**Առարկա՝ JavaScript**

*Խնդիր 1*

HTML տիրույթում ստեղծել շախմատի տախտակ, որի վրա նշված են հորիզոնական և ուղղահայաց կոորդինատները։ Շախմատի տախտակից ներքև ստեղծել մեկ կոճակ և մեկ տեքստային դաշտ։ JavaScript-ի միջոցով կազմել ծրագիր, որի կատարման ժամանակ շախմատի տախտակի վրա որևէ դաշտ ընտրելիս այդ դաշտի ուղղահայաց, հորիզոնական և անկյունագծային շարքերը կներկվեն որևէ գույնով, տեքստային դաշտում կարտածվի ընտրված դաշտի կոորդինատները, իսկ կոճակին սեղմելիս ծրագիրը կբերվի սկզբնական վիճակի։

<!—index.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>SOAP Request with Fetch</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<h1>Chessboard</h1>

<div class="frame">

<div class="letters"></div>

<div board class="board"></div>

<div class="numbers"></div>

</div>

<p class="output">

<button onclick="restart()">

Reset

</button>

<br><br>

Field coordinates:

<span output></span>

</p>

<script src="app.js"></script>

</body>

</html>

/\* style.css \*/

body{

margin: 0;

padding: 0;

font-family: 'Courier New', Courier, monospace;

}

h1{

text-align: center;

}

.frame{

width: 450px;

height: 450px;

margin: 0 auto;

position: relative;

}

.board{

width: 400px;

height: 400px;

display: flex;

flex-wrap: wrap;

border: 1px solid #000;

padding: 25px;

}

.field{

width: 50px;

height: 50px;

transition: 0.1s;

box-shadow: inset 0px 0px 1.5px 0px #000;

}

.white{background: #fff;}

.black{background: #000;}

.row-col{background: lightskyblue;}

.diagonal{background: lightseagreen;}

.captured{background: rgb(139, 139, 197);}

.field:hover{

filter: invert(0.30);

}

.output{

width: 400px;

margin: 20px auto;

}

.letters{

width: 400px;

position: absolute;

bottom:0px;

display: flex;

justify-content: space-around;

padding: 0 25px ;

}

.letter{

/\* background: red; \*/

width: 50px;

text-align: center;

}

.numbers{

height: 400px;

width: 20px;

position: absolute;

top:25px;

left:8px;

display: flex;

flex-direction: column;

justify-content: space-around;

}

.number{margin-top: 5px;}

//app.js

const board = document.querySelector("[board]");

const letters\_board = document.querySelector(".letters");

const numbers\_board = document.querySelector(".numbers");

const output = document.querySelector("[output")

const field\_matrix = [];

const letters = ["A", "B", "C", "D", "E", "F", "G", "H"];

const numbers = [8, 7, 6, 5, 4, 3, 2, 1];

const coordinates = Array.from(letters, letter => Array.from(numbers, number => letter + number));

const transposedCoordinates = coordinates[0].map((\_, colIndex) => coordinates.map(row => row[colIndex]));

for(let i = 0; i < 8; i++){

const field = document.createElement("div");

field.textContent = numbers[i]

field.classList.add("number");

numbers\_board.appendChild(field);

}

for(let i = 0; i < 8; i++){

const field = document.createElement("div");

field.textContent = letters[i]

field.classList.add("letter");

letters\_board.appendChild(field);

}

for (let i = 0; i < 8; i++) {

const row = [];

for (let j = 0; j < 8; j++) {

const field = document.createElement("div");

field.classList.add("field");

field.setAttribute("i", i);

field.setAttribute("j", j);

field.setAttribute("coordinate", transposedCoordinates[i][j])

if ((i + j) % 2 === 0) {

field.classList.add("white");

} else {

field.classList.add("black");

}

board.appendChild(field);

row.push(field);

}

field\_matrix.push(row);

}

board.addEventListener("click", (event) => {

const clickedField = event.target;

if (clickedField.classList.contains("field")) {

const x = Number(clickedField.getAttribute("i"));

const y = Number(clickedField.getAttribute("j"));

restart();

output.textContent = clickedField.getAttribute("coordinate")

draw(x, y);

}

});

const draw = (x, y) => {

// row and column

for (let i = 0; i < 8; i++) {

field\_matrix[i][y].classList.add("row-col");

field\_matrix[x][i].classList.add("row-col");

}

// diagonals

for (let i = x, j = y; i < 8 && j < 8; i++, j++) {

field\_matrix[i][j].classList.add("diagonal");

}

for (let i = x, j = y; i >= 0 && j >= 0; i--, j--) {

field\_matrix[i][j].classList.add("diagonal");

}

for (let i = x, j = y; i < 8 && j >= 0; i++, j--) {

field\_matrix[i][j].classList.add("diagonal");

}

for (let i = x, j = y; i >= 0 && j < 8; i--, j++) {

field\_matrix[i][j].classList.add("diagonal");

}

field\_matrix[x][y].classList.add("captured");

}

const restart = () => {

field\_matrix.forEach((row) => {

output.textContent = ""

row.forEach((field) => {

field.classList.remove("row-col");

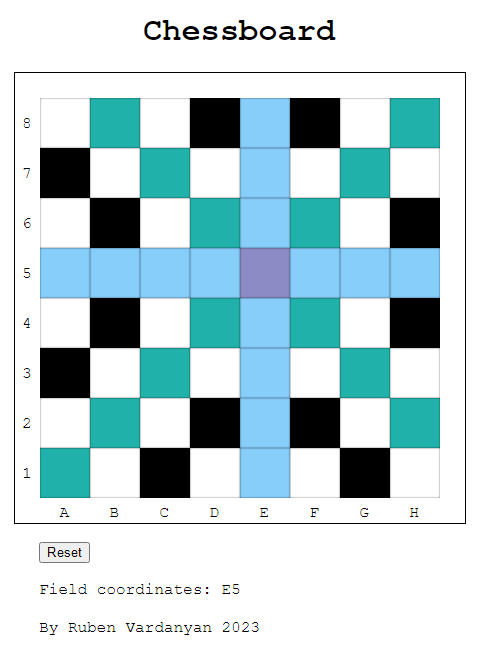
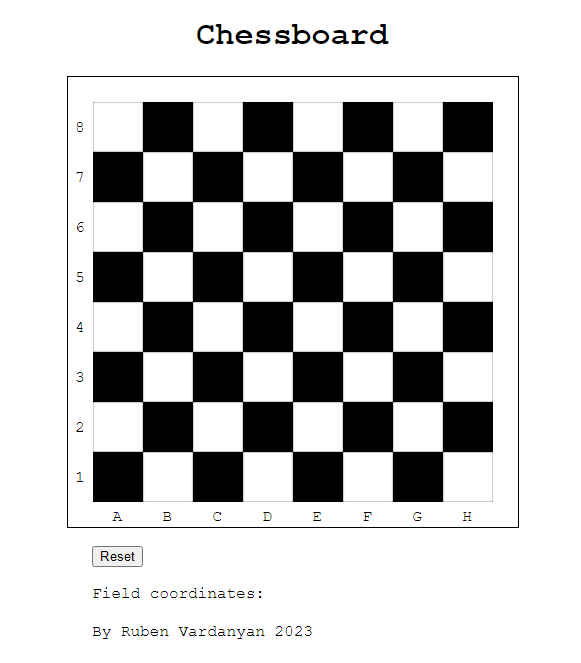
field.classList.remove("diagonal");

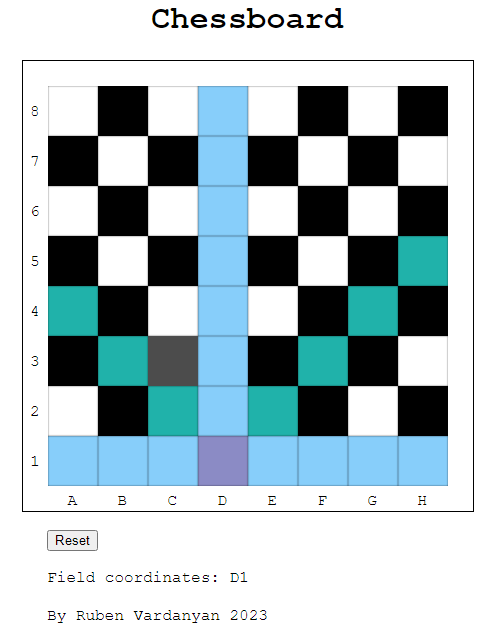
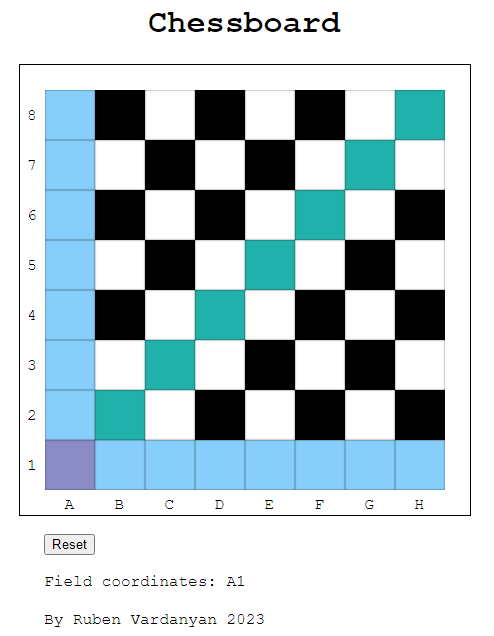
field.classList.remove("captured");

});

});

};.number{margin-top: 5px;}



*Խնդիր 2*

HTML տիրույթում ստեղծել ամսաթվի ներմուծման դաշտ, որտեղ օգտագործողը պետք է մուտքագրի իր ծննդյան ամսաթիվը։ JavaScript-ի միջոցով ստեղծել ծրագիր, որը կհաշվի և կարտածի օգտագործողի տարիքն ըստ իր ներմուծած ամսաթվի։

<!—index.html -->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<link rel="stylesheet" href="style.css">

</head>

<body>

<div class="main">

<h1>Age calculator</h1>

<p>Enter your birth date</p>

<input type="date" id="age" name="trip-start" value="2005-03-16" min="1900-01-01" max="2023-12-13" />

<button class="btn" onclick="calculateAge()">

Calculate Your Age

</button>

<p class="age\_output"></p>

</div>

<script src="app.js"></script>

</body>

</html>

/\* style.css \*/

body{

margin: 0;padding: 0; font-family:'Courier New', Courier, monospace;

background: #131417;

}

div{box-sizing: border-box;}

.main{

width: 350px; position: absolute; top: 50%; left: 50%;

transform: translate(-50%,-50%); background: #2c303a;

border-radius: 10px; padding: 0 20px; color: #f8f8f8;

}

h1, p{text-align: center;}

#age{

display: block;

margin: 25px auto;

padding: 15px;

border-radius: 10px;

border: none;

outline: none;

background: #47cf73;

}

#age:focus{background: #248c46; color: #f8f8f8;}

.btn{

display: block; margin: 25px auto; padding: 10px 16px; border-radius: 5px;

border: none; outline: none; background: #444857; color: #f8f8f8;font-size: 15px;

}

.btn:hover{background: #3d404f;}

//app.js

function calculateAge() {

var dob = new Date(document.getElementById('age').value);

var today = new Date();

var age = today.getFullYear() - dob.getFullYear();

if (today.getMonth() < dob.getMonth() ||

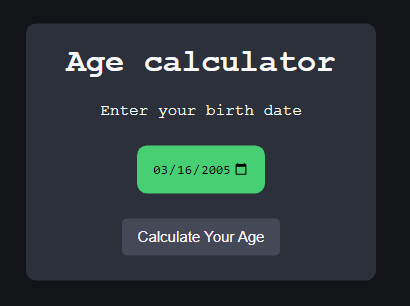
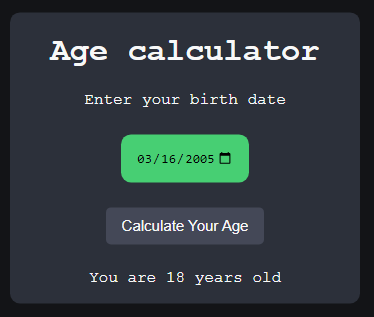
(today.getMonth() === dob.getMonth() && today.getDate() < dob.getDate())) {

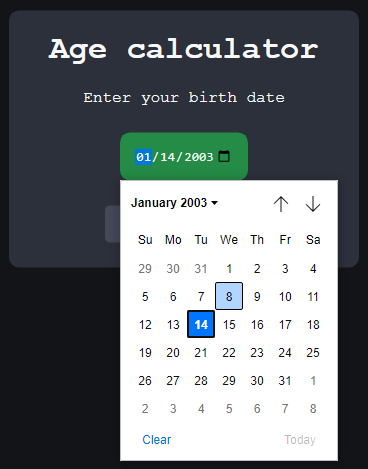
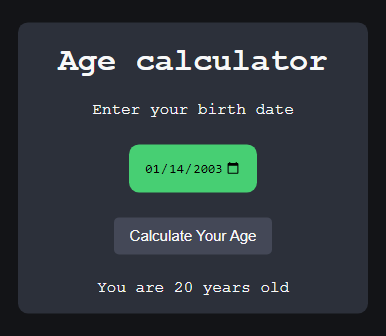
age--;

}

document.querySelector('.age\_output').textContent = ` You are ${age} year${age > 1 ? "s" : ""} old`

}

**Օգտագործված գրականություն**

* <https://www.w3schools.com/js/js_htmldom.asp>
* <https://www.geeksforgeeks.org/dom-document-object-model/>
* <https://developer.mozilla.org/en-US/docs/Web/API/Document_Object_Model/Introduction>
* <https://learn.javascript.ru/dom-nodes>
* <https://www.javascripttutorial.net/javascript-dom/>