**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"**

Інститут **КНІТ**

Кафедра **ПЗ**

ЗВІТ

До лабораторної роботи № 1

**З дисципліни:** *“* *Програмування в Інтернет”*

**На тему:** *“* Структура [DOM](http://www.w3.org/TR/2004/PR-DOM-Level-3-Core-20040205/introduction.html) та методи доступу до вузлів дерева.*”*

**Лектор:**

проф. каф. ПЗ

Яковина В.С.

**Виконав:**

ст. гр. ПЗ-23 Михалевич П.-І. В.

**Прийняв:**

доц. каф. ПЗ

Грицай О.Д.

« » 2022 р.

∑= .

Львів – 2022

**Тема:** Структура [DOM](http://www.w3.org/TR/2004/PR-DOM-Level-3-Core-20040205/introduction.html) та методи доступу до вузлів дерева.

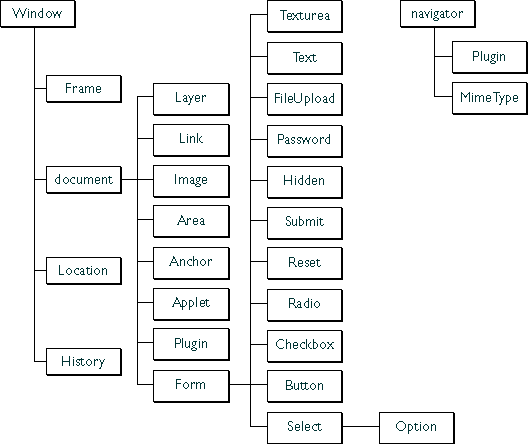
**Мета:** Ознайомитись з ієрархічною структурою об’єктів JavaScript та об’єктами документа і броузера.

**Теоретичні відомості**

## Модель DOM

web-сторінка може мати вигляд дерева, вузли якого є об’єктами, до властивостей яких доступаються операторами мови програмування Javascript.

Модель DOM (Document Object Model, об’єктна модель документа) містить низку стандартних глобальних об’єктів. Зокрема, це window, navigator, document, screen, history, location. Ієрархію основних об’єктів у моделі DOM подано   
на рис. 1.2.



*Рис. 1.2. Дерево моделі DOM об’єкта window браузера*

Перерахуємо глобальні об’єкти та їх призначення, а також розглянемо методи та властивості глобальних об’єктів DOM:

* [window](http://www.w3schools.com/htmldom/dom_obj_window.asp) – вікно браузера, найвищий об’єкт ієрархії DOM. Усі глобальні змінні стають частинами об’єкта window. Його методи такі:

[alert](http://www.w3schools.com/htmldom/met_win_alert.asp), [blur](http://www.w3schools.com/htmldom/met_win_blur.asp), [clearInterval](http://www.w3schools.com/htmldom/met_win_clearInterval.asp), [clearTimeout](http://www.w3schools.com/htmldom/met_win_clearTimeout.asp), [close](http://www.w3schools.com/htmldom/met_win_close.asp), [confirm](http://www.w3schools.com/htmldom/met_win_confirm.asp), [focus](http://www.w3schools.com/htmldom/met_win_focus.asp), [moveBy](http://www.w3schools.com/htmldom/met_win_moveBy.asp), [moveTo](http://www.w3schools.com/htmldom/met_win_moveTo.asp), [open](http://www.w3schools.com/htmldom/met_win_open.asp), [print](http://www.w3schools.com/htmldom/met_win_print.asp), [prompt](http://www.w3schools.com/htmldom/met_win_prompt.asp), [resizeBy](http://www.w3schools.com/htmldom/met_win_resizeBy.asp), [resizeTo](http://www.w3schools.com/htmldom/met_win_resizeTo.asp), [scrollBy](http://www.w3schools.com/htmldom/met_win_scrollBy.asp), [scrollTo](http://www.w3schools.com/htmldom/met_win_scrollTo.asp), [setInterval](http://www.w3schools.com/htmldom/met_win_setInterval.asp), [setTimeout](http://www.w3schools.com/htmldom/met_win_setTimeout.asp)

Властивості: [document](http://www.w3schools.com/htmldom/dom_obj_document.asp), [history](http://www.w3schools.com/htmldom/dom_obj_history.asp), [location](http://www.w3schools.com/htmldom/dom_obj_location.asp), [name](http://www.w3schools.com/htmldom/prop_win_name.asp)

* [navigator](http://www.w3schools.com/htmldom/dom_obj_navigator.asp) – інформація про браузер.

Властивості: [appName](http://www.w3schools.com/htmldom/prop_nav_appname.asp), [appVersion](http://www.w3schools.com/htmldom/prop_nav_appversion.asp), [browserLanguage](http://www.w3schools.com/htmldom/prop_nav_browserlanguage.asp), [cookieEnabled](http://www.w3schools.com/htmldom/prop_nav_cookieenabled.asp), [platform](http://www.w3schools.com/htmldom/prop_nav_platform.asp), [userAgent](http://www.w3schools.com/htmldom/prop_nav_useragent.asp)

* [screen](http://www.w3schools.com/htmldom/dom_obj_screen.asp) – інформація про екран, який використовує браузер.

Властивості: [availHeight](http://www.w3schools.com/htmldom/prop_screen_availheight.asp), [availWidth](http://www.w3schools.com/htmldom/prop_screen_availwidth.asp), [colorDepth](http://www.w3schools.com/htmldom/prop_screen_colordepth.asp), [height](http://www.w3schools.com/htmldom/prop_screen_height.asp), [pixelDepth](http://www.w3schools.com/htmldom/prop_screen_pixeldepth.asp), [width](http://www.w3schools.com/htmldom/prop_screen_width.asp)

* [history](http://www.w3schools.com/htmldom/dom_obj_history.asp) – список викликаних сторінок.

Властивості: [length](http://www.w3schools.com/htmldom/prop_his_length.asp)

Методи: [back](http://www.w3schools.com/htmldom/met_his_back.asp), [forward](http://www.w3schools.com/htmldom/met_his_forward.asp), [go](http://www.w3schools.com/htmldom/met_his_go.asp)

* [location](http://www.w3schools.com/htmldom/dom_obj_location.asp) – URL поточної web-сторінки.

Властивості: [host](http://www.w3schools.com/htmldom/prop_loc_host.asp), [hostname](http://www.w3schools.com/htmldom/prop_loc_hostname.asp), [href](http://www.w3schools.com/htmldom/prop_loc_href.asp), [pathname](http://www.w3schools.com/htmldom/prop_loc_pathname.asp), [port](http://www.w3schools.com/htmldom/prop_loc_port.asp), [protocol](http://www.w3schools.com/htmldom/prop_loc_protocol.asp), [search](http://www.w3schools.com/htmldom/prop_loc_search.asp)

Методи: [assign](http://www.w3schools.com/htmldom/met_loc_assign.asp), [reload](http://www.w3schools.com/htmldom/met_loc_reload.asp), [replace](http://www.w3schools.com/htmldom/met_loc_replace.asp)

* [document](http://www.w3schools.com/htmldom/dom_obj_document.asp) – поточна Web-сторінка.

Властивості: [anchors](http://www.w3schools.com/htmldom/coll_doc_anchors.asp), body, [cookie](http://www.w3schools.com/htmldom/prop_doc_cookie.asp), [domain](http://www.w3schools.com/htmldom/prop_doc_domain.asp), [forms](http://www.w3schools.com/htmldom/coll_doc_forms.asp), [images](http://www.w3schools.com/htmldom/coll_doc_images.asp), [links](http://www.w3schools.com/htmldom/coll_doc_links.asp), [referrer](http://www.w3schools.com/htmldom/prop_doc_referrer.asp), [title](http://www.w3schools.com/htmldom/prop_doc_title.asp), [URL](http://www.w3schools.com/htmldom/prop_doc_url.asp)

Методи: [close](http://www.w3schools.com/htmldom/met_doc_close.asp), [getElementById](http://www.w3schools.com/htmldom/met_doc_getelementbyid.asp), [getElementsByName](http://www.w3schools.com/htmldom/met_doc_getelementsbyname.asp), [open](http://www.w3schools.com/htmldom/met_doc_open.asp), [write](http://www.w3schools.com/htmldom/met_doc_write.asp), [writeln](http://www.w3schools.com/htmldom/met_doc_writeln.asp), [getElementsByTagName](http://www.w3schools.com/htmldom/met_doc_getelementsbytagname.asp)

Приклад ієрархічного дерева DOM для об’єкта document наведено на рис. 1.3.

За змістом вузли дерева поділяють на три типи:

* **елементів** (HTML теги) – породжені вузли або атрибути;
* **тексту**  (текст або блоки) – не мають породжених вузлів чи атрибутів;
* атрибутів (пари атрибут/значення) – не мають породжених вузлів чи атрибутів.

**Типи вузлів та методи доступу до них**

За місцем розташування на дереві виділено такі типи вузлів: parent – батьківський, child – породжений, sibling – сусідній, firstchild – перший породжений, lastchild – останній породжений.

Назви вузлів є складовими назвами властивостей вузлів та параметрів методів, які ними оперують (ліквідують, модифікують, додають тощо). Вузли DOM-дерева мають такі властивості:

* firstChild, lastChild – початковий та кінцевий породжені вузли;
* childNodes – масив породжених вузлів;
* nextSibling, previousSibling – сусідні вузли (по одному), що мають одного батька;
* parentNode – елемент, що породив цей вузол.

Мова програмування дає змогу змінювати склад вузлів, вигляд та зміст web-сторінки. На моделі DOM програміст має можливість виконувати операції двох типів: на рівні вузлів для зміни структури дерева додаванням або відніманням вузла та заміни вузла; на рівні вузла для зміни змісту елемента. Зокрема, для першого типу операцій застосовуються методи:

* [appendChild](http://www.w3schools.com/dom/met_node_appendchild.asp)(node): розміщає вузол у кінці списку породжених вузлів до заданого;
* [insertBefore](http://www.w3schools.com/dom/met_node_insertbefore.asp)(newChild, oldChild): розміщає новий вузол перед вузлом oldChild у списку породжених вузлів до заданого;
* [removeChild](http://www.w3schools.com/dom/met_node_removechild.asp)(node): знищує наведений вузол зі списку породжених вузлів до заданого;
* [replaceChild](http://www.w3schools.com/dom/met_node_replacechild.asp)(newChild, oldChild): заміняє пород­жений на новий вузол.

Властивості набувають значення

* + nodeType: 1 – елемент, 2 – атрибут, 3 – текст, 4 – коментарі, 9 – документ.
  + nodeName: повертає версію тега, наприклад, "div" чи "article". Текстові вузли мають назву "#text" . Назва вузла document є "#document" ,
  + nodeValue: текст вузла або значення атрибута.

Доступ до вузлів DOM за тегами чи ідентифікаторами здійснюється методами:

* document.getElementById("id") – елемент;
* element.getElementsByTagName("tag") – усі по­роджені вузли;

document.createElement("tag") - команда створює новий порожній вузол для подання елемента певного типу.

**Завдання**

1. Пройти наступні частини [курсу по html](https://www.codecademy.com/learn/learn-html) на codecademy: [Elements and Structure](https://www.codecademy.com/courses/learn-html/lessons/intro-to-html/exercises/intro) та [Tables](https://www.codecademy.com/courses/learn-html/lessons/html-tables/exercises/why-tables)

2. Розробити web-сторінку яка відображатиме поле для гри у шахи з положенням фігур головоломки узятої із [списку головоломок](https://www.chess.com/puzzles/problems) (головоломку обрати відносно номеру розташування у списку відповідно до вашого порядкового номера). Сторінка має:

* містити елементи HTML для занесення даних гравця,
* використовувати HTML таблицю,
* містити посилання на правила шахів та головоломку.
* містити зображення фігур.

**Хід роботи**

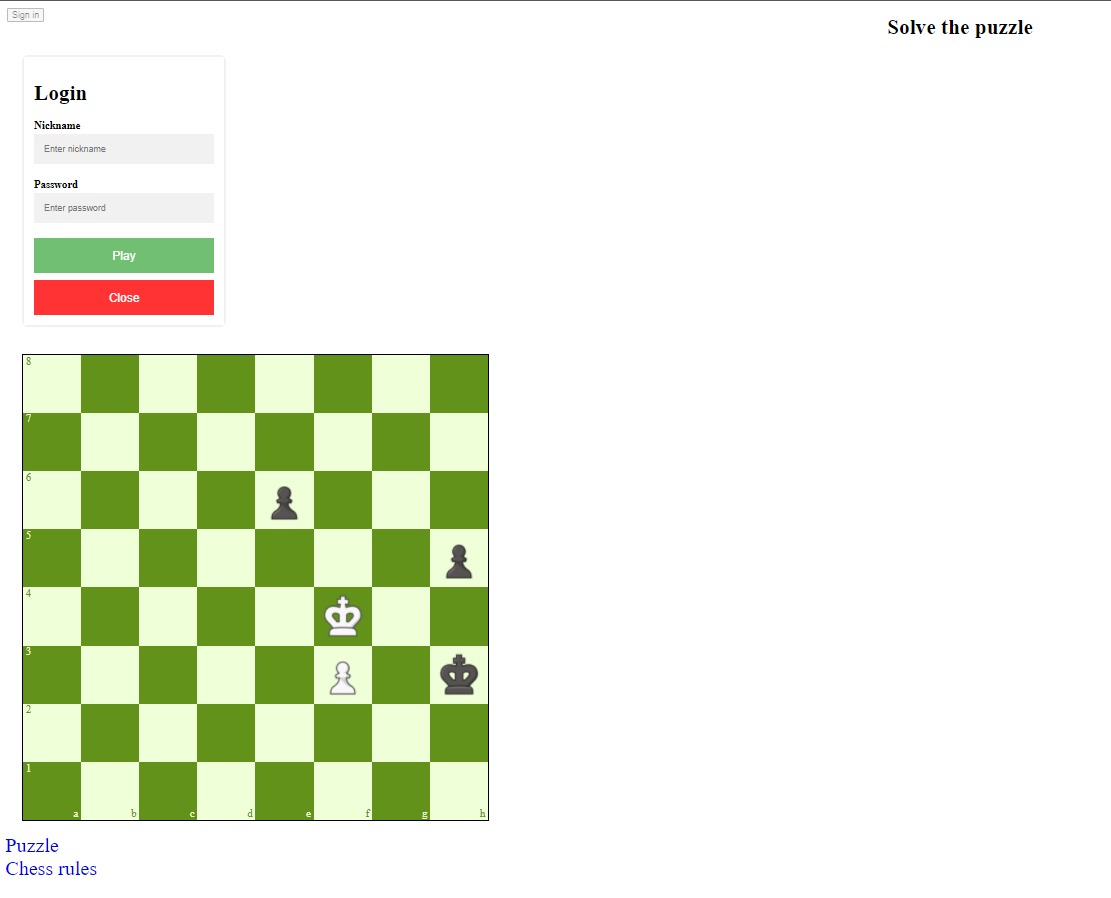
****

Рис.1. Скріншот вигляду сайту

Код програми:

Chess.html

<!DOCTYPE html>

<html>

    <head>

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

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

        <meta charset="utf-8">

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

        <title>

            Chess game

        </title>

    </head>

    <body>

        <h1 style="text-align: center;">

            Solve the puzzle

        </h1>

        <button type="submit" class="openForm" onclick="openForm()">Sign in</button>

        <div class="Form" id="Form">

            <form class="registration">

                <h1>Login</h1>

                <label class="label" for="email"><b>Nickname</b></label>

                <input type="text" placeholder="Enter nickname" name="email" required>

                <label class="label" for="psw"><b>Password</b></label>

                <input type="password" placeholder="Enter password" name="psw" required>

                <button type="submit" class="btn">Play</button>

                <button type="submit" class="btn cancel" onclick="closeForm()">Close</button>

            </form>

        </div>

        <div>

            <table>

                <tr>

                    <td><div class = "cell"><span class = "number green">8</span></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                </tr>

                <tr>

                    <td><div class = "cell"><span class = "number white">7</span></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                </tr>

                <tr>

                    <td><div class = "cell"><span class = "number green">6</span></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"><img class="figure black pawn" src="Resources/bp.png" alt="black pawn" onclick="TagCell(event)"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                </tr>

                <tr>

                    <td><div class = "cell"><span class = "number white">5</span></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"><img class="figure black pawn" src="Resources/bp.png" alt="black pawn" onclick="TagCell(event)"></div></td>

                </tr>

                <tr>

                    <td><div class = "cell"><span class = "number green">4</span></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"><img class="figure white king" src="Resources/wk.png" alt="white king" onclick="TagCell(event)"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                </tr>

                <tr>

                    <td><div class = "cell"><span class = "number white">3</span></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"><img class="figure white pawn" src="Resources/wp.png" alt="white pawn" onclick="TagCell(event)"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"><img class="figure black king" src="Resources/bk.png" alt="black king" onclick="TagCell(event)"></div></td>

                </tr>

                <tr>

                    <td><div class = "cell"><span class = "number green">2</span></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                    <td><div class = "cell"></div></td>

                </tr>

                <tr>

                    <td><div class = "cell"><span class = "number white">1</span><span class = "letter white">a</span></div></td>

                    <td><div class = "cell"><span class = "letter green">b</span></div></td>

                    <td><div class = "cell"><span class = "letter white">c</span></div></td>

                    <td><div class = "cell"><span class = "letter green">d</span></div></td>

                    <td><div class = "cell"><span class = "letter white">e</span></div></td>

                    <td><div class = "cell"><span class = "letter green">f</span></div></td>

                    <td><div class = "cell"><span class = "letter white">g</span></div></td>

                    <td><div class = "cell"><span class = "letter green">h</span></div></td>

                </tr>

            </table>

        </div>

        <nav>

            <ul>

                <li><a href="https://www.chess.com/puzzles/problem/491418">Puzzle</a></li>

                <li><a href="https://en.wikipedia.org/wiki/Rules\_of\_chess">Chess rules</a></li>

            </ul>

        </nav>

    </body>

</html>

Chess.css

table {

    width: 700px;

    height: 700px;

    border: 1px solid black;

    border-collapse: collapse;

    margin-left: 25px;

}

td {

    width: 12.5%;

    height: 12.5%;

}

table tr:nth-child(even) td:nth-child(even), table tr:nth-child(odd) td:nth-child(odd){

    background-color: rgb(239, 255, 215);

}

table tr:nth-child(odd) td:nth-child(even), table tr:nth-child(even) td:nth-child(odd){

    background-color: rgb(98, 146, 26);

}

img.figure {

    position: absolute;

    width: 85%;

    height: 85%;

    top: 50%;

    left: 50%;

    transform: translate(-50%, -50%);

}

img.circle {

    position: absolute;

    width: 30%;

    height: 30%;

    top: 50%;

    left: 50%;

    transform: translate(-50%, -50%);

}

ul {

    font-size: 30px;

    list-style-type: none;

    margin: 0;

    margin-top: 20px;

    padding: 0;

}

span.green{

    color: rgb(98, 146, 26);

}

span.white{

    color: rgb(239, 255, 215);

}

a {

    text-decoration: none;

}

a:hover {

    color: red;

    text-decoration: underline;

}

.Form {

    border: 3px solid #f1f1f1;

    border-radius: 10px;

    max-width: 300px;

    margin-left: 25px;

    margin-top: 15px;

    margin-bottom: 40px;

}

.openForm {

    position: absolute;

    top: 10px;

    left: 10px;

    text-decoration: none;

    opacity: 0.5;

}

.openForm:hover {

    opacity: 1;

}

.registration {

    max-width: 300px;

    padding: 15px;

    background-color: white

}

.registration input[type=text], .registration input[type=password] {

    width: 89%;

    padding: 15px;

    margin: 5px 0 22px 0;

    border: none;

    background: #f1f1f1;

}

.registration .btn {

    font-size: large;

    background-color: #4CAF50;

    color: white;

    padding: 16px 20px;

    border: none;

    cursor: pointer;

    width: 100%;

    margin-bottom:10px;

    opacity: 0.8;

}

.registration .cancel {

    background-color: red;

    margin-bottom:0px;

}

img.figure:hover{

    opacity: 0.75;

}

span.letter{

    position: absolute;

    bottom: 0px;

    right: 3px;

}

span.number{

    position: absolute;

    top: 0px;

    left: 3px;

}

div.cell{

    position: relative;

    width: 100%;

    height: 100%;

}

Chess.js

function closeForm() {

    document.getElementById("Form").style.display = "none";

}

function openForm() {

    document.getElementById("Form").style.display = "block";

}

function GetTr(img){

    return img.parentElement.parentElement.parentElement;

}

function GetTd(img){

    return img.parentElement.parentElement;

}

function GetDiv(img){

    return img.parentElement;

}

var isWhiteTurn = true;

const circleElements = [];

const enemyElements = [];

var checkedElement;

function TagCell(e) {

    var img = document.createElement("img");

    img.src = "Resources/redCircle.png";

    img.className = "circle";

    img.setAttribute("onclick","MoveFigure(event);");

    var figure = e.srcElement || e.target;

    while(circleElements.length){

        GetDiv(circleElements[circleElements.length - 1]).removeChild(circleElements[circleElements.length - 1]);

        circleElements.pop();

    }

    if(enemyElements.includes(figure)){

        var cell = GetDiv(figure);

        cell.removeChild(figure);

        cell.appendChild(checkedElement);

        isWhiteTurn = !isWhiteTurn;

        while(enemyElements.length){

            enemyElements[enemyElements.length-1].style.border = "";

            enemyElements[enemyElements.length-1].style.borderRadius = "";

            enemyElements.pop();

        }

        return;

    }

    if((isWhiteTurn && figure.className.includes("black")) || (!isWhiteTurn && figure.className.includes("white"))){

        return;

    }

    while(enemyElements.length){

        enemyElements[enemyElements.length-1].style.border = "";

        enemyElements[enemyElements.length-1].style.borderRadius = "";

        enemyElements.pop();

    }

    checkedElement = figure;

    // White and black kings.

    if (figure.className.includes("white king") || figure.className.includes("black king")){

        // without border.

        if((GetTr(figure).rowIndex != 7 && GetTr(figure).rowIndex != 0)

            && (GetTd(figure).cellIndex != 0 && GetTd(figure).cellIndex!= 7)){

            var childrens = GetTr(figure).previousElementSibling.children;

            var leftUpperElement;

            for (var i = 0; i < childrens.length; ++i) {

                if(childrens[i].cellIndex == (GetTd(figure).cellIndex - 1)){

                    leftUpperElement = childrens[i];

                    break;

                }

            }

            TagSquare(leftUpperElement, 3, 3, figure.className);

        }

        // Right column

        else if(GetTd(figure).cellIndex == 7){

            // Right lower

            if(GetTr(figure).rowIndex == 7){

                var childrens = GetTr(figure).previousElementSibling.children;

                var leftUpperElement;

                for (var i = 0; i < childrens.length; ++i) {

                    if(childrens[i].cellIndex == (GetTd(figure).cellIndex - 1)){

                       leftUpperElement = childrens[i];

                       break;

                    }

                }

                TagSquare(leftUpperElement, 2, 2, figure.className);

            }

            // Right upper

            else if(GetTr(figure).rowIndex == 0){

                var leftUpperElement = GetTd(figure).previousElementSibling;

                TagSquare(leftUpperElement, 2, 2, figure.className);

            }

            // Others

            else{

                var childrens = GetTr(figure).previousElementSibling.children;

                var leftUpperElement;

                for (var i = 0; i < childrens.length; ++i) {

                    if(childrens[i].cellIndex == (GetTd(figure).cellIndex - 1)){

                        leftUpperElement = childrens[i];

                        break;

                    }

                }

                TagSquare(leftUpperElement, 2, 3, figure.className);

            }

        }

        // Left column

        else if(GetTd(figure).cellIndex == 0){

            // Left lower

            if(GetTr(figure).rowIndex == 7){

                var childrens = GetTr(figure).previousElementSibling.children;

                var leftUpperElement;

                for (var i = 0; i < childrens.length; ++i) {

                    if(childrens[i].cellIndex == GetTd(figure).cellIndex){

                       leftUpperElement = childrens[i];

                       break;

                    }

                }

                TagSquare(leftUpperElement, 2, 2, figure.className);

            }

            // Left upper

            else if(GetTr(figure).rowIndex == 0){

                var leftUpperElement = GetTd(figure);

                TagSquare(leftUpperElement, 2, 2, figure.className);

            }

            // Others

            else{

                var childrens = GetTr(figure).previousElementSibling.children;

                var leftUpperElement;

                for (var i = 0; i < childrens.length; ++i) {

                    if(childrens[i].cellIndex == GetTd(figure).cellIndex){

                        leftUpperElement = childrens[i];

                        break;

                    }

                }

                TagSquare(leftUpperElement, 2, 3, figure.className);

            }

        }

        // Upper and lower row without left and right cols

        else {

            // Lower

            if(GetTr(figure).rowIndex == 7){

                var childrens = GetTr(figure).previousElementSibling.children;

                var leftUpperElement;

                for (var i = 0; i < childrens.length; ++i) {

                    if(childrens[i].cellIndex == (GetTd(figure).cellIndex - 1)){

                       leftUpperElement = childrens[i];

                       break;

                    }

                }

                TagSquare(leftUpperElement, 3, 2, figure.className);

            }

            // Upper

            else{

                var leftUpperElement = GetTd(figure).previousElementSibling;

                TagSquare(leftUpperElement, 3, 2, figure.className);

            }

        }

    }

    // White and black pawn.

    else if (figure.className.includes("white pawn") || figure.className.includes("black pawn")){

        if((GetTr(figure).rowIndex != 0 && figure.className.includes("white")) || (GetTr(figure).rowIndex != 7 && figure.className.includes("black"))){

            var childrens;

            if(figure.className.includes("white")){

                childrens = GetTr(figure).previousElementSibling.children;

            }else if(figure.className.includes("black")){

                childrens = GetTr(figure).nextElementSibling.children;

            }

            for (var i = 0; i < childrens.length; ++i) {

                if((childrens[i].cellIndex == (GetTd(figure).cellIndex - 1) && (childrens[i].children)[0].hasChildNodes())

                    || (childrens[i].cellIndex == (GetTd(figure).cellIndex + 1) && (childrens[i].children)[0].hasChildNodes())){

                    var divChilds = (childrens[i].children)[0].children;

                    for(var j = 0; j < divChilds.length; ++j){

                        if(divChilds[j].className.includes("figure")

                        && ((divChilds[j].className.includes("black")  && figure.className.includes("white"))

                        || (divChilds[j].className.includes("white") && figure.className.includes("black")))){

                            enemyElements.push(divChilds[j]);

                            break;

                        }

                    }

                }

                else if(childrens[i].cellIndex == GetTd(figure).cellIndex){

                    if(!(childrens[i].children)[0].hasChildNodes()){

                        (childrens[i].children)[0].appendChild(img);

                        circleElements.push(img);

                    }else{

                        var divChilds = (childrens[i].children)[0].children;

                        var hasFigure = false;

                        for(var j = 0; j < divChilds.length; ++j){

                            if(divChilds[j].className.includes("figure")){

                                hasFigure = true;

                            }

                        }

                        if(!hasFigure){

                            (childrens[i].children)[0].appendChild(img);

                            circleElements.push(img);

                        }

                    }

                }

                else if(childrens[i].cellIndex == (GetTd(figure).cellIndex + 2)){

                    break;

                }

            }

        }

    }

    for(var enIdx = 0; enIdx < enemyElements.length; ++enIdx){

        enemyElements[enIdx].style.border = "2px dashed red";

        enemyElements[enIdx].style.borderRadius = "3px";

    }

}

function TagSquare(leftUpperElement, colCount, rowCount, figureClassName){

    var i = 0;

    var j = 0;

    var currentElement = leftUpperElement;

    while (true){

        while(true){

            var img = document.createElement("img");

            img.src = "Resources/redCircle.png";

            img.className = "circle";

            img.setAttribute("onclick","MoveFigure(event);");

            if(!(currentElement.children)[0].hasChildNodes()){

                (currentElement.children)[0].appendChild(img);

                circleElements.push(img);

            }else{

                var divChilds = (currentElement.children)[0].children;

                var hasFigure = false;

                for(var m = 0; m < divChilds.length; ++m){

                    if(divChilds[m].className.includes("figure")){

                        hasFigure = true;

                        if((divChilds[m].className.includes("white") && figureClassName.includes("black"))

                        || (divChilds[m].className.includes("black") && figureClassName.includes("white"))){

                            enemyElements.push(divChilds[m]);

                        }

                    }

                }

                if(!hasFigure){

                    (currentElement.children)[0].appendChild(img);

                    circleElements.push(img);

                }

            }

            ++j;

            if (j == colCount){

                j = 0;

                break;

            }

            currentElement = currentElement.nextElementSibling;

        }

        ++i;

        if (i >= rowCount){

            break;

        }

        var childrens = currentElement.parentElement.nextElementSibling.children;

        for (var n = 0; n < childrens.length; ++n) {

            if(childrens[n].cellIndex == leftUpperElement.cellIndex ){

                currentElement = childrens[n];

            }

        }

    }

}

function MoveFigure(e){

    var figure = e.srcElement || e.target;

    var cell = GetDiv(figure);

    while(circleElements.length){

        GetDiv(circleElements[circleElements.length - 1]).removeChild(circleElements[circleElements.length - 1]);

        circleElements.pop();

    }

    while(enemyElements.length){

        enemyElements[enemyElements.length-1].style.border = "";

        enemyElements[enemyElements.length-1].style.borderRadius = "";

        enemyElements.pop();

    }

    cell.appendChild(checkedElement);

    isWhiteTurn = !isWhiteTurn;

}

**Висновок**

На даній лабораторній роботі я вивчив основи html, css та js. Навчився використовувати основні теги html та стилі css, та створив простий сайт за допомогою цього, також продумав логіку ходів шахматних фігур за допомогою js.