



TECHNIKI

PROGRAMOWANIA

Wykład 3:

Języki programowania – (X)HTML

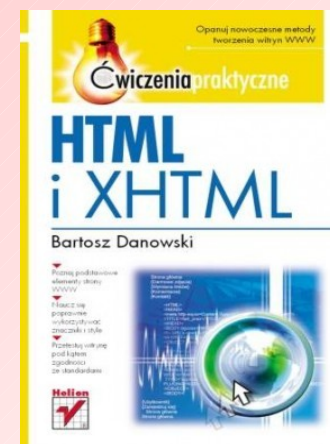
HTML - bibliografia



HTML i XHTML krok po kroku
Faithe Wempen
ISBN: 83-7243-527-8



HTML, XHTML i CSS. Biblia
Bill Karow, Bryan Pfaffenberger,
Chuck White, Steven M. Schafer
ISBN: 83-7361-725-6



HTML i XHTML. Ćwiczenia praktyczne
Bartosz Danowski
ISBN: 83-246-0568-1

Czym jest HTML?



HTML – dominujący język wykorzystywany do tworzenia stron internetowych.



Dokument (X)HTML, czyli (**Extensible**) **Hypertext Markup Language** - (**Rozszerzalny**) **Hipertekstowy** **Język Oznaczenia**, jest plikiem tekstowym, gdzie wpisuje się wszystkie polecenia, dotyczące formatowania tekstu, wstawiania grafiki i inne.



Pozwala opisać strukturę informacji zawartych w dokumencie nadając znaczenie poszczególnym fragmentom tekstu (formując linki, nagłówki, akapity, listy, itp.) oraz osadzić w tekście dodatkowe obiekty np. statyczne grafiki, interaktywne formularze, dynamiczne animacje.

Czym jest HTML?



Ważną cechą języka, która wyraźnie przyczyniła się do rozpowszechnienia sieci WWW jest niezależność od systemu operacyjnego oraz parametrów sprzętowych komputera, na którym strony te będą oglądane.

HTML a XHTML

- ◎ Pierwsza oficjalna odsłona języka HTML 4.0 odbyła się 18 grudnia 1997, a ostatnia poprawka do wersji 4.01 została wprowadzona 24 grudnia 1999.
- ◎ Język XHTML, który jest aktualnym standardem w tworzeniu serwisów WWW, opiera się na specyfikacji HTML 4.01, tzn. definiuje dokładnie te same znaczniki, atrybuty i sposoby budowania dokumentów.
- ◎ Jednak wprowadza przy tym pewne obostrzenia, które pozwalają dostosować składnię do języka XML, który od jakiegoś czasu wyznacza trendy rozwoju aplikacji przenośnych.

Dokument HTML



Dokument HTML ma postać pliku tekstowego, gdzie tekst przeplatany jest komendami (*ang. tags*) języka.



Plik źródłowy jest interpretowany przez "przeglądarkę" (*ang. browser*) która prezentuje dokument w jego ostatecznym kształcie.



Najpopularniejsze "przeglądarki" *MS Internet Explorer*, *Opera* i wiele wyszukiwarek opartych na rdzeniu zwanym Mozilla (są to zarówno popularny *Netscape* jak i mniej znane *Galeon* czy *Phoenix*), nie działają identycznie, a dokument powinien we wszystkich wyglądać tak samo.

Dokument HTML

- Do tworzenia dokumentów HTML można stosować każdy edytor plików tekstowych.
- W klasie programów pomagających w tworzeniu dokumentów można wyróżnić takie, które działają na poziomie kodu oraz takie, które działają na poziomie dokumentu, generując kod automatycznie.
 - Stosowanie pierwszych jest żmudniejsze, lecz daje pełną kontrolę nad dokumentem (oczywiście przy znajomości HTML).
 - Stosowanie drugich jest szybsze i nie wymaga znajomości komend HTML ale wprowadzanie poprawek może być utrudnione, zaś niektóre efekty mogą być nie do uzyskania, jeśli wykraczają poza możliwości programu.

Narzędzia (X)HTML

Pajaczek (shareware - płatny)

Najbardziej znany polski edytor HTML oferujący szerokie wsparcie dla najnowszych technologii internetowych, z których warto wymienić takie jak HTML, XHTML, XML, PHP, JavaScript, VBScript, Perl, MathML, SVG, SMIL, P3P, ASP i inne.

Wsparcie to dotyczy nie tylko ułatwionej edycji plików źródłowych w tych językach - program oferuje bowiem kontekstową dokumentację dla wybranego polecenia języka, przeglądanie za pomocą serwera czy funkcję dynamicznego kodu, która uwalnia od konieczności pamiętania poleceń języka.

Narzędzia (X)HTML

CoreEditor (płatny - dostępna wersja DEMO)

Tańszy konkurent Pajęczka, następca darmowego EdHTML przepisany zupełnie od nowa. Bardzo dobrze spisuje się jako edytor dla osób programujących w Pascalu, Asemblerze, Delphi, C, Java itd. Niemniej największy nacisk położony został na wsparcie dla takich technologii jak xHTML, CSS, PHP, JavaScript i MySQL.

Ager Web Edytor (darmowy)

Oprócz HTML i CSS, wspiera również programowanie w językach JavaScript i PHP. Posiada generatory do automatycznego podłączenia pluginów takich jak: Media Pleyer, QuickTime, Flash, RealMedia, a także kreatory znaczników HTML.

Narzędzia (X)HTML

EdHTML (darmowy)

Uniwersalny edytor do HTMLa, wspierający również XHTML, CSS, JavaScript i PHP. Bardzo dobrze koloruje składnię, pozwala na proste testowanie stron we wbudowanych przeglądarkach. Program zawiera wiele wbudowanych narzędzi, np. kompresor HTML, konwerter polskich znaków itp.

Edytor Znaczników HTML - ezHTML (darmowy)

Zapewnia dostęp do wszystkich możliwości języka HTML. Można także skorzystać z listy znaczników zawierającej wszystkie tagi standardu HTML 4.0.

Narzędzia (X)HTML

HateML Pro (darmowy)

Edytor przeznaczony zarówno dla początkujących jak i zaawansowanych webmasterów. Wspomaga tworzenie stron z użyciem XHTML, PHP, CSS, SQL (m.in. MySQL, Oracle, MSSQL7, MSSQL2000, Interbase6), JavaScript i VBScript. Posiada wbudowany inspektor tagów, parser HTML i PHP oraz prosty edytor WYSIWYG.

Program został również wyposażony w narzędzia mające na celu optymalizację kodu - m.in. narzędzie HTMLTidy służące do sprawdzania poprawności kodu, narzędzie kompresujące kod, oraz możliwość bezpośredniego wysłania kodu do narzędzia Validator W3C.

Narzędzia (X)HTML

kED (darmowy)

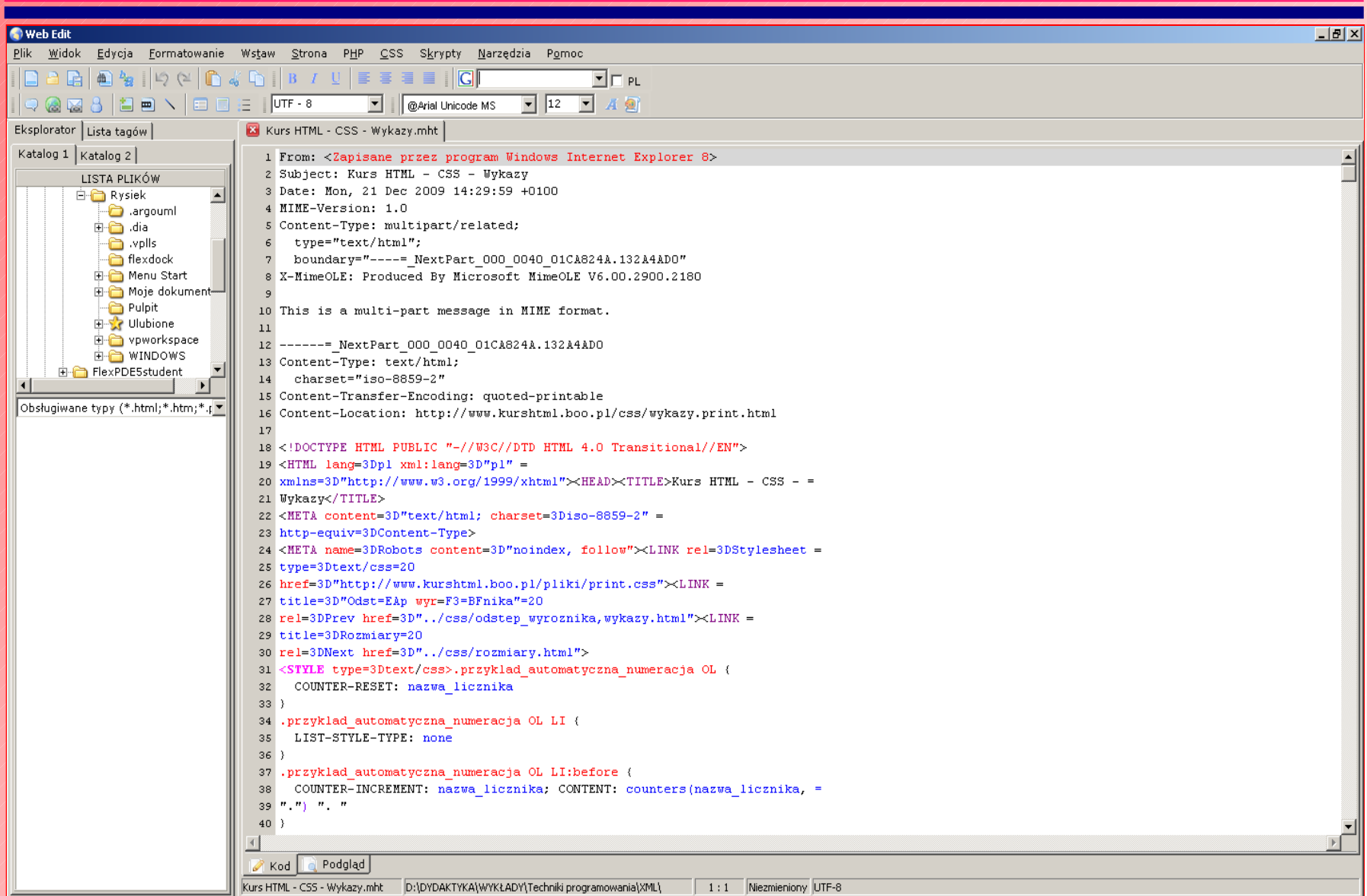
Edytor plików HTML, CSS, XML, PHP, JavaScript. Edycja wielu dokumentów jednocześnie, rozbudowana edycja tabel (w tym oczyszczanie tabel zapisanych przez MS-Word). Możliwość wyboru dodatkowych zewnętrznych przeglądarek.

Web Edit (darmowy)

Wsparcie m.in. języków: HTML i XHTML, CSS, PHP. Zawiera generatory: tabel, formularzy, list, bloków tekstowych, wideo, Flash, obrazów.

Inteligentna funkcja podpowiedzi kodu ułatwia wstawianie atrybutów bez potrzeby ich pamiętania.

Narzędzia (X)HTML



The screenshot shows the 'Web Edit' application window. The menu bar includes 'Plik', 'Widok', 'Edycja', 'Formatowanie', 'Wstaw', 'Strona', 'PHP', 'CSS', 'Skrypty', 'Narzędzia', and 'Pomoc'. The toolbar contains various icons for file operations and editing. The left pane shows a file explorer with a tree view of folders and files, including 'Rysiek', '.argouml', '.dia', '.vpls', 'flexdock', 'Menu Start', 'Moje dokument', 'Pulpit', 'Ulubione', 'vpworkspace', 'WINDOWS', and 'FlexPDESstudent'. The main editor area displays the source code of an email, which is a multi-part message in MIME format. The code includes headers for 'From', 'Subject', 'Date', 'MIME-Version', 'Content-Type', 'Content-Location', and 'Content-Transfer-Encoding'. It also contains HTML markup for a document type declaration, language settings, and a list of links. The list is styled with automatic numbering and includes a 'before' style for the list items. The status bar at the bottom shows the file name 'Kurs HTML - CSS - Wykazy.mht', the path 'D:\DYDAKTYKA\WYKLADY\Techniki programowania\XML', and the encoding 'UTF-8'.

```
1 From: <Zapisane przez program Windows Internet Explorer 8>
2 Subject: Kurs HTML - CSS - Wykazy
3 Date: Mon, 21 Dec 2009 14:29:59 +0100
4 MIME-Version: 1.0
5 Content-Type: multipart/related;
6   type="text/html";
7   boundary="-----_NextPart_000_0040_01CA824A.132A4AD0"
8 X-MimeOLE: Produced By Microsoft MimeOLE V6.00.2900.2180
9
10 This is a multi-part message in MIME format.
11
12 -----_NextPart_000_0040_01CA824A.132A4AD0
13 Content-Type: text/html;
14   charset="iso-8859-2"
15 Content-Transfer-Encoding: quoted-printable
16 Content-Location: http://www.kurshtml.boo.pl/css/wykazy.print.html
17
18 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
19 <HTML lang=3Dpl xml:lang=3D"pl" =
20   xmlns=3D"http://www.w3.org/1999/xhtml"><HEAD><TITLE>Kurs HTML - CSS - =
21   Wykazy</TITLE>
22 <META content=3D"text/html; charset=3Diso-8859-2" =
23   http-equiv=3DContent-Type>
24 <META name=3DRobots content=3D"noindex, follow"><LINK rel=3Dstylesheet =
25   type=3Dtext/css=20
26   href=3D"http://www.kurshtml.boo.pl/pliki/print.css"><LINK =
27   title=3D"Odst=EAp wyr=F3=BFnika"=20
28   rel=3DPrev href=3D"./css/odstep_wyroznika,wykazy.html"><LINK =
29   title=3DRozmiary=20
30   rel=3DNext href=3D"./css/rozmiary.html">
31 <STYLE type=3Dtext/css>.przyklad_automatyczna_numeracja OL {
32   COUNTER-RESET: nazwa_licznika
33 }
34 .przyklad_automatyczna_numeracja OL LI {
35   LIST-STYLE-TYPE: none
36 }
37 .przyklad_automatyczna_numeracja OL LI:before {
38   COUNTER-INCREMENT: nazwa_licznika; CONTENT: counters(nazwa_licznika, =
39   ".") ". "
40 }
```

Struktura dokumentu (strony) HTML

□ Typowy dokument HTML zdefiniowany jest poprzez komendy podane poniżej:

Prolog

<HTML>

- deklaracja dokumentu

- początek kodu HTML

<HEAD>

- początek nagłówka

treść nagłówka

</HEAD>

- koniec nagłówka

<BODY>

- początek części właściwej

zawartość części właściwej

</BODY>

- koniec części właściwej

</HTML>

- koniec kodu HTML

Struktura dokumentu (strony) HTML

Prolog

- Prolog to jednolinijkowa deklaracja typu dokumentu, określająca m.in. wersję używanego języka HTML.
- Wersje różnią się od siebie zestawem używanych komend i ich zakresem.
- Obecnie najpopularniejsze prologi to:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Frameset//EN">
```

Oznaczają one dokumenty zakodowane przy pomocy wersji 4.0 języka HTML. W pierwszym przypadku to wersja najprostsza (domyślna), która nie dopuszcza "niestandardowych" elementów języka. Wersja "transitional" pozwala na rozszerzenie języka o dodatkowe elementy zaś "frameset" jest przeznaczona dla obsługi stron z ramkami.

Struktura dokumentu (strony) HTML

<HTML>

Komenda **<HTML>** występuje bezpośrednio po prologu i sygnalizuje początek dokumentu zakodowanego przy pomocy języka HTML, zaś **</HTML>** oznacza koniec.

<HEAD>

Pomiędzy początkiem komendy **<HEAD>** i jej końcem **</HEAD>** znajduje się nagłówek dokumentu zawierający wiele ważnych informacji, które zwykle nie są wyświetlane ale niejednokrotnie wpływają na interpretację całości lub części dokumentu.

<BODY>

Pomiędzy początkiem komendy **<BODY>** i jej końcem **</BODY>** znajduje się cały dokument interpretowany przez przeglądarkę - atrybuty tej komendy definiują kolor tła, kolor tekstu i elementów hipertekstowych.

Struktura dokumentu (strony) HTML - przykład

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<HTML>
  <HEAD>
    .....
    <TITLE> Jan Kowalski - moja strona </TITLE>
    .....
  </HEAD>
  <BODY>
    .....
    Nazywam się Jan Kowalski. Na razie to wszystko. Witajcie!
    .....
  </BODY>
</HTML>
```

Formatowanie tekstu – strona kodowa

- Dużą popularność w Internecie zdobył sobie standard **ISO 8859-2**, gdyż był konsekwentnie implementowany w systemach unixowych.
- Z chwilą "wejścia" do Internetu komputerów klasy PC, drugim sposobem kodowania najczęściej stosowanym dla języka polskiego jest strona kodowa Windows czyli **CP 1250**.
- Deklarację kodu umieszczamy w nagłówku, czyli pomiędzy komendą `<HEAD>` i jej końcem `</HEAD>`, zwykle przed tytułem `<TITLE>`.

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;  
charset=windows-1250">
```

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

Formatowanie tekstu – akapit

- Aby zaznaczyć przejście do nowej linii należy zastosować komendę **
**.
- Główną komendą formatującą akapit jest **<P>**.
- Jeśli całość tekstu ujęto pomiędzy **<P>** i **</P>** to można stosować atrybut **ALIGN** oznaczający:
 - **ALIGN=left** - równanie do lewej
 - **ALIGN=center** - centrowanie
 - **ALIGN=right** - równanie do prawej

<P>Ten tekst domyślnie wyrówna się do lewej</P>

<P ALIGN=center> Ten będzie na środku</P>

<P ALIGN=right> Ten zaś wyrówna się do prawej</P>

Formatowanie tekstu – atrybuty czcionki

□ Czcionce można nadać atrybuty bezwzględne takie jak:

- **pogrubienie** (** .. **),
- *pochylenie* (**<I> .. </I>**),
- podkreślenie (**<U> .. </U>**),
- indeksu górnego (**^{...}**),
- indeksu dolnego (**_{...}**),
- migotania (**<BLINK> ... </BLINK>**).

□ Chcąc zachować oryginalny format fragmentu tekstu w postaci pliku tekstowego (czyli z zachowaniem wszystkie odległości i przejścia do nowej linii), należy użyć komendy (**<PRE> ... </PRE>**).

Formatowanie tekstu – wielkość czcionki

□ Wielkość czcionki można regulować poprzez definiowanie nagłówków dokumentu i jego części lub bezpośrednią zmianę wielkości fontu.

- Definicja atrybutu nagłówka ma postać:

<Hx> ... **</Hx>** - gdzie x jest liczbą od 1 do 4.

```
<H1> To jest nagłówek nr 1 </H1>
```

```
<H2> To jest nr 2 </H2>
```

```
<H3> a to nr 3 </H3>
```

- Zmianę wielkości czcionki w tekście uzyskuje się

p *Tekst dwukrotnie powiększony*, *przechodzimy na pojedyncze*, *teraz jest fragment pisany normalnym rozmiarem,* *a to już o jeden pomniejszony*.

Formatowanie tekstu – rysowanie linii

- Do rysowania linii służy komenda **<HR>**
- Komenda ta ma też kilka atrybutów pozwalających tworzyć proste ornamenty:
 - **SIZE=liczba** - wyznacza grubość linii,
 - **WIDTH=liczba/procent** - wyznacza jej szerokość (domyślnie, całe okno czyli 100%),
 - **ALIGN=left/right/center** - wyrównanie do lewej/centrowanie/do prawej,
 - **NOSHADE** - usunięcie "cienia"

```
<HR SIZE=10>
```

```
<HR WIDTH=70%><HR WIDTH=60%><HR WIDTH=50%>
```

```
<HR WIDTH=40% ALIGN=left NOSHADE>
```

```
<HR SIZE=5 WIDTH=40% ALIGN=right>
```

Listy i wyliczenia – lista nienumerowana

- Lista nienumerowana jest przykładem najprostszej listy, którą definiuje się w następujący sposób:

**** - początek listy
**** - pierwsza pozycja listy
**** - druga pozycja listy
**** - trzecia pozycja listy
**** - koniec listy

- Listy mogą być wielostopniowo zagnieżdżane.
- Do oznaczenia pozycji listy można wykorzystać jeden z trzech typów punktatorów:

- **TYPE=disc**
- **TYPE=circle**
- **TYPE=square**

```
<UL TYPE=square>  
  <LI>pierwsza pozycja listy  
  <LI>druga pozycja listy  
</UL>
```

Listy i wyliczenia – lista numerowana

- Pozycje listy mogą być automatycznie numerowane.
- Zaletą użycia numerowanej listy jest możliwość uzupełniania jej w dowolnym punkcie bez konieczności przenumerowania pozycji.

```
<OL>
```

```
<LI>pierwsza pozycja listy
```

```
<OL TYPE=a>
```

```
<LI>pozycja 1 podlisty pierwszej
```

```
<LI>pozycja 2 podlisty pierwszej
```

```
<LI>pozycja 3 podlisty pierwszej
```

```
</OL>
```

```
<LI>druga pozycja listy
```

```
<LI>trzecia pozycja listy
```

```
</OL>
```



Listy i wyliczenia – lista numerowana

- Gdy numerowanie nie zaczyna się od 1, można zadać liczbę (literę) startową poleceniem **START**.

```
<OL START=5>
```

```
  <LI>pierwsza pozycja listy
```

```
  <LI>druga pozycja listy
```

```
  <LI>trzecia pozycja listy
```

```
</OL>
```

Listy i wyliczenia – lista definicyjna

□ Nazwa tej listy bierze się stąd, że każda pozycja ma drugi element struktury pozwalający zamieścić opis pozycji, tak jak opis hasła w encyklopedii.

```
<DL>
```

```
<DT> termin pierwszy
```

```
<DD> to jest opis do pierwszego terminu
```

```
<DT> termin drugi
```

```
<DD> to opis drugiego terminu
```

```
</DL>
```

Rezultat wygląda następująco:

termin pierwszy

to jest opis do pierwszego terminu

termin drugi

to opis drugiego terminu

Osadzanie grafiki w tekście

Ilustracją



niej Ozdobnikiem



ka HTML jest

stu.

niej przygotowany

, który pozwala na

wa formaty

(Osadzany obraz zapamiętywany jest w odrębnym pliku

Linia rozdzielającą



Niestandardowym napisem

To jest niestandardowy napis

Osadzanie grafiki – polecenie

- Składnia polecenia jest następująca:

```
<IMG SRC="adres pliku" atrybuty>
```

Adres pliku może być względny lub bezwzględny.

- adres względny może wyglądać następująco:

```
SRC="graf.jpg"
```

jeśli plik *graf.jpg* jest w tym samym folderze co dokument

```
SRC="obrazki/graf.jpg"
```

jeśli plik *graf.jpg* jest w folderze podrzędnym (o nazwie *obrazki*) w stosunku do foldera dokumentu

```
SRC="../graf.jpg"
```

jeśli plik *graf.jpg* jest w folderze nadrzędnym

-adres bezwzględny ma postać adresu internetowego

```
SRC="http://ebib.oss.wroc.pl/images/xfile.gif"
```

Osadzanie grafiki – atrybuty polecenia

□ Aby móc wkomponować obrazek w jak najlepszy sposób mamy do dyspozycji szereg atrybutów:

- **ALIGN**
- **ALT**
- **WIDTH / HEIGHT**
- **BORDER**
- **VSPACE / HSPACE**

ALIGN służy ustaleniu położenia tekstu wobec obrazka.

ALIGN=top wyświetla tekst od szczytu obrazka.

ALIGN=middle wyświetla tekst od środka wysokości obrazka.

ALIGN=bottom wyświetla tekst od dołu obrazka.

ALIGN=left wyświetla obrazek po lewej stronie tekstu.

ALIGN=right wyświetla obrazek po prawej stronie tekstu.

Osadzanie grafiki – atrybuty polecenia ****

ALT

wyświetla opis obrazka gdy jest on niewidoczny lub gdy jest wskazywany przez kursor

```
<IMG SRC="graf.jpg" ALT="To jest opis">
```

WIDTH / HEIGHT

pozwała zdefiniować szerokość i wysokość obrazu inną niż oryginalna

```
<IMG SRC="graf.jpg" ALIGN=middle WIDTH=80 HEIGHT=60>
```

BORDER otacza obrazek ramką o podanej grubości

```
<IMG SRC="graf.jpg" ALIGN=middle BORDER=10>
```

VSPACE / HSPACE

określa wolne miejsce (margines) wokół obrazka.

```
<IMG SRC="graf.jpg" ALIGN=left VSPACE=10 HSPACE=10>
```

Osadzanie grafiki w tle strony

- ❑ Obrazki mogą występować również w roli tła strony, dzięki atrybutowi polecenia `<BODY>`.
- ❑ Obrazek zadeklarowany jako tło, jest powtarzany tyle razy ile potrzeba do wypełnienia całego okna przeglądarki.

```
<BODY BACKGROUND="adres pliku">
```

Połączenia (linki) – hierarchia stron

- ❑ Łączenie stron poprzez "linki" jest podstawową cechą języka HTML (litera H oznacza hipertekst czyli dynamiczny system stron).
- ❑ Planując system stron zwykle zaczyna się od strony głównej, która nazywa się najczęściej **index.htm** lub **index.html** (ma to związek z konfiguracją serwerów HTTP, które przyjmują te nazwy jako nazwy domyślne głównych dokumentów systemu stron).
- ❑ Połączenie można stworzyć do dowolnej strony w Internecie o znanym adresie (bezwzględny).
- ❑ W przypadku stron znajdujących się na tym samym komputerze adres może być względny, tak samo jak przy adresie pliku z obrazkiem.

Połączenia (linki) – hierarchia stron

- Składnia linku ma następującą postać:

```
<A HREF="adres pliku ze stroną">aktywny napis linku</A>
```

- Zwykły link odsyła do początku strony – lecz można odesłać również do dowolnego punktu strony, pod warunkiem, że zostanie tam umieszczony punkt orientacyjny o następującej składni:

```
<A NAME="dowolna nazwa">napis, od którego ma zostać wyświetlona strona</A>
```

- Linkowanie nie musi mieć charakteru hierarchicznego, a strony mogą odsyłać do siebie wielokrotnie, na różnych poziomach.

Połączenia (linki) – hierarchia stron

Strona główna (index.htm)

Na tej stronie są trzy linki:

- [do strony podrzędnej 1](#)
- [do strony podrzędnej 2](#)
- [do drugiej połowy strony podrzędnej 2](#)

Odpowiednie kody linków mogą wyglądać następująco:

```
<A HREF="pod1.htm">do strony podrzędnej 1</A>
```

```
<A HREF="pod2.htm">do strony podrzędnej 2</A>
```

```
<A HREF="pod2.htm#połowa">do drugiej połowy strony podrzędnej 2</A>
```

Strona podrzędna 1 (pod1.htm)

Na tej stronie mogą być różne linki ale ostatnim powinien być powrót do strony nadrzędnej, który może wyglądać tak:

```
<= Powrót do strony głównej
```

zaś jego kod następująco:

```
<A HREF="index.htm"> <= Powrót do strony głównej</A>
```

Strona podrzędna 2 (pod2.htm)

Założmy, że w połowie strony zaczyna się rozdział 2 oznaczony napisem:

ROZDZIAŁ II

Aby stał się on punktem orientacyjnym kod powinien wyglądać następująco:

```
<A NAME="połowa">ROZDZIAŁ II</A>
```

Połączenia (linki) – wygląd połączeń

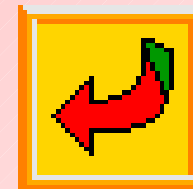
- Link może wyglądać różnie - najprostszą postacią jest oczywiście tekst, ale może być to również obrazek lub fragment formularza.

Przykład 1: <<< Powrót do spisu treści

```
<<< <A href="index.html#lekcja5">Powrót do spisu treści</A>
```

Przykład 2:

```
<A href="index.html#lekcja5"><IMG  
SRC="obrazki/strzalki/yhome.gif"> </A>
```



Przykład 3:

```
<FORM><INPUT TYPE="button"  
VALUE=" Powrót " onClick="parent.location.href  
='index.html#lekcja5'"></FORM>
```

Powrót

Połączenia (linki) – linki do innych protokołów

- Najczęściej używamy linków do innych stron HTML czyli łączymy się poprzez protokół HTTP. Inne protokoły obsługiwane przez przeglądarki to GOPHER i FTP.
- Wpisanie w miejsce protokołu MAILTO odsyła do adresu poczty elektronicznej i uruchamia wbudowanego lub zewnętrznego klienta poczty.
- Najważniejsze protokoły wyglądają więc następująco:

```
<A HREF="http://adres.strony.www">
```

```
<A HREF="ftp://adres.servera.ftp">
```

```
<A HREF="gopher://adres.gophera">
```

```
<A HREF="mailto:adres@poczty.elektronicznej">
```

```
<A HREF="telnet://adres.odległego.komputera">
```

Definiowanie koloru

- Kolory mogą pojawiać w poleceniu `<BODY>`, `` oraz jako atrybuty tabel.
- Kolory możemy deklarować poprzez podanie ich szesnastkowego kodu RGB lub nazwy.
- Kod RGB (**R**ed, **G**reen, **B**lue) ma postać:

`#rrggbb`

gdzie:

#	jest symbolem kodu szesnastkowego (operującego "cyframi" od 0 do F)
rr	dwoma cyframi koloru czerwonego
gg	dwoma cyframi koloru zielonego
bb	dwoma cyframi koloru niebieskiego

Definiowanie koloru

☐ Łatwiejsze do zapamiętania są oczywiście nazwy umowne, szczególnie jeśli znamy język angielski.

Ich stosowanie należy jednak zawsze przetestować w kilku najpopularniejszych przeglądarkach (IE, Netscape, Mozilla, Opera).

[Lista kolorów](#)

☐ Jeśli *nk* oznacza numer lub nazwę koloru to globalna definicja kolorów na stronie wygląda następująco:

```
<BODY BGCOLOR=nk TEXT=nk LINK=nk VLINK=nk ALINK=nk>
```

gdzie:

BGCOLOR	- oznacza kolor tła (domyślnie szary lub biały)
TEXT	- kolor tekstu (domyślnie czarny)
LINK	- kolor połączenia (domyślnie ciemno-niebieski)
VLINK	- kolor używanego połączenia (domyślnie fioletowy)
ALINK	- kolor linku nad którym zatrzymał się kursor.

Definiowanie koloru

- Lokalnie, możemy deklarować kolory poprzez atrybut fontu:

```
<FONT COLOR=nk>tutaj znajduje się kolorowany tekst</FONT>
```

Przykład:

```
<CENTER><FONT SIZE="+2">  
<FONT COLOR=red>U</FONT>  
<FONT COLOR=blue>W</FONT>  
<FONT COLOR=yellow>A</FONT>  
<FONT COLOR=green>G</FONT>  
<FONT COLOR=magenta>A</FONT>  
<FONT COLOR=gray>!</FONT>  
</FONT></CENTER>
```

Efekt:

U W A G A !

Tabele

- ❑ Tabela jest użyteczną strukturą, która pozwala nie tylko tworzyć zestawienia ale również budować ogólny szkielet strony, lub dokładnie kontrolować położenie różnych elementów względem siebie i całości strony.
- ❑ Tabele definiuje się w trzech krokach:
 - ① najpierw definiuje się ramy tabeli (**TABLE**)
 - ② potem definiuje się wiersz (**TR**)
 - ③ zaś w wierszu poszczególne komórki (**TD** lub **TH**)
- ❑ Struktura wiersza wraz z zawartymi w niej komórkami powtarzana jest tyle razy ile jest wierszy w tabeli, zaś powtarzające się komórki tworzą kolumny.
- ❑ Każdemu elementowi tabeli można nadać odpowiednie atrybuty.

⊙ BORDER=*grubość*

określa grubość ramki (tylko obrys); BORDER bez podania liczby powoduje rysowanie cienkiej ramki; pominięcie tego parametru powoduje, że linie ramki są niewidoczne.

⊙ CELSPACING=*ilość "spacji" pomiędzy komórkami*

określa odległość pomiędzy komórkami tabelki i, co za tym idzie, grubość wewnętrznych linii.

⊙ CELLPADDING=*ilość "spacji" pomiędzy brzegiem komórki a jej zawartością (domyślnie 1)*

określa odległość pomiędzy brzegiem komórki a jej zawartością.

- ⊙ **WIDTH**=*szerokość w pikselach lub procentach*
pozwała narzucić konkretną szerokość tabeli, podając albo bezwzględną ilość pikseli albo ilość procentów odnoszącą się do szerokości okna przeglądarki.
- ⊙ **HEIGHT**=*wysokość w pikselach lub procentach*
tak samo jak WIDTH tylko dotyczy wysokości.
- ⊙ **ALIGN**=*left/center/right* (położenie tabeli)
umieszcawia tabelę z lewej, pośrodku lub po prawej stronie okna przeglądarki.
- ⊙ **VALIGN**=*top/middle/bottom*
atrybut umiejscawia tabelę w pionie.
- ⊙ **BGCOLOR**=*nk*
określa nazwę lub numer (*nk*) koloru tła tabeli.

- ⊙ **WIDTH**=*szerokość w pikselach lub procentach*
pozwała narzucić konkretną szerokość tabeli, podając albo bezwzględną ilość pikseli albo ilość procentów odnoszącą się do szerokości okna przeglądarki.
- ⊙ **HEIGHT**=*wysokość w pikselach lub procentach*
tak samo jak WIDTH tylko dotyczy wysokości.
- ⊙ **ALIGN**=*left/center/right* (położenie tabeli)
umieszcawia tabelę z lewej, pośrodku lub po prawej stronie okna przeglądarki.
- ⊙ **VALIGN**=*top/middle/bottom*
atrybut umieszcawia tabelę w pionie.
- ⊙ **BGCOLOR**=*nk*
określa nazwę lub numer (*nk*) koloru tła tabeli.

Atrybuty wiersza <TR>...</TR>

- ⊙ **ALIGN**=*left/center/right*

wyrównuje tekst w wierszu do lewej, prawej lub centruje.

- ⊙ **VALIGN**=*top/middle/bottom*

wyrównuje tekst w pionie.

- ⊙ **BGCOLOR**=*nk*

kolor tła w wierszu.

- ⊙ **BACKGROUND**=*"adres do pliku"*

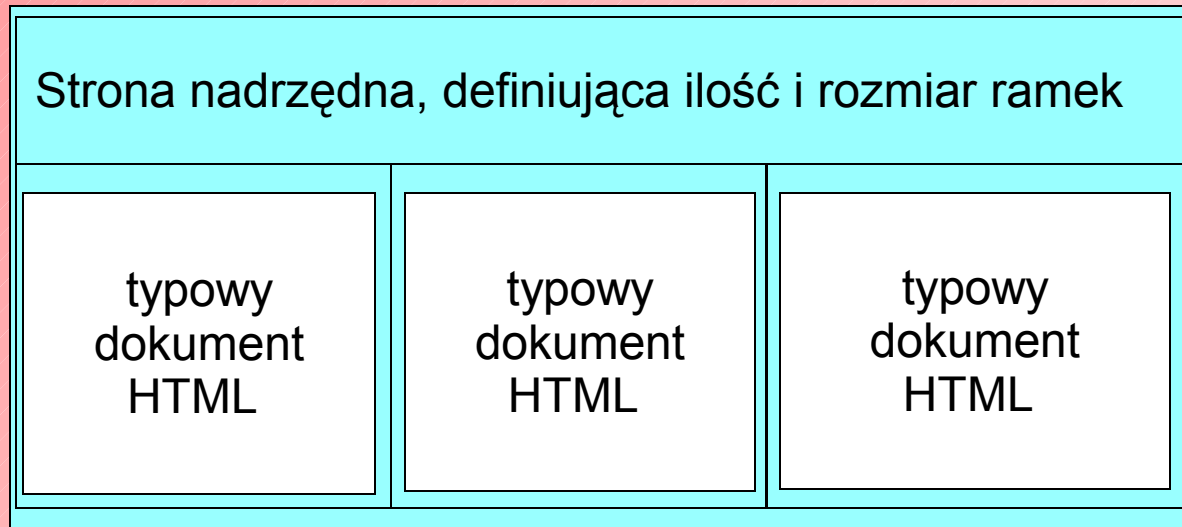
obrazek w tle w danym wierszu.

Atrybuty kolumny tabeli <TD>...</TD>

- ⊙ WIDTH=*szerokość w pikselach lub procentach*
- ⊙ HEIGHT=*wysokość w pikselach lub procentach*
- ⊙ ALIGN=*left/center/right*
- ⊙ VALIGN=*top/middle/bottom*
- ⊙ BGCOLOR=*nk*
- ⊙ BACKGROUND="*adres do pliku*"
analogiczne do atrybutów wiersza lub tabeli.
- ⊙ NOWRAP
zapobiega zawijaniu wierszy w komórce.
- ⊙ COLSPAN=*n*
pozwala połączyć sąsiednie komórki w poziomie.
- ⊙ ROWSPAN=*n*
pozwala łączyć wiersze, czyli komórki w pionie.

Ramki

- ❑ Ramki pozwalają na wyświetlanie kilku dokumentów na jednej stronie.
- ❑ Okno przeglądarki może zostać podzielone na dowolną ilość mniejszych okien w dowolny sposób.



Ramki – składnia strony z ramkami

- ❑ Strona nadrzędna ramki różni się od "zwykłej" strony HTML deklaracją **FRAMESET** w miejscu BODY.
- ❑ Ogólne ramy mają więc poniższą postać.

```
<HTML>  
<HEAD>  
</HEAD>  
<FRAMESET>  
    definicja ramek  
</FRAMESET>  
</HTML>
```


Ramki – składnia strony z ramkami

□ Deklaracja **FRAMESET**, oznaczająca zestaw ramek ma dwa atrybuty: **ROWS** i **COLS**.

⊙ **ROWS**="*lista wierszy - podział w poziomie*,"
lista wierszy może być zdefiniowana w liczbach bezwzględnych (pikselach) lub procentowo.

Przykładowo:

<FRAMESET ROWS="20%,60%,20%">

oznacza podział okna przeglądarki na trzy poziome części, zajmujące odpowiednio: 20, 60 i 20 procent wysokości okna.

<FRAMESET ROWS="100,*">

oznacza podział okna przeglądarki na dwie pionowe części o szerokości: pierwsze okno 100 pikseli, drugie okno - reszta (oznaczona "*").

Ramki – składnia strony z ramkami

□ Deklaracja **FRAMESET**, oznaczająca zestaw ramek ma dwa atrybuty: **ROWS** i **COLS**.

⊙ **COLS="lista kolumn - podział w pionie"**

tak samo jak w przypadku wierszy liczbę i wielkość kolumn podaje się w liczbach pikseli lub procentach.

Przykładowo:

```
<FRAMESET COLS="40%,60%">
```

oznacza podział okna przeglądarki na dwie kolumny zajmujące odpowiednio 40 i 60% szerokości okna.

```
<FRAMESET COLS="40,* ,40">
```

oznacza podział okna przeglądarki na trzy kolumny, z których dwie skrajne zajmują po 40 pikseli, zaś środkowa resztę szerokości okna.

Ramki – deklaracja okna strony podrzędnej

- Po zadeklarowaniu liczby kolumn i wierszy następują deklaracje każdego okna wraz z podaniem wszystkich jego atrybutów.
- Służy do tego deklaracja **FRAME**, w której można zadeklarować następujące atrybuty:
 - ⊙ **SRC="adres do pliku"**
wskazuje na plik wyświetlany w danym oknie.
 - ⊙ **NAME="nazwa okna"**
służy do identyfikacji danego okna.
 - ⊙ **MARGINWIDTH="wartość"**
dodaje marginesy w ramce wyrażone w pikselach, po prawej i lewej stronie.
 - ⊙ **MARGINHEIGHT="wartość"**
jak powyżej, tylko marginesy dolny i górny.

Ramki – deklaracja okna strony podrzędnej

⊙ SCROLLING="yes|no|auto"

oznacza użycie suwaków w oknie - jeśli mają być to atrybut przyjmuje wartość "yes", jeśli nie to "no", "auto" powoduje wyświetlenie suwaków tylko gdy dokument nie mieści się w całości w wyznaczonym oknie.

⊙ NORESIZE

służy do zastrzeżenia stałej wielkości okna.

⊙ FRAMEBORDER="yes|no"

pozwała wyświetlać lub nie granice pomiędzy ramkami. Domyślnie granice są wyświetlane.

⊙ FRAMESPACING=„wartość"

określa ilość wolnego miejsca dookoła ramki (w pikselach).

Ramki – przykład

Przypuśćmy że chcemy uzyskać poniższy układ okien:

jeden	dwa
	trzy

wtedy dokument z ramkami będzie zdefiniowany następująco:

```
HTML>
  <FRAMESET COLS="50%,50%">
    <FRAME SRC="f-doka.html" NAME="jeden">
    <FRAMESET ROWS="50%,50%">
      <FRAME SRC="f-dokb.html" NAME="dwa">
      <FRAME SRC="f-dokd.html" NAME="trzy">
    </FRAMESET>
  </FRAMESET>
</HTML>
```

Metadane

☐ **Metadane** są to dane o dokumencie zawarte w samym dokumencie. Nie są one wyświetlane ani interpretowane przez przeglądarkę, są natomiast odczytywane przez roboty katalogujące Internet - im bardziej starannie sformułujemy metadane tym lepiej dokument zostanie zaindeksowany w takich serwisach internetowych.

☐ Metadane występują w nagłówku dokumentu czyli w sekcji pomiędzy poleceniem **<HEAD>** i **</HEAD>**.

☐ *Deklaracja każdego elementu metadanych składa się z pary "nazwa-zawartość":*

```
<META HTTP-EQUIV="nazwa elementu" CONTENT="zawartość">
```

lub

```
<META NAME="nazwa elementu" CONTENT="zawartość">
```


Metadane

- ⊙ HTTP-EQUIV może określać:
 - Creation-date (*data powstania dokumentu*)
 - Expires (*data ważności dokumentu*)
 - Reply-to (*kontakt email z autorem*)

- ⊙ NAME może określać:
 - author (*dane autora*)
 - keywords (*słowa kluczowe*)
 - Description (*opis*)
 - ROBOTS (*polecenia dla robotów indeksujących*)
 - ALL - wszystkie strony
 - INDEX - indeksuj
 - NOFOLLOW - bez indeksowania linków
 - NOINDEX - nie indeksuj

Metadane - przykład

Przykładowa definicja nagłówka tego dokumentu mogłaby wyglądać następująco:

```
HEAD>
```

```
<META HTTP-EQUIV="Content-Type" CONTENT="text/html;  
    charset=windows-1250">
```

```
<META HTTP-EQUIV="Creation-date" CONTENT="20.12.98">
```

```
<META HTTP-EQUIV="Reply-to"
```

```
    CONTENT="arw@win.oss.wroc.pl">
```

```
<META NAME="author" CONTENT="Aleksander Radwański">
```

```
<META NAME="keywords" CONTENT="HTML, manual in Polish">
```

```
<META NAME="ROBOTS" CONTENT="INDEX, NOFOLLOW">
```

```
<TITLE>Sposób definiowania metadanych</TITLE>
```

```
</HEAD>
```

Monitorowanie

□ **Monitorowanie** wykorzystania stron polega na notowaniu następujących parametrów:

- ile razy była odwiedzana?
- kto i kiedy ją odwiedzał?
- przy użyciu jakiej przeglądarki?

□ Informacje te są zawarte w logu, automatycznie tworzonym przez serwer WWW. Problem w tym, że zapis dotyczy wszystkich stron umieszczonych na serwerze, a dostęp do niego ma tylko administrator.

□ Dlatego też popularne są różne niezależne narzędzia, takie jak:

- liczniki
- książki gości
- indywidualne logi


Dynamizowanie strony - skrypty

- ❑ Jednym ze sposobów uaktywnienia strony jest stosowanie skryptów CGI.
- ❑ Pod tą nazwą kryją się różne języki skryptowe: **Perl**, **PHP**, **ASP**, **JavaScript**, **Embperl**.
- ❑ Języki skryptowe mogą działać na różnych poziomach i ich stosowanie zależne jest najczęściej od konfiguracji serwera WWW.
- ❑ Języki skryptowe są pełnowartościowym językami programowania zaawansowanych aplikacji, ale można też wykorzystać je również do szybkiego tworzenia niewielkich skryptów, pozwalających na dynamiczne wyświetlanie stron WWW, lub sterowanie różnymi procesami poprzez przeglądarkę (np. do przetwarzania wprowadzonych danych).

Dynamizowanie strony – języki skryptowe

 **Perl** - to jeden z najstarszych i najbardziej rozwiniętych języków skryptowych.

- Standardowo w interpreter języka Perl wyposażone są wszystkie serwery Unixowe, ale istnieją też odmiany interpretera dla systemów Windows i NetWare a nawet DOS!
- Obszerne archiwum gotowych skryptów - CPAN - znajduje się w sieci pod adresem: <http://www.cpan.org/>

 **ASP** - to technologia firmy Microsoft stosowana na serwerach Windowsowych.

- Z tego powodu jej zastosowanie jest ograniczone jest tylko do serwerów tego typu.
- Podobnie jak Perl, skrypty ASP są umieszczone w odrębnych plikach i muszą być umieszczane w odpowiednim miejscu przez administratora.

Dynamizowanie strony – języki skryptowe

 **PHP** - jest stosunkowo nowym i popularnym językiem skryptowym.

- Jego stosowanie nie wymaga szczególnych uprawnień, jeśli wcześniej obsługa PHP zostanie skonfigurowana na serwerze WWW (większość serwerów posiada taką możliwość, ale nie zawsze jest ona aktywna).
- Kod wpisywany jest bezpośrednio do pliku dokumentu, który zwykle ma rozszerzenie "php" zamiast "html".
- Język PHP jest specjalnie dostosowany do współpracy z serwerem WWW, dzięki czemu programy w PHP są proste choć równie skuteczne jak "perlowe".
- W sieci istnieje wiele archiwów gotowych skryptów.

Dynamizowanie strony – języki skryptowe

 **JavaScript** - to język skryptowy obsługiwany nie przez serwer tylko przez przeglądarkę.

- Należy odróżnić go języka Java, który jest bardziej rozbudowanym i wszechstronnym językiem programowania używanym nie tylko w połączeniu z serwisem WWW.
- Różnica pomiędzy Java a JavaScript polega na tym, że programów Javy (nazywanych apletami) nie włącza się do testu dokumentu HTML a zagnieżdża w nim na zasadzie oddzielnego okna o podanej wielkości.

 **Embperl** - to najnowszy pomysł w dziedzinie języków skryptowych.

- Embperl oparty jest na Perlu, ale skrypt (podobnie jak w przypadku PHP i JavaScript) wpisany jest bezpośrednio w kod dokumentu. Stosowanie tego języka wymaga jednak uruchomienia specjalnego modułu na serwerze WWW.

Dynamizowanie strony – skrypty - przykłady

jv_pop3.html	- wyskakujące okno z tekstem
jv_bgcl4.html	- zmiana koloru tła przyciskami
jv_clrpk.html	- ustalanie koloru tła w kodzie szesnastkowym (RGB)
jv_lnktp.html	- napis w polu zależnie od wskazanego linku
jv_pop2.html	- okno otwierane i zamykane jednym klawiszem
jv_pswr3.html	- wchodzenie do strony przez "niby-hasło"
jv_scrit.html	- skrolowanie strony w dół
jv_scri2.html	- skrolowanie strony w prawo
jv_sort.html	- sortująca się tabela
jv_bbnr.html	- odbijający się napis (gorzej wygląda w IE)
jv_conv.html	- konwersja miar odległości, objętości i wagi
jv_detec.html	- wykrywanie przeglądarki
jv_fade.html	- cieniowany, kolorowy tekst
jv_hibye.html	- komunikat na przywitanie i pożegnanie
jv_tisp2.html	- liczenie czasu pobytu na stronie
jv_headl.html	- ruchome napisy na klawiszu

Style

- Style kaskadowe (**CSS**) występują na trzech poziomach:
 - lokalnie;
 - w nagłówku dokumentu
 - w wyodrębnionym pliku (styl zewnętrzny).
- Styl lokalny "nakrywa" styl z nagłówka, ten zaś "nakrywa" styl zewnętrzny.
- Można więc stworzyć zewnętrzny plik z ogólnymi regułami, od których wyjątki będą definiowane na poziomie dokumentu lub określonego znacznika (lokalnie).

Style - rodzaje

□ Przykładowo definicja stylu nagłówkowego H1 może wyglądać następująco:

- *lokalnie*

```
<H1 style="color: blue; font-style: italic">
```

- *w nagłówku*

```
<STYLE>
```

```
H1 {color: blue; font-style: italic}
```

```
</STYLE>
```

- *w pliku zewnętrznym*

```
<link rel=stylesheet type="text/css"  
      href="nazwa_pliku_stylów.html">
```

Style - składnia

Ogólna składnia deklaracji stylu wygląda następująco:

```
selektor {cecha-1: wartość-1; cecha-2: wartość-2; ... cecha-n: wartość-n}
```

gdzie:

- selektorem będzie jeden ze znaczników HTML, takich jak np.:
H1, H2 ... H6, P, TR, TD, LI itd.

- zaś cechą:

czcionka

font-family: font-size: font-style: font-weight:

kolor i tło

color: background-color: background-image:
background-position: background-repeat:

tekst

letter-spacing: line-height: text-align: text-decoration:
text-indent: text-transform: vertical-align:

listy

list-style-image: list-style-position: list-style-type:

Serwery WWW

- ❑ **Serwer WWW** to zwykły komputer, posiadający stały numer IP oraz uruchomiony program serwerowy (protokół http).
- ❑ Serwer może nazywać się dowolnie, ale tradycją jest by nazwa serwera WWW to było po prostu **www**, co sprzyja to łatwemu zapamiętywaniu adresów.
- ❑ Serwer WWW udostępnia zawartość wybranego folderu poprzez protokół http, a przeglądarki komunikujące się z serwerami pobierają w ten sposób zawartość stron.
- ❑ Folder, w którym zlokalizowane są dokumenty nazywa się *document root* i określa się go w konfiguracji serwera - np.: `C:\www` albo `C:\apache\htdocs` (w systemach Windowsowych) lub `/var/www` czy też `/usr/web/www` (w systemach Unixowych).

Publikowanie dokumentu

- ❑ Publikując dokumenty rzadko korzysta się z własnego komputera, bowiem uruchamiając na nim serwer nie można go już wyłączać (serwery internetowe działają bez przerw, nie wiadomo bowiem kiedy i skąd nastąpi połączenie).
- ❑ Zwykle zatem używa się serwerów macierzystych instytucji (np. uniwersytetów, bibliotek) lub darmowych serwerów publicznych.
- ❑ Można też publikować dokumenty na serwerach komercyjnych (zwykle wiąże się to z dodatkowymi udogodnieniami - większą ilością miejsca na dysku, stabilniejszym i łatwiej dostępnym kontem itp. - ale też opłatą).

Publikowanie dokumentu

- Korzystając z odległego serwera otrzymuje się do dyspozycji tzw. konto wiążące się z dostępem do określonego folderu, zwykle o ograniczonej pojemności (5 do 20 MB).
- Jedyne problem to transport plików na odległy serwer.
 - **FTP** - jest to najpowszechniejszy, najbardziej uniwersalny i najszybszy sposób transportu plików.
 - **Ładowanie przez przeglądarkę** (tzw. *Upload*)- jest wolniejszy ale w przypadku małych plików wystarczająco efektywny.
 - **Zdalne edytory WWW** - w przypadku gdy chcemy stworzyć szybko prostą stronę, można skorzystać z automatu tworzącego strony WWW wprost na zdalnym serwerze.