

Практичне заняття № 3

Тема заняття: Оформлення HTML документа. Карти. Створення форм

Зміст роботи:

Ввімкнути комп'ютер і запустити стандартну програму Блокнот;

Завдання1

Створіть html-файл, результат якого показаний на малюнку. Для цього додайте зображення. Далі вкажіть на зображенні посилальні області: скарб, акула, восьминіг, корабель. При наведенні курсору на посилальну область повинна з'являтися підказка. При натисканні на посилальну область повинно відкриватися зображення в окремому вікні.

Карта сокровищ



Завдання2

Створіть html-файл, в якому потрібно створити форму:

- 1) Додайте на сторінку порожню форму, потім задайте їй адресу відправки <https://echo.htmlacademy.ru> і метод відправки даних - post.
- 2) Додайте у форму поле вводу з підписом Ваш логін за допомогою тега `<input>`, задайте йому тип text і ім'я login.

Форма входу

Ваш логін

- 3) Оберніть підпис Ваш логін і текстове поле всередину тега label для зв'язку підпису та текстового поля
- 4) Додайте друге текстове поле з ім'ям password і ідентифікатором password-field після першого поля.

- 5) Додайте підпис Ваш пароль і зв'яжіть його з другим полем за допомогою атрибуту for.
- 6) Поміняйте у поля введення пароля тип поля на password.

Форма входу

Ваш логін

Ваш пароль

- 7) Додайте кнопку для відправки форми з написом Відправити, а потім надішліть форму.
- 8) Додайте у форму багаторядкове текстове поле з ім'ям comment і ідентифікатором comment-field. Задайте цьому полю висоту 10 рядків. І початкове значення Текст коментаря.

Введіть ваш коментарий:

Текст коментаря

Отправить

- 9) Додайте у форму чекбокс з ім'ям subscribe і ідентифікатором subscribe-field. Зробіть, щоб галочка в чекбоксі стояла за замовчуванням.

☒ Підписати Вас на розсилку?

- 10) Додайте у форму перемикач з ім'ям question-one. Задайте перемикачу значення *чол.*

Яка Ваша стать?

☒ чоловік

- 11) Додайте у форму другий перемикач з ім'ям question-one і значенням жін. Зробіть його обраним за замовчуванням.

Яка Ваша стать?

☐ чоловік

☒ жінка

- 12) Додайте у форму список, що розкривається з ім'ям year. Додайте в цей список варіант зі значенням 1998 і підписом 1998 рік. І другий варіант зі значенням 1999 та підписом 1999 рік.

Ваш рік народження:

1998

1998

1999

- 13) Зробіть список, що розкривається, мультіселектом. Збільште його розмір до 7. Зробіть, щоб за замовчуванням було відзначено неділю.

Ваші улюблені дні тижня:

Понеділок

Вівторок

Середа

Четвер

П'ятниця

Субота

Неділя

- 14) Задайте для форми атрибут enctype необхідний для завантаження файлів. Додайте поле для завантаження файлів з ім'ям avatar.

Виберіть зображення для свого аватара:

Виберіть файл

Файл не вибран

Завантажити

- 15) Додайте приховане поле з ім'ям account_number і значенням 15. Надішліть форму і подивіться, які дані були передані

Оплата послуг

Сума платежу грн.

Відправити

Завдання3

Створити 2 форми за зразком:

Регистрация

Ваш логин (email):

Пароль:

Информация о себе:

☒ Подписаться на рассылку

Профиль

Ваш пол: ☐ Мужской ☒ Женский

Месяц и год рождения:

Январь 1990

Знакомые технологии:

HTML
CSS
JavaScript
Node.js

Фото: No file selected

Методичні рекомендації по виконанню роботи:

1) **Зображення-карту** створюють у три етапи.

Перший етап - створення самого зображення з допомогою добре нам знайомого тега ``:

```
<IMG SRC = "map.gif">
```

Другий етап - створення карти, особливого елемента Web-сторінки, який описує набір "гарячих" областей зображення-карти. Сама карта на Web-сторінці ніяк не відображується.

Карту створюють за допомогою парного тега `<MAP>`:

```
<MAP NAME = "<ім'я карти>">
```

```
</ MAP>
```

Після створення карти слід прив'язати її до створеного на першому етапі зображенню. Для цього ми застосуємо обов'язковий в даному випадку атрибут `USEMAP` тега ``. Його значення - ім'я,що прив'язане до зображення карти, причому на початку цього імені обов'язково слід поставити символ `#` ("решітка"). (В імені, заданого атрибутом `NAME` тега `<MAP>`, символ `#` присутнім не повинен.)

```
<IMG SRC = "map.gif" USEMAP = "#samplemap">
```

На третьому етапі створюють опису самих "гарячих" областей в карті. Їх поміщують всередину відповідного тега `<MAP>` і формують за допомогою одинарних тегів `<AREA>`:

```
<AREA [SHAPE = "rect | circle | poly"] COORDS = "<набір параметрів>" HREF = "<інтернет-адресу гіперпосилання>" | NOHREF TARGET = "<мета гіперпосилання>">
```

Атрибути тега `<area>`

shape	Обриси області. Можливі значення атрибуту: rect - прямокутник circle - круг poly - багатокутник Необов'язковий атрибут. Якщо атрибут <code>SHAPE</code> відсутній, створюється саме прямокутна область
--------------	---

coords	Координати області. Для прямокутника - координати лівого верхнього і правого нижнього кутів. Для кола - координати центру і радіус. Для багатокутника - координати всіх кутів.
href	Адреса файлу, на який робиться посилання.
nohref	Означає, що ця область не діє. Зручний для вирізання дірок.
alt	Текстовий опис області. Служить альтернативою при відключеній в браузері завантаженні зображень.
title	Спливаючий текст, що з'являється при наведенні курсору мишки на задану область.

Приклад:

```
<IMG SRC="map.gif" USEMAP="#samplemap">
```

...

```
<MAP NAME="samplemap">
```

```
<AREA SHAPE="circle" COORDS="50,50,30" HREF="page1.html">
```

```
<AREA SHAPE="circle" COORDS="50,150,30" HREF="page2.html">
```

```
<AREA SHAPE="poly" COORDS="100,50,100,100,150,50,100,50" NOHREF>
```

```
<AREA SHAPE="rect" COORDS="0,100,30,100" HREF="appendix.html"
```

```
TARGET="_blank">
```

```
</MAP>
```

Тут ми створили дві круглі "гарячі" області, що вказують на Web-сторінки page1.html і page2.html, багатокутну область, не посилається нікуди, і прямокутну область, що посилається на Web-сторінку appendix.html. причому остання "гаряча" область при натисканні на ній відкриє Web-сторінку в новому вікні Web-оглядача.

2) Створення форм

Щоб створити форму, потрібно використовувати парний тег <form>, усередині якого розміщуються поля форми. У тега <form> є два важливі атрибути:

- **action** задає адреса, URL, відправки форми;
- **method** задає метод надсилання форми.

Приклад:

```
<form action = "https://echo.htmlacademy.ru" method = "get">
```

```
  поля форми
```

```
</form>
```

Для відправки форми зазвичай використовують методи **get** або **post**. Якщо не вказати атрибут **method**, то буде використаний **get**.

Метод **get** посилає дані форми в рядку запиту, тобто вони видні в адресному рядку браузера і слідують після знака питання.

Наприклад:

<https://www.google.com/search?q=htmlacademy>

Метод **get** краще використовувати в пошукових формах, тому що він дозволяє отримати посилання на результати пошуку і передати її комусь.

Метод **post** посилає дані в тілі HTTP-запиту і використовується, коли потрібно відправити багато даних і посилання на результат обробки цих даних не потрібна. Наприклад, при редагуванні особистого профілю.

Текстове поле введення

Більшість полів форм створюється за допомогою одиночного тега **input**. У цього тега два обов'язкові атрибути:

- **type** задає тип поля;
- **name** задає ім'я поля.

Тип поля впливає на те, як воно буде відображатися і поводитися. Найпоширеніший тип - це **text**, який позначає текстове поле. Він же використовується за умовчанням.

Приклад:

```
<form action = "https://echo.htmlacademy.ru" method = "get">  
  <input type = "text" name = "search">  
</form>
```

Ім'я поля потрібно, щоб правильно обробити дані на сервері. Зазвичай, ім'я поля має бути унікальним в межах форми, хоча є винятки. Для завдання імені поля використовують латинські букви і цифри.

Ідентифікатор і значення за замовчуванням

Атрибут **id** поля введення позначає ідентифікатор. Він повинен бути унікальним не тільки в межах форми, а й на всій сторінці.

Зазвичай ідентифікатори використовують для підвищення зручності роботи з формою, наприклад, створюють підписи, пов'язані з дрібними полями. При клацанні по таким підписам активується зв'язане поле. І це зручно, так як по великій підписи потрапити легше, ніж по маленькому полю. Також ідентифікатори використовують в JavaScript для роботи з полями.

Ідентифікатор на відміну від імені поля не передається на сервер. Краще використовувати ідентифікатори, що відрізняються від імені поля, особливо актуально це для полів множинного вибору. Атрибут **value** задає значення поля вводу за замовчуванням. Це теж підвищує зручність.

Підпис для поля введення

Можна подумати, що зробити підпис до поля дуже просто. Пишемо текст поряд з полем і все готово:

Підпис `<input type = "text" name = "username">`

Насправді цього недостатньо - ми отримали просто шматок тексту і поле, які розташовані поруч один з одним, але логічно ніяк не пов'язані.

Є спеціальний тег, який дозволяє сміливо сказати: «Цей шматок тексту дійсно підпис до цього поля!». Це парний тег **<label>**.

Він пов'язує текст і поле введення логічно. А ще якщо клацнути по тексту в такого підпису, то курсор переміститься у відповідне поле.

Створювати підписи до полів за допомогою **<label>** - хороший прийом. Використовуйте його.

Перший спосіб створити підпис - просто обернути текст підпису та тег поля в тег **<label>**, ось так:

`<label>`

```
Підпис <input type = "text" name = "username">
</ label>
```

Треба відзначити, що при Огортання тексту в тег <label> він візуально ніяк не змінюється, адже головне завдання підпису - створити логічний зв'язок

Зв'яжемо підпис і поле по id

Іноді обернути поле і текст підпису в тег <label> не можна. Наприклад, коли вони розміщені в різних осередках таблиці.

У цьому випадку можна зв'язати підпис з полем за допомогою атрибуту **id**. Алгоритм такий:

1. Додаємо до поля введення ідентифікатор за допомогою атрибуту **id**.
2. Обертаємо текст підпису в тег <label>.
3. Додаємо тегу <label> атрибут **for**.
4. У атрибут **for** записуємо таке ж значення, що і в атрибуті **id** у поля.

наприклад:

```
<label for = "user-field-id"> Ім'я користувача </ label>
```

...

багато-багато інших мегів

...

```
<input id = "user-field-id" type = "text" name = "username">
```

Поле для введення пароля

Щоб зробити поле справжнім полем для введення пароля, в якому текст буде відображатися «зірочками», потрібно просто змінити значення атрибута **type** на **password**.

Кнопка відправки форми

Така кнопка створюється за допомогою тега <input> з типом **submit**.

Напис на кнопці можна задати за допомогою атрибуту **value**. Для кнопки відправки форми задавати ім'я необов'язково. Але якщо ім'я задано, то на сервер будуть відправлятися ім'я і значення кнопки.

Багаторядкове поле введення

Багаторядкове текстове поле створюється за допомогою парного тега <textarea>. У нього є атрибути **name** і **id**, які аналогічні атрибутам текстового поля.

Атрибут **rows** приймає цілочисельне значення і задає висоту багаторядкового поля в рядках. Атрибут **cols** задає ширину поля в символах. В якості ширини символу береться деяка «усереднена ширина».

Атрибут **value** у багаторядкового поля відсутня, а значення за замовчуванням задається по-іншому. Текст, розташований між відкриває і закриває тегом <textarea> і є значенням за замовчуванням. Ось так:

```
<textarea> Значення за замовчуванням </ textarea>
```

Чекбокс або «галочка»

Поле-галочка - це тег <input> з типом **checkbox**.

Галочка працює за принципом «або так, або ні». Якщо галочка стоїть, то браузер посилає змінну з ім'ям поля на сервер, якщо галочки ні, то не надсилається нічого. Таким чином, атрибут **value** не є обов'язковим.

Щоб галочка стояла за замовчуванням, потрібно додати до тегу атрибут **checked**. Ось так:

```
<input type = "checkbox" checked>
```

Чекбокс не має на увазі вибір одного елемента з декількох. Тому якщо в одній формі є кілька чекбоксів, то імена у них повинні бути різними.

Перемикач або «радіобаттон»

Поле-перемикач - це тег **<input>** з типом **radio**.

Зазвичай перемикачі розміщують групами по кілька штук. Причому у перемикачів з однієї групи має бути однакове ім'я і різні значення, які задаються за допомогою **value**.

Таким чином, атрибут **value** є для перемикачів обов'язковим. Браузер відправляє на сервер значення **value** вибраного перемикача.

Група перемикачів

Тепер додамо ще один варіант відповіді в наш перемикач. Для цього потрібно додати ще один **<input>** з таким же ім'ям, але іншим значенням **value**.

Подібним чином можна створювати групи перемикачів з будь-якою кількістю варіантів.

Щоб зробити який-небудь варіант в перемикачі обраним за замовчуванням, потрібно додати до відповідного тегу **<input>** атрибут **checked**, як у поля-галочки.

До речі, ім'я поля у перемикачів однієї групи має бути однаковим, але ідентифікатори завжди повинні бути унікальними.

Список, що розкривається або «селект»

Список, що розкривається так само, як і перемикач, дозволяє вибрати один варіант відповіді з декількох.

Список, що розкривається створюється за допомогою парного тега **<select>**, у якого є знайомі атрибути **name** і **id**.

Варіанти відповідей задаються за допомогою парних тегів **<option>**, які повинні розташовуватися всередині тега **<select>**.

Наприклад:

```
<select name = "theme">
  <option value = "light"> Світла тема </ option>
  <option value = "dark"> Темна тема </ option>
  ...
</ select>
```

В атрибуті **value** тега **<option>** задається значення варіанта відповіді, а всередині цього тега розташовується підпис варіанти відповіді.

Якщо при відправленні форми у обраного варіанту заданий **value**, то на сервер відправиться значення цього атрибута. В іншому випадку буде відправлений текст підпису.

«Мультиселект»

Список, що розкривається можна перетворити на так званий «мультиселект», тобто список, в якому можна вибрати не один, а кілька варіантів.

Щоб зробити це, потрібно додати до тега **<select>** атрибут **multiple**.

Вибрати декілька варіантів можна, клацаючи по ним з затиснутою клавішею **Ctrl** на Windows або **Command** на OS X.

Висоту мультіселекта можна змінювати за допомогою атрибуту **size** тега **<select>**.

Щоб відзначити як вибрані за замовчуванням одне або кілька значень, потрібно до відповідних тегам **<option>** додати атрибут **selected**.

Поле для завантаження файлів

Поле для завантаження файлів - це тег **<input>** з типом **file**. Для цього поля обов'язковим атрибутом є ім'я.

Щоб поле запрацювало і браузер зміг передати вибраний файл на сервер, необхідно додати формі атрибут **enctype** зі значенням **multipart/form-data**. Чи не полю, а формі.

До речі, зовнішній вигляд таких полів дуже сильно відрізняється в залежності від операційної системи і дуже погано змінюється за допомогою стилів.

Приховане поле

І ще одне невидиме і дуже корисне поле. Це приховане поле. Його використовують, коли у формі потрібно відправити якісь додаткові службові дані, які не вводяться користувачем.

Наприклад, це можуть бути реквізити замовлення або номер користувача у формі оплати.

Приховане поле - це тег **<input>** з типом **hidden**.