

Temat 18

Temat: Formularze

Formularze i elementy sterujące, takie jak np. `<input>` mają wiele specjalnych właściwości i zdarzeń.

Większość tych właściwości przydaje się w specyficznych sytuacjach. Najczęściej używaną właściwością jest **value**, która zawiera wartość danego pola. Wartość value zwraca dynamiczną wartość, jaka jest wpisywana do danego pola.

Do elementów formularza możemy dostać się na kilka różnych sposobów.

Pola typu text, button, number...

Korzystamy ze znanych narzędzi dostępu do elementów drzewa DOM oraz z właściwości value pól.

Przykład 1a Przygotuj stronę html wykorzystując poniższy kod. Zapisz przykład jako T18p1a.html

```
<body>
  <form action='' method='POST'>
    <input type="text" name="inputText" id="inputText" placeholder="Pole testowe">
    <input type="button" name="inputButton" id="inputButton" value="Wypisz">
  </form>
  <p id="output"></p>
  <script>
    const inputT = document.querySelector("#inputText");
    const button = document.querySelector("#inputButton");
    const output = document.querySelector("#output");

    button.addEventListener("click", () => {
      console.log(inputT.value);
      output.innerText=inputT.value;
    })
  </script>
</body>
```

Przykład 1b Przygotuj stronę html wykorzystując poniższy kod. Zapisz przykład jako T18p1b.html

```
<body>
  <form action='' method='POST'>
    <input type="text" name="inputText" id="inputText" placeholder="Pole testowe">
    <input type="button" name="inputButton" id="inputButton" value="Wypisz">
  </form>
  <p id="output"></p>
  <script>
    const input = document.querySelectorAll("input");
    const output = document.querySelector("#output");

    input[1].addEventListener("click", () => {
      console.log(input[0].value);
      console.log(input[1].value);
      output.innerText=`Pole tekstowe: ${input[0].value} , Przycisk: ${input[1].value}`;
    })
  </script>
</body>
```

Przykład 1c Przygotuj stronę html wykorzystując poniższy kod. Zapisz przykład jako T18p1c.html

```
<body>
  <form action='' method='POST'>
    <input type="text" name="inputText" id="inputText" placeholder="Pole testowe">
    <input type="button" name="inputButton" id="inputButton" value="Wypisz">
  </form>
  <p id="output"></p>
  <script>
    const input = document.getElementsByTagName("input");
    const output = document.getElementById("output");

    input[1].addEventListener("click", () => {
      console.log(input[0].value);
      console.log(input[1].value);
      output.innerText=`Pole tekstowe: ${input[0].value} , Przycisk: ${input[1].value}`;
    })
  </script>
</body>
```

Kolekcja `document.forms`

Formularze dokumentów są elementami specjalnej kolekcji `document.forms`.

To tak zwana „kolekcja nazwana”: jest zarówno nazwana, jak i uporządkowana. W celu uzyskania dostępu do formularza możemy użyć zarówno nazwy (atrybutu „name”, jak i indeksu elementu w dokumencie.

Właściwości `name` każdego elementu powinny być nadane, aby prawidłowo przesłać formularz na serwer, stąd możliwość dotarcia do nich poprzez wykorzystanie tej właściwości.

Przykład 2 Przygotuj stronę html wykorzystując poniższy kod. Zapisz przykład jako T18p2.html

```
<body>
  <form action='' method='POST' name="myForms">
    <input type="text" name="inputText" placeholder="Pole testowe">
    <input type="button" name="inputButton" value="Wypisz">
  </form>
  <form action='' method='POST' name="myForms2">
    <input type="number" name="inputNumber" placeholder="Wybierz liczbę">
    <input type="button" name="inputButton2" value="Wyświetl">
  </form>
  <p></p>
  <p></p>
  <script>
    /*const input = document.getElementsByTagName("input");*/
    const output = document.querySelectorAll("p");
    /* dostęp do formularza:*/
    console.log(document.forms.myForms);
    console.log(document.forms.myForms2);
    console.dir(document.forms.myForms);
    console.log(document.forms[0]);
    console.log(document.forms[1]);
```

Do poszczególnych elementów formularza możemy dostać się z wykorzystaniem kolekcji: `form.elements` lub atrybutu „name” poszczególnych elementów:

```
/*Pobieranie elementu z formularza z wykorzystaniem elements:*/
console.log(document.forms[0].elements[0]);
console.log(document.forms[0].elements[1].value);

/*Pobieranie elementu z formularza z wykorzystaniem name:*/
console.log(document.forms.myForms.inputText.value);
const inputT = document.forms.myForms.inputText;
const button1 = document.forms.myForms.inputButton;

button1.addEventListener("click", () => {
    console.log(inputT);
    const out = inputT.value;
    output[0].innerText=`Pole tekstowe: ${out} `;
})

/* drugi formularz*/
const inputN = document.forms.myForms2.inputNumber;
const button2 = document.forms.myForms2.inputButton2;
console.log(button2.value);

const show =() => {
    console.log(inputN);
    const outN = inputN.value;
    output[1].innerText=`Pole liczbowe: ${outN} `;
}
button2.addEventListener("click", show);

</script>
```

Wszystkie dane z pól formularza są tekstem.

Podczas pracy z liczbami warto pamiętać o zmianie typu otrzymanych z formularza danych.

Przykład 3. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako T18p3.html

```
<body>
  <form action='' method='GET'>
    <fieldset>
      <legend>Kalkulator</legend>
      <label>a:
        <input type='number' name='numberA' value=''>
      </label>
      <label>b:
        <input type='number' name='numberB' value=''>
      </label>
      <input type='button' value="Oblicz">
    </fieldset>
  </form>
  <p id="output"></p>

  <script>
    const input = document.querySelectorAll("input");
    const output = document.querySelector("#output");

    input[2].addEventListener("click", () => {
      console.log(input[0].value);
      console.log(input[1].value);
      let a = parseFloat(input[0].value);
      let b = parseFloat(input[1].value);
      let sum = a+b;
      output.innerText=`suma liczb: ${a} + ${b} = ${sum}`;
    })
  </script>
</body>
```

lub

```
<body>
  <form action='' method='GET'>
    <fieldset>
      <legend>Kalkulator</legend>
      <label>a:
        <input type='number' name='numberA' value=''>
      </label>
      <label>b:
        <input type='number' name='numberB' value=''>
      </label>
      <input type='button' value="Oblicz">
    </fieldset>
  </form>
  <p id="output"></p>
  <script>
    const inputN = document.querySelectorAll("input[type='number']");
    const btn = document.querySelector("input[type='button']");
    const output = document.querySelector("#output");

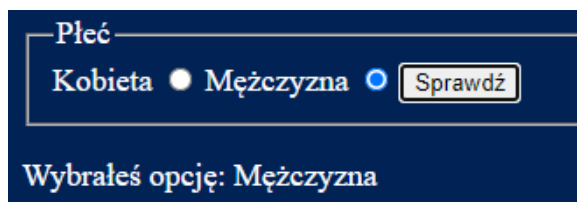
    btn.addEventListener("click", () => {
      let a = parseFloat(inputN[0].value);
      let b = parseFloat(inputN[1].value);
      let sum = a+b;
      output.innerText=`suma liczb: ${a} + ${b} = ${sum}`;
    })
  </script>
</body>
```

Pola typu RADIO

Do pól formularza typu radio odwołujemy się poprzez wartość checked oraz value.

Przykład 4. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako T18p4.html

Strona internetowa zawierająca przyciski typu radio do zaznaczenia jednej z dwóch opcji (płeć – kobieta albo mężczyzna). Po kliknięciu przycisku Sprawdz na stronie pojawia się komunikat informujący o wyborze użytkownika.



The screenshot shows a web form with a title "Płeć" inside a fieldset. There are two radio buttons: "Kobieta" (unselected) and "Mężczyzna" (selected). To the right is a button labeled "Sprawdz". Below the fieldset, a message states "Wybrałeś opcję: Mężczyzna".

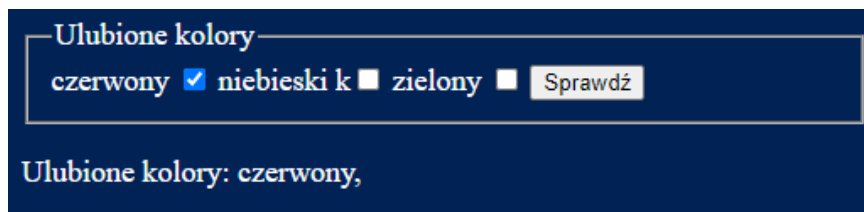
```
<form action='' method='GET'>
  <fieldset>
    <legend>Płeć</legend>
    <label>Kobieta
      <input type='radio' name='gender' value='Kobieta'>
    </label>
    <label>Mężczyzna
      <input type='radio' name='gender' value='Mężczyzna'>
    </label>
    <input type="button" value="Sprawdz">
  </fieldset>
</form>
<p id="output"></p>
<script>
  const input = document.querySelectorAll("input[type='radio']");
  const output = document.querySelector("#output");
  const btn = document.querySelector("input[type='button']");

  btn.addEventListener("click", () => {
    let valueR;
    for(let i = 0; i < input.length; i++){
      if(input[i].checked){
        valueR = input[i].value;
      }
    }
    output.innerText=`Wybrałeś opcję: ${valueR}`;
  })
</script>
```

Pola typu CHECKBOX

Przykład 5. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako T18p5.html

Strona internetowa zawierająca 3 pola wyboru koloru. Użytkownik może zaznaczyć dowolną ilość kolorów. Po kliknięciu przycisku wyświetlane jest podsumowanie.



```
<form action='' method='post'>
  <fieldset>
    <legend>Ulubione kolory</legend>
    <label>czerwony
      <input type='checkbox' name='color' value='czerwony'>
    </label>
    <label>niebieski
      <input type='checkbox' name='color' value='niebieski'>
    </label>
    <label>zielony
      <input type='checkbox' name='color' value='zielony'>
    </label>
    <input type="button" value="Sprawdź">
  </fieldset>
</form>
<p id="output"></p>

<script>
  const check = document.querySelectorAll("input[name='color']");
  const output = document.querySelector("#output");
  const btn = document.querySelector("input[type='button']");

  btn.addEventListener("click", () => {
    let valueC="";
    for(let i = 0; i < check.length; i++){
      if(check[i].checked){
        valueC += `${check[i].value}, `;
      }
    }
    output.innerText=`Ulubione kolory: ${valueC}`;
  })
</script>
```

Uwagi:

Wstawiając pola wyboru (ang. checkbox) używamy znaczników input z atrybutem type ustawionym na "checkbox". Aby sprawdzić, czy pole jest zaznaczone używamy właściwości checked obiektu, która ma wartość true jeśli pole jest zaznaczone, a wartość false, gdy jest odznaczone.

Pola typu SELECT

Do wstawienia pola kombi (ang. combobox), czyli listy rozwijanej używamy znacznika `select`, a elementy listy umieszczamy w znacznikach `option`. Atrybut `name` jest przypisany do znacznika `select`, a wartości atrybutu `value` są przypisane do poszczególnych elementów listy (znacznik `option`). Właściwość **`selectedIndex`** przechowuje indeks wybranej opcji.

Przykład 6. Przygotuj stronę html wykorzystującą poniższy kod. Zapisz przykład jako [T18p6.html](#)

Strona internetowa zawierająca listę rozwijaną. Po kliknięciu przycisku pojawia się komunikat o wybranej opcji.



```
<form action='' method='post'>
  <fieldset>
    <legend>Ulubione kolory</legend>
    <select name='color'>
      <option value='czerwony'>czerwony</option>
      <option value='niebieski'>niebieski</option>
      <option value='zielony'>zielony</option>
      <option value='fioletowy'>fioletowy</option>
    </select>
    <input type="button" value="Sprawdź">
  </fieldset>
</form>
<p id="output"></p>
```

```
<script>
  const selectC = document.querySelector("select[name='color']");
  const output = document.querySelector("#output");
  const btn = document.querySelector("input[type='button']");

  /* selectedIndex zwraca numer wybranego pola listy*/
  btn.addEventListener("click", () => {
    let valueC = selectC.value;
    let sIndex = selectC.selectedIndex
    output.innerHTML=`Ulubiony kolor: ${valueC}
    to pole o indeksie: ${sIndex}`;
  })
</script>
```

Pola typu TEXTAREA

Obszar tekstowy (ang. textarea) wstawiamy używając znacznika `textarea`. Umożliwia on wprowadzenie wielu linii tekstu. Obszar tekstowy często w formularzach używa się z polem wyboru, którego zaznaczenie powoduje wyświetlenie obszaru tekstowego, a odznaczenie ukrywa go.

Przykład 7. Przygotuj stronę html wykorzystując poniższy kod. Zapisz przykład jako T18p7.html



```
<form action='' method='post'>
  <fieldset>
    <legend>Formularz</legend>
    <p>Dodatkowe informacje: </p>
    <p>
      <label>
        <input type='checkbox' name='showText' value="pokaż">Pokaż/ukryj
      </label>
      <textarea rows="3" cols="50" name="longText" placeholder="Dodatkowe informacje:"></textarea>
    </p>
    <input type="button" value="Wyślij informacje">
  </fieldset>
</form>
<p>Podsumowanie:</p>
<p id="output"></p>
<script>
  const check = document.querySelector("input[type='checkbox']");
  const textA = document.querySelector("textarea");
  const btn = document.querySelector("input[type='button']");
  const output = document.querySelector("#output");

  check.addEventListener("click", () => {
    if(check.checked){
      textA.classList.add('show');
    }else{
      textA.classList.remove('show');
    }
  })
  btn.addEventListener("click", () => {
    if(check.checked){
      output.innerText=`Twoja wiadomość: ${textA.value}`;
    }
  })
</script>
```