

## Tworzenie stron www instrukcja 04

---

### Formularze

Formularz jest używany w bardzo wielu przypadkach. Za jego pomocą zdobywamy informacje o odwiedzających stronę, tworzymy ankiety, książki gości, a nawet zgłaszamy zapytanie do internetowych serwisów katalogujących czy dokonujemy zakupów w sieciowych centrach handlowych.

Język HTML posiada zestaw znaczników odpowiedzialnych za generowanie elektronicznych formularzy. Dodatkowo HTML 5 wprowadza kilka bardzo przydatnych elementów, które znacznie upraszczają budowę własnych formularzy. *Jedną z najważniejszych i długo oczekiwanych nowości jest funkcja sprawdzania zawartości danych wprowadzonych do formularza.*

### Ramy formularza

Ramy formularzy na stronie definiuje znacznik `<form></form>`.

Pierwszym atrybutem tego znacznika jest `action=" "`, który określa rodzaj akcji, jaka ma zostać wykonana po naciśnięciu przycisku OK. Jako wartość `action=" "` można podać adres odpowiedniego skryptu bądź adres e-mail. Dalej mamy atrybut `method=" "`, gdzie określa się metodę, która zostanie użyta do wykonania formularza. Do wyboru mamy dwie metody: `post` i `get`. Pierwsza z nich, `post`, służy do zgłaszania długich żądań do serwera, natomiast `get` zgłasza krótkie żądania. Jest to dość ogólne wyjaśnienie, ale uznałem, że zupełnie wystarczy dla Twoich potrzeb. Jeśli zajmiesz się pisanem skryptów, wtedy będziesz musiał poszerzyć wiadomości z tego zakresu.

Pora na przykład definicji ram formularza:

```
<form action="mailto:email@domena.pl" method="post">  
...  
</form>
```

Tak ustawiony formularz wyśle na adres e-mail zawartość wszystkich pól, jakie wstawimy w dalszej części niniejszej publikacji.

Każda współczesna przeglądarka posiada mechanizmy pozwalające na automatyczne zapamiętywanie danych wprowadzonych do formularzy, a następnie wykorzystanie ich w przyszłości do automatycznego wypełniania innych formularzy. Rozwiązanie to jest bardzo wygodne, ale niestety często może być przyczyną poważnych problemów z wyciekami ważnych danych. Dlatego w propozycji piątej specyfikacji języka HTML dodano nowy atrybut o nazwie `autocomplete=" "`, który może przybierać wartości `on` lub `off`. Element ten pozwala na kontrolowanie mechanizmu autouzupełniania formularzy. Dokładniej mówiąc, określa on, czy przeglądarka może zapisać dane wprowadzone do formularza do swojej "pamięci". Domyślnym ustawieniem każdego formularza jest

wartość on. Innymi słowy, jeżeli do formularza nie dodasz atrybutu autocomplete="on", to i tak funkcja ta będzie włączona. Ustawienie wartości off spowoduje, że przeglądarka nie będzie zapamiętywać danych wprowadzonych do formularza.

Atrybut autocomplete="" może występować również w odniesieniu do konkretnego pola. Dzięki temu możesz np. zablokować zapamiętywanie haseł lub numerów kart kredytowych, pozwalając jednocześnie na pamiętanie adresu czy nazwiska.

Listing widoczny poniżej wyłącza automatyczne wypełnianie pól formularza danymi przechowywanymi w pamięci przeglądarki:

```
<form action="http://helion.pl/search.cgi" method="get"
  autocomplete="off">
  ...
</form>
```

Drugi nowy atrybut nosi nazwę novalidate="" i przybiera tylko jedną wartość, novalidate. Ten element pozwala wyłączyć domyślne sprawdzanie zawartości wysłanego formularza.

```
<form action="http://helion.pl/search.cgi" method="get"
  novalidate="novalidate">
  ...
</form>
```

Mówiąc prościej, jeżeli w kodzie strony w znaczniku <form> znajdzie się atrybut novalidate="novalidate", formularz zostanie wysłany bez sprawdzania zawartości. W praktyce oznacza to, że wysłany zostanie również formularz z błędnymi danymi lub z pustymi polami.

### **Pola typu input**

Formularz elektroniczny tak jak jego papierowy odpowiednik składa się z odpowiednich pól służących do różnych celów. Pierwsza grupa pól formularza to input — czyli pola pozwalające wprowadzić określone typy danych. Oto przykład ogólnej definicji:

```
<input />
```

Przyjrzyjmy się bliżej poszczególnym atrybutom znacznika <input />. Pierwszy z nich nosi nazwę type="" i określa rodzaj pola formularza. Do wyboru mamy następujące typy pól:

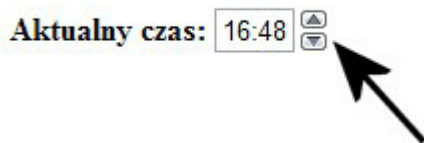
Przyjrzyjmy się bliżej poszczególnym atrybutom znacznika <input />. Pierwszy z nich nosi nazwę type="" i określa rodzaj pola formularza. Do wyboru mamy następujące typy pól:

- type="text" — wstawia zwykłe pole tekstowe.

- `type="search"` — zwykłe pole tekstowe wykorzystywane podczas budowy wyszukiwarki osadzonej na stronie WWW. Cechą charakterystyczną pola typu `search` jest to, że po wprowadzeniu treści pojawia się znak `x`, za pomocą którego możemy sprawnie usunąć wpisany tekst
- `type="tel"` — pole tekstowe służące do zbierania informacji na temat numeru telefonu.
- `type="url"` — pole tekstowe wykorzystywane do wprowadzania adresu strony WWW w formacie `http://www.adresstrony.pl`.
- `type="email"` — pole tekstowe wykorzystywane do wprowadzania adresu e-mail.
- `type="datetime"` — pole do wprowadzania liczb oznaczających datę i czas dla strefy UTC w formacie `rrrr-mm-ddThh:mmZ`. Jeżeli przeglądarka poprawnie obsługuje ten element, to po kliknięciu tego pola kursorem myszy powinien się pojawić wygodny kalendarz, z którego możemy wybrać odpowiednią datę.



- `type="date"` — pole do wprowadzania liczb oznaczających datę w formacie `rrrr-mm-dd` (np. 2010-12-16). Jeżeli przeglądarka poprawnie obsługuje ten element, to po kliknięciu tego pola kursorem myszy powinien się pojawić wygodny kalendarz, z którego możemy wybrać odpowiednią datę. Kalendarz jest podobny do tego z rysunku 3.30.
- `type="month"` — pole do wprowadzania liczb oznaczających datę w formacie `rrrr-mm` (np. 2010-12). Jeżeli przeglądarka poprawnie obsługuje ten element, to po kliknięciu tego pola kursorem myszy powinien się pojawić wygodny kalendarz, z którego możemy wybrać odpowiednią datę. Kalendarz jest podobny do tego z rysunku 3.30.
- `type="week"` — pole do wprowadzania liczb oznaczających datę w formacie `rrrr-numer tygodnia` (np. 2010-W50). Jeżeli przeglądarka poprawnie obsługuje ten element, to po kliknięciu tego pola kursorem myszy powinien się pojawić wygodny kalendarz, z którego możemy wybrać odpowiednią datę. Kalendarz jest podobny do tego z rysunku wyżej.
- `type="time"` — pole do wprowadzania liczb oznaczających czas w formacie `hh:mm:ss` (np. 23:59:59). Jeżeli przeglądarka poprawnie obsługuje ten element, to obok pola powinny pojawić się dodatkowe przyciski, za pomocą których możemy ustawić odpowiednią godzinę.



Ze względu na dużą ilość możliwych opcji dostępnych dla formularzy ograniczymy się jedynie do prostego przykładu.

#### Ćwiczenie 28a

1. Utwórz folder o nazwie „28 Formularz”.
2. Otwórz plik przygotowany w ćwiczeniu 2.
3. Pomiedzy znacznikami <body></body> umieść swój formularz.
4. W pierwszej kolejności dodaj ramy formularza oraz określ sposób wysyłki i ukryte pola niezbędne do tego, aby wysłać formularz. Wystarczy, że w kodzie swojej strony umieścisz poniższy listing.

```
<form action="http://www.roy.icenter.pl/form/" method="post" />
<input type="hidden" name="to" value="twój@adres-email.pl" />
<input type="hidden" name="subject" value="Temat wiadomości" />
```
5. Dodaj dwa pola typu input. Pierwsze z nich opisz jako Imię, a drugie jako e-mail.
6. Dodaj pole typu textarea. W tym miejscu będzie można wprowadzić treść wiadomości.
7. Skorzystaj ze znaczników <fieldset></fieldset> oraz <legend></legend>, aby zgrupować pola formularza tak, by wyglądał podobnie do tego z rysunku
8. Zapisz dokument i otwórz go w przeglądarce. Spróbuj uzupełnić dostępne pola i wysłać formularz.
9. Jeżeli masz ochotę, spróbuj dodać kolejne pola formularza. Możesz przetestować działanie funkcji sprawdzania poprawności danych wpisanych do formularza.

Zostaw swój komentarz

**Imię:**

**E-mail:**

**Komentarz:**

Rysunek 1 Tak powinien wyglądać Twój formularz

```

9 <body>
10 <form action="http://www.roy.icenter.pl/form/" method="post" method="post" />
11 <fieldset>
12 <legend> Zostaw swój komentarz
13 </legend>
14 <input type="hidden" name="to" value="twój@adres-email.pl" />
15 <input type="hidden" name="subject" value="Testowa wiadomość" />
16 <dl>
17 <dt><strong>Imię:</strong>
18 </dt>
19 <dd>
20 <input type="text" name="imię" size="30" />
21 </dd>
22 <dt><strong>E-mail:</strong>
23 </dt>
24 <dd>
25 <input type="email" name="imię" size="30" />
26 </dd>
27 <dt><strong>Komentarz:</strong>
28 </dt>
29 <dd>
30 <textarea name="opinie" rows="30" cols="50"></textarea>
31 </dd>
32 </dl>
33 <p>
34 <input type="submit" value="Wyślij" />
35 </p>
36 </fieldset>
37 </form>
38 </body>

```

Rysunek 2 Tak powinien wyglądać kod

## Ćwiczenie 28b

1. Utwórz plik HTML z następującym kodem:

```

1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <form action="demo_form.asp">
6   Data urodzenia: <input type="date" name="bday">
7   <input type="submit">
8 </form>
9
10 </body>
11 </html>
12

```

2. Efekt sprawdź w przeglądarce Chrome.

## Ćwiczenie 28c

1. Utwórz plik HTML z następującym kodem:

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <form >
6      Ulubiony kolor: <input type="color" name="favcolor"><br>
7      <input type="submit">
8  </form>
9
10 </body>
11 </html>
```

2. Efekt sprawdź w przeglądarce Chrome.

## Ćwiczenie 29

1. Utwórz folder o nazwie „29 Rozbudowany formularz”.
2. Do folderu wgraj plik arkusza stylów ZookeeperForm.css.

Utwórz dokument HTML następującej treści:

```
<!DOCTYPE html>
<html lang="pl">
<head>
  <meta charset="utf-8">
  <title>Opiekun zwierząt w zoo</title>
  <link href="ZookeeperForm.css" rel="stylesheet">
</head>

<body>

<h1>Opiekun zwierząt w zoo</h1>
<h2>Formularz podania o pracę</h2>
<form action="#">
  <p><i>Proszę wypełnić ten formularz. Obowiązkowe pola oznaczono znakiem
</i><em>*</em>.</p>
  <fieldset>
    <legend>Dane kontaktowe</legend>
    <label for="first_name">Imię <em>*</em></label>
    <input id="first_name"><br>
    <label for="family_name">Nazwisko <em>*</em></label>
    <input id="family_name"><br>
    <label for="telephone">Nr telefonu</label>
    <input id="telephone"><br>
    <label for="email">E-mail <em>*</em></label>
    <input id="email"><br>
  </fieldset>
  <fieldset>
    <legend>Szczegółowe dane osobowe</legend>
    <label for="age">Wiek <em>*</em></label>
    <input id="age"><br>
    <label for="gender">Płeć</label>
    <select id="gender">
      <option value="female">Kobieta</option>
      <option value="male">Mężczyzna</option>
    </select><br>
  </fieldset>
</form>
```

```

        <label for="comments">Kiedy zdałaś/łeś sobie sprawę, że opiekun zwierząt to
zawód dla Ciebie?</label>
        <textarea id="comments"></textarea>
    </fieldset>

    <fieldset>
        <legend>Zaznacz swoje ulubione zwierzęta</legend>
        <label for="zebra"><input id="zebra" type="checkbox"> zebra</label>
        <label for="cat"><input id="cat" type="checkbox"> kot</label>
        <label for="anaconda"><input id="anaconda" type="checkbox"> anakonda</label>
        <label for="human"><input id="human" type="checkbox"> człowiek</label>
        <label for="elephant"><input id="elephant" type="checkbox"> słoń</label>
        <label for="horse"><input id="horse" type="checkbox"> koń</label>
        <label for="pigeon"><input id="pigeon" type="checkbox"> gołąb</label>
        <label for="crab"><input id="crab" type="checkbox"> krab</label>
    </fieldset>
    <p><input type="submit" value="Wyślij formularz"></p>
</form>
</body>
</html>

```

### 3. Uzupełnij formularz o nowe elementy. Efekt sprawdź w przeglądarce.

---

#### Ćwiczenie 30

1. Utwórz folder „30 Pasek stanu i miernik”.  
Utwórz dokument HTML zawierający kod:

```

<!DOCTYPE html>

<html lang="pl">
<head>
    <meta charset="utf-8">
    <title>Znaczniki <progress> i <meter></title>
</head>
<body>
    <div>
        <h1>Paski postępu</h1>
        <p>Postęp: <progress max=100 value=50>50%</progress><br>
        Postęp: <progress>Wykonywanie zadania w toku... </progress>
    </div>

    <div>
        <h1>Wskaźniki</h1>
        <p>Twoja walizka waży: <meter min="2" max="20" value="16">13 kilogramów</meter><br>
        Twoja walizka waży: <meter min="2" max="20" low="4" high="15" value="16">16
        kilogramów</meter>*
        <p><small>* Bagaże cięższe niż 15 kilogramów podlegają dodatkowej opłacie.</small></p>
    </div>
</body>
</html>

```



```

<div class="separatedSection">
<p>Naszym celem jest zebranie 50 000 dolarów na rzecz fundacji FRL (Fundacja Ratowania
Lemingów).</p>
<p>Do tej pory zebraliśmy 14 000 dolarów. <meter max="50000" value="14000"></meter>

</div>
</body>
</html>

```

2. Dodaj własny przykład paska postępu i wskaźnika.
3. Efekt wypróbuj w przeglądarce.

## Kaskadowe arkusze stylów

Pojęcie kaskadowe arkusze stylów przewijało się już na naszych pierwszych zajęciach. Nadszedł w końcu czas abyśmy przyjrzeni się temu zagadnieniu bliżej. Od wersji HTML 3.2 wprowadzono pewne innowacje w strukturze języka i zaczęto zastępować część znaczników innymi. Nowe znaczniki nazwano kaskadowymi arkuszami stylów i sukcesywnie je rozbudowywano. Oficjalna specyfikacja CSS ma numer 3. Dzięki stylom możemy mieć pełną kontrolę nad formatowaniem dokumentu. Żadne z poleceń języka HTML nie pozwalało na regulowanie odstępów pomiędzy blokami oraz nakładanie bloków na siebie. Warto wspomnieć o tym, że style umożliwiają kontrolę tła poszczególnych części dokumentu, właściwości czcionek zastosowanych na stronie, tabel, formularzy i wielu innych elementów.

Kaskadowe arkusze stylów, w skrócie CSS, możemy określić jako narzędzie do formatowania wyglądu dokumentów. Natomiast za pomocą samych stylów nie możemy stworzyć strony. Podobnie jak języki HTML kaskadowe arkusze stylów są standaryzowane przez W3C i tam należy szukać odpowiedniej specyfikacji.

## Budowa stylu

Konstrukcja stylu nie należy do zbyt skomplikowanych, co widać w poniższym przykładzie.

```
selektor {cecha:wartość; cecha:wartość; etc.;}
```

*Selektor* jest odpowiednikiem znacznika występującego w języku HTML, może to być np. <p> lub <div>. Drugi człon stylu jest ujęty w nawias klamrowy { } i zawiera polecenia formatujące. Polecenie formatujące — *cecha* — jest określane przez specyfikację, natomiast *wartość* jest zmienną definiowaną przez nas w zależności od aktualnych potrzeb. Przykładowy prosty styl definiujący kolor bloku tekstu jako czerwony, o wysokości czcionki równej 14 punktów, wygląda tak:

```
p
{
    color:red;

    font-size:14pt;
}
```

Należy zwrócić uwagę na to, że cecha jest oddzielona od wartości za pomocą dwukropka, a po wartości zawsze występuje średnik. Stosowanie średnika po ostatniej parze cechy i wartości nie jest wymagane, ale oszczędzi wielu niespodzianek.

Zapis stylu może przybrać inną postać, jednak jego działanie pozostaje niezmiennie.

```
p {color:red; font-size:14pt;}
```

Jeżeli zapomnimy rozdzielić poszczególnych par cech i wartości za pomocą średników, to styl nie będzie działał poprawnie.

### Metody osadzania stylów na stronie

Kaskadowe arkusze stylów mogą być osadzone w dokumencie HTML na kilka sposobów. Specyfikacja przewiduje następujące rozwiązania:

- Style wewnętrzne:
- lokalne,
- zagnieżdżone.
- Style zewnętrzne:
- dołączone,
- importowane.

Pierwszym rozwiązaniem jest umieszczenie deklaracji stylu bezpośrednio w formatowanym znaczniku. Rozwiązanie takie nosi nazwę **stylu lokalnego**, gdyż wpływa wyłącznie na zawartość znacznika, w którym się znajduje.

```
<p>Treść dokumentu</p>
```

```
<p style="color:green; font-size:15pt;">Treść dokumentu</p>
```

Wadą powyższego rozwiązania jest spore utrudnienie w przypadku zmiany sposobu formatowania jakiegoś elementu oraz niepotrzebne powiększanie objętości kodu. Weźmy na przykład dokument, w którym umieszczono czterdzieści bloków tekstu i każdy z nich ma wyglądać identycznie. Aby to osiągnąć, do każdego z nich należy wpisać odpowiednie style. Wiąże się to z wykonaniem dodatkowej pracy, stratą czasu oraz zwiększeniem objętości kodu. Sytuację taką przewidzieli twórcy kaskadowych arkuszy stylów, dzięki czemu mamy inne możliwości umieszczania CSS bezpośrednio w nagłówku `<head></head>` strony. Rozwiązanie to nazywane jest **stylem zagnieżdżonym**.

```

<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Tytuł przykładowej strony</title>
    <style type="text/css">
      p
      {
        color:green;
        font-family:Arial;
      }
    </style>
  </head>
  <body>
    <p>Przykładowy blok tekstu.</p>
  </body>
</html>

```

Style wewnętrzne — zarówno lokalne, jak i zagnieżdżone — poza zaletami mają również dość poważne wady. Weźmy jako przykład rozbudowany serwis, np. portal Onet.pl, na który składają się setki lub tysiące podstron. Każda z podstron jest formatowana w podobny sposób i wykorzystuje ten sam zestaw stylów. W takim przypadku style lokalne odpadają, a zagnieżdżanie definicji w nagłówkach <head></head> również nie jest najlepszym pomysłem. Na szczęście istnieje jeszcze trzecia możliwość, **styl zewnętrzny**, która opiera się na znaczniku <link /> umieszczanym bezpośrednio w nagłówku strony, dzięki czemu taki arkusz również działa w całym dokumencie. Znacznik ten pozwala na podłączenie zewnętrznego arkusza stylów. Odpowiedni wpis ma następującą postać:

```

<!doctype html>
<html>
  <head>
    <meta charset="UTF-8" />
    <title>Tytuł przykładowej strony
    </title>
    <link rel="stylesheet" href="arkusz.css" type="text/css" />
  </head>
  <body>
    <p>Przykładowy blok tekstu</p>
  </body>
</html>

```

Atrybut href="" odpowiada za określenie nazwy oraz lokalizacji wybranego arkusza. Następny element, rel="", definiuje relację, które w przypadku CSS zawsze mają wartość stylesheet. Ostatni element, type="", zawsze ma wartość text/css. Dzięki temu przeglądarka "wie", z jakim arkuszem ma do czynienia. Działanie tego elementu docenimy w przyszłości, gdy powstaną inne arkusze i będzie trzeba z nich skorzystać.

Zewnętrzny arkusz jest zwykłym plikiem tekstowym, a jego zawartość może wyglądać tak jak w poniższym przykładzie.

```
/* To jest przykład zewnętrznego arkusza stylów*/  
p  
{  
color:green;  
font-family:Arial;  
}
```

Zewnętrzny arkusz powinien być zapisany w pliku o dowolnej nazwie, która nie zawiera polskich znaków. Bezwzględnie taki dokument musi mieć rozszerzenie zgodne ze wzorem *nazwa-pliku.css*.

---

Specyfikacja CSS daje nam jeszcze możliwość zaimportowania arkusza z innej strony oraz zdefiniowania kilku zewnętrznych arkuszy dla tej samej strony. Jednak na tym etapie te rozwiązania nie są Ci potrzebne. Dlatego zdecydowałem się o nich wspomnieć jedynie w formie uwagi.

---

### Ćwiczenie 31 Osadzanie stylów na stronie

Stwórz prostą stronę i umieść na niej dwa bloki tekstu, a następnie podłącz zewnętrzny arkusz stylów. Korzystając z CSS, przypisz czerwony kolor dla dodanego tekstu. Aby wykonać to ćwiczenie, postępuj zgodnie z instrukcjami zamieszczonymi w kolejnych podpunktach.

1. Utwórz folder o nazwie „31 Osadzanie stylów na stronie”.
2. Otwórz plik przygotowany w ćwiczeniu 2.
3. Pomiedzy znacznikami <body></body> dodaj dwa bloki tekstu. W tym celu skorzystaj ze znacznika <p></p>.
4. Sprawdź, czy w nagłówku strony znajduje się deklaracja odpowiedzialna za podłączenie zewnętrznego arkusza stylów. Przypomnę, że odpowiedni zapis powinien mieć postać:

```
<link rel="stylesheet" href="arkusz.css" type="text/css" />
```

5. Zapisz dokument HTML. Nadaj mu nazwę *szablon.html*.
6. Za pomocą edytora tekstowego utwórz plik o nazwie *arkusz.css*. W związku z tym, że jeszcze nie znamy CSS, spróbuj przepisać poniższą deklarację.

```
p{color:red;}
```

7. Otwórz dokument HTML i sprawdź, czy tekst ma kolor czerwony.