

# Internet- технології

Мова розмітки HTML.  
Тіло документу.  
Форми та керуючі елементи

# Форми та керуючі елементи

В інтерфейсі користувача вебсторінки можна умовно виділити елементи прості (щось представляють чи позначають) та **інтерактивні** (дозволяють із ними взаємодіяти). Зазвичай взаємодія реалізується за допомогою **форм**.

Форма – це інструмент, за допомогою якого HTML-документ може надіслати **обробнику** деяку інформацію (у тому числі і введену користувачем) у вигляді пар «ім'я=значення».

Форма характеризується адресою обробника (адреса програми на веб-сервері, яка буде обробляти вміст даних форми) і включає набір **керуючих елементів** форми, які являють собою стандартні поля для введення інформації, кнопки, перемикачі та інші. Зокрема, кнопка надсилання даних форми на сервер (кнопка submit).

# Форми. Приклад форми авторизації

**<form>**

**<label** for="username">Username:</label>

**<br>**

**<input** type="text" id="username" name="username">

**<br><br>**

**<label** for="password">Password:</label>

**<br>**

**<input** type="password" id="password" name="password"><br><br>

**<input** type="submit" value="Submit">

**</form>**



Username:

Password:

Submit

# Форми. Методи відправки даних

Встановлює форму на веб-сторінці тег `form`. Серед його атрибутів слід зазначити `action`, який вказує, куди відправляти дані, і `method` – який вказує, яким чином це зробити – тобто якого типу буде HTTP-запит браузера до обробника.

Значення атрибуту `method`:

**get** – передача відбувається за допомогою гет-запиту браузера, у такого запиту немає тіла, тож усі дані передаються в адресному рядку самого запита. Пари «ім'я=значення» приєднуються в цьому випадку до адреси обробника після знаку питання і поділяються між собою амперсандом (символ `&`).

`handler.php?username=SomeName&password=qwerty`

**post** – на сервер параметри запиту відправляються у тілі пост-запиту браузера. Це дозволяє надсилати більше даних, ніж доступно методу `get`, і без явного відображення їх користувачеві браузера. Адреса у такому випадку не змінюється (не додаються параметри запиту).



# Форми. Поширені керуючі елементи

Тег	Опис
<b>&lt;button&gt;</b>	Створює кнопку на веб-сторінці.
<b>&lt;input&gt;</b>	Дозволяє створювати різні елементи інтерфейсу.
<b>&lt;textarea&gt;</b>	Створює поле для багаторядкового тексту.
<b>&lt;fieldset&gt;</b>	Призначений для групування елементів форми.
<b>&lt;label&gt;</b>	Встановлює мітку для елемента форми. Рекомендується для кожного елемента <b>&lt;input&gt;</b> встановлювати відповідний <b>&lt;label&gt;</b>
<b>&lt;details&gt;</b>	Елемент, що розкривається.
<b>&lt;select&gt;</b>	Створює розкривний список.
<b>&lt;option&gt;</b>	Визначає окремі пункти списку, створюваного за допомогою контейнера <b>select</b> .
<b>&lt;optgroup&gt;</b>	Групує елементи списку.

## Керуючі елементи. `<input>`


Переважає більшість керуючих елементів форми (кнопки, перемикачі, текстові поля та інші) створюється за допомогою тегу `<input>`.

Елемент `<input>` не обов'язково розміщувати саме всередині контейнеру `<form>`, що визначає форму, але якщо введені користувачем дані повинні бути відправлені на сервер, де їх обробляє серверний скрипт-обробник, то керуючий елемент повинен відноситися до форми.

Основний атрибут тегу `<input>`, що визначає вид елемента – `type`. Він дозволяє задавати такі елементи форми, як текстове поле (`text`), поле з паролем (`password`), перемикач (`radio`), прапорець (`checkbox`), приховане поле (`hidden`), кнопка (`button`), кнопка для надсилання форми (`submit`), кнопка для очищення форми (`reset`), поле для відправлення файлу (`file`) та інші. Для кожного елемента існує свій список атрибутів, які визначають його вигляд та характеристики.

# Керуючі елементи. <input>. Поширені типи

Тип	Описание
text	Звичайне поле для введення.
email	Для введення адреси електронної пошти (є вимоги щодо шаблону).
number	Введення числового значення.
range	Повзунок для вибору значення у певному діапазоні.
password	Поле для прихованого введення.
submit	Кнопка для відправки даних форми.
checkbox	Поле для галочки.
radio	Можливість вибору серед значень.
date	Поле для вибору дати календаря.
time	Поле вибору часу.
color	Елемент для вказівки кольору.

Text	<input type="text" value="Firstname"/>
Email	<input type="email" value="Email Address"/>
Number	<input type="number" value="Amc"/>
Range	<input type="range" value=""/>
Password	<input type="password" value="Password"/>
Submit	<input type="submit" value="Send form"/>
Checkbox	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Radio	<input type="radio"/> <input type="radio"/> <input type="radio"/>
Date	<input type="text" value="ДД . ММ . ГГГГ"/> 
Time	<input type="text" value="-- : --"/>
Color	<input type="color"/>



## Керуючі елементи. <select>

<select> дозволяє створити елемент інтерфейсу у вигляді списку, що розкривається, а також список з одним або множинним вибором. Кожен пункт створюється за допомогою елемента <option>, який має бути вкладений у контейнер <select>. Якщо планується надсилати дані списку на сервер, потрібно помістити <select> всередину форми. Ширина списку визначається найширшим текстом, вказаним у елементі <option>, і може змінюватися з допомогою стилів.

```
<select name="ComOS">
```

```
  <optgroup label="PortMaster 2">
```

```
    <option value="pm2_3.7">PortMaster 2 и ComOS 3.7</option>
```

```
    <option value="pm2_3.5">PortMaster 2 и ComOS 3.5</option>
```

```
  </optgroup>
```

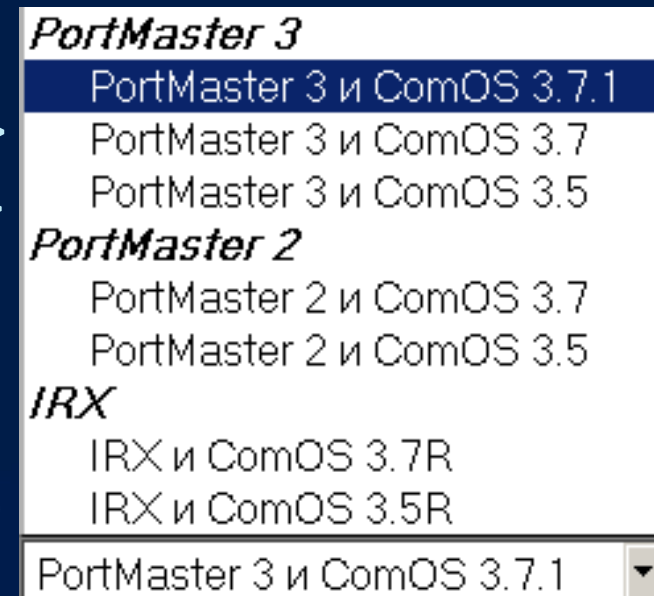
```
  <optgroup label="IRX">
```

```
    <option value="IRX_3.7R">IRX и ComOS 3.7R</option>
```

```
    <option value="IRX_3.5R">IRX и ComOS 3.5R</option>
```

```
  </optgroup>
```

```
</select>
```





# Керуючі елементи. <details>

Тег details визначає елемент, що розкривається, з додатковою інформацією. Він часто використовується для створення інтерактивних елементів, які можна відкривати та закривати без використання JavaScript. У відкритому стані він розширюється та показує прихований контент. За замовчуванням details знаходиться у закритому стані. Всередині details можуть розміщуватися будь-які теги. Для відображення заголовка details використовується тег summary.

Індикатором відкритого стану details є атрибут open (цей атрибут може використовуватися як CSS-селектор, details[open] або JavaScript-селектор, document.querySelector('[open])).

<details>

    <summary>Заголовок - видима частина елемента "details"</summary>

    <p>Прихований контент – додаткова інформація</p>

</details>

► Заголовок - видима частина елемента "details"

▼ Заголовок - видима частина елемента "details"

Прихований контент – додаткова інформація

# Контрольні запитання

- Для чого використовуються форми?
- Які існують методи відправки даних формою?
- Який метод відправки рекомендується використовувати при передачі великих обсягів даних?
- Назвіть основні керуючі елементи.
- Відмінності використання елементів, створених за допомогою `<input type="button">` та `<button>`.
- Для чого використовується елемент `<label>`? Які його обов'язкові атрибути?
- Якими тегами можна створити елемент для введення тексту?
- Для чого використовується елемент, що створений за допомогою `<input type="hidden">`?
- Як створити елемент для вказівки дати?