getElementById("inAge").selectedIndex].text);

if(document.getElementById("m").checked){

document.getElementById("sex").append("male")

}

else if(document.getElementById("f").checked){

document.getElementById("sex").append("female")

}

else{

document.getElementById("sex").append("unknown")

}

for(let i=0;i<checkBoxes.length;i++){

if(document.getElementById(checkBoxes[i]).checked){

document.getElementById("inter").append(checkBoxes[i] + ", ")

}

}

document.getElementById("opin").append(document.getElementById("inOpin").value);

for(let i = 0; i < ids.length; i++){

document.getElementById(ids[i]).style.display = "block";

}

}

Завдання 2

Використовуючи список, що розкривається згiдно до свого варіанту з

четвертого завдання третьої лабораторної роботи та подію change створити

скрипт, який , аналізуючи властивості selectedIndex, чи text, чи value масиву

options демонстрував би фотографію та короткий опис відповідного міста у

відповідному блоці, змінюючи значення CSS властивостей display (парні

номери варіантів) чи z-index (непарні номери варіантів)

"use strict";

let links = ["1.jpg", "2.jpg", "3.jpg", "4.jpg", "5.jpg", "6.jpg", "7.jpg"]

function change(){

document.getElementById("img").src = links[document.getElementById("sel").selectedIndex];

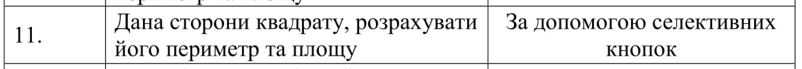
}

Завдання 3

Створити форму та розробити JS-скрипт з вирішення геометричної задачі

згідно свого варіанту. У випадку введення до полів форми некоректних даних

вивести відповідне повідомлення



"use strict";

function check(){

if(document.getElementById("sel1").selectedIndex == 3 & document.getElementById("sel2").selectedIndex == 2){

alert("Верно");

}

else{

alert("Ложь");

}

}

Завдання 4

У верхній частині сторінки створити форму, що містить дві кнопки з

написами Додати та Видалити. Для першої кнопки розробити скрипт, що

додає (методи createElement та appendChild стандартного об'єкту

document) на сторінку новий блок з випадковими (функція Мath.random())

координатами, розмірами, кольором фону та додатковими параметрами

згідно до свого варіанту, на другу кнопку - скрипт, що видаляє (метод

removeChild) доданий блок згiдно свого варіанту.



"use strict";

let newDiv = document.createElement("div");

function add(){

newDiv.style.position = "relative";

newDiv.style.background = String("#" + Math.floor(Math.random()\*16777215).toString(16));

newDiv.style.height = String(Math.floor(Math.random() \* 90) + 1 + "vh");

newDiv.style.width = String(Math.floor(Math.random() \* 50) + 1 + "vw");

newDiv.style.left = String(Math.floor(Math.random() \* 50) + 1 + "vw");

newDiv.style.border = String(Math.floor(Math.random() \* 50) + 1 + "px solid #" + Math.floor(Math.random()\*16777215).toString(16));

document.getElementById("top").appendChild(newDiv);

}

function del(){

document.getElementById("top").removeChild(newDiv);

}

Завдання 5

Створити форму, що містить порожній маркований список з предметної

області згідно свого варіанту, дві кнопки з написами Додати та Видалити та

текстове поле. Для першої кнопки розробити скрипт, що додає (методи

createElement Ta appendchild стандартного об'єкту document) у кінець

списку вміст текстового поля, якщо воно не порожне, на другу кнопку

скрипт, що видаляє (метод removeChild) елемент списку згідно свого

варіанту.



"use strict";

function add(){

let newLine = document.createElement("li");

if (document.querySelector('input').value != "") {

newLine.innerHTML = document.querySelector('input').value;

document.getElementById("list").appendChild(newLine);

}

}

function del(){

document.querySelector('ul > li').remove();

}

Завдання 6

Реалізувати минуле завдання, замінивши маркований список таблицею, яка

мiстила б не менше ніж п'ять стовпчиків (полів) тієї ж предметної області

відповідно одне текстове поле введення стількома полями скільки стовпчиків

має створена таблиця. Розроблені скрипти для відповідних кнопок мають виконувати ті ж самі функції для рядків (записів) таблиці згідно з відповідними варіантами.

"use strict";

function add(){

let newLine = document.createElement("tr");

document.querySelector("table").appendChild(newLine);

let arr = Array.from(document.querySelectorAll('[type="text"]'));

for(let i = 0; i < arr.length; i++){

let newTd = document.createElement("td");

newTd.innerHTML = arr[i].value;

newLine.appendChild(newTd);

}

}

function del(){

document.querySelector('table > tr').remove();

}

Завдання 7

Модифiкувати п'яте завдання цієї роботи таким чином, щоб елементи додані

до списку мали б подію (метод addEventListener) змінювати колір фону та

тексту на інші значення (наприклад інверсні) під час одинарного клацання

лівою кнопкою миші, тобто виділялись би, а кнопка Видалити б видаляла б

видiленi елементи.

Модифікувати шосте завдання цієї роботи таким чином, щоб рядки додані до

таблиці мали б подію (метод addEventListener) змінювати колір фону та

тексту на iншi значення (наприклад інверсні) під час одинарного клацання

лівою кнопкою миші, тобто видiлялись би, а кнопка Видалити б видаляла б

виділені рядки.

"use strict";

function changeBg(e){

if (window.getComputedStyle(e.target).backgroundColor == "rgb(255, 255, 255)") {

e.target.style.backgroundColor = "rgb(125, 125, 230)";

e.target.style.color = "red";

}

else{

e.target.style.backgroundColor = "rgb(255, 255, 255)";

e.target.style.color = "black";

}

}

document.querySelector("ul").addEventListener("click", function(e){

changeBg(e);

})

function add(){

let newLine = document.createElement("li");

if (document.querySelector('input').value != "") {

newLine.innerHTML = document.querySelector('input').value;

document.getElementById("list").appendChild(newLine);

}

}

function del(){

let arr = Array.from(document.querySelectorAll("li"));

for(let i = 0; i < arr.length; i++){

if(arr[i].style.color == "red"){

arr[i].remove();

}

}

}

"use strict";

let table = document.querySelector("table");

function changeBg(i){

table.rows[i].style.backgroundColor = "rgb(125, 125, 230)";

table.addEventListener("click", function(e){

let cell = e.target;

let i = cell.parentNode.rowIndex;

let row = cell.parentNode;

if (window.getComputedStyle(row).backgroundColor == "rgb(125, 125, 230)") {

row.style.backgroundColor = "rgb(255, 255, 255)";

}

else{

row.style.backgroundColor = "rgb(125, 125, 230)";

}

})

}

document.querySelector("table").addEventListener("click", function(e){

let cell = e.target;

let i = cell.parentNode.rowIndex;

changeBg(i);

})

function add(){

let newLine = document.createElement("tr");

document.querySelector("table").appendChild(newLine);

let arr = Array.from(document.querySelectorAll('[type="text"]'));

for(let i = 0; i < arr.length; i++){

let newTd = document.createElement("td");

newTd.innerHTML = arr[i].value;

newLine.appendChild(newTd);

}

}

function del(){

let arr = [];

for (let i = 0; i < table.rows.length; i++) {

if (window.getComputedStyle(table.childNodes[i+1]).backgroundColor == "rgb(125, 125, 230)") {

table.childNodes[i+1].remove();

}

}

}

Завдання 6

Реалізувати минуле завдання, замінивши маркований список таблицею, яка

мiстила б не менше ніж п'ять стовпчиків (полів) тієї ж предметної області

відповідно одне текстове поле введення стількома полями скільки стовпчиків

має створена таблиця. Розроблені скрипти для відповідних кнопок мають виконувати ті ж самі функції для рядків (записів) таблиці згідно з відповідними варіантами.

"use strict";

function add(){

let newLine = document.createElement("tr");

document.querySelector("table").appendChild(newLine);

let arr = Array.from(document.querySelectorAll('[type="text"]'));

for(let i = 0; i < arr.length; i++){

let newTd = document.createElement("td");

newTd.innerHTML = arr[i].value;

newLine.appendChild(newTd);

}

}

function del(){

document.querySelector('table > tr').remove();

}

Завдання 8

Модифікувати скрипти минулого завдання так, щоб кнопка Додати не була

доступною (атрибут disabled) доти, доки усi текстові поля (подія keyup) не

будуть заповнені інформацією для додання в таблицю чи список, а кнопка

Видалити була б недоступною доти, доки хоча б один елемент чи рядок не

був би виділеним.

"use strict";

let list = document.getElementById("list");

let addBut = document.getElementById("b1");

let delBut = document.getElementById("b2");

let inp = document.getElementById("inp")

function changeBg(e){

if (window.getComputedStyle(e.target).backgroundColor == "rgb(255, 255, 255)") {

e.target.style.backgroundColor = "rgb(125, 125, 230)";

e.target.style.color = "red";

}

else{

e.target.style.backgroundColor = "rgb(255, 255, 255)";

e.target.style.color = "black";

}

let items = list.getElementsByTagName("li");

for(let i = 0; i < items.length; ++i){

delBut.disabled = true;

if (window.getComputedStyle(items[i]).color == "rgb(255, 0, 0)") {

delBut.disabled = false;

return;

}

}

}

document.querySelector("ul").addEventListener("click", function(e){

changeBg(e);

})

document.querySelector("input").addEventListener("input", updateVal);

function updateVal(e){

addBut.disabled = false;

if(e.target.value == ""){

addBut.disabled = true;

}

}

function add(){

let newLine = document.createElement("li");

if (document.querySelector('input').value != "") {

newLine.innerHTML = document.querySelector('input').value;

document.getElementById("list").appendChild(newLine);

}

inp.value = "";

addBut.disabled = true;

}

function del(){

let arr = Array.from(document.querySelectorAll("li"));

for(let i = 0; i < arr.length; i++){

if(arr[i].style.color == "red"){

arr[i].remove();

}

}

delBut.disabled = true;

}

"use strict";

let table = document.getElementById("tab");

let addBut = document.getElementById("b1");

let delBut = document.getElementById("b2");

let arr = Array.from(document.querySelectorAll('[type="text"]'));

function changeBg(i){

table.rows[i].style.backgroundColor = "rgb(125, 125, 230)";

table.rows[i].style.color = "red";

table.addEventListener("click", function(e){

let cell = e.target;

let i = cell.parentNode.rowIndex;

let row = cell.parentNode;

if (window.getComputedStyle(row).backgroundColor == "rgb(125, 125, 230)") {

row.style.backgroundColor = "rgb(255, 255, 255)";

row.style.color = "black";

}

else{

row.style.backgroundColor = "rgb(125, 125, 230)";

row.style.color = "red";

}

})

for(let i = 0, row; row = table.rows[i]; ++i){

delBut.disabled = true;

if (window.getComputedStyle(row).color == "rgb(255, 0, 0)") {

delBut.disabled = false;

return;

}

}

}

table.addEventListener("click", function(e){

let cell = e.target;

let i = cell.parentNode.rowIndex;

changeBg(i);

})

document.querySelectorAll(".inpq").forEach(item => {

item.addEventListener("input", event => {

for (let i = 0; i < arr.length; i++) {

if (arr[i].value == "") {

addBut.disabled = true;

return;

}

addBut.disabled = false;

}

})

})

function add(){

let newLine = document.createElement("tr");

document.querySelector("table").appendChild(newLine);

for(let i = 0; i < arr.length; i++){

let newTd = document.createElement("td");

newTd.innerHTML = arr[i].value;

newLine.appendChild(newTd);

}

}

function del(){

let arr = [];

for (let i = 0; i < table.rows.length; i++) {

if (window.getComputedStyle(table.childNodes[i+1]).backgroundColor == "rgb(125, 125, 230)") {

table.childNodes[i+1].remove();

}

}

delBut.disabled = true;

}

Завдання 9

Модифікувати скрипти минулого завдання так, щоб подвійне клацання на

елемент списку чи рядок таблиці виводив би вiдповiдну інформацію до

текстових полів, а нова кнопка Зберігти дозволяла б зберігати редаговану

інформацію у відповідному місці списку чи таблиці. Під час редагування

кнопки Додати та Видалити мають бути недоступними.

"use strict";

let list = document.getElementById("list");

let addBut = document.getElementById("b1");

let delBut = document.getElementById("b2");

let inp = document.getElementById("inp")

let idx;

function changeBg(e){

if (window.getComputedStyle(e.target).backgroundColor == "rgb(255, 255, 255)") {

e.target.style.backgroundColor = "rgb(125, 125, 230)";

e.target.style.color = "red";

}

else{

e.target.style.backgroundColor = "rgb(255, 255, 255)";

e.target.style.color = "black";

}

let items = list.getElementsByTagName("li");

for(let i = 0; i < items.length; ++i){

delBut.disabled = true;

if (window.getComputedStyle(items[i]).color == "rgb(255, 0, 0)") {

delBut.disabled = false;

return;

}

}

}

document.querySelector("ul").addEventListener("click", function(e){

changeBg(e);

})

document.querySelector("ul").addEventListener("dblclick", function(e){

inp.value = e.target.innerHTML;

idx = Array.prototype.indexOf.call(list.childNodes, e.target);

//alert(idx)

})

document.querySelector("input").addEventListener("input", updateVal);

function updateVal(e){

addBut.disabled = false;

if(e.target.value == ""){

addBut.disabled = true;

}

}

function add(){

let newLine = document.createElement("li");

if (document.querySelector('input').value != "") {

newLine.innerHTML = document.querySelector('input').value;

document.getElementById("list").appendChild(newLine);

}

inp.value = "";

addBut.disabled = true;

}

function save(){

list.childNodes[idx].innerHTML = inp.value;

inp.value = "";

}

function del(){

let arr = Array.from(document.querySelectorAll("li"));

for(let i = 0; i < arr.length; i++){

if(arr[i].style.color == "red"){

arr[i].remove();

}

}

delBut.disabled = true;

}

"use strict";

let table = document.getElementById("tab");

let addBut = document.getElementById("b1");

let delBut = document.getElementById("b2");

let saveBut = document.getElementById("b3");

let arr = Array.from(document.querySelectorAll('[type="text"]'));

let r;

function changeBg(i){

table.rows[i].style.backgroundColor = "rgb(125, 125, 230)";

table.rows[i].style.color = "red";

table.addEventListener("click", function(e){

let cell = e.target;

let i = cell.parentNode.rowIndex;

let row = cell.parentNode;

if (window.getComputedStyle(row).backgroundColor == "rgb(125, 125, 230)") {

row.style.backgroundColor = "rgb(255, 255, 255)";

row.style.color = "black";

}

else{

row.style.backgroundColor = "rgb(125, 125, 230)";

row.style.color = "red";

}

})

for(let i = 0, row; row = table.rows[i]; ++i){

delBut.disabled = true;

if (window.getComputedStyle(row).color == "rgb(255, 0, 0)") {

delBut.disabled = false;

return;

}

}

}

table.addEventListener("click", function(e){

let cell = e.target;

let i = cell.parentNode.rowIndex;

changeBg(i);

})

table.addEventListener("dblclick", function(e){

let cell = e.target;

r = cell.parentNode.rowIndex;

for (let i = 0; i < arr.length; i++) {

arr[i].value = table.rows[r].cells[i].innerHTML;

}

saveBut.disabled = false;

})

document.querySelectorAll(".inpq").forEach(item => {

item.addEventListener("input", event => {

for (let i = 0; i < arr.length; i++) {

if (arr[i].value == "") {

addBut.disabled = true;

return;

}

addBut.disabled = false;

}

})

})

function add(){

let newLine = document.createElement("tr");

document.querySelector("table").appendChild(newLine);

for(let i = 0; i < arr.length; i++){

let newTd = document.createElement("td");

newTd.innerHTML = arr[i].value;

newLine.appendChild(newTd);

}

}

function save(){

for (let i = 0; i < arr.length; i++) {

table.rows[r].cells[i].innerHTML = arr[i].value;

}

saveBut.disabled = true;

}

function del(){

let arr = [];

for (let i = 0; i < table.rows.length; i++) {

if (window.getComputedStyle(table.childNodes[i+1]).backgroundColor == "rgb(125, 125, 230)") {

table.childNodes[i+1].remove();

}

}

delBut.disabled = true;

}