

HTML / HTML dla zielonych - Kurs HTML

W tym rozdziale dowiesz się...

Kurs języka HTML dla zupełnie początkujących. Napisz własną stronę WWW nawet w ciągu jednego dnia i wprowadź ją do internetu za darmo. Stworzenie strony WWW jest naprawdę proste. Zapraszam!

Zaczynamy!

Czy tworzenie stron internetowych naprawdę jest tak trudne jak mówią?

Edytory

Który edytor HTML wybrać: Pajaczek, CoreEditor, Bluefish, Brackets, PSPad, gedit, Kate, Quanta Plus, SCREAM, Smultron?

Ramy dokumentu

Jak wygląda typowy dokument HTML? Co to są podstrony?

Wpisywanie tekstu

W jaki sposób wpisuje się tekst na stronach WWW? Jakie są zasady poprawnego wpisywania znaków interpunkcyjnych w tekście komputerowym?

Znaczniki

Co to są znaczniki HTML?

Koniec linii

W jaki sposób przenieść tekst do następnej linijki w dokumencie HTML?

Akapit

W jaki sposób układać tekst na ekranie? Jak dodać nowy akapit? Jak wyśrodkować lub wyjustować tekst? Co to są atrybuty HTML?

Tekst pogrubiony

W jaki sposób pogrubić (wytłuszczyć) tekst na stronie WWW?

Tekst pochylony

W jaki sposób pochylić tekst na stronie WWW (kursywa)?

Tekst podkreślony

W jaki sposób podkreślić tekst na stronie WWW?

Wielkość czcionki

W jaki sposób zmienić rozmiar czcionki na stronie WWW?

Kolor czcionki

W jaki sposób zmienić kolor czcionki na stronie WWW?

Rodzaj czcionki

W jaki sposób zmienić rodzaj czcionki na stronie WWW?

Łączenie parametrów

W jaki sposób zmienić wygląd tekstu na stronie WWW?

Kolor tła oraz tekstu

W jaki sposób zmienić kolor tła oraz kolor tekstu na stronie WWW?

Wstawienie obrazka

W jaki sposób wstawić obrazek (grafikę, zdjęcie) na stronę WWW?

Ustawienie obrazka

W jaki sposób określić ustawienie obrazka (grafiki, zdjęcia) na stronie WWW?

Wyśrodkowanie obrazka

W jaki sposób ustawić obrazek (grafikę, zdjęcie) lub inne elementy na środku ekranu (wyśrodkowanie, centrowanie)?

Odsyłacz do podstrony

Do czego służą odsyłacze (hiperłącza, linki, odnośniki hipertekstowe)?

Odsyłacz do adresu internetowego

Jak wstawić link (odsyłacz, hiperłącze, odnośnik hipertekstowy) na stronie WWW?

Odsyłacz pocztowy

Jak wstawić adres e-mail na stronie WWW?

Odsyłacz obrazkowy

W jaki sposób wstawić na stronie WWW odnośnik (link, hiperłącze, odsyłacz hipertekstowy) obrazkowy (graficzny), czyli klikalny przycisk?

Jak zrobić dobrą stronę

Czego unikać, aby Twoja strona WWW nie odstraszała internautów?

Podsumowanie

HTML

Quiz

HTML

Ćwiczenia

HTML

Zaczynamy!

Czy tworzenie stron internetowych naprawdę jest tak trudne jak mówią?

Widzę, że jesteś "zielony/zielona"... ale nic się nie martw. Jeśli koniecznie chcesz "zmienić kolor", przeczytaj umieszczony poniżej tekst. Pozwoli Ci on, stworzyć Twoją pierwszą stronę internetową, nawet w ciągu jednego dnia. Jeśli uważasz, że pisanie stron w języku HTML jest dla Ciebie "czarną magią", a sama strona jest jakimś tajemniczym i bardzo skomplikowanym dokumentem, to się mylisz. Napisanie krótkiej strony internetowej jest prostsze niż Ci się wydaje. Zatem nie trać już czasu na wymówki typu: „*Ja się niczego nie nauczę!*” i tym podobne, bo to nieprawda. Zaczynij już lepiej czytać.

Mam tylko jedną prośbę: postaraj się przeczytać w miarę uważnie i po kolei całą treść na tej stronie. Jeśli pominiesz jakiś punkt lub przeczytasz go zbyt pobieżnie, może to spowodować, że nie zrozumiesz następnych.

Źródła

Opis tutaj zawarty, został opracowany na podstawie następujących specyfikacji:

- HTML 4.01 Specification
- XHTML 1.0 The Extensible HyperText Markup Language
- XHTML 1.1 - Module-based XHTML
- HTML5 - A vocabulary and associated APIs for HTML and XHTML
- Differences from HTML4

Edytory

Który edytor HTML wybrać: Pajęczek, CoreEditor, Bluefish, Brackets, PSPad, gedit, Kate, Quanta Plus, SCREAM, Smultron?

Co jest potrzebne, żeby napisać stronę internetową? Teoretycznie może to być dowolny edytor tekstu, no i oczywiście dobre chęci. Dokument HTML, czyli Hypertext Markup Language - Hipertekstowy Język Oznaczania, jest po prostu plikiem tekstowym, gdzie wpisujemy wszystkie polecenia, dotyczące formatowania tekstu, wstawiania grafiki i inne. Lecz pisanie strony w zwykłym edytorze tekstu byłoby bardzo uciążliwe. Dlatego powstało wiele wyspecjalizowanych edytorów, które ułatwią, przyspieszą i uprzyjemnią Twoją pracę. Poniżej znajdziesz listę popularnych edytorów HTML (kolejność wg licencji i alfabetycznie). Informacje częściowo zostały zaczerpnięte ze stron producentów.

Windows

Pajęczek (shareware - płatny)

Najbardziej znany polski edytor HTML rozwijany od lat. Oferuje 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.

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.

Bluefish (darmowy) ★★★★★

Szybki i lekki edytor dla webmasterów-programistów. Wspiera m.in.: HTML, PHP, C, Java, JavaScript, JSP, SQL, XML, Python, Perl, CSS, ColdFusion, Pascal. Zawiera kreatory znaczników HTML, a także specjalne dla szkieletu strony, tabel, ramek, albumów zdjęć itp. Automatyczne domyka znaczniki. Posiada wbudowane narzędzie wyszukiwania i zamiany w wielu plikach jednocześnie oraz przeglądarkę dokumentacji znaczników i funkcji. Integruje się z innymi aplikacjami. Istnieje polska wersja językowa.

Brackets (darmowy) ★★★★★

Lekki, a zarazem bardzo zaawansowany i nowoczesny edytor tekstu przeznaczony przede wszystkim dla webmasterów. Dzięki podglądowi na żywo bezpośrednio w przeglądarce można na bieżąco weryfikować efekty zmian wprowadzanych w tworzonym kodzie HTML i CSS. Dodatkowo obsługuje instalowanie wielu darmowych rozszerzeń, które pozwalają znacznie zwiększyć możliwości programu i jeszcze lepiej dopasować go do własnych potrzeb.

PSPad (darmowy) ★★★★★

Szybki i lekki edytor produkcji czeskiej, jednak radzący sobie z kodowaniem polskich znaków lepiej niż niejeden rodzimy program. Oczywiście posiada polską wersję językową. **Uwaga:** zaraz po instalacji należy w ustawieniach programu usunąć skróty klawiaturowe kolidujące z kombinacjami

służącymi do uzyskania polskich znaków diakrytycznych (np. *Alt+L*)!

Wspiera m.in. następujące języki: HTML, XHTML, XML, CSS, JavaScript, PHP, C/C++, Java, Pascal, Perl, Python, RSS, SQL (MySQL, Interbase, MSSQL, Oracle, Sybase), Visual Basic, Assembler. Pozwala korzystać z szablonów dokumentów. Ma wbudowany podgląd dokumentów za pomocą Internet Explorera i Mozilli. Integruje się z zewnętrznymi programami, między innymi edytorem CSS. Obsługuje mechanizm wtyczek. Zawiera narzędzie do kompresji i dekompresji kodu HTML oraz rozbudowany eksplorator kodu z listą funkcji i elementów zawartych w edytowanym dokumencie. Posiada zaimplementowaną funkcję dopasowywania nawiasów, podpowiedzi oraz kreatory znaczników i funkcji, które można samodzielnie tworzyć i modyfikować. Obsługuje makrokody. Zawiera wbudowane narzędzie do porównywania plików oraz wyszukiwania i zamiany w wielu plikach jednocześnie. Przydatną funkcją jest także próbnik kolorów ekranu, tabela znaków *ASCII* [*American Standards Committee for Information Interchange*] oraz sprawdzanie pisowni.

Linux

Bluefish (darmowy) ★★★★★

Oprócz wersji dla systemu Windows (opis powyżej) edytor ten dostępny jest także dla Linuksa.

Brackets (darmowy) ★★★★★

Oprócz wersji dla systemu Windows (opis powyżej) edytor ten dostępny jest także dla Linuksa.

gedit (darmowy) ★★★★★

Oficjalny edytor środowiska *GNOME* [*GNU Network Object Model Environment*], cechujący się prostotą. Wspiera m.in.: C, C++, Java, HTML, XML, Python, Perl. Automatycznie dodaje wcięcia linii kodu oraz dopasowuje nawiasy. Obsługuje wtyczki, np.: statystyki dokumentu, integracja z zewnętrznymi aplikacjami, makrokody, lista znaczników.

Kate (darmowy) ★★★★★

Wchodzi w skład środowiska *KDE* [*K Desktop Environment*]. Dostępna również uproszczona wersja pod nazwą *KWrite*. Podświetla składnię, dopasowuje nawiasy oraz automatycznie dodaje wcięcia linii kodu. Obsługuje autouzupełnianie, a także mechanizm wtyczek. Wbudowana jest funkcja sprawdzanie pisowni oraz konwersja kodowania znaków.

Quanta Plus (darmowy)

Wchodzi w skład środowiska *KDE* [*K Desktop Environment*]. Podświetla składnię m.in.: HTML, XHTML, XML, XSLT, CSS, JavaScript, Java, PHP, C/C++, C#, Python, Perl, SQL (MySQL, PostgreSQL), Assembler, ASP, JSP, Pascal. Dopasowuje i automatycznie domyka nawiasy, jak również dodaje wcięcia linii kodu. Zawiera liczne kreatory znaczników, zaawansowane autouzupełnianie, podpowiedzi i automatyczne zamykanie znaczników. Umożliwia podgląd dokumentu w kilku trybach oraz przeglądarkach (Konqueror, Firefox, Mozilla, Netscape, Opera, Lynx). Posiada edytor znaczników i atrybutów, a także zaawansowany eksplorator struktury dokumentu. Załączona jest dokumentacja dla języków HTML, CSS, JavaScript, PHP. Zawiera szablony dokumentów i skryptów. Obsługuje wtyczki oraz integruje się z innymi aplikacjami. Posiada zintegrowane narzędzie wyszukiwania i zamiany w wielu plikach jednocześnie, porównywania plików oraz edytor map odsyłaczy. Wspiera *CVS* [*Concurrent Versions System*]. Wbudowana jest funkcja sprawdzania pisowni.

SCREAM - Site CRreation and Editing EnvironMent (darmowy)

Zawiera eksplorator linków z możliwością sprawdzenia martwych odnośników. Udostępnia kreatory, autouzupełnianie, podpowiedzi i automatycznie zamykanie znaczników. Obsługuje szablony dokumentów (interpretuje pliki programu *Dreamweaver*). Wspiera *CVS* [*Concurrent Versions System*]. Posiada wbudowany eksplorator struktury dokumentu. Integruje się z innymi aplikacjami. Zawiera narzędzie wyszukiwania i zamiany w wielu plikach jednocześnie.

Mac OS X

Smultron (płatny)

Oferuje podświetlanie składni, tworzenie biblioteki *snippetów*, czyli często używanych bloków kodu. Posiada eksplorator funkcji występujących w edytowanym dokumencie. Obsługuje autouzupełnianie i automatyczne domykanie znaczników. Zawiera wbudowane narzędzie wyszukiwania i zamiany w wielu plikach jednocześnie. Obsługuje *AppleScript*.

Bluefish (darmowy) ★★☆☆☆

Oprócz wersji dla systemu Windows (opis powyżej) edytor ten dostępny jest także dla Mac OS X.

Brackets (darmowy) ★★☆☆☆

Oprócz wersji dla systemu Windows (opis powyżej) edytor ten dostępny jest także dla Mac OS X.

Jeśli nie wiesz, który edytor HTML będzie dla Ciebie odpowiedni albo wolisz poszukać programu obcojęzycznego, zobacz: Ranking edytorów HTML (online). Dodatkowe opisy i porównania wraz z plikami do ściągnięcia znajdziesz również np. w serwisie DobreProgramy.pl lub w magazynie Komputer Świat.

Istnieją również **edytory graficzne**, tzw. **WYSIWYG** (ang. *What You See Is What You Get*, czyli *To Co Widzisz Jest Tym Co Otrzymasz*). Jest nim choćby **FrontPage** dostępny w pakiecie biurowym *Microsoft Office*. Aby się nimi posługiwać, nie musisz wcale znać języka HTML, ponieważ stronę tworzy się w nich jak w zwykłym edytorze tekstu, a cały kod HTML generuje automatycznie program. Ale nie ma nic za darmo (no może oprócz tego kursu :-). Takie edytory nie dają Ci pełnej kontroli nad zawartością strony. Możesz korzystać z bogatych narzędzi, lecz czasami nauka wszystkich możliwości takiego edytora, może zabrać więcej czasu, niż nauka podstaw HTML. Dodatkowo generują "nadmiarowy" kod, co zwiększa objętość strony oraz powoduje, że bardzo trudno ją później modyfikować przy użyciu edytorów tekstowych. Często wywołuje to również nieprawidłowy sposób wyświetlania w niektórych popularnych przeglądarkach internetowych - np. strony tworzone w edytorze FrontPage, najlepiej wyświetlają się w przeglądarce Internet Explorer. Poza tym istnieją przypadki, w których znajomość choćby podstaw HTML jest wprost niezbędna, ponieważ edytory graficzne po prostu nie udostępniają wszystkich możliwości lub uzyskany efekt nie do końca jest zgodny z oczekiwaniami.

Chociaż stronę WWW można napisać w zwykłym edytorze tekstu, absolutnie nie polecam tej metody! Po pierwsze łatwo w ten sposób popełnić błędy literowe w nazwach poleceń HTML, co spowoduje błędy w wyświetlaniu strony. Wyspecjalizowane edytory HTML od razu o nich informują, ponieważ każde poprawne polecenie HTML zwykle jest w nich napisane innym kolorem (jest to tzw. *funkcja kolorowania składni*) - niepoprawne komendy nie zmieniają koloru. Cała treść w zwykłym edytorze tekstu jest jednolita, co sprawia, że nie zauważymy błędów.

Jeśli do tworzenia dokumentów HTML użyjesz np. *windowsowskiego Notatnika* (popularny *Notepad*), to polskie znaki diakrytyczne (czyli popularne "ogonki") mogą zostać błędnie zapisane - w przeglądarce internetowej pojawiają się w ich miejscu "krzaczk".

Na różnych forach i grupach dyskusyjnych czasem można spotkać trochę żartobliwe wypowiedzi typu: „*Prawdziwy webmaster nie używa żadnych dodatkowych edytorów, bo tak jest zbyt łatwo*”. Nigdy nie wierz w takie bajki! Prawdziwego webmastera cechują w największym skrócie: wiedza, zdolności oraz umiejętność wykorzystania właściwych narzędzi, które znacznie ułatwiają, a przede wszystkim przyspieszają pracę. Ponadto, z niezrozumiałych powodów, osoby początkujące (choć nie tylko) często twierdzą, że nie używają edytora HTML, bo chcą się "dobrze" nauczyć języka HTML. Takie rozumowanie jest całkowicie błędne! Czy używając zwykłego długopisu nauczymy się pisać gorzej niż gdybyśmy to robili przy pomocy gęsiego pióra i płamiącego atramentu? Przeciwnie - tekst napisany długopisem będzie wyglądał bardziej estetycznie (brak kleksów :-). Przecież zasady pisowni zawsze są takie same, bez względu na narzędzie jakiego używamy. Podobnie wykorzystanie tekstowych edytorów HTML wcale nie oznacza, że gorzej nauczymy się języka HTML. Przyspieszy to tylko naszą pracę i uchroni od popełnianych błędów składniowych, czyli raczej pozwoli nam jeszcze lepiej poznać język HTML. Większość doświadczonych

webmasterów-profesjonalistów wykorzystuje tekstowe edytory HTML i Tobie też radzę jak najszybciej zaopatrzyć się w taki. Im wcześniej to zrobisz, tym lepiej dla Ciebie.

Z powyższych powodów jeszcze raz gorąco zachęcam do zainstalowania edytora HTML - najlepiej jakiegoś polskiego darmowego (freeware) - oferta jest dość bogata, a większość z nich udostępnia zupełnie wystarczające funkcje, aby stworzyć świetną stronę WWW.

Ramy dokumentu

Jak wygląda typowy dokument HTML? Co to są podstrony?

Stronę WWW tworzy się podobnie jak zwykły dokument tekstowy. Po uruchomieniu edytora HTML należy wybrać opcję z menu: *Plik/Nowy* (lub *File/New*). Teraz można już zacząć pisanie strony. Jednak ponieważ dokument HTML nie jest całkowicie zwykłym plikiem tekstowym (zawiera *hipertekst*, osadzone obrazki i musi być poprawnie wyświetlany w różnych systemach operacyjnych na całym świecie), dlatego wymyślono specjalny szablon dokumentu HTML, który powinien być przestrzegany.

Oto jak przykładowo powinny wyglądać ramy każdego dokumentu w formacie HTML (istnieją również inne podobne szablony):

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <meta name="description" content="Tu wpisz opis zawartości strony">
  <title>Tu wpisz tytuł strony</title>
</head>
<body>

  Tu wpisuje się treść strony

</body>
</html>
```

UWAGA!

Powyższy szablon dotyczy tzw. *kodu źródłowego*, dlatego jeżeli używasz edytora graficznego (np.: FrontPage), **koniecznie** musisz się przełączyć w tryb bezpośredniej edycji kodu źródłowego HTML. Dodatkowo jeśli używasz zwykłego edytora tekstu (np. *windowsowskiego Notatnika - Notepad*), to niektóre polskie znaki mogą zostać błędnie zakodowane - zamiast nich pojawią się "krzaczki"! Jeszcze raz radzę zaopatrzyć się w jakiś edytor HTML - najlepiej polski.

Większość edytorów HTML przy tworzeniu nowego dokumentu automatycznie wpisuje swój własny szablon - zbliżony do powyższego. Jeśli chcesz, możesz z niego skorzystać, modyfikując opisane poniżej fragmenty. Proponuję jednak zastąpić go szablonem przedstawionym w tym rozdziale. Również w tym przypadku wszystkich zmian należy dokonywać bezpośrednio w trybie edycji kodu źródłowego HTML! Jeśli posiadasz edytor graficzny, musisz się do niego przełączyć, natomiast edytory tekstowe najczęściej posiadają tylko taki tryb - wtedy nie trzeba nic przełączać.

Wskazówka dla bardzo początkujących

Nie musisz za każdym razem, gdy stworzysz nową stronę, przepisywać całego tekstu powyższego szablonu. Wystarczy w przeglądarce internetowej naprowadzić wskaźnik myszki na początek tekstu (wskaźnik powinien zmienić kształt) i wcisnąć lewy przycisk myszki. Trzymając cały czas wciśnięty lewy przycisk, przeciągnij myszką do końca tekstu szablonu, po czym zwolnij przycisk - tekst zostanie zaznaczony. Teraz wcisnij na klawiaturze jednocześnie klawisze: **Ctrl+C**, co wywoła skopiowanie zaznaczonego tekstu do schowka systemowego. Następnie w edytorze HTML po otwarciu nowego pliku wystarczy wcisnąć: **Ctrl+V**, a do dokumentu zostanie wklejony tekst ze schowka.

Jeśli jesteś bardziej doświadczonym użytkownikiem komputera, takie rady pewnie Cię śmieszą. Proszę jednak o wyrozumiałość - każdy kiedyś zaczynał. Przypomnij sobie Twoje początki...

Większości tekstu w powyższym szablonie nie powinno się zmieniać. Natomiast można, a nawet należy, zmienić tekst, który został *wyróżniony* - odpowiednie fragmenty zostaną szczegółowo opisane poniżej.

Cała treść która znajduje się pomiędzy <head> oraz </head> (w szablonie powyżej) jest nazywana *nagłówkiem dokumentu* (treść nagłówkowa), natomiast wszystko pomiędzy <body> oraz </body> stanowi tzw. *ciało dokumentu* (treść właściwa).

A teraz wyjaśnimy sobie ważniejsze linijki.

- W miejsce tekstu: *Tu wpisz opis zawartości strony*, należy wpisać słowny opis, co znajduje się na Twojej stronie internetowej. Z wiadomości tej korzystają wyszukiwarki internetowe - takie jak Google. Tekst który tutaj wpiszesz, pojawi się w wyszukiwarce po odnalezieniu przez nią Twojej strony. Jeśli nie wpiszesz tutaj nic, wyszukiwarka wyświetli najczęściej kilka pierwszych linijek, które znajdują się na stronie. Ciekawy, ale niezbyt długi opis może zachęcić internautów do odwiedzenia Twojego serwisu! Każda pojedyncza strona, która znajduje się w serwisie, powinna mieć swój własny opis, przedstawiający jej treść.
- Pomiedzy <title> oraz </title> (w miejsce tekstu: *Tu wpisz tytuł strony*) należy wpisać tytuł, jaki będzie nosiła Twoja strona. Może on zawierać dowolny tekst, zaleca się jednak, aby nie był zbyt długi (najwyżej kilka wyrazów). Powinien on odnosić się do treści zawartych na danej stronie. Tytuł pojawia się na belce tytułowej przeglądarki internetowej (na samej górze okna programu), a także w wyszukiwarkach internetowych, po odnalezieniu Twojej strony - dlatego właśnie nie może on być zbyt długi. Trafny tytuł może również zachęcić internautów do obejrzenia Twojej strony!
- I najważniejsze - w miejsce tekstu: *Tu wpisuje się treść strony* (pomiedzy <body> oraz </body>) należy wpisać właściwą zawartość strony. Może to być po prostu zwykły tekst, który ma się pojawić na stronie. Są to również tzw. *znaczniki* sterujące wyglądem dokumentu (które zostaną opisane dalej).

Dla ciekawskich ;-)

Jeśli już teraz chcesz koniecznie wiedzieć, co oznaczają nieopisane powyżej linijki, zobacz rozdziały: Składnia (X)HTML5 oraz Deklaracja strony kodowej.

UWAGA!

W żadnym wypadku nie należy pomijać w powyższym szablonie następującej linijki:

```
<meta charset="utf-8">
```

Co to są podstrony?

Praktycznie wszystkie serwisy internetowe składają się z pewnej liczby podstron, a nie tylko z jednej, bardzo długiej strony głównej. Każda podstrona to po prostu osobny plik (z rozszerzeniem ***.html** lub ***.htm**), który tworzy się w taki sam sposób, jak stronę główną (przy użyciu szablonu przedstawionego powyżej). Np. ten serwis składa się z następujących podstron (zobacz w spisie treści): "HTML dla zielonych", "Nagłówek i treść", "Tekst" itd. Taka organizacja treści ułatwia użytkownikowi nawigację oraz zapobiega niepotrzebnemu wczytywaniu całego serwisu od razu, co trwałoby prawdopodobnie bardzo długo.

Wszystkie podstrony zapisuje się zwykle w tym samym katalogu - jako osobne pliki - lub w różnych katalogach (jeśli chcemy je jakoś posegregować tematycznie). Z jednej podstrony użytkownik może automatycznie przechodzić do innych, klikając odpowiednie odnośniki (jeśli je umieścimy).

Jeszcze jedno:

Plik będący stroną startową (czyli główną) musi nazywać się **index.html** lub **index.htm**. Twoja witryna internetowa może składać się z dowolnej ilości podstron - każda z nich to osobny plik, tworzony w ten sam sposób, jak strona główna. Inne podstrony mogą nazywać się dowolnie, byleby miały rozszerzenie ***.html** lub ***.htm**. Jednak stosowanie bardzo długich nazw może być uciążliwe - szczególnie w przypadku gdy internauta będzie chciał się bezpośrednio przenieść do którejś z podstron Twojego serwisu poprzez wpisanie jej adresu w przeglądarce (każda strona ma swój unikatowy adres, do którego można się odnosić).

UWAGA! Bardzo ważne: w nazwach plików lepiej nie używać:

- wielkich liter
- znaków specjalnych, np.: \ / : * ? " < > |
- spacji (jeśli musisz, w zamian używaj podkreślnika "_")
- polskich liter (ą, ć, ę, ł, ń... itd.)

Jeśli nie zastosujesz się do tych zaleceń, może się okazać, że po wprowadzeniu strony do Internetu, pliki takie nie wczytają się (nawet jeśli na Twoim komputerze wszystko jest w porządku)!

W dalszych punktach na tej stronie zostanie zamieszczony opis podstawowych poleceń, które mogą Ci być pomocne w napisaniu Twojej pierwszej najprostszej strony internetowej. Wszystkie je należy umieszczać pomiędzy <body> oraz </body> (w ramach właściwej treści strony). Jeśli chcesz wprowadzić na stronę zwykły tekst, możesz go po prostu bezpośrednio wpisać - bez używania żadnych dodatkowych poleceń.

Wpisywanie tekstu

W jaki sposób wpisuje się tekst na stronach WWW? Jakie są zasady poprawnego wpisywania znaków interpunkcyjnych w tekście komputerowym?

Jeśli chcemy umieścić na stronie WWW zwykły tekst, możemy wpisać go bezpośrednio z klawiatury - w miejsce właściwej treści dokumentu (patrz: Ramy dokumentu). Nie trzeba przy tym stosować żadnych dodatkowych poleceń. Należy jednak pamiętać, że przeglądarka internetowa automatycznie zawinie wszystkie wiersze, dlatego w edytorze HTML tekst możemy wpisywać dowolnie, np. po wpisaniu:

To jest zwykły tekst...
 To jest zwykły tekst...
 To jest zwykły tekst...
 To jest zwykły tekst...

Na ekranie otrzymamy:

To jest zwykły tekst... To jest zwykły tekst... To jest zwykły tekst...

W edytorze możemy zakończyć linię (klawisz Enter) w dowolnym miejscu - tam gdzie będzie nam wygodnie. Nie należy również przedzielać wyrazów z jednej linii do drugiej za pomocą myślnika:

To jest zwykły tekst... To jest zwykły tekst... To jest **zwy-**
kły tekst...

Zaznaczony wyraz powinniśmy wpisać normalnie. Przeglądarka sama ustawi go w odpowiedniej linii.

Na koniec kilka praktycznych porad, które początkowo łatwo zbagatelizować, ale uwierz mi - naprawdę okazują się niezwykle przydatne:

- Dbaj o poprawność ortograficzną i stylistyczną tekstu! O ile błędy literowe mogą się zdarzyć każdemu, to rażące błędy ortograficzne, popełniane zbyt często, odstraszą potencjalnych czytelników. Jeśli masz kłopoty z ortografią (jak większość Polaków :-), sprawdzaj tekst w słowniku komputerowym (np. w *Wordzie*). Wiele edytorów HTML ma własny słownik.
- Używaj znaków interpunkcyjnych, a zwłaszcza przecinków! Jeśli nie będziesz ich używać, tekst który napiszesz, może okazać się zupełnie niezrozumiały dla czytającego. Przypominam, że przecinki stawiamy pomiędzy zdaniami prostymi, wchodzącym w skład dłuższego zdania złożonego (zdanie złożone zawiera kilka czasowników, czyli słów odpowiadających na pytanie: "co robi?"). Stawia się je prawie zawsze przed: "że", "iż" (wyjątki: "..., chyba że...", "..., dlatego że...", "..., (po)mimo że/iż..."), a najczęściej również przed: "a", "ale", "lecz", "aby", "żeby", "ponieważ", "bo", jak również przed wyrażeniami zawierającymi "który" (np.: "z którym", "w którym", "po którym", "wewnątrz którego"

itd.). Nie stawiamy przecinków przed ani za spójnikami: "i", "oraz", "lub", "albo", "bądź", "ani", "czy" (jeśli pełni rolę spójnika).

Kropki używamy - poza zakończeniem zdania - również po skrótach (np.: "prof. [profesor]", "inż. [inżynier]"). Wyjątkiem są skróty, kończące się na ostatnią literę pełnego wyrazu (np.: "mgr [magister]", "nr [numer]", ale "dyr. [dyrektor]"). Ponadto często stosuje się nawiasy dla przedstawienia jakiejś uwagi pobocznej (w ich miejsce można również użyć myślników).

- Oddzielaj nowymi akapitami fragmenty tekstu, które nieco różnią się od siebie tematycznie. Bardzo długi "czysty" tekst niezbyt wygodnie się czyta.

Nie musisz od razu stać się literatem... chyba że piszesz stronę literacką ;-). Nie chodzi mi o to, aby teksty na Twojej stronie były wprost idealne pod względem gramatycznym i ortograficznym. Nie popadajmy w paranoję. To jest kurs HTML, a nie nauka poprawnej polszczyzny. Z pewnością również w tym serwisie znajdziesz błędy - wybaczenie, nie jestem polonistą. Jak głosi słynny cytat: „*Mylić się jest rzeczą ludzką... i nic co ludzkie nie jest mi obce*” (co, nie wiesz kto to powiedział... ja też nie :-), jednak przydatne może okazać się chociaż minimum wiedzy przedstawione tutaj. Ważne, aby Twoja strona nie odstraszała internautów poprzez niepotrzebne błędy, których przecież można bardzo łatwo uniknąć.

Jeśli napiszesz swoją stronę internetową "byle jak", może to sugerować, że również informacje na niej zawarte nie są wiarygodne!

To wszystko była powtórka, lecz jest coś, czego najczęściej nie uczą w szkole, a co jest niezbędne przy pracy z tekstem komputerowym - są to zasady wpisywania znaków interpunkcyjnych:

- **Podstawowe znaki przestankowe:** kropka ("."), przecinek (","), wykrzyknik ("!"), pytajnik ("?"), dwukropek (":"), średnik (";"), wielokropek (". . ."). Przed tymi znakami nigdy nie stawiamy spacji! Spację stawiamy natomiast po nich.

Wyjątek stanowi sytuacja, gdy bezpośrednio po sobie następuje kilka takich znaków - wtedy spację stawiamy tylko po ostatnim z nich.

Drugi wyjątek stanowią krótkie kilkuwyrazowe skróty, w których każdy skrócony wyraz jest zakończony kropką - wtedy spację stawiamy tylko na końcu takiego skrótu (np.: "m.in. [między innymi]" - między innymi, "Sp. z o.o. [Spółka z ograniczoną odpowiedzialnością]" - spółka z ograniczoną odpowiedzialnością; ale "100 tys. zł [100 tysięcy złotych]").

Trzeci wyjątek stanowią daty, godziny i liczby - jeśli występuje wewnątrz nich kropka, przecinek lub dwukropek, nie stawiamy po nim spacji (np.: "15.07.1410 r.", "9:08", "12,5" - notacja polska, "12.5" - notacja angielska).

Ponadto należy zauważyć, że pojedyncze kropki w wielokropku nie oddzielamy spacjami, a jeśli stanowi on koniec zdania, nie stawiamy dodatkowej kropki. Podobnie jeżeli na końcu zdania znajduje się skrót zakończony kropką, nie stawiamy już po nim drugiej kropki. Nie stawiamy spacji po wielokropku, jeśli rozpoczynamy od niego nowy fragment tekstu - dla wskazania kontynuacji jakiejś wcześniejszej wypowiedzi.

Jeżeli chodzi o jednostki fizyczne, zawsze stawiamy przed nimi spację (nie dotyczy znaku procenta). Tuż po nich nie stawia się kropki, chyba że znajdują się na końcu zdania.

Poprawnie: ...wyraz... wyraz. wyraz, wyraz! wyraz? wyraz: wyraz; wyraz...
np.: tel./fax m.in. 15.07.1410 r., 9:08, 1 kg, 100%, itp.

Niepoprawnie: ... wyraz ... wyraz , wyraz ! wyraz,wyraz wyraz
:wyraz...wyraz ... wyraz. . . wyraz... . np. : tel. / fax m. in. 15. 07.
1410r. , 9: 08, 1 kg., 1kg, 100 %, itp..

- **Nawiasy i cudzysłowy.** Treść znajdująca się w nawiasie nigdy nie powinna być od niego oddzielona spacjami (dotyczy to zarówno początku jak i końca)! Przed otwarciem nawiasu zawsze stawiamy spację. Zwykle stawiamy ją również po jego zamknięciu. Wyjątek stanowi sytuacja, w której bezpośrednio po nawiasie występuje podstawowy znak przestankowy - wtedy nie stawiamy spacji między nimi. Wszystko to odnosi się również do cudzysłowu.

Poprawnie: wyraz (treść) wyraz "treść" wyraz (treść), wyraz "treść"! wyraz - (...) - ("treść!") "treść" (itd.).

Niepoprawnie: wyraz (treść) wyraz(treść)wyraz wyraz(treść) wyraz "treść" , wyraz "treść" ? -(...)-"treść"("treść! ") (itd.) .

- **Pauza (myślnik)** - wpisujemy ją oddzielając z obu stron spacjami. Wyjątek stanowią łączniki wchodzące w skład złożonych wyrazów (np.: "biało-czerwony"), końcówki (np.: "PC-y"), nr telefonu, kody pocztowe itp. - wtedy nie rozdzielamy ich spacjami.

Poprawnie: To jest - jak głoszą stare pisma - poprawnie. biało-czerwony 20-letni PC-y 99-999 e-mail

Niepoprawnie: To jest- jak głoszą stare pisma-niepoprawnie. biało - czerwony, biało- czerwony, 20 - letni 99 - 999 e - mail

- **Operatory matematyczne** ("+", "-", "*", "/" "=") piszemy najczęściej oddzielając je z obu stron spacjami. Wyjątek stanowią znaki minus i plus, które nie oznaczają działania matematycznego, lecz znak liczby - wtedy nie stawiamy po nich spacji. Ponadto nawias występujący w nazwach funkcji nie poprzedzamy spacją.

Poprawnie: $f(x, y) = -2x + 3y + 4$

Niepoprawnie: $f(x,y) = -2x + 3y + 4$

Wiem, że uwagi tutaj przedstawione mogą niektórych trochę śmieszyć (szczególnie pierwsza część).

Bądźcie tolerancyjni dla tych, którzy nie są tak doświadczeni. Przypomnijcie sobie, czy nigdy nie trafiliście na stronę WWW, na której wręcz roiło się od denerwujących błędów. Co wtedy można pomyśleć o takim serwisie? Mam nadzieję, że uczestnikom tego kursu nigdy to się nie przytrafi.

Znaczniki

Co to są znaczniki HTML?

Poza zwykłym tekstem na stronę możesz wprowadzić *znaczniki* (tzw. *tagi*). **Znacznik** jest to specjalny tekst umieszczony w nawiasach ostrych - np.: . Jest on częścią składni języka HTML i pozwala sterować wyglądem strony. Dzięki niemu możesz np. ustalić kolor tła, rodzaj formatowania tekstu, wstawić obrazek czy tabelę itd. Znacznik nie jest widoczny, na ekranie. Widoczne są tylko efekty jego działania (np. wstawienie obrazka). Ja sam aby umieszczony powyżej znacznik był widoczny, musiałem się posłużyć pewną "sztuczką" (jeśli nie możesz wytrzymać i już teraz chcesz wiedzieć jaką, zajrzyj na stronę: Znaki specjalne).

Ponieważ znaki: "<" (znak mniejszości) oraz ">" (znak większości) są zarezerwowane dla znaczników, nie powinny się one pojawić w normalnej treści strony. Jeżeli musimy ich użyć, należy wpisywać zamiast nich odpowiednio: < oraz >. Ponadto znak "&" (*ampersand* - angielskie "and" - Shift+7) należy zastępować przez: &

Istnieją znaczniki *otwierające* (np.:) oraz *zamykające* (np.:). Zauważ, że znacznik zamykający rozpoczyna się ukośnikiem (czyli znakiem: "/") i ma taką samą nazwę jak otwierający. Pomiędzy znacznikami otwierającym i zamykającym może znaleźć się jakiś tekst, który chcemy np. poddać formatowaniu (w tym przypadku będzie to wytłuszczenie tekstu), np.:

`Ten tekst zostanie wytłuszczony.`

albo

``

Ten tekst zostanie wytłuszczony.

``

(oba powyższe sposoby są równoważne).

Znacznik otwierający musi znaleźć się zawsze przed znacznikiem zamykającym. Jest to chyba dosyć logiczna zasada, ponieważ raczej nie można zamknąć jeszcze nieotwartych drzwi ani otworzyć niezamkniętych... no chyba że mówimy o drzwiach obrotowych ;-). Należy również pamiętać, że prawie każdy znacznik (poza nielicznymi wyjątkami) trzeba zamknąć za pomocą odpowiedniego znacznika zamykającego, co oznacza, że nie można zapominać o wstawianiu znaczników zamykających!

W języku HTML nie ma żadnego znaczenia jakimi literami wpisujemy znaczniki, np. znacznik `<html>` możemy równie dobrze wpisać tak: `<hTmL>`. Przyznasz jednak, że ten drugi sposób jest mało czytelny. Dlatego w tym kursie wszystkie znaczniki będą pisane małymi literami.

UWAGA!

W języku XHTML, będącym odmianą języka HTML, wszystkie znaczniki muszą być pisane obowiązkowo małymi literami. Dlatego zalecam już od początku uczyć się właśnie tej zasady!

Praktycznie każdy edytor HTML oferuje:

- **Kolorowanie składni** polega na tym, że jeśli wpisujemy poprawny znacznik, jego składnia przyjmie określony kolor - zależny od ustawień programu. Dzięki temu od razu uwidaczniają się wszystkie błędy - najczęściej proste "literówki". Jest to chyba najważniejsza zaleta stosowania edytorów HTML. Dlatego właśnie nie poleca się używania zwykłych edytorów tekstu, do tworzenia stron WWW.
- **Automatyczne zamykanie znaczników** polega na automatycznym dopisaniu przez program odpowiedniego znacznika zamykającego, zaraz po tym jak użytkownik poda znacznik otwierający. Dzięki temu, że nie trzeba tego robić ręcznie, praca przebiega dużo sprawniej, a dodatkowo nie zapomnimy o zamykaniu znaczników, ponieważ program robi to automatycznie za nas.
- **Generatory** (lub inaczej kreatory - ang. *wizards*) pomagają przy tworzeniu różnorodnych znaczników (poprzez specjalne okna dialogowe).

Koniec linii

W jaki sposób przenieść tekst do następnej linijki w dokumencie HTML?

`
`

Powyższy znacznik (`
`) stosuje się gdy chcemy natychmiastowo zakończyć linię. Zapytasz zapewne: Po co go stosować, nie można po prostu nacisnąć Enter i przenieść kursor tekstowy do następnej linii? Otóż nie można. Przeglądarka internetowa ignoruje wszelkie znaki przejścia do następnej linii za pomocą klawisza Enter (ignoruje również postawienie obok siebie więcej niż jednej spacji - zobacz: Znaki specjalne). Na przykład jeśli wpiszesz w edytorze taki tekst:

To jest pierwsza linia...
a to jest druga linia.

w przeglądarce pojawi się:

To jest pierwsza linia... a to jest druga linia.

Linia zostanie zakończona automatycznie tylko wtedy, gdy podany tekst będzie zbyt długi, aby zmieścić się w jednym wierszu. Jeśli jednak koniecznie chcemy natychmiast zakończyć linię, możemy to zrobić, stawiając w miejscu gdzie ma być ona zakończona, znacznik `
`. Np. wpisanie w edytorze:

To jest pierwsza linia...

a to jest druga linia.

spowoduje wyświetlenie tekstu:

To jest pierwsza linia...
a to jest druga linia.

Jeśli postawisz obok siebie więcej niż jeden znacznik
, możesz "zejść" kilka linijek niżej, np. wpisanie:

To jest pierwsza linia...

...a to jest kolejna linia.

spowoduje wyświetlenie:

To jest pierwsza linia...

...a to jest kolejna linia.

Zwróć uwagę, że samego znacznika
 nie widać na ekranie przeglądarki. Widać jedynie efekty jego działania, tzn. zakończenie linii.

Znacznik
 w języku HTML nie ma znacznika zamykającego (to jest właśnie jeden z nielicznych wyjątków)!

Natomiast pisząc stronę w języku XHTML, jesteśmy zobowiązani zamknąć każdy znacznik. Jak to zatem zrobić? Zasada jest prosta: wszystkie "tagi", które zgodnie z HTML nie mają znaczników zamykających, w języku XHTML zamykamy stawiając / tuż przed zamknięciem nawiasu ostrego znacznika, np.:
 (zgodnie z HTML wystarczyłoby wpisać:
). Mocno zalecam już od początku trzymać się tej zasady, ponieważ stosując ją, Twoje strony będą zgodne zarówno ze składnią języka HTML, jak i XHTML!

Akapit

W jaki sposób układać tekst na ekranie? Jak dodać nowy akapit? Jak wyśrodkować lub wyjustować tekst? Co to są atrybuty HTML?

<p>Tu wpisz treść akapitu</p>

Akapit (w pewnych warunkach nazywany *paragrafem*) to pewien ustęp w tekście. Następujące po sobie akapity, są rozdzielone linijką przerwy. Treść akapitu należy wpisać pomiędzy znacznikami <p> oraz </p>. Przyjęło się, że praktycznie każdy zwykły tekst na stronie WWW umieszcza się w akapitach. Pojedynczy akapit przedstawia ustęp w tekście, który nieco różni się tematycznie od poprzedniego. Zamiast stosować dwa znaczniki końca linii:

, można po prostu objąć wybrany fragment tekstu paragrafem. Efekt będzie identyczny, a dodatkowo przeglądarka lepiej wyświetli taki tekst. Dzięki temu strona będzie wyglądała estetyczniej i łatwiej będzie można odszukać na niej interesujące informacje.

Akapit (paragraf) jest bardzo ważny w składni HTML, ponieważ pozwala w określony sposób sformatować tekst na ekranie (ułożyć go w podany sposób). Robi się to podając *atrybuty znacznika*. Atrybut wpisuje się zawsze wewnątrz znacznika otwierającego - bezpośrednio po jego nazwie (oddzielony od niej spacją), a przed znakiem zamknięcia nawiasu ostrego, czyli przed ">". Każdy znacznik ma ściśle określone atrybuty, które obsługuje. W przypadku akapitu można zastosować m.in. następujące:

1. Wyrównanie tekstu do lewej strony (domyślnie)

<p style="text-align: left">Treść akapitu</p>

lub po prostu

<p>Treść akapitu</p>

2. Wyrównanie tekstu do prawej

<p style="text-align: right">Treść akapitu</p>

3. Wyśrodkowanie tekstu

<p style="text-align: center">Treść akapitu</p>

4. Justowanie tekstu (wyrównanie do obu marginesów jednocześnie)

<p style="text-align: justify">Treść akapitu</p>

We wszystkich przypadkach **wyróżnione** zostały właśnie *atrybuty znacznika* wraz z ich *wartościami* (wartość atrybutu podaje się w cudzysłowie po znaku równości).

W miejsce tekstu: *Treść akapitu*, należy wpisać tekst, który ma zostać sformatowany w sposób określony przez parametr.

UWAGA!

W języku XHTML wszystkie atrybuty muszą być pisane obowiązkowo małymi literami. Dlatego zalecam już od początku uczyć się właśnie tej zasady!

Przykład:

style="text-align: **left**",
wyrównanie do lewej,
wyrównanie do lewej (style="text-align: left"),
(style="text-align: left")...

style="text-align: **right**",
wyrównanie do prawej,
wyrównanie do prawej (style="text-align: right"),
(style="text-align: right")...

style="text-align: **center**",
wyśrodkowanie,
wyśrodkowanie (style="text-align: center"),
(style="text-align: center")...

justowanie, czyli wyrównanie do obu marginesów jednocześnie (style="text-align: **justify**"); justowanie, czyli wyrównanie do obu marginesów jednocześnie (style="text-align: justify"); justowanie, czyli wyrównanie do obu marginesów jednocześnie (style="text-align: justify"); justowanie, czyli wyrównanie do obu marginesów jednocześnie (style="text-align: justify"); justowanie, czyli wyrównanie do obu marginesów jednocześnie (style="text-align: justify"); justowanie, czyli wyrównanie do obu marginesów jednocześnie (style="text-align: justify"); justowanie, czyli wyrównanie do obu marginesów jednocześnie (style="text-align: justify"); justowanie, czyli wyrównanie do obu marginesów jednocześnie (style="text-align: justify"); justowanie, czyli wyrównanie do obu marginesów jednocześnie (style="text-align: justify"); justowanie, czyli wyrównanie do obu marginesów jednocześnie (style="text-align: justify")...

Tekst pogrubiony

W jaki sposób pogrubić (wytluszczyć) tekst na stronie WWW?

Tu wpisz tekst

Znacznik ten pozwala pogrubić (wytluszczyć) część tekstu (ang. "bold"), który się wewnątrz niego znajduje. Podobna funkcja jest zwykle dostępna w zwykłych edytorach (np.: WordPad), podczas tworzenia normalnego dokumentu tekstowego.

Przykład:

Ten tekst jest pogrubiony (wytluszczony)

Tekst pochylony

W jaki sposób pochylić tekst na stronie WWW (kursywa)?

`<i>Tu wpisz tekst</i>`

Pozwala napisać tekst pismem pochylonym, czyli kursywą (ang. "italic").

Przykład:

Ten tekst jest napisany pismem pochylonym, czyli kursywą

Tekst podkreślony

W jaki sposób podkreślić tekst na stronie WWW?

`<u>Tu wpisz tekst</u>`

Pozwala podkreślić fragment tekstu (ang. "underline").

Przykład:

Ten tekst jest podkreślony.

Wielkość czcionki

W jaki zmienić rozmiar czcionki na stronie WWW?

`Tu wpisz tekst`

gdzie jako "rozmiar" należy wpisać:

`xx-small`

Czcionka najmniejsza

`x-small`

Czcionka bardzo mała

`small`

Czcionka mała

`medium`

Czcionka średnia

`Large`

Czcionka duża

`x-Large`

Czcionka bardzo duża

`xx-Large`

Czcionka największa

Przykład:

Czcionka o rozmiarze `xx-small`

Czcionka o rozmiarze `x-small`

Czcionka o rozmiarze `small` (domyślna)

Czcionka o rozmiarze **medium**

Czcionka o rozmiarze **large**

Czcionka o rozmiarze **x-large**

Czcionka o rozmiarze **xx-large**

Kolor czcionki

W jaki sposób zmienić kolor czcionki na stronie WWW?

```
<span style="color: kolor">Tu wpisz tekst</span>
```

gdzie jako "**kolor**" można wpisać:

black		(czarny)
white		(biały)
silver		(srebrny)
gray		(szary)
maroon		(kasztanowy)
red		(czerwony)
purple		(purpurowy)
fuchsia		(fukcja)
green		(zielony)
lime		(limonowy)
olive		(oliwkowy)
yellow		(żółty)
navy		(granatowy)
blue		(niebieski)
teal		(zielonomodry)
aqua		(akwamaryna)

Jeśli za mało Ci tych szesnastu kolorów, zajrzyj na stronę pt.: Kolory.

Edytory HTML posiadają często specjalny selektor kolorów, za pomocą którego można w prosty sposób wybrać barwę o dowolnym odcieniu.

Przykład:

Ten tekst został napisany czcionką koloru czerwonego

Rodzaj czcionki

W jaki sposób zmienić rodzaj czcionki na stronie WWW?

```
<span style="font-family: rodzaj">Tu wpisz tekst</span>
```

lub

```
<span style="font-family: rodzaj1, rodzaj2, rodzaj3...">Tu wpisz tekst</span>
```

gdzie zamiast **rodzaj, rodzaj1, rodzaj2, rodzaj3...** należy wpisać rodzaje czcionek (np.: *Arial, 'Courier New', 'Times New Roman', Verdana i inne*). Jeżeli nazwa czcionki składa się z kilku

wyrazów, to trzeba ją objąć w znaki apostrofu (np.: `Tekst`).

W miejsce kropek można wpisać dalsze rodzaje czcionek rozdzielone przecinkami (same kropki należy usunąć).

Uwaga! Należy być ostrożnym z używaniem tego polecenia, ponieważ jeśli użytkownik oglądający Twoją stronę nie będzie posiadał podanej czcionki w swoim systemie operacyjnym, tekst zostanie napisany czcionką domyślną (w systemie Windows najprawdopodobniej będzie to 'Times New Roman', chociaż to też nie jest pewne). Poza tym nie każda czcionka potrafi zapisać poprawnie polskie znaki. Koniecznie sprawdź rezultat w praktyce!

W przypadku drugiego polecenia zostanie użyta taka czcionka, którą pierwszą w kolejności podawania będzie posiadał użytkownik. Jest to lepszy sposób, gdyż pozwala się zabezpieczyć na wypadek nieposiadania jednego rodzaju czcionki przez użytkownika.

W systemie Windows standardowo dostępne powinny być czcionki: 'Times New Roman', Arial, 'Courier New'. Dodatkowo od jakiegoś czasu z Internet Explorerem dostarczane są: Verdana, Tahoma, 'Trebuchet MS', Georgia. Przy definiowaniu czcionki dobrze jest wykorzystywać te właśnie rodzaje, a także takie które domyślnie występują w innych systemach operacyjnych (np.: Helvetica - podobna do Arial).

Przykład:

Czcionka Arial

Czcionka 'Courier New'

Czcionka 'Times New Roman'

Czcionka Verdana

Łączenie parametrów

W jaki sposób zmienić wygląd tekstu na stronie WWW?

Wszystkie powyższe parametry (atrybuty i znaczniki) dotyczące tekstu można łączyć, np. po wpisaniu w edytorze:

```
<p style="text-align: center"><span style="font-size: large; color: red; font-family: 'Courier New' "><b><i><u>  
To jest jakiś tekst  
</u></i></b></span></p>
```

otrzymamy:

To jest jakiś tekst

Zauważ, że znaczniki zamykamy w kolejności odwrotnej jak je otwieraliśmy. Czyli najpierw zamykamy znacznik, który został otwarty jako ostatni (w naszym przykładzie jest to: `<u>`), a na końcu zamykamy ten znacznik, który otworzyliśmy jako pierwszy (czyli `<p>`).

Dodatkowo wartości atrybutu `style="..."` odnoszące się do tego samego znacznika (w naszym przypadku jest to znacznik ``), można połączyć wypisując je po kolei rozdzielone od siebie znakami średnika (w naszym przypadku są to wartości: `"font-size: large; color: red; font-family: 'Courier New'"`). Kolejność wpisywania zarówno wartości atrybutu jak i znaczników (otwierających) jest dowolna.

Kolor tła oraz tekstu

W jaki sposób zmienić kolor tła oraz kolor tekstu na stronie WWW?

```
<body style="background-color: kolor tła; color: kolor tekstu">  
Tu jest właściwa treść strony  
</body>
```

Zamiast wyrazów "**kolor tła**" oraz "**kolor tekstu**" należy wpisać definicję koloru odpowiednio: tła strony oraz tekstu na całej stronie. Robi się to tak samo jak w przypadku czcionek.

Jeśli chcemy ustalić kolor tła oraz tekstu **na całej stronie** możemy użyć dodatkowych wartości atrybutu dla znacznika <body>. Wartości te ("background-color: ...; color: ...") nie wpisujemy w miejscu właściwej treści strony - tak jak w przypadku znaczników - lecz w atrybucie style="..." wewnątrz znacznika otwierającego <body> (przed właściwą treścią) - tak jak pokazano powyżej - ponieważ nie są one oddzielnymi *znacznikami* tylko *wartościami atrybutu*. Należy również pamiętać, że na stronie może się znajdować tylko jeden znacznik <body>. Wszystkie atrybuty które się do niego odnoszą (jak na przykład powyższy), wpisuje się do tego samego znacznika (otwierającego), a nie wstawia się nowego w innym miejscu strony.

█ Kolor tła oraz tekstu całej strony powinniśmy ustalać zawsze jednocześnie.

Nawet jeśli ustalimy kolor tekstu na całej stronie, możemy go później lokalnie (miejscowo) zmieniać za pomocą polecenia ... (zobacz: Kolor czcionki).

Wstawienie obrazka

W jaki sposób wstawić obrazek (grafikę, zdjęcie) na stronę WWW?

```

```

Zamiast tekstu: "**Tu podaj względną ścieżkę dostępu do obrazka**", należy podać miejsce na dysku, gdzie znajduje się nasz obrazek, który chcemy umieścić na stronie. Jeżeli obrazek znajduje się w tym samym katalogu co strona, na której chcemy go wstawić, wystarczy wpisać tutaj samą nazwę pliku obrazka, nie zapominając przy tym o podaniu rozszerzenia (".jpg" lub ".gif").

Natomiast w miejsce "**Tu podaj tekst alternatywny**" wpisuje się krótką informację, która pojawi się w przypadku, kiedy obrazek nie zostanie wyświetlony (np. jeśli użytkownik wyłączy wyświetlanie grafiki w swojej przeglądarce internetowej).

UWAGA!

Znacznik nie posiada w języku HTML znacznika zamykającego!

Przykład:



Poniżej znajdziesz przykłady poprawnych ścieżek dostępu, dla różnych struktur katalogów na dysku. We wszystkich przypadkach naszym zamiarem będzie wstawienie obrazka o nazwie plik.gif do dokumentu strona.html - drogę "przejścia" zaznaczono kolorem czerwonym i strzałkami. Przy tworzeniu ścieżki

zawsze "startujemy" od `strona.html` i zmierzamy do `plik.gif`. Plik `index.html` stanowi stronę główną, którą się teraz nie zajmujemy.

1. Poprawnie: `plik.gif`

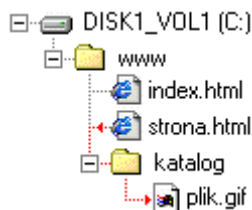
Niepoprawnie: `C:\www\plik.gif`, `plik.GIF`, `plik`



Jak widać obrazek `plik.gif` oraz dokument `strona.html` znajdują się w tym samym katalogu. Nie ma przy tym znaczenia czy oba pliki są w katalogu głównym serwisu czy w jakimś podkatalogu - ścieżka dostępu jest taka sama. W takim przypadku wystarczy podać nazwę pliku bez żadnych dodatków. Jest to najprostsza struktura katalogów (szczególnie pierwsza - wszystko w jednym katalogu). Stosuje się ją szczególnie w serwisach składających się z niewielu podstron. Jeśli masz problemy ze swobodnym tworzeniem bardziej skomplikowanych ścieżek dostępu, polecam Ci właśnie ten model, ponieważ możliwość pomyłki jest tutaj naprawdę niewielka.

2. Poprawnie: `katalog/plik.gif`

Niepoprawnie: `C:\www\katalog\plik.gif`, `katalog\plik.gif`, `katalog/plik.GIF`, `katalog/plik`



W przypadku kiedy `plik.gif` znajduje się w jakimś podkatalogu, a dokument `strona.html` nie, przed nazwą pliku trzeba podać nazwę tego podkatalogu oraz ukośnik (zwykły, nie odwrócony!). Taka struktura jest stosowana np. gdy mamy dużo plików (np. zdjęć) i dla utrzymania porządku chcemy je trzymać w osobnym podkatalogu.

3. Poprawnie: `../plik.gif`

Niepoprawnie: `C:\www\plik.gif`, `katalog/plik.gif`, `..\plik.gif`, `../plik.GIF`, `../plik`



Sytuacja analogiczna jak poprzednio, lecz teraz `plik.gif` jest w katalogu głównym, a `strona.html` w podkatalogu. W takim przypadku przed nazwą pliku trzeba wstawić `../` co oznacza "wyjście" do katalogu nadrzędnego, czyli z `katalog/` do `www/`. Struktura przydatna w przypadku wielu działów w serwisie, z których każdy składa się z dużej ilości podstron umieszczonych w osobnych katalogach.

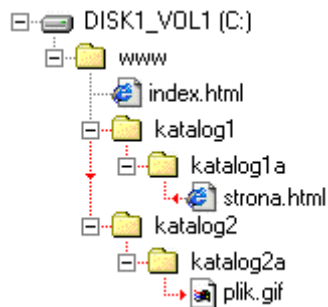
4. Poprawnie: `../katalog2/plik.gif`

Niepoprawnie: `C:\www\katalog2\plik.gif`, `katalog2/plik.gif`, `..\katalog2\plik.gif`, `../katalog2/plik.GIF`, `../katalog2/plik`



Jest to nieco zmodyfikowany poprzedni wariant. Teraz również plik.gif znajduje się w podkatalogu - podobnie jak strona.html z tym, że są to dwa różne katalogi. Budowanie ścieżki w tym przypadku rozpoczynamy w podkatalogu o nazwie katalog1/, w którym znajduje się nasza strona.html. Następnie wychodzimy do katalogu nadrzędnego www/ co jest realizowane przez wstawienie ../ na początku ścieżki dostępu. Teraz możemy swobodnie wejść do drugiego podkatalogu katalog2/ (realizuje się to przez podanie jego nazwy), w którym znajduje się nasz plik.gif, a na końcu - jak zawsze - podajemy nazwę pliku. Struktura przydatna w przypadku wielu działów w serwisie oraz dużej ilości plików. Wtedy w katalogu głównym znajduje się zazwyczaj jedynie strona główna index.html, a wszystkie podstrony oraz pliki są umieszczone w katalogach podrzędnych.

5. **Poprawnie:** ../../katalog2/katalog2a/plik.gif
Niepoprawnie: C:\www\katalog2\katalog2a\plik.gif,
katalog2/katalog2a/plik.gif, ../katalog2/katalog2a/plik.gif



Ten przykład jest już dość zaawansowany. Jeśli go zrozumiesz, możesz powiedzieć, że ścieżki dostępu "masz w małym palcu" :-)

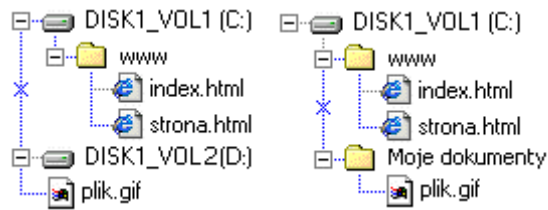
"Znajdujemy się" w katalogu: katalog1a/ (bo tutaj jest nasza strona, na której chcemy wstawić obrazek). Pierwsze dwie kropki w ścieżce dostępu powodują wyjście o jeden poziom wyżej - czyli do katalogu: katalog1/. Kolejne dwie kropki odpowiadają za wyjście o następny poziom wyżej, czyli do katalogu: www/. Teraz możemy już wejść do katalogu podrzędnego: katalog2/, a potem do: katalog2a/, który znajduje się wewnątrz niego. Na końcu wystarczy podać nazwę pliku.

Powyższe sposoby podawania ścieżki dostępu obowiązują dla wszystkich rodzajów plików - również dla podstron. Częstym błędem popełnianym przez osoby początkujące, jest podanie bezwzględnej ścieżki dostępu, np.:

C:\www\katalog2\katalog2a\plik.gif (BŁĄD!)

Jest to karygodny błąd!!! Po wprowadzeniu strony do Internetu, zostaje ona zapisana na dysku *komputera-serwera* (nie na Twoim). Struktura (wygląd) drzewa katalogów będzie tam z całą pewnością inna niż na Twoim dysku lokalnym. Dlatego podana ścieżka bezwzględna tam nie istnieje. Co gorsze, błędu możesz nie wykryć od razu, ponieważ na Twoim komputerze obrazek wczyta się poprawnie, ale już znajomy nic nie zobaczy! Zauważ również, że w ścieżkach używamy ukośników ("/"), a nie odwróconych ukośników ("\").

Jest jeszcze jeden rodzaj poważnego błędu:



Obie ścieżki wyglądają poprawnie, ale są absolutnie niedopuszczalne! Nie wolno wstawiać do dokumentu plików z innego dysku. W ogóle nie można się odwoływać do plików, które znajdują się "powyżej" katalogu głównego serwisu - u nas jest to `www/` (katalog główny serwisu można rozpoznać po tym, że bezpośrednio w nim znajduje się strona główna `index.html`).

Pamiętaj: stosuj zawsze względne, a nie bezwzględne ścieżki dostępu! Używaj również zwykłych, a nie odwróconych ukośników. Wszystkie pliki Twojej strony powinny znajdować się w obrębie katalogu głównego serwisu, w żadnym razie powyżej niego!

Niektóre edytory HTML umożliwiają automatyczne wstawienie względnej ścieżki dostępu (dopiero po zapisaniu strony na dysku), poprzez wybranie pliku w specjalnym selektorze. Dzięki temu tworzenie zawiłych ścieżek, może być dużo łatwiejsze.

Przy okazji zwracam uwagę, że strona startowa `index.html` (patrz powyżej) musi znajdować się w głównym katalogu publicznym na serwerze. Podczas umieszczania strony w Internecie, na serwer wysyłamy wszystkie pliki znajdujące się w katalogu `www/`, ale nie sam katalog. `index.html` musi bezwzględnie znajdować się w katalogu głównym na Twoim koncie WWW, w przeciwnym razie Twoja strona w ogóle się nie wyświetli! Podczas umieszczania strony w Internecie, nie zapomnij przekopiować na serwer również wszystkich plików obrazków!

UWAGA!

Bardzo ważne: w nazwach plików lepiej nie używać:

- wielkich liter
- znaków, np.: \ / : * ? " < > |
- spacji (jeśli musisz, w zamian używaj podkreślnika "_")
- polskich liter (ą, ć, ę, ł, ń... itd.)

Jeśli nie zastosujesz się do tych zaleceń, może się okazać, że po wprowadzeniu strony do Internetu, pliki takie nie wczytają się (nawet jeśli na Twoim komputerze wszystko jest w porządku)!

Na koniec jeszcze jedna rada: obrazki wstawione na stronę, wydatnie zwiększają jej objętość, a co za tym idzie, wydłużają czas wczytywania strony. Jeśli na Twoim dysku lokalnym strona wczytuje się błyskawicznie, nie myśl, że będzie tak samo po wprowadzeniu jej do Internetu. Oczywiście im większy jest obrazek, tym więcej miejsca zajmuje na dysku. Poza tym zbyt wiele nieprzemysłanej grafiki tylko zaciemnia treść. Dlatego staraj się z tym nie przesadzać. Należy również pamiętać, aby najlepiej wszystkie obrazki były zapisane w jednym z trzech formatów:

- **GIF** - dla rysunków składających się z mniej niż 256 kolorów, np. przyciski menu i inne "ręcznie" rysowane ozdobniki. Format ten obsługuje przezroczystość.
- **JPG** - dla zdjęć wielokolorowych. Nie obsługuje przezroczystości.
- **PNG** - pozwala zapisywać z bardzo dobrą jakością zarówno ręcznie rysowane grafiki jak i zdjęcia. Obsługuje kompresję oraz zarówno pełną przezroczystość jak i półprzezroczystość. Niestety zwłaszcza ta ostatnia własność może nie być interpretowana przez starsze przeglądarki (np. *MSIE [Microsoft Internet Explorer] 6.0*).

gdzie zamiast: **"Tu podaj względną ścieżkę dostępu do obrazka"** należy podać lokalizację na dysku, gdzie znajduje się żądany obrazek, a w miejsce **"Tu podaj tekst alternatywny"** wpisuje się krótką informację, która pojawi się w przypadku, kiedy obrazek nie zostanie wyświetlony.

Polecenie pozwala wyśrodkować obrazek, czyli ustawić go na środku ekranu.

Przykład:



Odsyłacz do podstrony

Do czego służą odsyłacze (hiperłącza, linki, odnośniki hipertekstowe)?

(w obrębie tego samego serwisu)

Praktycznie na każdej stronie WWW spotyka się odsyłacze (inaczej: odnośniki lub hiperłącza). Najczęściej stanowi je specjalnie wyróżniony krótki tekst (lub obrazek), po kliknięciu którego, następuje przeniesienie do innej strony. Tak jak każda książka składa się z rozdziałów, tak samo serwisy internetowe składają się zwykle z podstron. Każda podstrona jest osobnym plikiem HTML (z rozszerzeniem *.html lub *.htm) i zawiera treść, która dość znacznie różni się tematycznie od pozostałych (tworzy się ją w taki sam sposób jak stronę główną). Taka organizacja ułatwia użytkownikowi nawigację oraz zapobiega niepotrzebnemu wczytywaniu całego serwisu od razu (co trwałoby prawdopodobnie bardzo długo). Umieszczenie wszystkiego w jednym pliku (index.html), absolutnie nie jest wskazane (chyba że Twoja strona jest naprawdę krótka)!

Aby umożliwić użytkownikowi swobodne przechodzenie pomiędzy podstronami naszego serwisu, trzeba je w jakiś sposób połączyć między sobą. Należy umieścić w wybranym miejscu odsyłacze do wszystkich podstron. Zwykle przeznaczają się na to osobną stronę, która stanowi spis treści i zawiera ułożone kolejno odsyłacze do wszystkich podstron serwisu.

Składnia odsyłacza do podstrony (w obrębie tego samego serwisu) jest następująca:

```
<a href="względna ścieżka dostępu do podstrony">opis odsyłacza</a>
```

Zasady wpisywania **ścieżki dostępu** są takie same jak w przypadku obrazków.

Natomiast zamiast: **"opis odsyłacza"**, należy wpisać krótki tekst, który pojawi się na ekranie i po kliknięciu którego nastąpi przejście do podanej podstrony (adres strony jest widoczny tylko w pasku statusu przeglądarki).

||| Za pomocą tego typu odsyłacza można się odnosić do dowolnego pliku znajdującego się na koncie FTP
razem ze stroną. Dzięki temu można zbudować dział *download*, tzn. odsyłacze do pobierania plików.

Przykład:

Aby zobaczyć przykład działania takiego odsyłacza możesz kliknąć tutaj (nastąpi przejście do strony głównej tego kursu).

Odsyłacz do adresu internetowego

Jak wstawić link (odsyłacz, hiperłącze, odnośnik hipertekstowy) na stronie WWW?

(w obrębie całego Internetu)

Użytkownika można odsyłać również do cudzych stron, a nie tylko do własnych i nie jest to w żaden sposób łamanie praw autorskich (jeśli nie podpiszemy się jako autorzy takiej strony :-)) - przeciwnie - jest to darmowa reklama dla strony, do której następuje odwołanie.

Składnia odsyłacza do adresu internetowego (w obrębie całego Internetu) jest następująca:

```
<a href="adres internetowy">opis odsyłacza</a>
```

Zasady wpisywania "**opisu odsyłacza**" są takie same jak w przypadku odsyłacza do podstrony. Natomiast jako: "**adres internetowy**" należy wpisać adres miejsca w Internecie, gdzie chcemy się przenieść (np.: <http://www.onet.pl>).

Adres strony internetowej zawsze musi rozpoczynać się od <http://> lub <https://>

Odsyłacz taki jest przydatny, jeśli chcemy umieścić w swoim serwisie tzw. linki do dowolnych stron w obrębie całego Internetu. Jeśli surfując w sieci, trafimy na jakąś ciekawą stronę, możemy skopiować jej adres (z paska adresu przeglądarki) i umieścić w odsyłaczu. Dzięki temu wszyscy użytkownicy odwiedzający naszą witrynę, będą mogli przenieść się bezpośrednio do wskazanej strony, bez potrzeby wpisywania jej adresu.

Przykład:

Aby zobaczyć przykład działania takiego odsyłacza możesz kliknąć tutaj (nastąpi przejście do strony głównej portalu internetowego **onet.pl**).

Odsyłacz pocztowy

Jak wstawić adres e-mail na stronie WWW?

```
<a href="mailto:adres poczty e-mail">opis odsyłacza</a>
```

Zamiast: "**adres poczty e-mail**" należy wpisać adres poczty elektronicznej (np.: jan_kowalski@example.com).

Ten odsyłacz jest przydatny, gdy pragniesz umieścić na stronie adres swojej skrzynki poczty elektronicznej (e-mail). Dzięki temu będziesz w stałym kontakcie z internautami odwiedzającymi Twoją stronę.

Pamiętaj, że podanie swojego adresu na stronie WWW, może spowodować, że zaczną do Ciebie przychodzić niechciane wiadomości - reklamówki (tzw. *spam*). Dobrym pomysłem może być założenie sobie darmowego konta pocztowego w dowolnym portalu internetowym i podanie adresu takiego konta na swojej stronie - darmowe konto zawsze można zmienić. Zakładając stronę WWW na darmowym serwerze, często dostajemy również konto e-mail. Wtedy można z niego skorzystać.

Przykład:

Jako przykład użycia takiego odsyłacza możesz wysłać list do mnie (tylko proszę, jeśli nie masz nic konkretnego do napisania, nie przysyłaj mi pustych listów :-))

Wyślij list do autora kursu HTML

Odsyłacz obrazkowy

W jaki sposób wstawić na stronie WWW odnośnik (link, hiperłącze, odsyłacz hipertekstowy) obrazkowy (graficzny), czyli klikalny przycisk?

Wszystkie odsyłacze przedstawione do tej pory, miały postać tekstową. Jeśli chcesz umieścić w swoim serwisie np. menu z obrazkowymi przyciskami, należy w tym celu użyć następującego polecenia:

gdzie jako "adres" można podać:

- względną ścieżkę dostępu do dowolnej podstrony Twojego serwisu (np.: `index.html`),
- adres internetowy poprzedzony przez "http://" (np.: `http://www.onet.pl`),
- adres poczty elektronicznej poprzedzony przez "mailto:" (np.: `mailto:jan_kowalski@example.com`).

Natomiast zamiast: "Tu podaj względną ścieżkę dostępu do obrazka" należy podać lokalizację na dysku, gdzie znajduje się żądany obrazek, a w miejsce "Tu podaj tekst alternatywny" wpisuje się krótką informację, która pojawi się w przypadku, kiedy obrazek nie zostanie wyświetlony.

Jak widać odsyłacza obrazkowego możemy użyć w połączeniu z dowolnym typem odnośników (do podstrony, do adresu internetowego lub pocztowy). Jednak najczęściej w ten sposób tworzy się menu nawigacyjne serwisu (odsyłacze do podstron).

Obrazki przycisków najlepiej zapisywać w formacie **GIF**. Jeśli nie masz zacięcia artystycznego, nie musisz samodzielnie rysować wszystkich grafik. W Internecie na pewno znajdziesz wiele stron, gdzie możesz darmowo pobrać gotowe przyciski.

Przykład:

Kliknij na tym przycisku: 

Jak zrobić dobrą stronę

Czego unikać, aby Twoja strona WWW nie odstraszała internautów?

Na koniec chciałbym poruszyć bardzo ważną sprawę, o której wiele osób zdaje się zapominać. Znajomość podstaw HTML w zupełności wystarcza do napisania strony internetowej, lecz aby stworzyć dobry serwis, który będzie często odwiedzany i którym będzie się można naprawdę pochwalić, trzeba czegoś więcej:

- I. Zastanów się dobrze nad tematyką swojej strony. Oczywiście nikt nie zabroni Ci publikacji informacji o tym, że np. Twój pies wabi się Azor, a kot - Mruczek, ale czy coś takiego naprawdę może przyciągnąć czytelników? Nie zapominaj, że część osób łączy się z Internetem ze smartfonu czy tabletu, płacąc za ilość danych pobranych z sieci, więc czy ktoś będzie tracił czas na czytanie takich informacji? Postaraj się znaleźć temat, który Cię interesuje, jest oryginalny i może zaciekać innych. Zwykle lepsze rezultaty przynosi skupienie się na jednej dziedzinie, niż opisywanie wszystkiego po trochu.
- II. Uważaj z doбором kolorów na stronie, zwłaszcza jeśli chodzi o tło oraz tekst. Nieciekawe lub rażąco złe zestawienie barw jest najważniejszym czynnikiem, który zdradza, że autor serwisu jest kompletnym amatorem. Wyobraź sobie co pomyślą np. klienci jakiejś firmy, która posiadałaby stronę z zielonym tłem i czerwonym tekstem (lub co gorsze odwrotnie)! Unikaj jak ognia jaskrawych odcieni w tle. Jeśli nie masz pomysłu, ustal białe tło i czarny tekst - tak jest najbezpieczniej. Dobre rezultaty dają również barwy całkiem ciemne lub delikatne pastelowe (jasne). Tło w żadnym razie nie może pogarszać czytelności strony, dlatego należy pamiętać o doborze odpowiedniego koloru tekstu.
- III. Nie przesadzaj z grafiką. Fakt, że wiesz jak wstawić na stronę obrazek nie oznacza, że musi się od nich roić. Pamiętaj, że to właśnie grafika najbardziej wpływa na szybkość ładowania strony, dlatego ogranicz jej ilość oraz rozmiary. Jeśli już musisz coś umieścić, postaraj się, aby było to naprawdę dobrej jakości. Tworząc samodzielnie obrazy używaj *antialiasingu*, czyli wygładzania poszarpanych krawędzi. Cechą po której na pierwszy rzut oka można rozpoznać dobrego projektanta jest oszczędne i umiejętne operowanie grafiką, która dobrze komponuje się z wyglądem strony.

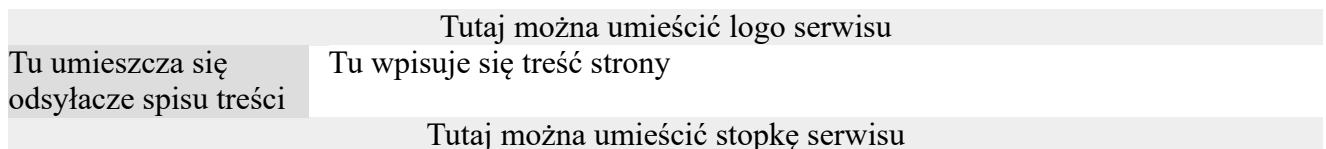
IV. Przyjrzyj się innym renomowanym witrynom. Zwróć uwagę na projekt szaty graficznej. Dużo nie zawsze znaczy dobrze. Czasem bardziej oszczędna forma wygląda lepiej, a przede wszystkim czytelniej, niż wyszukane ozdobniki. Najważniejszy jest dobry pomysł. Pamiętaj, że kolorystyka witryny oraz projekt oprawy graficznej to pierwsze rzeczy, na które zwracają uwagę użytkownicy. Jeśli się nie postarasz, Twoja strona już na wstępie będzie mówiła: "Tutaj nie ma nic godnego uwagi, możesz opuścić to miejsce". Większość dobrych stron w Internecie zbudowana jest następująco: na górze znajduje się banner lub logo serwisu, po lewej lub po prawej - menu, a obok - treść strony. Nie znaczy to, że musisz koniecznie podporządkować się takiemu kanonowi, lecz jeżeli chcesz uzyskać podobną strukturę, we właściwej treści strony (BODY) wpisz np.:

```
<header style="text-align: center">Tutaj można umieścić logo serwisu</header>
<nav style="width: 300px; float: left; overflow: hidden">
Tu umieszcza się odsyłacze spisu treści
</nav>
<article style="margin-left: 320px">

Tu wpisuje się treść strony

</article>
<footer style="text-align: center; clear: both">Tutaj można umieścić stopkę
serwisu</footer>
```

a otrzymamy:



V. Nawigacja w serwisie to bardzo ważna sprawa. Nieczytelny układ strony, menu w którym trudno cokolwiek znaleźć i inne podobne historie są częstą przyczyną zniechęcenia użytkowników, którzy w konsekwencji opuszczają witrynę. Przy projektowaniu systemu odsyłaczy spróbuj wczuć się w sposób myślenia internautów. Pamiętaj, że nie znają oni serwisu tak dobrze jak Ty i mogą nie chcieć zagłębiać się w szczegóły. Dlatego warto w widocznym miejscu umieścić hiperłącza do najciekawszych działów, jakiś system wyszukiwania lub indeks. W przypadku obszerniejszych witryn wskazane jest podzielenie ich na działy tematyczne. Sposób nawigacji powinien być przede wszystkim wygodny, praktyczny, logiczny i łatwo dostępny.

VI. Dbaj o poprawność ortograficzną i stylistyczną tekstu. Naucz się zasad wpisywania znaków interpunkcyjnych. Jeśli na stronie WWW występuje masa błędów, prawdopodobnie autor nie ma nic ważnego do przekazania.

VII. Postaraj się na bieżąco aktualizować treść strony i usuwać powstałe błędy. Udoskonalaj oprawę graficzną oraz system nawigacji, aby serwis nie stał się nudny. Informuj o wprowadzanych nowościach i utrzymuj kontakt za pośrednictwem wiadomości e-mail z użytkownikami. Otwórz się na konstruktywną krytykę.

‡ Jeżeli masz jakieś zastrzeżenia co do tego serwisu, wiesz jak można zrobić coś lepiej, napisz do mnie

Szablony WWW

Istnieje sposób na zrobienie profesjonalnie wyglądającej strony nawet bez dużego doświadczenia - są to *szablony WWW*. Wiele serwisów internetowych oferuje darmowo gotowe projekty graficzne witryn. Zawierają one całą strukturę dokumentu oraz wszystkie potrzebne grafiki. Jedyne co trzeba zrobić samodzielnie, to uzupełnić szablon własnym tekstem w odpowiednich miejscach. Jeśli nie masz zatem pomysłu na projekt graficzny swojego serwisu, radzę poszukać ciekawego szablonu w sieci. Dzięki temu Twoja strona na pewno nie będzie "straszyła" internautów ;-)

Radzę jednak uważać przy korzystaniu z takich rozwiązań. Niektóre szablony wyglądają nawet bardzo dobrze, ale są tak przeładowane grafiką, że wczytywanie takiej strony może potrwać bardzo długo. Dodatkowo szczególnie jeśli korzystasz z obcojęzycznych szablonów, należy sprawdzić sposób kodowania znaków. W nagłówku każdej strony HTML musi znajdować się linijka:

```
<meta charset="utf-8">
```

(zobacz: Ramy dokumentu). Brak podobnej linijki w nagłówku strony albo istniejące wpisy typu:

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

są poważnym błędem, który spowoduje niepoprawne wyświetlanie polskich znaków w przeglądarce!

Adresy stron z darmowymi szablonami WWW możesz znaleźć w serwisach:

- Tymex.org
- Webmade.org
- Open Source Web Design
- TemplateMo

Podsumowanie

To by było na tyle. Poznane tu polecenia powinny pozwolić Ci na napisanie prostej, ale zarazem dość dobrze wyglądającej strony WWW. **A więc do dzieła...**

Jeżeli będziesz mieć więcej czasu, zajrzyj do dalszych rozdziałów tego kursu. Znajdziesz tam o wiele więcej przydatnych znaczników, dzięki którym napiszesz jeszcze lepszą stronę.

Aby kontynuować **Kurs HTML**, kliknij poniższe linki:

- [Nagłówek i treść](#)
- [Tekst](#)
- [Odsyłacze](#)
- [Tabele](#)
- [Multimedia](#)
- [Widżety](#)
- [Formularze](#)

Dodatkowo w rozdziale: [I co dalej](#) możesz przeczytać szczegółowy opis, jak wprowadzić swoją stronę do Internetu (zupełnie za darmo).

Zapraszam...

Quiz

Sprawdź swoją wiedzę, nabytą w tym rozdziale, rozwiązując testowy **QUIZ** (online).

Ćwiczenia

Teoria bez praktyki jest niczym. Dlatego zachęcam do stałego eksperymentowania z kodem HTML w trakcie czytania kursu. Dzięki temu poznane informacje na pewno zostaną lepiej utrwalone w Twojej pamięci. Ponadto często dopiero na etapie ćwiczeń praktycznych okazuje się, że tak naprawdę nie do końca właściwie zrozumieliśmy teorię. Jednak przede wszystkim to praktyka sprawia, że stajemy się coraz lepsi w danej dziedzinie. To z kolei pobudza i rozwija naszą pasję, dzięki której możemy stać się prawdziwymi

mistrzami - „*praktyka czyni mistrza*”. Dlatego nie zaniedbuj tego etapu nauki. Zainstaluj edytor HTML i zacznij systematycznie ćwiczyć.

W ramach rozwoju umiejętności praktycznych zalecam wykonanie serii ćwiczeń. Będą one polegać na stworzeniu prostej strony internetowej bazującej na poznanych informacjach. W kolejnych rozdziałach będziemy ją stopniowo rozbudowywać, dlatego nie zapomnij zapisać jej sobie gdzieś na komputerze. Na tym etapie opublikowanie strony w internecie nie będzie jeszcze konieczne.

1. Znajdź temat, który szczególnie Cię interesuje. Może to być np. Twoja ulubiona książka, gra komputerowa, zespół muzyczny, dyscyplina sportu itp.

Jeżeli do Twojej strony zamierzasz zaczerpnąć informacje z internetu, pamiętaj o poszanowaniu praw autorskich! Nie zawsze można sobie swobodnie skopiować na swoją stronę dowolny tekst, jeżeli autor nie wyraził na to zgody. Każdy cytat powinien być wyraźnie oznaczony z podaniem jego źródła.

Możesz również stworzyć stronę o sobie, w której opiszesz krótko swoją biografię, zainteresowania, hobby itp.

2. Otwórz edytor HTML. Jeśli korzystasz z edytora graficznego, przełącz się do trybu bezpośredniej edycji kodu źródłowego HTML.
3. W edytorze utwórz nowy plik HTML i wklej w nim szkielet strony uzupełniając konieczne fragmenty.
4. Zacznij wpisywanie właściwej treści strony. Wyniki swojej pracy możesz na bieżąco oglądać w przeglądarce internetowej. W tym celu zapisz plik pod nazwą `index.html`, a następnie odszukaj go na dysku za pomocą dowolnego programu do przeglądania plików (w systemie Windows możesz użyć np. *Eksploratora plików*) i dwukrotnie kliknij ikonę stworzonego właśnie dokumentu. Dzięki temu Twoja strona otworzy się w przeglądarce. Za każdym razem kiedy coś na niej zmienisz, po prostu ponownie zapisz plik w edytorze, a następnie przełącz się do przeglądarki i odśwież stronę (klawisz *F5*), aby zobaczyć efekty.
5. Wróć do edytora HTML. Podziel tekst na akapity, tak aby był wygodniejszy w czytaniu.
6. Wybierz fragmenty tekstu, na które Twoim zdaniem czytelnik powinien zwrócić szczególną uwagę i umieść je w znacznikach pogrubienia.
7. Dodaj do strony ilustracje.

W przypadku zdjęć i grafik należy szczególnie pamiętać o poszanowaniu praw autorskich! Nie każdy obrazek można sobie swobodnie skopiować na swoją stronę, jeżeli autor nie wyraził na to zgody.

Jeśli tworzysz stronę o sobie, może to być po prostu Twoje zdjęcie.

8. Stwórz podstronę o nazwie `linki.html` (nowy plik HTML w edytorze) i umieść do niej odsyłacz na stronie głównej (czyli w pliku `index.html`). O tym co to są i jak się tworzy podstrony przeczytasz w podrozdziale: Ramy dokumentu.
9. Na podstronie `linki.html` wstaw odsyłacze do ciekawych stron w internecie, na których czytelnik będzie mógł znaleźć więcej informacji na opisywany przez Ciebie temat. Jeżeli tworzysz stronę o sobie, możesz tutaj umieścić odnośniki do Twojego profilu w serwisach społecznościowych: Facebook, Twitter, YouTube, LinkedIn itp.
10. Stwórz podstronę o nazwie `kontakt.html` (nowy plik HTML w edytorze) i umieść do niej odsyłacz na stronie głównej (czyli w pliku `index.html`). Wstaw na niej odsyłacz pocztowy z Twoim adresem e-mail.