

## **Методичні підходи до викладення мови "JavaScript" в педагогічних учбових закладах.**

Широке розповсюдження дистанційних форм навчання невід`ємно зв`язано з використанням комп`ютерних технологій як дидактичного засобу навчання[1]. Сучасні мереживі технології, котрі надали можливості оперативної передачі інформації, відкрили доступ до різноманітних інформаційних ресурсів, пропонують послуги проведення дискусій, конференцій тощо стали основоположним дидактичним засобом дистанційних форм навчання[2]. В цьому напрямку вагому роль приділяють гіпертекстові, як зручному інструменту для відображення учбового матеріалу[3], а також новітнім WEB-технологіям, що стрімко популяризуються у сьогоденні. Не треба забувати, що гіпертекстова інформація сучасного Інтернет - це мультимедійна інформація оздоблена інтерактивністю, котра підтримується як зі сторони серверу, так і з боку клієнта. Використання інтерактивності гіпертекстового документа у дистанційному навчанні призвело до розроблення ефективних методик програмної підтримки взаємодії людини, що навчається з стороною контролю та управління учбовим процесом[4]. Така можливість надання інтерактивності розкриває вагомо новітні підходи до реалізації дистанційних форм навчання.

Процес формування інформаційної культури майбутнього викладача інформатики невід`ємно зв`язується з процесом вивчення новітніх мереживих технологій, що набули широкого розповсюдження в різноманітних галузях сучасного життя і, перш за все, в освіті[5]. З цього приводу бачиться доцільним розроблення та впровадження в учбовий процес теоритично-практичного курсу, що охоплює наступні тематичні розділи:

- принципи практичного використання Інтернет в процесі розроблення науково-дослідницьких робіт студентів;
- мова JavaScript - принципи організації інтерактивності на стороні клієнта;
- CGI(common gateway interfase) - принцип обміну даними, інтерактивність, що реалізована на стороні серверу, основи мови Perl.

Цей курс розраховано на 36 лекційних годин та 72 години лабораторно-практичних робіт. В цій статті ми плануємо докладно розглянути саме методику викладання мови JavaScript [6, 7, 8]. Методична структура цього тематичного розділу чітко орієнтується на вирішення основних задач, що ставляться при вивченні даної теми. Ці задачі можуть бути сформульовані таким чином:

1. вивчення методів надання інтерактивності структурним елементам гіпертекстового документу;
2. вивчення принципів, які надають можливість збирати інформацію та обробляти її модулями JavaScript програми;
3. розкриття сутності гіпертекстового документа, як структурованої одиниці об'єктів браузера;
4. поглиблення розуміння основних принципів об'єктно-орієнтованого програмування на прикладі мови JavaScript

Вивчення мови JavaScript базується на чіткому знанні мови HTML(мова розмітки гіпертекстового документу). Кожний студент добре знає те, що вся інформація, котра розміщується між тегами <BODY>...</BODY> являє собою гіпертекстову інформацію, яка відображається браузером. З точки зору JavaScript - це **документ**, який реагує на такі події: onblur (документ став неактивним); onfocus (документ став активним); onload (документ загрузився в вікно); onunload(документ вивантажують з вікна браузера); onkeydown; onkeyup; onpress(події зв'язані з клавіатурою); onclick; onmouseup; onmousedown (події, що пов'язані зі станом миші в середовищі документа). Коли відбувається одна з подій, які ми щойно перелічили, ми можемо викликати запуск програмного модуля мови JavaScript. Це можна зробити так:

```
<BODY onload="ім`я функції([список фактичних параметрів])">
```

В самому документі мовою HTML ми утворювали різні структурні одиниці, а саме:

- малюнки <IMG SRC="URL">;
- гіперпосилки <A HREF="URL">...</A>;

- закладки <A HREF="#NAME">...</A>;
- форми <FORM>...</FORM>.

Всі перелічені структури HTML документа є елементами, що **вкладені** в документ. Це дуже важливо для подальшого розуміння загальної структури компонентів гіпертекстового документу. Крім того, кожна з цих одиниць реагує на власні події. Ці події можуть приводити до виконання деякого модуля JavaScript програми. Так при роботі з малюнками ми можемо відреагувати на наступні події: `onabord` (завантаження з мережі перервана користувачем); `onerror` (виникла помилка при завантаженні малюнка); `onkeydown`, `onkeyup`, `onkeypress` (події, що пов'язані з натиском миші на картині); `onload` (повне завантаження малюнка на екрані користувача). За допомогою мови JavaScript наступ одної з подій можна опрацювати таким чином:

```
<IMG SRC="URL" onabord="ім'я функції([список фактичних параметрів])">
```

Події, на котрі можуть реагувати решта елементів гіпертекстового документу також розглядаються на першому етапі вивчення мови JavaScript. Цей перший крок у процесі вивчення нової мови програмування дуже важливий, тому що:

- основне призначення мови JavaScript - це надання можливих форм інтерактивності гіпертекстового документу, а реагування на події, що виникають на стороні клієнта - це один з проявів інтерактивності;
- виділення структурних елементів гіпертекстового документу з самого початку орієнтує студентів на сприйняття HTML документа, як структурованої ієрархії основних елементів;
- вивчення мови програмування починається з написання функцій, що сприяє розвитку модульного підходу до програмування.

Дуже важливим є той факт, що студенти проводять асоціативну паралель між компонентами середовища Delphi та структурними елементами HTML документа. Це ж і не дивно, тому що кожний елемент гіпертекстового документу, як і компонента мови програмування Delphi є структурним

елементом візуального середовища; реагує на події запуском конкретного програмного модуля.

На першому етапі бачиться дотепним обробляти події основних елементів гіпертекстового документу за допомогою стандартної функції `alert(x)`, за допомогою якої поверх вікна браузера з'являється діалогове вікно, в котрому відображається текст переданий як фактичний параметр. Лише пізніше буде розглядатися `alert(x)`, як метод об'єкта браузера `window`.

Вивчення структурного елемента форми починається з того, що надається основне розуміння цього об'єкта, як інструмента, що збирає інформацію. Саме тому ввід інформації, до якого всі привикли в інших мовах програмування тут виконується за допомогою елементів форми і тільки потім ця інформація може оброблятися мовою JavaScript. Вивід інформації - це також важливий момент взаємодії, і це можна робити за допомогою форми. Форма, як і інші елементи документу реагує на ряд подій, але найважливішим тут бачиться така подія, як `onsubmit`. Ця подія виникає, коли результати заповнення форми відправляються на сервер для подальшого оброблення. Наступ цієї події обробляється функцією, яка може перевіряти наступне: чи правильно заповнені поля форми, чи присутня інформація в обов'язкових полях та інше. Викликається ця подія безпосередньо в тегу FORM (`<FORM onsubmit="ім'я функції([список фактичних параметрів])">`) В формі мовою HTML ми утворювали різні структурні одиниці, за допомогою яких й виконується процес зібрання інформації, а саме:

- однорядкове поле вводу тексту `<INPUT TYPE=text>`;
- однорядкове поле вводу пароля `<INPUT TYPE=password>`;
- однорядкове поле вводу скритої інформації `<INPUT TYPE=hidden>`;
- кнопка `<INPUT TYPE=button>`
- кнопка для передачі даних програмі оброблювачу `<INPUT TYPE=submit>`;
- кнопка для очищення полів форми `<INPUT TYPE=reset>`;
- радіо кнопки `<INPUT TYPE=radio>`;
- чекбокси `<INPUT TYPE=checkbox>`;

- вибір файлу на локальному комп'ютері користувача <INPUT TYPE=file>;
- меню вибору <SELECT>...</SELECT>;
- багаторядкове поле вводу <TEXTAREA>...</TEXTAREA>.

Всі перелічені структури HTML документа є елементами, що **вкладені** в форму. Тепер ми можемо скласти чітку структуру елементів гіпертекстового документу. Ця ієрархія являє собою частину загальної структури об'єктів браузера, з якою ми будемо знайомитись далі. Кожний гіпертекстовий документ відкривається в вікні браузера, тому він є вкладеним в об'єкт браузера вікно.

    window

        document

            images

            links

            anchors

            forms

                elements

Детальний аналіз основних подій, на які реагують основні елементи форми призводить до формування загальної картини нового сприйняття гіпертекстового документу як складної ієрархії об'єктів браузера, що взаємодіють між собою та реагують на широке коло подій.

І так, наш гіпертекстовий документ складається з об'єктів, але ми знаємо, що кожний об'єкт - це в свою чергу складна структурна одиниця, котра наділяється власними методами та властивостями. Як вже було сказано вище ввід та вивід інформації в мові JavaScript виконується за допомогою елементів форми. Саме тому аспекту приділяється увага на другому етапі. І це не випадково, ми вчимося реагувати на інформацію, що отримуємо від користувача. Складемо невеличку форму, в котрій буде однорядкове поле для вводу інформації, поле для виводу інформації та кнопка. Коли клієнт натискає на кнопку, інформація, що була записана в полі вводу, відображається в полі виводу.

...

```
<FORM NAME="form1">
```

```
<P> Поле вводу <INPUT TYPE=text NAME="input1" SIZE=20  
MAXLENGTH=30></P>
```

```
<P> Поле виводу <INPUT TYPE=text NAME="output1" SIZE=20  
MAXLENGTH=30></P>
```

```
<CENTER><INPUT TYPE=button VALUE= "продублюй інформацію"  
></CENTER>
```

```
</FORM>
```

...

Для того, щоб рішити нашу задачу нам необхідно опрацювати подію onclick для елемента форми кнопка. Це в тексті нашого документа буде виглядати саме так:

```
<INPUT TYPE=button VALUE= "продублюй інформацію"  
onclick="document.form1.output1.value=document.form1.input1.value">
```

document.	form1.	output1.	value
Звернення до кожного об'єкта браузера йде чітко в послідовності ієрархії об'єктів браузера. Коректніше було б написати window.document, але в зв'язку з тим, що об'єкт window є зовнішнім для всіх об'єктів браузера, запис прийнято спрощувати.	Ми працюємо з елементом форми. Ми вказуємо до якої форми відноситься цей елемент шляхом зазначення імені форми ( атрибут NAME тега FORM). Всі форми гіпертексту свого документу організують масив forms, тому ми вправі звернутися до потрібної нам форми за її порядковим номером в цьому масиві (document.forms[0])	Для того, щоб вказати саме з яким елементом форми ми працюємо ми зазначаємо ім'я цього елемента ( атрибут NAME тега, що зазначає відповідний елемент). Всі елементи форми організують масив elements, тому ми вправі звернутися до потрібного нам елемента за його порядковим номером в цьому масиві (document.form1.elements[0]).	До кожного об'єкта браузера використовуються власні методи та властивості саме value - це властивість, котра дорівнює значенню відповідного елемента.

(мал.1)

За допомогою схеми (мал.1) зрозуміло, що значенню поля для виводу інформації (output1) ми присвоїли значення поля для вводу інформації (input1) в момент, коли відбулася подія onclick для кнопки нашої форми. Після того, як ми суттєво розібралися як ми можемо звертатися до об'єктів браузера ми поступово будемо знайомитись з основними методами та властивостями цих об'єктів. Але на даному етапі ми можемо перейти до вивчення основних синтаксичних особливостей мови JavaScript. Як вже було з'ясовано основні події для кожного об'єкта гіпертекстового документа можуть оброблятися за допомогою функції.

Тому цілком зрозуміло те, що ми переходимо до вивчення правил написання основних модулів програми, принципів усталення програм в гіпертекстовий документ, правил обміну фактичних та формальних параметрів, принципів типізації даних, структури основних операцій та операторів, тощо. На цьому етапі пропонується знайомство зі стандартними класами мови JavaScript - Date, String, Array, Math, поглиблюється розуміння поняття інкапсуляції в об'єктно-орієнтованій мові програмування, досконало вивчаються методи та властивості об'єктів стандартних класів. Подальше вивчення мови JavaScript орієнтовано на поглиблення розуміння сутності об'єктно-орієнтованого програмування, розкриття поняття поліморфізму та спадковості. На цьому етапі пропонується скласти власний клас, утворити з нього новий клас на базі батьківського, додати нові методи до системних класів, реалізувати технологію динамічної спадковості.

Подальше вивчення мови JavaScript базується на поєднанні основних знань по програмуванню на мові JavaScript з одного боку з можливостями надання інтерактивності структурним елементам гіпертекстового документу з іншого та приводить до придбання навичок вирішення складних практичних задач. Тут ми поглиблено розглядаємо ієрархічну структуру об'єктів браузера, вивчаємо їх основні методи та властивості, вчимося керувати фреймами та вікнами за допомогою мови JavaScript. Саме в цьому розділі пропонується

завдання, в яких необхідно розробити навчаючі програми, анімаційні ефекти, програми тестування з контролем часу, тощо.

Весь курс по мові JavaScript розраховано на 18 лекційних та 36 лабораторно-практичних годин. Практична частина складається з 7 лабораторних завдань, на які виділяється в залежності від складності від 2 до 6 годин на виконання. Курс апробовано на базі Одеського національного університету ім. І.І. Мечнікова та на базі Південноукраїнського державного педагогічного університету ім. К.Д. Ушинського. Результати підтверджують ефективність розробленої методики, зацікавленість матеріалом з боку студентів та доцільність викладення даного розділу в вищих педагогічних закладах.

#### Література.

1. Основи дистанційного навчання. Дистанційний курс: Навчальний посібник. //За ред. В.М. Кухаренко Харків: ХДПУ, 1999. - 182 с.
2. Дистанційне навчання: Навчальний посібник. //За ред. Є.С. Полат. - М.:Гуманітарне видавництво центру ВАЛДОС, 1998. - 192с.
3. Ю.С. Рамський, І.С. Іваськів. Методика навчання основ WEB-програмування в курсі інформатики загальноосвітньої школи.// "Комп`ютер у школі та сім`ї", №1(13) 2001, с.18-20.
4. І.В. Клименко, Я.М. Степанова, І.О. Сафронова, Т.С. Іванова. Проблеми дистанційної освіти.// "Комп`ютер у школі та сім`ї", №6(18) 2001, с.13-15.
5. М.І. Жалдак. Система підготовки вчителя до використання інформаційних технологій в учбовому процесі. Дис. ... д-ра пед. наук. //М.: НІІ СІМО АПН СРСР, 1989, 48с.
6. Арон И. Волш. Основи програмування на Java для World Wide Web. //Пер. з англ. - Київ, "Діалектика", 1996. 511с.
7. Дарнелл Р. JavaScript: довідник. //пер. з англ. - С. Петербург, "Пітер", 1998, 416с.
8. Менджер Джейсон. JavaScript: основи програмування. // ВНУ Київ, 1999, 512с.