ФАХОВИЙ КОЛЕДЖ ПВНЗ БУКОВИНСЬКИЙ УНІВЕРСИТЕТ
ЦИКЛОВА КОМІСІЯ З ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

МЕТОДИЧНИЙ ПОСІБНИК

до лабораторних занять із дисципліни «Web-технології та Web-дизайн»
для студентів спеціальності 122 комп'ютерні науки
укладач: Артеменко О.І., к.т.н., доцент

СХВАЛЕНО
Протокол засідання
циклової комісії
«___» ____ 20__ р. №___

2020 рік

ЗМІСТ

1 ПОЯСНЮВАЛЬНА ЗАПИСКА

Методичний посібник до виконання лабораторних робіт розроблений відповідно до програми дисципліни «Web-технології та Web-дизайн». Теми лабораторних робіт охоплюють всі розділи предмету.

Метою виконання лабораторних робіт є набуття практичних умінь і навичок web-програмування та web-дизайну. Лабораторні роботи розширюють і поглиблюють теоретичні знання, дозволяють набути досвіду самостійного опрацювання та вирішення конкретних завдань.

Лабораторні роботи з предмету "Web-технології та Web-дизайн" для спеціальності 122 передбачають вивчення мови розмітки HTML, мов програмування JavaScript, PHP, каскадні таблиці стилів CSS, ознайомлення з методами веб-дизайну, веб-програмування, роботою з базами даних MySQL.

Лабораторні роботи тісно пов’язані з теоретичним матеріалом, попередніми та наступними лабораторними роботами. Лабораторні роботи забезпечені навчальними методичними посібниками, інструкціями і завданнями.

В результаті вивчення предмету студенти повинні знати: сучасні мови веб-програмування, основні веб-графіки та веб-дизайну.

Студенти повинні вміти: створювати веб-сторінки, оформлювати та публікувати веб-ресурси, забезпечувати інтерактивну та програмну підтримку веб-сайтів.

Лабораторні роботи виконуються в комп’ютерній лабораторії, проводяться поетапно після вивчення відповідного теоретичного матеріалу. Наведені в кожній лабораторній роботі теоретичні відомості охоплюють необхідний для підготовки і виконання роботи мінімум навчального матеріалу.

До початку кожної лабораторної роботи студенти повинні ознайомитись теоретичним матеріалом по темі лабораторної роботи, з порядком виконання, внести у звіт назву роботи, її мету, обладнання, яке використовується у даній роботі.

Результати студенти обов’язково оформляють у вигляді звіту та пред’являють викладачу. Звіт повинен містити номер лабораторної роботи, тему, завдання до роботи, тексти скриптів, результати їх виконання, висновок.

Захист робіт проходить шляхом перевірки правильності виконання завдань, звіту, усного опитування. Робота вважається зарахованою при наявності повністю оформленого звіту і позитивної оцінки при захисті.

2 ІНСТРУКЦІЯ З ТЕХНІКИ БЕЗПЕКИ

1 Загальні положення

Приступаючи до роботи з ПК, необхідно завжди пам’ятати, що це дуже складна і дорога апаратура, яка потребує обережного і акуратного ставлення до неї, високої самодисципліни на всіх етапах роботи з комп’ютером.

Напруга живлення ПК (220 В) є небезпечною для життя людини. Через це в конструкції блоків комп’ютерів, міжблочних з’єднувальних кабелів передбачена достатньо надійна ізоляція від струмопровідних ділянок. Користувач практично має справу лише з декількома вимикачами живлення і, здавалося б, застрахований від ураження електричним струмом. Однак в практичній роботі можуть зустрічатися непередбачені ситуації, і щоб вони не стали небезпечними для користувача, необхідно знати та чітко виконувати ряд правил техніки безпеки. Це допоможе не тільки уникнути нещасних випадків і зберегти здоров’я, але й гарантує збереження апаратури.

Особливо уважним треба бути при роботі з дисплеєм, електронно-променева трубка якого використовує високу напругу і є джерелом електромагнітного випромінювання. Неправильне поводження з дисплеєм та іншою електронною апаратурою може привести до тяжких уражень струмом, спричинити загоряння апаратури.

2 Вимоги безпеки під час виконання роботи

2.1 Перевірити робоче місце, щоб на ньому не було зайвих предметів

2.2 Під час роботи НЕОБХІДНО:

- суворо дотримуватись положень інструкції з експлуатації апаратури
- працювати на клавіатурі чистими сухими руками, не прикладати надмірних зусиль при роботі з клавіатурою та мишею
- працюючи з дискетами, оберігати їх від механічних пошкоджень, пливу магнітного поля та тепла, при вставленні дискети в дисковод переконатися в її правильній орієнтації
- електронно-променева трубка є джерелом електромагнітного випромінювання, тому працювати треба на відстані не менше 60 см від екрану, дотримуватись правильної постави, не нахиляючись і не сутулячись
- тривала робота на комп’ютері призводить до напруження зору, тому час від часу треба робити перерви в роботі

2.3 Під час роботи ЗАБОРОНЯЄТЬСЯ:

- знаходитись в лабораторії в верхньому одязі;
- торкатися до екрана і тильного боку дисплея, проводів живлення, з’єднувальних кабелів і пристроїв заземлення
- пересувати системний блок та монітор;
- порушувати порядок ввімкнення і вимкнення апаратури
- намагатися самостійно усунути виявлену несправність
- класти на апаратуру сторонні предмети
- працювати на комп’ютері з вологими руками, у вологій одежі
- працювати при недостатньому освітленні, високому рівні шуму
- не дозволяються сторонні розмови, подразнюючі шуми

2.4 Після закінчення роботи необхідно:

- зберегти робочу інформацію на носіях
- вимкнути комп’ютер в правильній послідовності
- прибрати робоче місце.

ПЕРЕЛІК ПОСИЛАНЬ

1 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 / Р. Никсон. – 3-е изд. – СПб. : Питер, 2015. – 688 с.
2 Макфарланд Д. Большая книга CSS3 / Д. Макфарланд. – 3-е изд. – СПб. : Питер, 2014. — 608 с.
3 Берд Д. Веб-дизайн. Руководство разработчика / Д. Берд. – СПб. : Питер, 2012. — 224 с.
4 Макнейл П. Веб-дизайн. Идеи, секреты, советы / П. Макнейл. – СПб. : Питер, 2012. — 272 с.
5 Сырых Ю.А. Современный веб-дизайн. Эпоха Веб 3.0 / Ю.А. Сырых. – 2-е изд. – М. : ООО «И.Д. Вильямс», 2013. – 368 с.
6 Adobe Systems Incorporated. Использование Adobe Dreamweaver CS5 и CS5.5. Справочное руководство
7 Фрейн Б. HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Б. Фрейн. – СПб. : Питер, 2014. – 304 с.
8 Хоган Б. HTML5 и CSS3. Веб-разработка по стандартам нового поколения / Б. Хоган. – 2-е изд. – СПб. : Питер, 2014. – 320 с.
9 Робсон Э. Изучаем HTML, XHTML и CSS / Э. Робсон, Э. Фримен. – 2-е изд. – СПб. : Питер, 2014. – 720 с.
10 Шмитт К. HTML5. Рецепты программирования / К. Шмитт, К. Симпсон. – СПб. : Питер, 2012. – 288 с.
11 Седерхольм Д. Пуленепробиваемый веб-дизайн. Библиотека специалиста / Д. Седерхольм. – 3-е изд. – СПб. : Питер, 2012. – 304 с.
12 Чекко Р. Графика на JavaScript / Р. Чекко. – СПб. : Питер, 2013. — 272 с.
13 Флэнаган Д. JavaScript. Подробное руководство, 6-е издание / Д. Флэнаган. – СПб. : Символ-плюс, 2012. – 1080 с.
14 Маклафлин Б. PHP и MySQL. Исчерпывающее руководство / Б. Маклафлин. – СПб. : Питер, 2013. – 512 с.
15 Роббинс Д. HTML5, CSS3 и JavaScript. Исчерпывающее руководство / Д. Роббинс; [пер. с англ. М. А. Райтман]. – 4-е издание. – М. : Эксмо, 2014. – 528 с.

ЛАБОРАТОРНЕ ЗАНЯТТЯ №1 (2 год.)

ТЕМА. Створення html-документів з таблицями

НАВЧАЛЬНА МЕТА: навчитися створювати таблиці заданого розміру; навчитися об'єднувати ячейки по горизонталі і вертикалі; навчитися створювати фіксовані таблиці.

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Таблиці використовуються для представлення табличних, статистичних даних, створення календарів, розкладів тощо. Раніше (до CSS) використовувались для створення багатоколоночних макетів. Складності при роботі з таблицями виникають на пристроях з невеликими екранами. Зараз дизайнери працюють над цією проблемою.

Теги:

<table></table> – вміст таблиці

<tr></tr> – рядок таблиці

<th></th> – заголовок таблиці

<td></td> – ячейка даних таблиці

Приклад html-коду таблиці:

```
<table>
  <tr><th>Страви</th><th>Калорії</th><th>Жири (г)</th></tr>
  <tr><td>Курячий бульйон</td><td>120</td><td>2</td></tr>
  <tr><td>Салат Цезар</td><td>400</td><td>26</td></tr>
</table>
```

Об’єднання ячейок

Об’єднання стовпців – атрибут colspan.

Приклад.

```
<table>
  <tr><th colspan="2">Жири</th></tr>
  <tr><td>Насичені жири</td><td>Ненасичені жири</td></tr>
</table>
```

Об’єднання рядків – атрибут rowspan.

Приклад.

```
<table>
 <tr><th rowspan="3">Порції</th><td>Маленька (150 г.)</td></tr>
 <tr><td>Середня (250 г.)</td></tr>
 <tr><td>Велика (2400 г.)</td></tr>
</table>
```

Доступність таблиць

1 Опис вмісту таблиці – <caption>.

Приклад.

```
<table>
  <caption>Харчова цінність</caption>
  <tr><th>Страви</th><th>Калорії</th><th>Жири (г)</th></tr>
  ...
</table>
```

2 Додавання опису структури таблиці. Для цього в HTML5 використовуються розширені

елементи таблиці.

Розширені елементи таблиці

Елементи групи рядків – *thead, tfoot, tbody.* Рядки та групи рядків можна охарактеризувати як ячейки, що відносяться до заголовка, нижнього колонтитулу чи основної чатини таблиці.

Елементи групи стовпців – *col (визначення стовпців), colgroup (групування стовпців).*

Приклад.

```
<table>
  <caption>Харчова цінність</caption>
  <col span="1" class="itemname">
  <colgroup id="data">
        <col span="1" class="calories">
        <col span="1" class="fat">
  </colgroup>
  <thead>
        <tr><th>Страви</th><th>Калорії</th><th>Жири (г)</th></tr>
  </thead>
  <tbody>
        <tr><td>Курячий бульйон</td><td>120</td><td>2</td></tr>
        <tr><td>Салат Цезар</td><td>400</td><td>26</td></tr>
  </tbody>
</table>
```

Спеціальні можливості – *scope, headers.* Інколи складно зрозуміти, який заголовок до яких ячейок відноситься. Атрибут *scope* співставляє заголовок таблиці з рядком, стовпцем, групою рядків (напр., *tbody*) чи групою стовпців, в яких він прописаний, використовуючи значення *row, column, rowgroup* чи *colgroup* відповідно.

Приклад. Ячейки зоголовка відносяться до поточного рядка.

```
<table>
 <tr><th scope="row">Марс</th><td>.95</td><td>.62</td><td>0</td></tr>
</table>
```

У випадку дійсно складних таблиць, де атрибуту *scope* недостатньо, щоб однозначно співставити ячейку із заголовком, в елементі *td* вказують атрибут *headers*.

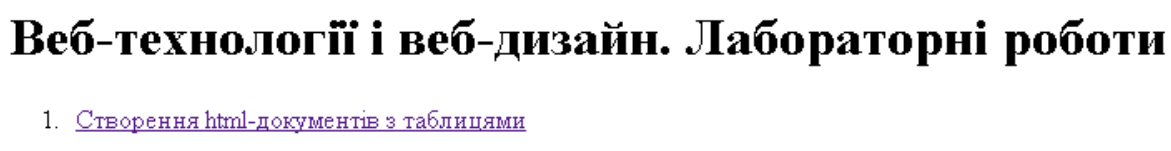
Приклад.

```
<th id="diameter">Діаметр вимірюється в кілометрах</th>
...багато інших ячейок...
<td headers="diameter">.38</td>
... багато інших ячейок...
```

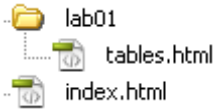
ХІД РОБОТИ

Зауваження. Підчас виконання завдань дотримуватись принципів розмітки HTML5.

1 Створити індексний файл index.html, який буде містити посилання на завдання лабораторних робіт:



2 Створити папку lab01, в якій створити файл tables.html:



3 У файлі tables.html у блок заголовку включити наступний код:

```
<style>
  td, th {
```

```
        border: 1px solid #666;
      }
</style>
```

4 У верхньому колонтитулі tables.html розмістити посилання на головну сторінку, вказати тему та мету роботи:

На головну

Лабораторна робота №1

ТЕМА. Створення html-документів з таблицями
НАВЧАЛЬНА МЕТА: навчитися створювати таблиці заданого розміру; навчитися об'єднувати ячейки по горизонталі і вертикалі; навчитися створювати фіксовані таблиці.

5 Реалізувати таблицю:

Таблиця 1

Альбом	Рік
Transmissions	1993
Luciana	1994
Beyond the Infinite	1995
Bible of Dreams	1997
Shango	2000
Labyrinth	2004
Gods & Monsters	2008
Inside The Reactor	2011
The Golden Sun Of The Great East	2013

Виділити в даній таблиці заголовок, основну частину, нижній колонтитул.

6 Реалізувати таблицю:

Таблиця 2

7:00	7:30	8:00
Lorem ipsum		
Lorem ipsum	Lorem ipsum	Lorem ipsum
Lorem ipsum	Lorem ipsum Lorem ipsum	

Для об’єднання ячейок використати атрибут colspan.

7 Реалізувати таблицю:

Таблиця 3

<table>
<tr><td>яблука</td><td rowspan="3">апельсини</td><td>персики</td></tr>
<tr><td>банани</td><td rowspan="2">ананаси</td></tr>
<tr><td>авокадо</td></tr>
</table>

Для об’єднання ячейок використати атрибут rowspan.

8 Реалізувати таблицю:

Таблиця 4

Заголовок таблиці

	Загальний заголовок для двох підзаголовків		Заголовок 3
	Заголовок 1	Заголовок 2	
Пункт А	данні А1	данні А2	данні А3
Пункт Б	данні Б1	данні Б2	данні Б3
Пункт В	данні В1	данні В2	данні В3

Співставити заголовки Пункт А, Пункт Б, Пункт В відповідним рядкам.

9 У нижньому колонтитулі стрінки вказати власне прізвище та групу.

10 Показати виконану роботу викладачу

11 Оформити звіт. Захистити роботу.

ЗМІСТ ЗВІТУ

1 Тема
2 Мета
3 Код
4 Схематичне зображення на екрані
5 Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ

1 Який тег служить для вставки таблиці в html -документ?
2 Який тег відповідає за початок рядка?
3 Який тег відповідає за початок ячейки?
4 Які теги використовуються для об’єднання ячейок?
5 Як можна об'єднати ячейки по горизонталі?
6 Як можна в таблиці об'єднати ячейки по вертикалі?
7 Який тег використовується для опису вмісту таблиці?
8 Який тег використовується для виділення заголовку таблиці?
9 Який тег використовується для виділення основної частини таблиці?
10 Який тег використовується для нижнього колонтитулу таблиці?
11 Які атрибути використовуються для співставлення заголовків рядкам, стовпцям, ячейкам?

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.

ЛАБОРАТОРНЕ ЗАНЯТТЯ №2 (2 год.)

ТЕМА. Створення форм за допомогою HTML

НАВЧАЛЬНА МЕТА: навчитися створювати форми і вказувати обробників форм; Навчитися створювати елементи управління форми.

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Форми призначені для організації взаємодії з користувачем. Вони дозволяють вводити текст, здійснювати вибір із запропонованих значень за допомогою списків або кнопок, організовувати інтерактивний обмін інформацією між Web-сторінкою і сервером.

Типова функціонуюча форма складається із двох частин:

1 створена за допомогою розмітки (відображається на екрані);

2 дотаток чи сценарій на стороні сервера, який обробляє введені користувачем дані і повертає відповідний результат.

Робота форми:

- відкриття та заповнення форми;
- відправка даних форми;
- збирання, перетворення, відправка інформації на сервер браузером для подальшої обробки;
- прийняття відправленої інформації додатком веб-серверу, її обробка;
- надсилання додатком відповіді браузеру. Тип відповіді залежить від вмісту та призначення форми.

Дані форми перетворюються браузером так як і URL, де пробіли та інші недопустимі символи представляються в шіснадцятковій формі. Напрклад, символ пробілу після збору даних форми представлений у вигляді значення %20, а символ слеша – %2F.

Інтеракетивна форма:

<form>…</form>

Являє собою контейнер для всього вмісту форми. Не може містити інший елемент <form>.

Зараз прийнято виділяти компоненти форми семантичними HTML-елементами, такими як списки чи .

Приклад.

```
<h1>Підписка на розсилку</h1>
<form action="/cgi-bin/mailinglist.php" method="post">
<fieldset>
<legend>Підписка на подкаст — кращий спосіб залишатись в курсі всіх цікавих новин проекту.</legend>
<ol>
<li><label for="name">Ім'я:</label><input type="text" name="name" id="name"></li>
<li><label for="email">Email:</label><input type="text" name="email" id="email"></li>
</ol>
<input type="submit" value="Підписатись">
</fieldset>
</form>
```

В атрибуті *action* вказується місцезнаходження (URL-адреса) додатку чи сценарію для обробки форми. Інколи код обробки даних форми знаходиться в поточному документі. В такому випадку атрибут *action* залишають пустим.

Атрибут *method* визначає спосіб передачі інформації на сервер. Наприклад, дані

name = Anna Sokolova

email = annsokol@example.com

після перетворення браузером матимуть вигляд

username=Anna%20Sokolova&email=annsokol%40example.com

Існує два способи їх передачі на сервер, які і вказуються в якості значення атрибуту *method*: POST і GET.

При виборі методу POST браузер посилає на сервер окремий запит, який складається із декількох спеціальних заголовків, після яких ідуть дані. Вміст цього запиту доступний лише серверу, в зв'язку із чим цей метод є оптимальним для передачі конфіденційних даних. Також він використовується при передачі великого об'єму даних, так як не має обмеження в кількості символів на відміну від метода GET.

При виборі методу GET перетворені дані форми вбудовуються в URL-запит, що відправляється на сервер. Адреса відокремлюється від даних знаком запитання:

get http://www.bandname.com/cgi-bin/mailinglist.php?username=Anna%20Sokolova&email=annsokol%40example.com

Цей метод підходить, якщо користувачам потрібно надати можливість зберігати в закладках результати відправки даних форми. Метод GET непридатний для завантаження файлів на сервер.

Атрибут *name* визначає ім'я змінної елемента форми.

Всі елементи форми повинні містити в собі атрибути *name,* щоб зв'язаний з нею додаток міг розподілити дані за типами. Атрибут name допустимо додавати і до елементів кнопок *submit* і *reset*, але це не обов'язково, так як у них особливі функції (відправка даних та очистка полів форми), не пов'язані зі збиранням даних.

При виборі імен елементів форми потрібно керуватися правилами іменування змінних. Імена повинні бути описовими. Не можна задавити імена як завгодно.

Елементи форми

Поля ведення тексту

<input type="text"> – однорядкове текстове поле

<textarea>…<textarea> – багаторядкове текстове поле

<input type="password"> – поле введення паролю

<input type="search"> – поле пошуку

<input type="email"> – адреса електронної пошти

<input type="tel"> – номер телефону

<input type="url"> – розміщення (url-адреса)

Кнопки скидання та відправки даних

<input type="submit"> – відправка даних форми на сервер

<input type="reset"> – скидання значень елементів форми

Приклад.

```
<p><input type="submit" value="Відправити"><input type="reset" value="Почати заново"></p>
```

Інші типи кнопок:

<input type="image"> – кнопка із зображенням

<input type="button"> – кнопка довільного значення

Елемент *<button>…</button>* інструмент для створення кнопки користувача.

Перемикачі та прапорці

<input type="radio"> – перемикач

<input type="checkbox"> – прапорець

Списки

Випадаючі та прокручувані списки створюються за допомогою елемента <select>.

<select>…</select> – список

<option>…</option> – пункт списку

<optgroup>…</optgroup> – логічна група пунктів списку

Елемент <select> Відображається як *випадаючий список*, якщо не вказаний атрибут size, або йому присвоєно значення 1. В іншому випадку буде створено *прокручуваний список*.

Елемент вибору файла

<input type="file"> – поле вибору файла

Приклад.

```
<form action="/client.php" method="POST" enctype="multipart/form-data">
```

```
<label>Завантажте фото<em>(необов'язково)</em><br>
<input type="file" name="photo" size="28"></label>
</form>
```

Важливо відмітити, що для форми, яка містить в собі елемент вибору файла, в якості типу кодування (*enctype*) слід вказувати значення *multipart/form-data* і відправляти дані методом POST.

Елементи дати й часу (HTML5)

<input type="date"> – вибір дати

<input type="time"> – вибір часу

<input type="datetime"> – вибір дати/часу з врахуванням часового поясу (не підтримується в жодному із браузерів)

<input type="datetime-local"> – вибір дати/часу без врахуванням часового поясу

<input type="month"> – вибір місяця року

<input type="week"> – вибір тижня

Введення чисел (HTML5)

<input type="number"> – введення чисел

<input type="range"> – повзунковий регулятор

Вибір кольору (HTML5)

<input type="color"> – палітра кольорів

Приклад.

```
<p><label>Ваш улюблений колір: <input type="color" name="color_"></label></p>
```

ХІД РОБОТИ

1 Створити форму form.html для заповнення анкети (див. зразок нижче). Збережіть файл у попередньо створеній папці lab02. Зв’язати сторінки index.html та form.html посиланнями.

2 Створити поля для введення ПІП користувача і пароля користувача (символи вводяться в поле Пароль повинні відображуватися зірочками):

Прізвище:*

Ім'я:*

По батькові:*

Пароль:* •••••••

3 Створити перемикачі для вибору статі. Реалізувати можливість вибору тільки одного варіанту. Один із варіантів повинен бути вибраний за замовчуванням:

Стать: Ч Ж

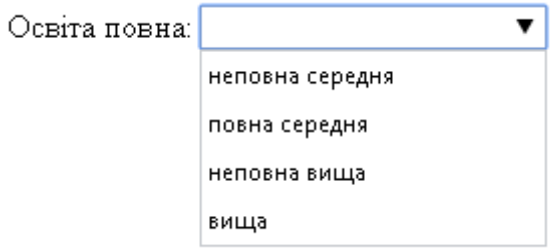
4 Створити поле для введення дати народження:

Дата народження:* дд.мм.гггг

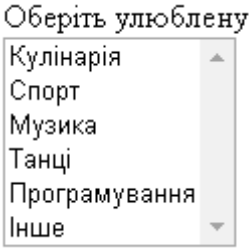
Сентябрь 2015

Пн	Вт	Ср	Чт	Пт	Сб	Вс
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4

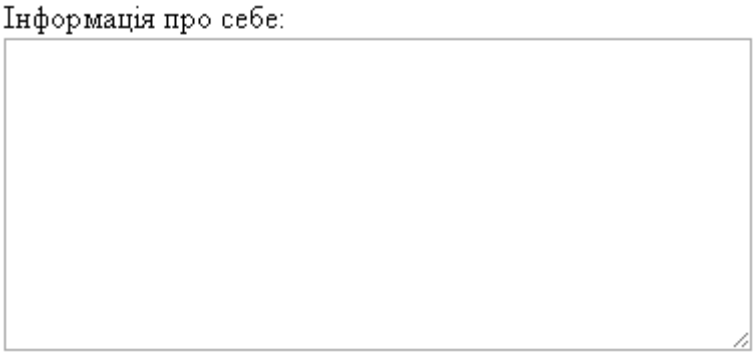
5 Створити поле для вибору освіти:



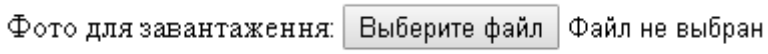
6 Створити список інтересів з можливістю вибору декількох елементів:



7 Додати поле для введення декількох рядків тексту з додатковою інформацією про користувача. Поле повинне мати розміри 10 рядків і 50 стовпців:



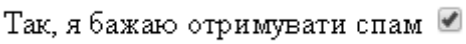
8 Створити поле з кнопкою для завантаження файлу зображення:



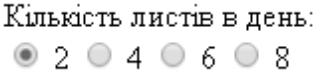
9 Створити поле для введення електронної адреси:

10 Створити поле для введення телефону:

11 Створити прапорець, з текстом "Так, я бажаю отримувати розсилку" з прапорцем, встановленим за умовчанням.



12 Створити перемикачі для вибору кількості отримуваних листів:



13 Створити дві кнопки. Першу для передачі даних форми на сервер, а другу для очищення форми.

14 Виділити на формі дві групи елементів: «Персональна інформація», «Контактна інформація» (див. зразок нижче).

15 Оформити звіт. Показати виконану роботу викладачу.

ЗМІСТ ЗВІТУ

1 Тема
2 Мета
3 Код
4 Схематичне зображення на екрані
5 Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ

1 Охарактеризувати принцип роботи форми.
2 Який синтаксис простої форми?
3 Охарактеризувати основні елементи форми.
4 Які є поля введення тексту?
5 Як реалізувати кнопки відправки та скидання даних форми?
6 Як створити групу перемикачів та прапорців?
7 Як створити список?
8 Як реалізувати вибір файлу у формі?
9 Охарактеризувати елементи дати й часу.
10 Які елементи призначені для забезпечення доступності форм?
11 Який атрибут тега <form> вказує на файл, який оброблятиме ці форми?
12 У чому відмінності тега <input type="reset"> і <input type="submit">?
13 Як додати на форму перемикач?
14 Для чого призначений тег <textarea>?
15 Які існують методи передачі даних форми на сервер?
16 За рахунок чого перемикачі можна розбивати на групи?
17 Який атрибут тега <input> дозволяє вказати значення, яке буде заповнено при першому відображенні форми?
18 Для чого призначений атрибут checked?

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.

Зразок form.html

На головну

Лабораторна робота №2

ТЕМА. Створення форм за допомогою HTML
НАВЧАЛЬНА МЕТА: навчитися створювати форми і вказувати обробників форм; навчитися створювати елементи керування форми.

Форма реєстрації на сайті

Персональна інформація

Прізвище:*

Ім'я:*

По батькові:*

Пароль:*

Стать: Ч Ж

Дата народження:* дд.мм.гггг

Освіта повна:

Оберіть улюблену категорію:
Кулінарія
Спорт
Музика
Танці
Програмування
Інше

Інформація про себе:

Фото для завантаження: Выберите файл Файл не выбран

Контактна інформація

Адреса електронної пошти:*
mailto@mailto.ua
Телефон:

Так, я бажаю отримувати розсилку

Кількість листів в день:
2 4 6 8

Очистити Відправити

ЛАБОРАТОРНЕ ЗАНЯТТЯ №3 (2 год.)

ТЕМА. Створення веб-сторінок з використанням CSS

НАВЧАЛЬНА МЕТА: закріпити навички роботи із блочними, плаваючими елементами, використання класів, застосування позиціонування.

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

В CSS можливе використання так званих псевдокласів – набору класів, що описують деякі елементи, які не можна описати звичайним способом.

селектор:псевдоклас { властивість: значення; ... }

При описі назва псевдокласу на відміну від звичного класу користувача відокремлена від назви тега двокрапкою.

:first-child - перший дочірній елемент даного елементу
:link - невідвідані посилання
:visited - відвідані посилання
:hover - елемент, над яким знаходиться курсор
:active - активний елемент
:focus - елемент, що має фокус введення
:lang - поточна мова
:first-line - перший рядок
:first-letter - перша буква
:before - вміст перед елементом
:after - вміст після елементу

Для спрощення роботи із стилями і зменшення текстового навантаження опису введені деякі додаткові можливості:

1) загальний стиль

* { властивість: значення; ... }

Такий знак означає, що всі теги в HTML-документі підкорятимуться правилам даного стилю

2) дочірній елемент

селектор>дочірній_селектор { властивість: значення; ... }

Такий знак означає, що правила стилю розповсюджуються на дочірній тег тільки вказаного тега, наприклад:

ol>li {font-size:13pt} - правила діють тільки на елементи нумерованого списку

3) вільний клас

#клас { властивість: значення; ... }

Такий знак означає, що правилам даного стилю підкорятимуться ті теги в HTML-документі, які будуть підключені до вказаного класу за допомогою спеціального атрибуту id:

id="ім’я класу"

4) групування селекторів

селектор1, селектор2, … { властивість: значення; ... }

При використовуванні однакових оголошень для різних тегов можливе їх угрупування – перелік через кому.

Наприклад:

h1, p, div { font-size: 13 pt; }

замість

h1 { font-size: 13 pt; }

p { font-size: 13 pt; }

div { font-size: 13 pt; }

5) групування значень

селектор { групуюча_властивість: значение1 значение2 ...; }

Деякі однотипні (ті, що характеризують однотипні параметри об'єкту) властивості можуть групуватися в спеціальні групуючі властивості, для яких значення перераховуються після двокрапки через пропуски. При цьому дотримується певна послідовність вказівки значень.

Наприклад:

h1 {font: bold normal 13pt/14pt arial }

замість

h1 {
font-family: arial;
font-size: 13 pt;
line-height: 14 pt;
font-weight: bold;
font-style: normal;
}

ХІД РОБОТИ

Примітка. За можливості потрібно використовувати групування селекторів та значень CSS.

1 Створіть html-файл traffic_light.html, що містить блочний елемент DIV.

```
<div>Це блочний елемент. Це блочний елемент.</div>
```

2 Створіть css-файл і підключіть його до html-документу.

3 Створіть клас k, в якому визначите розмір блоку 200x200 пікселів, фон (ясно-зелений) і рамку (суцільну, темно-зелену, шириною в 3 пікселі).

```
.k{
    width: 200px;
    height: 200px;
    background-color: #99ff99;
    border-style: solid;
    border-color: #009933;
}
```

4 Підключіть клас k до блоку в html-документі. Зверніть увагу на те, як зміниться блок.

5 Додайте ще п'ять таких же блоків розміром 200x200 пікселів. Блоки розташовуються один під одним.

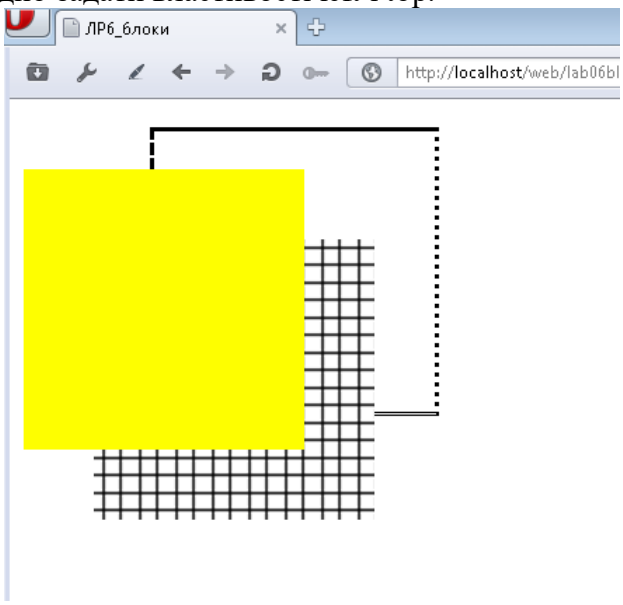
6 Для того щоб блоки розташовувалися один за одним необхідно додати в клас k властивість *float: left;*. Відновіть і подивіться зміни при зміні розмірів вікна.

7 Тепер блоки "липнуть" один до одного, між ними немає відстані. Щоб додати зовнішній відступ ліворуч додайте властивість *margin: 3px;*.

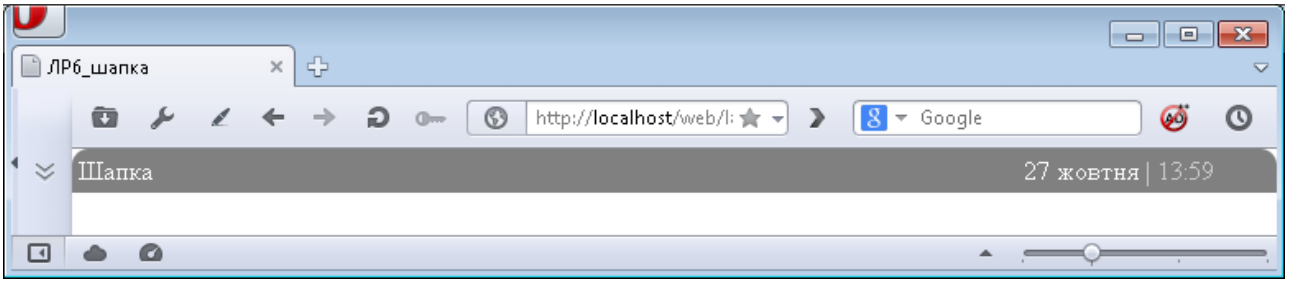
8 Використовуючи вільні класи модифікуйте попередній файл, створивши горизонтальний світлофор з трьох блоків. Ширину блоків зробіть динамічною, щоб вони змінювалася залежно від ширини вікна, але усі блоки повинні мати одну ширину. Задайте кожному блоку свій колір. Відступи для body приберіть за допомогою css.

9 Задайте першому блоку властивість *display: none;*, а другому *visibility: hidden;* У чому відмінності цих двох властивостей?

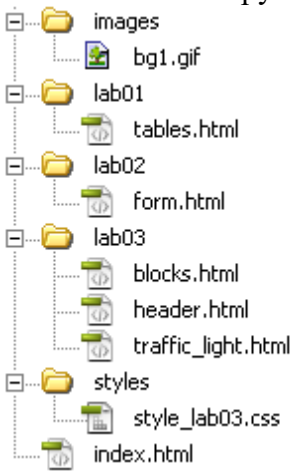
10 Створіть файл blocks.html, а в ньому три блоки (200x200 пікселів). У першому блоці зробіть рамку шириною 3px з різним типом лінії з різних сторін блоку. У другому блоці задайте фонове зображення (*background-image*). Файл для фонового зображення bg1.gif знаходиться у папці з інструкцією до лабораторної роботи. У третьому блоці, задайте фон. Розташуйте блоки один над одним. Для цього встановіть кожному блоку властивість *position: absolute;* і порядок розташування блоків зверху вниз: оголошення z-index: 5; (у тих блоків що нижче, значення індексу має бути менше, наприклад 10, 20 і 30). Що б блоки починалися не в одній точці, їм необхідно задати властивості left і top.



11 Використовуючи блоки, їх вкладеність, фон, вирівнювання, властивість *float*, властивість *margin-right*, створіть «гумову» шапку в окремому файлі header.html. Для обмеження стиснення та розтягування використайте властивості *min-width* та *max-width*. Для заокруглення використовуйте властивість *border-radius*. Для вирівнювання по центру вікна браузера - *margin: 0 auto;*.



12 Забезпечити структуру папок та файлів відповідно до поданого нижче зразка:



13 Додати посилання на створені сторінки до індексного файлу:

Веб-технології і веб-дизайн. Лабораторні роботи

1. Створення html-документів з таблицями
2. Створення форм за допомогою HTML
3. Створення веб-сторінок з використанням CSS
    1. Блоки
    2. Світлофор
    3. Шапка

14 Показати роботу викладачу. Оформити звіт. Захистити роботу.

ЗМІСТ ЗВІТУ

1 Тема
2 Мета
3 Код
4 Схематичне зображення на екрані
5 Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ

1 Як описати псевлоклас?
2 Яке призначення властивості float?
3 Що таке загальний стиль?
4 Що таке дочірній елемент? Як його описати в CSS?
5 Що називають вільним класом або ідентифікатором?

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.

ЛАБОРАТОРНА РОБОТА №4 (2 год.)

ТЕМА. Створення макетів сайту

НАВЧАЛЬНА МЕТА: навчитися створювати фіксовані та гумові макети сайту у дві колонки.

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Технологія, що дозволяє створити один сайт, що адаптується до пристроїв з різною шириною екрану отримала назву «адаптивний веб-дизайн (Responsive web design, RWD, термін введений Ітаном Маркоттом)», використовує різні прийоми, що змушують сторінку змінювати розмітку на основі ширини екрана браузера. Наприклад, на смартфоні сторінку можна викласти однією, легкою для читання колонкою, яка поміститься на вузькому екрані, а на більш широких моніторах підтримувати розмітку в кілька колонок.

RWD не є єдиною технологією і методом. У ньому зібрані воєдино кілька методів CSS і HTML для створення веб-сторінок, чия розмітка адаптується до різних екранів.

У RWD об'єднано три основні ідеї: гнучкі сітки (grid) для розмітки, гнучке середовище для зображень, а також відео- та медіазапити CSS, призначені для створення різних стилів для екранів різної ширини.

Оскільки виробники телефонів розуміють, що більшість веб-сайтів створені для екранів настільних комп'ютерів, вони змусили свої браузери зменшувати масштаб сайтів. Конкретний коефіцієнт зменшення варіюється залежно від характеристики конкретного телефону. Наприклад, Safari на iPhone працює так, ніби екран дійсно має ширину 980 пікселів, і зменшує сторінку, щоб вона помістилася в цих 980 пікселах.

Подібна поведінка мобільних браузерів дозволяє непогано справлятися з більшістю сайтів, але з адаптивним веб-дизайном вона поєднується не дуже добре. Оскільки адаптивні сайти призначені для отримання хорошого подання на смартфонах, зменшувати масштаб зображення на дисплеї не потрібно. Є досить простий спосіб скасування такої поведінки в браузерах мобільних пристроїв. Потрібно просто до розділу <head> веб-сторінки додати наступний код (саме підходяще місце для цього - безпосередньо перед тегом <title>):

<meta name = "viewport" content = "width = device-width">

Крім використання метатега viewport, є ще один спосіб. Розробники з CSS Working Group додали до CSS правило viewport, яке дозволяє робити все те ж саме, що і в метатегах viewport, але у таблиці стилів. Завдяки цьому можна відмовитися від додавання тега <meta> до кожного HTML-файлу сайту і просто додати одне правило viewport до своєї таблиці стилів:

@viewport {width: device-width; }

Це правило потрібно додати в самому початку таблиці стилів до оголошення самих стилів. На жаль, в даний час правило viewport ще не працює на всіх браузерах і вимагає для тих браузерів, які його не розуміють, додавати префікс виробника.

У CSS 3 введено таке поняття, як *медіазапити*. Вони дозволяють призначати стилі сторінкам на основі ширини і висоти вікна цільового браузера.

Використовуючи даний метод, можна створювати стилі користувача для браузерів мобільних телефонів, планшетних і настільних комп'ютерів і тим самим налаштовувати представлення сайту таким чином, щоб він виглядав найкращим чином на кожного типу пристрою.

Створення медіазапитів

У зазначеному нижче прикладі браузер завантажує зовнішню таблицю стилів *small.css*, коли хто-небудь переглядає ваш сайт за допомогою браузера, ширина вікна якого становить 480 пікселів. Дужки навколо запиту - *(width: 480px)* - є обов'язковим елементом. Якщо їх не поставити, браузер проігнорує запит.

<link href="css/small.css" rel="stylesheet" media="(width: 480px)">

Медіазапити зрозумілі більшості браузерів мобільних і настільних пристроїв, але незрозумілі Internet Explorer 8 і більш раннім версіями. Застарілі версії Internet Explorer можна змусити розуміти ваші медіазапити, якщо додати до тега <head> документа трохи коду JavaScript. Потрібно завантажити файл *respond.js* з адреси http://tinyurl.com/7w49a6z, помістити цей файл у сайт, а потім дати на нього посилання на сторінці, скориставшись тегом <script>. наприклад:

<!--[if lte IE 8]>

<script src = "respond.min.js"> </script>

<![endif]-->

Цей невеликий маневр змусить Internet Explorer 8, 7 і 6 розуміти медіазапити.

Для наближеного значення роздільної здатності:

<link href="css/small.css" rel="stylesheet" media="(max-width:480px)">

Для настільних ПК:

<link href="css/large.css" rel="stylesheet" media="(min-width:769px)">

Для планшетних ПК:

<link href="css/medium.css" rel="stylesheet" media="(min-width:481px) and (max-width:768px)">

Включення запитів в таблицю стилів

Використання директиви @import.

@import url(css/small.css) (max-width: 320px);

На жаль, як уже раніше зазначалося, Internet Explorer 8 і більш ранні версії розуміють медіазапити тільки при використанні програми JavaScript під назвою respond.js. Ця програма не працює з медіазапитами, що застосовують технологію @import, тому при використанні цієї директиви і медіазапита для завантаження таблиці стилів, призначеного для браузерів настільних систем, Internet Explorer 8 і більш ранні версії не завантажать цю таблицю стилів. Але ця проблема так гостро не стоїть, оскільки немає ніякої необхідності задіяти медіазапити для браузерів настільних систем.

Директиви *@import* повинні поміщатися в початок таблиці стилів. Вони не можуть йти після будь-яких стилів. В результаті можуть виникати проблеми з каскадністю, при яких стилі, визначені у зовнішній таблиці стилів і завантажені з допомогою директиви *@import*, будуть скасовуватися більш пізніми стилями в таблиці стилів. Цієї проблеми можна уникнути при наявності всього лише однієї зовнішньої таблиці стилів, яка містить тільки директиви *@import*. Перша з них призведе до завантаження основної таблиці стилів, призначеної для всіх пристроїв, а друга і третя приведуть до завантаження таблиць стилів з використанням медіазапитів:

@import url(css/base.css); /* Немає медіазапита, застосовувати до всіх */

@import url(css/medium.css) (min-width: 481px) and (max-width: 768);

@import url(css/small.css) (max-width: 480px);

Вбудовування медіа запиту в таблицю стилів. Медіазапити можна також вбудувати безпосередньо в таблицю стилів:

```
@media (max-width: 480px) {
body {
/* Сюди поміщаються властивості стилю */
  }
.style1 {
/* Сюди поміщаються властивості стилю */
  }
}
```

Директива *@media* працює як своєрідний контейнер для всіх стилів, що відповідають запиту.

Основна структура таблиці стилів

Можна сказати, що існує безліч різних способів використання медіазапитівв, які націлені на різні пристрої: з пріоритетом, що віддається настільним системам, з пріоритетом, що віддається мобільним системам, в окремих таблицях стилів, в єдиній таблиці стилів і т. д. Але спочатку потрібно завжди створювати єдину зовнішню таблицю стилів, включаючи в неї стилі

для дисплеїв настільних систем, а потім додаючи медіазапити зі змінами цього дизайну основного рівня під планшетні пристрої і телефони. Схематично структура для такого файлу повинна мати наступний вигляд:

```
/* Сюди поміщаються стилі, що перезапускають вихідні настройки браузера */
/* Сюди поміщаються стилі для браузерів настільних пристроїв і основні стилі для всіх пристроїв */
body {
/* Властивості для тіла документа */
}
/* Тільки для дисплеїв середньої ширини */
media (min-width: 481px) and (max-width: 768px) {
body {
/* Властивості, застосовувані тільки до браузерів планшетних пристроїв */
}
}
/* Тільки для дисплеїв малої ширини */
media (max-width: 480px) {
body {
/* Властивості, застосовувані тільки до браузерів телефонів */
}
}
```

ХІД РОБОТИ

1Створити фіксований макет веб-сторінки у дві колонки за зразком.

fixed.html:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8" />
<title>Фіксований макет у дві колонки</title>
<link href="../styles/fixed.css" rel="stylesheet" />
</head>
<body>
  <div class="layout">
    <div class="header">
      <h1>Hreader</h1>
    </div>
    <div class="sidebar">
      <ul>
        <li><a href="#">one</a></li>
        <li><a href="#">two</a></li>
        <li><a href="#">three</a></li>
        <li><a href="#">four</a></li>
        <li><a href="#">five</a></li>
        <li><a href="#">six</a></li>
      </ul>
    </div>
    <div class="content">
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec condimentum dolor id purus imperdiet gravida. Etiam rhoncus molestie lorem tristique dictum. Proin efficitur vehicula ipsum eget tristique. Phasellus tempor lacinia turpis, in congue sem dignissim porta. Fusce interdum accumsan velit in tempor. Vestibulum tempus erat nec turpis fringilla egestas. Donec scelerisque, justo sit amet porta efficitur, enim lectus tempus urna, vitae molestie mi ante sed justo. Nam tempor odio lectus, sed feugiat diam aliquam ac. Fusce congue bibendum enim et vehicula. Maecenas sagittis iaculis viverra.</p>
      <p>Cras non tempus risus, non tincidunt odio. Donec ac fringilla est. Sed nunc nisl, tincidunt in diam non, vehicula consectetur metus. Nam vel ipsum tincidunt, tristique quam eget, ultricies sapien. Nullam ac arcu ac diam gravida dapibus id eget nibh. Fusce mollis quis ipsum a porttitor. Vestibulum congue ultrices metus, nec porta erat aliquam nec. Nullam pharetra nunc erat. Cras id pharetra lacus. Quisque facilisis felis id rhoncus semper. Nulla accumsan semper condimentum. Etiam convallis velit at sem pretium auctor. Quisque eu est bibendum, gravida purus
```

```
ac, aliquam felis. Pellentesque aliquet pulvinar quam auctor mollis. Vivamus faucibus justo eget ligula laoreet, eget blandit erat vestibulum. Nunc lacus lorem, porta viverra dignissim in, porttitor sit amet justo.</p>
        <p>Morbi at augue consectetur, condimentum ante at, suscipit tellus. Duis dictum nulla id mollis interdum. Maecenas faucibus blandit metus non luctus. Sed eu orci mattis risus pretium malesuada. Mauris ac ex ac urna commodo dignissim. Mauris ut libero id massa aliquet egestas. Nunc eu sollicitudin ex. Fusce eget malesuada nisl, in iaculis eros. Nulla quis posuere tellus, ac fringilla ante. Integer fermentum turpis id urna volutpat gravida. Curabitur consequat tincidunt metus eu suscipit. Ut accumsan ultricies nisl ut imperdiet. Donec rutrum posuere convallis. Mauris lobortis, leo blandit porttitor tempus, leo magna interdum nunc, vitae tincidunt sapien augue eu lacus.</p>
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. In vel laoreet purus. Quisque in ex nec metus pulvinar elementum. Quisque ac efficitur tortor, a placerat neque. Curabitur vestibulum, libero ac dictum gravida, ligula libero dignissim nisi, et gravida velit urna vel est.</p>
    </div>
    <div class="footer">&copy;Copyright</div>
  </div>
</body>
</html>
```

fixed.css:

```
body {
  background: #f0f0f0;
}
.header {
  height: 100px;
  padding:20px;
  border-bottom: 2px solid #e0ecb8;
}
.layout {
  width: 800px;
  margin: 0 auto;
  background: #ffffff;
}
.content {
  width:600px;
  float:left;
  padding: 10px;
  background: #e0ecb8;
}
.sidebar {
  width: 200px;
  padding: 10px;
  float: right;
}
.sidebar ul  {
    margin-left: 0px;
    padding-left: 0px;
    list-style: none;
  }
.sidebar ul a {
  font-weight: bold;
  color: #e0ecb8;
  display: block;
  text-decoration: none;
  border-bottom: 1px dashed #e0ecb8;
}
.sidebar ul a:hover {
  color: #000000;
}
.footer {
  clear: both;
  padding: 10px;
  border-top: 2px solid #e0ecb8;
}
```

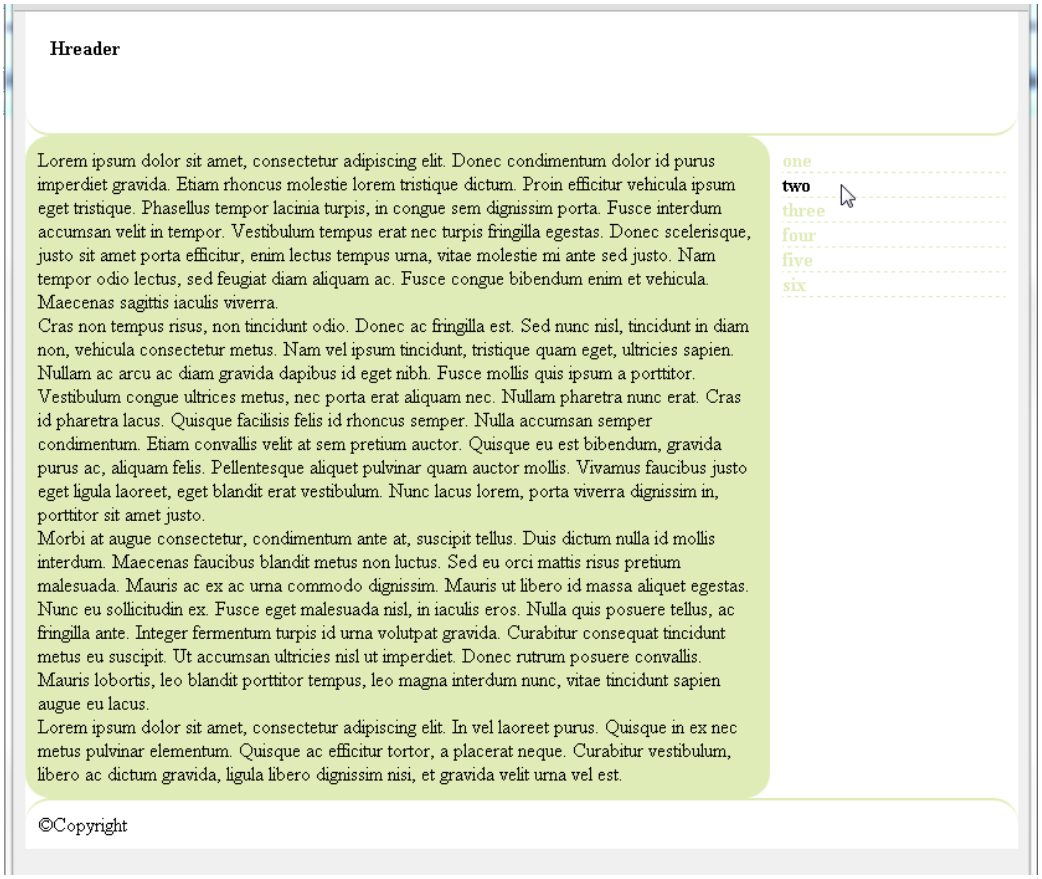
```
.header, .content, .sidebar, .footer{
   box-sizing: border-box;
   -webkit-box-sizing: border-box;
   -moz-box-sizing: border-box;
   border-radius: 20px 20px 20px 20px;
}
```

Підключити файл reset.css для скидання стандартних стилів:

reset.css

```
html, body, div, span, object, iframe, h1, h2, h3, h4, h5, h6, p, blockquote, pre, a, abbr, acronym, address, big, cite,
code, del, dfn, em, img, ins, kbd, q, s, samp,small, strike, strong, sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul, li,
fieldset, form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td, article, aside, canvas, details, embed,
figure, figcaption, footer, header, hgroup, menu, nav, output, ruby, section, summary, time, mark, audio, video {
   margin: 0;
   padding: 0;
   border: 0;
   font-size: 100%;
   vertical-align: baseline;
}
article, aside, details, figcaption, figure, footer, header, hgroup, menu, nav, section {
   display: block;
}
body {
   line-height: 1.2;
}
ol {
   padding-left: 1.4em;
   list-style: decimal;
}
ul {
   padding-left: 1.4em;
   list-style: square;
}
table {
   border-collapse: collapse;
   border-spacing: 0;
}
```

Зразок



Розібратись в структурі html та властивостях css.

2 Перетворити попередній макет веб-сторінки у дві колонки у гумовий не порушуючи структури html (зберегти файли під назвами fluid.html та fluid.css).
Вимоги:

- ширина макету може коливатись від 769 до 1200px;
- ширина для sidebar – 20%.

3 Використовуючи методи адаптивного веб-дизайну розробити макет сайту (зберегти файли під назвами rwd.html та rwd.css) відповідно до наступних вимог:

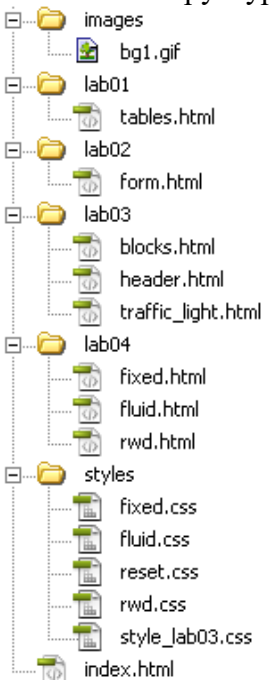
- для ПК використовується фіксований макет у дві колонки (див. завдання 1 ходу роботи);
- для планшетних пристоїв використовується гумовий макет у дві колонки (див. завдання 2 ходу роботи);
- для смартфонів використовується гумовий макет в одну колонку;
- передбачити відміну масштабування сайту мобільними браузерами;
- використовувати одну таблицю стилів (вище згадуваний rwd.css).

Структура таблиці стилів повинна мати вигляд:

```
/* Сюди поміщаються стилі, що перезапускають вихідні настройки браузера */
/* Сюди поміщаються стилі для браузерів настільних пристроїв і основні стилі для всіх пристроїв */
body {
/* Властивості для тіла документа */
}
/* Тільки для дисплеїв середньої ширини */
@media (min-width: 481px) and (max-width: 768px) {
body {
/* Властивості, застосовувані тільки до браузерів планшетних пристроїв */
}
}
/* Тільки для дисплеїв малої ширини */
@media (max-width: 480px) {
body {
```

```
/* Властивості, застосовувані тільки до браузерів телефонів */
}
}
```

4 Забезпечити структуру папок та файлів відповідно до поданого нижче зразка:



5 Додати посилання на створені сторінки до індексного файлу:

Веб-технології і веб-дизайн. Лабораторні роботи

1. Створення html-документів з таблицями
2. Створення форм за допомогою HTML
3. Створення веб-сторінок з використанням CSS
    1. Блоки
    2. Світлофор
    3. Шапка
4. Створення макетів сайту
    1. Фіксований у дві колонки
    2. Гумовий у дві колонки
    3. Адаптивний

6 Показати роботу викладачу. Оформити звіт. Захистити роботу.

ЗМІСТ ЗВІТУ

1 Тема
2 Мета
3 Код
4 Схематичне зображення на екрані
5 Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ:

1 Яке призначення властивості float?
2 Яке призначення властивості clear?
3 Яке призначення властивості box-sizing?
4 Які особливості створення фіксованого макету у дві колонки?
5 Які особливості створення гумового макету у дві колонки?
6 Охарактеризувати створення макету сайту з використанням RWD.

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.

ЛАБОРАТОРНА РОБОТА №5 (2 год.)

ТЕМА. Управління процесом виконання сценаріїв в JavaScript

НАВЧАЛЬНА МЕТА: навчитися використовувати конструкції JavaScript для управління процесом виконання сценаріїв.

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

JavaScript є мовою сценаріїв, який працює виключно на стороні клієнта всередині браузера. Для виклику цієї мови його код поміщається між відкриваючим і закриваючим HTML-тегами <*script*> і </*script*>:

```
<body>
<script>
  document.write("Текст");
</script>
```

Підключення файлів javascript:

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

В JavaScript функції використовуються для виділення фрагментів коду, що виконують конкретну задачу. Для створення функції її потрібно оголосити:

```
function product (a, b)
{
return a * b
}
```

Приклади використання

Збереження результату роботи функції у змінній:

```
var $ = function(id) { return document.getElementById(id); };
ui.prompt = $("prompt");
```

Неявне використання функції:

```
document.getElementById('showhide').onclick = function () {
changeDisplayState('toggleMe');
return false;
};
```

Інструкція if

Приклади:

```
if(a > 100) {
 b=2;
 document.write("a більше 100");
}

if(b == 10) document.write("b дорівнює 10");

if(a > 100) {
 document.write("a більше 100");
}
else{
 document.write("a менше чи дорівнює 100");
}

if(a > 100) {
 document.write("a більше 100");
}
else if(a < 100) {
 document.write("a менше 100");
}
else{
```

```
 document.write("а дорівнює 100");
}
```

Цикли for

Цикл *for* поєднує всі найкращі якості організації циклу в одній конструкції, яка дозволяє передати кожній інструкції три параметри:

–вираз ініціалізації;

–вираз умови;

–вираз модифікації.

Приклад. Використання циклу for

```
for(var count = 1; count <= 7; ++count) {
  document.write(count + " помножити на 7 дорівнює " + count * 7 + "<br>");
}
```

В першому параметрі циклу *for* можна присвоювати значення відразу декільком змінним, розділяючи вирази комами:

```
for(var i = 1, j = 1; i < 10; i++)
```

Точно так само в останньому параметрі можна здійснювати відразу декілька модифікацій:

```
for(i = 1; i < 10; i++, --j)
```

Або можна одночасно робити і те й інше:

```
for(var i = 1, j = 1; i < 10; i++, --j)
```

ХІД РОБОТИ

1 Скопіювати папку lab05 з матеріалами для роботи у власну папку.

2 У скопійованій папці lab05 створити сторінку lab_05_1.html. Вивести в тіло веб-сторінки всі непарні числа з вказаного діапазону. Початкове та кінцеве значення діапазону задавати за допомогою змінних. В зовнішньому файлі реалізувати відповідну функцію, яка повертатиме рядок парних чисел, розділених комою та пробілом. Після останнього значення коми та пробілу бути не повинно.

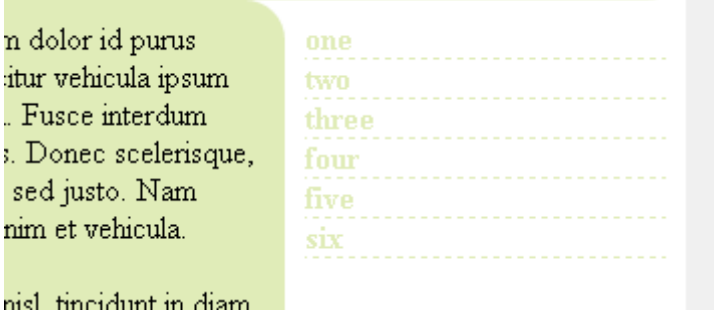
3 Відкрити файл lab_05_2.html. В тіло веб-сторінки інтегрувати наступний скрипт:

```
var changeDisplayState = function (id) {
                        var d = document.getElementById('showhide'), e = document.getElementById(id);
                        if (e.style.display === 'none' || e.style.display === '') {
                        e.style.display = 'block';
                        d.innerHTML = 'Hide paragraph';
                        }
                        else {
                            e.style.display = 'none';
                            d.innerHTML = 'Show paragraph';
                        }
                    };
document.getElementById('showhide').onclick = function () {
                                                changeDisplayState('toggleMe');
                                                return false;
                                            };
```

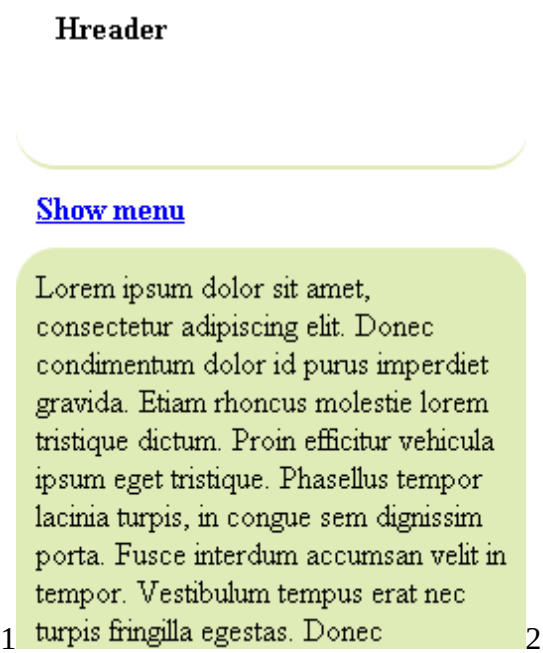
Розібратися в коді. Розставити коментарі. Протестувати сторінку та зробити висновки.

4 Вдосконалити скрипт з попереднього завдання, реалізувавши перевірку підтримки стандартів DOM.

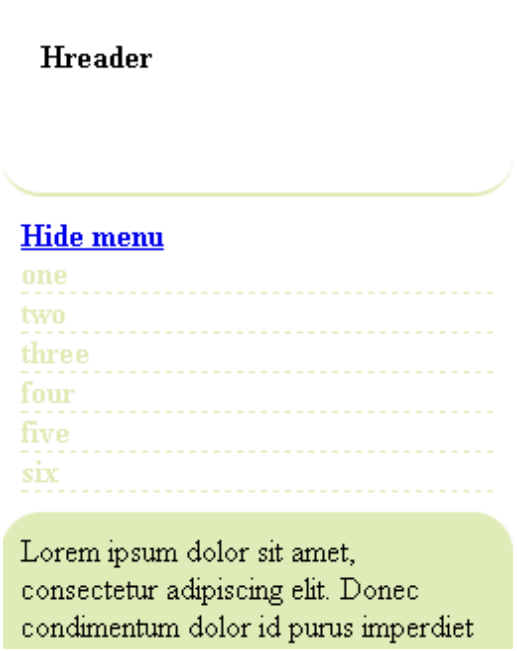
5 Відкрити сторінку lab_05_3.html. Пересвідчитись в правильності роботи адаптивного макету. Реалізувати приховання/відображення меню при застосуванні до сторінки стилю «для смартфонів» (макету в одну колонку). Приховання чи відображення меню повинне спрацьовувати при натисненні на відповідному посиланні, яке потрібно попередньо додати в макет. При застосуванні до сторінки інших стилів (для планшетів та ПК) посилання відображатись не повинне, треба відображати лише меню:



При застосуванні макету для смартфонів (в одну колонку), посилання повинне відображатись і при клацанні на ньому показувати/приховувати меню:



1



2

6 Показати роботу викладачу. Оформити звіт. Захистити роботу.

ЗМІСТ ЗВІТУ

1. Тема
2. Мета
3. Код
4. Схематичне зображення на екрані
5. Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ:

1. Охарактеризувати синтаксис та семантику інструкції try…catch.
2. Охарактеризувати синтаксис та семантику інструкції if.
3. Охарактеризувати синтаксис та семантику інструкції switch.
4. Охарактеризувати синтаксис та семантику оператора «?».
5. Охарактеризувати синтаксис та семантику інструкції while.
6. Охарактеризувати синтаксис та семантику інструкції do…while.
7. Охарактеризувати синтаксис та семантику інструкції for.
8. Як здійснити припинення роботи циклу?
9. Для чого призначена інструкція continue?

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.

ЛАБОРАТОРНА РОБОТА №6 (2 год.)

ТЕМА. Обробка масивів

НАВЧАЛЬНА МЕТА: навчитися використовувати функціонал JavaScript для роботи з масивами.

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Оголошення масивів:

```
var arrayname = new Array();
```

або

```
var arrayname = [];
```

Методи для роботи з масивами: push, shift, length, splice, reverse, sort, forEach, join.

Приклад застосування методу до масиву:

```
var pets = ["Кішка", "Собака", "Кролик", "Хом'як"];
document.write(pets.join() + "<br>");
document.write(pets.join(' ') + "<br>");
document.write(pets.join(' : ') + "<br>");
```

Масив аргументів

Складовою частиною кожної функції є масив аргументів - *arguments*. Завдяки йому можна визначити кількість змінних, переданих функції, і зрозуміти, що вони собою являють. Приклад:

```
displayItems("Собака", "Кішка", "Поні", "Хом'як", "Черепаха");
function displayItems(v1, v2, v3, v4, v5) {
  document.write(v1 + "<br>");
  document.write(v2 + "<br>");
  document.write(v3 + "<br>");
  document.write(v4 + "<br>");
  document.write(v5 + "<br>");
}
```

Використання масиву arguments у функції, якій можна передавати довільну кількість параметрів:

```
function displayItems() {
  for (j = 0; j < displayItems.arguments.length; ++j) {
    document.write(displayItems.arguments[j] + "<br>");
  }
}
```

ХІД РОБОТИ

Зауваження:

1. Для виконання роботи потрібно створити html-файл та файл для реалізації допоміжних функцій JavaScript.
2. Підчас виконання завдань потрібно обов’язково використовувати коментарі та дотримуватись стилю програмування.
3. Виведення інформації на екран повинне бути інформативним.

1Оголосити масив прізвищ студентів групи. В зовнішньому файлі реалізувати допоміжну функцію для виведення елементів масиву на екран. Функція повинна приймати два параметри: перший – власне, сам масив; другий – рядок-роздільник. Виконати такі дії над масивом:

1) вивести на екран всі елементи масиву;
2) додати ще одне прізвище в кінець масиву та вивести елементи масиву на екран;
3) вилучити перший елемент масиву та вивести решту на екран
4) додати два прізвища в середину масиву та вивести елементи масиву на екран;
5) вивести на екран всі елементи масиву в зворотньому порядку;
6) відсортувати масив за алфавітом та вивести на екран;
7) відсортувати масив за алфавітом в зворотньому порядку та вивести на екран.

Приклад результату виконання завдання:

Завдання 1
1) Оголошений масив:
Коваленко Петренко Лобанов Акриленко
2) Масив після додавання нового прізвища в кінець:
Коваленко Петренко Лобанов Акриленко Вернигора
3) Масив після вилучення першого елемента:
Петренко Лобанов Акриленко Вернигора
4) Масив після додавання нових прізвищ в середину:
Петренко Лобанов Мякушко Островський Акриленко Вернигора
5) Зворотній порядок елементів масиву:
Вернигора Акриленко Островський Мякушко Лобанов Петренко
6) Відсортовано за алфавітом:
Акриленко Вернигора Лобанов Мякушко Островський Петренко
7) Відсортовано за алфавітом в зворотньому порядку:
Петренко Островський Мякушко Лобанов Вернигора Акриленко

2 Вивести на екран всі елементи створеного в попередньому завданні масива за зразком. Використати метод forEach та неіменовану функцію.
Зразок:

Завдання 2
Відображення елементів масиву (за допомогою forEach):
0 - Петренко
1 - Островський
2 - Мякушко
3 - Лобанов
4 - Вернигора
5 - Акриленко

3 Оголосити масив довільних цілих чисел. Відсортувати оголощений масив спочатку за зростанням, а потім – за спаданням. Вивести інформацію на екран, забезпечивши інформативність.
Приклад результату виконання завдання:

Завдання 3
Масив цілих чисел:
11 23 6 112
Масив цілих чисел за зростанням:
6 11 23 112
Масив цілих чисел за спаданням:
112 23 11 6

4 Реалізувати функцію для вибірки елементів з масиву слів за заданими критеріями, які передаватимуться їй в якості відповідних параметрів (масив, початкова послідовність символів в слові, мінімальна кількість символів в слові).
Використовуючи дану функцію вивести на екран ті прізвища з масиву, створеного в першому завданні, які починаються на «Пе» та в яких не менше 6 символів (вхідні дані можна коригувати).

5 Дослідити яку задачу виконує наступний фрагмент коду:

```
document.write(fixNames("kOVALENKO", "PETRENKO", "IvANOv"));
function fixNames() {
  var s = "";
  for (j = 0; j < fixNames.arguments.length; ++j) {
    s += fixNames.arguments[j].charAt(0).toUpperCase() + fixNames.arguments[j].substr(1).toLowerCase()
+ " ";
  }
  return s.substr(0, s.length-1);
}
```

Реалізувати дану задачу з використанням масиву.

ЗМІСТ ЗВІТУ

1 Тема
2 Мета
3 Код
4 Схематичне зображення на екрані
5 Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ:

1 Як використовується масив аргументів?
2 Як оголосити числовий масив?
3 Як оголосити асоціативний масив?
4 Як оголошується багатовимірний масив?
5 Яке призначення методу push?
6 Яке призначення методу pop?
7 Яке призначення методу concat?
8 Яке призначення методу forEach?
9 Яке призначення методу join?
10 Яке призначення методу reverse?
11 Яке призначення методу sort?

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.

ЛАБОРАТОРНА РОБОТА №7 (2 год.)

ТЕМА. Методи об’єкта Window

НАВЧАЛЬНА МЕТА: навчитися використовувати на практиці методи об’єкта Window.

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Слідуючи концепції інтеграції JavaScript в існуючі системи, браузери підтримують включення скрипта, наприклад, в значення атрибуту події:

Видалити

Тут при натисненні на посилання функція confirm('Ви впевнені?'); викликає модальне вікно з написом «Ви впевнені?», а return false; блокує перехід за посиланням. Зрозуміло, цей код працюватиме тільки якщо в браузері є і включена підтримка JavaScript, інакше перехід за посиланням відбудеться без попередження.

Є і третя можливість підключення JavaScript — написати скрипт в окремому файлі, а по тому підключити його за допомогою конструкції:

<script type="text/javascript" src="http://Шлях_до_файла_зі_скриптом"> </script>

Об'єкт Window має три методи для відображення простих діалогів. Метод alert() виводить повідомлення і чекає, поки користувач закриє диалогове вікно. Метод confirm() пропонує користувачеві клацнути на кнопці OK або Cancel (Відміна) і повертає логічне значення. Метод prompt() виводить повідомлення, чекає введення рядка користувачем і повертає цей рядок. Нижче демонструється приклад використання усіх трьох методів:

```
do {
    var name = prompt("Введіть ваше ім'я"); //Поверне рядок
    var correct = confirm("Ви ввели '" + name + "'.\n " + //Поверне логічне значення
                          "Клацніть OK, щоб продовжити, " +
                          "або Отмена, щоб повторити введення.");
} while (!correct)
alert("Привіт, " + name + "."); //Виведе просте повідомлення
```

Методи **alert(), confirm() і prompt()** надзвичайно прості у використанні, але правила хорошого дизайну вимагають, щоб вони застосовувалися як можна рідше. Діалоги, подібні до цих, нечасто використовуються у Веб, і більшість користувачів вважатиме діалогові вікна, що виводяться цими методами, випадаючими із звичайної практики. Єдиний варіант, коли має сенс звертатися до цих методів, – це відладка. JavaScript -програмісти часто вставляють виклик методу alert() в програмний код, намагаючись діагностувати виниклі проблеми.

Зверніть увагу, що текст, який відображається методами alert(), confirm() і prompt() в діалогах, – це звичайний неформатований текст. Його можна форматувати тільки пропусками, переведеннями рядків і різними знаками пунктуації.

Методи confirm() і prompt() є блокуючими, тобто вони не повертають управління, поки користувач не закриє діалогові вікна, що відображаються ними. Це означає, що, коли виводиться одне з цих вікон, програмний код припиняє виконання, і поточний документ, що завантажується, якщо такий існує, припиняє завантажуватися до тих пір, поки користувач не відреагує на запит. У більшості браузерів метод alert() також є блокуючим і чекає від користувача закриття діалогового вікна, але це не є обов'язковою вимогою.

Об’єкт **Document** має важливий метод getElementById(), який повертає єдиний елемент документа (елемент являє собою пару із відкриваючого та закриваючого тегів HTML та все, що міститься між ними), спираючись на значення атрибута id елемента.

Перетворення в число

Строге перетворення можна здійснити унарним плюсом '+'

var s = "12.34";

alert (+s); / / 12.34

Строге - означає, що якщо рядок не є в точності числом , то результат буде NaN :
alert (+"12test"); / / NaN (Not-A-Number)
Єдиний виняток - пробільні символи на початку і в кінці рядка , які ігноруються :
alert (+" -12 "); / / -12
alert (+" \n34 \ n ") ; / / 34 , переклад рядка \ n є пробільних символом
alert (+"") ; / / 0 , порожній рядок стає нулем
alert (+"1 2") ; / / NaN , пробіл посередині числа - помилка
Аналогічним чином відбувається перетворення і в інших математичних операторах і функціях:
alert ('12 .34 '/ "-2") ; / / -6.17
М'яке перетворення : parseInt і parseFloat
У світі HTML / CSS багато значення не є в точності числами. Наприклад , метрики CSS: 10pt або - 12px .
Оператор '+' для таких значень поверне NaN :
alert (+"12px") / / NaN
Для зручного читання таких значень існує функція parseInt :
alert (parseInt ('12px')); / / 12
parseInt і її аналог parseFloat перетворять рядок символ за символом, поки це можливо.
При виникненні помилки повертається число, яке вийшло. parseInt читає з рядка ціле число, а parseFloat - дробове .
alert (parseInt ('12px')) / / 12 , помилка на символі ' p '
alert (parseFloat ('12.3.4')) / / 12.3 , помилка на другій точці
Звичайно, існують ситуації, коли parseInt / parseFloat повертають NaN. Це відбувається при помилці на першому ж символі :
alert (parseInt ('a123')); / / NaN

ХІД РОБОТИ

1 Створіть файл lab07test.js наступний код, що містить:

```
function verifyButton() {
    alert("Просте повідомлення");
    result = confirm("Перевіримо, яку кнопку ви натиснули");
    if (result) {
        document.write("<p>Ви натиснули кнопку OK</p>");
    } else {
        document.write("Ви натиснули кнопку Отмена");
    }
}
```

2 Створіть новий файл lab07test.html:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Перевірка натиснутої кнопки</title>
  </head>
  <body>
    <h1>Javascript</h1>
  </body>
</html>
```

3 Додати на сторінку кнопку і зробити так, щоб функція, створена в першому завданні спрацьовувала при натисненні саме на цю кнопку.

4 Створіть файл lab07_4.html за зразком:

```
<!doctype html>
  <head>
  <meta charset="UTF-8">
  <title>Validate</title>
  </head>
<body>
  <form method="post" action="handler.php">
    <label for="surname">Прізвище</label><br><input type="text" id="surname"><br>
    <label for="firstname">Ім'я</label><br><input type="text" id="firstname"><br>
```

```
    <label for="username">Ім'я користувача</label><br><input type="text" id="username"><br>
    <input type="submit" value = "Sent">
  </form>
</body>
</html>
```

В файлі lab07test.js реалізувати наступну функцію:

```
function validate(){
    var surname=document.getElementById("surname");
    if (surname.value==""){
        surname.style.backgroundColor="red";
    } else {
        surname.style.backgroundColor="white";
    }
    var firstname=document.getElementById("firstname");
    if (firstname.value==""){
        firstname.style.backgroundColor="red";
    } else {
        firstname.style.backgroundColor="white";
    }
    var username=document.getElementById("username");
    if (username.value==""){
        username.style.backgroundColor="red";
    } else {
        username.style.backgroundColor="white";
    }
}
```

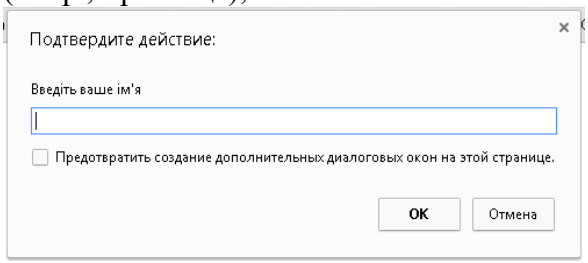
Розібратись в структурі функції, розставити коментарі.

Доопрацювати функцію таким чином, щоб виводилась повідомлення про помилку, якщо хоч б одне з полів не заповнене.

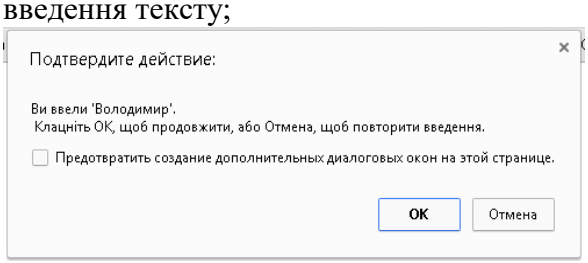
Причепити доопрацьовану функцію на кнопку відправки даних форми.

5 Реалізувати функцію, який буде виконувати наступні дії:

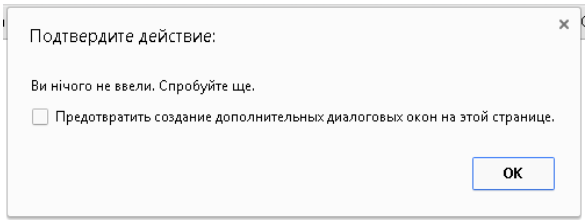
1) виводити користувачу діалогове вікно з проханням ввести довільний рядок тексту (напр., прізвище);



2) виводити на екран введений рядок та забезпечувати можливість повторного введення тексту;



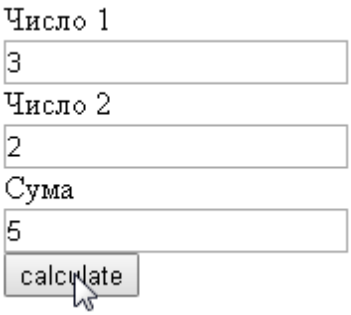
3) перевіряти коректність введення (чи не порожній рядок) та виводити відповідні повідомлення;



Функцію інтегрувати в lab07test.js, або в попередньо створений файл lab07_5.html. В html-файлі потрібно розмістити посилання, до якого «причепити» реалізовану функцію. Функція повинна спрацьовувати при наведенні на посилання курсором.

6 Реалізувати веб-інтерфейс для знаходження суми двох чисел.

Приклад веб-інтерфейсу:



ЗМІСТ ЗВІТУ

1 Тема
2 Мета
3 Код
4 Схематичне зображення на екрані
5 Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ:

1 Як реалізувати виведення тексту у вікні за допомогою JavaScript?
2 Для чого призначений метод alert()?
3 Для чого призначений метод confirm()?
4 Для чого призначений метод prompt()?
5 Охарактеризувати використання методу getElementById.

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.

ЛАБОРАТОРНА РОБОТА №8 (2 год.)

ТЕМА. Графічні та мультимедійні ефекти на веб-сторінках

НАВЧАЛЬНА МЕТА: навчитися інтегрувати у веб-документи відео- та аудіо об'єкти за допомогою елементів <audio> і <video>, створювати графічні ефекти за допомогою елементу <canvas>.

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Як і інші HTML-елементи, canvas - це просто елемент в складі веб-сторінки з певними розмірами, всередині якого можна для малювання графіки використовувати JavaScript. Полотно створюється за допомогою тега <canvas>, якому також потрібно присвоїти ідентифікатор, щоб в коді JavaScript було зрозуміло, до якого саме полотна йде звернення.

Приклад 1. Відображення прапору японії за допомогою полотна

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8"/>
  <title id="title">Flag of Japan</title>
  <script src="osc.js"></script>
</head>
<body>
  <canvas id= "mycanvas" width="320" height="240">
  Це элемент canvas с ідентифікатором <i>mycanvas</i>
  Цей текст відображається тільки в браузерах, що не підтримують HTML5
  </canvas>
  <script>
    //Створення об'єкта canvas
    var canvas = O('mycanvas');
    /*Команда викликає метод getContext нового, щойно створеного об'єкта canvas, запитуючи двовимірний доступ до полотна шляхом передачі значення '2d'*/
    var context = canvas.getContext('2d');
    /*Встановити для властивості fillStyle цього контексту значення 'red'*/
    context.fillStyle = 'red';
    /*Встановити для властивості border об'єкта canvas значення однопіксельної суцільної чорної лінії для контуру зображення прапора*/
    S(canvas).border = '1px solid black';
    /*Відкрити контур контексту, і позиція малювання переміщається до точки з координатами (160; 120)*/
    context.beginPath();
    context.moveTo(160, 120);
    /*Намалювати дугу з центром із заданими координатами. Її радіус - 70 пікселів, вона починається з кута 0° (що є правим краєм кола, якщо дивитися прямо на неї) і триває по всьому колу в радіанах, що визначаються значенням 2×π. Останнє значення false служить ознакою напрямку малювання дуги за годинниковою стрілкою; а значення true було б ознакою малювання, здійснюваного проти годинникової стрілки.*/
    context.arc(160, 120, 70, 0, Math.PI * 2, false);
    //Закрити контур
    context.closePath();
    /*Здійснити заливку контуру, використовуючи значення в властивості fillStyle*/
    context.fill();
  </script>
</body>
</html>
```

Методи для роботи з прямокутниками

fillRect, clearRect, strokeRect

Приклад 2. Заливка прямокутників

```
var canvas = O('mycanvas');
var context = canvas.getContext('2d');
S(canvas).background = 'lightblue';
context.fillStyle = 'blue';
context.strokeStyle = 'green';
context.fillRect(20, 20, 600, 200);
context.clearRect(40, 40, 560, 160);
context.strokeRect(60, 60, 520, 120);
```

Використання градієнтів

createLinearGradient, addColorStop, createRadialGradient

Приклад 3. Градієнти

```
var gradient = context.createLinearGradient(0, 80, 640,80);
gradient.addColorStop(0, 'white');
gradient.addColorStop(1, 'black');
context.fillStyle = gradient;
context.fillRect(80, 80, 480,80);
```

Методи та властивості для роботи з текстом

strokeText, textBaseLine, font, textAlign, fillText

Приклад 4. Запис тексту на полотні

```
var canvas = O('mycanvas');
var context = canvas.getContext('2d');
context.font = 'bold 140px Times';
context.textBaseline = 'top';
context.textAlign = 'left';
context.strokeText('Canvas text', 0, 0);

/*Заливка тексту за допомогою узору*/
image = new Image();
image.src = 'image.png';
  image.onload = function() {
    var pattern = context.createPattern(image, 'repeat');
    context.fillStyle = pattern;
    context.fillText('Canvas text', 0, 0);
    context.strokeText('Canvas text', 0, 0);
  }
```

Контури

beginPath, closePath, moveTo, LineTo, stroke, rect, fill, arc, bezierCurveTo

Приклад 5. Малювання контуру по точкам

```
context.beginPath();
context.moveTo(20, 100);
context.lineTo(20, 20);
context.lineTo(155, 20);
context.lineTo(155,100);
context.stroke();
context.closePath();
```

bezierCurveTo

Крива створюється між позиціями (24; 20) і (240; 220), але з невидимими натяжними точками за межами полотна (в даному випадку) в позиціях (720; 480) і (-240; -240):

Приклад 6. Використання bezierCurveTo

```
context.beginPath();
context.moveTo(240, 20);
context.bezierCurveTo(720, 480, −240, −240, 240, 220);
context.stroke();
context.closePath();
```

Методи для обробки зображень

drawImage, shadowOffsetX, shadowOffsetY, shadowBlur, shadowColor

Приклад 7. Завантаження зображення

```
var myimage = new Image();
myimage.src = 'image.png';
myimage.onload = function() {
```

```
  context.drawImage(myimage, 20, 20);
}
```

ХІД РОБОТИ

Зауваження. Всі необхідні виконання роботи матеріали містяться в папці d:\work\ps3\web\lab\lab08.

1 Створити html-документ lab08.html.

2 Підключити до створеного документа osc.js.

3 Додати канву 620x620.

4 Створити в канві наступну картинку:

Рекомендації та вимоги:

- текст потрібно залити лінійним градієнтом, в якому використати дві колірні опорні точки;
- для створення дуги в прапорі можна використати метод bezierCurveTo;
- для реалізації тризуба використати файл tryzub.png, метод drawImage та масштабування (метод scale);
- для кожного з елементів (текст, тризуб, прапор) потрібно застосовувати різний тип тіні (напр., змінювати колір).

5 Додати в документ елемент аудіо для прослуховування гімну. Відмовитись від стандартної панелі управління плеєра. Забезпечити сумісність із старими браузерами (вивести відповідне повідомлення та посилання на скачування).

6 Реалізувати управління за допомогою javascript: дві кнопки (одна для запуску, інша – пауза). На кнопки причепити відповідні функції. У функціях потрібно застосувати до створеного аудіооб’єкта методи play() та pause():

Результат:

ЗМІСТ ЗВІТУ

1 Тема
2 Мета
3 Код
4 Схематичне зображення на екрані
5 Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ:

1 Яке призначення елементу canvas?
2 Як створити полотно за допомогою JavaScript?
3 Які методи використовуються для роботи з прямокутниками?
4 Які методи JavaScript об’єкта canvas використовуються для створення градієнтних заливок?
5 Охарактеризувати методи та властивості для роботи з текстом на полотні.
6 Охарактеризувати методи та властивості для роботи з контурами.

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.

ЛАБОРАТОРНА РОБОТА №9 (2 год.)

ТЕМА. Використання функцій PHP

НАВЧАЛЬНА МЕТА: навчитися використовувати інструментарій PHP для створення і роботи з функціями та об'єктами.

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Функції використовуються для виділення блоків коду, що виконують конкретне завдання. Наприклад, якщо вам часто доводиться шукати якісь дані і виводити їх в певному форматі, то цілком розумно буде звернутися до функції.

```
function ім'я_функції([параметр [, ...]]) {
  //Інструкції
}
```

Імена функцій не чутливі до регістру.

У складі цих інструкцій повинна бути одна або кілька інструкцій *return*, які змушують функцію припинити виконання і повернути управління коду, що викликав функцію. Якщо інструкція *return* продовжена якимось значенням, то викликаючий код може його отримати.

Визначення класу включає в себе ім'я класу (чутливе до регістру), його властивості та методи:

У прикладі клас *Subscriber* оголошується підкласом *User* шляхом використання інструкції *extends:*

```
$object = new Subscriber;
$object->name = "Fred";
$object->password = "pword";
$object->phone = "012 345 6789";
$object->email = "fred@bloggs.com";
$object->display();
class User {
  public $name, $password;
  function save_user() {
    echo "Сюда розміщується код, що зберігає дані користувача";
  }
}
class Subscriber extends User {
  public $phone, $email;
  function display() {
    echo "Name: " . $this->name . "<br>";
    echo "Pass: " . $this->password . "<br>";
    echo "Phone: " . $this->phone . "<br>";
    echo "Email: " . $this->email;
  }
}
```

ХІД РОБОТИ

Зауваження. Всі реалізовані функції та класи повинні бути розміщені в окремому файлі (functions.php), котрий підключатиметься до робочої сторінки. Текстову інформацію задавати на англійській мові або транслітом.

1 Налаштувати веб-сервер. Налаштувати власний віртуальний хост.

2 Створити робочу сторінку lab09.php. Сформувати html-структуру за допомогою функціоналу PHP.

3 Створити скрипт functions.php.

4 Визначити поточну версію установки PHP.

5 Реалізувати та використати функцію, яка переводить милі в кілометри (1 миля=1.609344 км). Забезпечити приведення типів та захист від некоректних аргументів.

6 Реалізувати та використати функцію, яка приводить в порядок ПІБ (виправляє регістр), що передаються в якості параметрів. Функція повинна повертати рядкове значеня. Використати допоміжні функції: ucfirst, strtolower. Вхідні дані: pETrenko, IvaN, ivanOvych. Результат роботи функції:

Prizvyshe: Petrenko
Imya: Ivan
Po-batkovi: Ivanovych

7 Реалізувати та використати функцію, яка приводить в порядок ПІБ (виправляє регістр), що передаються в якості параметрів. Функція повинна повертати масив.

8 Реалізувати та використати функцію, яка приводить в порядок ПІБ (виправляє регістр), що передаються в якості параметрів. В якості аргументів функції повинні передаватися посилання на відповідні змінні.

9 Реалізувати клас FootballPlayer, який описуватиме відомості про футболістів: команда, прізвище, ім'я, дата народження, амплуа, номер. Реалізувати конструктор для початкової ініціалізації даних класу. Реалізувати метод для виведення інформації про об'єкт класу.

10 Розширити клас, створений у попередньому завданні, новим класом FootballPlayerStat, який міститиме наступну інформацію: зріст, захищені дані – вага, кількість проведених матчів, кількість забитих голів. Реалізувати конструктор для початкової ініціалізації даних класу. Реалізувати методи для зміни та отримання ваги, кількості проведених матчів, кількості забитих голів. Реалізувати метод для визначення відсотку результативності.

11 Протестувати роботу класів: створити екземпляр класу, провести ініціалізацію всіх властивостей, перевірити властивості та методи.

12 Показати роботу викладачу.

ЗМІСТ ЗВІТУ

1 Тема
2 Мета
3 Код
4 Схематичне зображення на екрані
5 Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ:

1 Як реалізувати функцію в PHP?
2 Яке призначення службового слова return?
3 Які є можливі способи передачі параметрів функції?
4 Як реалізувати клас в PHP?
5 Як розширити клас PHP?
6 Як створити екземпляр класу?
7 Як забезпечити початкову ініціалізацію об'єктів класу?

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.

ЛАБОРАТОРНА РОБОТА №10 (2 год.)

ТЕМА. Створення та обробка масивів в PHP

НАВЧАЛЬНА МЕТА: навчитися використовувати інструментарій PHP для роботи із масивами.

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Масиви з числовою індексацією

```
$paper[] = "Copier";
$paper[] = "Inkjet";
print_r($paper);
```

Або

```
$paper[0] = "Copier";
$paper[1] = "Inkjet";
print_r($paper);
```

Видобування елементів із масиву:

```
for ($j=0; $j<4; ++$j) echo "$j: $paper[$j]<br>";
```

Асоціативні масиви

```
$paper['copier'] = "Copier & Multipurpose";
$paper['inkjet'] = "Inkjet Printer";
echo $paper['laser'];
```

Ключове слово array

```
$p1 = array("Copier", "Inkjet", "Laser", "Photo");
echo "Элемент массива p1: " . $p1[2] . "<br>";
$p2 = array('copier' => "Copier & Multipurpose",
      'photo' => "Photographic Paper");
echo "Элемент массива p2: " . $p2['photo'] . "<br>";
```

Формат: *індекс => значення*

Цикл foreach…as

```
$paper = array( 'copier' => "Copier & Multipurpose",
        'photo' => "Photographic Paper");
foreach ($paper as $item => $description) echo "$item: $description<br>";
```

ХІД РОБОТИ

Примітка. Для ознайомлення із функціями використовувати довідник.

1 Створити файл form_count_elem.html в якому створити поле введення для вказівки числа цифр в одновимірному масиві (рис. 1).

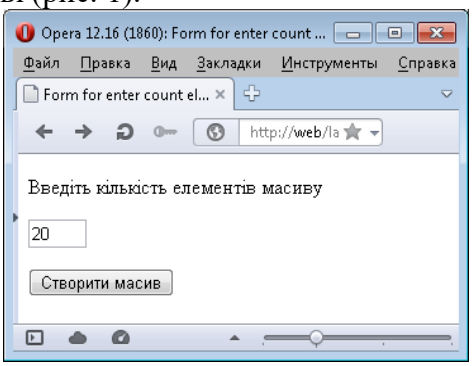


Рисунок 1 – Вигляд форми в браузері

2 Створити файл form_count_elem.php в якому, використовуючи функцію rand(min, max) заповнити масив двозначними випадковими числами. Вивести масив на екран в рядок з пропусками між словами. Алгоритм виконання наступний:

- В циклі for, в якому кількість операцій береться з поля введення у файлі form_count_elem.html, заповнити масив випадковими числами;
- Вивести заголовок "Масив на елементів заповнений випадковими числами" (див. рис. 2);
- В циклі вивести елементи масиву (див. рис. 2).

3 Використовую функцію sort() відсортувати масив за зростанням і вивести результат на екран (див. рис. 2).

4 Використовуючи функцію array_reverse() перевернути елементи масиву в зворотному порядку і результат вивести на екран (див. рис. 2).

5 Видалити останній елемент з масиву (функція array_pop()), вивести масив на екран (див. рис. 2).

6 Підрахувати суму елементів в масиві array_sum() і кількість елементів в масиві count(). Знайти і вивести на екран середнє арифметичне для елементів масиву (див. рис. 2).

7 Додати в масив значення "100" з ключем "maximum".

8 Використовуючи цикл foreach, вивести на екран усі елементи масиву (див. рис. 2).

9 Використовуючи функцію in_array визначити чи є в масиві число 50. Вивести на екран відповідне повідомлення (див. рис. 2).

10 Використовуючи функцію array_unique видалити з масиву значення, що повторюються. Порівняти результати (див. рис. 2).

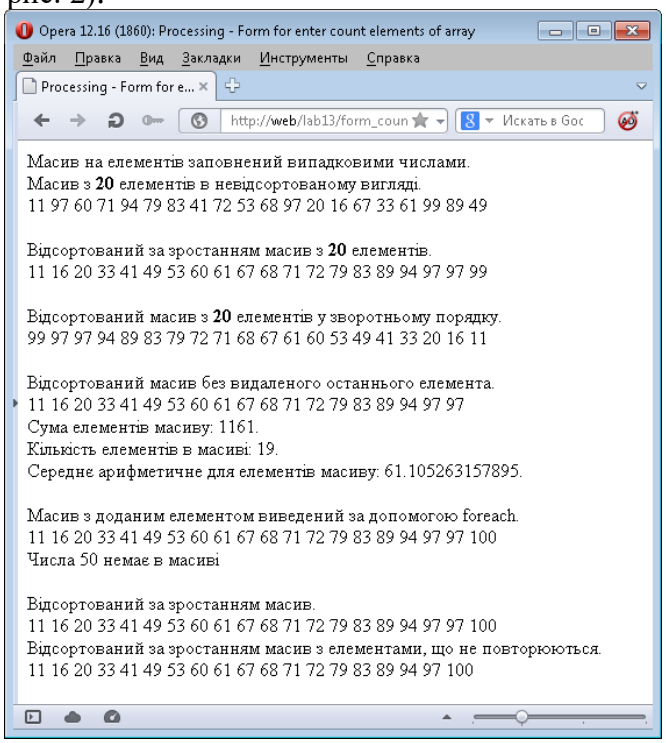


Рисунок 2 – Результат роботи скрипта form_count_elem.php

11 Показати роботу викладачу.

ЗМІСТ ЗВІТУ

1 Тема
2 Мета
3 Код
4 Схематичне зображення на екрані
5 Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ:

1 Який масив називають простим?
2 Який масив називають індексованим?
3 Як оголосити асоціативний масив?
4 Як отримати доступ до елемента індексованого масиву? Навести приклад.
5 Як отримати доступ до елемента асоціативного масиву? Навести приклад.
6 Як оголосити багатовимірний масив?
7 Як звернутися до елемента багатовимірного масиву?
8 Що називають ключем масиву?
9 Що називають індексом масиву?

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.

ЛАБОРАТОРНА РОБОТА №11 (2 год.)

ТЕМА. Взаємодія з базою MySQL за допомогою PHP

НАВЧАЛЬНА МЕТА: навчитися використовувати інструментарій PHP для роботи з MySQL.

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Команди MySQL

Команда	Дія
SHOW	Виведення елементів
ALTER	Внесення змін до бази даних або таблицю
BACKUP	Створення резервної копії таблиці
\c	Скасування введення
CREATE	Створення бази даних
DELETE	Видалення рядка з таблиці
DESCRIBE	Опис стовпців таблиць
DROP	Видалення бази даних або таблиці
EXIT (Ctrl+C)	Вихід
GRANT	Зміна привілеїв користувача
HELP (\h, \?)	Відображення підказки
INSERT	Вставка даних
LOCK	Блокування таблиці (таблиць)
QUIT (\q)	Те ж саме, що і EXIT
RENAME	Перейменування таблиці
SHOW	Список відомостей про об’єкти
TRUNCATE	Спустошення таблиці
UNLOCK	Розблокування таблиці (таблиць)
UPDATE	Оновлення існуючої записи
USE	Використання бази даних

Функції PHP для роботи з MySQL

mysql_connect, mysql_error, mysql_select_db, mysql_query, mysql_num_rows, mysql_fetch_row, mysql_result, mysql_close

Методи MySQLi

mysqli, connect_error, query, error, num_rows, data_seek, fetch_assoc, fetch_array

ХІД РОБОТИ

1 Підключитись до MySQL локального веб-серверу Denwer з командного рядка, використовуючи користувача root:

```
mysql -u root
```

2 Перевірити наявні бази даних:

```
SHOW databases
```

3 Створити базу даних publications:

```
CREATE DATABASE publications
```

4 Відкрити створену базу даних для змін:

```
USE publications
```

5 Створити нового користувача user з паролем mypasswd для хоста localhost та надати йому всі права на створену базу даних. Синтаксис команди:

GRANT ПРАВА ON база_даних.об’єкт TO 'ім’я_користувача@ім’я_хоста' IDENTIFIED BY 'пароль';

6 Підключитися до MySQL від імені створеного користувача та почати використовувати базу даних publications.

7 Створити таблицю classics:

CREATE TABLE classics (author VARCHAR(128), title VARCHAR(128), type VARCHAR(16), year CHAR(4)) ENGINE MyISAM;

8 Перевірити факт створення таблиці classics:

```
DESCRIBE classics
```

9 Заповнити таблицю даними:

INSERT INTO classics(author, title, type, year)
VALUES('Mark Twain','The Adventures of Tom Sawyer','Fiction','1876');
INSERT INTO classics(author, title, type, year)
VALUES('Jane Austen','Pride and Prejudice','Fiction','1811');
INSERT INTO classics(author, title, type, year)
VALUES('Charles Darwin','The Origin of Species','Non-Fiction','1856');
INSERT INTO classics(author, title, type, year)
VALUES('Charles Dickens','The Old Curiosity Shop','Fiction','1841');
INSERT INTO classics(author, title, type, year)
VALUES('William Shakespeare','Romeo and Juliet','Play','1594');

10 Змінити тип даних стовпця year:

```
ALTER TABLE classics MODIFY year SMALLINT
```

11 Додати до таблиці стовпець pages:

```
ALTER TABLE classics ADD pages SMALLINT UNSIGNED
```

12 Перейменувати стовпець type на category:

```
ALTER TABLE classics CHANGE type category VARCHAR(16)
```

13 Вилучити стовпець pages:

```
ALTER TABLE classics DROP pages
```

14 Додати індекси до таблиці classics:

```
ALTER TABLE classics ADD INDEX(author(20));
ALTER TABLE classics ADD INDEX(title(20));
ALTER TABLE classics ADD INDEX(category(4));
ALTER TABLE classics ADD INDEX(year);
```

15 Додати до таблиці стовпець isbn та заповнити його даними:

```
ALTER TABLE classics ADD isbn CHAR(13);
UPDATE classics SET isbn='9781598184891' WHERE year='1876'
UPDATE classics SET isbn='9780582506206' WHERE year='1811'
UPDATE classics SET isbn='9780517123201' WHERE year='1856'
UPDATE classics SET isbn='9780099533474' WHERE year='1841'
UPDATE classics SET isbn='9780192814968' WHERE year='1594'
```

16 Зробити isbn ключовим полем (додати первинний ключ) та переконатись у правильності результатів:

```
ALTER TABLE classics ADD PRIMARY KEY(isbn);
DESCRIBE classics;
```

17 Створити таблицю customers (покупці):

CREATE TABLE customers (name VARCHAR(128), isbn VARCHAR(13), PRIMARY KEY (isbn)) ENGINE MyISAM;

INSERT INTO customers(name,isbn) VALUES('Joe Bloggs','9780099533474');
INSERT INTO customers(name,isbn) VALUES('Mary Smith','9780582506206');
INSERT INTO customers(name,isbn) VALUES('Jack Wilson','9780517123201');

18 Виконати резервне копіювання бази даних у файл publications.sql, використовуючи утиліту mysqldump.exe.

19 Сформувати наступні запити до таблиці classics:

- вивести всі записи на екран, відсортувавши їх по полю title за спаданням;
- порахувати кількість рядків;
- вивести всіх авторів, виключаючи повторення;
- вивести інформацію про трьох авторів та назви їх творів, починаючи з третього запису;

- змінити автора Mark Twain на Mark Twain (Samuel Langhorne Clemens);
- визначити кількість видань по кожній категорії.

20 Визначити які книги були придбані покупцями (запит на об’єднання даних із двох таблиць).

21 Створити файл login.php, в якому описати параметри підключення до створеної бази даних.

22 Створити скрипт query.php для вибірки даних з таблиці classics створеної бази даних publications за допомогою функцій PHP для роботи з MySQL. Код потрібно супроводжувати коментарями.

23 Підключити login.php до query.php.Протестувати роботу query.php. Зробити висновки.

24 Створити скрипт query-mysqli.php для вибірки даних з таблиці classics створеної бази даних publications за допомогою за допомогою розширення PHP MySQLi. Код потрібно супроводжувати коментарями. В якості параметрів підключення використати login.php.

25 Протестувати роботу query-mysqli.php. Зробити висновки.

26 Показати роботу викладачу.

27 Оформити звіт. Захистити роботу.

ЗМІСТ ЗВІТУ

1 Тема
2 Мета
3 Код
4 Схематичне зображення на екрані
5 Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ:

1 Охарактеризувати процес використання MySQL за допомогою PHP.

2 Яка стандартна PHP-функція призначена для підключення до бази даних MySQL?

3 В якому випадку роботу функції mysql_result не можна визнати оптимальною?Який масив називають простим?

4 Яку команду потрібно використовувати для створення резервної копії бази даних publications в файлі publications.sql?

5 Для чого потрібна крапка з комою в запитах MySQL?

6 Які команди використовуються для перегляду доступних баз даних або таблиць?

7 Як на локальному хості створюється новий користувач MySQL з ім'ям newuser і паролем newpass, якого відкритий доступ до всього вмісту бази даних newdatabase?

8 Як переглянути структуру таблиці?

9 Для чого потрібен індекс в MySQL?

10 Обидва специфікатори, і SELECT DISTINCT, і GROUP BY, призводять до відображення тільки одного рядка для кожного значення в стовпці, навіть якщо таке значення мають кілька рядків. Яке основне розходження між SELECT DISTINCT і GROUP BY?

11 Як можна за допомогою інструкції SELECT ... WHERE повернути тільки ті рядки, в яких в якомусь місці стовпця author таблиці classics, міститься слово Langhorne?

12 Що повинно бути визначено в двох таблицях, щоб з'явилася можливість їх об'єднання?

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.

ЛАБОРАТОРНА РОБОТА №12 (2 год.)

ТЕМА. Обробка інформації в формах за допомогою PHP

НАВЧАЛЬНА МЕТА: навчитися використовувати функціонал PHP для обробки даних, що передаються за допомогою форм .

ОБЛАДНАННЯ ТА ОСНАЩЕННЯ: ПК, програмне забезпечення Notepad++, веб-браузер, зошит, ручка.

ОСНОВНІ ТЕОРЕТИЧНІ ПОЛОЖЕННЯ

Обробка даних форми:

- необхідно створити форму і усі елементи управління помістити у всередину пари тегів <form>…</form>.
- визначити в атрибуті action файл, який оброблятиме дані форми;
- визначити метод передачі даних в атрибуті method. Можливі варіанти GET і POST;
- у елементах управління заповнити атрибут name;
- створити файл php, який оброблятиме форму.

До змінних потрібно звертатися за шаблоном $_TYPE[“NAME”], при цьому TYPE змінювати на GET або POST, а NAME – на значення атрибута, вказаного у відповідному елементі керування.

Метод запиту GET

Формат запиту: GET сценарій?параметри HTTP/1.0

Приклад 1. Спосіб передачі змінних сценарію методом GET.

Ввести в рядку браузера:

http://web/script.php?name=Ivan&age=20

Можна використати форму:

```
<html><body>
<form action="script.php" method = “GET”>
Введіть ім’я: <input type=text name="name"><br>
Введіть вік: <input type=text name="age"><br>
<input type=submit value="GO!">
</form>
</body></html>
```

script.php:

```
<?php
        echo “Привіт $_GET[‘name’]. Вам $_GET[‘age’] років.”;
?>
```

Передані сценарію параметри відображаються в URL в адресному рядку браузера.

Метод запиту POST

Формат запроса: POST сценарий?параметры HTTP/1.0

Приклад 2. Спосіб передачі змінних сценарію методом POST.

Форма:

```
<html><body>
<form action="script.php" method = “POST”>
Введіть ім’я: <input type=text name="name"><br>
Введіть вік: <input type=text name="age"><br>
<input type=submit value="GO!">
</form>
</body></html>
```

script.php:

```
<?php
        echo “Привіт $_POST[‘name’]. Вам $_POST[‘age’] років.”;
```

```
?>
```

Метод POST використовується при передачі великих об'ємів даних, наприклад, при завантаженні файлів через WEB.

Передані сценарію параметри не відображаються в URL в адресному рядку браузера.

isset

(PHP3, PHP4, PHP5)

isset – визначає, чи встановлена змінна.

Повертає TRUE, якщо var існує; інакше FALSE. Якщо змінна була розустановлена/unset за допомогою функції unset(), вона не може бути isset().

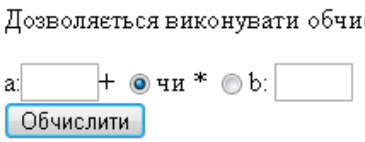
Приклад 2.

```
<?php
    $a = "test";
    $b = "anothertest";
    echo isset ($a); // TRUE
    echo isset ($a, $b); //TRUE
    unset ($a);
    echo isset ($a); // FALSE
    echo isset ($a, $b); //FALSE
    $foo = NULL;
    print isset ($foo); // FALSE
?>
```

ХІД РОБОТИ

1 Створити html-файл math_op.html який міститиме форму з двома полями для введення чисел і перемикач, який визначатиме, яку дію потрібно виконати з числами (просумувати або перемножити):

Вигляд у браузері



math_op.html

```
<!DOCTYPE html>
<html>
 <head>
  <title>form_math_op</title>
 </head>
 <body>
  <p>Дозволяється виконувати обчислення над цілими числами (до чотирьох знаків).</p>
  <form action="math_op.php" method="post">
    <p>a:<input type="text" name="a" size="4">+
    <input type="radio" name="act" value="plus" checked>чи *
    <input type="radio" name="act" value="myltiply">b:
    <input type="text" name="b" size="4"><br/>
    <input type="submit" value="Обчислити">
    </p>
  </form>
 </body>
</html>
```

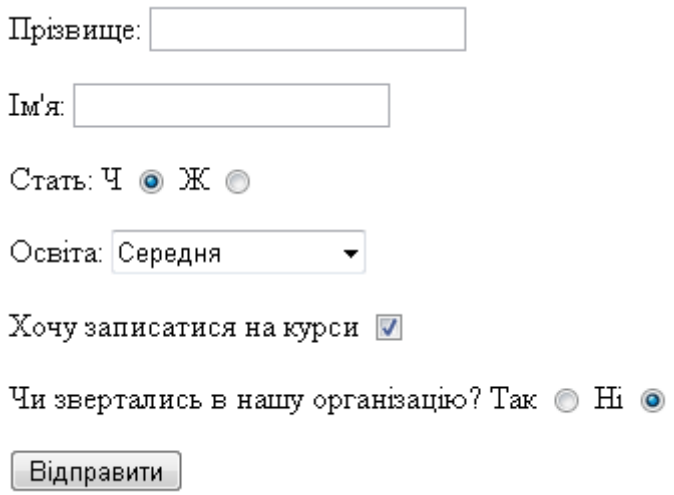
2 Зберегти файл на своєму сервері та перевірити його працездатність.

3 Створити php-файл math_op.php який оброблятиме інформацію введену у формі. Залежно від вибраного положення перемикача виконати відповідну дію. Додати в документ посилання для повернення на попередню сторінку:

math_op.php

```php
<?php
    printf("<!doctype html>
<head>
<meta charset=\"UTF-8\">
<title>lab12_math_op_rezult</title>
</head>
<body>");
/*Перевірка введених даних*/
    if (isset($_POST['a']) && preg_match('/^[0-9]{1,4}$/',$_POST['a'])){
        $a = $_POST['a'];
    } else {
        exit("Введені некоректні дані<br/><a href='lab12_math_op.html'>Перейти на попередню сторінку</a>");
    }
    if (isset($_POST['b']) && preg_match('/^[0-9]{1,4}$/',$_POST['b'])){
        $b = $_POST['b'];
    } else {
        exit("Введені некоректні дані<br/><a href='lab12_math_op.html'>Перейти на попередню сторінку</a>");
    }
/*Обчислення результатів*/
    if ($_POST['act']=='plus') {
        $c = $_POST['a'] + $_POST['b'];
        echo "Сума чисел $a та $b дорівнює $c";
    } else {
        $c = $_POST['a'] * $_POST['b'];
        echo "Добуток чисел $a та $b дорівнює $c";
    }
    echo "<br/><a href='lab12_math_op.html'>Перейти на попередню сторінку</a>";
    printf("</body>
</html>");
?>
```

4 Створити новий файл, в який включити форму, що містить наступні дані:



reg.html

```
<!DOCTYPE html>
<html>
 <head>
  <title>reg_form</title>
 </head>
 <body>
  <form name="reg" action="reg.php" method="post">
    <p>Прізвище:
    <input name="surname" type="text" size="20"></p>
    <p>Ім'я:
    <input name="name" type="text" size="20"></p>
    <p>Стать: Ч <input name="sex" type="radio" value="m" checked> Ж <input name="sex" type="radio"
value="v"></p>
    <p>Освіта:
    <select name="education">
      <option selected value="Середня">Середня</option>
      <option value="Неповна середня">Неповна середня</option>
      <option value="Вища">Вища</option>
      <option value="Неповна вища">Неповна вища</option>
```

```
    </select>
    <p>Хочу записатися на курси <input name="courses" value="courses" type="checkbox" checked></p>
    <p>Чи звертались в нашу організацію? Так <input name="first_time" type="radio" value="yes"> Ні <input
name="first_time" type="radio" value="no" checked></p>
    <p><input value="Відправити" type="submit"></p>
  </form>
 </body>
</html>
```

5 Створити reg.php, що динамічно формуватиме сторінку-результат, яку потрібно вивести у форматі: Шанований(а) Іванов Сергій. Ми раді вітати Вас на наших курсах. Сподіваємося на подальшу (продовження) співпрацю. В обробнику реалізувати перевірку існування глобальних змінних. Виведене повідомлення повинне формуватися в залежності від обраних перемикачів статі та курсів.

6 Показати роботу викладачу.

7 Оформити звіт. Захистити роботу.

ЗМІСТ ЗВІТУ

1 Тема
2 Мета
3 Код
4 Схематичне зображення на екрані
5 Висновки

КОНТРОЛЬНІ ЗАПИТАННЯ ДО ЗАХИСТУ:

1 В чому полягає спосіб передачі параметрів сценарію php методом POST?
2 В чому полягає спосіб передачі параметрів сценарію php методом GET?
3 Пояснити використання функції isset?
4 Як звертатися до змінних, переданих сценарію php методом POST?
5 Як звертатися до змінних, переданих сценарію php методом POST?

ПІДВЕДЕННЯ ПІДСУМКІВ І ОЦІНЮВАННЯ РІВНЯ ЗНАНЬ

Оцінювання рівня знань студентів проводиться за результатами захисту лабораторної роботи, який включає перевірку правильності отриманих результатів та усного опитування.