Kurs WWW

Paweł Rajba
pawel@ii.uni.wroc.pl
http://pawel.ii.uni.wroc.pl/

Wprowadzenie

- Przykład na początek
  - W firmie mamy dwa zespoły: programiści PHP i tych, co robią layout w HTMLu i CSSie.
  - Ci co chcą, żeby ci od layoutu grzebali w ich kodzie i na odwrót.
  - Potrzebujemy technologii, która pozwoli oddzielić część prezentacji od części tzw. logiki biznesowej
  - Do tego właśnie służą Template Engines

Wprowadzenie

- Celem template engines, jest oddzielenie kodu aplikacji od prezentacji. Dzięki temu mamy:
  - zwiększoną czytelność kodu HTML i PHP
  - łatwiejsze zarządzanie dużymi projektami
  - programista PHP może bez konsultacji z projektantem HTML+CSS dokonywać zmian,
  - podobnie projektant HTML+CSS może bez konsultacji z projektantem HTML+CSS dokonywać zmian
  - programista może modyfikować kod PHP bez konieczności dokonywania zmian w szablonach

Wprowadzenie

- SmartTemplates – charakterystyka
  - Jest bardzo szybki, oparty na PHP
  - Szablony są kompilowane, a nie parsowane przy każdym uruchomieniu; w razie potrzeby szablony są przekompilowywane
  - Język szablonów jest rozszerzalny, gdyż można pisać własne funkcje i modyfikatory zmiennych
  - Znaczniki szablonów są konfigurowalne, można używać { }, {{ }}, <!--{ }-->, itd.

Wprowadzenie

- SmartTemplates – charakterystyka c.d.
  - Konstrukcje if/elseif/else/endif są przekazywane bezpośrednio do PHP, stąd konstrukcja {if...} może być dowolnie skomplikowana
  - Możliwość nieograniczonego zagnieżdżania różnych konstrukcji np. ifów.
  - Kod PHP można osadzać bezpośrednio w szablonach, jednak nie jest to zalecane
  - Wbudowane buforowanie, można także tworzyć własne funkcje do obsługi buforowania
  - Architektura typu plug-in

Instalacja

- Smarty i dokumentację można pobrać z:
  - http://smarty.php.net/do_download.php?download_file=Smarty-2.6.9.tar.gz
  - http://smarty.php.net/distributions/manual/en/Smarty-2.6.9-docs.pdf
- Wymagania
  - PHP w wersji co najmniej 4.0.6

Instalacja

- Kopiujemy zawartość katalogu "lib" z pobranego pliku Smarty.tar.bz do katalogu:
  - który określa zmienna include_path w pliku php.ini np. include_path=".;d:\Programy\Smarty"
  - albo do dowolnego innego katalogu, ale wtedy musimy ustawić zmienną SMART_DIR
- Po skopiowaniu i restarcie serwera WWW, poniższy kod nie powinien zwrócić błędów

```
<?php
require("Smarty.class.php");
$smarty = new Smarty;
?>
```

Instalacja

- Jeśli poprzedni kod zwróci błąd, należy zrobić jedną z następujących rzeczy:
  - Podać bezwzględną ścieżkę do pliku:

```
<?php
require('d:/Programy/Smarty/Smarty.class.php');
$smarty = new Smarty;
?>
```

  - Ustawić ręcznie zmienną SMART_DIR

```
<?php
define('SMARTY_DIR', 'd:\Programy\Smarty\');
require(SMARTY_DIR.'Smarty.class.php');
$smarty = new Smarty;
?>
```

Instalacja

- Każda aplikacja oparta na smarty musi mieć utworzone następujące katalogi
  - templates
  - config
  - templates_c
    - tutaj serwer WWW musi mieć prawo do zapisu
  - cache
    - tworzymy, jeśli korzystamy z buforowania
    - tutaj serwer WWW musi mieć prawo do zapisu
- Tworzymy pliki tpl i php. I korzystamy

Pierwszy przykład

- Pliki:
  - templates/pierwszy.tpl
  - templates/naglowek.tpl
  - configs/konfiguracje.conf
  - pierwszy.php
- Po uruchomieniu zaglądamy do katalogu
  - templates_c

Cześć I

- Najpierw omówimy część tworzenia pliku szablonów

Komentarze

- Składnia
  - {* To jest komentarz *}

Zmienne

- Przykłady
  - {$foo} – prosta zmienna
  - {$foo[4]} – 5 element tablicy
  - {$foo.bar} – analogicznie jak w PHP $foo['bar']
  - {$foo.$bar} – analogicznie jak w PHP $foo[$bar]
  - {$foo->bar} – właściwość "bar" obiektu $foo
  - {$foo->bar()} – wynik metody "bar" obiektu $foo
  - {#foo#} – wartość zmiennej "foo" z pliku konf.
  - {$smarty.config.foo} – synonym dla {#foo#}
  - {$foo[bar]} – składnia dostępna w pętlach

Zmienne

- Przykłady c.d.
  - {$foo.bar.baz}
  - {$foo.$bar.$baz}
  - {$foo[4].baz}
  - {$foo[4].$baz}
  - {$foo.bar.baz[4]}
  - {$foo->bar($baz,2,$bar)}
  - {"foo"} – wartości statyczne

Zmienne

- Nadawanie wartości w PHP
  - Przykład

```
<?php
$smarty = new Smarty;
$smarty->assign('Contacts', array(
    'fax' => '555-222-9876',
    'email' => 'zaphod@slartibartfast.com',
    'phone' => array(
        'home' => '555-444-3333',
        'cell' => '555-111-1234')));
$smarty->display('index.tpl');
?>
```

Zmienne

- Nadawanie wartości w PHP
  - Przykład c.d., zawartość pliku index.tpl

```
{$Contacts.fax}<br />
{$Contacts.email}<br />
{* you can print arrays of arrays as well *}
{$Contacts.phone.home}<br />
{$Contacts.phone.cell}<br />
```

  - wyprodukuje

```
555-222-9876<br />
zaphod@slartibartfast.com<br />
555-444-3333<br />
555-111-1234<br />
```

Zmienne

- Zmienna {$smarty}
  - {$smarty.now}
    - liczba sekund od 1.1.1970
  - {$smarty.const}
    - pozwala na dostęp do stałych PHP
    - przykład: {$smarty.const.SMARTY_DIR}
  - {$smarty.config}
    - pozwala na dostęp do zmiennych z pliku konf.
    - synonimem jest {#nazwazmiennej#}
  - {$smarty.section}, {$smarty.foreach}
    - przydatne przy pętlach (będzie omawiane później)

Zmienne

- Zmienna {$smarty} c.d.
  - {$smarty.template}
    - nazwa aktualnie przetwarzanego pliku template
  - {$smarty.ldelim}, {$smarty.rdelim}
    - lewy i prawy ogranicznik w plikach template

Modyfikatory zmiennych

- Co to jest?
  - pozwalają na formatowanie wartości zmiennych

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - capitalize, count_characters
  - cat, count_words
  - date_format
  - default, escape
  - lower, nl2br
  - regex_replace
  - string_format
  - strip, upper
  - wordwrap

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - capitalize
    - Zamienia pierwsze litery słów na wielkie litery
    - Parametr typu boolean (opcjonalny) określa, czy zamieniać litery w słowach, w których są cyfry

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - capitalize, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('articleTitle',
                'next x-men film, x3, delayed.');
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$articleTitle}
      {$articleTitle|capitalize}
      {$articleTitle|capitalize:true}
      ```
    - ```
      next x-men film, x3, delayed.
      Next X-Men Film, x3, Delayed.
      Next X-Men Film, X3, Delayed.
      ```
      ```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - count_characters
    - Zlicza ilość znaków w zmiennej
    - Parametr typu boolean (opcjonalny) określa, czy przy zliczaniu mają być brane pod uwagę białe znaki

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - count_characters, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('articleTitle',
           'Cold Wave Linked to Temperatures.');
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$articleTitle}
      {$articleTitle|count_characters}
      {$articleTitle|count_characters:true}
      ```
    - ```
      Cold Wave Linked to Temperatures.
      29
      33
      ```
      ```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - count_words
    - Zlicza ilość słów
  - count_words, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('articleTitle',
          'Dealers Will Hear Car Talk at Noon.');
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$articleTitle}
      {$articleTitle|count_words}
      ```
    - ```
      Dealers Will Hear Car Talk at Noon.
      7
      ```
      ```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - cat
    - Dołącza do zmiennej napis podany w parametrze
    - Parametr typu string (opcjonalny) określa, jaki napis ma zostać dołączony

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - cat, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('articleTitle',
            "Psychics predict world didn't end");
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$articleTitle|cat:" yesterday."}
      ```
    - ```
      Psychics predict world didn't end yesterday.
      ```
      ```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - date_format
    - Formatuje datę i/lub czas do dowolnego formatu
    - Pierwszy parametr typu string (opcjonalny) określa format daty. Format tworzymy za pomocą odpowiednich zmiennych
    - Drugi parametr typu string (opcjonalny) określa domyślną datę, jeśli zmienna jest pusta.

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - date_format, przykład
    - <?php
      $smarty = new Smarty;
      $smarty->display('index.tpl');
      ?>
    - {$smarty.now|date_format}
      {$smarty.now|date_format:"%A, %B %e, %Y"}
      {$smarty.now|date_format:"%H:%M:%S"}
    - Feb 6, 2001
      Tuesday, February 6, 2001
      14:33:00

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - default
    - Ustawia wartość domyślną w przypadku, gdy zmienna jest nieustawiona
    - Parametr typu string (opcjonalny) określa, jaka ma być wartość domyślna

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - default, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('articleTitle',
         'Dealers Will Hear Car Talk at Noon.');
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$articleTitle|default:"no title"}
      {$myTitle|default:"no title"}
      ```
    - ```
      Dealers Will Hear Car Talk at Noon.
      no title
      ```
      ```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - escape
    - Zamienia niektóre znaczki na kody
    - Parametr typu string (opcjonalnie) określa, które znaczki i jak będą zamieniane. Dopuszczalne wartości:
      - html, htmlall
      - url,
      - quotes,
      - hex, hexentity,
      - javascript

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - escape, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('articleTitle',
            "'Stiff Opposition Expected
             to Casketless Funeral Plan'");
      $smarty->display('index.tpl');
      ?>
      ```
      ```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - escape, przykład c.d.

```
{$articleTitle}
{$articleTitle|escape}
{$articleTitle|escape:"html"}
  {* escapes & " ' < > *}
{$articleTitle|escape:"htmlall"}
  {* escapes ALL html entities *}
{$articleTitle|escape:"url"}
{$articleTitle|escape:"quotes"}
<a href="mailto:
  {$EmailAddress|escape:"hex"}">
  {$EmailAddress|escape:"hexentity"}</a>
```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - escape, przykład c.d.
    - 'Stiff Opposition Expected to Casketless Funeral Plan'
    - 'Stiff Opposition Expected to Casketless Funeral Plan'
    - 'Stiff Opposition Expected to Casketless Funeral Plan'
    - 'Stiff Opposition Expected to Casketless Funeral Plan'
    - %27Stiff+Opposition+Expected+to+Casketless+Funeral+Plan%27
    - \'Stiff Opposition Expected to Casketless Funeral Plan\'
    - <a href="mailto:%62%6f%62%40%6d%65%2e%6e%65%74">...

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - lower
    - zamienia wszystkie litery na małe
  - lower, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('articleTitle',
          'Two Convicts Evade Noose, Jury Hung.');
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$articleTitle}
      {$articleTitle|lower}
      ```
    - ```
      Two Convicts Evade Noose, Jury Hung.
      two convicts evade noose, jury hung.
      ```
      ```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - nl2br
    - znak nowej linii zamienia na sekwencję
  - nl2br, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('articleTitle', "Sun or
      rain expected\ntoday, dark tonight");
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$articleTitle|nl2br}
      ```
    - ```
      Sun or rain expected<br />today, dark
      tonight
      ```
      ```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - regex_replace
    - Wyszukuje i zamienia wyrażenia regularne
    - Składnia wyrażeń taka, jak w funkcji preg_replace() z PHP
    - Parametr pierwszy typu string określa wyrażenie, które będzie zamieniane
    - Parametr drugi typu string określa napis, który będzie wstawiany w miejsce wyrażenia
    - Oba parametry są obowiązkowe

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - regex_replace, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('articleTitle',
        "Infertility unlikely to\nbe passed on.");
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$articleTitle}
      {$articleTitle|regex_replace:"/[\r\t\n]/":"
      "}
      ```
    - ```
      Infertility unlikely to
      be passed on.
      Infertility unlikely to be passed on.
      ```
      ```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - string_format
    - Formatuje napisy i liczby
    - Parametr pierwszy typu string (obowiązkowy) określa, co formatować
    - Składnia formatowania taka jak w funkcji sprintf()

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - string_format, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('number', 23.5787446);
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$number}
      {$number|string_format:"%.2f"}
      {$number|string_format:"%d"}
      ```
    - ```
      23.5787446
      23.58
      24
      ```
      ```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - strip
    - Zamienia wszystkie sekwencja tabulatorów, znaków nowej linii i spacji na pojedynczą spację lub określony ciąg znaków
    - Parametr pierwszy typu string (opcjonalny) określa napis, na który sekwencje mają być zamieniane

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - strip, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('articleTitle',
      "Grandmother of\neight makes\t hole in
      one.");
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$articleTitle}
      {$articleTitle|strip}
      {$articleTitle|strip:" "}
      ```
    - Grandmother of
      eight makes hole in one.
      Grandmother of eight makes hole in one.
      Grandmother of eight makes hol
      e in one.

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - upper
    - Zamienia wszystkie litery na wielkie
  - upper, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('articleTitle',
         "If Strike isn't Settled Quickly ...");
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$articleTitle}
      {$articleTitle|upper}
      ```
    - ```
      If Strike isn't Settled Quickly ...
      IF STRIKE ISN'T SETTLED QUICKLY ...
      ```
      ```

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - wordwrap
    - Zawija wiersze w napisie
    - Parametr pierwszy typu integer określa, jakie długie mają być wynikowe wiersze; domyślnie 80
    - Parametr drugi typu string określa jaka sekwencja ma być po każdym wierszu; domyślnie "\n"
    - Parametr trzeci typu boolean określa, czy zawijać na końcu słowa (false), czy dokładnie na znaku (true); domyślnie false (mi to i tak nie działało)
    - Wszystkie parametry są opcjonalne

Modyfikatory zmiennych

- Lista wybranych modyfikatorów
  - wordwrap, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('cytat','Indeed? I heard it
          not: then it draws near the season.');
      $smarty->display('index.tpl');
      ?>
      ```
    - ```
      {$cytat}
      {$cytat|30:"<br>"}
      ```
    - Indeed? I heard it not: then it draws near the season.
      Indeed? I heard it not: then
      it draws near the season.

Modyfikatory zmiennych

- Modyfikatory można łączyć. Przykład.
- ```
  <?php
  $smarty = new Smarty;
  $smarty->assign('cytat','Indeed? I heard it
      not: then it draws near the season.');
  $smarty->display('index.tpl');
  ?>
  ```
- ```
  {$cytat}
  {$cytat|wordwrap:30:"<br>"|capitalize|cat:" ..."}
  ```
- Indeed? I heard it not: then it draws near the season.

  Indeed? I Heard It Not: Then
  It Draws Near The Season. ...

Funkcje

- Składnia
  - {funcname attr1="val1" attr2="val2"}
- Przykłady:
  - {config_load file="colors.conf"}
  - {include file="header.tpl"}
  - {if $highlight_name}
Welcome, {$name}!
{else}
Welcome, {$name}!
{/if}
  - {include file="footer.tpl"}

Atrybuty funkcji

- Wartości boolowskie mogą być przekazywane za pomocą następujących literałów:
  - true, on, yes lub false, off, no
- Przykłady
  - {include file="header.tpl"}
  - {include file=$includeFile}
  - {include file=#includeFile#}
  - {html_select_date display_days=yes}
  - <select name="company">
    {html_options values=$vals selected=$selected output=$output}
    </select>

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - capture
  - config_load
  - foreach,foreachelse
  - include
  - if,elseif,else
  - literal
  - php
  - section,sectionelse

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - capture
    - Pozwala przechwytywać wyjście do zmiennej
    - Składnia
      - ```
        {capture name=zmienna}
            {* tutaj jest wszystko *}
            {* przechwytywane do zmienna *}
        {/capture}
        ```
    - Dostęp do przechwyconego mamy poprzez zmienną **`{$smarty.capture.zmienna}`**
        ```

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - capture, przykład
    - ```
      {capture name=banner}
      {include file="get_banner.tpl"}
      {/capture}
      {if $smarty.capture.banner ne ""}
          <tr><td>
          {$smarty.capture.banner}
          </td></tr>
      {/if}
      ```
      ```

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - config_load
    - Funkcja do wczytywania zmiennych z plików konfiguracyjnych
    - Składnia
      - ```
        {config_load
            file="plik.conf"
            section="sekcja"
            scope="local | parent | global}
        ```
    - Znaczenie zasięgu
      - local – wczytywany tylko w lokalnym pliku szablonu
      - parent – wczytywane także w pliku, który wywołał bieżący plik szablonu
      - global – dostępne we wszystkich plikach szablonów
        ```

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - config_load, przykład
    - ```
      {config_load file="colors.conf"
                   section="Customer"}
      <html>
      <title>{#pageTitle#}</title>
      <body bgcolor="{#bodyBgColor#}">
      <table border="{#tableBorderSize#}"
      bgcolor="{#tableBgColor#}">
      <tr bgcolor="{#rowBgColor#}">
      <td>First</td><td>Last</td><td>Address</td>
      </tr>
      </table>
      </body>
      </html>
      ```
      ```

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - foreach, foreachelse
    - Służy do iterowania pojedynczych tablic asocjacyjnych
    - Jest prostszą wersją pętli section
    - Parametry:
      - from (array) – tablica źródłowa (wymagany)
      - item (string) – nazwa zmiennej iterującej (wymagany)
      - key (string) – nazwa zmiennej z kluczem (opcjonalny)
      - name (string) – nazwa pętli; potrzebna, jeśli chcemy mieć dostęp do zmiennych specjalnych (opcjonalny)

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - foreach, foreachelse c.d.
    - Zmienne związane z pętlą
      - iteration – numer bieżącej iteracji; numeracja zawsze zaczyna się od 1
      - first – prawdziwe, jeśli jest pierwsza iteracja
      - last – prawdziwe, jeśli jest ostatnia iteracja
      - total – liczba wszystkich iteracji; można używać wewnątrz i za pętlą
  - foreach, foreachelse, przykład
    - templates/drugi.tpl
    - templates/naglowek.tpl
    - drugi.php

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - include
    - Służy do dołączania innych plików tpl
    - Parametry
      - file="plik.tpl" – określa plik, która należy dołączyć (wymagany)
      - lista przypisań zmienna=wartość, które zostaną zrealizowane podczas wstawiania zawartości

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - include, przykłady
    - {include file="header.tpl" title="Main Menu"
      table_bgcolor="#c0c0c0"}
      {* body of template goes here *}
      {include file="footer.tpl" logo="http://example.com/logo.gif"}
    - {* absolute filepath *}
      {include file="/usr/local/include/tpls/header.tpl"}
    - {* absolute filepath (same thing) *}
      {include file="file:/usr/local/include/tpls/header.tpl"}
    - {* windows absolute filepath (MUST use "file:" *}
      {include file="file:C:/www/pub/tpls/header.tpl"}
    - {* include from template resource named "db" *}
      {include file="db:header.tpl"}

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - if,elseif,else
    - pozwala na warunkowe wykonanie kodu
  - Przykłady
    - ```
      {if $name eq "Fred"}
      Welcome Sir.
      {elseif $name eq "Wilma"}
      Welcome Ma'am.
      {else}
      Welcome, whatever you are.
      {/if}
      ```
      ```

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - if,elseif,else, przykłady
    - ```
      {* an example with "or" logic *}
      {if $name eq "Fred" or $name eq "Wilma"}
      ...
      {/if}
      ```
    - ```
      {* same as above *}
      {if $name == "Fred" || $name == "Wilma"}
      ...
      {/if}
      ```
    - W powyższym przykładzie spacje wokół == są ważne (inaczej nie zadziała)

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - if,elseif,else, przykłady
    - ```
      {* parenthesis are allowed *}
      {if ( $amount < 0 or $amount > 1000 ) and
            $volume >= #minVolAmt#}
      ...
      {/if}
      ```
    - ```
      {* you can also embed php function calls *}
      {if count($var) gt 0}
      ...
      {/if}
      ```
    - ```
      {* test if values are even or odd *}
      {if $var is not odd}
      ...
      {/if}
      ```
      ```

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - literal
    - pozwala umieszczenie kodu, który nie będzie parsowany
  - literal, przykład

```
{literal}
<script type="text/javascript">
<!--
function isblank(field) {
  if (field.value == '') { return false; }
  else { return true; }
}
// -->
</script>
{/literal}
```

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - php
    - Pozwala umieścić bezpośrednio kod PHP
  - php, przykład
    - ```
      {php}
      // including a php script directly
      // from the template.
      include("/path/to/display_weather.php");
      {/php}
      ```
      ```

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - section, sectionelse
    - Służy do iteracji zmiennych typu array
    - Parametry
      - name (string) – nazwa sekcji (wymagany)
      - loop (mixed) – określa ile razy sekcja będzie wyświetlana; zwykle jest to tablica (wymagany)
      - start – początkowy indeks iteracji; jeśli bedzie to wartość ujemna, zostanie policzona od końca

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - section, sectionelse
    - Parametry c.d.
      - step – określa, co który indeks będzie brany pod uwagę np. step=2 daje 0,2,4,...
      - max – określa ile razy maksymalnie sekcja będzie wyświetlona

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - section, sectionelse
    - Zmienne dostępne pod {$smarty.section.sectionname.varname}
      - index – bieżąca iteracja; numeracja od 0 i zwiększany o 1, chyba że step jest ustawiony
      - index_prev – poprzednia iteracja; numeracja od -1
      - index_next – następna iteracja; index+step
      - iteration – numer iteracji; numeracja od 1 (bez względu na inne ustawienia np. step)

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - section, sectionelse
    - Zmienne dostępne pod {$smarty.section.sectionname.varname} c.d.
      - first – prawdziwe, jeśli bieżąca jest pierwszą iteracją
      - last – prawdziwe, jeśli bieżąca jest ostatną iteracją
      - loop – ostatni indeks, który będzie wykorzystany w iteracjach; można użyć także za sekcją
      - total – liczba wykonań sekcji; można użyć także za sekcją

Funkcje wbudowane

- Lista wybranych funkcji wbudowanych
  - section, sectionelse
    - Przykład: drugi.tpl

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - assign, counter, cycle, debug, eval, fetch
  - html_checkboxes, html_image
  - html_options, html_radios
  - html_select_date, html_select_time
  - html_table
  - math, mailto
  - popup_init
  - popup
  - textformat
- Pliki: funkcje.tpl, funkcje.php

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - assign
    - Nadaje zmiennej wartość
    - Parametry
      - var : string – nazwa zmiennej
      - value : string – wartość dla zmiennej
      - oba parametry są wymagane
  - assign, przykład
    - `{assign var="name" value="Bob"}`
    - `The value of $name is {$name}.`
    - `The value of $name is Bob.`

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - counter
    - generuje elementy dowolnego ciągu arytmetycznego
    - parametry:
      - name : string – nazwa ciągu (default)
      - start : number – początkowy element (1)
      - skip : number – odstęp pomiędzy elementami (1)
      - direction : string – czy wstępujący, czy zstępujący (up)
      - print : boolean – czy drukować liczby (true)
      - assign : string – nazwa zmiennej, do której zostanie przekierowany wynik funkcji; liczba nie jest drukowana

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - counter, przykład
    - {counter start="8" skip="-2"}
      {counter}
      {counter}
      {counter}
    - 8
      6
      4
      2

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - cycle
    - pobiera cyklicznie wartości z pewnego zbioru
    - parametry
      - name : string – nazwa cyklu (default)
      - values : mixed – lista przecinkowa (chyba że ustawiony jest parametr delimiter) lub tablica
      - print : boolean – czy drukować wartości
      - delimiter : string – określa separator w values (,)
      - assign : string – nazwa zmiennej, do której zostanie przekierowany wynik funkcji; wartość nie jest drukowana

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - cycle, przykład
    - ```
      <table cellpadding="4"
             cellspacing="2" border="1">
      <tr><th>Imię</th><th>Nazwisko</th></tr>
      {section name=osoba loop=$osoby step=1}
      <tr style="background-color:
      {cycle values="aqua,lime"}">
      <td>{$osoby[osoba].imie}</td>
      <td>{$osoby[osoba].nazwisko}</td>
      </tr>
      {sectionelse}<tr><td>Brak danych.</td></tr>
      {/section}
      </table>
      ```
      ```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - debug
    - wypluwa okienko z informacją o zmiennych itp.

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - eval
    - pozwala traktować zawartość zmiennej jak szablon
    - parametry:
      - var : mixed – zmienna do przetworzenia
      - assign : string – zmienna, do której zostanie przekazany wynik zamiast wypisywania na ekran
  - eval, przykład
    - setup.conf:

```
title = Welcome to {$company}'s home page!
```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - eval, przykład c.d.
    - funkcje.tpl

```
{assign var="company" value="Piggis"}
{#title#}<br>
{eval var=#title#}
```

    - OUTPUT:

```
Welcome to {$company}'s home page!
Welcome to Piggis's home page!
```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_checkboxes
    - tworzy grupę złożoną pól wyboru na podstawie przekazanych danych
    - parametry:
      - name : string – nazwa listy
      - values : array – tablica wartości dla pól
      - output : array – tablica napisów, które będą obok pól
      - select : string/array – wybrany element
      - options : array – tablica asocjacyjna zawierająca wartości i napisy dla pól; alternatywa dla tablic values i output
      - separator : string – tekst, który będzie oddzielał kolejne pola wyboru
      - labels : boolean – czy wstawiać znacznik label

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_checkboxes, przykład
    - ```
      <?php
      require('Smarty.class.php');
      $smarty = new Smarty;
      $smarty->assign('cust_checkboxes', array(
          1000 => 'Joe Schmoe',
          1001 => 'Jack Smith',
          1002 => 'Jane Johnson',
          1003 => 'Charlie Brown'));
      $smarty->assign('customer_id', 1001);
      $smarty->display('index.tpl');
      ?>
      ```
      ```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_checkboxes, przykład c.d.
    - ```
      {html_checkboxes
          name="id"
          options=$cust_checkboxes
          selected=$customer_id
          separator="<br />"}
      ```
      ```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_checkboxes, przykład c.d.

```
<label><input type="checkbox" name="id[]"
   value="1000" />Joe Schmoe</label><br />
<label><input type="checkbox" name="id[]"
   value="1001" checked="checked" />
   Jack Smith</label><br />
<label><input type="checkbox" name="id[]"
   value="1002" />Jane Johnson</label><br />
<label><input type="checkbox" name="id[]"
   value="1003" />Charlie Brown</label><br />
```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_radios
    - Podobna to html_checkboxes

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_options
    - tworzy grupę elementów typu option na podstawie przekazanych danych
    - parametry:
      - values : array – wartości elementów (znacznik value)
      - output : array – teksty elementów
      - selected : string/array – wybrany element
      - options : array – tablica asocjacyjna zawierająca wartości i teksty; alternatywa dla values i output
      - name : string – nazwa grupy; dodaje znaczniki <select name="nazwagrupy"></select>

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_options, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('cust_options', array(
          1001 => 'Joe Schmoe',
          1002 => 'Jack Smith',
          1003 => 'Jane Johnson',
          1004 => 'Charlie Brown'));
      $smarty->assign('customer_id', 1001);
      $smarty->display('index.tpl');
      ?>
      ```
      ```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_options, przykład
    - ```
      {html_options
          name=$customer_id
          options=$cust_checkboxes
          selected=$customer_id}
      ```
    - ```
      <select name="1001">
      <option label="Joe Schmoe"
         value="1000">Joe Schmoe</option>
      <option label="Jack Smith" value="1001"
         selected="selected">Jack Smith</option>
      <option label="Jane Johnson"
         value="1002">Jane Johnson</option>
      <option label="Charlie Brown"
         value="1003">Charlie Brown</option>
      </select>
      ```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_table
    - służy do utworzenia tabelki na podstawie przekazanych danych
    - parametry:
      - loop : array – dane do tabelki
      - cols : integer – liczba kolumn; domyślnie 3
      - rows : integer – liczba wierszy
      - inner : string – określa, jak kolejne elementy będą wkładane: kolumnami czy wierszami; dostępne wartości: cols, rows; domyślnie cols
      - trailpad : string – ciąg znaków na miejsce pustych komórek; domyślnie

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_table
    - parametry c.d.
      - hdir : string – kierunek danych w wierszach; wartości: left, right; domyślnie right (z lewa na prawo)
      - vdir : string – kierunek wyświetlania wierszy; wartości: down, up; domyślnie down (od góry na dół)
      - table_attr : string – atrybuty dla znacznika table; domyślnie border="1"
      - tr_attr : string – atrybuty dla znaczników tr; jeśli przekażemy tablice, jej wartości będą przydzielane cyklicznie
      - td_attr : string – atrybuty dla znaczników td; jeśli przekażemy tablicę, j.w.

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_table, przykład
    - ```
      <?php
      $smarty = new Smarty;
      $smarty->assign('table_data',
          array(1,2,3,4,5,6,7,8,9,10));
      $smarty->assign('tr',
          array('bgcolor="#eeeeee"',
                'bgcolor="#dddddd"'));
      $smarty->assign('table','border="1"
                               cellpadding="4"
                               cellspacing="1"');
      $smarty->display('index.tpl');
      ?>
      ```
      ```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_table, przykład c.d.
    -
```
{html_table loop=$table_data
            cols=5
            rows=3
            tr_attr=$tr
            table_attr=$table}
```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_table, przykład c.d.

```
<table border="1" cellpadding="4"
cellspacing="1">
<tr bgcolor="#eeeeee">
<td>1</td><td>2</td><td>3</td><td>4</td>
</tr>
<tr bgcolor="#dddddd">
<td>5</td><td>6</td><td>7</td><td>8</td>
</tr>
<tr bgcolor="#eeeeee">
<td>9</td><td>10</td>
<td> </td><td> </td>
</tr>
</table>
```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_select_date
    - tworzy pola do wyboru daty; wstawia bieżącą datę
    - parametry: jest ich sporo
    - przykład: {html_select_date}
  - html_select_time
    - tworzy pola do wyboru czasu; wstawia bieżący czas
    - parametry: jest ich sporo
    - przykład: {html_select_time}

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - fetch
    - pobiera listę plików z lokalnego systemu lub zdalnego np. ftp
  - math
    - służy do obliczania wyrażeń arytmetycznych
    - należy używać z rozwagą – niewydajne i powolne
    - przykład

```
{* $height=4, $width=5 *}
{math equation="x + y" x=$height y=$width}
OUTPUT:
9
```

Przydatne funkcje

- Lista wybranych przydatnych funkcji
  - html_image
    - tworzy znacznik <image>
  - mailto
    - tworzy link typu mailto
    - potrafi zakodować ten link korzystając z javascriptu
  - popup_init, popup
    - służy do tworzenia denerwujących okienek
  - textformat
    - służy do formatowania bloku tekstu
    - przykład
      - {textformat} {* tekst do formatowania } {/textformat}

Pliki konfiguracyjne

- Kilka uwag
  - Poprzez te pliki nadajemy zmiennym wartości, z których możemy potem korzystać przy tworzeniu szablonów
  - Mogą służyć do przechowywania pewnych ustawień stron, np. kolorów
  - Wartości zmiennych mogą, ale nie muszą być w cudzysłowach (pojedynczych lub podwójnych)
  - Używając potrójnych cudzysłowów (""") można umieszczać wartości w kilku wierszach
  - Możemy przypisania umieszczać w sekcjach. Pojawienie się wiersza [Dane] tworzy sekcję o nazwie Dane; wtedy możemy wczytać przypisania tylko z tej sekcji

Pliki konfiguracyjne

- Kilka uwag c.d.
  - Przypisania, które nie są w żadnej sekcji (te na początku) są globalne i są zawsze wczytywane
  - Jeśli zmienna jest ustawiana kilka razy, wiążące jest ostatnie przypisanie
  - Do wczytywania plików służy funkcj {config_load}
  - Komentarze tworzymy umieszczając na początku wiersza znak #
  - Można ukryć pewne dane przed wczytaniem do szablonów, tworząc zmienne lub sekcje ukryte. Ukrycie polega na dopisaniu do początku nazwy kropkę

Pliki konfiguracyjne

- Przykład

```
# global variables
pageTitle = "Main Menu"
bodyBgColor = #000000
tableBgColor = #000000
rowBgColor = #00ff00

[Customer]
pageTitle = "Customer Info"

[Login]
pageTitle = "Login"
focus = "username"
Intro = """This is a value that spans more
than one line. you must enclose
it in triple quotes."""

[.Database]
host=my.example.com
db=ADDRESSBOOK
user=php-user
pass=foobar
```

Cześć II

- Teraz omówimy smarty w punktu widzenia programisty

Zmienne

- $template_dir
  - lokalizacja plików szablonów
  - domyślnie ./templates
- $compile_dir
  - lokalizacja skompilowanych szablonów
  - domyślnie ./templates_c
- $configs
  - lokalizacja plików konfiguracyjnych
  - domyślnie ./configs

Zmienne

- $plugins_dir
  - lokalizacja wtyczek
  - domyślnie plugins
  - zaleca się podanie pełnej ścieżki do katalogu (tak, żeby nie było potrzeby korzystać z include_path)
- $debugging
  - włącza uruchamianie konsoli debugowania dla bieżącego pliku szablonu

Zmienne

- $compile_check
  - smarty sprawdza, czy plik szablonu był zmieniany (timestamp); jeśli tak, tpl jest przekompilowywany
  - ustawienie na false wyłącza sprawdzanie i zwiększa wydajność, ale zmiana pliku szablonu nie będzie widoczna
- $force_compile
  - wymusza rekompilację przy każdym wywołaniu
  - przykrywa zmienną $compile_check

Zmienne

- $config_overwrite
  - ustala, czy kolejne wartości o tym samym kluczu w pliku konfiguracyjnym będą nadpisywane, czy będzie tworzona tablica
    - true – nadpisywanie
    - false – tworzenie tablicy
  - domyślnie true
- $config_read_hidden
  - ustala, czy będą wczytywane sekcje ukryte plików konfiguracyjnych
  - domyślnie false

Zmienne

- $left_delimiter, $right_delimiter
  - określają znaki, które będą tworzyć przejście w ,,tryb smarty"
  - domyślnie { i }
- $php_handling
  - określa sposób traktowania wstawek PHP bezpośrednio w pliku szablonu
    - SMARTY_PHP_PASSTHRU – pozostawia bez zmian
    - SMARTY_PHP_QUOTE – drukuje jako tekst
    - SMARTY_PHP_REMOVE – usuwa znaczniki PHP
    - SMARTY_PHP_ALLOW – wykonuje wstawki

Metody

- assign – ustawia wartość zmiennej
  - składnia
    - void assign (mixed var)
    - void assign (string varname, mixed var)
  - przykład
    - `$smarty->assign('Name', 'Fred');`
    - `$smarty->assign('Address', $address);`
    - ```
      $smarty->assign(
         array("city" => "Lincoln",
               "state" => "Nebraska"));
      ```
      ```

Metody

- append – dołącza element do tablicy
  - składnia
    - void append(mixed var)
    - void append(string varname, mixed var)
  - przykład
    - `$smarty->append("Name", "Fred");`
    - `$smarty->append("Address", $address);`
    - ```
      $smarty->append(
         array("city" => "Lincoln",
               "state" => "Nebraska"));
      ```
      ```

Metody

- clear_all_assign – usuwa wszystkie zmienne
  - składnia
    - void clear_all_assign (void)
- clear_assign – usuwa wybrane zmienne
  - składnia
    - void clear_assign (mixed var)
  - przykład
    - `$smarty->clear_assign("Name");`
    - ```
      $smarty->clear_assign(
          array("Name", "Address", "Zip"));
      ```
      ```

Metody

- clear_config – czyści zmienne konfiguracyjne
  - składnia
    - void clear_config ([string var])
  - znaczenie
    - jeśli podamy argument, to usuwana jest wybrana zmienna konfiguracyjna,
    - w przeciwnym razie usuwane są wszystkie zmienne konfiguracyjne
  - przykład
    - ```
      // clear all assigned config variables.
      $smarty->clear_config();
      ```
    - ```
      // clear one variable
      $smarty->clear_config('foobar');
      ```

Metody

- config_load – wczytuje plik konfiguracyjny
  - składnia
    - void **config_load** (string file [, string section])
  - przykłady
    - ```
      // load config variables and assign them
      $smarty->config_load('my.conf');
      ```
    - ```
      // load a section
      $smarty->config_load('my.conf', 'foobar');
      ```

Metody

- display – wyświetla plik szablonu
  - składnia
    - void **display** (string template)
  - przykłady

```
$smarty->display("index.tpl");
$smarty->display("/usr/include/tpl/hdr.tpl");
$smarty->display("file:/usr/include/tpl/hdr.tpl");
// windows filepath (MUST use "file:" prefix)
$smarty->display("file:C:/www/tpls/header.tpl");
// include from template resource named "db"
$smarty->display("db:header.tpl");
```

Metody

- fetch – przetwarza i zwraca plik szablonu
  - składnia
    - string **fetch** (string template)
  - Komentarz:
    - działa tak samo jak display, tylko zamiast wyświetlać zwraca wynik jako napis

Metody

- get_config_vars – pobiera zmienne pobrane z plików konfiguracyjnych
  - składnia
    - array **get_config_vars** ([string varname])
  - przykłady
    - ```
      // get loaded config template var 'foo'
      $foo = $smarty->get_config_vars('foo');
      // get all loaded config template vars
      $config_vars = $smarty->get_config_vars();
      // take a look at them
      print_r($config_vars);
      ```
      ```

Metody

- get_template_vars
  - składnia
    - array **get_template_vars** ([string varname])
  - przykłady

```
// get assigned template var 'foo'
$foo = $smarty->get_template_vars('foo');
// get all assigned template vars
$tpl_vars = $smarty->get_template_vars();
// take a look at them
print_r($tpl_vars);
```

Metody

- template_exists – sprawdza, czy istnieje wskazany plik szablonu
  - składnia
    - bool **template_exists** (string template)

Smarty i wtyczki

- Kilka uwag wstępnych
  - Wtyczki są zawsze ładowane na żądanie
  - Każda wtyczka jest ładowana tylko raz, nawet jeśli mamy kilka instancji Smarty
  - Katalog z wtyczkami może być napisem zawierającym ścieżke lub tablicą zawierającą ścieżki

Smarty i wtyczki

- Instalację wtyczki możemy przeprowadzić na dwa sposoby:
  - skopiować plik z wtyczką do jednego z katalogów, w którym przechowywane są wtyczki. Obowiązuje wtedy odpowiednie nazewnictwo zarówno plików jak i zawartych w nich funkcjach
  - utworzyć plik i zarejestrować z smarty odpowiedni obiekt. Do rejestracji obiektów mamy funkcje o nazwach register_* (np. register_function), a do wyrejestrowania - unregister_*.

Smarty i wtyczki

- Nazewnictwo
  - każdy plik wtyczki musi mieć nazwę wg wzoru
    - typ.nazwa.php
      - gdzie typem może być: function, modifier, block, compiler, prefilter, postfilter, outputfilter, resource, insert
      - a nazwa to nazwa wtyczki
  - natomiast wewnątrz pliku funkcja musi mieć nazwę
    - smarty_typ _nazwa()
  - W przypadku złego nazewnictwa, Smarty wypluje odpowiednie komunikaty o błędach

Smarty i wtyczki

- Utworzenie wtyczki z funkcją i modyfikatorem
  - Pliki
    - wtyczki.php
    - templates/wtyczki.tpl
    - plugins2/function.eightball.php
    - plugins2/modifier.capitalize.php
  - Robimy drobne modyfikacje
    - zmieniamy katalog plugins2 na plugins
    - zauważamy, że nie trzeba już niczego rejestrować