Міністерство освіти і науки України

Національний технічний університет України

«Київський Політехнічний Інститут»ґ

ім. І. Сікорського

Факультет прикладної математики

Кафедра «Системного програмування і спеціалізованих комп’ютерних систем»

**Лабораторна робота №3**

З дисципліни «Web дизайн»

На тему «**Розробка програм асинхронного обміну даними у Web-середовищі засобами бібліотеки JQuery. Ajax**»

**Виконав: Перевірив(ла):**

студентка групи КВ-43

Бойко Данило

Київ – 2017

**Лабораторна робота №3.**

**Метою**лабораторної роботи є вивчення основ асинхронного обміну даними мовою JavaScript за допомогою технології Ajax та бібліотеки JQuery при реалізації практичних задач Web-програмування.

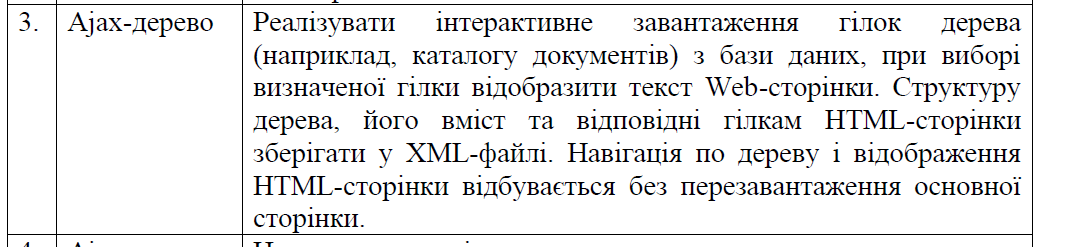
**Завдання**на лабораторну роботу:

1. Ознайомитись із синтаксисом та базовими можливостями бібліотеки JQuery.

2. Набути практичних навичок маніпулювання вмістом Web-сторінок.

3. Вивчити способи реалізації інтерактивного інтерфейсу користувача за допомогою засобів JQuery Ajax.

**Варіант 3.**



**Текст програми:   
1.** <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html>

<head><TITLE>Tree Example</TITLE>

<script type="text/javascript">

function tree\_toggle(event) {

event = event || window.event

var clickedElem = event.target || event.srcElement

if (!hasClass(clickedElem, 'Expand')) {

return // клик не там

}

// Node, на который кликнули

var node = clickedElem.parentNode

if (hasClass(node, 'ExpandLeaf')) {

return // клик на листе

}

// определить новый класс для узла

var newClass = hasClass(node, 'ExpandOpen') ? 'ExpandClosed' : 'ExpandOpen'

// заменить текущий класс на newClass

// регексп находит отдельно стоящий open|close и меняет на newClass

var re = /(^|\s)(ExpandOpen|ExpandClosed)(\s|$)/

node.className = node.className.replace(re, '$1'+newClass+'$3')

}

function hasClass(elem, className) {

return new RegExp("(^|\\s)"+className+"(\\s|$)").test(elem.className)

}

</script>

<style type="text/css">

.Container {

padding: 0;

margin: 0;

}

.Container li {

list-style-type: none;

}

/\* indent for all tree children excepts root \*/

.Node {

background-image : url(img/i.gif);

background-position : top left;

background-repeat : repeat-y;

margin-left: 18px;

zoom: 1;

}

.IsRoot {

margin-left: 0;

}

/\* left vertical line (grid) for all nodes \*/

.IsLast {

background-image: url(img/i\_half.gif);

background-repeat : no-repeat;

}

.ExpandOpen .Expand {

background-image: url(img/expand\_minus.gif);

}

/\* closed is higher priority than open \*/

.ExpandClosed .Expand {

background-image: url(img/expand\_plus.gif);

}

/\* highest priority \*/

.ExpandLeaf .Expand {

background-image: url(img/expand\_leaf.gif);

}

.Content {

min-height: 18px;

margin-left:18px;

}

\* html .Content {

height: 18px;

}

.Expand {

width: 18px;

height: 18px;

float: left;

}

.ExpandOpen .Container {

display: block;

}

.ExpandClosed .Container {

display: none;

}

.ExpandOpen .Expand, .ExpandClosed .Expand {

cursor: pointer;

}

.ExpandLeaf .Expand {

cursor: auto;

}

</style>

</head>

<body>

<div onclick="tree\_toggle(arguments[0])">

<div>Tree</div>

<ul class="Container">

<li class="Node IsRoot IsLast ExpandClosed">

<div class="Expand"></div>

<div class="Content">Root</div>

<ul class="Container">

<li class="Node ExpandClosed">

<div class="Expand"></div>

<div class="Content">Item 1</div>

<ul class="Container">

<li class="Node ExpandLeaf IsLast">

<div class="Expand"></div>

<div class="Content">Item 1.2</div>

</li>

</ul>

</li>

<li class="Node ExpandLeaf IsLast">

<div class="Expand"></div>

<div class="Content">Item 2</div>

</li>

</ul>

</li>

</ul>

</div>

</body>

</html>

**2.**

.Container {

padding: 0;

margin: 0;

}

.Container li {

list-style-type: none;

}

/\* indent for all tree children excepts root \*/

.Node {

background-image : url(img/i.gif);

background-position : top left;

background-repeat : repeat-y;

margin-left: 18px;

zoom: 1;

}

.IsRoot {

margin-left: 0;

}

/\* left vertical line (grid) for all nodes \*/

.IsLast {

background-image: url(img/i\_half.gif);

background-repeat : no-repeat;

}

.ExpandOpen .Expand {

background-image: url(img/expand\_minus.gif);

}

/\* closed is higher priority than open \*/

.ExpandClosed .Expand {

background-image: url(img/expand\_plus.gif);

}

/\* highest priority \*/

.ExpandLeaf .Expand {

background-image: url(img/expand\_leaf.gif);

}

.Content {

min-height: 18px;

margin-left:18px;

}

\* html .Content {

height: 18px;

}

.Expand {

width: 18px;

height: 18px;

float: left;

}

.ExpandOpen .Container {

display: block;

}

.ExpandClosed .Container {

display: none;

}

.ExpandOpen .Expand, .ExpandClosed .Expand {

cursor: pointer;

}

.ExpandLeaf .Expand {

cursor: auto;

}

.ExpandLoading {

width: 18px;

height: 18px;

float: left;

background-image: url(img/expand\_loading.gif);

**}**

**3.** function tree(id, url) {

var element = document.getElementById(id)

function hasClass(elem, className) {

return new RegExp("(^|\\s)"+className+"(\\s|$)").test(elem.className)

}

function toggleNode(node) {

// определить новый класс для узла

var newClass = hasClass(node, 'ExpandOpen') ? 'ExpandClosed' : 'ExpandOpen'

// заменить текущий класс на newClass

// регексп находит отдельно стоящий open|close и меняет на newClass

var re = /(^|\s)(ExpandOpen|ExpandClosed)(\s|$)/

node.className = node.className.replace(re, '$1'+newClass+'$3')

}

function load(node) {

function showLoading(on) {

var expand = node.getElementsByTagName('DIV')[0]

expand.className = on ? 'ExpandLoading' : 'Expand'

}

function onSuccess(data) {

if (!data.errcode) {

onLoaded(data)

showLoading(false)

} else {

showLoading(false)

onLoadError(data)

}

}

function onAjaxError(xhr, status){

showLoading(false)

var errinfo = { errcode: status }

if (xhr.status != 200) {

// может быть статус 200, а ошибка

// из-за некорректного JSON

errinfo.message = xhr.statusText

} else {

errinfo.message = 'Некорректные данные с сервера'

}

onLoadError(errinfo)

}

function onLoaded(data) {

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

var child = data[i]

var li = document.createElement('LI')

li.id = child.id

li.className = "Node Expand" + (child.isFolder ? 'Closed' : 'Leaf')

if (i == data.length-1) li.className += ' IsLast'

li.innerHTML = '<div class="Expand"></div><div class="Content">'+child.title+'</div>'

if (child.isFolder) {

li.innerHTML += '<ul class="Container"></ul>'

}

node.getElementsByTagName('UL')[0].appendChild(li)

}

node.isLoaded = true

toggleNode(node)

}

function onLoadError(error) {

var msg = "Ошибка "+error.errcode

if (error.message) msg = msg + ' :'+error.message

alert(msg)

}

showLoading(true)

$.ajax({

url: url,

data: node.id,

dataType: "json",

success: onSuccess,

error: onAjaxError,

cache: false

})

}

element.onclick = function(event) {

event = event || window.event

var clickedElem = event.target || event.srcElement

if (!hasClass(clickedElem, 'Expand')) {

return // клик не там

}

// Node, на который кликнули

var node = clickedElem.parentNode

if (hasClass(node, 'ExpandLeaf')) {

return // клик на листе

}

if (node.isLoaded || node.getElementsByTagName('LI').length) {

// Узел уже загружен через AJAX(возможно он пуст)

toggleNode(node)

return

}

if (node.getElementsByTagName('LI').length) {

// Узел не был загружен при помощи AJAX, но у него почему-то есть потомки

// Например, эти узлы были в DOM дерева до вызова tree()

// Как правило, это "структурные" узлы

// ничего подгружать не надо

toggleNode(node)

return

}

// загрузить узел

load(node)

}

}

**Результати роботи:**

