

12. FORMULARZE

Formularze to po prostu możliwość wymiany opinii internautów o stronie. To nie wszystko. Gdyby przyrzeć się uważnie są one wszędzie – w ankietach, w poczcie elektronicznej, na forach dyskusyjnych, w komunikatorach internetowych, po prostu wszędzie, gdzie możemy coś wpisać/zaznaczyć i coś się z tym stanie.

Atrybut ACTION

Formularze mają bardzo wiele różnych typów pól, co pozwala na bardzo precyzyjne komunikowanie się. Od razu powiem, że do obsługi formularzy najlepiej przygotować skrypty, działające po stronie serwera, ale ten kurs nie obejmuje tego tematu. Formularz tworzy się za pomocą znacznika **<FORM>** pomiędzy którym będzie widniało to, co jest uznawane jako formularz, bądź jego element. Jego najważniejszym atrybutem jest **ACTION**, który określa, gdzie mają trafić dane po kliknięciu na wyślij, którego wartości to:

- adres e-mail poprzedzony mailto: np.: login @witryna.pl
- adres skryptu obsługującego dany formularz np.: skrypt.php

Atrybut METHOD

Innym bardzo istotnym atrybutem jest **METHOD**, który informuje, gdzie mają być wysłane dane po kliknięciu na wyślij. Jego wartościami są:

- **get** - wysłanie danych wewnątrz adresu strony
- **post** - wysłanie danych do pamięci przeglądarki i następnie odczytanie ich

Oczywiście zalecam używanie drugiej metody, ponieważ nie ma możliwości stworzenia specjalnego linku nie wypełniając formularza i przy okazji sam link jest ładniejszy. **Uwaga:** odświeżenie strony, którą obsługuje skrypt, po wysłaniu formularza powoduje przesłanie formularza ponownie.

Typu pól formularza

Pierwszym polem formularza jest pole, także przycisk tworzony za pomocą znacznika **<INPUT>**. Chociaż na większości przeglądarek sam znacznik powinien zamienić się na pole tekstowe, to zgodnie ze specyfikacją powinniśmy użyć wszechstronnego atrybutu **TYPE**, którego wartości to:

KURS HTML

- **text** - najwyklesze pole tekstowe
- **password** - pole tekstowe maskujace znaki (gwiazdki, lub inne zmienniki)
- **checkbox** - kwadracik, ktory mozna zaznaczyc (mozna zaznaczyc wiele kwadratow)
- **radio** - koleczko, ktore mozna zaznaczyc (mozna zaznaczyc jedno koleczko)
- **submit** - przycisk wysylajacy formularz
- **image** - graficzna odmiana przycisku wysylajacego formularz
- **reset** - przycisk odswiezajacy formularz (przywracajacy domyslne ustawienia)
- **file** - pole umozliwiajace dodanie pliku
- **button** - przycisk robiaacy cos w polaczeniu z jakimis skryptem

Uwaga: Pamietajmy, ze samo pole to nic, niektorym polom powinniemy nadać takze wartosc – atrybut **VALUE**, gdyz odbiorca formularza nie dostanie obrazka z wypelnionym formularzem, tylko wlasnie wartosc VALUE, ale obowiazkowo kazdy powinien posiadac swoja nazwe – atrybut **NAME**. Atrybut **VALUE** opisaną wartosc bedzie miał tylko w polach: checkbox i radio w pozostalych bedzie to domyslna wartosc (domyslnie wypelnione pole tekstowe, lub napis na przycisku zamiast domyslnego, ale w tym przypadku nie jest to obowiazkowe, ale na przykladach zostanie wykorzystane).

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
Twoje imię:  
<INPUT TYPE="text" NAME="imie">  
</FORM>
```

Skoro poznalismy juz teoretycznie co robi znacznik VALUE, wykorzystajmy go w praktyce:

```
<FORM ACTION="mailto:login@witryna.pl" METHOD="post">  
Twoje imię administratora Krzychur HomePage:  
<INPUT TYPE="text" NAME="imie" VALUE="Krzysio">  
</FORM>
```

KURS HTML

Pole typu "password"

W celu zaszyfrowania jakiejś informacji tak, aby nie zobaczyła jej osoba siedząca obok nas używamy **password**. Jeżeli w tym przypadku użyjemy atrybutu VALUE atrybut zadziała poprawnie, ale i tak reprezentowane przez niego znaki zamienia się na zmienniki np.:

```
<FORM ACTION="mailto:login@witryna.pl" METHOD="post">
```

```
Twoje hasło do poczty elektronicznej:
```

```
<INPUT TYPE="password" NAME="haslo">
```

```
</FORM>
```

Opcja "checkbox" czyli zaznaczanie opcji

Teraz trochę kwadracików. Użytkownik wypełniający formularz po zaznaczeniu kilku kwadracików, bez wpisywania długich tekstów przekaże nam informacje. Pamiętajmy, że kwadraciki muszą mieć takie same nazwy (atrybut NAME), a ich wartości (atrybut VALUE), aby było poprawne ortograficznie.

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">
```

```
Czego brakuje na stronie:<BR>
```

```
<INPUT TYPE="checkbox" NAME="brak" VALUE="Artykułów">
```

```
Artykułów<BR>
```

```
<INPUT TYPE="checkbox" NAME="brak" VALUE="Wszystkiego">
```

```
Wszystkiego<BR>
```

```
<INPUT TYPE="checkbox" NAME="brak" VALUE="Niczego"> Niczego
```

```
</FORM>
```

Jeden z kilku - pola typu "radio"

Czasami musimy wybrać jedną opcję z kilku. W tym celu musimy skorzystać z pól typu **radio**. Pamiętajmy, że jeżeli chcemy wybrać jedną opcję z kilku pól, wszystkie pola muszą mieć taką samą nazwę (atrybut NAME), jednakże jeżeli chcemy w jednym formularzu stworzyć kilka takich list wyboru musimy każdej liście, w której chcemy zaznaczyć jakąś kropkę z kilku opcji musimy używać unikalnych nazw. Oto dwa przykłady wykorzystania pól typu radio. Pierwszym przykładem jest zastosowanie jednej listy wyboru, a drugiej dwóch:

KURS HTML

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">
```

```
Jak oceniasz serwis:<BR>
```

```
<INPUT TYPE="radio" NAME="ocena" VALUE="Dobrze"> Dobrze<BR>
```

```
<INPUT TYPE="radio" NAME="ocena" VALUE="Źle"> Źle<BR>
```

```
<INPUT TYPE="radio" NAME="ocena" VALUE="Średnio"> Średnio
```

```
</FORM>
```

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">
```

```
Jak oceniasz serwis:<BR>
```

```
<INPUT TYPE="radio" NAME="ocena" VALUE="dobrze"> Dobrze<BR>
```

```
<INPUT TYPE="radio" NAME="ocena" VALUE="zle"> Źle<BR>
```

```
<INPUT TYPE="radio" NAME="ocena" VALUE="srednio"> Średnio<BR><BR>
```

```
Często tutaj bywasz?<BR>
```

```
<INPUT TYPE="radio" NAME="bywanie" VALUE="tak"> Tak<BR>
```

```
<INPUT TYPE="radio" NAME="bywanie" VALUE="nie"> Nie<BR>
```

```
<INPUT TYPE="radio" NAME="bywanie" VALUE="pierwszy"> Jestem po raz  
pierwszy
```

```
</FORM>
```

Napis na przycisku typu "submit"

Jak na razie jeszcze nie było jednego z najważniejszych dokumentów. Mianowicie przycisku, który po naciśnięciu wysyła dane na odpowiedni adres zapisany w atrybucie ACTION znacznika <FORM>. Poniższe kody przycisków musimy umieścić najlepiej na końcu każdego formularza. Pierwszym z nich jest użyty domyślny napis na przycisku (w mojej przeglądarce jest to Wyślij zapytanie), a w drugim jest użyty atrybut VALUE, których przechowuje napis na przycisku:

KURS HTML

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<INPUT TYPE="submit">  
</FORM>
```

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<INPUT TYPE="submit" VALUE="Wyślij">  
</FORM>
```

Graficzny przycisk

Jeżeli użyjemy wartości przycisku **image** i użyjemy atrybutu **SRC**, w którym umieścimy ścieżkę do pliku graficznego możemy stworzyć przycisk wysyłający formularz, który jest grafiką np.:

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<INPUT TYPE="image" SRC="wyslij.gif">  
</FORM>
```

Powyższy przycisk jest odpowiedzialny za przesłanie danych dalej. Jeżeli chcemy stworzyć graficzny przycisk innego typu np. text, czy radio musimy użyć języka CSS i użyć deklaracji background np.:

```
<STYLE TYPE="text/css">  
input  
{  
background: url(http://www.webdiary.pl/modules/technologie/pole.gif)  
}  
</STYLE>
```

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<INPUT TYPE="text" NAME="tekst" STYLE=" ">  
</FORM>
```

KURS HTML

Przycisk typu "reset" - czyszczenie pól formularza

Istnieje przycisk, który pozwala na usunięcie danych z formularza, które ręcznie wpisaliśmy (wszystkie atrybuty VALUE zostaną domyślnie ponownie umieszczone w polach), jednakże uważam, że jest on beznadziejny, ponieważ użytkownik chcąc zaprzestać dalszemu wypełniania formularza, bądź rezygnacji z jego wysłania może po prostu opuścić stronę, a dane znikną.

Innym argumentem na to, aby zaprzestać jego używaniu jest to, że nie ma on żadnego przycisku cofającego, pomyłkowe kliknięcie (patrzac kątem oka Wyczyść i Wyślij można stwierdzić, że przyciski są jednakowe). Internauta po wypełnieniu ogromnego formularza na pewno nie wypełni go ponownie! Ale cóż:

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<INPUT TYPE="reset" VALUE="Wyczyść">  
</FORM>
```

Załączanie plików do formularza

Jeżeli chcemy załączyć do formularza plik z naszego komputera musimy użyć wartości **file**. Oglądając efekt w przeglądarce bez trudu zauważymy, że jest on połączeniem pola tekstowego i przycisku. W polu tekstowym możemy ręcznie wpisać ścieżkę załączanego dokumentu, a klikając na przycisk (u mnie nazywa się on przeglądaj...) otwieramy okienko, które pozwoli łatwiej zlokalizować dany plik, a po zaakceptowaniu jego ścieżka od razu pojawi się we wspomnianym polu tekstowym. Przykład pola pozwalającego na załączanie plików:

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<INPUT TYPE="file" NAME="plik">  
</FORM>
```

Uwaga: nie można załączać folderów, ale można pakować je w archiwa, które można następnie załączyć.

Pole typu "button"

Ostatnim przyciskiem jest **button**. Umieszczenie go, powoduje powstanie przycisku, który nic nie robi. Jest to stworzone dla skryptów, które po np. naciśnięciu takiego przycisku robią jakieś działanie, np. sprawdzają, czy formularz został wypełniony poprawnie, jednakże przypominam, że kurs nie obejmuje skryptów.

KURS HTML

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<INPUT TYPE="button" VALUE="Wolne klikanie">  
</FORM>
```

Wyłączania aktywności pola

Jest jeszcze wiele atrybutów, które edytują formularz. Jeżeli chcemy czasowo wyłączyć pewne pole, możemy użyć atrybutu **DISABLED** (do wszystkich pól <INPUT>), które nie pozwala zmieniać zawartości pól, ani klikać na nie np.:

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<INPUT TYPE="text" NAME="tekst" DISABLED><BR>  
<INPUT TYPE="submit" DISABLED>  
</FORM>
```

Nie należy mylić poprzedniego atrybutu, z **READONLY**, który oznacza, że pole jest tylko do odczytu, ale "prawdziwe" przeglądarki pozwalają zaznaczyć pole, skopiować jego domyślną zawartość itp. Atrybut ten działa w polach <INPUT> typu:

- **text**
- **password**

Przykład atrybutu READONLY:

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<INPUT TYPE="text" NAME="tekst" READONLY>  
</FORM>
```

Kolejny atrybut - **CHECKED**, oznacza, że coś jest domyślnie zaznaczone. Pamiętajmy, że w listach jednokrotnego wyboru atrybut ten istniejący w kilku opcjach może zostać nieprawidłowo odczytany przez przeglądarkę, albo zostanie użyty do jednego z pól. Atrybut działa w polach <INPUT> typu:

- **checkbox**
- **radio**

Przykład atrybutu CHECKED:

KURS HTML

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<INPUT TYPE="checkbox" NAME="test" VALUE="Test" CHECKED>  
</FORM>
```

W celu ograniczenia maksymalnej ilości znaków w polu używamy atrybutu **MAXLENGTH** o liczbowej wartości oznaczającej... wiadomo co. Atrybut działa w polach `<INPUT>` typu:

- **text**
- **password**

Przykład atrybutu **MAXLENGTH**:

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<INPUT TYPE="text" NAME="test" MAXLENGTH="10">  
</FORM>
```

Uwaga: jeżeli jednocześnie korzystamy z atrybutu **MAXLENGTH** i **VALUE**, to w przypadku umieszczenia w **VALUE** więcej znaków niż przewiduje to **MAXLENGTH** to albo:

- zostanie załadowany mniejszy ciąg znaków licząc
- przeglądarka popiełni jakiś błąd

Listy rozwijane

Jak na razie poznaliśmy tylko jeden znacznik formularza - `<INPUT>`. Jest jeszcze kilka, ale na początku zajmijmy się znacznikiem tworzącym listy rozwijane - `<SELECT>`. Jest to pierwszy opcjonalny znacznik, który poznaliśmy, który znajduje się wewnątrz formularza. Wewnątrz niego znajdują się znaczniki tworzące listę (pamiętamy lekcję o listach wypunktowanych i numerowanych), które są tworzone za pomocą opcjonalnego znacznika `<OPTION>`. W przypadku list rozwijanych atrybut **NAME** znajduje się w znaczniku `<SELECT>`, a **VALUE** w `<OPTION>`. Przykład rozwijanej listy:

KURS HTML

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">
<SELECT NAME="lista">
<OPTION VALUE="pozycja1">Pozycja 1</OPTION>
<OPTION VALUE="pozycja2">Pozycja 2</OPTION>
<OPTION VALUE="pozycja3">Pozycja 3</OPTION>
<OPTION VALUE="pozycja4">Pozycja 4</OPTION>
<OPTION VALUE="pozycja5">Pozycja 5</OPTION>
</SELECT>
</FORM>
```

Jeżeli chcemy widzieć od razu więcej niż jedną opcję, co odrobinę zmieni wygląd listy musimy użyć atrybutu **SIZE** o wartości większej niż 1, gdyż jedynka jest wartością domyślną. Wartość ta oznacza ilość pozycji widzianych bez potrzeby przewijania listy np.

```
<FORM ACTION="mailto:login@witryna.pl" METHOD="post">
<SELECT NAME="lista" SIZE="3">
<OPTION VALUE="pozycja1">Pozycja 1</OPTION>
<OPTION VALUE="pozycja2">Pozycja 2</OPTION>
<OPTION VALUE="pozycja3">Pozycja 3</OPTION>
<OPTION VALUE="pozycja4">Pozycja 4</OPTION>
<OPTION VALUE="pozycja5">Pozycja 5</OPTION>
</SELECT>
</FORM>
```

Chociaż wydaje się, że z takich typów list można wybrać jedną listę, to nie prawda. Można wybrać kilka, jeżeli przypiszemy atrybut **MULTIPLE** znacznikowi **SELECT** np.:

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">
<SELECT NAME="lista" MULTIPLE>
<OPTION VALUE="pozycja1">Pozycja 1</OPTION>
<OPTION VALUE="pozycja2">Pozycja 2</OPTION>
<OPTION VALUE="pozycja3">Pozycja 3</OPTION>
<OPTION VALUE="pozycja4">Pozycja 4</OPTION>
<OPTION VALUE="pozycja5">Pozycja 5</OPTION>
</SELECT>
</FORM>
```

KURS HTML

Uwaga: zastosowanie znacznika pozwalające wybór większej ilości opcji, lista wygląda tak, jakby nadano jej atrybut SIZE o wartości takiej, ile jest pozycji. Jeżeli chcemy wybrać kilka opcji, poprzez standardową listę musimy przypisać liście ten atrybut o określonej wartości, jednakże przy wyborze wartości 1 przyciski przewijania mogą być bardzo małe, albo w ogóle pominięte.

Uwaga: zamiast stosować listy wielokrotnego wyboru zalecam używać pól <INPUT> typu CHECKBOX, gdyż przeciętny internauta nie wie jak z listy wybrać kilka opcji, lub czy w ogóle jest to możliwe, a pisanie zbędnych informacji zmniejsza czytelność formularza.

Za pomocą atrybutu **DISABLED** także możemy zablokować listę, lub konkretne pozycje, prezentuję to na dwóch przykładach. Na pierwszym przykładzie zablokowana jest cała lista, a na drugim tylko jedna opcja:

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">
<SELECT NAME="lista" DISABLED >
<OPTION VALUE="pozycja1" >Pozycja 1</OPTION>
<OPTION VALUE="pozycja2">Pozycja 2</OPTION>
<OPTION VALUE="pozycja3">Pozycja 3</OPTION>
<OPTION VALUE="pozycja4">Pozycja 4</OPTION>
<OPTION VALUE="pozycja5">Pozycja 5</OPTION>
</SELECT>
</FORM>
```

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">
<SELECT NAME="lista">
<OPTION VALUE="pozycja1" DISABLED>Pozycja 1</OPTION>
<OPTION VALUE="pozycja2">Pozycja 2</OPTION>
<OPTION VALUE="pozycja3">Pozycja 3</OPTION>
<OPTION VALUE="pozycja4">Pozycja 4</OPTION>
<OPTION VALUE="pozycja5">Pozycja 5</OPTION>
</SELECT>
</FORM>
```

KURS HTML

Uwaga: w przypadku zablokowania pierwszego pola listy, domyślnie w okienku listy pojawi się kolejna opcja.

Jeżeli chcemy, aby inna opcja była domyślnie zaznaczona, wybranemu znacznikowi `<OPTION>` nadajemy atrybut **SELECTED** np.:

```
<FORM ACTION="mailto:login@witryna.pl" METHOD="post">
<SELECT NAME="lista">
<OPTION VALUE="pozycja1">Pozycja 1</OPTION>
<OPTION VALUE="pozycja2" SELECTED>Pozycja 2</OPTION>
<OPTION VALUE="pozycja3">Pozycja 3</OPTION>
<OPTION VALUE="pozycja4">Pozycja 4</OPTION>
<OPTION VALUE="pozycja5">Pozycja 5</OPTION>
</SELECT>
</FORM>
```

Uwaga: przypisując jednej opcji atrybuty **DISABLED**, **SELECTED** mogą nastąpić różne niezgodności:

- zaznaczenie zablokowanej opcji
- zaznaczenie pierwszej nie zablokowanej opcji
- inne zachowanie się przeglądarki

Duże pola tekstowe TEXTAREA

Jest jeszcze jedno pole formularza. Pozwala ono na wprowadzenie większej ilości tekstu – opcjonalny znacznik `<TEXTAREA>`. Przykładowe wykorzystanie tego znacznika widzimy poniżej:

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">
<TEXTAREA NAME="tekscior"></TEXTAREA>
</FORM>
```

Atrybut ten nie ma wartości **VALUE**. Chcąc, aby wewnątrz pola tekstowego był domyślnie wpisany tekst musimy umieścić go pomiędzy znacznikami `<TEXTAREA>`. Pamiętajmy, że jeżeli umieścimy tam jakieś znaczniki np. `
` znacznik pojawi się widocznie w polu tekstowym. Jeżeli zgodnie z przykładem chcemy umieścić tam kolejny wiersz po prostu piszemy tekst w kilku liniijkach np.:

KURS HTML

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<TEXTAREA NAME="tekscior">  
Tekst<BR>Tekst  
Tekst  
Tekst  
</TEXTAREA>  
</FORM>
```

Nie każdemu podoba się czcionka o stałej szerokości znaku w <TEXTAREA> (liter i, W, czy kropka zajmują tyle samo miejsca). W celu stworzenia normalniejszej czcionki wystarczy użyć arkusza stylów i określić krój czcionki np.:

```
<STYLE TYPE="text/css">  
textarea  
{  
font-family: Verdana  
}  
</STYLE>
```

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">  
<TEXTAREA NAME="tekscior"></TEXTAREA>  
</FORM>
```

W celu zmiany rozmiarów znacznika <TEXTAREA> zwykle na większe, musimy użyć języka CSS i za pomocą znanych już nam deklaracji, dostosować okienko do naszych potrzeb np.:

```
<STYLE TYPE="text/css">  
textarea  
{  
width: 300px;  
height: 100px  
}  
</STYLE>
```

KURS HTML

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">
<TEXTAREA COLS="60" ROWS="10" NAME="tekscior"></TEXTAREA>
</FORM>
```

W polu <TEXTAREA> tak samo jak w kilku innych polach formularza działają znaczniki READONLY i DISABLED. W związku z tym, że tekst w tym polu formularza możemy umieszczać w kilku liniijkach w przeciwieństwie do <INPUT> więc istnieje atrybut, który obsługuje zawijanie wierszy. Mam na myśli atrybut **WRAP**, który ma następujące wartości:

- **off** - wyłączone zawijanie przy pisaniu i wysyłaniu
- **physical** - włączone zawijanie przy pisaniu, ale jest wyłączone przy wysyłaniu
- **virtual** - włączone zawijanie przy pisaniu i wysyłaniu

Co to znaczy, że zawijanie jest włączone/wyłączone przy wysyłaniu? Mianowicie to, że tekst będzie tak samo widoczny np. w wyniku działania skryptu, bądź na poczcie elektronicznej – linijki będą zawinięte w tych samych miejscach, co były zawinięte w polu <TEXTAREA>. W każdym z przypadków można ręcznie przejść do kolejnej linijki, co też będzie widoczne później za pomocą klawisza Enter. Przykład użycia atrybutu WRAP np. z wartością physical:

```
<FORM ACTION="mailto:login @witryna.pl" METHOD="post">
<TEXTAREA WRAP="physical" NAME="tekscior"></TEXTAREA>
</FORM>
```

Uwaga: przy wpisaniu większego wyrazu niż mieści się w jednym wierszu pola tekstowego pojawi się poziomy pasek przewijania.

Dekorowanie formularzy za pomocą CSS

Czas upiększyć pola formularza. Na przeglądarkach o bardzo kolorowym, z wszelakimi animacjami interfejsie, wszystko jest ładne i piękne. Jednakże oprogramowanie, które głównie składa się z koloru szarego pozostawia wiele do życzenia. W zasadzie pole można zmodyfikować tak, jak chcemy, gdyż większość poznanych już deklaracji w polach formularza też zadziała. Jednakże dla mnie, najważniejsze jest wnętrze formularza, obramowanie i czcionka. Oto prosty i jednocześnie całkiem ciekawy styl, wraz z użytymi zagadkami, które jak narazie były omówione teoretycznie:

KURS HTML

```
<STYLE TYPE="text/css">
```

```
input.pole, textarea, select
```

```
{
```

```
font-size: 13px;
```

```
color: brown;
```

```
font-family: Arial, Verdana;
```

```
font-weight: bold;
```

```
border: 1px solid black;
```

```
background: lightgreen
```

```
}
```

```
textarea
```

```
{
```

```
width: 200px;
```

```
height: 300px
```

```
}
```

```
input, select
```

```
{
```

```
width: 200px
```

```
}
```

```
</STYLE>
```

```
<FORM>
```

```
<SELECT NAME="wybor">
```

```
<OPTION VALUE="1">Opcja</OPTION>
```

```
<OPTION VALUE="2">Opcja</OPTION>
```

```
</SELECT><BR><BR>
```

```
<INPUT TYPE="text" CLASS="pole" NAME="pole"><BR><BR>
```

```
<TEXTAREA NAME="tekscior"></TEXTAREA>
```

```
</FORM>
```

KURS HTML

Uwaga: formatując formularze, a konkretnie pola `<INPUT>` powinniśmy zawsze stosować klasy – na niektórych przeglądarkach może dziwnie wyglądać kwadracik z różnym tłem oblepiony w około obramowaniem...