

Programowanie aplikacji internetowych

Pracownia specjalistyczna, studia podyplomowe, rok 2018/2019

Temat: Język HTML i style CSS

1. Stwórz formularz HTML pozwalający na rejestrację użytkownika w aplikacji internetowej. Wykorzystaj szablon *zadanie1.html*. Formularz powinien zapewniać użytkownikowi możliwość wpisania takich danych, jak:
 - login,
 - hasło (niewidoczne przy wprowadzaniu),
 - email,
 - data urodzenia,
 - wybór kraju (spośród 5 dostępnych),
 - płeć,
 - akceptacja regulaminu,
 - przycisk *Utwórz konto* oraz *Anuluj*.Sprawdź różnicę w działaniu metody przekazywania danych formularza – GET oraz POST.
2. Pobierz archiwum *worzec.zip* i przeanalizuj strukturę strony *index.html* pod kątem wykorzystania stylów CSS do rozmieszczenia poszczególnych elementów (logo, treść podstrony, menu nawigacyjne, stopka).
3. Stworzony w punkcie nr 1 formularz rejestracyjny umieść w pliku *rejestracja.html* korzystając z wzorca *index.html*. Wszystkie pola tekstowe powinny mieć szerokość 150px. Tło każdego pola tekstowego powinno być jasnoszare. Przyciski na formularzu powinny mieć ciemnoszare tło oraz biały napis.
4. Stwórz dwa wpisy w menu nawigacyjnym: *strona główna* oraz *rejestracja*.

Podstawowe znaczniki formularza HTML

Podstawowe znaczniki języka HTML pozwalające budować formularze interakcji z użytkownikiem:

- `<form name="nazwa" method="post" action="index.html"></form>` – podstawowy znacznik tworzący formularz. Pole *method* odpowiada za tryb przekazywania danych (POST lub GET), zaś *action* określa adres strony, na którą powinno zostać przekazane sterowanie po wykonaniu postback'a. Pole *name* określa nazwę, pod jaką widoczne jest dane pole/formularz w po wykonaniu przetwarzania.
- `<input type="text" name="nazwa" value="wartość" />` – znacznik tworzący zwykłe pole tekstowe o nazwie *nazwa*. Klucz *value* określa wartość początkową wpisaną w pole,
- `<input type="password" name="nazwa" value="wartość" />` – znacznik tworzący pole tekstowe przeznaczone dla haseł (wprowadzane znaki są zastępowane np. kropkami) o nazwie *nazwa*. Klucz *value* określa wartość początkową wpisaną w pole,
- `<input type="submit" name="nazwa" value="Treść przycisku" />` – znacznik tworzący przycisk przetwarzania formularza o nazwie *nazwa*,
- `<select name="nazwa"> <option value="1">Treść 1</option> ... </select>` – znacznik tworzący listę wyboru wartości o nazwie *nazwa*. Może się składać z wielu linii *option*, przy czym których każda określa pojedynczy wiersz wyboru,
- `<input type="radio" name="nazwaGrupy" value="kodWyboru">Opis wyboru</input>` – znacznik tworzący listę przycisków typu *radiobutton* należących do grupy *nazwaGrupy* (tzn. zapewniających tryb pojedynczego wyboru w ramach grupy o tej samej nazwie), przy czym zaznaczoną opcję można rozpoznać jako *kodWyboru*,
- `<input type="checkbox" name="nazwa" value="wartość">Opis opcji</input>` – znacznik tworzący pole typu *checkbox*. W przypadku zaznaczenia checkbox'a na formularzu, będzie on dołączony do przesyłanych danych jako *nazwa = wartość*.

Składnia stylu CSS

Podstawowa składnia stylu CSS:

```
/* Komentarz */
selektor {
    właściwość:wartość;
    właściwość:wartość;
}
```

Selektorem może być nazwa tagu HTML, nazwa klasy lub identyfikator elementu (tagu).

Przykład:

```
p {
    color:red;
    text-align:center;
}
```

Identyfikatory i klasy selektorów

Następujący styl CSS będzie połączony z dowolnym tagiem HTML zawierającym identyfikator *id=id_elementu* (na stronie musi występować tylko jeden element o danym identyfikatorze):

```
#id_elementu {
    color:#ff0000;
}
```

Następujący styl CSS będzie połączony z dowolnym tagiem HTML określonym klasą *nazwa_klasy* (np. `<p class="nazwa_klasy">Jakiś tekst</p>`)

```
.nazwa_klasy {
    color:orange;
}
```

Klasy mogą być ograniczone tylko do danego typu elementu:

```
p.nazwa_klasy {
    color:black;
}
```

Łączenie wielu selektorów w przypadku, gdy posiadają takie same właściwości:

```
h1, h2, h3 {
    color:green;
}
```

Zagnieżdżanie stylów

Przykład zagnieżdżonych stylów CSS:

```
p {
    color:blue;
}

.marked {
    background-color:blue;
}

.marked p {
    color:white;
}
```

W tym przypadku, tag `<p>` będzie wyświetlony w kolorze niebieskim, jeśli będzie znajdował się poza elementem (np. `<div>`) ze zdefiniowaną klasą `.marked` lub w kolorze białym na niebieskim tle, jeśli zostanie umieszczony w elemencie z oznaczoną klasą `.marked`:

```
<p>Ten tekst będzie niebieski</p>
```

```
<div class="marked">
    <p>Ten tekst będzie biały na niebieskim tle</p>
</div>
```

Miejsca definiowania arkuszy stylów

Miejsca, w których można definiować arkusze stylów CSS:

- plik zewnętrzny:

```
<head>
    <link rel="stylesheet" type="text/css" href="mystyle.css" />
</head>
```
- w sekcji head:

```
<style type="text/css">
    hr {
        color:sienna;
    }
</style>
```
- bezpośrednio w znaczniku elementu:

```
<p style="margin-left:20px">Jakiś tekst</p>
```

Priorytet dziedziczenia stylów CSS

1. Styl tagu HTML.
2. Styl w sekcji *head*.
3. Styl z zewnętrznego pliku.
4. Ustawienia domyślne przeglądarki.

Dokumentacja dotycząca stylów CSS

1. Tutorial *w3schools*: <http://www.w3schools.com/css/>
2. Lista właściwości stylów: http://www.w3schools.com/css/css_reference.asp
3. Box model: http://www.w3schools.com/css/css_boxmodel.asp