

HTML z elementami JavaScript

Skrypt ćwiczeniowy



Wydanie I
Gdańsk, listopad 2000

Całość skryptu oraz żadna część nie mogą być modyfikowane, kopiowane oraz publikowane w formie elektronicznej
bądź drukiem bez zezwolenia autora.

Podręcznik ten stanowi integralną część materiałów edukacyjnych i nie jest samodzielnie używany. Pełne wykorzystanie
zawartych w nim informacji oraz realizacja ćwiczeń może być prowadzona przez aktywnych uczestników zajęć
ćwiczeniowych.

CZEŚCI	3
Wstęp	3
Zasady tworzenia stron WWW	3
Uwagi praktyczne	3
Coto jest znacznik?	3
Budowa szkieletu dokumentu	4
Tytuły	4
Akapit	4
Poziome linie	4
Listy wypunktowane i numerowane	5
Styl czcionki	5
Hiperłącza, czyli odsyłacze	5
Ćwiczenie nr 1 – Tytuły, listy, odsyłacze	6
Ćwiczenie nr 2 – atrybuty czcionek	7
Grafika	8
Kolor tła strony	8
Ćwiczenie nr 3 – wstawianie tła strony i obrazów graficznych	8
CZEŚCII	9
Tworzenie stron z ramkami	9
Konstruowanie ramek	9
Ćwiczenie nr 4 – ramki pionowe	10
Ramki kolumnowo-wierszowe	11
Ćwiczenie nr 5 – ramki kolumnowo-wierszowe	11
Ramki i spisy treści	12
Ćwiczenie nr 6 – ramki z spisem treści	12
Animacja Marquee, czyli przesuwający się tekst	14
Dźwięk w tle dokumentu	14
Tabele	15
Ćwiczenie nr 7 – tabela	16
CZEŚCIII	16
Umieszczanie skryptów JavaScript na stronach WWW	16
Jak umieścić skrypt JavaScript w dokumencie HTML?	16
CZEŚCIV – ZADANIA SPRAWDZAJĄCE	17
ZADANIE NR 1	17
ZADANIE NR 2	20
DODATEKA – ZNACZNIKI HTML	23
Znaczniki podstawowe	23
Znaczniki fizyczne	23
DODATEKB – WYBRANE ELEMENTY JAVASCRIPT	24
Umieszczanie skryptów w dokumencie HTML	24
Deklaracje zmiennych	24
Operatory arytmetyczne, porównania i logiczne	25
Operatory arytmetyczne	25
Operatory porównania	25
Operatory logiczne	25
Funkcje	25
Instrukcja wyboru	26
Instrukcja pętli	26
Formularze	27
Przydatne funkcje standardowej języka JavaScript	27
Funkcje daty i czasu systemowego	27
Funkcje: alert, confirm, prompt obsługujące okienka dialogowe	28
Funkcje zwracające dane o przeglądarce internetowej	28
DODATEKC – PRZYKŁADY SKRYPTÓW	28
Animowany tekst w pasku statusu	28
Pole tekstowe z zegarkiem	29
Rozwijanie menu w ramce	30
DODATEKD – Podstawy JavaScript	30
Wstęp	30
Dlaczego używamy skryptów JS na stronach WWW?	31

Jak umieścić skrypt JS w dokumencie HTML?	31
Podstawowe elementy JS	31
Obiekty i ich właściwości oraz metody	32
Predefiniowane obiekty JS	33
Podstawowe konstrukcje używane w JS	36
Komentarze	36
Okna komunikatów	36
Instrukcje podstawiania	36
Instrukcje warunkowe if	36
Deklaracje zmiennych	37
Instrukcje etli	37
Definicje funkcji	37
Przykładowe skrypty	38
Superanimacja w pasek statusu	40
Zadania ćwiczeniowe	43
Uwagi końcowe	46
LITERATURA	47

CZĘŚĆ I

Wstęp

Strona WWW czyli dokument hipertekstowy HTML jest zwykłym plikiem tekstowym, w którym znajdują się polecenia języka HTML. Dokument taki można utworzyć za pomocą dowolnego edytora tekstu np. Notatnika, jednak jest to metoda zbyt uciążliwa. Dlatego drugą metodą jest używanie specjalizowanych edytorów, które ułatwiają tworzenie dokumentu hipertekstowego. Najbardziej dostępnymi programami do tworzenia stron WWW są np. **Kreator stron WWW** w pakiecie MS-Office 97 Professional, **FrontPage Express** w systemie Windows 98, a ostatnio jest to program **Pajaczek** zalecany przez Ministerstwo Edukacji Narodowej do używania w szkołach.

W niniejszym skrypcie zajmijmy się programowaniem stron WWW w oparciu o możliwości programu Pajaczek.

Zasady tworzenia stron WWW

1. Każdy temat przyporządkowujemy jednej stronie.
2. Struktura witryny powinna być hierarchiczna.
3. Każda strona składa się z listy tematów (form menu) po lewej stronie okna oraz treści wybranego tematu (poprawiej stronie okna).
4. Wybór tematów z listy prowadzi użytkownika do strony opisującej dany temat lub listy podtematów.
5. Pamiętaj o korektach językowych tekstów.

Uwagi praktyczne

1. Jeden temat – jedna strona. Kilka nie będzie przeładowany, co strona będzie się ładować szybciej niż jedna ogromna strona.
2. Wiersze tekstu nie powinny być zbyt długie – są one trudne do odbioru, szczególnie gdy dostępujemy do nich z dala.
3. Oddzielaj fragmenty tekstu znacznikiem
. Pozwoli to na wprowadzenie drobiny wolnego miejsca.
4. Zachowaj umiar w korzystaniu z pogrubienia i kursywy. Czcionka małym rozmiarze, pisaną kursywą, jest całkowicie nieczytelna.
5. Unikaj stosowania czcionki podkreślanej, ponieważ tekst podkreślony wygląda jak hiperłącze.
6. Nie wyróżniaj zwykłego tekstu za pomocą znacznika nagłówka.
7. Definiuj łącza unikając kreślenia „tutaj”.
8. Nie twórz niepotrzebnych bloków nadmiarowych obrazów kolorowych.
9. Parametry rysunków (wielkość, ilość kolorów) w formacie GIF powinny zostać tak zoptymalizowane, aby strona nie ładowała się zbyt wolno, co może zniechęcić potencjalnego czytelnika.
10. Testuj swoją stronę staraj się być zadowolony z nich rozdzielczości ekranu, tak aby dobra proporcja optymalnej wielkości tekstu i rysunków dla różnych rozdzielczości – niekażdy posiada najnowszego monitora.
11. Sprawdź ortografię! Nawet ortograficzny błąd może być fatalny.

Czym jest znacznik?

Polecenie (znacznik, tag) HTML jest specjalnym ciągiem znaków objętym nawiasami prostymi.

Wszystkie polecenia (znaczniki) sterujące wyglądem i zachowaniem strony WWW. Znaczniki w przeglądarce WWW niesą wyświetlane. Są one jedynie interpretowane, tzn. pozwalają sformatować tekst.

Budowa szkieletu dokumentu

Cały dokument powinien być objęty parą znaczników `<HTML>` `</HTML>`. Między nimi powinny znaleźć się parę znaczników `<HEAD>` `</HEAD>`, które stanowią ramy dla informacji ogólnych. Ponadto informacje powinny być objęte kolejnymi znacznikami `<BODY>` `</BODY>` (patrz poniższy przykład).

```
<HTML>
<HEAD>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-2">
<TITLE>Moja strona WWW </TITLE>
</HEAD>
<BODY>
właściwa treść (ciało) dokumentu
</BODY>
</HTML>
```

Stosowanie strony kodowej ISO-8859-2 (czyli ISO Latin2) powoduje, że nasz dokument jest zgodny z Polską Normą.

Tytuły

Doprowadzanie tytułów w dokumencie służy poleceniu `<Hn>` `</Hn>`. Znak H oznacza „header”, natomiast cyfra to stopień tytułu (mamy ich sześć). Polecenie wprowadzające tytuł stopnia pierwszego składa się z znacznika otwierającego `<H1>` i zamykającego `</H1>`. A oto przykład:

```
<H1> Witam na mojej stronie </H1>
<H3> Moje zainteresowania </H3>
<H3> Moje sukcesy </H3>
```

Akapit

Do tworzenia akapitów służy polecenie `<P>`, które wstawia interlinię między poszczególnymi fragmentami tekstu. Pamiętaj o stosowaniu znacznika zamykającego `</P>`.

Przykład:

```
<P> Wycieczki rowerowe </P>
<P> Biologia i zoologia </P>
```

Jeśli chcemy przełamać tekst, nie wprowadzając przy tym znaku końca akapitu, to stosujemy pojedyncze polecenie `
`, które przenosi tekst do nowej linii, przy tym nie wprowadzając dodatkowej interlinii.

Przykład:

```
Wycieczki rowerowe <BR>
Biologia i zoologia <BR>
```

Poziome linie

Aby wstawić poziomą linię, używamy polecenia `<HR>`.

```
....
<BODY>
<H1> Witam na mojej stronie </H1>
<HR>
<H3> Moje zainteresowania </H3>
<P> Wycieczki rowerowe </P>
<P> Biologia i zoologia </P>
<HR>
<H3> Moje sukcesy </H3>
<P> Imię i cew Ogólnopolskich Zawodach Strzeleckich </P>
</BODY>
....
```

Listy wypunktowane i numerowane

Aby utworzyć listę wypunktowaną, należy użyć polecenia ``. Poszczególne pozycje listy wprowadzamy za pomocą polecenia ``

Przykład:

```
<UL>
<LI>Wycieczki rowerowe </LI>
<LI>Biologia i zoologia </LI>
<LI>Muzyka rockowa </LI>
<LI>Rozmowy internetowe </LI>
</UL>
```

Aby utworzyć listę numerowaną, należy użyć polecenia ``. Poszczególne pozycje listy wprowadzamy za pomocą polecenia ``

Przykład:

```
<OL>
<LI>Imię i nazwisko </LI>
<LI>Założenie i siedziba </LI>
</OL>
```

Stylizacja

Stylizacja: **pogrubienie** (bold), *pochylenie* (italic) i podkreślenie (underlined) ustawiamy za pomocą następujących par znaczników: ``, `<I></I>`, `<U></U>`.

```
<B>Wycieczki rowerowe </B>
<I>Biologia i zoologia </I>
<U>Muzyka rockowa </U>
```

Hiperłącza, czyli odsyłacze

Odsyłacz to prosta metoda odwołania do innych stron, które znajdują się w światowej sieci. W trakcie przeglądania strony WWW klikamy w taki odsyłacz, który automatycznie zostaje prześwietlony i przenosi nas na stronę WWW – w ten sposób za pomocą kliknięcia myszką na odsyłaczach możemy przetranszować do różnych miejsc w Internecie.

Przykładowy odsyłacz możemy wygenerować następująco:

```
<A HREF="http://www.arminius.gdansk.wp.pl">ARMINIUS.C. </A>
```

na ekranie przeglądarki:

[ARMINIUS.C.](http://www.arminius.gdansk.wp.pl)

Inne przykłady:

```
<a href="http://www.pg.gda.pl">Politechnika Gdańska</a>
<a href="http://www.univ.gda.pl">Uniwersytet Gdański</a>
<a href="http://www.gdansk.gda.pl">Nasze Miasto Gdańsk</a></p>
<a href="ei.html">Elementy informatyki</a></p>
```

Ćwiczenie nr 1 – Tytuły, listy, odsyłacze

Wykorzystując edytor stron HTML Pajązek (lub Notatnik w Windows) doplikuj nazwę **Moja strona WWW.html** wpisując następujące znaczniki:

```
<HTML>
<HEAD>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-2">
<TITLE>Moja strona WWW </TITLE>
</HEAD>
<BODY>
<H1>Witam na mojej stronie </H1>
<HR>
<H3>Moje zainteresowania </H3>
<UL>
<LI><B>Wycieczki rowerowe </B></LI>
<LI><I>Biologia i zoologia </I></LI>
<LI><U>Muzyka rockowa </U></LI>
</UL>
<HR>
<H3>Moje sukcesy </H3>
<OL>
<LI>Imię i nazwisko w Ogólnopolskich Zawodach Strzeleckich </LI>
<LI>Założenie sieci dostępu do Internetu </LI>
</OL>
<HR>
<H3>Firma mojej rodziny: </H3>
<A href="http://www.arminius.gdansk.wp.pl">ARMINIUS.S.C. </A>
</BODY>
</HTML>
```

Powyższy kod HTML wpisał w edytorze internetowym i zbiegł się z oczekiwaniami

Witam na mojej stronie

Moje zainteresowania

- **Wycieczki rowerowe**
- *Biologia i zoologia*
- Muzyka rockowa

Moje sukcesy

1. I miejsce w Ogólnopolskich Zawodach Strzeleckich z kuszy
2. Założenie sieci osiedlowej z dostępem do Internetu

Firma mojego taty:

[ARMINIUS S.C.](http://www.arminius.gdansk.wp.pl)

Ćwiczenie nr 2 – atrybuty czcionek

Teraz w pliku o nazwie **MojaStronaWWW.html** zmień kod HTML tak aby wyglądał następująco:
(dodane zostały znaczniki **** i **** - kolor czcionki czerwony)

```
<HTML>
<HEAD>
<meta http-equiv="content-type" content="text/html; charset=iso-8859-2">
<TITLE>MojaStronaWWW </TITLE>
</HEAD>
<BODY>
<H1>Witam na mojej stronie </H1>
<FONT SIZE="3"> Teraz przedstawię... </FONT>
<FONT SIZE="5"> moje sukcesy... i zainteresowania.... </FONT>
<H3><FONT COLOR="#FF0000"> Moje zainteresowania </FONT></H3>
<UL>
<LI><B>Wycieczki rowerowe </B></LI>
<LI><I>Biologia i zoologia </I></LI>
<LI><U>Muzyka rockowa </U></LI>
</UL>
<H3>Moje sukcesy </H3>
<OL>
<LI> I miejsce w Ogólnopolskich Zawodach Strzeleckich z kuszy </LI>
<LI> Założenie sieci osiedlowej z dostępem do Internetu </LI>
</OL>
<HR>
<H3>Firma mojego taty: </H3>
<A HREF="http://www.arminius.gdansk.wp.pl">ARMINIUS S.C. </A>
</BODY>
</HTML>
```

Witam na mojej stronie

Teraz przedstawi ę... moje sukcesy... i zainteresowania....

Moje zainteresowania

- Wycieczki rowerowe
- *Biologia i zoologia*
- Muzyka rockowa

Moje sukcesy

1. I miejsce w Ogólnopolskich Zawodach Strzeleckich z kuszy
2. Założenie sieci osiedlowej z dostępem do Internetu

Firma mojego taty:

[ARMINIUS S.C.](#)

Grafika

Oczywi ście oprócz testów strony WWW zawieraj ą grafik ę, któr ą czynie bardziej atrakcyjnymi. Jednak grafik ę nale ży stosowa ć z umiarem. Tak zwane graficznie ozdoby, zw łaszcz ą g dy nie s ą ż ądnymi i istotnymi tre ściami niepotrzebnym, „wodotryskiem”. Na stronach WWW s ą stosowane dwa formaty bitowych plików graficznych: GIF i JPG (JPEG) – któr ę s ą kompresowane, aby zajmowa ły mniej miejsca co z tego wynika – ich przesy łanie z serwera WWW do komputera odbiorcy strony trwa ł ą znacznie krócej.

Grafik ę mo żemy wprowadzamy za pomoc ą polecenia:

``

Uwaga: g dy plik graficzny (obrazek) znajdujesi ę w tym samym katalogu co dokument, wystarczy poda ć jego nazw ę. Natomiast g dy plik graficzny s ą przechowywane w odr ębnym katalogu, nale ży nazw ę pliku poprzedzi ć ście żką dost ępu.

Naprzyk ład:

``

lub

``

Kolor t łastrony

Kolor t łastrony mo żna ustawi ć za pomoc ą parametru BGCOLOR w znaczniku BODY (`<BODY BGCOLOR= "kolor">`).

Ćwiczenie nr 3 – wstawianie t łastrony i obrazów graficznych

Utwórz plik o nazwie **Grafika1.html** w któr ą umie śc ą nast ępuj ący kod HTML:

```
<HTML>
<HEAD>
```



```

<metahttp-equiv="content-type"content="text/html;c  harset=iso-8859-2">
<TITLE>GrafikanastronieWWW  </TITLE>
</HEAD>
<BODYBGCOLOR="gray">
<H1>Witamna mojej stronie  </H1>
<FONTSIZE="3"> Tutajb ędzieobrazek...</ FONT>
<IMGSRC=" C:/Windows/WINLOGO.GIF">
</BODY>
</HTML>

```

Powyższy kod HTML przegł ądarce internetowej ędziewygl ąda nast ępująco:



Zapomoc ą przedstawionych znaczników mo żemy ju żbudowa ęprostestrony WWW.

CZĘŚĆ II

Tworzenie stron z ramkami

Celemu życia ramek jest umieszczenie w jednej ramce spisu treści w drugiej - wywoływanych dokumentów (tak zawsze istnieje ścieżka do bieżącej strony HTML). Wszystkie strony WWW muszą być do niego odpowiednie, szczególnie w pozycjach spisu treści. Ramka przeznaczona na dokumenty, aby dynamicznie powiększać lub zmniejszać zawartość (jeśli nie została zablokowana przez autora strony).

Ramki ułatwiają poruszanie się w wielostronicowych dokumentach HTML, lecz powodują wolniejsze wczytywanie stron. Błąd w konstrukcji dokumentu z wieloma ramkami może spowodować dziwne zachowanie strony.

Przeglądarki Netscape i Internet Explorer potrafią obsługiwać, jednak ramki nie są zgodne z standardem HTML 3.2 i dlatego nie zawsze są stosowane przez autorów stron.

Konstruowanie ramek

W szczególności definiujemy liczbę, wielkość i pozycję ramek oraz zachowanie innych dokumentów w momencie aktywowania ączy hiperlinków. Pozostałe strony, wchodzące w skład całego, wielostronicowego dokumentu, są stronami podrzędnymi. Uwaga: na podstawie strony nie powinno być żadnych znaczników poza blokiem nagłówkowym <TITLE> oraz definicją samych ramek. Znaczniki <BODY></BODY> są umieszczane w ramach <NOFRAMES> są interpretowane przez przeglądarki i nie akceptują ramek. Pomiędzy znacznikami <BODY></BODY> jest umieszczana "bezzramkowa" wersja strony.

Paraznaczników <FRAMESET></FRAMESET> jest szkieletem całej konstrukcji i obejmuje bardziej szczegółowe znaczniki, dotyczące liczb ramek, podziału strony i wielkości ramek. Z kolei paraznaczników <NOFRAMES></NOFRAMES> służy do wprowadzenia elementów strony, które będą widoczne dla posiadacza przeglądarki innej niż Navigator i Internet Explorer. W przypadku używania Navigatora lub Internet Explorera, wszelkie informacje są ignorowane.

Znacznik **<FRAME>** służy do wprowadzenia konkretnej ramki wraz z parametrami, ustalającymi jej zawartość i zagęszczenie stron oraz decydującymi o tym, czy ramka zawiera suwak, a także o definiowaniu jej min i max.

Parametry FRAMESET

Parametr **COLS="x,y"**, umieszczany w znaczniku **FRAMESET** służy do liczenia i określania szerokości ramek (podajemy jej w pikselach lub w procentach szerokości strony) np. **<FRAMESET COLS="18%,82%">**.

Parametr **<FRAMESRC="nazwa_pliku.htm">** przypisuje ramce dokument jako źródło dokumentu, na którym będzie wyświetlana zawartość pliku.

Parametr **<FRAMESCROLLING=yes>** powoduje wyświetlenie ramki z suwakami.

Parametr **<FRAMESCROLLING=no>** powoduje wyświetlenie ramki bez suwaków.

Parametr **<FRAMESCROLLING=auto>** powoduje wyświetlenie ramki z suwakami tylko wtedy, gdy dokument będzie większy niż ramka.

Parametr **<FRAME NAME="nazwa_ramki">** służy do nadawania unikalnej nazwy dla ramki w celu ustawienia punktu odniesienia dla odsyłaczy.

Jeśli użyjemy parametru **<FRAMENORESIZE>** to zabezpieczymy ramkę przed zmianą rozmiaru.

Domyślna postać definicji ramki wygląda następująco:

<FRAMESCROLLING=yes NAME="nazwa_ramki" SRC="nazwa_pliku. htm">

Aby zmieniła postać kolumnowa ramek na wierszową, należy w znaczniku **<FRAMESET COLS="18%,82%">** zamienić następujący znacznik **<FRAMESET ROWS="18%,82%">**.

Ćwiczenie nr 4 - ramki pionowe

Wykonaj poniższe zadania w WWW, wykorzystując poprzednio wykonaną stronę i sprawdź, czy jej wygląd w przeglądarce jest podobny do przedstawionego w treści ćwiczenia.

Przykład pliku **RAMKI1.html**

```
<HTML>
<TITLE>Ramki1 </TITLE>
<FRAMESET COLS="18%,82%">
<FRAMESCROLLING =&auto;yes" NAME="GrafikanaWWW" SRC="Grafika1.html">
<FRAMESCROLLING="auto" NAME="MojastronaWWW" SRC="MojastronaWWW.html">
</FRAMESET>
<NOFRAMES>
<BODY BGCOLOR="gray">
<A HREF="Grafika1.html">GrafikanaWWW </A>
<A HREF="MojastronaWWW.html">MojastronaWWW </A>
</BODY>
</NOFRAMES>
</HTML>
```

Wygląd pliku **RAMKI1.html** w przeglądarce:



Ramkikolumnowo-wierszowe

Zanim przystąpimy do następnego ćwiczenia stworzymy sześć prostych stron WWW o następujących nazwach: Strona1.html, Strona2.html, Strona3.html, Strona4.html, Strona5.html, Strona6.html.

Najpierw dzielimy stronę na wiersze, a następnie w drugim i trzecim wierszu deklarujemy dodatkowy podział na kolumny - trzy w drugim i dwie w trzecim. Odpowiednik kodu w pliku **Ramki2.html** ma następującą postać:

```
<HTML>
<TITLE>Ramki2</TITLE>
<FRAMESET ROWS=25,*,25>
<FRAME NAME="Strona1" SRC=Strona1.html>
<FRAMESET COLS=30%,*,30%>
<FRAME NAME="Strona2" SRC=Strona2.html>
<FRAME NAME="Strona3" SRC=Strona3.html>
<FRAME NAME="Strona4" SRC=Strona4.html>
</FRAMESET>
<FRAMESET COLS=50%,50%>
<FRAME NAME="Strona5" SRC=Strona5.html>
<FRAME NAME="Strona6" SRC=Strona6.html>
</FRAMESET>
</FRAMESET>
<NOFRAMES>
<BODY BACKGROUND="gray">
<A HREF="Strona1.html">Strona1</A>
<A HREF="Strona2.html">Strona2</A>
<A HREF="Strona3.html">Strona3</A>
<A HREF="Strona4.html">Strona4</A>
<A HREF="Strona5.html">Strona5</A>
<A HREF="Strona6.html">Strona6</A>
</BODY>
</NOFRAMES>
</HTML>
```

Ćwiczenie nr 5 - ramkikolumnowo-wierszowe

Utwórz w pliku **Ramki2.html** przedstawioną wyżej konstrukcję i aby utrzymać tę postać strony w przeglądarce.



Ramki i pstryki

Aby wykonać witrynę internetową zawierającą cztery ramki – jedną kolumnową po lewej stronie zawierającą spis stron WWW, oraz drugą ramkę po prawej stronie (oczywiście znajdziemy nazwę głównej ramki) należy użyć znaczników:

```
<FRAMESET COLS="25%,*">-25% przeglądanego obszaru przeznaczamy na pstryki
<FRAME SCROLLING="auto" NAME="SPIS TREŚCI" SRC="SpisTresci.htm">-ramka po lewej stronie
<FRAME SCROLLING="auto" NAME="Main" SRC="Main.htm">-do ramki Main będą ładowane
poszczególne strony wybierane za pomocą odsyłaczy w pstrykach.
```

Spis treści (ramka po lewej stronie) to prosta lista zawierająca odsyłacze do odpowiednich stron. Kliknięcie na odsyłacz spowodowało załadowanie wskazanej w parametrze **HREF** strony, ale zarazem ładowała się również ramka o nazwie **Main**.

Stworzymy więc następującą konstrukcję:

```
<A HREF="Strona1.html" TARGET="Main">Strona1</A>
<A HREF="Strona2.html" TARGET="Main">Strona2</A>
<A HREF="Strona3.html" TARGET="Main">Strona3</A>
<A HREF="Strona4.html" TARGET="Main">Strona4</A>
<A HREF="Strona5.html" TARGET="Main">Strona5</A>
<A HREF="Strona6.html" TARGET="Main">Strona6</A>
```

Parametr **HREF="nazwa_ładowanej_strony"** dotyczy nazwy pliku, w którym jest zapisana dana strona WWW.

Parametr **TARGET="nazwa_ramki"** dotyczy docelowego miejsca, do którego ma być łaadowana strona.

Ćwiczenie nr 6 – ramki i pstryki

W pliku **Main.html** utwórz stronę, która będzie pełniła rolę twojej strony głównej w serwisie internetowym.

```
<HTML>
<HEAD>
<TITLE>Main</TITLE>
</HEAD>

<BODY BGCOLOR="gray">
```

```

<H1>STRONAGŁÓWNA</H1>
<IMG SRC="INFORMAlogo.jpg">
<IMG SRC="C:/Windows/WINLOGO.GIF">
</BODY>
</HTML>

```

Następnie utwórz w pliku **Ramki3.html** przedstawioną poniżej konstrukcję:

```

<HTML>
<HEAD>
<TITLE>RAMKI3</TITLE>
</HEAD>

<FRAMESET COLS="20%,*">
<FRAME SCROLLING="auto" NAME="SPISTRE ŚCI" SRC="SpisTresci.html">
<FRAME SCROLLING="auto" NAME="Main" SRC="Main.html">
<H1>STRONAZRAMKAMI</H1>
</FRAMESET>

<NOFRAMES>
<BODY BACKGROUND="gray">
<H1>STRONABEZRAMEK</H1>
<UL>
<LI><A HREF="Strona1.html" TARGET="Main">Strona1</A>
<LI><A HREF="Strona2.html" TARGET="Main">Strona2</A>
<LI><A HREF="Strona3.html" TARGET="Main">Strona3</A>
<LI><A HREF="Strona4.html" TARGET="Main">Strona4</A>
<LI><A HREF="Strona5.html" TARGET="Main">Strona5</A>
<LI><A HREF="Strona6.html" TARGET="Main">Strona6</A>
</UL>
</BODY>
</NOFRAMES>
</HTML>

```

Następnie utwórz w pliku **SpisTresci.html** przedstawioną poniżej konstrukcję:

```

<HTML>
<HEAD>
<TITLE>Spis Treści</TITLE>
</HEAD>

<BODY BACKGROUND="gray">
<H3>SPISTRE ŚCI</H3>

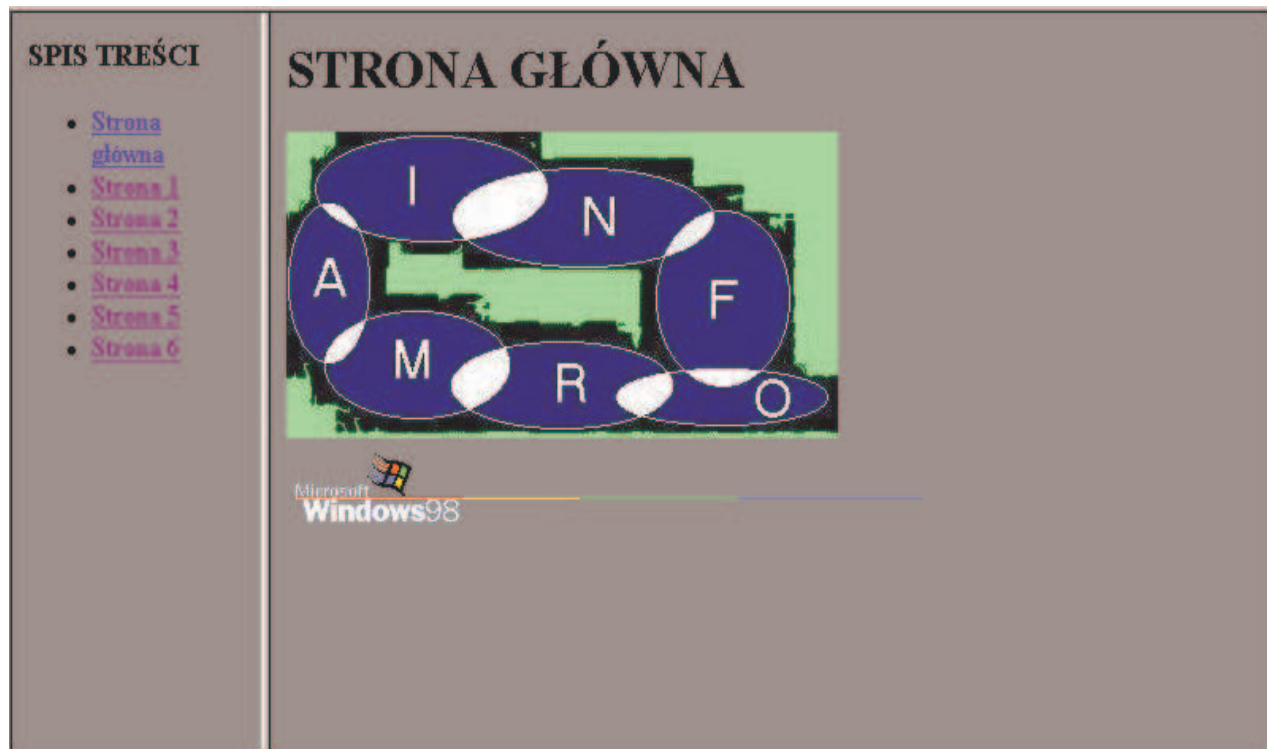
<UL TYPE="square">
<FONT SIZE="3"><B>
<LI><A HREF="Main.html" TARGET="Main">Strona główna</A>
<LI><A HREF="Strona1.html" TARGET="Main">Strona1</A>
<LI><A HREF="Strona2.html" TARGET="Main">Strona2</A>
<LI><A HREF="Strona3.html" TARGET="Main">Strona3</A>
<LI><A HREF="Strona4.html" TARGET="Main">Strona4</A>
<LI><A HREF="Strona5.html" TARGET="Main">Strona5</A>
<LI><A HREF="Strona6.html" TARGET="Main">Strona6</A>
</B></FONT>
</UL>

```

```
</BODY>
</HTML>
```

W pliku **INFORMAlogo.jpg** znajdziesz rysunek. Je śl ichceszmo żesz utworzy ć swój własny rysunek. Pamiętaj jednak o zmianie parametru `` w pliku **MAIN.HTML**.

Wygląd witryny internetowej składa się z plików: **RAMKI3.HTML**, **MAIN.HTML**, **SpisTresci.HTML**, **Strona1.html**, **Strona2.html**, **Strona3.html**, **Strona4.html**, **Strona5.html**, **Strona6.html**. Każde przypomina poniższy rysunek:



Sprawdź czy po kliknięciu na odpowiednie linki (tzw. linki) w prawym oknie zostanie wyświetlona odpowiednia strona WWW.

Jeśli wszystkie działania jak to sobie zaplanowali śmy, to znaczy żew przyszło ścib ędzieszosi ągać sukcesy w sztuce budowania stron internetowych. Gratulacje!!! :-)))

Animacja Marquee, czyli przesuwający się tekst

Animacja to efektowny napis, który „pływa” w poprzek ekranu. Definiowanie takiego animowanego napisu wygląda następująco:

```
<MARQUEE>To jest animowany tekst</MARQUEE>
```

Dźwięk w tle dokumentu

Aby umieścić w tle strony (dokumentu HTML) dźwięk (podkład muzyczny) należy po prostu żyć się następującą konstrukcją w sekcji **HEAD**:

```
<HEAD>
```

```
...
```

```
<BGSOUND SRC="pinkpant.mid" LOOP="1">
```

```
...
```

```
</HEAD>
```

Parametr **LOOP** określa, jak many ód twarzany plik dźwiękowy (liczbę powtórzeń dźwięku) podany w parametrze **SRC**:

Wartość **infinite** – odtwarzanie bez przerwy

Przykład dla pliku dźwiękowego WAV (tyłkodla Microsoft IE) `<BGSOUNDSRC="dzwiek.wav" LOOP=infinite>`

Tabele

Każda tabela składa się z wierszy. Każdy wiersz składa się z danych.

Znaki definiujące tabelę: `<TABLE>` / `</TABLE>` (ang. Table – tabela).

Znaki definiujące pojedynczy wiersz tabeli: `<TR>` / `</TR>` (ang. Table row – wiersz tabeli).

Znaki definiujące pojedynczy daną w wierszu tabeli: `<TD>` / `</TD>` (ang. Table data – dane w wierszu).

Przykład:

```
<TABLE>
<TR>
<TD>Imię</TD><TD>Nazwisko</TD><TD>Wiek</TD>
</TR>
<TD>Grzegorz</TD><TD>Juszek</TD><TD>20</TD>
<TR>
<TD>Jan</TD><TD>Kowalski</TD><TD>18</TD>
</TR>
<TR>
<TD>Urszula</TD><TD>Kasolik</TD><TD>19</TD>
</TR>
</TABLE>
```

Wygląd w przeglądarce:

Imię	Nazwisko	Wiek
Grzegorz	Juszek	20
Jan	Kowalski	18
Urszula	Kasolik	19

Tabela z obramowaniem definiujesz za pomocą następującego znacznika: `<TABLE BORDER>` / `</TABLE>`

Przykład:

```
<TABLE BORDER>
<TR>
<TD>Imię</TD><TD>Nazwisko</TD><TD>Wiek</TD>
</TR>
<TD>Grzegorz</TD><TD>Juszek</TD><TD>20</TD>
<TR>
<TD>Jan</TD><TD>Kowalski</TD><TD>18</TD>
</TR>
<TR>
<TD>Urszula</TD><TD>Kasolik</TD><TD>19</TD>
</TR>
</TABLE>
```

Wygląd w przeglądarce:

Imię	Nazwisko	Wiek
Grzegorz	Juszek	20
Jan	Kowalski	18
Urszula	Kasolik	19

Za pomocą parametru BORDER można zmienić wygląd i grubość obramowania – na przykład na: `<TABLE BORDER=5>` lub `<TABLE BORDER=10>`

Wygląd w przeglądarce:

Imię	Nazwisko	Wiek
Grzegorz	Juszek	20
Jan	Kowalski	18
Urszula	Kasolik	19

Imię	Nazwisko	Wiek
Grzegorz	Juszek	20
Jan	Kowalski	18
Urszula	Kasolik	19

Ćwiczenie nr 7 – tabela

W pliku `tabela.html` utwórz tabelkę, która w przeglądarce będzie wyglądała następująco:

Pierwsza komórka	Druga komórka
Trzecia komórka	Czwarta komórka

CZĘŚĆ III

Umieszczanie skryptów JavaScript na stronach WWW

Celemu żywania skryptów jest w dokumentach internetowych HTML jest wzbogacenie ich o żywość stron WWW w postaci interesujących elementów takich jak np. przyciski, formularze, umieszczanie tekstu w wierszu statusu przeglądarki, otwieranie nowego okna bez zamykania okna aktualnego itp.

Jak umieścić skrypt JavaScript w dokumencie HTML?

Aby umieścić skrypt napisany w JavaScript na stronie internetowej używa się znaczników: `<SCRIPT LANGUAGE="JavaScript">` `</SCRIPT>`.

Například, aby wyświetlić napis na stronie WWW należy wewnątrz dokumentu HTML wpisać sekcję **BODY** następująco:

```
<HTML>
<HEAD>
To jest HTML + JavaScript
</HEAD>
```



```

<BODY>
<SCRIPT LANGUAGE="JavaScript">
    document.write("To jest JavaScript!")
</SCRIPT>
</BODY>
</HTML>

```

Bardziej skomplikowany skrypt mo ̇ze wygl ąda ̇ nast ̇ epuj ąco:

```

<script language="JavaScript">
<!--

function go () {
if (document.selecter.select1.options[document.selecter.select1.selectedIndex].value != "none")
{
location = document .selecter.select1.options[document.selecter.select1.selectedIndex].value
}
}
}
//-->
</script>

<script language="JavaScript">
<!--
document.write('<formname = "selecter"><selectname = "select1" size = 1>');
document.write('<optionvalue = none>Wybierz test z listy i kliknij przycisk bok   ...');
document.write('<optionvalue = none>-----');
document.write('<optionvalue = "ei_inf_test1.htm">Test nr 1 - Podstawy przetwarzania informacji ');
document.write('<optionvalue = "ei_inf_test2.htm">Test nr 2 - Podstawy budowy komputera IBMPC ');
document.write('<optionvalue = "ei_inf_test3.htm">Test nr 3 - Struktura informacji w komputerze ');
document.write('<optionvalue = "ei_inf_test4.htm">Test nr 4 - System operacyjny Windows ');
document.write('<optionvalue = "ei_inf_test5.htm">Test nr 5 - Edytory tekstów ');
document.write('<optionvalue = "ei_inf_test6.htm">Test nr 6 - Arkusze kalkulacyjne ');
document.write('</select>');
document.write('<INPUT TYPE = "button" VALUE = "Uruchom test" onclick = "go()">');
document.write('</form>');
//-->
</script>

```

CZĘŚĆ IV – ZADANIA SPRAWDZAJ ąCE

ZADANIE NR 1

Temat: Tworzenie stron WWW zawieraj ących podstawowe znaczniki HTML.

Treść zadania: Tworzenie szkieletu dokumentu, wstawianie tytułu, akapitu, poziomych linii, list numerowanych i wypunktowanych, hiperłączy (odsyłączy), animacji typu MARQUEE, zmiana stylu czcionek.

Ucze ̇n powinien umie ̇ć: Utworzy ̇ć szkielet dokumentu HTML oraz sprawnie posługiwac ̇ się znacznikami: <HTML>, <HEAD>, <TITLE>, <BODY>, <P>, <HR>, , , , <A>,
, <MARQUEE> oraz znacznikami fizycznymi.

Ucze ̇n mo ̇ze w trakcie wykonywania zadania korzysta ̇ć z: Skryptu ̇wiczeniowego HTML, programów Pającek, FrontPage, Microsoft Word, Notatnik i innych pomocy edukacyjnych.

Wykonaj nast ̇ epuj ące strony WWW:

Zadanie Nr 1.html – zawieraj ąca:

W sekcji <HEAD>.....</HEAD> umie ̇ć nast ̇ epuj ące znaczniki:

```

<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=iso-8859-2">
<META HTTP-EQUIV="Creation-date" CONTENT="0000-00-00">

```

```
<META NAME="Description" CONTENT="Zadanie nr 1">
<META NAME="Author" CONTENT="Imię i nazwisko ucznia">
<TITLE>Zadanie nr 1</TITLE>
```

Tło strony głównej w pliku **backgrnd.gif**.

Tekst tytułowy <H4>....</H4>: dzisiejsza data, imię i nazwisko autora, symbol klasy i grupy.

Tekst tytułowy na środku <H1>....</H1>: Zadanie nr 1.

Linia pozioma.

Obrazek **polska.gif** na środku.

Linia pozioma.

Na środku tekst LISTA HIPERŁĄCZY.

Dwa hiperłącza do oddoszczesć: do pliku **stronaa.html** i do pliku **stronab.html**.

Użycie obrazka **kulka.gif** do znaczników hiperłącza.

Tekst animacji typu MARQUEE: **To jest zadanie sprawdzające**

Tekst wyrównany do prawego marginesu: Strona utworzona w Pracowni Informatyki ZSBA (styl czcionki **STRONG**).

stronaa.html – zawiera jąca:

W sekcji <HEAD>....</HEAD> umieści następujące znaczniki:

```
<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=iso-8859-2">
```

```
<META HTTP-EQUIV="Creation-date" CONTENT="0000-00-00">
```

```
<META NAME="Description" CONTENT="Strona A">
```

```
<META NAME="Author" CONTENT="Imię i nazwisko ucznia">
```

```
<TITLE>Strona A</TITLE>
```

Tło strony głównej w pliku **backgrnd.gif**.

Tekst tytułowy <H4>....</H4>: imię i nazwisko autora.

Tekst tytułowy na środku <H1>....</H1>: Strona A.

Linia pozioma.

Tekst po lewej stronie, pogrubiony: Lista ksiązeko Internetcie.

Lista wypunktowana:

Christian Crumlish, Internet dla początkujących

Dariusz Boncler, Moja strona WWW

Maria Sokół, Tworzenie stron WWW - Ćwiczenia praktyczne

Michael Sullivan-Trainor, Infostrada

Linia pozioma.

Po lewej stronie tekst LISTA HIPERŁĄCZY.

Dwa hiperłącza do oddoszczesć: do pliku **ZadanieNr1.html** i do pliku **stronab.html**.

Użycie obrazka **kulka.gif** do znaczników hiperłącza.

Tekst wyrównany do prawego marginesu: Strona utworzona w Pracowni Informatyki ZSBA (styl czcionki **STRONG**).

stronab.html – zawiera jąca podobne elementy jak **stronaa.html**, z tym że:

W sekcji <HEAD>....</HEAD> umieści następujące znaczniki:

```
<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=iso-8859-2">
```

```
<META HTTP-EQUIV="Creation-date" CONTENT="0000-00-00">
```

```
<META NAME="Description" CONTENT="Strona B">
```

```
<META NAME="Author" CONTENT="Imię i nazwisko ucznia">
```

```
<TITLE>Strona B</TITLE>
```

Tekst tytułowy na środku <H1>....</H1>: Strona B.

Linia pozioma.

Tekst po lewej stronie, pogrubiony: Lista podręczników.

Lista numerowana:

Byron Pfaffenberger, Słownik terminów komputerowych.

Byron Spinney, Ethernet - Porady praktyczne.

Tomasz Zygorowicz, PC i sieć komputerowe.

H. Małysiak, B. Pochopień, E. Wróbel, Mikrokomputery klasy IBM PC.

Krzysztof Wojtuszkiewicz, Urządzenia peryferyjne i interfejsy.

Dwa hiperłącza do oddoszczesć: do pliku **ZadanieNr1.html** i do pliku **stronaa.html**.

Widok stron WWW:

1. stronawpliku **ZadanieNr1.html**:

Gdańsk, dnia 0000-00-00

Imię i nazwisko: XXXXXXXXXXXXXXXXXXXXXXXX

Klasa: XXXX, Grupa: 0

Zadanie nr 1



LISTA HIPERŁĄCZY:

- [Strona A](#)
- [Strona B](#)

To jest zadanie sprawdzające

Strona utworzona w Pracowni Informatyki ZSBA

Uwaga: czas na wykonanie zadania: 2x45 minut).

Poprawne wykonanie następujących elementów:	Ocena:
zapisanie plików ZadanieNr1.html, stronaa.html, stronab.html	Dopuszczająca
zapisanie plików(j.w.)+tło strony, napisy, linie poziome	Dostateczna
zapisanie plików(j.w.)+tło strony, napisy, linie poziome+obrazki, animowany tekst, listy: numerowane i wypunktowane	Dobra
zapisanie plików(j.w.)+tło strony, napisy, linie poziome+obrazki, animowany tekst, listy: numerowane i wypunktowane+hiperłącza	Bardzo dobra

2. stronawpliku **stronaa.html**:

Imię i nazwisko: XXXXXXXXXXXXXXXXXXXXXXXX

Strona A

Lista książek o Internecie:

- Christian Crumlish, Internet dla początkujących.
- Dariusz Boncler, Moje strony WWW
- Maria Sokół, Tworzenie stron WWW - Ćwiczenia praktyczne
- Michael Sullivan - Trainor, Infostrada

LISTA HIPERŁĄCZY:

- [Strona główna](#)
- [Strona B](#)

Strona utworzona w Pracowni Informatyki ZSBA

3. stronawpliku **stronab.html**:



Imię i nazwisko: XXXXXXXXXXXXXXXXXXXXXXXX

Strona B

Lista podręczników:

1. Btyan Pfaffenberger, Słownik terminów komputerowych.
2. Byron Spinney, Ethernet - Porady praktyczne.
3. Tomasz Zydorowicz, PC i sieci komputerowe.
4. H. Małysiak, B. Pochopień, E. Wróbel, Mikrokomputery klasy IBM PC.
5. Krzysztof Wojtuszkiewicz, Urządzenia peryferyjne i interfejsy.

LISTA HIPERŁĄCZY:

-  [Strona główna](#)
-  [Strona A](#)

Strona utworzona w Pracowni Informatyki ZSBA

ZADANIENR2

Temat: Tworzenie stron WWW zawierających klasyczne odnośniki URL oraz menu wybierające adres URL.

Treść zadania: Tworzenie szkieletu dokumentu, wstawianie tytułu, akapitu, poziomych linii, list numerowanych i wypunktowanych, hiperłączy (odsyłaczy), animacji typu MARQUEE, zmiana stylu czcionek, wstawianie obrazków.

Uczeń powinien umieć: Utworzyć szkielet dokumentu HTML oraz sprawnie posługiwać się znacznikami: <HTML>, <HEAD>, <TITLE>, <BODY>, <P>, <HR>, , , , <A>,
, <MARQUEE>, , <SCRIPT> oraz znacznikami fizycznymi.

Uczeń może w trakcie wykonywania zadania korzystać z: Skryptu ćwiczeniowego HTML, programów Pajączek, FrontPage, Microsoft Word, Notatnik i innych pomocy edukacyjnych.

UtwórzdwiestronyWWW(pierwszaonazwie **witryna.html**,drugaonazwie **ei.html**).

Uwaga:czasnawykonaniezadania:2x45minut).

Wskazówkido **witryna.html**:

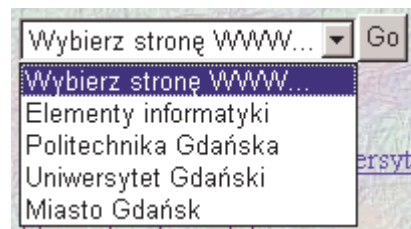
1. Spytaj nauczyciela o miejsce przechowywania następujących plików: **artus.jpg**, **mouse.jpg**, **muppets.mid**, **pastel.jpg**, **poland.gif**, **zsbalogo.gif**.
2. Skopiuj ww. pliki do katalogu w którym będziesz tworzyć strony WWW.

3. W obu plikach HTML zastosuj stronę kodową ISO-8859-2: za pomocą znacznika `<META http-equiv="Content-Type" content="text/html; charset=iso-8859-2">` w sekcji HEAD.
4. Wykonaj stronę `witryna.html` wg następujących wskazówek:
 - `<title>Zadanie kontrolne z HTML</title>`
 - podkład muzyczny z pliku `muppets.mid` jednokrotnie odtwarzanego,
 - wstaw logo szkoły po lewej stronie (plik: `zsbalogo.gif`),
 - wstaw tekst () po prawej stronie: Twoje imię, nazwisko, klasa, grupa + „Gdańsk,” + dzisiejsza data,
 - tekst ma być pogrubiony,
 - wstaw tekst „**Moja prywatna witryna**” (wycentrowany, wielk. fontu = 5),
 - wstaw obrazek (plik `poland.gif`) – wycentrowany,
 - wstaw obrazek (plik `artus.jpg`) po prawej stronie,
 - wstaw tytuł „**Moje zainteresowania:**” (znacznik H4),
 - wstaw listę wypunktowaną:
 - Podróże
 - Dyskoteka
 - Życie pod namiotami
 - Grafika komputerowa

wstaw tytuł „**Nie lubię:**” (znacznik H4),

- wstaw listę wypunktowaną:
 1. Głupoty
 2. Chamstwa
 3. Uczenia się na pamięć
 4. Pacanów
- Wstaw linię poziomą,
- Wstaw tekst przesuwający się poziomo (animację Marquee): „**Zostań naszym studentem**”,
- Wstaw cztery odnośniki do następujących adresów URL:
 - `http://www.pg.gda.pl` (napis: **Politechnika Gdańska**)
 - `http://www.univ.gda.pl` (napis: **Uniwersytet Gdański**)
 - `http://www.gdansk.gda.pl` (napis: **Nasze Miasto Gdańsk**)
 - `ei.html` (napis: **Elementy informatyki**)

Widok menu rozwijanego (tylko na ocenę celującą):

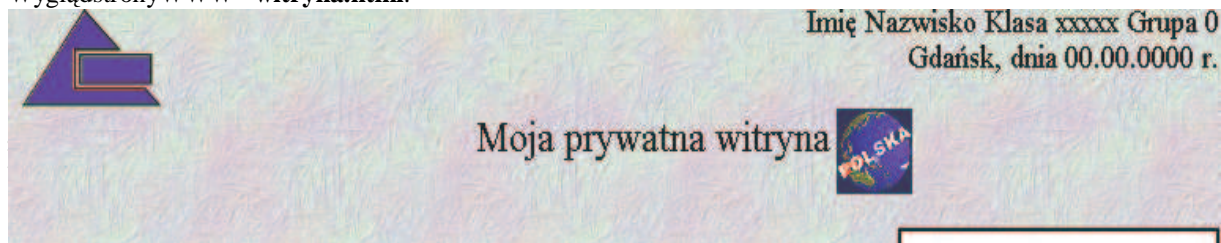


(dot. utworzenia skryptu na ocenę celującą)

Adresy URL w menu:

`ei.html` (napis: **Elementy informatyki**)
`http://www.pg.gda.pl` (napis: **Politechnika Gdańska**)
`http://www.univ.gda.pl` (napis: **Uniwersytet Gdański**)
`http://www.gdansk.gda.pl` (napis: **Miasto Gdańsk**)

Wygląd strony WWW `witryna.html`:



Wskazówki do **ei.html**:

1. Wykonaj stronę **ei.html** wg następujących wskazówek:

- `<META NAME="Author" CONTENT="Autor dokumentu">`
`<TITLE>ELEMENTY INFORMATYKI</TITLE>`
- wstaw napis po środku „**ELEMENTY INFORMATYKI**”,
- wstaw tekst animowany „Zapraszamy na lekcje z elementów informatyki”,
- wstaw obrazek z pliku `mouse.jpg`.

Wygląd strony WWW **ei.html**:



Zasady oceniania zadania nr 2

<i>Poprawne wykonanie następujących elementów:</i>	<i>Ocena:</i>
zapisanie plików witryna.html, ei.html .	<i>Dopuszczająca</i>
zapisanie plików (j.w.) + tło strony, napisy, linie poziome.	<i>Dostateczna</i>

zapisanieplików(j.w.)+tłostrony,napisy,liniepziome+obrazki,animowanytekst,listy:numerowanaiwypunktowana.	<i>Dobra</i>
zapisanieplików(j.w.)+tłostrony,napisy,liniepziome+obrazki,animowanytekst,listy:numerowanaiwypunktowana+hiperłącza.	<i>Bardzodobra</i>
zapisanieplików(j.w.)+tłostrony,napisy,liniepziome+obrazki,animowanytekst,listy:numerowanaiwypunktowana+hiperłącza+skryptJavaScriptobsługującymenu.	<i>Celująca</i>

DODATEK – ZNACZNIKI HTML

Znaczniki podstawowe

NAZWA	OPIS
<BODY>....</BODY>	Treść dokumentu
<A>.....	Hiperłącze, zakotwiczenie
 	Wymuszone złamanie wiersza
<BUTTON>....</BUTTON>	Przycisk
<FORM>....</FORM>	Formularz
<FRAME>	Ramka
<FRAMESET>....</FRAMESET>	Układ ramek
<HEAD>....</HEAD>	Nagłówek dokumentu
<HR>	Linia pozioma
<HTML>....</HTML>	Określenie typu dokumentu
	Obraz osadzony
<INPUT>	Pole formularza
	Element listy
<MENU>....</MENU>	List menu
....	List numerowana
<P>	Akapit
<SCRIPT>....</SCRIPT>	Wyrażenie skryptowe
<TD>...</TD>	Komórka tabeli
<TH>....</TH>	Komórka nagłówka tabeli
<TITLE>....</TITLE>	Tytuł dokumentu
<TR>....</TR>	Wiersz tabeli
....	List wypunktowana

Znaczniki fizyczne

Czcionka pogrubiona

Przykład:

To jest **czcionka pogrubiona (bold)**

Czcionka pochylona

<I></I>

Przykład:

To jest *czcionka pochylona (italic)*

Czcionka migająca

<BLINK></BLINK>

Przykład:

To jest **czcionka migająca (blink)** – jest to rozszerzenie Netscape'a, niedziałające w specyfikacji HTML4.

Czcionka podkreślona

<U></U>

Przykład:

To jest czcionka podkreślona (underline)

Czcionka o stałej szerokości znaku

<TT></TT>

Przykład:

To jest czcionka monotypiczna, o stałej szerokości znaku (fixed width = teletype)

Czcionka przekreślona

<STRIKE></STRIKE>

Przykład:

~~Czcionka przekreślona~~

Superskrypt (indeks górny)

Przykład:

Czcionka^{superskryptem}

Subskrypt (indeks dolny)

Przykład:

Czcionka_{subskrytem}

Dużaczcionka (+1 punkt)

<BIG></BIG>

Przykład:

Duża czcionka

Małaczcionka (-1 punkt)

<SMALL></SMALL>

Przykład:

Małaczcionka

DODATEK B – WYBRANE ELEMENTY JAVASCRIPT

Umieszczanie skryptów w dokumencie HTML

Aby umieścić skrypt w języku JavaScript na żywo w dokumencie HTML, użyj następującej konstrukcji:

<SCRIPT LANGUAGE="JavaScript">

<!--

tutaj powinien zostać umieszczony deklaracja zmiennych, funkcje oraz polecenia języka JavaScript

-->

</SCRIPT>

Deklaracje zmiennych

W języku JavaScript istnieją następujące typy danych:

- tekstowy,
- liczbowy,
- logiczny

- null.

Aby zadeklarować zmienną typu tekstowego należyć przed życiem tej zmiennej umieścić słowo **var**, nazwę zmiennej oraz przypisać jej wartość początkową w nawiasie średnikkończy instrukcję np.
var tekst = "To jest wartość tekstowa";

Aby zadeklarować zmienną typu liczbowego należyć przed życiem tej zmiennej umieścić słowo **var**, nazwę zmiennej oraz przypisać jej liczbę wartość początkową (średnikkończy instrukcję) np.
var liczba = 123;

Aby zadeklarować zmienną typu logicznego (**true** lub **false**) należyć przed życiem tej zmiennej umieścić słowo **var**, nazwę zmiennej oraz przypisać jej wartość logiczną (średnikkończy instrukcję) np.
var koniec = false;

Aby zadeklarować zmienną typu pustego należyć przed życiem tej zmiennej umieścić słowo **var**, nazwę zmiennej oraz przypisać jej wartość **null** (średnikkończy instrukcję) np.
var nic = null;

Operatory arytmetyczne, porównania i logiczne

Zapomóż operatorów można wykonywać podstawowe operacje arytmetyczne i logiczne takie jak dodawanie, odejmowanie, mnożenie, dzielenie, porównanie lewej strony z prawą oraz koniunkcja, alternatywa, negacja.

Operatory arytmetyczne

OPERATOR	OPIS
+	dodawanie liczb
-	odejmowanie liczb
*	mnożenie liczb
/	dzielenie liczb
--	zmniejszenie wartości o 1
++	zwiększenie wartości o 1

Operatory porównania

OPERATOR	OPIS
==	czy lewa strona jest równa prawej
<=	czy lewa strona jest mniejsza lub równa prawej
>=	czy lewa strona jest większa lub równa prawej
!=	czy lewa strona jest nierówna prawej
<	czy lewa strona jest mniejsza od prawej
>	czy lewa strona jest większa od prawej

Operatory logiczne

OPERATOR	OPIS
	Alternatywa dwóch wyrażeń
&&	Koniunkcja dwóch wyrażeń
!	Zaprzeczenie (negacja wyrażenia)

Funkcje

Składni definicji funkcji wygląda następująco:

```
function nazwa_funkcji(argumenty)
{
  ...
  polecenia JavaScript;
  ...
  return wartość_zwracana;
}
```

Definicja funkcji musi rozpoczynać się słowem kluczowym **function**. Potem następuje nazwa funkcji. W nawiasach okrągłych umieszczane są argumenty przekazywane do wnętrza funkcji. Jeśli liczba argumentów jest większa od 1, to argumenty należą do części przecinkami. Nawias klamrowy { rozpoczyna działanie funkcji, a nawias klamrowy } kończy działanie funkcji. Pomiędzy tymi nawiasami umieszczasz polecenia w języku JavaScript – każde polecenie kończysz średnikiem. Polecenie **return** wartość zwracaną; zwraca konkretną wartość funkcji. W przypadku gdy to polecenie nie zostanie użyte funkcja zwraca wartość **null**.

Przykład funkcji:

```
function Dzielenie (x,y)
{
    var iloraz=0;
    iloraz =x /y;
    return iloraz ;
}
```

UWAGA: Funkcje umieszczamy w sekcji <HEAD>....</HEAD> dokumentu HTML (znaki // oznaczają komentarz) np.:

```
<HEAD>
...
<SCRIPT LANGUAGE="JavaScript">
<!--
//algorytm zamiany liczby binarnej (cztery bity) na dziesiętną
// Autor: Jerzy Kluczewski
function Bin2Dec()
{
    var x3 = 0;
    var x2 = 0;
    var x1 = 0;
    var x0 = 0;
    var y = 0;
    y = x3*8+x2*4+x1*2+x0;
    document.form1.xdec.value = y;
    return;
}
...
//-->
</SCRIPT>
</HEAD>
```

Instrukcja wyboru

Składnia instrukcji wyboru wygląda następująco:

```
if( warunek)
    {poleceniaJavaScript }
else
    {poleceniaJavaScript };
```

Przykład:

```
if(y!=0)
    {iloraz=x/y;}
else
    {alert('Niemożnadieli przez zero!');};
```

Instrukcja pętli

W języku JavaScript istnieje dwa rodzaje instrukcji pętli:

- **for** – jest to cięgi instrukcji wykonywanych z góry okre ślonej ilo ści razy,
- **while** – jest to cięgi instrukcji wykonywanych tak długo jak długo jest spełniony podany warunek.

Składnia instrukcji pętli **for** ma następującą postać:

```
for(zmienna; warunki; indeks){ poleceniaJavaScript };
```

Przykład:

```
var zmienna = 200;  
var krok == 1;  
for(krok; krok < 101; krok++){ zmienna = zmienna + 1};
```

Składnia instrukcji pętli **while** ma następującą postać:

```
while(warunek){ poleceniaJavaScript};
```

Przykład:

```
var zmienna = 200;  
var krok == 1;  
while(krok <= 100){ zmienna = zmienna + 1};
```

Formularze

Każdy formularz musi mieć swój własną nazwę **nazwa_formularza**. Formularz składa się z pól. Każde pole musi mieć swój własną nazwę.

Składnia adresu danego pola w formularzu jest następująca: **document.forms["nazwa_formularza"].nazwa_pola;**

Aby pola formularza były widoczne w dokumencie HTML nale ży użyć sekcji **<BODY>....<BODY>** następującego znacznika:

```
<FORM NAME="nazwa_formularza">  
<INPUT TYPE="text" NAME="nazwa_pola1" SIZE="20">  
<INPUT TYPE="button" NAME="nazwa_pola2" VALUE="Naciśnij" onClick="funkcja(argumenty)">  
</FORM>
```

Parametry pola **TYPE** okre ślają jego rodzaj i danego pola nale ży okre ślić w danym polu formularza. Mog ą to być pola tekstowe lub przyciski sterujące. Każde typ pola posiada swoją własną właściwość i metody.

Dla pola typu tekstowego, **TEXT**”mo żna okre ślić następujące właściwości:

NAME – nazwa pola tekstowego,

VALUE – aktualna zawarto ść pola,

SIZE – długo ść pola

Dla pola typu przycisk, **BUTTON**”mo żna okre ślić następujące właściwości:

NAME – nazwa przycisku,

VALUE – tekst przycisku,

Metoda **onClick** – okre ślają funkcję, która będzie wywołana po kliknię ciu przycisku.

Przydatne funkcje standardowej języka JavaScript

Funkcje daty i czasu systemowego

Aby pobra ć z systemu operacyjnego aktualn ą dat ę i czas JavaScript dysponuje następującymi funkcjami:

getDate, getMonth, getYear, getHours, getMinutes, getSeconds.

Przykład:

```
<SCRIPT LANGUAGE="JavaScript">  
var dzisiaj = new Date ;  
var dzien = dzisiaj.getDate();  
var miesiac = dzisiaj.getMonth();  
var rok = dzisiaj.getYear();  
var godziny = dzisiaj.getHours();
```

```
var minuty = dzisiaj.getMinutes();  
var sekundy = dzisiaj.getSeconds();  
</SCRIPT>
```

Funkcje: alert, confirm, prompt obsługujące okienka dialogowe

Okienka dialogowe czyli okna służące do komunikacji strony WWW z użytkownikami mogą być następujące:

- informacyjne **alert**,
- potwierdzające (z przyciskami „OK” i „Anuluj”) **confirm**,
- do prowadzenia dialogów **prompt**,

Okienko informacyjne wyświetlane jest za pomocą funkcji **alert**, której składnia jest następująca:

```
alert('tekst_komunikatu');
```

Przykład:

```
alert('Niewolno dzielić przez zero');
```

Okienko potwierdzające obsługiwane jest za pomocą funkcji **confirm**, której składnia jest następująca:

```
confirm('tekst_komunikatu').
```

 W przypadku naciśnięcia przycisku „OK” funkcja **confirm** zwraca wartość **true**, a w przeciwnym razie wartość **false**.

Przykład:

```
if(alert('Czy napewno chcesz zamknąć okno?')){window.close();}
```

Okienko do prowadzenia dialogów obsługiwane jest za pomocą funkcji **prompt**, której składnia jest następująca:

```
prompt('tekst_komunikatu', 'domyślna_wartość').
```

 W przypadku naciśnięcia przycisku „OK” funkcja **prompt** zwraca wartość wprowadzoną do pola okna, a w przeciwnym razie wartość domyślną.

Przykład:

```
nazwisko=alert('Podaj swoje nazwisko', 'Kowalski');
```

Funkcje zwracające dane o przeglądarce internetowej

Funkcja **navigator.appName** zwraca nazwę przeglądarki.

Funkcja **navigator.appVersion** zwraca numer wersji przeglądarki.

Funkcja **navigator.language** zwraca wersję językową przeglądarki.

Przykład:

```
<SCRIPT LANGUAGE="JavaScript">  
dokument.write("Nazwa:" + navigator.appName + "<BR>")  
dokument.write("Wersja:" + navigator.appVersion + "<BR>")  
dokument.write("Język:" + navigator.language + "<BR>")  
</SCRIPT>
```

DODATEK C – PRZYKŁADY SKRYPTÓW

Animowany tekst w pasku statusu

W sekcji <HEAD>...</HEAD> należy umieścić następujący skrypt:

```
<script language="JavaScript">  
<!-- Autorskryptu: Bart Jellema  
var helloText="TO JEST ANIMOWANY TEKST W PASKU STATUSU PRZEGLĄDARKI"  
var theText=""  
var started=false  
var step=0  
var times=1  
function welcomeText()  
{  
  times--  
  if(times==0)  
  {  
    if(started==false)
```

```

{
  started=true;
  window.status=hellotext;
  setTimeout("anim()",1);
}
thetext=hellotext;
}
}
function showstatustext(txt)
{
  thetext=txt;
  setTimeout("welcometext()",4000)
  times++
}
function anim()
{
  step++
  if(step==7){step=1}
  if(step==1){window.status='>==='+thetext+'===<'<}
  if(step==2){window.status='>=='+thetext+'==<='<}
  if(step==3){window.status='>=>'+thetext+'<=<'<}
  if(step==4){window.status='>=>'+thetext+'<=<='<}
  if(step==5){window.status='>=>'+thetext+'<=<='<}
  if(step==6){window.status='>=>'+thetext+'<=<='<}
  setTimeout("anim()",200);
}
//Autorskryptu: BartJellema-->
</script>

```

Poletekstowe zegarkiem

Poniższy skrypt wyświetla zegarek w polutekstowym:

```

<script language="javascript">
<!--W3eJAVAScriptPreset
var timerID=null;
var timerRunning=false;
function stopclock()
{
  if(timerRunning)
    clearTimeout(timerID)
    timerRunning=false;
}
function startclock()
{
  stopclock();
  showtime();
}
function showtime()
{
  var now=new Date();
  var hours=now.getHours();
  var minutes=now.getMinutes();
  var seconds=now.getSeconds();
  var timeValue=""+hours);
  timeValue+=((minutes<10)?"":0)+":"+minutes;
  timeValue+=((seconds<10)?"":0)+":"+seconds;
  document.clock.face.value=timeValue;
  timerID=setTimeout("showtime()",1000);
  timerRunning=true;
}
//-->
</script>

```

```
<formname="clock"onSubmit="0">
<inputtype="text"name="face"size="24">
</form>
<script>startclock();</script>
```

Rozwijanie menu w ramce

W sekcji <BODY>...</BODY> należy umieścić następujący skrypt:

```
<script language="JavaScript">
<!--
function go()
{
if
(document.selector.select1.options[document.selector.select1.selectedIndex].value
!= "none")
{
location =

document.selector.select1.options[document.selector.select1.selectedIndex].value
}
}
//-->
</script>

<script language="JavaScript">
<!--
document.write('<form name="selector"><select name="select1" size=1>');
document.write('<option value=none>Wybierz stronę WWW...');
document.write('<option value="ei.html">Elementy informatyki');
document.write('<option value="http://www.pg.gda.pl">Politechnika Gdańska');
document.write('<option value="http://www.univ.gda.pl">Uniwersytet Gdański');
document.write('<option value="http://www.gdansk.gda.pl">Miasto Gdańsk');
document.write('</select>');
document.write('<INPUT TYPE="button" VALUE="Go" onclick="go()">');
document.write('</form>');
//-->
</script>
```

DODATEK D – Podstawy JavaScript

Wstęp

Język JavaScript (wskrócie **JS**) jest obiektowym językiem interpretowanym przez przeglądarki dokumentów HTML i jest bardzo łatwy do przyswojenia przez każdego początkującego HTML-owca. Dokumenty pisanego w HTML z użyciem JS nazywamy stronami WWW dynamicznym HTML (**DHTML**).

Składnia języka JS przypomina bardzo składnię języka C++ oraz w pewnym zakresie języka Pascal, a więc wystarczy mieć trochę wiedzy z matematyki i języków oraz duży zapas do tworzenia atrakcyjnych stron WWW.

Dlaczego żywamy skryptów JS na stronach WWW?

Celemu żywaniaskryptów jest w dokumentach internetowych HTML jest wzbogaceniemo żliwościstronWWWw postaciinteresuj ącychelementówtakichjaknp.przyciski,formularze,umi eszczanietekstuwierszustatusu przeglądarki,otwieranienowegookna bezzamykaniaoknaaktualnegoitp.

Wystarczywklei ćodpowiednikoddoswojegodokumentuHTMLiwszystkocoonpotra fib ędziedziała ć.Poniewa ż skryptys ąbez żadnegokodowaniaumieszczanewHTMLumo żnaskopiowa ćskryptyzodwiedzanychstron. Oczywiście żetoby ćnaruszeniemprawautorskich,wi ęćlepiejkorzysta ćżserwisówoferuj ącychskryptyzadarmo dowykorzystaniaNiejestjednaknaruszeniemprawautor skichanalizowanieciekawychrozwi ązańizdobywanie wiedzynaprzykładach-tojestwr ęćzgodnepolecenia.

Jednakbezmy ślnaskopiowanieskryptówniepozwalawpełnidoceni ćmo żliwościtegoj ęzyka.Dodatkowo,bez znajomościchocia żpodstawnieb ędziemo żnaznale żćipoprawi ćbł ędów,anidostosowa ćskryptówdowłasnych potrzeb.Dlategowartozapozna ćsi ężpodstawamiJS.

Jak umie śććskrypt JS w dokumencie HTML?

Aby umie śććskrypt napisany w JS na le żyposłu życi ęznacznikiemHTML:

<SCRIPT LANGUAGE="JavaScript"> </SCRIPT>

Například,abywy świetlićnapisnastronieWWWnale ży(wewn ątrzdokumentuHTML)wpisa ćdosekcji **BODY** następuj ący skrypt:

```
<HTML>
<HEAD>
<TITLE>To jest HTML + JavaScript</TITLE>
</HEAD>
<BODY>
<SCRIPT LANGUAGE="JavaScript">
    document.write("To jest JavaScript!")
</SCRIPT>
</BODY>
</HTML>
```

Podstawowe elementy JS

Podstawowe elementy j ęzykaskryptowegoJavaScriptto: operatory, zdarzenia oraz obiekty.

Operatory

WJSopróćznanych powszechnieoperatorów +, -, *, /, wyst ępuj ą nast ępuj ąceoperatory¹:

Opis operacji	Symbol
Dzielenie modulo (reszta z dzielenia)	%
Podstaw praw ą stron ę do lewej	=
Równa się	==
Mniejsze od	<
Wi ększe od	>
Różne od	!=
Wi ększe lub równe	>=
mniejsze lub równe	<=
Nie mniejsze od	!<

¹ Operatory stosuje się po prawej stronie znaku „=” (tzw. Instrukcja podstawiania - do zmiennej "wynik" wstawiana jest wyliczona wartość: **wynik=liczba2 operatorliczba2**)

Predefiniowane obiekty JS

Array

Obiekt `Array` służy do tworzenia tablic w dokumentach HTML. Obiekt posiada metody do wykonywania różnych operacji na tablicach (np. łączenie, sortowanie), tak że w łatwy sposób można je modyfikować.

Tworzenie tablic odbywa się według schematu:

```
Tablica=new Array(pierwszy element, drugi element, ..., ostatni element);
```

Metody:

METODA	OPIS
<code>concat</code>	Łączy dwie tablice, tworząc nową
<code>join</code>	Łączy wszystkie elementy tablicy włańcuch tekstowy
<code>pop</code>	Usuwa ostatni element tablicy
<code>push</code>	Dodaje jeden lub kilka elementów tablicy
<code>reverse</code>	Odwraca kolejność elementów w tablicy
<code>shift</code>	Usuwa pierwszy element tablicy
<code>splice</code>	Dodaje lub usuwa elementy tablicy
<code>sort</code>	Sortuje elementy tablicy

Boolean

Obiekt `Boolean` służy do obsługi zmiennych logicznych np. do zamiany wartości zmiennych logicznych.

Przykłady:

```
zmienna=new Boolean()           tworzy zmienną o wartości false
zmienna=new Boolean(" ")         tworzy zmienną o wartości false
zmienna=new Boolean(false)       tworzy zmienną o wartości false
zmienna=new Boolean(true)        tworzy zmienną o wartości true
zmienna=new Boolean(1)           tworzy zmienną o wartości true
```

Obiekt `Boolean` posiada jedną właściwość. Jest nią `valueOf`. Zwraca on wartość obiektu `Boolean`.

Date

Obiekt `Date` służy do korzystania z daty i godziny. Nie posiada żadnych właściwości, ale zaoferuje duży zbiór metod do manipulowania datą i godziną.

Deklaracja obiektu odbywa się według schematu:

```
dzisiaj=new Date("rok, miesiąc, dzień, godzina, minuta, sekunda"); gdzie:
rok: liczba w zeszłym wieku (0-11), miesiąc: 0-11 (0-styczeń), dzień: 1-31
godzina: 0-23, minuta: 0-59, sekunda: 0-59
```

Metody:

METODA	OPIS
<code>getDate</code>	Zwraca dzień miesiąca
<code>getDay</code>	Zwraca dzień tygodnia
<code>getHours</code>	Zwraca godzinę
<code>getMinutes</code>	Zwraca minutę
<code>getMonth</code>	Zwraca numer miesiąca

getSeconds	Zwracasekundy
getYear	Zwracarok
setDate	Ustawiadzie ńmiesi ąca
setDay	Ustawiadzie ńtygodnia
setHours	Ustawiagodzin ę
setMinutes	Ustawiaminut ę
setMonth	Ustawianumermiesi ąca
setSeconds	Ustawiasekundy
setYear	Ustawiarok
toGMTString	Zamieniadat ęnała ńcuch
getTime	Zwracailo ńćmilisekundod1970r.
setTime	Ustawiailo ńćmilisekundod1970r.

Math

Obiekt `Math` reprezentuje funkcje stałe matematyczne. Jego metody to funkcje trygonometryczne, logarytmiczne, wykładnicze oraz inne, rzadziej używane. Natomiast wartości stałe matematyczne.

Właściwości - stałe matematyczne:

STAŁA	OPIS	PRZYBLI ŻENIE
E	Stała Eulera	2.718281828459045
LN10	Logarytm naturalny liczby 10	2.30585092994046
LN2	Logarytm naturalny liczby 2	0.693147180559945
LOG10E	Baza logarytmu dziesiętnego	0.4342944819032518
LOG2E	Baza logarytmu opodstawie 2	1.4426950408889634
PI	Liczba PI	3.141592653589793
SQRT1_2	Pierwiastek kwadratowy z 1/2	0.7071067811865476
SQRT2	Pierwiastek z 2	1.414213562373095

Metody:

FUNKCJA	OPIS	PRZYKŁAD	WYNIK
abs(liczba)	Warto ńć bezwzgl ędna	Math.abs(-5)	5
cos(alfa)	Cosinus k ątawradianach	Math.cos(Math.PI)	-1
sin(alfa)	Sinus k ątawradianach	Math.sin(Math.PI)	0
tan(alfa)	Tangens k ątawradianach	Math.tan(0)	0
acos(alfa)	Arcuscosinus k ątawradianach	Math.acos(1)	0
asin(alfa)	Arcussinus k ątawradianach	Math.asin(0)	0
atan(alfa)	Arcustangens k ątawradianach	Math.atan(1)	0.785398163397448
exp(x)	Zwracaliczb ę Eulerapodniesion ą do x pot ęgi	Math.exp(1)(0)	2.718281828459045
log(x)	Odwrotno ńć powoy ńszej funkcji	Math.log(1)	0
ceil(x)	Najmniejsza łąkowita, niemniejsza od x	Math.ceil(8.1)	9
floor(x)	Najwi ększa łąkowita, niewi ększa od x	Math.floor(8.1)	8
log(x)	Odwrotno ńć powoy ńszej funkcji	Math.log(1)	0
max(x,y)	Zwracaliczb ę wi ększ ą z podanych	Math.max(4,3.9)	4
min(x,y)	Zwracaliczb ę mniejsz ą z podanych	Math.min(4,9)	4
pow(x,y)	Liczba x do pot ęgi y	Math.pow(3,3)	27
random()	Zwracaliczb ę losow ą z przedziału 0-1	Math.random()	0.536296745237586
round(x)	Zaokr ąglienie x do liczby łąkowitej	Math.round(0.7)	1

String

Obiekt String służydooperacjinała ńcuchachtekstowych.

Tworzenieobiektuodbywasie wedługschematu:

```
Lancuch = new String(string);
```

gdziestringjestdowolnymła ńcuchemtekstowym

Obiekt Stringposiadajedn ąwła ściwość.Jestni ą String.length.Zwracaonadługo śćła ńcuchatekstowego.

Metody:

METODA	OPIS
anchor("nazwa")	Definiujenazw ękotwicy
big,blink,bold,fixed,italics,small,strike,sub,sup	Odpowiednioformatuj ątekst
charAt(pozycja)	Zwracaznakzokre ślonejpozycji
charCodeAt(pozycja)	Zwracakodznakuzokre ślonejpozycji
indexOf(łańcuch1,ła ńcuch2)	Zwracapozycj ępierwszegowyst ąpieniała ńcucha1wła ńcuchu2
lastIndexOf(łańcuch1,ła ńcuch2)	Zwracapozycj ęostatniegowyst ąpieniała ńcucha1wła ńcuchu2
link	Tworzyodno śnikztekstu
concat(łańcuch2)	Doł ącząła ńcuch2doaktualnegoła ńcucha
split(znakpodziału,ilo ść)	Łańcuchzostajepodzielonynapewn ąilo śćpodła ńcuchów, przyczymka żdysieko ńczyznakiempodziału
slice(początek,koniec)	Zwracała ńcuchozadanypocz ątkuiko ńcu
substr(początek,długo ść)	Zwracała ńcuchopewnejdługo śćiizadanypocz ątku
toLowerCase	Zamieniała ńcuchnamałelitery
toUpperCase	Zamieniała ńcuchnawielkielitery

Istniejejeszczewi ęcejpredefiniowanychobiektówzktórychnajciekawszeto:

Document–reprezentujeaktualnieprzeł ądanydokument

Właściwości:

WŁAŚCIWOŚĆ	OPIS
bgColor	Kodkolorutła
fgColor	Kodkolorugłównegoelementów
lastModified	Dataostatniejmodyfikacjistrony

Metody:

METODA	OPIS
close()	Zamykadokument.
open()	Otwieradokument.
write("tekst") writeln("tekst")	Umieszczanastronieła ńcuch„tekst”(writeln powoduje żenast ępnymy elementwy świetlanyjestwnowejlinii).

Navigator-reprezentuje żywaną wla śnieprzeł ądarkę

Właściwości:

WŁAŚCIWOŚĆ	OPIS
appName	Nazwaprogramuprzeł ądarki
appVersion	Wersjaprogramuprzeł ądarki
language	Języknarodowyprzeł ądarki

Metody:

METODA	OPIS
javaEnabled()	Zwracawartość trueje śliprzeł ądarkaobsługujeskryptuJavy,w przeciwnymraziezwraca false.

Podstawowekonstrukcje żywanewJS

Podstawowekonstrukcje żywanewJS to: komentarze, okna dialogowe, okna komunikatu , instrukcje podstawiania, instrukcje warunkowe, deklaracje zmiennych, p ętle, definicje funkcji.

Komentarze:

<!-- To jest komentarz
// To jest komentarz

Okna dialogowe

Okno dialogowe typu "Prompt" słu ży do wprowadzania tekstów z klawiatury.

```
window.prompt(informacja, tekst_startowy)
```

Okna komunikatów

Okno komunikatu "Alert" słu ży do wprowadzania tekstu komunikatu

```
window.alert("tekst alarmu")
```

Instrukcje podstawiania

```
masa = 5.65;  
imie = "Jurek";  
nazwisko = "Kluczewski";  
czy_koniec = false;
```

Instrukcje warunkowe if

Instrukcja warunkowa steruje wykonaniem instrukcji w zależności od tego, czy warunek zmienna1 == zmienna2 jest spełniony:

```
if(zmienna1 == zmienna2)  
{ ciąg instrukcji }
```

```
Else  
{ ciąg instrukcji }
```

Deklaracjezmiennych

Nazwazmiennejmo żeby ćdowolna,podwarunkiem, żeśpełniaonast ępującewarunki:

- Nazwazaczynasi ęodlityeralboznakupodkre ślenia" _".
- Niezawieraspacji.
- Niejesttośłowozarezerwowanetz.n.takie,któremaj użspecjalneznaczeniewJS.

Deklaracjainicjuj ączienn ąmaposta ć:

```
var nazwa_zmiennej="Jaka jest siła Dżawaskryptów???";
```

Instrukcjep ętli

Instrukcja if

Instrukcjap ętli ifsterujewykonanieminstrukcji,gdyzgórywiadomoileraz ydanyci ąginstrukcjimaby ćwykonany.

Instrukcjap ętlimaposta ć:

```
for (zmienna; warunki; modyfikacja_indeksu) { polecenia }
```

gdzie zmiennajesttowarto śćindeksup ętli, warunkis ąwarunkamizako Ńczeniap ętli,a modyfikacja_indeksujestwyr ążeniemmodyfikuj ącymwarto śćindeksup ętli.Poleceniaznajduj ącesi ęwnawiasachklamrowychb ędąwykonywanedot ąda żindeksprzestanięsp ęłnia ćwarunek.

Przykład:

```
var n=1;  
for (n; n<11; n++) { document.write("<TD>"+n+"</TD>"); }
```

Instrukcja while

Instrukcja whilepowoduje, żepoleceniab ędąwykonywanetakdługo,jakdługowarunekjestsp ęłniony.

Postaćinstrukcji whilejestnast ępująca::

```
while(warunek) { polecenia }
```

Przykład:

```
var koniec=false;  
var n=0;  
while (!koniec) { n++; if (n>10) {koniec=true}; }
```

Definicjefunkcji

Definicjęfunkcjirozpoczynasi ęsłowem function.

```
function nazwa_funkcji(argumenty)
{ polecenia;
return wartość
}
```

Definicję funkcjonalną znajdziesz za pomocą edytorów znaczników. Należy pamiętać, aby argumenty oddzielać **przecinkami**, polecenia **średnikami**. Polecenie `return` zwraca konkretną wartość jako wynik działania funkcji. Można nie podawać, wtedy funkcja zwróci wartość `null`.

Przykład:

```
function odwrotnie(tekst)
{
odwrotnie="";
dlugosc=tekst.length;
for (dlugosc;dlugosc>0;dlugosc--)
{ odwrotnie=odwrotnie+tekst.charAt(dlugosc-1); }
}
```

Przykładowe skrypty

```
<!--
////////////////////////////////////
// Przyciski nawigacji przeglądarki //
////////////////////////////////////

document.write("<CENTER>");
document.write("<FORM NAME=\"Historia\">");
document.write("<INPUT TYPE=\"BUTTON\" VALUE=\"Poprzednia\"
onClick=\"history.back()\">");
document.write("<INPUT TYPE=\"BUTTON\" VALUE=\"Główna strona\"
onClick=\"location='[/Tutaj podaj adres strony głównej]'/\">");
document.write("<INPUT TYPE=\"BUTTON\" VALUE=\"Następna\"
onClick=\"history.forward()\">");
document.write("</FORM>");
document.write("</CENTER>");
```

```
////////////////////////////////////
// Wyświetla dane o przeglądarce //
////////////////////////////////////

<!--
document.writeln("<p class=text> Używasz " + navigator.appName + " (" +
navigator.appCodeName + ") " + navigator.appVersion);
```

```
////////////////////////////////////
// Wyświetla teksty w zależności od pory dnia //
////////////////////////////////////

<!--
```

```

d = new Date();
hour = d.getHours();
if(hour < 5)
{
    document.write("Natychnmiasť spać!!! Zobacz która jest godzina");
}
else
if(hour < 8)
{
    document.write("Jak dobrze wstać skoro świt...");
}
else
if(hour < 12)
{
    document.write("Mamy piękne przedpołudnie...");
}
else
if(hour < 20)
{
    document.write("Żeglowanie po internecie przed 20-tą???");
}
else
{
    document.write("Już niedługo nastanie ciemność - zbliża się noc");
}

```

```

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=iso-8859-2">
<META NAME="Description" CONTENT="Zegarek w statusie">
<META NAME="Author" CONTENT="Jerzy Kluczewski">
<TITLE>Zegarek w wierszu statusu</TITLE>

<script Language="JavaScript">
<!--
////////////////////////////////////
// Wyświetla zegarek w wierszu statusu      //
////////////////////////////////////
var timerID = null;
var timerRunning = false;
function stopclock (){
if(timerRunning)
clearTimeout(timerID);
timerRunning = false;
}
function showtime ()
{
var now = new Date();
var hours = now.getHours();
var minutes = now.getMinutes();
var seconds = now.getSeconds()
var timeValue = "Aktualny czas "
timeValue += ((hours < 10) ? "0" : "") + hours
timeValue += ((minutes < 10) ? ":0" : ":") + minutes
timeValue += ((seconds < 10) ? ":0" : ":") + seconds
window.status = timeValue;
timerID = setTimeout("showtime()",1000);
timerRunning = true;

```

```

function startclock ()
{
stopclock();
showtime();
}
</script>

</HEAD>

<BODY ONLOAD="startclock()">

<H1><P STYLE=" text-align: justify; font-family: Arial Black">
Aktualny czas pokazuje zegarek w linii statusu</P></H1>

</BODY>
</HTML>

```

Superanimacja w pasku statusu

```

<HTML>
<HEAD>
<META HTTP-EQUIV="Content-type" CONTENT="text/html; charset=iso-8859-2">
<TITLE>Animacja w pasku statusu</TITLE>

<SCRIPT LANGUAGE="JavaScript">

<!--
// _____
// ANIMACJA W PASKU STATUSU PRZEGLĄDARKI
// Autorzy: (c) 1996 - Tomer and Yehuda Shiran
// Ten skrypt ukazał się w książce ww. autorów.
// Zgodnie z panującymi zwyczajami umieściłem ten komentarz jako
// wyraz poszanowania ich praw autorskich i formę podziękowania.
// Tłumaczenie na język polski: Jerzy Kluczewski
// _____

// odstęp czasowy pomiędzy wstawianiem kolejnych znaków
var speed = 10 // wartość dodatnia [milisekundy]

// czas pauzy pomiędzy zakończeniem cyklu animacji i rozpoczęciem nowego cyklu
var pause = 2000 // wartość dodatnia [milisekundy]

// wartości początkowe
var timerID = null
var bannerRunning = false

// utworzenie tablicy i wpisanie ciągów znakowych, które będą animowane
var ar = new Array()
ar[0] = "Witaj na stronie, która powstała dzięki Javie"
ar[1] = "Pierwsze środowisko obiektowe Java powstało dzięki"
ar[2] = "Billowi Joy w roku 1990"
ar[3] = "James Gosling, Patrick Naughton i Mike Sheridan w roku 1991"
ar[4] = "przyczynili się do powstania języka Java"
ar[5] = "Później Gosling opracował język Object Application Kernel"
ar[6] = "którego nazwę zmieniono na Java"
ar[7] = "Na bazie Javy powstał język JavaScript w którym niniejszy"
ar[8] = "skrypt napisali Tomer i Yehuda Shiran"

```



```
var message = 0 //indeks aktualnego napisu
```

```
var state = "" // stan animacji (pusty ciąg)
```

```
clearState() // resetowanie stanu animacji
```

```
// zatrzymanie aktywnej animacji
```

```
function stopBanner()
```

```
{
    if (bannerRunning) clearTimeout(timerID) // jeżeli animacja aktywna to zatrzymaj ją
    // zatrzymaj stoper
    timerRunning = false
} // of stopBanner()
```

```
// start animacji
```

```
function startBanner()
```

```
{
    // zatrzymaj animację
    stopBanner()
    // rozpocznij animację od aktualnej pozycji
    showBanner()
} // of startBanner()
```

```
// stan animacji to łańcuch znaków "0" o długości równej aktualnemu napisowi
```

```
function clearState()
```

```
{
    // inicjalizacja pustego łańcucha o indeksie "message"
    state = ""
    // utworzenie łańcucha o tej samej długości zawierającego znaki "0"
    for (var i = 0; i < ar[message].length; ++i) { state += "0" }
} // of clearState()
```

```
// wyświetlanie aktualnego napisu
```

```
function showBanner()
```

```
{
    // jeżeli aktualny napis został wykonany
    if (getString())
    {
        // zwiększ wskaźnik napisu
        message++
        // jeżeli nowy wskaźnik napisu jest poza tablicą to rozpocznij od pierwszego napisu
        if (ar.length <= message) message = 0
        // nowy napis jest najpierw wyświetlany jako pusty
        clearState()
        // wyświetl następny znak po czasie "pause" w milisekundach
        timerID = setTimeout("showBanner()", pause)
    }
    else
    {
        // inicjalizuj pusty ciąg
        var str = ""
        // buduj łańcuch do wyświetlania ze znaków których status jest "1"
        for (var j = 0; j < state.length; ++j)
        {
            str += (state.charAt(j) == "1") ? ar[message].charAt(j) : " "
        }
        // umieść łańcuch częściowy w pasku statusu przeglądarki
        window.status = str
        // dodaj następny znak po czasie "speed" milisekund
        timerID = setTimeout("showBanner()", speed)
    }
} // of showBanner()
```

```

function getString()
{
// ustaw zmienną "full" na "true"
var full = true
// ustaw na "false" jeśli znaleziono wolne miejsce w łańcuchu (znak nie wyświetlany)
for (var j = 0; j < state.length; ++j)
{
// jeśli znak o indeksie "j" w napisie nie jest znakiem wyświetlanym
if (state.charAt(j) == 0) full = false
}
// zwraca "true" jeśli cały łańcuch jest zapełniony (aby nie wchodzić do pętli)
if (full) return true
// szukaj pustych na losowych pozycjach (pusty przerywa pętlę)
while (1)
{
// generuj liczbę losową (z zakresu od 0 do state.length - 1 == message.length - 1)
var num = getRandom(ar[message].length)
// jeśli jest puste miejsce to przerwij pętlę
if (state.charAt(num) == "0") break
}
// na pozycji "num" zamień znak "0" na "1"
state = state.substring(0, num) + "1" + state.substring(num + 1, state.length)
// zwraca "false" ponieważ łańcuch nie jest pełny (znaleziono puste miejsca)
return false
} // of getString()

```

```

function getRandom(max)
{
// tworzy instancję obiektu Date (aktualny czas)
var now = new Date()
// wylicza liczbę losową (kombinacja liczby pseudolosowej i czasu)
var num = now.getTime() * now.getSeconds() * Math.random()
// ucina liczbę do zakresu pomiędzy 0 i (max - 1)
return num % max
} // of getRandom(max)

```

-->

</SCRIPT>

</HEAD>

<BODYonLoad="startBanner()">

<CENTER><H2>PrzykładstronyWWWzanimacj ą±wpaskustatusu</H2></CENTER>

</BODY>

</HTML>

Zadania ćwiczeniowe

Zadanie nr 1

Wykonaj stronę WWW zawierającą poniższy formularz.

Wprowadź dane i naciśnij przycisk "Oblicz"

Wartość weksla mln. zł.

Stopa dyskonta w skali rocznej %

Od daty płatności weksla upłynęło dni

Kwota odsetek zł.

Przycisk „Oblicz kwotę odsetek” umożliwi rozwiązanie następującego zadania:

Bank komercyjny przyjął do dyskonta weksla na 1 mln zł przystopie dyskontowej 10%, a do dni płatności brakuje jeszcze 36 dni. Jaką kwotę gotówki otrzyma klient za weksel po potrąceniu kwoty odsetek z tytułu dyskonta? Bank nabywający weksel (czyli dyskontujący weksel) nie płaci jego pełnej kwoty, lecz po potrąceniu kwoty pewną kwotę nazywaną kwotą z dyskontem. Kwota należna z banku odsetek liczonej jest według wzoru:

$$D = \frac{W \times \% \times t}{100 \times 360}$$

gdzie:

D - kwota odsetek z tytułu dyskonta,
W - suma weksla,
% - stopa dyskonta w skali rocznej,
t - liczba dni upływających od daty płatności weksla.

Zadanie nr 2

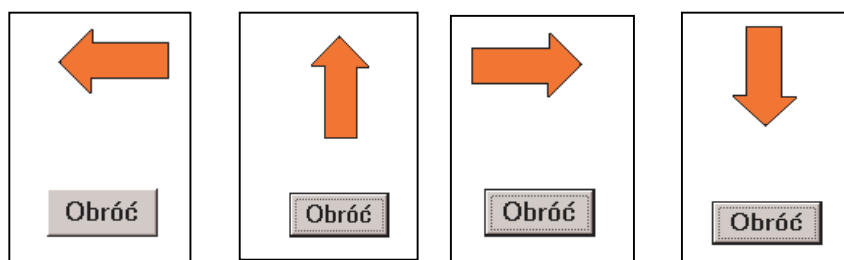
Wykonaj stronę WWW zawierającą następujący formularz:

Wprowadź ciąg znaków i naciśnij przycisk "Podziel"

Po wpisaniu ciągu znaków w górny pole tekstowe i naciśnięciu przycisku „Podziel” w dolnych polach tekstowych powinny wyświetlić się dwa ciągi znaków: do jednego ciagu wpisane zostaną znaki numerów parzystych, do drugiego – numerów nieparzystych (choć z numerów pozycji zajmowanych przez znaki wciagu).

Zadanie nr 3

Zapomocą dostępnych narzędzi wykonaj stronę WWW zawierającą rysunek strzałki, który będzie obracany o 90° zgodnie z ruchem wskazówek zegara po każdym kliknięciu w przycisk „Obróć”. Wykorzystaj odpowiednie programy do tworzenia rysunków strzałki.



Zadanie nr 4

Wykonaj stronę WWW zawierającą następujący formularz, w którym przycisk „Kontrola” powinien kontrolować, czy pierwsza z pól tekstowych jest pusta, a przycisk „Wyczyść” kasuje www.pola.

Imię	<input type="text"/>
Nazwisko	<input type="text"/>
Data urodzenia	<input type="text"/>
Płeć:	<input type="radio"/> Kobieta <input checked="" type="radio"/> Mężczyzna
Miejscowość	<div> <div>Częstochowa ▲</div> <div> <div>Darłowo</div> <div>Elbląg</div> <div>Gdańsk ▼</div> </div> </div>
<div>Wyczyść</div> <div>Kontrola</div>	

Zadanie nr 5

Utwórz stronę WWW, która zawiera następujący formularz:

Podaj kąt w stopniach, a potem naciśnij przycisk *Oblicz*

x =

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
sin(x)	cos(x)	tg(x)	ctg(x)

Powpisaniu wartości kąta (w stopniach) w pierwszym polu i naciśnięciu przycisku Oblicz w czterech następujących polach powinny pojawić się wartości funkcji trygonometrycznych dla wprowadzonego kąta np.:

Podaj kąt w stopniach, a potem naciśnij przycisk *Oblicz*

x =

<input type="text" value="0.7071067811865475"/>	<input type="text" value="0.7071067811865476"/>	<input type="text" value="0.9999999999999999"/>	<input type="text" value="1.0000000000000002"/>
sin(x)	cos(x)	tg(x)	ctg(x)

Zadanie nr 6

Wykonaj stronę WWW zawierającą następujący formularz:

Wprowadź dwie liczby i naciśnij przycisk + lub - lub * lub /

<input type="text"/>	<input type="text"/>		
x	y		
<input type="button" value="+"/>	<input type="button" value="-"/>	<input type="button" value="*"/>	<input type="button" value="/"/>
<input type="text"/>			

Powpisaniu dwóch liczb w polach oznaczonych x oraz y i naciśnięciu jednego z czterech przycisków w polu poniżej powinien pojawić się wynik wybranego działania arytmetycznego.

Zadanie nr 7

Wprowadź ciąg znaków i naciśnij przycisk "Zmień"

Zmień

Po wpisaniu ciągu znaków w górny pole tekstowe i naciśnięciu przycisku „**Zmień**” w dolnym polu powinien pojawić się wprowadzony ciąg znaków zamieniony literami wielkimi na małe, na przykład:

Wprowadź ciąg znaków i naciśnij przycisk "Zmień"

Architektura Starego Miasta

Zmień

aRCHITEKTURA sTAREGO mIASTA

Uwagi końcowe

JavaScript to nazwa języka programowania opracowanego przez firmy *Sun Microsystems* i *Netscape*. Umieszczenie na stronach WWW aktywnych elementów, takich jak przycisk zmieniający wygląd, gdy przesuniesz

nadnimimyszką, tekst przesuwają, a cyfry wpasują status przeglądarki czy odsyłacze otwierają nowe okna. To są najbardziej popularne zastosowania, ale można też tworzyć bardziej skomplikowane i pożyteczniejsze skrypty. JS jest obsługiwany przez wszystkie (dwie) główne przeglądarki, czyli IE (Internet Explorer) i NN (Netscape Navigator), ale także przez niektóre mniej znane, jak np. *Opera 3.20*. Język ten jest także stosowany do innych celów, niezwiązanych bezpośrednio z przeglądarką WWW – w JS można napisać ASP (Active Server Pages), można go wykorzystać przy automatyzacji zadań w różnych aplikacjach (np. MS Office, Star Office, czy system Windows 98).

Warto jednak zauważyć, że obsługa *JavaScript* w różnych przeglądarkach nie jest jednakowa, szczególnie jeśli Microsoft promuje własne "narzędzie", czyli *JScript*, zgodną z implementacją Netscape'a, ale rozszerzoną o dodatkowe funkcjonalności języka np. *Regular Expressions*.

LITERATURA

1. Maria Sokół. Tworzenie stron WWW – ćwiczenia praktyczne.
2. Dariusz Boncler. Moje strony WWW.
3. Paweł Wimmer. Podręcznik HTML (zainstalowany w Pracowni wraz z Pakietem).
4. Michał Gwóźdź. Internetowy Kurs JavyScript (źródło: <http://www.ipts.w.pl>).