**Temat: Tworzenie formularzy.**

1. Typy pól formularza.

* checkbox — pole wyboru,
* password — pole tekstowe do wpisywania haseł,
* text — pole tekstowe,
* textarea — rozszerzone pole tekstowe.
* radio — pole wyboru,
* select — lista wyboru,

i inne

1. Przyciski na formularzu.

* reset — przycisk reset,
* submit — przycisk submit,
* button — klasyczny przycisk.

**Pamiętaj, że nie wystarczy wpisywać typ pola, aby formularz działał poprawnie.**

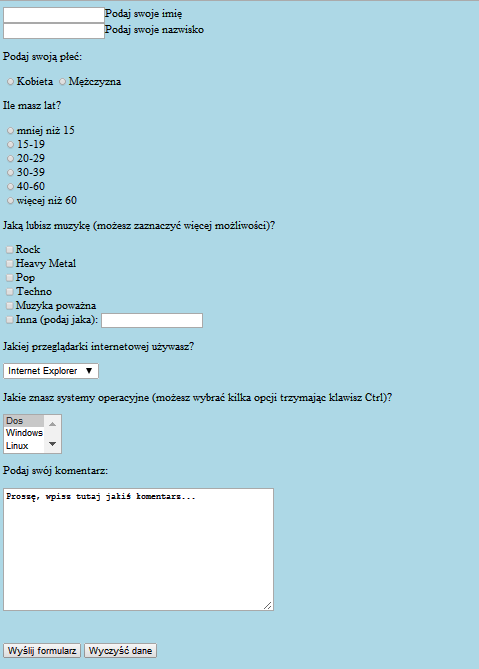
**Znaczniki obsługujące pola, potrzebują także innych atrybutów!**

**Zadanie 12.**

Utwórz poniższy formularz.

<form action="mailto:twój\_adres\_poczty@domena" method="post" enctype="text/plain">

</form>

****

**Materiały z podręcznika wydawnictwa WSIP:**

Każdy formularz zaczyna się od znacznika otwierającego <f orm> i kończy się znaczni­kiem zamykającym </f orm>. Pomiędzy tymi znacznikami znajdują się wszystkie inne polecenia dotyczące formularza. Podstawowe elementy formularza to pola tekstowe i listy oraz przycisk powodujący wysłanie danych.

Znacznik <form> może mieć następujące atrybuty:

* action="adres e-mail". To najważniejszy atrybut, ponieważ ustala adres poczty elektronicznej, na który będzie wysyłany formularz. Jako wartość tego atrybutu można również podać adres skryptu PHP lub CGI, który będzie wywoływany po zatwierdzeniu formularza. Dane wprowadzone do formularza zostaną przekazane do wybranego skryptu. Dodanie parametru ?subject=temat umożliwia podanie tytułu formularza.
* method="post". Określa on sposób (metodę) przekazania informacji do wskaza­nego skryptu lub pod wskazany adres. Może przyjąć jedną z dwóch wartości: post lub get. Wartością domyślną jest get, ale zaleca się stosowanie metody post.
* enctype=" ". Ustala on sposób kodowania przesyłanych danych. Wybierany bywa tylko wtedy, gdy formularz przekazywany jest metodą post. Dla metody get zawsze ma ustawioną wartość "application/x-www-form-urlencoded". Domyślnie dla metody post też przyjmuje tę wartość. Przy przesyłaniu pliku na serwer należy ustawić dla tego atrybutu wartość "multipart/form-data". Wysyłając zwykły formularz, atrybut ten można więc pominąć, natomiast przy wysyłaniu pliku należy nadać atrybutowi wartość "multipart/form-data".

Przykład:

<form action="mailto:twój\_adres\_poczty@domena" method="post" enctype="text/plain">

treść formularza

</form>

**Pole INPUT**

Pole INPUT służy do wprowadzania danych tekstowych do formularza i jest definiowane przy użyciu znacznika <input>.

<form action="mailto:nauka&gmail.com" method="post">

Podaj swój adres e-mail:

<input name="adres" value="">

<input type="submit" value="Wyślij">

</form>

Znacznik <input> ma następujące atrybuty:

* name — określa nazwę danego pola oraz nazwę zmiennej, do której zostanie pod­stawiona wprowadzona wartość. Należy zwrócić uwagę, aby wartość tego atrybutu była różna dla wszystkich pól formularza.
* value — określa domyślną wartość pola.
* size — określa liczbę znaków mieszczących się w widocznej części pola.
* maxlength — określa maksymalną liczbę znaków, jaką można wpisać do pola.
* type — określa typ pola.

Typy pól

Typ pola może przyjmować następujące wartości:

* type="text" — pole tekstowe.
* type="checkbox" — pole zaznaczenia opcji (można zaznaczyć opcję lub usunąć zaznaczenie).
* type="radio" — pole wyboru (z grupy opcji można zaznaczyć tylko jedną). Kliknięcie wartości powoduje zaznaczenie danej opcji, ale równocześnie usuwa zaznaczenie innej.
* type="password" — pole wprowadzania hasła (wprowadzane znaki nie są wi­doczne).  
  Po wpisaniu hasła pojawią się kropki zamiast wprowadzonej wartości. Należy jednak pamiętać, że mimo to dane wysyłane na serwer nie są szyfrowane.
* type="file" — pozwala dołączyć do formularza plik, który po zatwierdzeniu for­mularza zostanie przesłany na serwer. Przy tworzeniu pola typu file musi powstać pole, gdzie użytkownik może wpisać ścieżkę dostępu i nazwę pliku, oraz przycisk *Przeglądaj,* za pomocą którego użytkownik może wyszukać plik. Formularz zawie­rający pole tego typu musi być przekazywany metodą post i należy ustawić sposób kodowania danych na enctype="multipart/form-data"
* type="submit" — przycisk, który uruchamia wykonanie akcji zdefiniowanej w atrybucie action. Wartość przypisana do tego pola jest wyświetlana jako napis na przycisku
* type="reset" — przycisk, którego kliknięcie usuwa zawartość wszystkich pól formularza.
* type="hidden" — pole tekstowe niewidoczne na stronie Pole jest niewidoczne i niedostępne dla użytkownika, ale jego wartość jest przesyłana wraz z formularzem. Sie ma znaczenia, w którym miejscu formularza zostaną umieszczone pola ukryte. Mogą one służyć *do* zapisywania informacji z formularzy wcześniej wypełnionych przez użytkownika.
* type="button" — przycisk dowolnego przeznaczenia.

<input type="button" value="Przycisk">

Utworzony w ten sposób przycisk po kliknięciu nie spowoduje żadnego działania. » osiu jego wykorzystania należy zdefiniować program realizujący obsługę zdarzeń.

**Pole BUTTON**

Pole BUTTON służy do tworzenia w formularzu przycisków i jest definiowane za pomocą znacznika <button>. Ma działanie podobne do znacznika <input>, który dla atrybutu - przyjmuje jedną z wartości: button, reset lub submit.

<button type="submit" name="przycisk1">Wyślij</button>

**<button** type="reset" name="przycisk2">Usuń</button>

<**button** type="button" name="przycisk3">Powrót</button>

Różnica polega na tym, że przyciski tworzone za pomocą znacznika <button> mogą zawierać elementy ustawiające ich wygląd Znacznik <button> może mieć atrybut style powodujący zmianę wyglądu tekstu na przycisku.

**Etykieta**

Opis pola, który wyświetla się na formularzu, nie jest związany z tym polem. Używając znacznika <label>, można związać tekst etykiety z polem, którego dotyczy opis.

Definicję etykiety można zapisać następująco:

<label><input type="checkbox" name="nazwa" value="wartość">Etykieta pola</label>

lub

<input type="checkbox" name="nazwa" value="wartość" id="pole">

<label for="pole">Etykieta pola</label>

Po zdefiniowaniu takiego polecenia wyświetli się pole oraz skojarzona z nim etykieta. Efekt jest taki, że kliknięcie zarówno w obrębie pola, jak i etykiety powoduje zazna­czenie opcji lub usunięcie zaznaczenia. Tekst etykiety może zostać zdefiniowany przed definicją pola INPUT lub po tej definicji. Etykiety można przypisywać do wszystkich typów pól formularza.

Grupowanie elementów formularza

Jeżeli projektowany formularz zawiera wiele elementów różnego rodzaju, można te elementy pogrupować za pomocą znacznika <fieldset>. Powinien on znaleźć się za znacznikiem <form>, ale przed znacznikami definiującymi poszczególne elementy.

Sposób grupowania elementów:

<fieldset>

<legend align="kierunek">

</legend>

Definiowanie elementów grupy

</fieldset>

Do podpisania grupy elementów został zastosowany znacznik <legend>, w którym zdefiniowano nagłówek grupy (Wykształcenie:). Atrybut align="left" powoduje wyrównanie nagłówka do lewej. Atrybut align może przyjmować wartości: left, right, top i bottora. Znacznik <legend> może zostać pominięty.

Pole SELECT

Pole SELECT służy do wyświetlania w formularzu listy wartości i zachęca do wybrania jednej lub kilku z nich. Jest definiowane przy użyciu znacznika <select>, wewnątrz którego mogą wystąpić znaczniki <option>. Znacznik <select> definiuje listę, a po­szczególne wartości tej listy są definiowane za pomocą znacznika <option>.

Definicja listy jest następująca:

<select name="lista">

<option> Pierwszy element listy</option> <option> Drugi element listy</option> <option> Trzeci element listy</option> </select>

Dla znacznika <select> można używać następujących atrybutów:

* size="n" — n określa liczbę pozycji widocznych w polu SELECT. Jeżeli atrybut nie zostanie zdefiniowany, użytkownik nie będzie mógł wybrać żadnej wartości z listy.
* multiple — można na liście zaznaczyć więcej niż jedną pozycję.

Dla znacznika <option> można używać atrybutów:

* selected — pozycja, dla której został zdefiniowany ten atrybut, będzie automa­tycznie zaznaczona.
* value=" wartość" — określa wartość przypisaną pozycji listy.

Listy zagnieżdżone

Stosując znaczniki <select> i <option>, definiujemy listę wartości. Zdefiniowanej liście można nadać strukturę wielopoziomową, tworząc podmenu dla każdej jej wartości. Do tego celu służy znacznik <optgroup> z atrybutem label=" tytuł grupy", gdzie tytuł grupy to nazwa podmenu.

<optgroup label="tytuł">

</optgroup>

Dla każdego podmenu należy utworzyć przy użyciu znacznika <option> oddzielną listę wartości i przed nim wstawić znacznik <optgroup>.

Pole TEXTAREA

Do wprowadzenia w formularzu większej ilości tekstu można użyć pola TEXTAREA, które definiujemy za pomocą znacznika <textarea>. Znacznik <textarea> może zawierać następujące atrybuty:

* name="nazwa" — nazwa przypisana do pola;
* rows = "n" — liczba wierszy w polu tekstowym;
* cols = "n" — liczba kolumn (znaków) w polu tekstowym;
* wrap — w trakcie wprowadzania tekstu linie są dostosowywane do marginesów.   
  Atrybut wrap może przyjmować wartości:
* off — wyłączenie zawijania tekstu;
* physical — zawijanie tekstu na stronie i przesyłanie tekstu zawiniętego;
* virtual – zawijanie tekstu na stronie, ale przesyłanie bez zawijania

Nieaktywne pola formularza

Cały formularz można uczynić nieaktywnym poprzez zdefiniowanie atrybu­tu disabled. Można również dodać atrybut disabled do dowolnego elemen­tu formularza, powodując, że tylko ten element stanie się nieaktywny. Najlepsze efekty daje związanie atrybutu disabled ze znacznikiem <input>. Przykład: <input disabled name="nazwisko" value="Podaj nazwisko" size="25">.

Inną metodą może być uniemożliwienie wprowadzenia w polu zmian przez użytkownika. Do tego celu służy atrybut readonly, którego użycie spowoduje zablokowanie pola. Przy­kład: <input readonly name="nazwisko" value="Podaj nazwisko" size="25">. Atrybut readonly może być używany dla pól tekstowych, pól haseł, pól wyboru i blo­ków tekstu.

Klawisze skrótów

Do definiowanych w formularzu elementów można przypisywać klawisze skrótów. Zastosowanie kombinacji klawiszy spowoduje przejście do wybranego elementu. Do definiowania klawiszy skrótów służy atrybut accesskey umieszczony w definicji elementu, którego będzie dotyczył.

Usprawnienia w formularzach

Język HTML 5 wprowadza nowe znaczniki i atrybuty usprawniające obsługę formularzy. Aby automatycznie ustawić kursor w wybranym polu formularza, należy dodać atrybut autofocus.

Do wyłączania autouzupełniania w wybranych polach lub w całym formularzu służy atrybut autocomplete przyjmujący wartość on lub off.

Przykład 1.72

<input type="text" autocomplete="off">

Dla wybranego elementu formularza można zdefiniować listę proponowanych wartości wyświetlanych jako lista rozwijana. Użytkownik może wprowadzić własną wartość spoza listy.

Przykład 1.73

<input type="text" name="blok" list="jezyki">

<datalist id="jezyki">

<option> Angielski</option>

<option> Francuski</option>

<option> Włoski</option>

</datalist>

W języku HTML 5 znajdują się też elementy walidacji formularzy

Atrybut required służy do sprawdzenia, czy pole zostało wypełnione. Jeżeli takie pole formularza będzie puste, nie zostanie on zatwierdzony.

Przykład 1.74

<input type="text" required>

Inny element walidacji formularza to sprawdzanie, czy wartość wprowadzona do pola jest zgodna z zadeklarowanym wyrażeniem regularnym.

Przykład 1.75

<input type="text" pattern="[0-9]{2}-[0-9]{2}">

Oprócz typów pól dostępnych w HTML 4 język HTML 5 udostępnia grupę pól do wprowadzania danych określonego typu. Są to między innymi:

* <input type="email" > — do wprowadzania adresu e-mail,
* <input type="url" > — do wprowadzania adresu URL,
* <input type="number" min="l" max="7" step="l" value="2"> — do wprowadzania liczby całkowitej z podanego przedziału,
* <input type="date" value="2013-06-01"> — do wprowadzania daty,

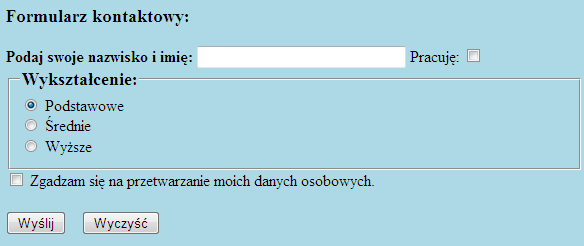
• <input type="color" value="#F0F0F0"> — do wprowadzania koloru.

Podsumowanie

Z formularzami na stronach internetowych spotykamy się bardzo często. Przy budowaniu formularzy konieczna jest znajomość tworzenia i działania omówionych elementów. Należy pamiętać, że aby w pełni wykorzystać formularze, nie wystarczy znać tylko język HTML. Niezbędna jest jeszcze znajomość języka przetwarzania po stronie serwera (ang. server-side) i języka po stronie klienta (ang. client-side). Formularz projektujemy w języku HTML, kod przetwarzający dane, które zostały wprowadzone do formularza, piszemy w języku przetwarzania po stronie serwera (np. PHP, Perl), a obsługę zdarzeń związa­nych z formularzem tworzymy w języku po stronie klienta (np. JavaScript, VBScript).

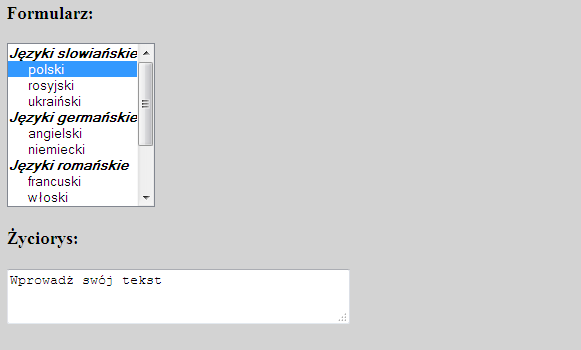
**Zadanie 13.**

Przygotuj poniższy formularz (patrz [Grupowanie elementów formularza](#grupowanie))



**Zadanie 14.**

Przygotuj poniższy formularz (patrz [Listy zagnieżdżone](#zagnieżdżanie)) Niewidoczne pozycje listy znajdziesz na kolejnej stronie.



Języki słowiańskie

polski

rosyjski

ukraiński

Języki germańskie

angielski

niemiecki

Języki romańskie

Francuski

włoski

hiszpański

**Zadanie 15.**

Przygotuj poniższy formularz

