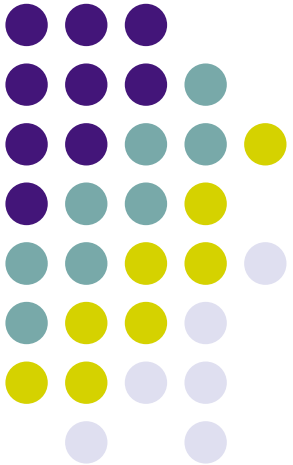
EUROPE DIRECT
Opole

Warsztaty programowania stron www

Część I

Dr inż. Szczepan Paszkiel



Strona www

- Hipertekstowy dokument
- Odczytywana przez przeglądarkę internetową
- Kompatybilność pomiędzy przeglądarkami
- Wtyczki do obsługi elementów

- Konsorcjum W3C

Przeglądarka www

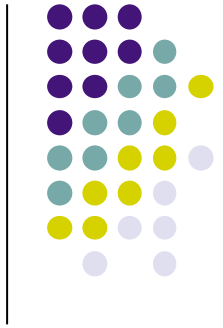
Wyszukiwarka

YAHOO!

Google

Microsoft Bing

Serwery internetowe

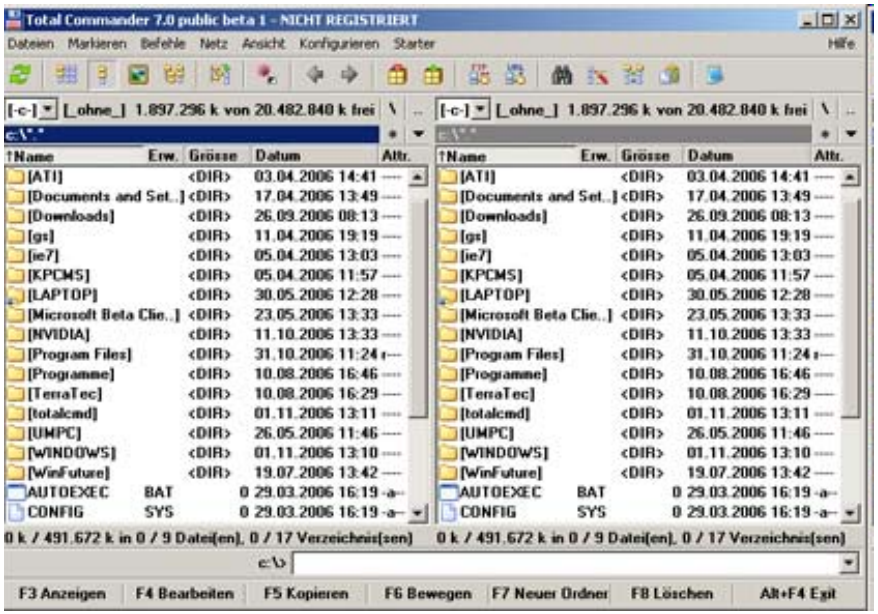


- Komputer – host – serwer FTP
- Web Serwer - jest to program działający na serwerze internetowym, obsługujący żądania protokołu komunikacyjnego HTTP
- Apache, IIS

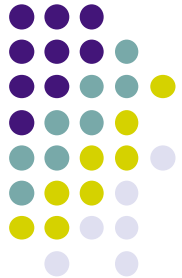
- HTTPS, FTP, SMTP, POP3, IMAP

Klient FTP

- Total Commander
- TurboFTP
- AbleFTP
- FileZilla
- Konqueror

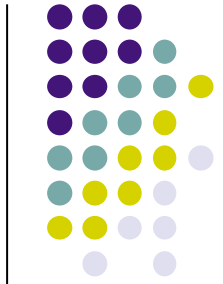


Aplikacje do tworzenia witryn www



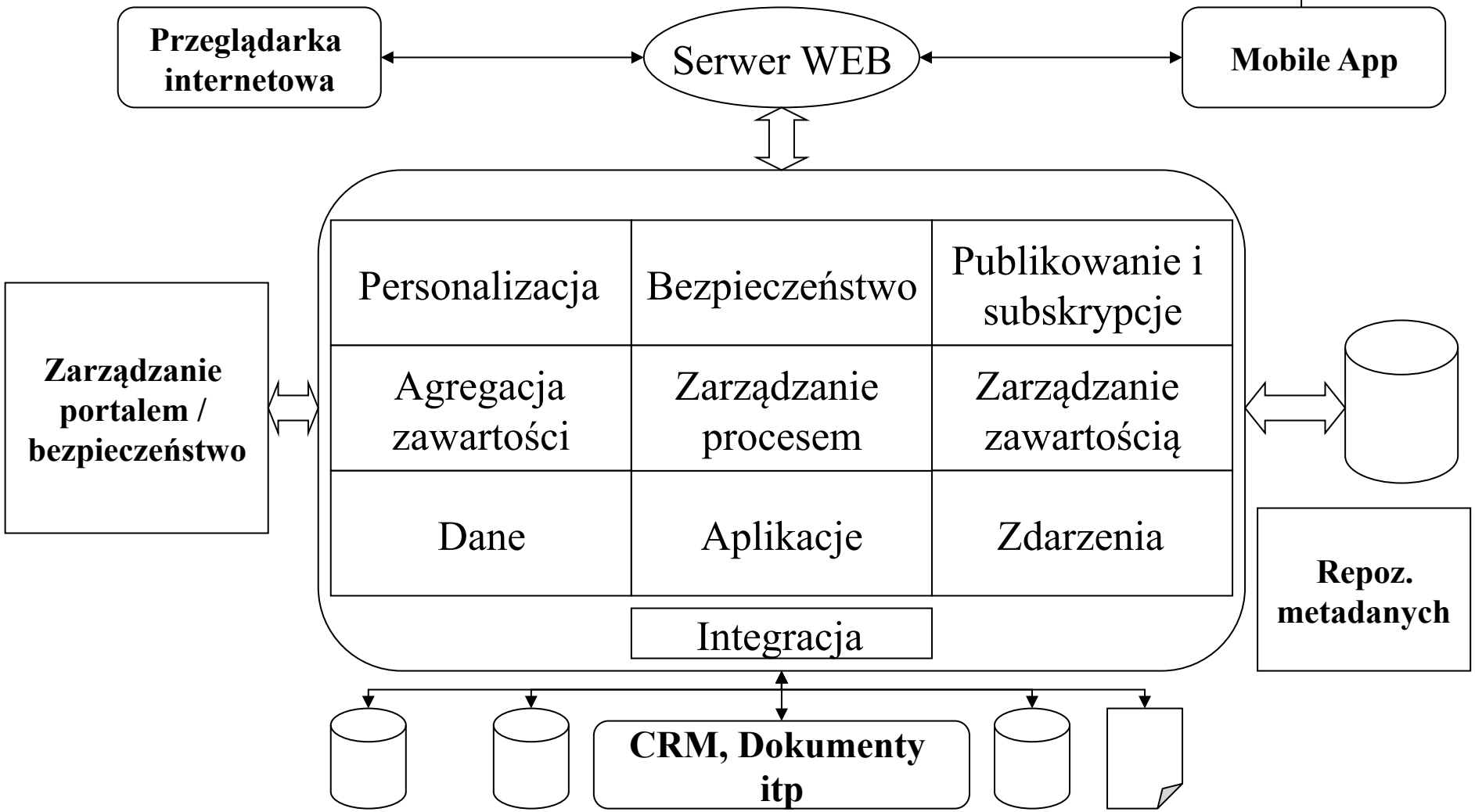
- Microsoft Expression Web
- Adobe Dreamweaver
- Pajączek NxG Professional
- Notepad2
- Adobe Flash Builder
- ConTEXT

Usługi internetowe



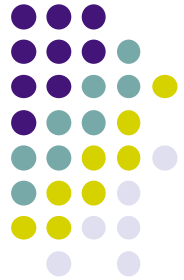
- WWW – przeglądarki, odsyłacze hipertekstowe
- Poczta elektroniczna
- Grupy dyskusyjne – tablica ogłoszeń
- FTP
- Komunikatory
- E-bank – szyfrowanie
- Wideokonferencje

Schemat budowy serwisu



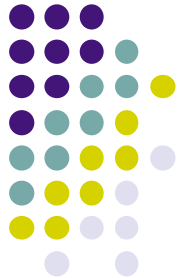
Na podstawie mat. Firmy **SYBASE**

Wymagania stawiane serwisom internetowym



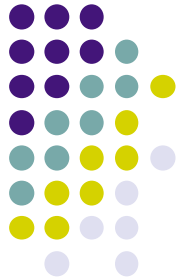
- skalowalność
- ciągłość działania (24/7)
- indeksowanie i wyszukiwanie
- zarządzanie zawartością
- bezpieczeństwo
- kategoryzacja
- indywidualizacja
- różnorodność repozytoriów
- stworzenie jednolitego środowiska pracy

Sposoby publikacji danych w Internecie



- Statyczne strony WWW
  - robimy kilka stron dla różnych przypadków dodajemy pliki z grafiką wrzucamy na serwer WWW, dobre do małych rozwiązań
- strony dynamiczne (technologie JS, PHP)
  - stworzenie schematu wyglądu stron; strony generowane są na podstawie schematu i danych pobieranych z bazy danych (pojawia się możliwość swobodnej zmiany treści bez ingerencji w pliki html)
- publikacje oparte o systemy zarządzania treścią
  - jw. + wiele innych cech czyli zarządzanie schematami, projektami

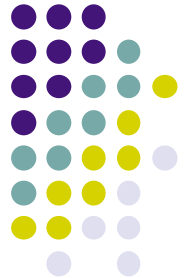
Hiperłącza



- Odnośnik, link to zamieszczone w dokumencie elektronicznym odwołanie do innego dokumentu lub innego miejsca w danym dokumencie.

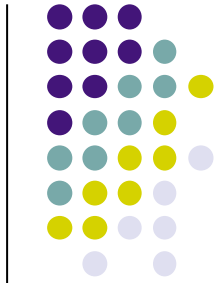
- Odwołanie takie związane jest z fragmentem tekstu lub obrazem – uaktywnienie hiperłącza (kliknięcie lub nadejście odpowiedniego momentu) powoduje otwarcie dokumentu docelowego.

Technologie tworzenia stron



- HTML/DHTML - CSS
- PHP+MySQL
- ASP <-> XML – język znaczników
- JSP – oparte na języku JAVA – dynamiczne witryny
- Java Script
- Applety JAVA
- Adobe Flash

Systemy CMS

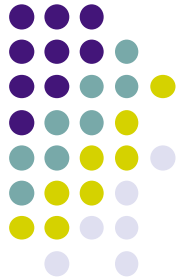


- **System zarządzania treścią** - *Content Management System*

Jest to aplikacja internetowa, pozwalająca na łatwe utworzenie serwisu WWW oraz jego późniejszą aktualizację i rozbudowę przez redakcyjny personel nietechniczny.

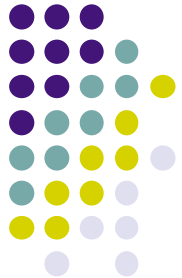
Kształtowanie treści i sposobu ich prezentacji w serwisie internetowym zarządzanym przez CMS odbywa się za pomocą prostych w obsłudze interfejsów użytkownika, zazwyczaj w postaci stron WWW zawierających rozbudowane formularze i moduły.

Domeny



- **.com** – komercyjne
- **.edu** – edukacja, szkolnictwo
- **.gov** – rządowe, polityczne
- **.net** – sieciowe
- **.org** – organizacje
- **.int** – organizacje międzynarodowe (od 1988)
- **.mil** – militarne

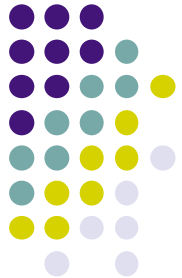
DNS Domain Name System



- Zapewnia przyjazność adresów internetowych dla człowieka

- Umożliwia lokalizację maszyny (adres logiczny) na podstawie znakowej nazwy.

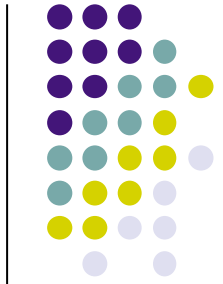
Adresy URL



- Jest to stosowany w Internecie system adresowania, pozwalający wskazać dowolny dokument.

- URL jest czymś w stylu ścieżki dostępu do pliku występującej np. w systemach operacyjnych MS-DOS czy Unix.

Adres URL



- Adres dokumentu HTML powinien być budowany według następującego wzorca:

http://[domena]/[kategoria strony]/[nazwa strony]

gdzie:

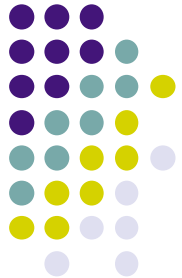
- *[domena]* - domena serwisu,
- *[kategoria strony]* – nazwa kategorii nadrzędnej serwisu zapisana w sposób przyjazny dla wyszukiwarek, w przypadku krótkich nazw kategorii można umieścić większą liczbę kategorii nadrzędnych,
- *[nazwa strony]* – przyjazna dla wyszukiwarek nazwa przeglądanego dokumentu HTML.

Czego należy unikać tworząc URL?

Budując adresy URL łatwo jest przedobrzyć. Aby tego uniknąć należy unikać:

- tworzenia długich adresów URL (mających 100-200 znaków),
- tworzenia adresów zawierających wiele niepotrzebnych parametrów, w tym te odpowiedzialne za identyfikację sesji lub użytkownika,
- wykorzystywania generycznych nazw dla stron, takich jak *strona1.html*,
- tworzenia zbyt opisowych nazw stron, na przykład: *strona-poswiecona-tematyce-ogolnorozwojowej-z-nastawieniem-na-to-i-to-i-to.html* .

Podstawy języka HTML

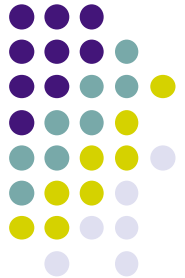


- Dokument HTML jest zwykłym plikiem tekstowym, w którym znajdują się polecenia HTML.

- Wynika stąd, że dokument taki można utworzyć za pomocą najprostszego edytora tekstów, ręcznie dodając znaczniki.

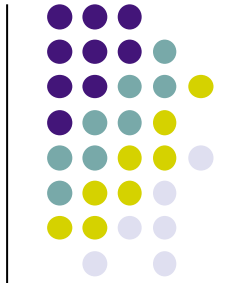
- index.html
- index.htm

Zasady tworzenia stron www



- Stosuj się do standardów
- Nie nasycaj strony zbyt wielką ilością grafiki
- Ostrożnie wykorzystuj cudzą grafikę
- Podawaj datę aktualizacji serwisu
- Nie bądź anonimowy
- Wstaw licznik odwiedzin
- Poinformuj innych o swoich stronach

Struktura strony www



<HTML>

<HEAD> informacje nagłówkowe </HEAD>

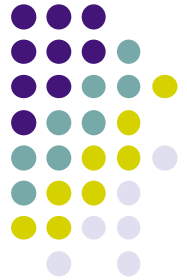
<BODY>

właściwa treść (ciało) dokumentu

</BODY>

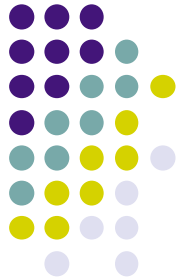
</HTML>

Znacznik <HEAD>



```
<HEAD>
<meta http-equiv="content-type" content="text/html;
    charset=iso-8859-2">
<TITLE>Tytuł strony</TITLE>
</HEAD>
```

Znacznik <BODY>

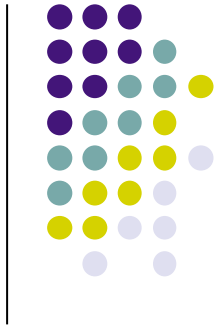


<BODY BGCOLOR="kolor">
<BODY BACKGROUND="URL albo ścieżka/nazwa_pliku.gif">

<P>To jest treść pierwszego akapitu</P>
<P ALIGN=LEFT> </P>
<P ALIGN=RIGHT> </P>
<P ALIGN=CENTER> </P>
<P ALIGN=JUSTIFY> </P>

<!-- ... -->

Znacznik <BODY>



To jest pierwszy wiersz
To jest drugi wiersz

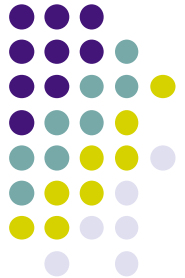
<HR> - pozioma linia

<HR WIDTH=300>
<HR ALIGN=center>
<HR COLOR="nazwa_koloru">

<H1>…</H1>
…..
<H5>…</H5>

<CENTER> </CENTER>

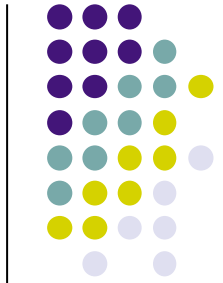
Wykazy



```
<UL>
 <LI>Uważam, że UFO istnieje</LI>
 <LI>Uważam, że pozaziemskie cywilizacje mogą
 nam pomóc</LI>
 <LI>Uważam, że należy się przygotować na
 spotkanie</LI>
 </UL>
```

```
<UL TYPE=disc>
<UL TYPE=circle>
<UL TYPE=square>
```

Formatowanie tekstu



<B>pogrubiony (bold)</B>
<I>pochylony (italic, kursywa)</I>
<U>podkreślony (underlined)</U>
<BLINK> migający napis </BLINK>
<STRIKE> przekreślony napis </STRIKE>
<MARQUEE>Tekst animacji</MARQUEE>

<FONT SIZE="x">Tekst objęty definicją</FONT>
x = 1...7

<FONT COLOR="kolor">Tekst objęty poleceniem</FONT>

Linki wewnętrzne oraz zewnętrzne

<A HREF="adres.strony.internetowej">Jakaś nazwa tej strony</A>

<A HREF="http://host.domena">Tekst</A>

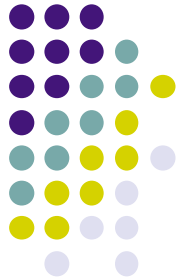
<A HREF="linki.html">Linki</A>

<A HREF="plik_tekstowy.txt">Plik tekstowy TXT</A>

<A HREF="plik_graficzny.gif">Plik graficzny GIF</A>

<A HREF="mailto:autor@jego.adres">Imię i nazwisko</A>

Grafika na stronie www

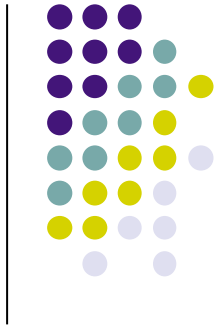


<IMG SRC="nazwa_pliku">

<IMG SRC="tucows.gif" WIDTH=200 HEIGHT=50>

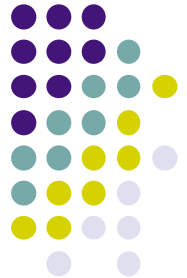
<IMG SRC="tucows.gif" WIDTH=120 HEIGHT=160
BORDER=5>

I strona www



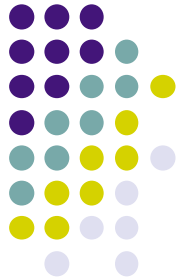
- Notepad2

Kaskadowe arkusze stylów



- CSS (*Cascading Style Sheets*) jest to język opisujący sposób, w jaki przeglądarki mają wyświetlać zawartość odpowiednich elementów HTML.
- Kaskadowe arkusze stylów służą do definiowania sposobu wyświetlania elementów HTML. Pozwalają określać rozmiar i kolor czcionki, definiować odstępy i rozmieszczenie tekstu oraz obrazów na stronie, a pełna lista ich możliwości jest znacznie dłuższa.
- Znaczniki HTML zostały pierwotnie zaprojektowane jako narzędzia definiowania zawartości dokumentu. I tak znacznik nagłówka określał: „To jest nagłówek", znacznik akapitu stwierdzał: „To jest akapit tekstu", znacznik tabeli informował: „To jest tabela", a o układzie strony decydowała przeglądarka.
- Wraz z rozbudową możliwości przeglądarek zaczęły pojawiać się coraz to nowe znaczniki i atrybuty. Tworzenie stron WWW, których zawartość byłaby dobrze odseparowana od układu dokumentu, stawało się coraz trudniejsze.

Kaskadowe arkusze stylów



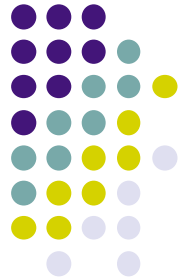
Co daje stosowanie arkuszy CSS?

Podstawowe zalety stylów CSS to możliwość szybkiej i prostej modyfikacji stylu oraz błyskawiczna wręcz aktualizacja postaci dokumentu w przypadku takich zmian. One naprawdę zaoszczędzą Twój czas! Wyobraź sobie, że w rozbudowanej witrynie trzeba zmienić sposób formatowania dziesiątków nagłówków czy połączeń! To wiele godzin pracy, jeśli będziesz ręcznie wyszukiwał elementy i zmieniał ich atrybuty, lub kilka chwil, jeśli zastosujesz w dokumencie arkusze CSS – wówczas modyfikacja stylu to parę stuknięć w klawisze, a aktualizacja to automat.

Czym są kaskadowe arkusze stylów?

Kaskadowy arkusz stylów to mechanizm definiujący formatowanie dla danej strony przy zastosowaniu stylów złożonych, które przeglądarka zinterpretuje w określonym porządku zwanym kaskadą.

Rodzaje arkuszy stylów



Definicja stylu może pojawić się w konkretnym elemencie HTML – wówczas mówimy o **stylu wpisanym**,

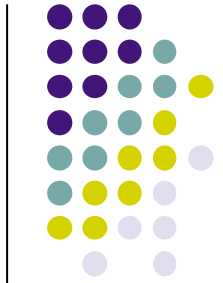
w obrębie elementu `head` strony HTML (to znaczy między znacznikami `<head> </head>`) – takie arkusze stylów nazywa się **osadzonymi**,

lub może zostać pobrana z pliku zewnętrznego – jest to wtedy **zewnętrzny lub łączony arkusz stylów**.

Wszystkie typy arkuszy CSS – **wpisane, osadzone** i **zewnętrzne** – można stosować jednocześnie.

Łączone arkusze stylów są przechowywane w zewnętrznym pliku o rozszerzeniu nazwy .**css**. Składnia takiego arkusza jest podobna jak w przypadku arkusza osadzonego, a sformatowanie strony wymaga jedynie umieszczenia połączenia do pliku zawierającego definicję stylu.

Jak działa kaskada stylów?



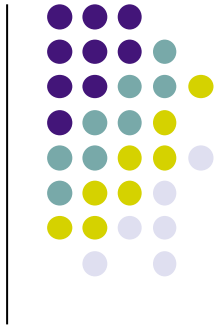
Jaki styl zostanie zastosowany, jeśli w dokumencie zdefiniowano kilka arkuszy stylów ?

W przypadku obecności różnych arkuszy stylów na jednej stronie, hierarchia ich ważności rośnie w następującej kolejności:

1. Domyślne ustawienia przeglądarki
2. Zewnętrzny arkusz stylów
3. Osadzony arkusz stylów (umieszczony między znacznikami `<head> </head>`)
4. Styl wpisany (dotyczący konkretnego elementu HTML)

Najwyższy priorytet ma styl wpisany, co oznacza, że jego ustawienia są dominujące względem zadeklarowanych w sekcji `head` oraz pobranych z pliku zawierającego zewnętrzny arkusz stylów. Dominują także nad domyślnymi ustawieniami przeglądarki.

Ogólna postać CSS



Postać arkusza stylu CSS zależy od typu arkusza. W przypadku stylu wpisanego – a więc umieszczonego w konkretnym znaczniku – ma ona taką oto postać:

```
<znacznik style="właściwość: wartość;">
```

Ogólna postać osadzonego arkusza CSS jest następująca:

```
<style type="text/css">
<!--
selektor {właściwość: wartość;}
-->
</style>
```

Zawarta w obrębie elementu style definicja następującą składnię:

```
selektor{właściwość: wartość}
```

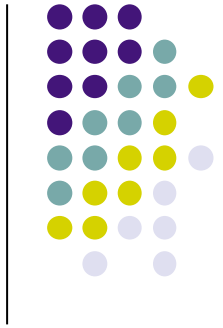
Selektorem nazywa się znacznik czy też element, który chcesz zdefiniować, **właściwość** to jego atrybut, który zmieniasz przypisując mu nową **wartość**. Właściwość i wartość rozdzielone są dwukropkiem oraz zawarte w nawiasach klamrowych, tak jak w tym przykładzie:

```
body {color: black}
```

Jeśli wartość ma postać wielowyrazową – na przykład `sans serif` – umieszcza się ją w cudzysłowie:

```
p {font-family: "sans serif"}
```

Ogólna postać CSS



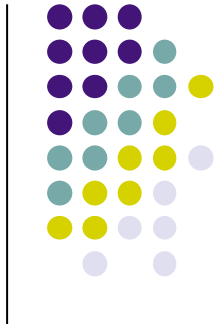
Oto przykład rzeczywistej definicji:

```
<style type="text/css">
<!--
body {font-family: Verdana, Arial, Helvetica;}
-->
</style>
```

Między znacznikami `<style>` i `</style>` umieszczana jest lista znaczników HTML wraz z właściwościami arkusza, które je definiują. Definicja zaprezentowana powyżej zawiera tylko jeden symboliczny selektor definiowany przez jedną właściwość CSS. W przypadku większej liczby właściwości, wszystkie muszą być umieszczone w nawiasie klamrowym `({})` oraz oddzielone średnikami `(;)`.

Zwróć uwagę na **znaczniki komentarza html**, obejmujące wszystkie znaczniki i właściwości CSS – dzięki temu, że zostały zastosowane, przeglądarki, które nie potrafią obsługiwać arkuszy stylów nie wyświetlą śmieci, lecz potraktują kod jako komentarz. Te z kolei, którym arkusze CSS nie są obce, wiedzą, że między znacznikami zawarte są definicje stylów.

Ogólna postać CSS



Co oznacza ta definicja stylu?

W zaprezentowanej definicji stylu zdefiniowany jest krój pisma, który zostanie zastosowany do tekstu strony w sekcji body. Innymi słowy tekst w tej sekcji będzie pisany czcionką `Verdana` (lub czcionką `Arial`, ewentualnie czcionką `Helvetica`, jeśli `Verdana` nie zostanie znaleziona na komputerze użytkownika – jak widzisz, arkusze stylu uwzględniają nawet ewentualny brak zaplanowanej przez twórcę strony czcionki). Styl czcionki zostanie zastosowany także do tekstu umieszczonego między znacznikami `<p>` i `<div>`, natomiast tekst umieszczony w komórkach tabeli wymaga odrębnej definicji stylu.

W jaki sposób w jednej definicji stylu określić kilka właściwości?

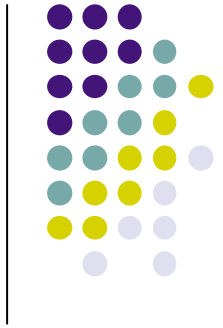
Jeśli chciałbyś zdefiniować w definicji stylu kilka właściwości, musisz je rozdzielać średnikami, tak jak w poniższym wyrażeniu przykładowym, w którym do tekstu stosowane jest wyrównanie i kolor:

```
p {text-align: center; color: red}
```

Aby definicja była bardziej czytelna, każdą z właściwości możesz umieszczać w osobnym wierszu:

```
 p
{
text-align: center;
color: black;
font-family: arial
}
```

Ogólna postać CSS



Przypisanie jednej wartości kilku elementom?

W definicji stylu elementy (inaczej selektory) można **grupować**. Selektory w grupie oddzielane są przecinkami. Oto przykład definicji, w której wszystkim poziomom nagłówka (od h1 do h6) przypisany zostaje kolor czerwony:

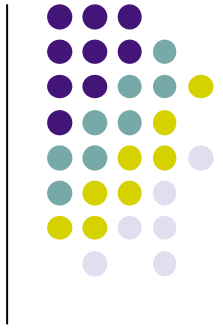
```
h1, h2, h3, h4, h5, h6
{
color: red
}
```

Czy jednemu elementowi można przyporządkować różne style?

Powiedzmy, że chcesz w swoim dokumencie zastosować do akapitów dwa różne sposoby wyrównania tekstu: pewne akapity chcesz dosunąć do prawego marginesu, a pozostałe wycentrować. W takim przypadku przydatny będzie atrybut `class`. Pozwala on zdefiniować różne style dla tego samego elementu – inaczej mówiąc pozwala zdefiniować **klasy stylu**.
Klasę określa się w taki oto sposób:

```
element.nazwa_klasy{właściwość: wartość}
```

Atrybut class



W naszym przykładzie musimy więc zdefiniować dwie klasy: pierwsza, nazwijmy ją prawy, w której ustawimy sposób wyrównania tekstu do prawego marginesu, i klasa center, w której tekst jest wyśrodkowany:

```
p.prawy {text-align: right}
p.center {text-align: center}
```

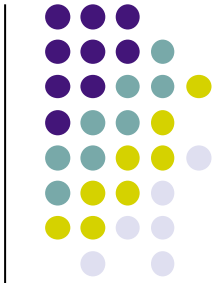
Teraz wystarczy wstawić nazwę klasy do tych znaczników, w których chcemy mieć określony sposób formatowania i gotowe:

```
<p class="prawy">
```

Ten akapit zostanie wyrównany do prawej.

```
</p>
<p class="center">
Ten akapit zostanie wyśrodkowany.
</p>
```

Atrybut class



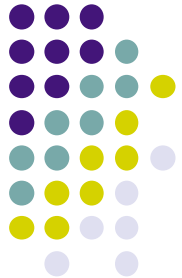
Nazwy klas mogą być dowolne, pamiętaj jednak, by **nie stosować w nich polskich liter**.
Nazwę selektora w definicji klasy można pominąć:

```
.nazwa_klasy{właściwość: wartość}
```

Wówczas zdefiniowany zostanie styl, który można zastosować do wielu elementów. W przykładzie przedstawionym poniżej klasa prawy została zastosowana zarówno do elementu `h1`, jak i do akapitu `p`:

```
<h1 class="prawy">
Ten nagłówek został wyrównany do prawego marginesu.
</h1>
<p class="prawy">
Ten akapit został wyrównany do prawego marginesu.
</p>
```

Atrybut id



Czy można narzucić styl pojedynczym wystąpieniom danego elementu?

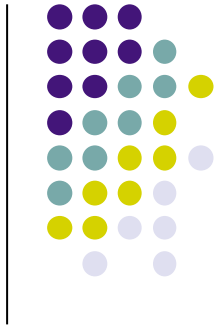
Pojedynczym wystąpieniom danego typu można narzucić styl korzystając z atrybutu id. Atrybut id pozwala więc oznaczać elementy podobnie jak atrybut `class` – `id` jest czymś w rodzaju identyfikatora elementów. Nazwa atrybutu id musi być jednoznaczna. W dokumencie może istnieć **tylko jeden** element o danym `id`:

```
<p id="wstep">
```

Do tego akapitu zostaną zastosowane ustawienia stylu zdefiniowane w definicji id o nazwie `wstep`.

```
</p>
```

Atrybut id



Atrybut `id` można zdefiniować na dwa sposoby: tak, by styl stosowany był do dowolnego elementu o określonym `id` lub tak, by styl był stosowany do określonego elementu o danym `id`.

Pierwszy przypadek, styl zostanie zastosowany do każdego elementu oznaczonego identyfikatorem `wstep`:

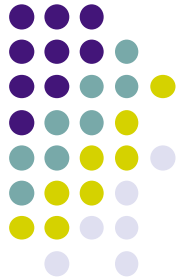
```
#wstep
{
font-size:110%;
font-weight:bold;
color:#0000ff;
background-color:transparent
}
```

Drugi przypadek, styl będzie zastosowany tylko do elementu `p` o `id="wstep"`:

```
p#wstep
{
font-size:110%;
font-weight:bold;
color:#0000ff;
background-color:transparent
}
```

Zwróć uwagę, że w obu przypadkach ustawienia stylu, które będą stosowane do elementów oznaczonych atrybutem `id` identyfikuje się nazwą atrybutu `id` poprzedzoną znakiem `#`, o tak: `#nazwa_id`.

Styl wpisany



Styl wpisany jest wprowadzany w wybranym znaczniku i nie ma wpływu na pozostałe znaczniki. Jeśli chcesz na przykład zastosować kursywę do zawartości znacznika `<h1>`, odpowiednią informację musisz umieścić w tym właśnie znaczniku.

Ogólna postać definicji stylu wpisanego:

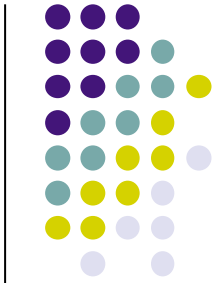
```
<znacznik style="definicja">
```

Przykład:

```
<h1 style="font-size: 20pt; font-weight: bold; font-family: Arial, sans-
serif">Treść tytułu</h1>
```

Styl wpisany powinien być stosowany tylko wtedy, gdy konieczne jest zastosowanie specyficznego stylu do pojedynczego wystąpienia danego elementu, z reguły tylko tam, gdzie niezbędne są drobne korekty stylu. Nie spełnia on bowiem zasadniczego celu kaskadowego arkusza stylów, a więc globalnej kontroli stylu całej strony.

Styl osadzony



Osadzony arkusz stylów to definicja stylu umieszczona między znacznikami `<style></style>`. Element ten powinien znaleźć się w nagłówku dokumentu. Konfiguruje on atrybuty stylu dla całej strony. Oto przykład:

```
<head>
<style type="text/css">
<--!
hr {color: sienna}
p {margin-left: 20px}
body {background-image: url("images/kot.gif")}
-->
</style>
</head>
```

style.css

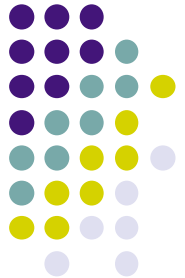
```
H1
{
text-align: center;
color: green;
}


UL
{
text-align: left;
color: white;
list-style-type: disc;
font-weight: bold;
}

.klasa2
{
color: red;
background: none
}
```

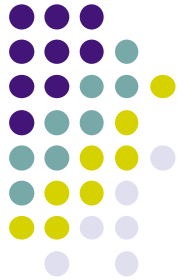
```
body
{
  font-family: Verdana, 'Trebuchet
     MS', Arial, Helvetica, sans-serif;
  font-size: 8pt;
  font-variant: small-caps;
  font-style: italic;
  color: black;
  background-image: url(2.gif);
  margin: 15mm;
  margin-top: 10mm;
  margin-bottom: 15mm;
  text-align: center;
  list-style-type: square;
}
```

Do plików HTML

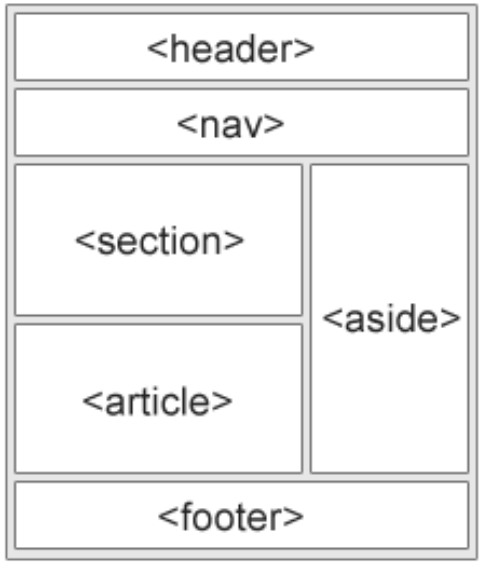


- <LINK REL="Stylesheet" HREF=”style.css" TYPE="text/css">

Podłączenie CSSa pod HTML

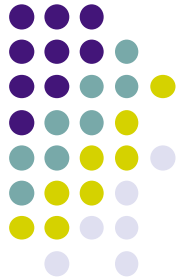


Website Layout - HTML5



header	Defines a header for a document or a section
nav	Defines a container for navigation links
section	Defines a section in a document
article	Defines an independent self-contained article
aside	Defines content aside from the content (like a sidebar)
footer	Defines a footer for a document or a section
details	Defines additional details
summary	Defines a heading for the details element

Kod strony w HTML 5 + CSS



Dziękuję za uwagę.

Opole, 2021