**JAVASCRIPT – REFERAT**

Javascript (JS) jest obiektowym skryptowym językiem programowania, najczęściej stosowanym na stronach internetowych. Dzięki niemu zyskujemy możliwość sterowania różnymi elementami html’a (a także css’a) bez potrzeby zmiany kodu źródłowego strony. My zaprezentujemy rezultaty działania naszych skryptów, na przykładach okien dialogowych, formularza oraz ramek.

**OKNA DIALOGOWE**

JavaScript posiada trzy gotowe do wykorzystania okna dialogowe, z których można skorzystać przy interakcji z użytkownikami. Wszystkie trzy okna wyświetlają (stały) tytuł i komunikat, który można ustawić oraz wstrzymują wykonanie skryptu, dopóki użytkownik nie zareaguje.

• **alert** (komunikat) zawiera tylko przycisk OK. Tego okna używa się do zawiadomienia użytkownika, że coś się wydarzyło lub ma się wydarzyć. Komunikat może być zwykłym tekstem w cudzysłowie lub dowolnym wyrażeniem, które tworzy tekst.

• **confirm** (komunikat) zawiera przycisk OK i Cancel (anuluj) - zwraca wartość true lub false w zależności od tego, który przycisk został kliknięty. Tego okna używa się, gdy proponuje się użytkownikowi prosty wybór tak lub nie. Zwykle stosuje się w tekście funkcję if.

• **prompt (**komunikat, wartość\_domyślna) zawiera pole tekstowe oraz przycisk OK i Cancel. Zwraca ono wprowadzony tekst. Tego okna używa się do otrzymania informacji na stronie wtedy, gdy nie chce się korzystać z formularza. Można ustawić wartość\_domyślną wyświetlaną w polu tekstowym lub po prostu ” ” w celu wyczyszczenia tego pola. Jeżeli nie ustawi się wartości domyślnej, w polu tekstowym wyświetlany jest napis undefined (niezdefiniowane).

Aby przechwycić wprowadzony tekst, przypisać trzeba wywołanie tego okna do zmiennej:

ccnum = prompt (”Podaj numer swojej karty kredytowej”, ” ”)

Następnie możesz użyć ccnum gdziekolwiek w kodzie.

Ciekawostka 🡪 **Własne okna dialogowe**

Systemowe okna dialogowe są proste w użyciu, ale mają ograniczone możliwości. Nie można zmienić ich tytułu, wielkości czy używanej w nich czcionki. Jeżeli jednak ktoś chciałby zastosować ładniejsze okno dialogowe, według własnych upodobań i preferencji to można takie okno samemu stworzyć.

**PRZYKŁAD** (krótkie omówienie poniższego kodu)

**Pokaz okien dialogowych**

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title>SYSTEMOWE OKNO DIALOGOWE</title>

</head>

<body>

<script>

alert("Pokaz systemowych okien dialogowych")

if (confirm("Naprawdę chcesz to zobaczyć?"))

alert("Zaczynamy")

else

alert("Mimo wszystko i tak pójdziemy dalej")

ccnum = prompt("Podaj numer swojej karty kredytowej ;)", "")

</script>

</body>

</html>

**KRÓTKIE ĆWICZENIE :** Napisać kod wyświetlający okno dialogowe typu alert z tekstem „Uwaga za kilka tygodni kolokwium. Najwyższy czas na naukę ☺”

Otwieramy VS 2012 🡪File 🡪 New Project

Wybieramy ASP.NET Empty Web Aplication (nazywamy je odpowiednio np. „ćw 1”)

Następnie klikamy po prawej stronie w naszą aplikację i wybieramy Add 🡪 New Item i wybieramy HTML Page.

**FORMULARZE**

Formularze są kluczowe dla JavaScript. Stanowią one interakcyjną część HTML – a i powiązanie pomiędzy programującym, użytkownikiem strony i kodem JavaScript.

Sam formularz to niewiele więcej niż tylko pojemnik, ale może zawierać przyciski wykorzystywane do kontroli kodu oraz obszary tekstu i inne elementy, w których użytkownik może wpisać dane lub dokonać wyboru. Jeżeli chce się uzyskać informację zwrotną od użytkownika, najłatwiej osiągnąć to, korzystając z elementów formularza.

Krótkie omówienie podstawowych elementów formularza ( z prezentacji)

**POLE TEKSTOWE**

Do wprowadzania pół jak powyżej było mówione służy input które pozwolą użytkownikowi wpisać jakąś informację lub wybrać jedną z opcji.

Pole tekstowe tworzymy za pomocą definicji:

<input type="text" name="nazwa\_pola" value="wartość początkowa" size="rozmiar pola w znakach" maxlength="maksymalny rozmiar pola">

Type="text" to typ pola, w tym wypadku tekstowego. Name to nazwa pola, w naszym przykładzie "nazwisko".

Naturalnie podanie wartości początkowej value nie jest konieczne, wystarczy zostawić pusty cudzysłów - wartość początkowa ma jedynie sugerować użytkownikowi charakter wprowadzanych danych.

Size to wielkość widocznego pola w znakach (gdy zawartość pola będzie dłuższa, będzie ono "przewijane" w prawo), natomiast maxlength to maksymalna liczba znaków, jaką można wprowadzić.

**POLE OPCJI**

Pole opcji (przełącznika) jest polem, w którym użytkownik strony wybiera jedną z dostępnych możliwości

W polu stosowany jest type="radio", który wprowadza tzw. przycisk radiowy. Użytkownik będzie wybierał tylko jedną z możliwych wartości. Kliknięcie na jakiejś wartości powoduje jej zaznaczenie, ale zarazem usuwa zaznaczenie z innej. Typ ten stosujemy wtedy, gdy nie można podać kilku wartości, gdyż jest to nielogiczne. Trudno być na przykład jednocześnie kobietą i mężczyzną.

<p><input type="radio" name="plec" value="kobieta" checked="checked"> Kobieta</p>

<p><input type="radio" name="plec" value="mezczyzna"> Mężczyzna</p>

Nazwą pola name jest "plec", wartość value przybiera dwie możliwe wartości. Jeśli podamy przy jednej z możliwości checked="checked", pole to będzie domyślnie zaznaczone.

**POLE WYBORU**

Pole wyboru umożliwia dokonanie wyboru kilku możliwości jednocześnie, np. zaznaczenie kilku języków, którymi się włada, czy kilku używanych przeglądarek internetowych.

W polu stosowany jest type="checkbox", który wprowadza pole w postaci kwadratu. Kliknięcie w nim powoduje zaznaczenie pola, ponowne kliknięcie usuwa zaznaczenie. Jeśli zdefiniujemy kilka pól, użytkownik strony będzie mógł zaznaczyć parę z nich lub wszystkie, zależnie od treści.

<p><input type="checkbox" name="jezyk" value="angielski"> angielski</p>

<p><input type="checkbox" name="jezyk" value="francuski"> francuski</p>

...

Name jest nazwą pola - w tym wypadku jest to język. Każde kolejne pole zawiera tę samą nazwę, gdyż pytamy o ten sam rodzaj danych (tę samą zmienną).

Value zmienia się, zależnie od języka. W kolejnych wierszach musimy wpisać różne języki.

Za poleceniem input wpisujemy jeszcze raz nazwę języka, aby była widoczna dla użytkownika.

**LISTA WYBORU**

Lista wyboru select służy do tworzenia rozwijanych menu, zawierających kilka opcji. Użytkownik strony może wybrać jedną z nich lub kilka. Składnia jest następująca:

<div>

<select name="nazwa\_listy" size="liczba wyświetlanych pozycji">

<option> pierwsza pozycja

<option> druga pozycja

<option> trzecia pozycja

...

</select>

</div>

Przykład kodu tworzącego formularz (omówienie) – przejście do VS 2012

Sercem naszego kodu JavaScript jest funkcja „check”, która sprawdza czy do formularza został wpisany adres email, jeżeli to nie nastąpiło dostajemy odpowiedni komunikat w formie alertu.

**ZNACZNIKI FORMULARZA**

Początek i koniec formularza jest zaznaczony za pomocą znaczników <FORM> i </FORM>. Znacznik <FORM> może posiadać wiele opcji.

Można skonfigurować formularz tam, aby po zakończeniu wypełniania go użytkownicy wysyłali e – mailem jego zawartość na konkretny adres e – mail.

Otwierający znacznik <FORM> musi być rozszerzony o słowa kluczowe METHOD i ACTION, łącznie z adresem na który dane z formularza mają zostać przesłane:

<FORM METHOD = post ACTION = mailto: adres\_email>

Dane gromadzone są główie przez znaczniki <INPUT … > Mają one szereg odmian, a ich cechy definiowane są przez opcję TYPE = ...

Znacznik < INPUT TYPE = text tworzy pojedynczą linię do wprowadzania tekstu. Wprowadzony tekst wymaga drugiej opcja NAME = … w celu określenia zmiennej, w której zostaną zapisane dane.

<INPUT TYPE = text NAME = imię>

Powyższy znacznik powoduje utworzenie zmiennej o nazwie imię. Powinna być wyświetlona na ekranie jako pusta linia do wprowadzania danych, o szerokości 20 znaków.

Wracając do funkcji ACTION aby dane zostały przesłane na podany w ACTION e – mail trzeba gdzieś w formularzu umieścić przycisk Submit.

<INPUT TYPE = Submit VALUE = ”Wyślij”>

Opcja TYPE = Submit definiuje, że jest to przycisk, który wysyła informację zwrotną.

Znacznik INPUT posiada jeszcze jedną, podobną opcję, Reset, która także tworzy przycisk, ale służy on do wyczyszczenia zawartości formularza.

<INPUT TYPE = Reset VALUE = ”Wyczyść formularz”>

Formularz zawiera różnego typu pola, które są wypełniane informacjami - np. imię i nazwisko, zamawiany towar, dalsze informacje o sobie, komentarz itd. Pola te są podzielone na kilka podstawowych grup:

input, z dodatkowymi parametrami type, name i value, służy do tworzenia pól, gdzie czytelnik strony może wpisać informację lub wybrać jakąś opcję.

select służy do tworzenia rozwijalnych list z opcjami, spośród których czytelnik wybiera interesujące go pozycje.

textarea jest poleceniem do tworzenia większego pola na dodatkowy komentarz czytelnika.

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<title></title>

<script>

function check(t) {

g = false;

if (t.email.value.length == 0) {

alert('Nie podałeś swojego emaila');

} else {

t.action = 'mailto:' + t.email.value;

g = true;

}

return (g);

}

</script>

<form enctype="text/plain"action="mailto:sylwia.filus@wp.pl?subject=Dane z formularza"method="post"

name="Formularz" onSubmit="return check(this)"

>

<div>

<p>Twój email: <input type="text" name="email" size="30" maxlength="40"

style="position: absolute; left: 150px;

background-color: cornsilk; font-style: italic; color: gray "/></p><br />

<p>Imię:<input type="text" name="Imię" size="30" maxlength="40"

style="position: absolute; left: 150px;

background-color: cornsilk; font-style: italic; color: gray "/></p><br />

<p>Nazwisko: <input type="text" name="Nazwisko" size="30" maxlength="40"

style="position: absolute; left: 150px;

background-color: cornsilk; font-style: italic; color: gray "/></p><br />

Płeć: <input type="radio" name="plec" value="K" /> Kobieta

<input type="radio" name="plec" value="M" /> Mężczyzna<br />

<p>Data urodzenia:</p>

<p>Dzień: <input type="text" name="Dzień" size="30" maxlength="40"

style="position: absolute; left: 150px;

background-color: cornsilk; font-style: italic; color: gray "></p>

<p>Miesiąc: <select name=”Miesiąc” size="5" </p><br />

<div>

<option> Styczeń

<option> Luty

<option> Marzec

<option> Kwiecień

<option> Maj

<option> Czerwiec

<option> Lipiec

<option> Sierpień

<option> Wrzesień

<option> Październik

<option> Listopad

<option> Grudzień

</select>

</div>

<p>Rok: <input type="text" name="Rok" size="30" maxlength="40"

style="position: absolute; left:150px; background-color: cornsilk; font-style: italic; color: gray "></p>

Upodobania: <select name=like>

<option value=1>Komputer

<option value=2>Film

<option value=3>Sport

</select> <br />

<input type="reset" value="Wyczyść formularz" /><br />

<input type="submit" value="Wyślij" />

</div>

</form>

</head>

<body>

</body>

</html>

**KRÓTKIE ĆWICZENIE:** dot. formularza

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<script>

function check(t) {

g = false;

if (t.imie.value.length == 0) {

alert('Nie podałeś swojego imienia');

} else {

t.action = 'mailto:' + t.email.value;

g = true;

}

return (g);

}

</script>

</head>

<body>

<form name=Dane\_osobowe onSubmit="return check(this)"

method=post action="mailto:nonamne">

Imię: <input type="text" name="imie">

Nazwisko: <input type="text" name="nazwisko">

<input type="submit" value="Wyślij">

</form>

</html>

**RAMKI**

Strona składająca się z ramek zbudowana jest z co najmniej 3 dokumentów:

dokumentu opisującego zestaw ramek (frameset)

pozostałych stron, które wchodzą w zestaw ramek

Każdą ramkę możemy traktować jak okno, w którym znajduje się dokument.

Kluczowe znaczenie ma para znaczników <frameset> </frameset>. Stanowią one, z dodatkowymi parametrami, ramy dla całej konstrukcji i między nimi są umieszczane bardziej szczegółowe znaczniki, dotyczące liczby ramek, podziału strony i wielkości ramek.

Aby konstrukcja mogła działać, konieczne jest podzielenie okna przeglądarki - w pionnie lub poziome. Poniższy kod (kontynuujemy nasz przykład) zawiera informację o podzieleniu okna na dwie kolumny. Służy do tego atrybut cols="x,y", umieszczany w otwierającym znaczniku frameset.

Sam podział na ramki to nie wszystko. Należy im jeszcze przypisać jakieś dokumenty, zazwyczaj w formacie HTML (choć mógłby to być na przykład plik w formacie PDF, który moglibyśmy obejrzeć za pomocą specjalnej wtyczki (plugin) wspomagającej przeglądarkę.

Aby przypisać ramce dokument, musimy podać źródło src dokumentu. Należy w tym celu użyć konstrukcji:

<frame src="nazwa\_pliku.htm" ...>

**Plik index.html**

<!DOCTYPE html>

<HTML>

<HEAD>

<TITLE>Przykład użycia ramek</TITLE>

<HEAD>

</HEAD>

<FRAMESET>

<FRAMESET COLS="\*, 45%">

<FRAME SRC= ramka1.html NAME = "ramka1">

<FRAME SRC= ramka2.html NAME = "ramka2">

</FRAMESET>

<BODY>

</BODY>

</HTML>

Aby zmienić kolor ramki należy zastosować w JavaScript

Instrukcja parent określa dokument definiujący zestaw ramek, czyli stojący najwyżej w zestawie ramek.

Aby odwołać się do dowolnej ramki z naszego zestawu skorzystamy z instrukcji:

*parent.ramka2.document.bgColor = kolor;*

Kolory są właściwościami obiektu document, więc odwołując się do nich trzeba w nazwie dołączyć słowo”document” (bgColor oznacza kolor tła)

**Plik ramka1.html**

<!DOCTYPE html>

<HTML>

<TITLE>Ramka1</TITLE>

<HEAD>

<SCRIPT LANGUAGE = "JavaScript">

function zmienKolor(kolor) {

parent.ramka2.document.bgColor = kolor;

}

</SCRIPT>

</HEAD>

<BODY>

<H2>Kliknij na przycisk, aby zmienić kolor w drugiej ramce</H2>

<FORM>

<INPUT

TYPE = "button"

NAME = "przycisk1"

VALUE = "czerwony"

onClick = "zmienKolor('red')";

>

<BR><BR>

<INPUT

TYPE = "button"

NAME = "przycisk2"

VALUE = "zielony"

onClick = "zmienKolor('green')";

>

<BR><BR>

<INPUT

TYPE = "button"

NAME = "przycisk3"

VALUE = "niebieski"

onClick = "zmienKolor('blue')";

>

</BODY>

</HTML>

**Plik Ramka2.html**

<!DOCTYPE html>

<HTML>

<TITLE>Ramka1</TITLE>

<HEAD></HEAD>

<BODY>

<H2>Tutaj będą się zmieniać kolory</H2>

</BODY>

</HTML>