

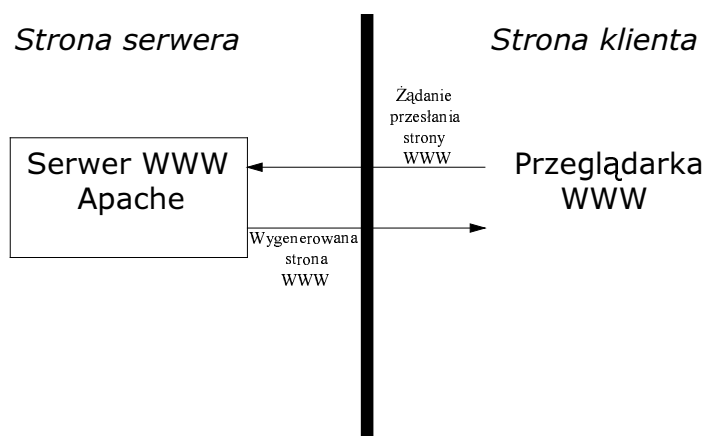
Technologia klient – serwer.

Najczęściej stosowanym rodzajem interakcji pomiędzy współdziałającymi programami jest **model klient – serwer**. Technologia ta stanowi podstawę większości komunikacji sieciowej.

Serwerem określa się każdy program, który oferuje usługę dostępną lokalnie lub poprzez sieć. Serwer przyjmuje zamówienia, wykonuje usługę i zwraca wyniki zamawiającemu.

Działający program nazywa się **klientem**, gdy wysyła zamówienie do serwera i czeka na jego realizację.

Komunikacja przeglądarki z serwerem WWW



Protokół HTTP

W przypadku interakcji pomiędzy serwerem WWW a przeglądarką wykorzystywany jest protokół **HTTP (Hypertext Transfer Protocol)**. Protokół ten określa formę żądań klienta dotyczących danych i formę odpowiedzi serwera na te żądania.

Pojedyncza operacja w protokole HTTP nazywana jest **transakcją**.

Przykładowa transakcja sieciowa może więc mieć następujący przebieg:

Wysłanie żądania – po wpisaniu w przeglądarce adresu URL np.:

<http://www.dmcs.p.lodz.pl:80/>, przeglądarka interpretuje jego poszczególne części.

Kolejne linie przykładowego żądania

GET /. HTTP/1.1

*Accept : image/gif, image/x-xbitmap, image-jpeg, image/pjpeg, */**

Accept-Language: pl

Accept-Encoding: gzip

User_Agent : Mozilla/4.0 (compatible; MSIE 5.01; Windows NT)

Host : www.dmcs.p.lodz.pl

Connection : Keep-Alive

Odpowiedź serwera

Po otrzymaniu żądania serwer odszukuje żądany zasób przypisany do adresu „/” i przesyła go do przeglądarki, poprzedzając nagłówkiem odpowiedzi.

Kolejne linie przykładowej odpowiedzi

HTTP/1.1 200 OK.

Date: Mon, 06 Dec 2000 15:32:54 GMT

Serwer Apache/1.3.6 Unix

Last-Modified: Fri, 04 Oct 2000 12:09:34 GMT

Etag: "8d7jk-876-901i7ki2"

Accept-Ranges: bytes

Content-length: 232

Connection: close

Content-type: text/html

<title> DMCS </title>

.....

Serwer Apache

- Można znaleźć pod adresem <http://www.apache.org>
- Darmowy i dostępny na wszystkie platformy.
- Standardowo w większości dystrybucji Linuksa.
- Ponad 80% witryn działa w oparciu o Apache.

Standardowe miejsce w systemie:

Windows: C:\Program Files\Apache Group\Apache

Linux: /usr/local/apache

Ważniejsze podkatalogi:

- bin
- conf
- logs
- htdocs → index.html

Instalacja na Linuksie:

```
tar -xzf apache_1.3.9.tar.gz
```

```
cd apache_1.3.9
```

```
./configure --with-layout=Apache --prefix=/usr/local/apache --with-port=80
```

```
make
```

```
make install
```

Pożyteczne zasoby:

Specyfikacje:

<http://www.w3c.org> :

Specyfikacja HTML

Specyfikacja CSS

Edytory html:

HomeSite

<http://www.allaire.com>

Web Pager Xpress

<http://adl.sonik.pl/webpager> (darmowy)

Pajaczek 2000

<http://www.creamsoft.com.pl> (komercyjny)

1'st Page 2000

<http://www.evrsoft.com/1stpage>

Grafika:

<http://www.clipartcity.com>

Prawa autorskie i zasady webmasterskie:

Można:

Umieszczać na swoich stronach linki do cudzych stron.

Nie można:

Kopiować innych stron i wstawiać ich do swojego serwisu.

Kopiować cudzych grafik bez pozwolenia.

Dobre zasady:

- Poprawność językowa stron (także brak literówek).
- Używanie polskich znaków w stronie kodowej ISO-8859-2.
- Umieszczanie grafiki w rozsądnej ilości i nie na stronie głównej.
- Uprzedzanie o stronach nasyconych grafiką powyżej 50 kb.
- Informowanie o wielkości pobieranych plików.
- Podawanie daty aktualizacji serwisu (ew. Co nowego?).
- **Korzystanie z generatorów stron tylko w fazie ogólnego projektu strony.**

WWW (World Wide Web)

WWW jest to sieć zasobów informacyjnych opierająca się na trzech mechanizmach:

- Jednolity schemat nazw stosowany wewnątrz całej sieci – **URI** (Universal Resource Identifier)
- Protokół służący do dostępu do nazwanych zasobów poprzez sieć
- HyperText służący do łatwego poruszania się po zasobach.
-

Każdy zasób musi być identyfikowany poprzez jednoznaczny **URI** składający się zwykle z trzech części:

- Nazwa mechanizmu używanego aby uzyskać zasoby (http, mailto, ftp)
- Nazwa komputera przechowującego dany zasób
- Nazwa zasobu, podana w postaci ścieżki.

Generalnie URI są case-sensitive i zawsze należy je pisać w takiej postaci.

URI może zawierać w sobie także identyfikator fragmentu zasobu. Jest ona oddzielona od adresu za pomocą znaku # (hasz).

Przykładowy URI:

```
http://somesite.com/html/top.html#section_2
```

URI mogą być podawane w postaci względnej:

```
<A href="page.html">MyPage</A>  
<IMG src="../../../icons/logo.gif" alt="logo">
```

HTML

Jednym z wielu realizacji języków hipertekstowych, stworzonych za pomocą **SGML**-a (Standard Generalized Markup Language – język służący do tworzenia znaczników opisujących dokumenty) jest **HTML** (Hypertext Markup Language). W niedługim czasie od swego powstania (1990) prawie całkowicie zdominował Internet.

Wybrany fragment dokumentu ograniczany jest znacznikami nadającymi pewne cechy.

Każdy znacznik składa się ze słowa kluczowego zawartego pomiędzy znakami „<” i „>”. Większość znaczników istnieje parami, tzn. znacznik otwierający i znacznik zamykający. Znacznik zamykający różni się tym, że zawarty jest pomiędzy znakami „</” i „>”.

Znaczniki nazywane są zamiennie **tagami**.

Każdy dokument hipertekstowy powinien składać się z trzech zasadniczych bloków:

```
<!DOCTYPE HTML PUBLIC ...>
```

```
<HTML>
```

```
  <HEAD>
```

```
    .....
```

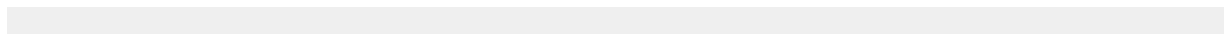
```
  </HEAD>
```

```
  <BODY>
```

```
    .....
```

```
  </BODY>
```

```
</HTML>
```



DTD (Document Type Definition) – jest to struktura opisująca zawartość dokumentu

Specyfikacja HTML 4 przewiduje trzy wersje prologu.

Jeśli dokument jest zgodny ze specyfikacją HTML 4.0.1, możemy stosować prolog:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
```

Wersja z adresem sieciowym:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"  
"http://www.w3.org/TR/html4/strict.dtd">
```

Częściej stosowaną wersją prologu jest tzw. wersja przejściowa ("luźna"):

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
```

Wersja z adresem sieciowym DTD ma postać:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"  
"http://www.w3.org/TR/html4/loose.dtd">
```

Szerszą odmianą Transitional jest prolog dla strony zawierającej ramki :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN">
```

I jego wersja z adresem:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Frameset//EN"  
"http://www.w3.org/TR/html4/frameset.dtd">
```

Jeśli dokument jest zgodny z wcześniejszą specyfikacją HTML 3.2, powinniśmy stosować prolog:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
```

lub

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2//EN">
```

Parsery przyjmują takie deklaracje jako podstawę badania poprawności składniowej. Najlepiej skorzystać z oficjalnego parsera World Wide Web Consortium, opartego na specyfikacji HTML 4.0.:

W3C HTML Validation Service (<http://validator.w3c.org>)

Przykładowy fragment DTD

```
<!--===== Forms =====>
<!ELEMENT FORM - - (%block;|SCRIPT)+ -(FORM) -- interactive form -->
<!ATTLIST FORM
  %attrs;                                -- %coreattrs, %il8n, %events --
  action      %URI;                      #REQUIRED -- server-side form handler --
  method      (GET|POST)                 GET        -- HTTP method used to submit the
form--
  enctype     %ContentType;              "application/x-www-form-urlencoded"
  accept      %ContentTypes;             #IMPLIED  -- list of MIME types for file
upload --
  name        CDATA                     #IMPLIED  -- name of form for scripting --
  onsubmit    %Script;                  #IMPLIED  -- the form was submitted --
  onreset     %Script;                  #IMPLIED  -- the form was reset --
  accept-charset %Charsets;             #IMPLIED  -- list of supported charsets --
>
```

Podstawowe elementy HTML

`<title> My Page </title>`

- zawarty zawsze w sekcji *head*
 - powinien możliwie najwierniej odzwierciedlać zawartość strony
 - nie powinien przekraczać 40 znaków
-

`<meta>`

Jest to opcjonalne polecenie, które bardziej szczegółowo informuje o zawartości dokumentu, np. dla celów katalogowych czy indeksów, wykorzystywanych przez odpowiednie programy, np. wyszukiwarki sieciowe. Zawiera ono trzy atrybuty: HTTP-EQUIV, NAME i CONTENT w kombinacji pierwszy z trzecim lub drugi z trzecim. HTTP-EQUIV definiuje zmienne systemowe, natomiast NAME - zmienne użytkownika. Najbardziej zalecane META:

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

Uwaga! Aby dokument został poprawnie wyświetlony, też musi być zapisany w ISO-8859-2 !!!

`<META NAME="Description" CONTENT="Some description">`

Zaleca się nie przekraczać 200 znaków.

`<META NAME="Keywords" CONTENT="keyword1, keyword2, keyword3">`

Zaleca się nie przekraczać 1000 znaków.

`<META NAME="Author" CONTENT="firstname and lastname">`

`<META NAME="Generator" CONTENT="Editor name">`

`<META HTTP-EQUIV="expires" CONTENT="Wed, 26 Apr 2001 08:21:57 GMT">` (Data zdefiniowana w formacie RFC 850)

`<META HTTP-EQUIV="Pragma" CONTENT="no-cache">`

`<META HTTP-EQUIV="Refresh" CONTENT="x">`

`<META HTTP-EQUIV="Refresh" CONTENT="x; URL=http://.../strona.html">`

<META name="robots" content="index, nofollow">

Zamiast pisać **index, follow**, możemy podać **all**.

Zamiast **noindex, nofollow** możemy użyć **none**.

Uwaga: nie wszystkie roboty sieciowe rozpoznają tę dyrektywę.

<base>

Definiuje bazowy adres dokumentu. Np:

<BASE HREF="http://www.firma.com/dokument.html">

Polecenie w tej postaci zapewnia poprawność relatywnych odsyłaczy w dokumencie.

<body>

<BODY BACKGROUND="URL albo ścieżka/nazwa_pliku.gif">

Deprecated(przestarzałe - zastąpione innymi formami):

<BODY BGCOLOR="kolor">

<BODY TEXT="kolor">

<BODY LINK="kolor1" VLINK="kolor2" ALINK="kolor3">

Czcionki (Deprecated):

 Przykład:

To jest czcionka pogrubiona (bold)

<I> </I> Przykład:

To jest czcionka pochylona (italic)

<BLINK> </BLINK> Przykład:

To jest czcionka migająca (blink) - jest to rozszerzenie Netscape'a, nie znajdujące się w specyfikacji HTML 4.

<U> </U> Przykład:

To jest czcionka podkreślona (underline)

<TT> </TT> Przykład:

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

<STRIKE> </STRIKE> Przykład:

~~Czcionka przekreślona~~

Alternatywnie jest używane równoważne polecenie <S></S>.

 Przykład:

Czcionka z ^{superskrytem}

 Przykład:

Czcionka z _{subskrytem}

<BIG> </BIG> Przykład:

Duża czcionka

<SMALL> </SMALL> Przykład:

Mała czcionka

Znaczniki logiczne - style

<CITE> </CITE> Przykład:

To jest cytat (citation)

<DFN> </DFN>

<DFN>, czyli Defining Instance, jest poleceniem stosowanym do wyróżniania jakiegoś bloku tekstu, np. pojawiającego się po raz pierwszy w danym tekście terminu. Przeglądarka interpretuje DFN jako kursywę (IE4) lub nie wyróżnia czcionki (NC4). Na przykład:

Internet Explorer 6 to najnowsza przeglądarka firmy Microsoft.

Przy najrozmaitszych rewizjach tekstów możemy się posłużyć poleceniem , które sygnalizuje zmiany. Pokazuje ono przekreślony fragment tekstu, np.:

Dokument dotyczący naszej polityki sprzedaży nie jest już dostępny.

Polecenie jest interpretowane przez IE4 i 5.

Gdy chcemy wyraźnie podkreślić jakąś nowość, możemy się posłużyć poleceniem INS.

<ins>To jest nowe opracowanie tematu.</ins>

To jest nowe opracowanie tematu

Polecenie jest interpretowane przez IE4 i 5.

Skrót i akronim:

<ABBR lang="pl">Prof.</ABBR>

<ACRONYM lang="en">WWW</ACRONYM>

 Przykład:

To jest czcionka mocno wyróżniona (strong)

 Przykład:

To jest czcionka wyróżniona (emphasis)

<CODE> </CODE> Przykład:

To jest kod (code)

<KBD> </KBD> Przykład:

Czczionka z klawiatury

<SAMP> </SAMP> Przykład:

Przykład (sample)

<VAR> </VAR>Przykład:

Zmienna (variable)

Kolory (Deprecated)

Składnia:

Nazwy kolorów:

1) w postaci angielskich nazw: *red, blue, white itd.*

2) w postaci kodu szesnastkowego określającego kolory RGB:

#000000 - czarny

#FFFFFF- biały

#FF0000- czerwony

Wielkość czcionki (Deprecated)

Czcionka normalna ma przypisaną wartość 3 . Pozostałe wartości są zawarte w przedziale od 1 do 7.

Czcionka bazowa (Deprecated)

Czcionka ma domyślną wielkość 3, ale w dowolnym momencie możemy ją zmienić za pomocą polecenia:

<BASEFONT SIZE="x">

Krój czcionki (Deprecated)

Jeszcze jedną możliwość urozmaicenia dokumentu daje polecenie . Pozwala ono zmienić krój czcionki dla danego dokumentu. Można także podać listę krojów oddzielonych przecinkami.

Nagłówki (Deprecated)

`<h1>Największy</h1>` **Największy**

.....

`<h6>Najmniejszy </h6>` **Najmniejszy**

Atrybut title:

`<H1 TITLE="To jest dymek do nagłówka h1">Nagłówek H1</H1>`

Wyrównanie nagłówków:

`<Hx ALIGN=CENTER> </Hx>`

Przykład:

To jest tytuł na środku strony

`<Hx ALIGN=RIGHT> </Hx>`

To jest tytuł justowany do prawego marginesu

`<Hx ALIGN=LEFT> </Hx>`

To jest tytuł justowany do lewego marginesu

Znacznik akapitu

`<p> </p>`

Znacznik końca wiersza

`<br>`

Pozioma linia:

<HR> Przykład:



Linia może być pozbawiona cieniowania:

<HR NOSHADE>



Linia może mieć dowolną grubość:

<HR SIZE=5>



Linia może mieć określoną długość w pikselach:

<HR WIDTH=300>



lub w procencie szerokości strony:

<HR WIDTH=50%>



Linia może być umieszczona na środku (domyślnie):

<HR ALIGN=center>

<HR ALIGN=center SIZE=8 WIDTH=200>



Może być justowana do np. lewego brzegu strony:

<HR ALIGN=left>

<HR ALIGN=left NOSHADE SIZE=4 WIDTH=300>

Linia może mieć kolor:

<hr color="nazwa_koloru">

<PRE> </PRE>

Tekst preformatowany, wyświetlany czcionką monotypiczną, uwzględnia spacje i tabulatory.

<BLOCKQUOTE></BLOCKQUOTE>

Blok cytowany, uwzględnia znaki tabulacji. Nadaje się do fragmentów wierszy.

<Q lang="pl"> </Q>

Cytat (" lub "). Nie jest interpretowany przez wszystkie przeglądarki.

<ADDRESS> </ADDRESS>

Polecenie jest interpretowane jako kursywa.

<CENTER> </CENTER>

Polecenie służy do środkowania na stronie wskazanego elementu

<FIELDSET>element</FIELDSET>

Otacza tekst obramowaniem. Ciekawym zastosowaniem jest użycie go w połączeniu ze znacznikiem LEGEND:

```
<FIELDSET>  
<LEGEND>Elementy</LEGEND>  
pierwszy element  
drugi element  
trzeci element  
</FIELDSET>
```

Odsyłacze(linki)

Odsyłacz jest wskazaniem jakiegoś innego miejsca. Kliknięcie na takim wskazaniu przenosi użytkownika do docelowego miejsca, przy czym może to być z powodzeniem miejsce na tej samej stronie, inna strona w ramach tego samego, złożonego dokumentu, a nawet strona na drugim końcu świata. Można wyróżnić dwa rodzaje linków: do etykiet i do stron.

1) Etykiety

```
<A NAME="nazwa_etykiety"> </A>
```

2) Linki

```
<A HREF="miejsce_docelowe">Tekst, na którym należy kliknąć</A>
```

Należy pamiętać, że adresy linków to nic innego jak URI, czyli są case-sensitive!

Linki mogą wskazywać na dowolnego typu plik. Jeżeli jest to plik html, zostanie otworzony bezpośrednio w przeglądarce. Także pliki popularnych formatów takich jak .txt, .gif zostaną przez nią wyświetlone. Jeśli przeglądarka potrafi wyświetlić zawartość danego pliku, zrobi to. W przeciwnym razie zaproponuje zapisanie go na dysk.

Przykłady linków:

Odsyłacz do innej strony HTML (w tym samym katalogu):

```
<A HREF="nazwa_strony.html">strona</A>
```

Odsyłacz do innej strony w katalogu podrzędnym

```
<A HREF="katalog_podrzeczny/nazwa_strony.html">strona</A>
```

Odsyłacz do strony w innym katalogu, równorzędnym w hierarchii:

```
<A HREF="../katalog_rownorzeczny/nazwa_strony.html">strona</A>
```

Odsyłacz do etykiety:

```
<A HREF="strona.htm#etykieta">Tekst</A>
```

Odsyłacz będący adresem email

Napisz do mnie

Bardzo ważny jest napisanie konstrukcji mailto:aaa@aaa.com.pl bez żadnych spacji!

Aby w momencie wysyłania listu pojawił się jakiś domyślny tekst:

Adres odbiorcy, do którego idzie kopia listu:

Ukryta kopia, czyli BCC (żaden inny odbiorca listu nie zobaczy tego adresu):

Można też zdefiniować fragment ciała listu:

Pocztę może też otrzymać kilka osób jednocześnie, adresowanych w polu głównym. Adresy trzeba rozdzielić średnikami:

Aby złączyć wszystko razem, należy posłużyć się znakiem & (ampersand):

Wyślij pocztę

Inne rodzaje odsyłaczy:

Konstrukcja wskazująca na katalog FTP ma postać:

Jakiś tekst

Konstrukcja wskazująca na bezpośredni plik:

Jakiś tekst

Konstrukcja wskazująca na plik na dysku lokalnym:

,

Odsyłacz do grupy dyskusyjnej:

```
<A HREF=news:pl.comp.www></A>
```

Odsyłacz do grupy dyskusyjnej na konkretnym serwerze:

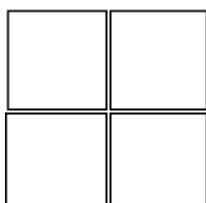
```
<A HREF="news://news.fuw.edu.pl">news.fuw.edu.pl</a>
```

Odsyłacz do konkretnej grupy dyskusyjnej:

```
<A HREF="news://news.fuw.edu.pl/pl.comp.www">grupa dyskusyjna  
pl.comp.www</a>
```

Odsyłacze będące mapami bitowymi

Niech będzie dany obrazek o wymiarach 100x100:



Aby podzielić go na cztery odsyłacze do różnych stron należy posłużyć się poleceniem:

```
<IMG SRC="odsyłacz.gif " USEMAP="#mapa1">  
<MAP NAME ="mapa1">  
<AREA SHAPE=RECT COORDS="1, 1, 50, 50" HREF="1.html">  
<AREA SHAPE=RECT COORDS="51, 1, 100, 50" HREF="2.html">  
<AREA SHAPE=RECT COORDS="1, 51, 51, 100" HREF="3.html">  
<AREA SHAPE=RECT COORDS="51, 51, 100, 100" HREF="4.html">  
</MAP>
```

Oprócz prostokątów można tworzyć mapy z użyciem okręgów i obszarów nieregularnych:

W przypadku okręgu stosujemy polecenie:

```
<AREA SHAPE=CIRCLE COORDS=" x, y, r" HREF="dokument.html">
```

W przypadku obszaru nieregularnego (przynajmniej 3 pary współrzędnych):

```
<AREA SHAPE=POLYGON COORDS=" 23, 33, 82, 16, 82, 80"  
HREF="dokument.html">
```

Listy i wykazy

Wykaz nieuporządkowany (unordered list):

Pierwszy punkt

Drugi punkt

Trzeci punkt

Przykład:

- Pierwszy punkt
- Drugi punkt
- Trzeci punkt

Zagnieżdżony wykaz nieuporządkowany:

Przykład:

- Punkt 1
- Punkt 2
 - Podpunkt 2.1
 - Podpunkt 2.2
 - Podpunkt 2.2.1
 - Podpunkt 2.2.2
- Punkt 3
- Punkt 4

Wykaz uporządkowany (ordered list):

Pierwszy punkt

Drugi punkt

Trzeci punkt

Przykład:

1. Pierwszy punkt
2. Drugi punkt
3. Trzeci punkt

Parametr **TYPE=n** pozwala określić typ numerowania listy

<OL TYPE=A> numerowanie według wielkich liter

<OL TYPE=a> numerowanie według małych liter

<OL TYPE=I> numerowanie według wielkich liczebników rzymskich

<OL TYPE=i> numerowanie według małych liczebników rzymskich

<OL TYPE=1> numerowanie według liczebników arabskich

Parametr **START=x** pozwala rozpocząć numerowanie listy od x

<OL START=x>

Lista uporządkowana też może być zagnieżdżana.

Lista definicyjna:

<DL> początek listy definicyjnej

<DT> pierwszy termin

<DD> wyjaśnienie pierwszego terminu

<DT> drugi termin

<DD> wyjaśnienie drugiego terminu

</DL> koniec listy definicyjnej

Polecenie DT powinno się znaleźć bezpośrednio za otwierającym tagiem DL

Przykład:

aaaa

opis aaaa

bbbb

opis bbbb

cccc

opis cccc

Tabele

Definicja tabeli:

```
<TABLE>  
  
<TR> <TD> </TD><TD> </TD><TD> </TD></TR>  
  
<TR> <TD> </TD><TD> </TD><TD> </TD></TR>  
  
<TR> <TD> </TD><TD> </TD><TD> </TD></TR>  
  
</TABLE>
```

Obramowanie tabeli:

```
<TABLE BORDER=10> </TABLE>
```

a1	a2	a3	a4	a5
b1	b2	b3	b4	b5
c1	c2	c3	c4	c5

Odległość między komórkami:

```
<TABLE BORDER CELSPACING=8> </TABLE>
```

a1	a2	a3	a4	a5
b1	b2	b3	b4	b5
c1	c2	c3	c4	c5

Marginesy dla komórek:

```
<TABLE BORDER CELSPACING=5 CELLPADDING=15> </TABLE>
```

a1	a2	a3	a4	a5
b1	b2	b3	b4	b5
c1	c2	c3	c4	c5

Szerokość tabeli:

```
<TABLE BORDER WIDTH=600> </TABLE>
```

a1	a2	a3	a4	a5
b1	b2	b3	b4	b5
c1	c2	c3	c4	c5

Zamiast wartości absolutnej w pikselach możemy także użyć wartości procentowej, np. pół szerokości ekranu przeglądarki.

```
<TABLE BORDER WIDTH=50%> </TABLE>
```

a1	a2	a3	a4	a5
b1	b2	b3	b4	b5
c1	c2	c3	c4	c5

Szerokość komórki:

```
<TD WIDTH=100> </TD>
```

a1	a2	a3	a4	a5
b1	b2	b3	b4	b5
c1	c2	c3	c4	c5

Wyrównanie tabeli:

<TABLE BORDER ALIGN=right> </TABLE>

<TABLE BORDER ALIGN=left> </TABLE>

Poziome wyrównanie danych w komórkach:

<TD ALIGN=center></TD>

<TD ALIGN=left> </TD>

<TD ALIGN=right> </TD>.

Wyrównanie jest wyraźnie widoczne, gdy samodzielnie zdefiniujemy szerokość komórki za pomocą WIDTH.

a1 - do lewej	a2 - środkowanie	a3 - do prawej
b1 - do lewej	b2 - środkowanie	b3 - do prawej
c1 - do lewej	c2 - środkowanie	c3 - do prawej

Wysokość tabeli: (nie wchodzi w skład specyfikacji):

<TABLE BORDER WIDTH="50%" HEIGHT="30%">

Pionowe wyrównanie danych w komórkach:

<TD VALIGN=top> </TD>

<TD VALIGN=middle> </TD>

<TD VALIGN=bottom> </TD>.

Kolor tła tabeli:

. <TABLE BORDER HEIGHT=200 BGCOLOR=yellow>

Kolory - *Black, Olive, Teal, Red, Blue, Maroon, Navy, Gray, Lime, Fuchsia, (white), Green, Purple, Silver, Yellow, Aqua*

Kolor obramowania tabeli:

<TABLE BORDER=5 BORDERCOLOR=red>.

a1	a2	a3	a4	a5
b1	b2	b3	b4	b5
c1	c2	c3	c4	c5

Odcień obramowania tabeli:

<TABLE BORDER=5 BORDERCOLORDARK=black
BORDERCOLORLIGHT=white>

a1	a2	a3	a4	a5
b1	b2	b3	b4	b5
c1	c2	c3	c4	c5

Tytuł i podpis tabeli:

<TABLE BORDER WIDTH=300>

<CAPTION ALIGN=top>Tytuł tabeli umieszczony u góry</CAPTION>

Microsoft Internet Explorer pozwala dodatkowo dosunąć podpis do lewej lub prawej strony, przy użyciu parametru VALIGN. Na przykład:

<TABLE BORDER WIDTH=300>

<CAPTION VALIGN=top **ALIGN=left**>Tytuł tabeli umieszczony u góry</CAPTION>

<TABLE BORDER WIDTH=300>

<CAPTION VALIGN=bottom **ALIGN=right**>Tytuł tabeli umieszczony u dołu z prawej strony</CAPTION>

Nagłówki wierszy i kolumn:

```
<TABLE BORDER CELLPADDING=10 WIDTH=400>
```

```
<TR><TD></TD><TH>1991</TH> <TH>1992</TH> <TH>1993</TH>  
<TH>1994</TH> <TH>1995</TH> </TR>
```

```
<TR ALIGN=center><TH>aaa</TH>  
<TD>1150</TD><TD>1240</TD><TD>1380</TD><TD>1420</TD><TD>1550</TD></TR>
```

```
<TR ALIGN=center><TH>bbb</TH>  
<TD>800</TD><TD>900</TD><TD>980</TD><TD>1150</TD><TD>1320</TD>  
></TR>
```

```
</TABLE>
```

	1991	1992	1993	1994	1995
aaa	1150	1240	1380	1420	1550
bbb	800	900	980	1150	1320

Non breaking space:

Aby komórki puste były dobrze wyświetlane przez przeglądarki należy wstawiać w nie znak "nie łamliwej" spacji: ** **

Łączenie komórek w pionie i poziomie:

Aby komórka rozciągała się na x wierszy należy użyć parametru ROWSPAN:

```
<TD ROWSPAN=x>
```

Aby komórka rozciągała się na x column należy użyć parametru COLSPAN:

```
<TD COLSPAN=x>.
```

Zagnieżdżanie tabel:

100	200	300	
		AAA	BBB
		CCC	DDD
400	500	600	

Nagłówek, ciało i stopka tabelki

```
<TABLE WIDTH=50% BORDER=5 CELLPADDING=5>
<THEAD>To jest część nagłówkowa</THEAD>
<TBODY>
<TR>
<TD>Komórka 11</TD>
</TR>
<TR>
<TD>Komórka 21</TD>
</TR>
</TBODY>
<TFOOT>To jest stopka</TFOOT>
</TABLE>
```

Obramowanie tabel

`<TABLE BORDER FRAME=x>`.

x=void pozwala usunąć zewnętrzne obramowanie tabeli.

x=above wstawia górną krawędź obramowania, a **x=below** - dolną.

x=vsides wyświetla lewą i prawą krawędź obramowania tabeli, a **x=hsides** - górną i dolną.

x=lhs wyświetla lewą krawędź obramowania, a **x=rhs** - prawą.

x=box wyświetla wszystkie krawędzie obramowania.

Linie wewnętrzne tabel:

`<TABLE BORDER RULES=x>`

x=none powoduje usunięcie linii wewnętrznych.

x=rows powoduje wyświetlenie tylko poziomych linii wewnętrznych.

x=cols powoduje wyświetlenie tylko pionowych linii wewnętrznych.

x=all powoduje wyświetlenie wszystkich linii wewnętrznych.

RAMKI

Strona (zestaw stron) składająca się z ramek ma następującą budowę:

- **strona główna** definiująca rozmieszczenie ramek .
- **strony podrzędne** będące treścią ramek.

Przykładowa strona zawierająca ramki:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>DMCS</TITLE>
```

```
</HEAD>
```

```
<FRAMESET COLS="25%,*%">
```

```
<FRAME SCROLLING="auto" NAME="Menu" SRC="menu.htm">
```

```
<FRAME SCROLLING="yes" NAME="Main" SRC="main.htm">
```

```
<NOFRAMES>
```

```
<BODY>
```

Zawartość strony dla przeglądarek nie obsługujących ramek

```
</BODY>
```

```
</NOFRAMES>
```

```
</FRAMESET>
```

```
</HTML>
```

Usunięcie linii dzielących ramki:

```
<frameset cols="20%,*" framespacing="0" frameborder="0" border="0">
```

Podział strony na wiersze:

```
<FRAMESET ROWS="200,*,200">
```

Parametry ramek:

Źródło ramki:

<FRAME SRC="nazwa_pliku.htm">

Belki przewijające:

<FRAME SCROLLING=yes>

<FRAME SCROLLING=no>

<FRAME SCROLLING=auto>

Nazwa ramki:

<FRAME NAME="nazwa">

Uniemożliwienie zmiany rozmiaru ramki:

<FRAME NORESIZE>

Zagnieżdżanie definicji ramek:

<FRAMESET ROWS=200,*,200>

 <FRAME SRC=strona6.htm>

<FRAMESET COLS=35%,*,35%>

 <FRAME SRC=strona7.htm>

 <FRAME SRC=strona8.htm>

 <FRAME SRC=strona9.htm>

</FRAMESET>

<FRAMESET COLS=50%,50%>

 <FRAME SRC=strona10.htm>

 <FRAME SRC=strona11.htm>

</FRAMESET>

</FRAMESET>

Zmiana zawartości ramek z poziomu innych ramek:

```
<A HREF="aaa.htm" TARGET="main">Main</A>
```

Zastrzeżone wartości parametru target:

```
TARGET="_blank"
```

```
TARGET="_self"
```

```
TARGET="_top"
```

Pływające ramki (inline frames):

```
<IFRAME WIDTH=100 HEIGHT=100 SRC="doc.htm">Twoja przeglądarka nie  
akceptuje pływających ramek!</IFRAME>
```

Aby pozbyć się obramowania ramki należy użyć wewnątrz definicji parametru frameborder:

```
FRAMEBORDER=0
```

Aby nie było suwaków:

```
SCROLLING=no
```

Elementy multimedialne

Powszechnie są przyjęte dwa standardy graficzne: JPG i GIF (znaczna kompresja). Nie jest przyjętym standardem format BMP.

Wstawienie obrazka:

```

```

Wysokość i szerokość obrazka:

```

```

Ramka wokół obrazka:

```
 <!-- Brak ramki : border = 0 -->
```

Odległość od oblewającego tekstu:

```

```

Pływający tekst (tylko IE):

<MARQUE>tekst<MARQUE>

Dźwięki:

Wszystkie elementy dźwiękowe wymagają posiadania przez klienta odpowiednich wtyczek. Element dźwiękowy musi więc być jedynie dodatkiem do strony, a nie jego główną treścią.

Plik dźwiękowy WAV (styl Netscape) - <embed src="../wav/complete.wav" border=0 width=x height=y>

Plik dźwiękowy WAV (styl Microsoft) <bgsound src="../wav/complete.wav" loop=infinite>

Plik dźwiękowy MIDI (styl Netscape) - <embed src="../midi/airwolf.mid" border=0>

Parametry:

<i>loop=x</i>	liczba powtórzeń dźwięku
<i>width=x height=y</i>	określa rozmiary wyświetlanego obrazka wtyczki
<i>autostart=true false</i>	nakazuje lub zakazuje automatycznego startu odtwarzania
<i>hidden=true false</i>	nakazuje schować wyświetlić obrazek wtyczki
<i>palette=foreground background</i>	pozwala uniknąć ewentualnego migotania palety kolorów, gdy przeglądarka pracuje w trybie 256 kolorów
<i>pluginspace="url"</i>	wskazuje miejsce, gdzie znajduje się wtyczka, jeśli użytkownik jej jeszcze nie zainstalował. Adres ten powinien być zgodny ze wskazaniami producenta wtyczki.

Tekst dla starszych przeglądarek nie obsługujących polecenia EMBED:

<EMBED SRC="plik.mid" WIDTH="200" HEIGHT="150">
<NOEMBED>Potrzebujesz Navigatora lub Internet Explorera aby to usłyszeć</NOEMBED>
</EMBED>

Inne pliki dźwiękowe: każdy ma osobny zestaw parametrów. Im bardziej popularny format dźwiękowy, tym większa szansa, że zostanie usłyszany.

Pliki flash:

Różne umieszczanie w zależności od przeglądarki. Wymaga kodu w języku JavaScript.

Znaki specjalne:

(HTML Reference Library 4)

Numer	Nazwa	Opis	Reprezentacja w przeglądarce
&	&	Ampersand	&
<	<	Less than	<
>	>	Greater than	>
 	 	Non-breaking space	
¡	¡	Inverted exclamation	¡
¢	¢	Cent sign	¢
£	£	Pound sterling sign	£
¤	¤	General currency sign	¤
¥	¥	Yen sign	¥
¦	¦	Broken vertical bar	
§	§	Section sign	§
¨	¨	Umlaut (dierisis)	¨
©	©	Copyright	©
ª	ª	Feminine ordinal	ª
«	«	Left angle quote, guillemot left	«
¬	¬	Not sign	¬
­	­	Soft hyphen	
®	®	Registered trademark	®
¯	¯	Macron accent	¯
°	°	Degree sign	°
±	±	Plus or minus	±
²	²	Superscript two	²
³	³	Superscript three	³
´	´	Acute accent	´
µ	µ	Micro sign	µ
¶	¶	Paragraph sign	¶
·	·	Middle dot	·
¸	¸	Cedilla	¸
¹	¹	Superscript one	¹
º	º	Masculine ordinal	º
»	»	Right angle quote, guillemot right	»
¼	¼	Fraction (one quarter)	¼
½	½	Fraction (one half)	½
¾	¾	Fraction (three quarters)	¾
¿	¿	Inverted question mark	¿

À	À	Capital A, grave accent	À
Á	Á	Capital A, acute accent	Á
Â	Â	Capital A, circumflex accent	Â
Ã	Ã	Capital A, tilde	Ã
Ä	Ä	Capital A, umlaut (dierisis)	Ä
Å	Å	Capital A, ring	Å
Æ	Æ	Capital AE diphthong (ligature)	Æ
Ç	Ç	Capital C, cedilla	Ç
È	È	Capital E, grave accent	È
É	É	Capital E, acute accent	É
Ê	&Eirc;	Capital E, circumflex accent	Ê
Ë	Ë	Capital E, umlaut (dierisis)	Ë
Ì	Ì	Capital I, grave accent	Ì
Í	Í	Capital I, acute accent	Í
Î	Î	Capital I, circumflex accent	Î
Ï	Ï	Capital I, umlaut (dierisis)	Ï
Ð	Ð	Capital Eth, Icelandic	Ð
Ñ	Ñ	Capital N, tilde	Ñ
Ò	Ò	Capital O, grave accent	Ò
Ó	Ó	Capital O, acute accent	Ó
Ô	Ô	Capital O, circumflex accent	Ô
Õ	Õ	Capital O, tilde	Õ
Ö	Ö	Capital O, umlaut (dierisis)	Ö
×	×	Multiply sign	×
Ø	Ø	Capital O, slash	Ø
Ù	Ù	Capital U, grave accent	Ù
Ú	Ú	Capital U, acute accent	Ú
Û	Û	Capital U, circumflex accent	Û
Ü	Ü	Capital U, umlaut (dierisis)	Ü
Ý	Ý	Capital Y, acute accent	Ý
Þ	Þ	Capital Thorn, Icelandic	Þ
ß	ß	Small sharp s, German (sz ligature)	ß
à	à	Small a, grave accent	à
á	á	Small a, acute accent	á
â	â	Small a, circumflex accent	â
ã	ã	Small a, tilde	ã
ä	ä	Small a, umlaut (dierisis)	ä
å	å	Small a, ring	å
æ	æ	Small ae diphthong (ligature)	æ
ç	ç	Small c, cedilla	ç
è	è	Small e, grave accent	è

é	é	Small e, acute accent	é
ê	ê	Small e, circumflex accent	ê
ë	&euuml;	Small e, umlaut (dierisis)	ë
ì	ì	Small i, grave accent	ì
í	í	Small i, acute accent	í
î	î	Small i, circumflex accent	î
ï	ï	Small i, umlaut (dierisis)	ï
ð	ð	Small eth, Icelandic	ð
ñ	ñ	Small n, tilde	ñ
ò	ò	Small o, grave accent	ò
ó	ó	Small o, acute accent	ó
ô	ô	Small o, circumflex accent	ô
õ	õ	Small o, tilde	õ
ö	ö	Small o, umlaut (dierisis)	ö
÷	÷	Division sign	÷
ø	ø	Small o, slash	ø
ù	ù	Small u, grave accent	ù
ú	ú	Small u, acute accent	ú
û	û	Small u, circumflex accent	û
ü	ü	Small u, umlaut (dierisis)	ü
ý	ý	Small y, acute accent	ý
þ	þ	Small thorn, Icelandic	þ
ÿ	ÿ	Small y, umlaut (dierisis)	ÿ
Œ	Œ	Latin Capital OE (ligature)	Œ
œ	œ	Latin Small OE (ligature)	œ
Š	Š	Capital S with caron	Š
š	š	Small s with caron	š
Ÿ	Ÿ	Capital Y with dierisis	Ÿ
ˆ	ˆ	Circumflex accent	^
˜	˜	Small tilde	~
–	–	En dash	–
—	—	Em dash	—
‘	‘	Left single quotation mark	`
’	’	Right single quotation mark	'
‚	‚	Single low-9 quotation mark	,
“	“	Left double quotation mark	"
”	”	Right double quotation mark	"
„	„	Double low-9 quotation mark	„
†	†	Dagger	†
‡	‡	Double Dagger	‡
‰	‰	Per mille (thousand) sign	‰

‹	&lquo;	Single left-pointing angle quotation mark	<
›	&rquo;	Single right-pointing angle quotation mark	>
€	€	Euro sign*	€

Formularze

<FORM ACTION="url" METHOD="Post">

....

</FORM>

Atrybuty:

action - adres zasobu, do którego trafi request. Może to być ta sama strona.

*method=**get/post* - metoda użyta do przesłania zawartości formularza.

get - dane idą w URL

post - dane idą w treści formularza

enctype - Standard użyty przy kodowaniu. Wartość domyślna to :
"application/x-www-form-urlencoded". Jeżeli jednym z pól formularza jest pole typu file, należy użyć typu: "multipart/form-data".

Pole typu input:

Np:

Pole tekstowe:

<input name="nazwa_pola" value="wartosc_pocatkowa" type="text">

Przycisk:

<input name="nazwa_pola" value="tekst_przycisku" type="button">

Atrybuty:

type - typ kontrolki. Domyślnym typem jest tekst. Dozwolone atrybuty to :

text|password|checkbox|radio|submit|reset|file|hidden|image|button

name - nazwa kontrolki

value - wartość początkowa

size - rozmiar w pikselach (lub znakach dla pól typu password lub text)

CSS (Cascade Style Sheets)

Składnia stylu ma postać:

selektor { cecha: wartość }

np:

P {font-family: Arial}

H1 {font-size: 20pt}

Poszczególne pary cecha-wartość (property-value) rozdziela się od siebie średnikami:

<P STYLE="font-size: 20pt; font-weight: bold; background: green; color: red">Czcionka 20pt, pogrubiona, koloru czerwonego na zielonym tle</P>

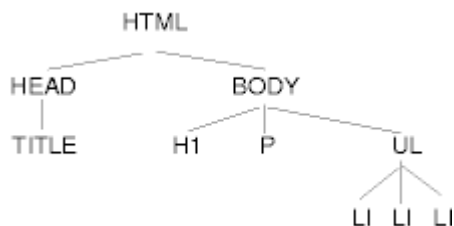
Jeżeli chcemy przypisać konkretny styl dla większej ilości selektorów, oddzielamy je przecinkami:

H1, H2, H3 {font-family:Helvetica; color:yellow}

Pojęcia wykorzystywane przy definiowaniu stylów:

Document tree (drzewo dokumentu) - Drzewo przedstawiające hierarchiczny układ elementów w postaci kolejnych tagów html. Każdy element oprócz elementu *root* (HTML) ma dokładnie jednego rodzica.

np:



Child (dziecko) - element znajdujący się o 1 poziom niżej w hierarchii w stosunku do danego elementu

Descendant (potomek) - element znajdujący się o 1 lub więcej poziomów niżej w hierarchii w stosunku do danego elementu

Parent (rodzic) - element znajdujący się o 1 poziom wyżej w hierarchii w stosunku do danego elementu

Ascendant (przodek) - element znajdujący się o 1 lub więcej poziom wyżej w hierarchii w stosunku do danego elementu

Własności dla całego dokumentu można przypisać w elemencie <BODY>. To samo można uzyskać posługując się tzw. selektorem uniwersalnym:

```
* { color:#0000FF }
```

Descendant selector (selektor potomka):

Można nadawać cechy atrybutom posiadającym określonych przodków np:

1) h1 h2 b {color: blue} - definicja ta określa następującą własność:

Jeżeli pod elementem <h1> znajdzie się element <h2> i pod <h2> znajdzie się element , to fragment objęty znacznikiem uzyska kolor niebieski.

2) UL UL LI {color:green}

- p1
- p2
 - p2.1
 - p2.2
- p3

Child selector (selektor dziecka):

Przykład:

```
p>u {color=green}
```

Oznacza, że jeżeli podkreślenie wystąpi wewnątrz akapitu, będzie miało kolor zielony.

Sibling selector (selektor braci):

```
H1 + p {margin-left: +30 mm}
```

Oznacza, że jeżeli <h1> i <p> mają wspólnego rodzica, to akapit zostanie odsunięty w lewo.

Selektory atrybutów:

Ogólna postać:

element [atrybut relacja wartość] {deklaracja stylu}

np:

```
p [align=right] {color: blue}
```

Prosty selektor atrybutu:

np:

```
h3 [title] {color: #000000 }
```

Oznacza, że jeżeli element `<h3>` będzie posiadał atrybut `title` (o dowolnej wartości), to zostanie on przedstawiony w kolorze białym.

```
[title] {color: #000000 }
```

Oznacza, że jeżeli jakikolwiek element będzie posiadał atrybut `title` (o dowolnej wartości), to zostanie on przedstawiony w kolorze białym.

Selektor atrybutu o określonej wartości:

element [atribut="wartość"] { deklaracja stylu } lub:

```
[atribut="wartość"] { deklaracja stylu } lub
```

```
[atribut~="wartość"] { deklaracja stylu }
```

W powyższym przypadku "wartość" może stanowić część wartości atrybutu, aby warunek był spełniony.

Klasy elementów

Każdemu elementowi można nadać tzw. klasę stylu css. Polega to na dopisaniu atrybutu `class` do jakiegokolwiek elementu.

Przykładowa definicja stylu dla klas:

```
*.klasa1 {font-family:Arial}
```

i przykładowe wykorzystanie:

```
<p class="klasa1"> Ten tekst będzie czcionką Arial </p>
```

Każdemu elementowi można nadać odpowiedni styl za pomocą klasy w zależności od pojawiających się okoliczności.

Przykład definicji różnych stylów dla tego samego znacznika pokazuje następujący przykład:

```
P {font-family: Arial; font-size: 12pt}  
P.uwagi {font-family: Arial; font-size: 12pt; font-weight: bold}  
P.wazne {font-family: Times; font-size: 12pt; color: blue}  
P.przypisy {font-family: Arial; font-size: 8pt}
```

Natomiast w samym tekście będziemy przywoływać klasy za pomocą polecenia

```
<p class=konkretna_klasa>Akapit</p>
```

Selektor ID

Służy wyodrębnieniu konkretnego elementu w drzewie. Musi być unikalny. Musi zaczynać się od litery, może składać się z liter, cyfr, łączników i kropek.

```
#[etykieta ID] {deklaracja stylu}
```

np:

```
H1#title {letter-spacing: 0.5em}
```

Przykładowe wykorzystanie w tekście:

```
<h1 id="title">tytul</h1>
```

Selektory odsyłaczy

1) Odsyłacz na stronie:

```
a:link {color:blue; background: red}
```

2) Odwiedzony odsyłacz:

```
a:visited {color:yellow}
```

3) Odsyłacz reagujący na pojawienie się nad nim kursora myszki:

```
A:hover {background:yellow; color:red}
```

4) Odsyłacz aktywny w danym momencie:

```
A:active {background:blue; color:red}
```

Powyższe odsyłacze należy zadeklarować w takiej właśnie kolejności.

Selektory specyficzne dla tekstu

Selektor first-line

P:first-line {text-transform: uppercase}

Spowoduje wypisanie pierwszej linii akapitu dużymi literami.

Selektor first-letter

P:first-letter { font-size: 200%; color:red }

Spowoduje wypisanie pierwszej litery akapitu dwukrotnie większą czcionką i w czerwonym kolorze.

Kaskadowość i dziedziczenie

Dziedziczenie polega na tym, że elementy leżące niżej w drzewie dziedziczą cechy elementów leżących nad nimi, chyba że je specjalnie przeddefiniujemy.

np

```
<table class="czcionka_arial">
```

```
<tr> <td>Tekst czcionką Arial</td>
```

```
<td class="czcionka_courier">Tekst czcionką Courier</td></tr>
```

```
</table>
```

Dziedziczenie nie zawsze działa w Netscape.

Kaskadowość polega na tym, że style mogą być przeddefiniowywane. Style mogą być umieszczane w trzech miejscach dokumentu HTML: importowane w nagłówku, umieszczane w sekcji HEAD lub lokalnie definiowane. Dla danego elementu liczy się zawsze "najbliższa definicja".

Zastosowanie: ogólne arkusze dla całego serwisu, bardziej szczegółowe dla konkretnych działów, jeszcze bardziej szczegółowe dla poszczególnych stron.

Sposoby umieszczania stylów na stronie

1) Styl lokalny, np:

```
<p style="font-size: 12pt; font-weight: bold; font-family: 'Andale mono', 'Courier New', monospace">Treść akapitu</p>
```

2) Rozciąganie stylu - ma zastosowanie, gdy stylem chcemy objąć jakiś kawałek strony, nie będący np. akapitem

```
<span style="definicja stylu"> rysunek , tekst, akapit ,</span>
```

W poleceniu tym możemy także posłużyć się zdefiniowaną wcześniej klasą

3) Wydzielone bloki:

```
<div style="definicja stylu"> rysunek , tekst, akapit ,</div>
```

Zastosowanie tym się różni, że należy je stosować do logicznie wydzielonych fragmentów stron.

4) Umieszczanie definicji w sekcji <head> np:

```
<head>
<STYLE TYPE="text/css">
<!--
BODY {margin-left: 1cm; margin-right: 1cm}
P {font-size: 10pt; font-family: Arial, Helvetica; font-weight: normal}
-->
</STYLE>
</head>
```

5) Importowanie stylu z zewnętrznego arkusza

```
<style>

<!--

@import url("http://www.polbox.com.pl/lupus/style/toolstyl.css");

H1 {font-size: 30pt; font-family: Helvetica}

-->

</style>
```

Style czcionek

Rodziny czcionek, np:

`<p style="font-family: Times">Treść akapitu</p>`

`<p style="font-family: Times, 'Times New Roman', 'Times New Roman CE'">Treść akapitu</p>`

Poszczególne style rozdzielane są od siebie przecinkami i znakiem spacji (spacja jest ważna!). Jeżeli rodzina czcionek składa się z kilku wyrazów, należy ująć ją w znaki ' i ' ("uszy"). Cała definicja powinna być ujęta w znaki cudzysłowiu.

Można się także posługiwać pewnymi słowami kluczowymi:

`<p style="font-family: serif">Treść akapitu</p>`

Inne dostępne słowa kluczowe: -serif, cursive, fantasy, monospace itp.

Styl czcionki:

`<p style="font-style: normal">Treść akapitu</p>`

Dostępne słowa kluczowe: italic, oblique.

Wariant czcionki:

`<p style="font-variant: normal">Treść akapitu</p>`

Inny dopuszczalny styl: small-caps. np: SMALL-CAPS.

Waga czcionki:

`<p style="font-weight: waga">Treść akapitu wyświetlana za pomocą wybranej wagi</p>`

Waga czcionki decyduje o stopniu pogrubienia. Dopuszczalne są następujące wartości: normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900.

Wielkość czcionki:

1) podawane jako słowa kluczowe:

`<p style="font-size: słowo kluczowe">Treść akapitu</p>`

Słowa kluczowe: xx-small, x-small, small, medium, large, x-large, xx-large.

Wielkości relatywne: larger, smaller (w stosunku do domyślnej).

2) podawane w jednostkach miary:

<p style="font-size: 1cm"></p>

Jednostki: centymetry (cm), cale (in), milimetry (mm), punkty (pt)

3) podawane w procentach:

<p style="font-size: 150%"></p>

Umieszczanie i kolejność atrybutów

<p style="font: small-caps bold 14pt/18pt Times, 'Times New Roman', serif"></p>

AKAPIT MAŁE KAPITALIKI, POGRUBIONY, 14 PKT, 18 PKT ODSTĘPU MIĘDZY LINIAMI BAZOWYMI WIERSZY, TIMES.

Poprawna kolejność atrybutów:

font-style->font-variant->font-weight->font-size->line-height->font-family

Własności tekstu

Odstępy między literami:

<p style="letter-spacing: 3mm">3-milimetrowe odstępy między literami.</p>

Dekoracja tekstu:

To jest odsyłacz do Katedry DMCS.

Możliwe wartości: underline,overline,line-through,blink (NC).

Pusta przestrzeń:

<p style="white-space: normal">Tutaj jest siedem spacji</p>

Tutaj jest siedem spacji.

<p style="white-space: pre">Tutaj jest siedem spacji</p>

Tutaj jest siedem spacji.

Transformacja tekstu:

Istnieją cztery transformacje tekstu: none (brak), capitalize (pierwsze litery wyrazów będą duże), uppercase (wszystko dużymi literami), lowercase (wszystko małymi literami). Np:

`<p style="text-transform: capitalize">To jest jakiś tekst.</p>`

To Jest Jakiś Tekst.

`<p style="text-transform: uppercase">To jest jakiś tekst.</p>`

TO JEST JAKIŚ TEKST.

Wyrównanie tekstu:

`<p style="text-align: right">To jest tekst wyrównany do prawego marginesu.</p>`

Dopuszczalne wartości: left, right, center, justify

Wcięcie tekstu:

`<p style="text-indent: 10%">To jest akapit, w którym pierwszy wiersz będzie wcięty o 10% szerokości strony</p>`

To jest akapit, w którym pierwszy wiersz będzie wcięty o 10% szerokości strony. To jest akapit, w którym pierwszy wiersz będzie wcięty o 10% szerokości strony.

Odległość można także podawać posługując się jednostkami miar.

Odstępy między wierszami:

`<p style="line-height: 1cm">To jest akapit, w którym wiersze są odsunięte od siebie o 1 cm...</p>`

Można podawać także w postaci procentów.

Kolor i tło

Kolor można przypisać dowolnemu elementowi.

Kolor tekstu:

```
<h3 style="color: #00FF00">Zielony tytuł trzeciego stopnia</h3>
```

```
<table border>
```

```
<tr>
```

```
<td style="color: #FF0000">Zielony tekst</td>
```

```
</tr>
```

```
</table>
```

Kolor tła:

```
<p style="background-color: blue">Akapit na tle Aqua</p>
```

Akapit na tle Aqua. Akapit na tle Aqua. Akapit na tle Aqua. Akapit na tle Aqua.

Kolory systemowe - można wykorzystywać na stronie kolory występujące w ustawieniach komputera klienta.

Według specyfikacji dostępne są następujące kolory:

ActiveBorder - Active window border.

ActiveCaption - Active window caption.

AppWorkspace - Background color of multiple document interface.

Background - Desktop background.

ButtonFace - Face color for three-dimensional display elements.

ButtonHighlight - Dark shadow for three-dimensional display elements (for edges facing away from the light source). Shadow color for three-dimensional display elements.

ButtonShadow - Shadow color for three-dimensional display elements.

ButtonText - Text on push buttons.

CaptionText - Text in caption, size box, and scroll bar arrow box.

GrayText - Grayed (disabled) text. This color is set to #000 if the current display driver does not support a solid gray color.

Highlight - Item(s) selected in a control.

HighlightText - Text of item(s) selected in a control.

InactiveBorder - Inactive window border.

InactiveCaption - Inactive window caption.

InactiveCaptionText - Color of text in an inactive caption.

InfoBackground - Background color for tooltip controls.

InfoText - Text color for tooltip controls.

Menu - Menu background.

MenuText - Text in menus.

Scrollbar - Scroll bar gray area.

ThreeDDarkShadow - Dark shadow for three-dimensional display elements.

ThreeDFace - Face color for three-dimensional display elements.

ThreeDHighlight - Highlight color for three-dimensional display elements.

ThreeDLightShadow - Light color for three-dimensional display elements (for edges facing the light source).

ThreeDShadow - Dark shadow for three-dimensional display elements.

Window - Window background.

WindowFrame - Window frame.

WindowText - Text in windows.

Tło w postaci grafiki

`<h3 style="background-image: url(obrazek.gif)">Tytuł na tle obrazka</h3>`

Powtarzanie tła (gdy obrazek jest mniejszy od tła obiektu):

`<h3 style="background-image: url(obrazek.gif); background-repeat: repeat-x">Tytuł na tle obrazka</h3>`

Możliwe wartości: repeat-x, repeat-y, repeat, no-repeat

Zaczeplenie tła:

Obrazek będący tłem może być "zaczeplony" na dwa sposoby:

- może tkwić nieruchomo w stosunku do treści strony
- może tkwić nieruchomo w stosunku do ekranu

W tym drugim przypadku należy zastosować definicję:

```
BODY {background: url(nazwa_obrazka); background-attachment: fixed}
```

Pozycja tła:

Jeżeli tło jest mniejsze od wypełnianego obszaru można ustalić jego pozycję względem obszaru (można stosować także w tabelkach):

background-position: wartość

gdzie wartość może być następującymi słowami kluczowymi: top, bottom, center, left, right lub ich kombinacją, np:

background-position: top right.

Można także podać wartości liczbowe lub procentowe (liczone od lewego górnego rogu wypełnianego obszaru)(x,y):

background-position: 2cm 3cm

background-position: 30% 50%

Skrócony zapis wielu atrybutów tła:

```
BODY {background: url(pcq.gif) fixed no-repeat center}
```

Suwaki(tylko IE)

Powyższe rozszerzenie nie wchodzi w skład specyfikacji CSS i jest jedynie rozszerzeniem Explorera począwszy od wersji 5.5.

Pełen zestaw atrybutów dotyczący suwaków:

- Scrollbar-Base-Color: kolor powierzchni suwaka i strzałek oraz rynny suwaka
- Scrollbar-Face-Color: kolor powierzchni suwaka i strzałek, a także kolor składowy rynny suwaka
- Scrollbar-Arrow-Color: kolor strzałek
- Scrollbar-Highlight-Color: kolor składowy rynny suwaka
- Scrollbar-Shadow-Color: kolor płytszego cienia
- Scrollbar-Dark-Shadow-Color: kolor głębszego cienia
- Scrollbar-3dLight-Color: kolor zewnętrznego cienia

Strona eksperymentalna Microsoftu:

<http://msdn.microsoft.com/workshop/samples/author/dhtml/refs/scrollbarColor.htm>

Marginesy

Możemy definiować następujące marginesy:

- margin-top
- margin-bottom
- margin-left
- margin-right, np:

`<p style="margin-top: 1cm">Akapit</p>`

Wartość odstępu można definiować także z użyciem procentów.

Istnieje możliwość zdefiniowania marginesów z użyciem jednego polecenia *margin*:

`<p style="margin: 2cm">Akapit</p>` - dwucentymetrowy margines ze wszystkich stron

`<p style="margin: 2cm 3cm">Akapit</p>`- dwucentymetrowy margines z góry i dołu oraz trzycentymetrowy z lewej i prawej

`<p style="margin: 2cm 3cm 1cm 4cm">Akapit</p>` - indywidualne marginesy dla każdej ze stron.

Można także zdefiniować także tzw. wewnętrzne marginesy posługując się poleceniem *padding-x*, gdzie x jest analogiczne do polecenia *margin-x*. Sposób mieszania atrybutów także jest identyczny.

Obramowania

Są bardzo różnie realizowane przez różne przeglądarki. Najbardziej uniwersalne wartości są następujące:

border-x-width - grubość obramowania, gdzie x może być: top,bottom,left,right

border-color - kolor obramowania

border-style - styl obramowania, może przyjmować następujące wartości: none, dotted, dashed, solid, double, groove, ridge, inset, outset.

Można posługiwać się słowem kluczowym *border* do podawania definicji mieszanych, np:

border: thick blue double

Wykazy

Rodzaj wyróżnika wykazu:

list-style-type - może przyjmować następujące wartości: disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none, np:

```
<ul style="list-style-type: disc">
<li>pierwszy punkt
<li>drugi punkt
<li>trzeci punkt
</ul>
```

Obrazek będący wyróżnikiem wykazu:

```
<ul style="list-style-image: url(dot.gif)">
```

Wykazy zajmujące więcej niż jedną linię:

Można zdecydować, jak ma być "zawijany" wielolinijkowy wykaz, czyli czy wyróżnik wykazu ma się znajdować na zewnątrz czy wewnątrz tekstu, np:

list-style-position: inside

- [illegible]

list-style-position: outside

- [illegible]

Rozmiary elementów

Szerokość:

`<p style="width: 150px"> </p>` Akapit o szerokości 150 pikseli.

Można także odnosić do wykazów czy komórek tabeli.

Wysokość:

`<td style="height: 150px"> </td>`

Często pojawiają się także atrybuty definiujące maksymalną i minimalną wysokość i szerokość (max-height, min-height, max-width, min-width). Na razie nie są obsługiwane.

Pozycjonowanie:

np:

`<span style="position: relative; left: 200px; top: 50px; width: 400px">Ten akapit jest odsunięty od lewego górnego rogu miejsca jego zamocowania o 200,50 pikseli i ma szerokość 400 pikseli</span>`

Obrazek umieszczony na stałe w lewym górnym rogu ekranu:

`<div style="position: absolute; left: 0%; top: 0%"></div>`

Pozycjonowanie absolutne powoduje wstawienie danego elementu w określoną część strony bez względu na obecność innych elementów na stronie. Można w ten sposób uzyskiwać efekty nakładania się elementów na siebie.

Pozycjonowanie względne odnosi się do miejsca wstawiania elementu na stronie. Jest łatwiejsze do kontrolowania i raczej należy używać tego sposobu.

Kursory

Są obsługiwane jedynie przez nowsze przeglądarki. Chodzi tu o kształt wskaźnika myszki po ustawieniu się nad danym elementem. np:

```

```

Po najechaniu myszką na obrazek.gif kursor zmieni kształt na znak zapytania.

Możliwe wartości:

- *crosshair* (krzyżyk)
- *hand* (ręka)
- *wait* (klepsydra)
- *text* (typowy kursor tekstowy I)
- *default* (typowy dla danego elementu)
- *auto* (zgodny z domyślnymi ustawieniami przeglądarki)
- *x-resize* - powoduje zmianę kierunku pokazywania strzałki kursora. Dozwolone są cztery kierunki świata oraz ich kombinacje. *X* może przyjmować następujące wartości: *n,w,e,s* lub kierunki pośrednie, np: *ne*.

Drukowanie stron WWW

Można wymusić na przeglądarkach "przełamywanie stron" w określonych miejscach, gdy strona zaczyna być drukowana. Strony można przełamywać za pomocą jednego z dwóch poleceń:

page-break-before: wartość - przełamanie strony przed danym elementem

page-break-after: wartość - przełamanie strony po danym elemencie

Na chwilę obecną jedyną możliwą wartością jest *always*.

np:

```
<H1 style="page-break-before: always">Rozdział 2</H1>
```

Wygodnym rozwiązaniem jest posłużenie się w tym miejscu klasami:

Definicja klasy (np. w sekcji HEAD):

```
.newpage {page-break-before: always}
```

Użycie klasy w tekście:

```
<h1 class=newpage>Tytuł rozdziału</h1>
```

JavaScript

Java i JavaScript to dwa różne języki. Poza podobną nazwą mają niewiele wspólnego.

Różnice pomiędzy Javą i JavaScriptem:

JavaScript	Java
Interpretowany przez klienta	Wykonywany przez klienta
Tworzy skrypty, które mogą być bezpośrednio wstawiane do stron HTML	Tworzy "aplety", które są wywoływane ze strony HTML, ale nie umieszczane w niej
Oparty na obiektach	Zorientowany obiektowo
Łatwy	Wymaga doświadczenia programistycznego

Sposoby umieszczania skryptów na stronach :

Skrypty umieszcza się bezpośrednio na stronach HTML. Od tagów HTML oddziela się je za pomocą jednej z par znaczników:

- a) `<SCRIPT>` i `</SCRIPT>`.
- b) `<SCRIPT language="JavaScript">` i `</SCRIPT>`.
- c) `<SCRIPT language="JavaScript1.1">` i `</SCRIPT>`.
- d) `<SCRIPT language="JavaScript1.2">` i `</SCRIPT>`.

Ukrywanie skryptu w znakach komentarza:

```
<SCRIPT><!-- treść skryptu --></SCRIPT>
```

```
<NOSCRIPT>Przeglądarka nie obsługuje JavaScript</NOSCRIPT>
```

Komentarze wewnątrz skryptu:

(identycznie do C/C++/Java):

```
/* wielolinijkowy
```

```
komentarz*/          (komentarzy wielolinijkowych nie wolno zagnieżdżać !)
```

```
// jednolinijkowy komentarz
```

Proste przykłady wykorzystania skryptów

1. Wypisanie tekstu:

```
<html>
<head>
</head>
<body>
<br>
tekst
<br>
  <script language="LiveScript">
    document.write("To jest JavaScript")
  </script>
<br>
tekst
</body>
</html>
```

2. Wywołanie funkcji(funkcje należy deklarować w części HEAD):

```
<html>
<head>
  <script language="JavaScript">
    function pushbutton() {
      alert("Czesc!");
    }
  </script>
</head>
<body>
<form>
  <input type="button" name="nazwa" value="Czesc" onclick="pushbutton()">
</form>
</body>
</html>
```

3. Odczytanie zawartości pola formularza:

```
<html>
<head>
<script language="JavaScript">
<!-- Umieszczenie wszystkiego w znakach komentarza html ukrywa tekst
skryptu przed starszymi przeglądarkami.
  function getname(str) {
    alert("Hi, "+ str+"!");
  }
// end hiding contents -->
</script>
</head>
<body>
```

Wpisz swoje imię:

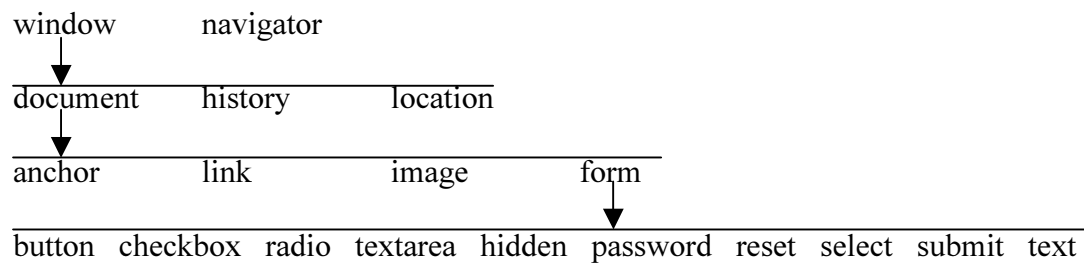
```
<form>
  <input type="text" name="name" onBlur="getname(this.value)" value="">
</form>
</body>
</html>
```

4. Wyświetlenie ostatniej daty aktualizacji dokumentu:

```
<html>
<body>
This is a simple HTML- page.
<br>
Last changes:
  <script language="LiveScript">
    <!-- hide script from old browsers
      document.write(document.lastModified)
    // end hiding contents -->
  </script>

</body>
</html>
```

Hierarchia obiektów w JavaScript



Odwoływanie się do właściwości obiektu:

1) za pomocą operatora kropki (.) - tak jak w C++

2) za pomocą tablicy właściwości obiektu np:

`document[1]` - wykorzystywane przy wykonywaniu tej samej czynności na wielu obiektach w pętli

3) za pomocą tablicy asocjacyjnej

`document["href"]`

Wyrażenia

`x=7;`

`str = "Some string";`

`(bool1 == true) && (bool2 == false);`

`str = "Some" + " string";`

Operatory

Operatory przypisania

Symbol	Składnia	Opis
%=	$x\%=y$	Wykonuje przypisanie $x=x\%y$
=	$x=y$	Wykonuje przypisanie $x=x*y$
/=	$x/=y$	Wykonuje przypisanie $x=x/y$
+=	$x+=y$	Wykonuje przypisanie $x=x+y$
-=	$x-=y$	Wykonuje przypisanie $x=x-y$
=	$x=y$	Przypisuje wartość y do zmiennej x

Operatory bitowe

Symbol	Składnia	Opis
<	$x<y$	Wykonuje bitową operację AND, która wyświetla 1, jeśli obie zmienne wynoszą 1.
^	x^y	Wykonuje bitową operację XOR, która wyświetla 1, jeśli jedna ze zmiennych (ale nie obie jednocześnie) wynosi 1..
	$x y$	Wykonuje bitową operację OR, która wyświetla 1, jeśli jedna ze zmiennych wynosi 1.
<<	$x<<y$	Wykonuje przesunięcie bitów w lewo o podaną liczbę miejsc.
>>	$x>>y$	Wykonuje przesunięcie bitów w prawo o podaną liczbę miejsc.

Operatory porównania

Symbol	Składnia	Opis
!=	$x!=y$	Zwraca true, jeśli zmienne nie są równe.
<	$x<y$	Zwraca true, jeśli pierwsza zmienna jest mniejsza niż druga.
<=	$x<=y$	Zwraca true, jeśli pierwsza zmienna jest mniejsza niż druga lub jej równa.
==	$x==y$	Zwraca true, jeśli zmienne są równe.
>	$x>y$	Zwraca true, jeśli pierwsza zmienna jest większa niż druga.
>=	$x>=y$	Zwraca true, jeśli pierwsza zmienna jest większa niż druga lub jej równa.

Operator dekrementacji

Symbol	Składnia	Opis
--	x--	Zmniejsza wartość zmiennej o 1.

Operator inkrementacji

Symbol	Składnia	Opis
++	x++	Zwiększa wartość zmiennej o 1.

Operatory logiczne

Symbol	Składnia	Opis
!	!x	Neguje wyrażenie.
&&	x&& y	Operator logiczny AND zwraca true, jeśli obie zmienne są prawdziwe (true).
	x y	Operator logiczny OR zwraca true, jeśli co najmniej jedna ze zmiennych jest prawdziwa (true).

Operatory arytmetyczne

Symbol	Składnia	Opis
-	x-y	Wykonuje operację odejmowania.
-	-x	Wykonuje negację zmiennej.
%	x%y	Zwraca resztę z dzielenia pierwszej zmiennej przez drugą (dzielenie modulo).
*	x*y	Wykonuje operację mnożenia.
/	x/y	Wykonuje operację dzielenia.
?:	x?:y	Sprawdza prawdziwość warunku.
+	x+y	Wykonuje operację dodawania.

Operator łańcuchowy (konkatenacji)

Symbol	Składnia	Opis
+	"text1"+"text2"	Łączy dwa ciągi znaków w jeden.

Instrukcje sterujące

- break
- continue
- else
- for
- for...in
- function
- goto
- if...else
- in
- return
- while
- with

break

Instrukcja zatrzymuje bieżącą pętlę *for* lub *while* i przekazuje sterowanie do instrukcji następującej po zatrzymanej pętli.

continue

Instrukcja kończy wykonywanie bloku instrukcji w pętli *for* lub *while* i kontynuuje wykonywanie pętli w następnej iteracji.

for

Instrukcja jest poleceniem pętli. Pętla, którą tworzy, będzie wykonywana określoną liczbę razy.

Składnia:

for([initial expression];)[condition;][increment-expression]) {statements}

np:

```
for(i=1; i<5; i++) {
```

```
// instrukcje
```

```
}
```

for..in

Instrukcja *for...in* iteruje (powtarza) zdefiniowaną zmienną na wszystkich własnościach obiektu.

np:

```
for (i in obj) {  
    // instrukcje  
}
```

goto

Instrukcja przenosi wykonywanie skryptu do innej części skryptu, zdefiniowanej przez etykietę

np:

```
goto label1;
```

if..else

Instrukcja wykonuje zbiór poleceń, jeśli zdefiniowany warunek jest spełniony. Jeśli warunek nie jest spełniony, może zostać wykonany inny zbiór poleceń, określony przez instrukcję *else*.

Składnia:

```
if(condition){statements} [else { statements}]
```

return

Zwraca wartość funkcji.

while

Instrukcja *while* tworzy pętlę, która bada wartość wyrażenia, a jeśli jest ono prawdziwe, wykonuje blok poleceń. Następnie pętla jest powtarzana dotąd, dopóki wyrażenie jest prawdziwe.

Składnia:

```
while(condition) {statements}
```

with

Instrukcja *with* wskazuje domyślny obiekt dla zbioru poleceń. W ramach tego zbioru poleceń wszystkie odwołania do własności, które nie wskazują wyraźnie jakiegoś obiektu, są przyjmowane jako należące do domyślnego obiektu.

Składnia:

```
with(object){statements}
```

np:

```
with (document) {
```

```
write " Wypisanie tekstu na ekran";
```

```
}
```

Zdarzenia

Zdarzenia to akcje, które są podejmowane w przeglądarce internetowej, za pomocą myszki lub klawiatury. Wyróżnia się następujące rodzaje zdarzeń:

Zdarzenie	Opis
blur	Zachodzi, gdy użytkownik przenosi fokus (wyjmuje kursor) z pola tekstowego, klikając w jakimkolwiek innym miejscu formularza lub naciskając klawisz Tab, aby przejść do innego pola formularza.
Składnia: <i>onBlur</i>	Używany z: checkbox, textarea, text
Przykład:	<INPUT TYPE="TEXT" NAME="FIELD" onBlur="DoSthOnBlur">
click	Zachodzi, gdy użytkownik klika na odsyłaczu lub elemencie.
Składnia: <i>onClick</i>	Używany z: button, checkbox, radio, reset, submit, href
Przykład:	<INPUT TYPE="BUTTON" VALUE="Click Here" onClick="DoSthonClick">
change	Zachodzi, gdy użytkownik zmienia wybór, tekst lub elementy obszaru tekstowego w formularzu
Składnia: <i>onChange</i>	Używany z: checkbox, textarea, text
Przykład:	<INPUT TYPE="TEXT" NAME="FIELD" SIZE=20 onChange="DoSthonChange">
focus(select)	Zachodzi, gdy użytkownik wstawia kursor (fokus) do jakiegoś elementu formularza, posługując się klawiszem tabulatora lub myszką
Składnia: <i>onFocus</i>	Używany z: checkbox, textarea, text
Przykład:	<INPUT TYPE="TEXT" NAME="FIELD" SIZE=10 onFocus="DoSthonFocus">

load	Zachodzi, gdy zostaje załadowana strona.
Składnia: onLoad	Używany z: dokumentami
Przykład:	<BODY onLoad="DoSthonLoad">
mouseover	Zachodzi, gdy użytkownik przenosi kursor myszki nad obrazek, odsyłacz czy kotwicę
Składnia: onMouseOver	Używany z: href, img
Przykład:	A HREF="http://www.altavista.digital.com" onMouseOver="DoSthonMouseOver">
submit	Zachodzi, gdy użytkownik wysyła formularz
Składnia: onSubmit	Używany z: form
Przykład:	FormName.onSubmit="DoSthonSubmit"
unload	Zachodzi, gdy użytkownik opuszcza stronę
Składnia: onUnload	Używany z: dokumentami
Przykład:	<BODY onUnload="DoSthonUnload">

Obiekty przeglądarki sieciowej

Większość przeglądarek posiada pewną ilość obiektów, które można wykorzystać do współpracy z nią. Łączna suma tych obiektów stanowi **model obiektów przeglądarki**. Oczywiście różne przeglądarki mają nieco różne modele.

Wszystkie strony WWW będą miały obiekty: **window**, **navigator**, **document**, **history**, **location**. Reszta obiektów zależy od budowy strony.

Obiekt window

Przeglądarka tworzy przynajmniej jeden obiekt window dla każdego dokumentu. Obiekt **window** można traktować jako faktyczne okno, a obiekt **document** jak treść, która pojawia się w tym oknie.

Wybrane własności:

document, history, location

defaultStatus - Domyślna wiadomość wyświetlana w pasku statusu

frames - Tablica zawierająca listę wszystkich ramek w oknie

name - Nazwa okna

parent - Odnosi się do okna, które zawiera układ ramowy

self - Dotyczy bieżącego okna

status - Wyświetla wiadomość w pasku statusu

top - Dotyczy okna położonego najwyżej w hierarchii (głównego)

Wybrane metody:

alert(string) - Tworzy okienko dialogowe z alertem

close() - Zamyka dokument

confirm(string) - Tworzy okienko dialogowe z tekstem *string* i z przyciskami OK i Cancel

open() - Otwiera nowe okno

setTimeout(expression, msec) - Wykonuje *expression* po upływie podanej liczby milisekund. Zwraca *timerId*, który może być użyty przez *clearTimeout()*

clearTimeout(timerId) - Przerywa polecenie *setTimeout*

scroll(x,y) - Przesuwa okno do podanych współrzędnych x,y

Obiekt Location

Obiekt przedstawia bieżący URL wraz ze wszystkimi jego parametrami, takimi jak port, nazwa hosta, ścieżka, argumenty wywołania.

Wybrane własności:

host - zawiera nazwę hosta oraz numer portu np: `www.dmcs.p.lodz.pl:80`

hostname - wyświetla nazwę hosta lub adres IP bieżącego hosta

href - określa cały adres URL

path - zawiera ścieżkę do bieżącego dokumentu

port - określa port komunikacyjny serwera (zwykle 80)

protocol - określa nazwę protokołu dla transmisji danych wraz z pierwszym dwukropkiem (`http:`, `ftp:`, etc.)

search - zawiera wszystkie argumenty wywołania występujące po znaku zapytania (?).

Obiekt history

Obiekt **history** zawiera informację o bieżącej sesji użytkownika przeglądarki. Dane składają się z pewnej liczby odwiedzonych witryn, ich adresów URL i ich nazw.

Wybrane właściwości:

current - zawiera URL bieżącego wpisu do historii

length - zawiera liczbę wpisów na liście historii

previous - zawiera URL poprzedniego wpisu

next - zawiera URL następnego wpisu

Wybrane metody:

back() - cofa się o jeden wpis w historii

forward() - przesuwa się do przodu o jeden wpis w historii

go(num) - przesuwa się o *num* wpisów w historii (*num* może być dodatnie lub ujemne)

go(string) - przechodzi do najnowszego wpisu w historii, którego tytuł lub URL zawiera *string* jako podłańcuch.

Obiekt document

Obiekt ustawia właściwości i metody bieżącego dokumentu.

Wybrane właściwości:

alinkColor - określa kolor aktywnego odsyłacza (active link color)

anchors - tablica odzwierciedlająca wszystkie kotwice dokumentu

bgColor - określa kolor tła

cookie - określa wartości cookies dla danego dokumentu

fgColor - określa kolor tekstu

forms - tablica odzwierciedlająca wszystkie formularze w dokumencie

lastModified - informuje, kiedy dokument był ostatnio modyfikowany

linkColor - określa kolor odsyłaczy

links - tablica odzwierciedlająca wszystkie odsyłacze w dokumencie

location - wyświetla bieżący adres URL dokumentu

referrer - adres URL dokumentu, z którego przywołano bieżący dokument

title - wyświetla zawartość znacznika <TITLE>

vlinkColor - określa kolor odsyłacza do odwiedzanej już wcześniej strony (visited link color)

Wybrane metody:

close() - zamyka bieżący dokument

write(string) - wypisuje zawartość stringu

writeln(string) - wypisuje zawartość stringu a następnie biały znak

Obiekty link, area, anchor

Obiekt **link** powstaje, gdy znacznik <a> korzysta z atrybutu *href*, obiekt **anchor** powstaje, gdy znacznik <a> korzysta z atrybutu *name*. Obiekt **area** powstaje dla każdego znacznika <area>.

Obiekty link i area można przywoływać za pomocą tablicy obiektu link. Posiadają one te same cechy co obiekt location oraz obsługują zdarzenia: *onMouseOver*, *onMouseOut*, *OnClick*.

Obiekt form

Obiekt form służy do uzyskiwania od użytkownika różnych rodzajów informacji. Obsługuje następujące zdarzenia: onSubmit(), onReset().

Wybrane właściwości:

name - wartość atrybutu name

method - wartość atrybutu method

action - wartość atrybutu action

encoding - wartość atrybutu encoding

target - okno docelowe po wysłaniu odpowiedzi

elements - tablica elementów obiektu form

length - liczba elementów obiektu form

Wybrane metody:

reset() - czyści formularz do początkowych ustawień

submit() - przesyła formularz

Obiekty typu input

Są to obiekty typu input=text, input=radio itp. Obsługiwane przez nie zdarzenia to: onFocus, onBlur, onSelect, onSubmit, onChange.

Wybrane metody:

focus() - uaktywnienie dla danego obiektu

blur() - dezaktywowanie danego obiektu

submit() - przesyła formularz

Wybrane właściwości:

name - wartość atrybutu name

value - wartość atrybutu value

defaultValue - początkowa zawartość pola (dla pól typu tekstowego)

checked - stan pola wyboru (dla pól typu checked i radio)

defaultChecked - początkowy stan pola wyboru (dla pól typu checked i radio)

Obiekt select

Wybrane właściwości:

name - wartość atrybutu name

options - tablica z elementami listy wyboru

length - długość powyższej tablicy

selectedIndex - indeks aktualnie wybranej opcji

Obiekt image

Każdemu znacznikowi odpowiada jedna instancja obiektu image. Obsługuje następujące dodatkowe zdarzenia: *onError* (gdy wystąpi błąd w trakcie ładowania obrazu), *onAbort* (uruchamiane, gdy użytkownik zatrzyma ładowanie obrazu), *onLoad*

Wybrane właściwości:

border - wartość atrybutu border

height - wartość atrybutu height

name - wartość atrybutu name

src - wartość atrybutu src

width - wartość atrybutu width

complete - wskazuje, czy obrazek został całkowicie załadowany

Przykład:

```
<html>
<head>
<script language="JavaScript">
<!--
  function changeImage(i,j) {
    document.images[i].src = "obrazek"+j+".gif";
  }
-->
</script>
</head>
<body>

  <a href = "doc.htm" onMouseOver="changeImage(0,2)"
onMouseOut="changeImage(0,1)"> </a>

<hr>

  <a href = "doc.htm" onMouseOver="changeImage(1,2)"
onMouseOut="changeImage(1,1)"> </a>

</body>

</html>
```

Wbudowane obiekty JavaScript

Obiekty te nie są częścią modelu obiektów przeglądarki a stanowią część języka JavaScript.

Obiekt Date

Składnia:

1. `variable = new Date()` //data bieżąca
2. `variable = new Date(hours, minutes, seconds)`
3. `variable = new Date(year, month, day)`

Wybrane metody:

getDate() - Wyświetla dzień miesiąca dla ustalonej daty (liczba całkowita z zakresu 1-31)

getDay() - Wyświetla dzień tygodnia dla ustalonej daty (liczba całkowita, od 0=Niedziela do 6=Sobota)

getHours() - Wyświetla godzinę dla ustalonej daty (liczba całkowita z zakresu 0-23)

getMinutes() - Wyświetla minuty dla ustalonej daty (liczba całkowita z zakresu 0-59)

getMonth() - Wyświetla miesiąc dla ustalonej daty (liczba całkowita z zakresu 0=Styczeń - 11=Grudzień)

getSeconds() - Wyświetla sekundy dla bieżącego czasu (liczba całkowita z zakresu 0-59)

getTime() - Pokazuje ustaloną datę z użyciem liczb (liczba milisekund od 1 stycznia 1970 00:00:00)

getTimezoneOffset() - Wyświetla w minutach różnicę czasu między czasem lokalnym i czasem Greenwich (GMT)

getFullYear() - Wyświetla rok dla ustalonej daty (liczba dwucyfrowa)

setDate() - Ustawia dzień miesiąca dla aktualnego obiektu *date*

setHours() - Ustawia godzinę dla aktualnego obiektu *date*

setMinutes() - Ustawia minuty dla aktualnego obiektu *date*

setMonth() - Ustawia miesiąc dla aktualnego obiektu *date*

setSeconds() - Ustawia liczbę sekund dla aktualnego obiektu *date*

setTime() - Ustawia datę i godzinę dla aktualnego obiektu *date*, w milisekundach od 1/1/70 00:00:00

setYear() - Ustawia rok dla aktualnego obiektu *date* (rok jest liczbą całkowitą większą od 1900)

Obiekt Math

Obiekt umożliwia dostęp do funkcji i stałych matematycznych.

Składnia: Math.property lub Math.method, gdzie property lub method jest jednym z podanych niżej elementów.

Własności:

E - Zwraca stałą Eulera, która wynosi ok. 2.718

LN2 - Zwraca logarytm dwóch, tj. ok. 0.693

LN10 - Zwraca logarytm z dziesięciu, tj. ok. 2.302

LOG2E - Zwraca logarytm o podstawie 2 z liczby E, czyli ok. 1.442

LOG10E - Zwraca logarytm o podstawie 10 z E, czyli ok. 0.434

PI - Zwraca wartość liczby Pi, czyli ok. 3.14159

SQRT1_2 - Zwraca pierwiastek kwadratowy z 0.5, czyli ok. 0.707

SQRT2 - Zwraca pierwiastek kwadratowy z 2, czyli ok. 1.414

Metody:

abs(liczba) - Zwraca wartość absolutną liczby

acos(liczba) - Zwraca arcus cosinus z liczby (podanej w radianach)

asin(liczba) - Zwraca arcus sinus z liczby (podanej w radianach)

atan(liczba) - Zwraca arcus tangens z liczby (podanej w radianach)

ceil(liczba) - Zwraca najmniejszą liczbę całkowitą, większą lub równą podanej liczbie

cos(liczba) - Zwraca cosinus liczby (podanej w radianach)

exp(liczba) - Zwraca wartość E podniesionej do potęgi wyrażonej podanym argumentem

floor(liczba) - Zwraca największą liczbę całkowitą mniejszą lub równą podanej liczbie

log(liczba) - Zwraca logarytm naturalny liczby

max(liczba1,liczba2) - Zwraca większą z dwóch liczb

min(liczba1,liczba2) - Zwraca mniejszą z dwóch liczb

pow(liczba1,liczba2) - Zwraca wartość liczby1 podniesionej do potęgi liczby2

random() - Zwraca wartość pseudolosową z przedziału 0-1

round(liczba) - Zwraca zaokrąglenie danej liczby do najbliższej liczby całkowitej

sin(liczba) - Zwraca sinus liczby (podanej w radianach)

sqrt(liczba) - Zwraca pierwiastek kwadratowy liczby

tan(liczba) - Zwraca tangens liczby (podanej w radianach)

Tworzenie nowych okien za pomocą `window.open()`

Nowe okna przeglądarki otwiera się z kilku powodów:

- okienka logowania
- zmuszenie użytkownika do przeczytania dodatkowych uwag/wyjaśnień
- przedstawienie treści równoległej (np. wyszukiwarki sieciowe)

Przykład:

```
<script language="JavaScript">
<!--
myWindow = window.open("window.htm", "MyWindow", "toolbar=no, location=no");
-->
```

Parametry metody `open`

1. Nazwa pliku html umieszczanego w nowym oknie
2. Nazwa okna (można jej użyć jako atrybutu `target` znacznika `<a>`)
3. Konfiguracja opcji dla tworzonego okna porozdzielana przecinkami. Możliwe ustawienia:

toolbar = [yes/no] - ustawia wyświetlenie paska narzędzi

location = [yes/no] - ustawia wyświetlenie paska lokalizacji

directories = [yes/no] - ustawia wyświetlenie paska katalogu

status = [yes/no] - ustawia wyświetlenie paska statusu

menubar = [yes/no] - ustawia wyświetlenie paska menu

scrollbars = [yes/no] - ustawia wyświetlenie pasków przewijania

resizable = [yes/no] - ustawia czy można zmieniać rozmiar okna

width = [l.pikseli] - ustawia szerokość okna w pikselach

height = [l.pikseli] - ustawia wysokość okna w pikselach

Komunikacja pomiędzy oknami:

Okno macierzyste - okno potomne : *self.nazwa_okna_potomnego*

Okno potomne - okno macierzyste : *opener*, np: *opener.document.Myform.Mytext.value = "tekst"*

Standardowe okna pomocnicze w JavaScript

1. *alert("text")* - okno z tekstem i przyciskiem OK.
2. *confirm("text")* - okno z tekstem i przyciskami OK i Cancel. Zwraca true lub false.
3. *prompt("text", "default_input")* - okno z tekstem i polem do wprowadzenia pewnych informacji przez użytkownika. Default_input to tekst domyślny, który pojawi się w polu do wypełnienia. Zwraca wprowadzoną przez użytkownika wartość.

np:

```
<script>

<!--

imie = window.prompt("Jak masz na imię?" , "Jasiu");

-->

</script>
```

Komunikacja pomiędzy ramkami

self lub *window* - ramka bieżąca

parent - obiekt zawierający bieżącą ramkę

top - najwyższy obiekt w hierarchii zawierający bieżącą ramkę

np:

parent.MyFrame2 - ramka leżąca na tym samym poziomie zagnieżdżenia o nazwie MyFrame2

Działania na stringach

Jako string rozumie się ciąg znaków umieszczony pomiędzy znacznikami ' ' lub " ".

Składnia: string.property lub string.method, gdzie string jest nazwą ciągu znaków, a property lub method pochodzą z poniższej listy.

Własności: length

Metody:

big() - Nadaje atrybut big

blink() - Nadaje atrybut blink

bold() - Nadaje atrybut bold

charAt(indeks) - pobiera znak z pewnej pozycji stringu

fontcolor(kolor) - Określa kolor stringu

fontsize(wielkość) - Określa wielkość czcionki w stringu

indexOf(łańcuch, indeks) - Szuka pierwszego wystąpienia poszukiwanej wartości w stringu

italics() - Nadaje atrybut italics

lastIndexOf(łańcuch, indeks) - Szuka ostatniego wystąpienia poszukiwanej wartości w stringu

small() - Nadaje atrybut small

strike() - Nadaje atrybut strike through (linia przekreślająca ciąg)

sub() - Nadaje atrybut subscript

substring(indeksPoczątku, indeksKońca) - Zwraca łańcuch rozpoczynający się od pewnego znaku i kończący na innym znaku

sup() - Nadaje atrybut superscript

toLowerCase() - Nadaje atrybut lower case

toUpperCase() - Nadaje atrybut upper case

Predefiniowane funkcje w JavaScript

JavaScript definiuje kilka przydatnych funkcji:

escape()

Funkcja zwraca heksadecymalny zapis łańcucha znaków w zmiennej kodowany w ISO-8859-1.

Składnia: `escape("Text")`

isNaN()

Funkcja określa, czy wartość przekazana do niej jest liczbą, czy też nie.

Składnia: `isNaN(value)`

parseFloat()

Funkcja konwertuje łańcuch znaków na liczbę zmiennoprzecinkową.

Składnia: `parseFloat("14.78")`

parseInt()

Funkcja konwertuje łańcuch znaków na liczbę całkowitą.

Składnia: `parseInt("string" [rad])`

unescape()

Ta funkcja, używana niekiedy w skryptach z hasłem lub szyfrujących, dekoduje łańcuch znaków zakodowany za pomocą *escape*.

Składnia: `unescape("TextToUnescape")`

Przykłady programów w JavaScript

a) zegarek pojawiający się na belce statusu

```
<script Language="JavaScript">
<!--
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 = "" + ((hours > 12) ? hours - 12 : hours)
    timeValue += ((minutes < 10) ? ":0" : ":") + minutes
    timeValue += ((seconds < 10) ? ":0" : ":") + seconds
    timeValue += (hours >= 12) ? " P.M." : " A.M."
    window.status = timeValue;
    timerID = setTimeout("showtime()",1000);
    timerRunning = true;
}
function startclock () {
    stopclock();
    showtime();
}
-->
</script>
```

BODY

```
<BODY ONLOAD="startclock()">
```

b) sygnalizowanie nowości na stronie (powinno być robione po stronie serwera)

```
<SCRIPT LANGUAGE="JavaScript">
  <!--
  function newItem(expireDate)
  {
    exp = new Date(expireDate);
    cur = new Date();
    if (cur.getTime() < exp.getTime())
      document.write("<IMG SRC='new.gif' WIDTH=31 HEIGHT=12
BORDER=0 ALT='new'>" );
  }
  -->
</SCRIPT>
```

W ramach BODY

```
<SCRIPT LANGUAGE="JavaScript">
  <!--
  newItem("9/30/2000");
  // -->
</SCRIPT>
```

c) wysłanie adresu strony znajomemu(powinno być robione po stronie serwera)

```
<A HREF="javascript:location.href='mailto:?SUBJECT=' + document.title + '&BODY=' +
escape(location.href)">Wyślij ten adres znajomemu</A>
```

d) nawigacja przeglądarki

```
<form name="Historia">
  <input type="button" value="&lt;&lt;" onClick="history.back()">
  <input type="button" value="Strona główna" onClick="location='jakas_strona.html'">
  <input type="button" value="&gt;&gt;" onClick="history.forward()">
</form>
```

e) rozpoznawanie rodzaju przeglądarki

```
<SCRIPT LANGUAGE="JavaScript">
  <!--
  function checkBrowser () {
    userAg = navigator.userAgent;
    bName = navigator.appName;
    bVer = parseInt(navigator.appVersion);
    if (bVer >= 4) {
      if (bName.substring(0,9) == "Microsoft")
        // Redirect IE 4.0
        window.location.href="ie4.htm";
      else
        if (bName.substring(0,8) == "Netscape")
          // Redirect Netscape 4.0
          window.location.href="nn4.htm";
        else
          window.location.href="inne.htm";
    } else {
      if (bVer == 3) {
        if (bName.substring(0,8) == "Netscape")
          // Redirect Netscape 3.0
          window.location.href="nn3.htm";
        else
          // Redirect all other browsers
          window.location.href="inne.htm";
      }
    }
  }
  // -->
</SCRIPT>
<NOSCRIPT>
<META HTTP-EQUIV="Refresh" CONTENT='1; URL=inne.htm'>
</NOSCRIPT>
```

f) arkusz styli zależny od rodzaju przeglądarki

```
<script language="JavaScript">
  <!--
    if(-1 != navigator.userAgent.indexOf("MSIE"))
    {
      // Internet Explorer
      document.write('<link rel="stylesheet" type="text/css" href="ie.css">');
    }
    else if (-1 != navigator.userAgent.indexOf("Mozilla"))
    {
      // Netscape
      document.write('<link rel="stylesheet" type="text/css" href="nc.css">');
    }
    else
    {
      // other
      document.write('<link rel="stylesheet" type="text/css" href="inny.css">');
    }
  //-->
</script>
```

g) uniwersalne tło dźwiękowe

```
<SCRIPT language="JavaScript">
  if (navigator.appName == "Netscape") {
    document.write('<EMBED SRC="../multimed/jazzball.mid" AUTOSTART=TRUE
LOOP=FALSE WIDTH=145 HEIGHT=55></EMBED>') }

  else { if (navigator.appName == "Microsoft Internet Explorer")
    document.write('<BGSOUND SRC="../multimed/jazzball.mid" LOOP="1">')
  }
  //-->
</SCRIPT>
```

Metody śledzenia zachowania użytkownika

Przykłady potrzeb identyfikacji użytkownika:

- sklep internetowy - poruszanie się pomiędzy stronami
- indywidualne strony domowe na serwerach Netscape
- zmienne nagłówki, informacje, pytania do użytkownika

Sposoby identyfikacji użytkownika:

- kodowanie informacji w adresie URL
- kodowanie informacji w ukrytych zmiennych formularza
- zachowywanie na serwerze
- przechowywanie na komputerze klienta w postaci *cookie*

Mechanizm cookie (persistent client state HTTP cookie)

Cookies – pozwalają zachowywać informacje na komputerze przeglądarki użytkownika do ponownego użytku. Zapisywane są one na dysku lokalnym użytkownika w postaci jawnej zwykle w postaci pliku tekstowego. Umożliwiają zapamiętywanie danych w konwencji *zmienna = wartość* przez określony czas (tzw. czas życia cookie). Nazwa mechanizmu jak i sam mechanizm został stworzony przez firmę Netscape.

W JavaScript cookie jest właściwością obiektu przeglądarki document . Jej parametry to-
name = value-expires = date-path = path-domain = domainname-secure = secure

Przykładowa funkcja do obsługi cookie:

```
function setCookie(name, value) { document.cookie = name + "=" + value;}
```

```
function getCookie(name) {  
    var result = null;  
    var myCookie = " " + document.cookie + ";";  
    var searchName = " " + name + "=";  
    var startCookie = myCookie.indexOf(searchName);  
    var endCookie;  
    if(startCookie != -1 ) {  
        startCookie += searchName.length;  
        endCookie = myCookie.indexOf(";", startCookie);  
        result = myCookie.substring(startCookie, endCookie);  
    }  
    return result;  
}
```

Ograniczenia:

- dla każdego serwera lub domeny można stworzyć tylko 20 cookies
- łączna liczba cookies dla przeglądarki musi być mniejsza niż 300
- cookie może mieć maksymalnie rozmiar 4KB (łącznie z parametrami name i value)
- całkowity rozmiar pliku z cookie nie może przekraczać więcej niż 1,2 Mb
- cookie są widoczne jedynie dla serwera który je utworzył

Przy przekroczeniu tych limitów przeglądarka będzie próbowała usunąć starsze cookies, przy czym najpierw usunie cookies, których dawno nie używano.

Wady mechanizmu cookies

- przechowywanie informacji w niezaszyfrowanym pliku
- przeglądarki mogą nie akceptować lub akceptować warunkowo cookie

Informacje o cookie:

http://www.netscape.com/newsref/std/cookie_spec.html

<http://www.cookiecentral.com>

SSI(server-side include)

Cechy SSI:

Umożliwia analizowanie stron HTML po stronie serwera i interpretację poleceń SSI.

Wymaga konfiguracji serwera, przy czym w zależności od serwera i systemu operacyjnego robi się to inaczej.

Nie istnieje oficjalny standard SSI, więc nie ma on jednolitej składni na wszystkich serwerach WWW.

Ogólna składnia poleceń SSI:

`<#!#polecenie nazwa_znacznika="parametr"-->`

Przykłady przydatnych poleceń SSI:

echo

np:

Dzisiejsza data to: `<#!#echo var="DATE_LOCAL"-->`

Twój adres to: `<#!#echo var="REMOTE_ADDR"-->`

Ta strona to: `<#!#echo var="DOCUMENT_NAME"-->`

include

np:

`<#!# include file="email.htm"-->`

Inne: `exec`(wykonanie skryptu CGI), `fsize`(rozmiar pliku) , `flastmod`(ostatnia data modyfikacji określonego pliku)

SSI nie jest używane często, zostało wyparte przez technologie takie PHP i Java. Można go używać jedynie przy nieskomplikowanych stronach WWW.

CGI (Common Gateway Interface)

- Jest to prawdopodobnie najstarsza technologia służąca do tworzenia aplikacji WWW.
- Programy CGI mogą być pisane w każdym języku, lecz najczęściej wykorzystywany jest Perl lub C.
- Typowym zastosowaniem CGI jest pobranie informacji ze strony WWW, przetworzenie danych, zwrócenie wyników i przekazanie ich użytkownikowi.
- Dzięki programom CGI możliwe jest także pobieranie informacji z bazy danych i przekazywanie ich za pośrednictwem serwera do przeglądarki.
- Typowym miejscem dla skryptów CGI jest katalog /cgi-bin znajdujący się w katalogu zawierającym pliki serwera.
- Zostało wyparte przez technologie takie jak ASP, PHP, Java.

Ogólny przegląd języka PHP

Skrypty PHP są one włączane bezpośrednio do dokumentu HTML. Dzięki temu część informacji jest osadzona na stałe, a część powstaje dynamicznie za sprawą odpowiedniego skryptu. Standardowo kod PHP jest oddzielany od kodu HTML za pomocą następujących znaczników (choć możliwe są także inne sposoby):

- rozpoczynającego skrypt PHP : `<? lub <?php`
- kończącego skrypt PHP : `?>`

Odbiorca otrzymuje jedynie rezultaty wykonanego wcześniej skryptu bez jego postaci źródłowej. Jest to o tyle ważne, że w kodzie mogą być zapisane takie informacje jak hasła do bazy danych. Oczywiście pisząc skrypty zawierające tego typu informacje należy zadbać, aby nie były one dostępne z poziomu powłoki systemu. Najprostszy skrypt PHP ma następującą postać :

```
<HTML>
  <BODY>
    <?php
      echo " To jest skrypt PHP ";
    ?>
  </BODY>
</HTML>
```

Działanie jego polega jedynie na wypisaniu tekstu „To jest skrypt PHP” i właśnie ten tekst ukaże się w przeglądarce. Otwierając w niej kod źródłowy tego dokumentu, można będzie zobaczyć:

```
<HTML><BODY>
To jest skrypt PHP
</BODY></HTML>
```

Parser PHP przetworzył instrukcję *echo* będącą poleceniem wypisania tekstu i do przeglądarki dotarł gotowy kod HTML.

Elementy języka PHP

a) Komentarze

```
// To jest komentarz 1 zajmujący jedną linię  
# To jest komentarz 2 zajmujący jedną linię  
/* To jest komentarz 3  
zajmujący więcej niż jedną linię */
```

b) **Wyrażenia i instrukcje** – instrukcje są kolejnymi krokami programu, zapisanymi w postaci kodu. Każda instrukcja składa się z wyrażenia zakończonego średnikiem. Wyrażenie może składać się z wywołań funkcji, zmiennych, operatorów i słów kluczowych. Przykład instrukcji:

```
$suma = $a + $b;
```

c) **Operatory** – priorytety operatorów są zaczerpnięte z języka C.

d) Instrukcje sterujące

W PHP występują trzy rodzaje pętli: *for*, *while*, oraz *do...while* a ich działanie i budowa jest identyczna do tych występujących w języku C. Można także korzystać z instrukcji *break* powodującej wyjście z pętli i instrukcji *continue* powodującej powrót do jej początku.

Istnieją dwie instrukcje warunkowe: *if...else* i *switch*. Ich budowa jak i działanie jest identyczne do języka C.

Zarówno w przypadku pętli jak i instrukcji warunkowych dopuszcza się użycie niektórych składni zaczerpniętych z języków skryptowych powłok.

e) **Zmienne** – są to obiekty zajmujące przestrzeń w pamięci, które mają swoją nazwę i typ, służące do przechowywania informacji. Nazwy zmiennych w PHP są zawsze poprzedzone znakiem dolara (\$) i nie wymagają deklaracji typu. Język PHP przeprowadza automatyczną konwersję typów, jednak czasami warto zastosować zarówno jawną deklarację jak i

konwersję typów. Rozróżnia się zmienne następujących typów: liczby całkowite, liczby zmiennopozycyjne, łańcuchy, tablice, obiekty.

f) **Funkcje** – typowa funkcja to fragment kodu przyjmujący pewne argumenty, wykonujący pewną akcję i zwracający pewną wartość. Oczywiście funkcja nie musi przyjmować argumentów jak i nie musi zwracać wartości. W PHP jest zdefiniowany bogaty zestaw funkcji wbudowanych, a także można definiować własne. Funkcje tworzy się posługując słowem kluczowym `function`. Przykład definicji i wywołania funkcji:

```
<HTML>
```

```
    <BODY>
```

```
<?php
```

```
    function suma($a,$b)
```

```
    {
```

```
        $suma=$a+$b;
```

```
        return $suma;
```

```
    }
```

```
    $jeden=1;
```

```
    $dwa=dwa;
```

```
    $trzy=suma($jeden,$dwa);
```

```
?>
```

```
    </BODY>
```

```
</HTML>
```

Interfejs API ODBC w języku PHP.

Funkcja	Opis działania
<i>odbc_close(integer connection)</i>	Zamyka bieżące połączenie z bazą danych.
<i>integer odbc_connect(string dsn, string user, string password)</i>	Nawiązuje połączenia ze źródłem danych ODBC. Jako wynik zwraca identyfikator połączenia.
<i>integer odbc_do(integer connection, string query)</i>	Wykonuje zapytanie na podanym połączeniu. Jako wynik zwraca identyfikator zbioru wyników.
<i>boolean odbc_fetch_row(integer result, integer row)</i>	Pobiera wskazany wiersz ze zbioru wyników. Argument <i>row</i> jest opcjonalny i jeśli nie zostanie podany, funkcja pobierze następny wiersz ze zbioru wyników. Jeżeli w zbiorze wyników nie zostały już żadne wiersze do pobrania, to funkcja zwróci <i>false</i> .
<i>string odbc_result(integer result, string field)</i>	Zwraca wartość wskazanego pola aktualnego wiersza zbioru wyników. Do pola można odwoływać się podając jego nazwę lub indeks liczbowy.

Przykład pobrania informacji z bazy danych:

```
(1)  <?php
(2)      $sDB = "pracownicy";      //Selected database
(3)      $sUser = "nazwa_użytkownika";      //Username
(4)      $sPassword = "password";      //User password
(5)      $con=odbc_connect("$sDB","$sUser","$sPassword");
(6)      $query_osoby="SELECT tytuł_naukowy,imie,nazwisko FROM osoby";
(7)      if(!$result_osoby=odbc_do($con,$query_osoby)) print("Błąd wykonania
        zapytania");
(8)      while($row_osoby=odbc_fetch_row($result_osoby))
(9)      {
(10)          $Tytuł=odbc_result($result_osoby,'tytuł_naukowy');
(11)          $Imie=odbc_result($result_osoby,'imie');
(12)          $Nazwisko=odbc_result($result_osoby,'nazwisko');
(13)  ?>
(14)      <a href="<?php echo "index.php";?>?strona=prac2&id=<?php
        echo"$ID_osoby";?>&ID_jezyka=1" target=_top>
        <?php echo"$Tytuł $Imie $Nazwisko";?> </a>
(15)  <?php
(16)      }
(17)  ?>
```

Opis programu:

- (1) – Znacznik rozpoczynający kod PHP.
- (2) – (4)– Deklaracja zmiennych zawierających informacje służące do połączenia z bazą.
- (5) – Utworzenie połączenia z bazą (zmienna \$con zawiera identyfikator połączenia).
- (6) - Umieszczenie pod zmienną \$query_osoby tekstu zapytania SQL.
- (7) – Wykonanie zapytania do bazy danych (zbiór wyników znajduje się w zmiennej \$result_osoby).
- (8) – Pobieranie poszczególnych wierszy ze zbioru wyników (dopóki nie zostaną pobrane wszystkie).
- (10) – (12) – Umieszczenie poszczególnych części zbioru wyników pod odpowiednimi zmiennymi.
- (13) - Znacznik zamykający kod PHP.
- (14) – Utworzenie hiperłącza do strony prywatnej danej osoby.