Bazy Danych i Usługi Sieciowe

Wstęp do usług sieciowych

Paweł Witkowski

Wydział Matematyki, Informatyki i Mechaniki

Jesień 2014

Plan wykładu

1. Protokół HTTP
2. HTML
3. CSS
4. JavaScript
5. Język PHP
6. Składnia PHP
7. Dostęp do bazy danych
8. Przetwarzanie danych z formularzy

Protokół HTTP

1. Protokół warstwy aplikacji modelu OSI
2. Określa zasady udostępniania i przesyłania rozproszonych informacji i multimediów
3. Pierwsza wersja, HTTP/0.9, w roku 1990
4. Obecnie używana wersja to HTTP/1.1
5. Zapewnia działanie World Wide Web

Model Klient-Serwer

1. Klient wysyła żądanie (*request*) do serwera
2. Żądanie zawiera URL - Uniform Resource Locator
3. Serwer zwraca zasób, jeżeli jest dostępny (*response*)
4. HTTP jest protokołem bezstanowym
5. Istnieją mechanizmy zapewniające emulację stanów – sesje, ciasteczka

URL

Adres URL

`http://host[:port][/path][?query]`

1. `host` – adres serwera, w postaci domenowej lub adresu IP
2. `port` – numer portu, na którym serwer odbiera połączenia TCP, standardowo 80
3. `path` – ścieżka dostępu do zasobu, jeżeli nieobecna podaje się "/"
4. `query` – dodatkowe informacje

Żądanie (*Request*)

1. Wybór metody stosowanej do zasobu
2. Adres zasobu
3. Wersja protokołu HTTP
4. Dane

Metody

1. OPTIONS
2. GET
3. HEAD
4. POST
5. PUT
6. DELETE
7. TRACE
8. CONNECT

Odpowiedź (*Response*)

1. Wersja protokołu HTTP
2. Status odpowiedzi – trzycyfrowy kod
    1. 1xx - informacyjne
    2. 2xx - sukces, potwierdzenie
    3. 3xx - przekierowanie
    4. 4xx - błąd po stronie klienta
    5. 5xx - błąd po stronie serwera
3. Dane

Status odpowiedzi - przykłady

1. 200 - OK
2. 301 - Zasób przeniesiony (*Moved Permanently*)
3. 401 - Brak autoryzacji (*Unauthorized*)
4. 403 - Dostęp zabroniony (*Forbidden*)
5. 404 - Nie znaleziono zasobu (*Not Found*)
6. 500 - Błąd serwera (*Internal Server Error*)

HTML

- **H**yper**T**ext **M**arkup **L**anguage
- Język oparty na znacznikach
- Opisuje strukturę stron www
- Pozwala na umieszczanie odnośników do zasobów zewnętrznych
- Umożliwia tworzenie formularzy do zbierania danych
- Umożliwia zanurzanie obrazów, animacji, dźwięków, filmów i innych typów danych

Historia HTML

1990 Tim Berners-Lee tworzy podstawy standardu języka znaczników służących do opisu dokumentów publikowanych w Internecie – HTML 1.0

1995 Dzięki współpracy z twórcami przeglądarek powstaje HTML 2.0

1996 HTML 3.2 jako statndard opracowany przez niezależną organizacę W3C (World Wide Web Consortium)

1997 HTML 4.0

1999 HTML 4.1 – najdłużej istniejący standard języka; warianty *strict* i *transitional*

2000 XHTML 1.0 – standard zgodny z XML

2008 Pierwsze specyfikacje standardu HTML5

2014 28 października, HTML5 z oficjalną rekomendacją W3C

Dokumenty HTML

- Pliki tekstowe, bez formatowania
- Treść razem ze znacznikami
- Przygotowywanie w dowolnym edytorze tekstowym
- *Notatnik, Notepad++, Gedit*
- Edytory wspomagające
- *Aptana, Dreamweaver, Netbeans, Visual Web Developer, ...*

HTML

- Znaczniki HTML służą to definiowania w treści strony
  - Akapitów
  - Nagłówków
  - List
  - Tabel
  - Obrazków
  - Odnośników
  - Pól formularzy
  - Innych obiektów

Używanie znaczników

- Znacznik otwierający - np. <**p**>
- Tekst w znaczniku - np. treść akapitu
- Znacznik zamykający - np. </**p**>
- Znaczniki samozamykające - np. koniec linii <**br** />
- Atrybuty znaczników - np. cel odnośnika
- <**a href=”http://www.uw.edu.pl”**>UW</**a**>

Nagłówek i treść

- Nagłówek strony (**head**)
- Treść strony (**body**)

Podstawowa struktura

```
1  <html>
2    <head>
3    ...
4    </head>
5    <body>
6    ...
7    </body>
8  </html>
```

Akapity i końce linii

- Przeglądarki ignorują końce linii i wielokrotne spacje
- Strukturę tekstu definiujemy wyłącznie za pomocą znaczników
- Akapity - znacznik <**p**>
- Koniec wiersza - znacznik <**br** />

Akapity i nagłówki

```
1 <p>Lorem ipsum dolor sit amet,
2 consectetur adipisicing elit, <br />
3 sed do eiusmod tempor incididunt
4 ut labore et dolore magna aliqua. </p>
```

Nagłówki

- Nagłówki dzielą się na poziomy
- Znaczniki **h1,h2,h3,h4,h5,h6**
- Zgodnie z konwencją, nagłówek **h1** powinien oznaczać tytuł strony

Nagłówki

```
1 <h1>Naglowek 1</h1>
2 <h2>Naglowek 2</h2>
3 <h3>Naglowek 3</h3>
4 <h4>Naglowek 4</h4>
5 <h5>Naglowek 5</h5>
6 <h6>Naglowek 6</h6>
```

Listy

- Wypunktowanie **ul**
- Numerowanie **ol**
- Elementy listy **li**

Listy

```
1 <ol>
2   <li>Element 1</li>
3   <li>Element 2</li>
4   <li>Element 3</li>
5   <li>Element 4</li>
6 </ol>
```

Tabele

- Tabela **table**
- Wiersz **tr**
- Komórka **th, td**

Tabele

```
1 <table>
2  <tr><th> A </th><th> B </th></tr>
3  <tr><td> A1 </td><td> B1 </td></tr>
4  <tr><td> A2 </td><td> B2 </td></tr>
5  <tr><td> A3 </td><td> B3 </td></tr>
6 </table>
```

Odnośniki

- Aktywna zawartość strony www
- Dostęp do zasobów po kliknięciu
- Tekst odnośnika - napis (lub obrazek) w który należy kliknąć na stronie
- Cel odnośnika - atrybut **href** - adres zasobu (strony www, pliku, ...)

Odnośniki

```
1 <a href="http://www.uw.edu.pl">Uniwersytet
      Warszawski</a>
2 <a href="pobieranie/plik.pdf">Plik do pobrania</a>
```

Obrazy

- Formaty **JPG, GIF, PNG, SVG**
- Adres lokalny - ścieżka do pliku i nazwa
- Adres w Internecie - pełny adres strony i ścieżka z nazwą
- Tekst alternatywny - atrybut **alt**
  - Opis obrazka zrozumiały dla wyszukiwarek
  - Wyświetlany gdy nie można wyświetlić obrazka

Obrazy

Obrazy

```
1 <img src="images/obrazek1.jpg"
2   alt="Pierwszy obrazek"/>
3 <img src="http://www.uw.edu.pl/images/obrazek2.jpg
    "
4   alt="Drugi obrazek"/>
```

Inne elementy tekstu

- <**em**>Wzmocnienie (emfaza)</**em**>
- <**strong**>Mocne wyróżnienie</**strong**>
- <**sup**>Indeks górny</**sup**>
- <**sub**>Indeks dolny</**sub**>
- <**q**>Krótki cytat</**q**>
- <**blockquote**>Długi cytat blokowy</**blockqote**>
- <**cite**>Podanie źródła</**cite**>
- <**abbr**>Skrót</**abbr**>
- <**code**>Fragment kodu programu</**code**>

Parametry strony

- Deklaracja **DOCTYPE**
- Znaczniki w sekcji **head**

Parametry strony

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
    Transitional//EN"
2   "http://www.w3.org/TR/xhtml1/DTD/
      xhtml1-transitional.dtd">
3 <html xmlns="http://www.w3.org/1999/xhtml">
4 <head>
5 <meta http-equiv="Content-Type"
6   content="text/html; charset=utf-8" />
7 <meta name="description" content="Opis" />
8 <title>Tytul strony</title>
9 </head>
```

Parametry strony w HTML5

- Uproszczona deklaracja **DOCTYPE**
- Uproszczona deklaracja kodowania znaków **head**
- Brak odnośnika do przestrzeni nazw XML

Parametry strony

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="utf-8" />
5 <meta name="description" content="Opis" />
6 <title>Tytul strony</title>
7 </head>
```

Sekcja head

- Typ zawartości i kodowanie znaków
- **<meta charset="utf-8" />**
- Opis strony (m. in. dla wyszukiwarek)
- **<meta name="description" content="Opis" />**
- Tytuł strony (wyświetlany m. in. w nagłówku okna i wynikach wyszukiwania)
- **<title>**Tytuł strony**</title>**

Modyfikacje wyglądu

- Kaskadowe arkusze stylów
- CSS - **C**ascading **S**tyle **S**heets
- Określają dla elementów HTML m. in.
  - Rodzaj, krój i wielkość czcionki
  - Tło jako kolor lub obraz
  - Marginesy, dopełnienia
  - Układ

HTML

- HTML Służy do definiowania struktury dokumentu:
  - akapity
  - nagłówki
  - listy
  - tabele
  - odnośniki
  - sekcje
- Nie powinien służyć do definiowania formatu

Elementy blokowe i sąsiadujące

- HTML definiuje dwa rodzaje elementów (znaczników)
- **Blokowe**
  - Zawierają złamanie wiersza przed i po elemencie
  - Zajmują całą dostępną szerokość
  - **p, h1-h6, ul, ol, li, table, div**
- **Sąsiadujące**
  - Umieszczane w bieżącym wierszu
  - Szerokość zależna od zawartości
  - **a, abbr, cite, q, span**

Modyfikacje wyglądu

- Kaskadowe arkusze stylów
- CSS - **C**ascading **S**tyle **S**heets
- Określają dla elementów HTML m. in.
  - rodzaj, krój i wielkość czcionki
  - tło jako kolor lub obraz
  - marginesy, dopełnienia
  - układ (*layout*)

Oszczędność czasu i pracy

- Definicje stylów przechowywane w oddzielnych plikach
- Ujednolicenie wyglądu wszystkich podstron
- Łatwość modyfikacji
- Kompletna zmiana wyglądu za pomocą podmiany pliku ze stylami
- Style globalne i lokalne
- Style dla przeglądarek i dla wydruku

Kolejność kaskadowa

- Definicje stylów można umieszczać
  - w zewnętrznym pliku
  - w sekcji **head** dokumentu HTML
  - jako wartość atrybutu **style** znacznika HTML
- W przypadku sprzeczności obowiązującym jest styl *najbardziej szczegółowy*

Składnia

- Nazwa znacznika HTML
- Lista właściwości w nawiasach {...}
- *właściwość* : *wartość* ;

Przykładowy układ

```
1  p {
2     background-color: #EEEEEE;
3     color: #000000;
4     margin: 10px;
5  }
```

Umieszczanie definicji stylów

- w zewnętrznym pliku
- **<link type="text/css" rel="stylesheet" href="mojstyl.css" />**
- w sekcji **head** dokumentu HTML
- **<style> ... </style>**
- jako wartość atrybutu **style** znacznika HTML
- **<p style="background-color: #6C8CD5; margin: 20px;">**

Czcionki

- Właściwość **font-family**
- Nazwa czcionki, np.: *Arial, Georgia, Tahoma, Courier, ...*
- Lista czcionek na wypadek braku obsługi przez przeglądarkę
- Na końcu czcionki generyczne:
  - szeryfowa (**serif**)
  - bezszeryfowa (**sans-serif**)
  - stałej szerokości (**monospace**)

font-family

```
h1 {    font-family: Georgia, Times New Roman, serif; }
p {    font-family: Trebuchet MS, Arial, sans-serif; }
```

Właściwości czcionki

- Właściwość **font-size**
- Jednostki: **px, pt, pc, cm, mm, em, ex, %**
- Właściwość **font-weight**
- Pogrubienie: **bold**;
- Właściwość **font-style**
- Kursywa: **italic**

Właściwości czcionki

```
h2 {
  font-family: Trebuchet MS, Arial, sans-serif;
  font-size: 24px;
  font-weight: bold;
  font-style: italic;
}
```

Kolory

- Kolor tła: **background-color**
- Kolor czcionki: **color**
- Kolor obramowania: **border-color**
- Sposoby określania koloru
  - Nazwa: **white, black, green, red,...**
  - Kod RGB: **rgb(255, 255, 255), rgb(0, 0, 0), rgb(0, 200, 0), rgb(150, 0, 0), ...**
  - Kod HEX: **#FFFFFF, #000000, #336633, #CC8C00, ...**

Właściwości czcionki

```
h3 {
  background-color: #EEEEEE;
  color: #111166;
}
```

Tekst

- Wyrównanie: **text-align**
  - do lewej lub do prawej **left, right**
  - wyśrodkowanie **center**
  - wyjustowanie **justify**
- Dekoracja: **text-decoration**
  - podkreślenie **underline**
  - naddkreślenie **overline**
  - przekreślenie **line-through**
- Wcięcie pierwszego wiersza: **text-indent**

Model pudełkowy

- Elementy HTML mogą być oddzielane od siebie za pomocą
  - marginesów (**margin**)
  - odstęp od innych elementów i krawędzi okne
  - dopełnienia (**padding**)
  - odstęp zawartości od obramowania elementu
  - ramek (**border**)
  - pomiędzy marginesem a dopełnieniem

Model pudełkowy

Marginesy i dopełnienia

- Zazwyczaj dla elementów blokowych
- Jednostki: **px, cm, mm, %**
- Marginesy:
  - **margin-top, margin-right, margin-bottom, margin-left**
  - **margin: 10px;**
  - **margin: 20mm 5mm 0 5mm;**
  - **margin: 10% 5%;**
- Dopełnienia:
  - **padding-top, padding-right, padding-bottom, padding-left**
  - **padding: 10px;**
  - **padding: 20mm 5mm 0 5mm;**
  - **padding: 10% 5%;**

Obramowanie

- Zazwyczaj dla elementów blokowych
- Szerokość: **border-width**
- Jednostki: **px, cm, mm, %**
- Style obramowania: **border-style**
- **solid, dotted, dashed, double, inset**
- Położenie
  - **border-top**
  - **border-right**
  - **border-bottom**
  - **border-left**

Sekcje **div**

- Sekcje strony
- Znacznik <**div**>
- Atrubuty **class**, **id**
- Służą grupowania elementów na stronie
- Można na przykład utworzyć sekcje
  - nagłówek
  - lewa kolumna
  - obszar główny
  - stopka
  - pojemnik na wszystko
- Formatowanie sekcji za pomocą stylów

Sekcje **div**

Przykładowy układ

```
1  <div id="pojemnik">
2    <div id="naglowek">
3      ...
4    </div>
5    <div id="lewa-kolumna">
6      ...
7    </div>
8    <div id="zawartosc">
9      ...
10   </div>
11   <div id="stopka">
12     ...
13   </div>
14 </div>
```

Sekcje w HTML5

Przykładowy układ

```
1   <div id="pojemnik">
2     <header>
3       ...
4     </header>
5     <nav>
6       ...
7     </nav>
8     <article>
9       ...
10    </article>
11    <footer>
12    ...
13    </footer>
14  </div>
```

Formatowanie sekcji

- Sekcje strony są formatowane za pomocą stylów
  - Układ
  - Wymiary
  - Marginesy
  - Obramowania
  - Tło

Formatowanie sekcji

body, #pojemnik, #naglowek

```
1   body {
2    background: #666666;
3    color: #000000; }
4   #pojemnik {
5    width: 780px;
6    background: #FFFFFF;
7    margin: 0 auto;
8    border: 1px solid #000000; }
9   header {
10   background: #876ED7;
11   padding: 0 10px 0 20px; }
```

Formatowanie sekcji

#lewa-kolumna, #zawartosc, #stopka

```
1  nav {
2    float: left;
3    width: 200px;
4    background-color: #FFD073;
5    padding: 15px 10px 15px 20px; }
6  article {
7    margin: 0 0 0 250px;
8    padding: 0 20px; }
9  footer {
10   padding: 0 10px 0 20px;
11   background-color: #FFFF73; }
```

Efekt końcowy

JavaScript

1. Język skryptowy
2. Interpretowany przez przeglądarkę
3. Kod jest zawarty w dokumencie HTML lub oddzielnym pliku
4. Umożliwia budowę stron interaktywnych
5. Umożliwia modyfikacje strony w reakcji na działania użytkownika
6. Nie ma nic wspólnego z językiem Java

Przykłady

Wyświetlanie daty

```
1 <script type="text/javascript">
2 document.write("<p>" + Date() + "</p>");
3 </script>
```

Powrót do poprzedniej strony

```
1 <a href="javascript:history.back()">Powrot</a>
```

Przykłady c.d.

Wyświetlanie napisu w zależności od pory dnia

```
1  <script type="text/javascript">
2  var d=new Date();
3  var time=d.getHours();
4  if (time<12) {
5    document.write("<p>Good morning</p>");
6  }
7  else if (time > 12 && time < 17) {
8    document.write("<p>Good morning</p>");
9  }
10 else {
11   document.write("<p>Good evening</p>");
12 }
13 </script>
```

JavaScript i DOM

1. Odnajdowanie elementów
2. Modyfikacja struktury
3. Dodawanie nowych elementów

Język PHP

1. **P**HP: **H**ypertext **P**reprocessor
2. Język skryptowy interpretowany na serwerze www
3. Kod może być zanurzany w plikach HTML
4. Służy to tworzenia interaktywnych aplikacji internetowych
5. Skrypty PHP generują kod HTML wysyłany do klienta przez serwer

Przykład

Witaj świecie - HTML + PHP

```
1 <html>
2     <head>
3         <title>PHP: Witaj swiecie</title>
4     </head>
5     <body>
6         <p><?php echo "Witaj swiecie!"; ?></p>
7     </body>
8 </html>
```

Przykład c.d.

Witaj świecie - wygenerowany HTML

```
1 <html>
2     <head>
3         <title>PHP: Witaj swiecie</title>
4     </head>
5     <body>
6         <p>Witaj swiecie!</p>
7     </body>
8 </html>
```

Widok w oknie przeglądarki

Witaj świecie!

Jak działa PHP?

1. Serwer www dostaje żądanie zasobu zawierającego kod PHP
2. Przekazuje zasób do interpretera PHP, który wykonuje instrukcje zawarte w znacznikach <?php i ?>
3. Interpreter PHP zastępuje fragmenty w znacznikach <?php i ?> przez kod HTML wygenerowany przez skrypty
4. Serwer www dostaje wygenerowany kod HTML od interpretera PHP
5. Kod HTML jest wysyłany do klienta

Przetwarzanie dokumentu PHP

1. Znaczniki HTML poza fragmentami ograniczonymi <?php i ?> są przepisywane bez zmian
2. Kod otoczony <?php i ?> jest interpretowany, wykonywany i zastępowany przez wypisywany tekst

Przetwarzanie dokumentu PHP

```
1 <h1>Tytul</h1>
2 <?php echo '<h2>Podtytul</h2>'; ?>
3 <p>Dalszy ciag tekstu.</p>
4 <p><?php echo 'Zakonczenie'; ?></p>
```

Co można zrobić przy pomocy PHP?

1. Używać wielokrotnie fragmentów kodu w różnych plikach HTML
2. Przetwarzać dane z formularzy
3. Wyświetlać zawartość z bazy danych na serwerze
4. Wysyłać wiadomości e-mail
5. Generować obrazy
6. **Wykonywać operacje na plikach**

Instrukcje, zmienne, komentarze

Koniec instrukcji - ;

```
1 echo 'Witaj swiecie';
```

Zmienne - $

```
1 echo $napis;
```

Komentarze - // lub /*...*/

```
1 echo 'Witaj'; //  Komentarz do konca linii \\
2 /* Komentarz \\
3 w kilku \\
4 liniach */
```

Instrukcja warunkowa

if-else

```
1 <?php
2 if ($zmienna1 == $zmienna2 ) {
3    echo '<p>Zmienne sa rowne</p>';
4 } else {
5    echo '<p>Zmienne nie sa rowne</p>';
6 }
7 ?>
```

Instrukcja warunkowa w kawałkach

if-else

```
1 <?php if ($zmienna1 == $zmienna2) {?>
2   <p>Zmienne sa rowne.</p>
3 <?php } else { ?>
4   <p>Zmienne nie sa rowne.</p>
5 <?php } ?>
```

Typy danych

1. Cztery typy skalarne
   1. `boolean`
   2. `integer`
   3. `float`
   4. `string`
2. Dwa typy złożone
   1. `array`
   2. `object`
3. Dwa typy specjalne
   1. `resource`
   2. `NULL`

Typy danych - przykłady

```php
$logiczna = TRUE;
$napis1 = "Tekst";
$napis2 = 'Inny tekst';
$liczba1 = 12;
$liczbaNapis = "15";
$wynik = $liczba1 + $liczbaNapis;
echo $wynik; // Zostanie wypisane 27
```

Wartości logiczne

1. Słowa kluczowe `TRUE` i `FALSE`
2. Wielkość liter nie ma znaczenia
3. Liczby i napisy mogą być rzutowane na wartości logiczne
4. Jako `FALSE` uznawane są
   1. liczba całkowita 0
   2. liczba zmiennoprzecinkowa 0.0
   3. pusty łańcuch znaków ""
   4. łańcuch znaków "0"
   5. pusta tablica
   6. pusty obiekt
   7. NULL
5. Pozostałe wartości interpretowane są jako `TRUE` (w tym np. -1)

Łańcuchy znaków

Pojedynczy cudzysłów

```
1 echo 'Jakis tekst';
2 echo 'Arnold once said: "I\'ll be back"'
```

Podwójny cudzysłów - dodatkowe przetwarzanie

```
1 $zmienna = 5;
2 echo "Jakis tekst \n i dalszy ciag w nowej linii";
3 echo "Wartoscia zmiennej jest: $zmienna";
```

Operacje na łańcuchach znaków

echo

```
1 <?php echo 'Witaj swiecie';?>
```

echo ze zmienną

```
1 <?php
2   $imie = 'Tomasz';
3   echo "Witaj $imie";
4 ?>
```

Operatory

Operatory arytmetyczne

Operator	Opis	Wartość
5 - 3	odejmowanie	2
5 + 3	dodawanie	8
5 * 3	mnożenie	15
15 / 3	dzielenie	5
16 % 5	reszta z dzielenia	1

Operatory

Operatory porównania

Operator	Opis	Kiedy prawda
$a == $b	równe	$a i $b są równe bez kontroli typu
$a === $b	tożsame	$a i $b są równe i tego samego typu
$a < $b	mniejsze	$a jest mniejsze od $b
$a > $b	większe	$a jest większe od $b
$a <= $b	mniejsze równe	$a jest mniejsze lub równe $b
$a >= $b	większe równe	$a jest większe lub równe $b
$a != $b	nierówne	$a i $b są różne bez kontroli typu
$a !== $b	nietożsame	$a i $b są różne lub róznych typów
$a <> $b	nierówne	$a i $b są różne bez kontroli typu

Operatory

Operatory logiczne

Operator	Opis	Kiedy prawda
$a and $b	koniunkcja	$a i $b są prawdą
$a && $b	koniunkcja	$a i $b są prawdą
$a or $b	alternatywa	$a lub $b są prawdą
$a || $b	alternatywa	$a lub $b są prawdą
$a xor $b	alternatywa wykluczająca	$a albo $b są prawdą
!$a	negacja	$a nie jest prawdą

Operatory

Operatory przypisania

Wyrażenie	Wartości zmiennych po wykonaniu
`$a = 5`	a=5
`$b = $a = 5`	a=5, b=5
`$c = $a++`	a=6, c=5
`$d = ++$a`	a=7, c=6
`$a += 3`	a=10
`$a -= 5`	a=5
`$a *= 3`	a=15
`$a /= 5`	a=3
`$a .= ' słonie'`	a= 3 słonie

Dostęp do bazy danych

1. Istnieją rozszerzenia pozwalające na połączenia z różnymi bazami danych
   1. SQLite;
   2. MySQL
   3. PostgreSQL
   4. MSSQL
   5. Oracle
2. Do połączenia z bazą konieczne są dane użytkownika i hasło
3. PHP musi mieć uprawnienia do połączenia z bazą ze swojego serwera

Dostęp do bazy danych

Połączenie

```
1 <?php
2   $db_string = 'mysql:host=labdb.bioexploratorium.
        pl;dbname=nazwabazy;charset=utf8';
3   $username = 'nazwauzytkownika';
4   $password = 'tajnehaslo';
5   $db_conn = new PDO($db_string,$username,
        $password);
6 ?>
```

Dostęp do bazy danych c.d.

Zapytanie

```
1 <?php
2   $pracownicy_sql = "SELECT * FROM pracownicy";
3   $pracownicy_stmt = $db_conn->prepare(
        $pracownicy_sql);
4   $pracownicy_stmt->execute();
5 ?>
```

Dostęp do bazy danych c.d.

Wyświetlanie danych

```
1  <table>
2    <tr>
3    <th>Imie</th>
4    <th>Nazwisko</th>
5    </tr>
6    <?php
7    while ($pracownik = $pracownicy_stmt->fetch()) {
8      print "<tr>"
9        . "<td>" . $pracownik['imie'] . "</td>"
10       . "<td>" . $pracownik['nazwisko'] . "</td>"
11       . "</tr>";
12     }
13   ?>
14 </table>
```

Dostęp do bazy danych c.d.

Wygenerowany kod HTML

```
<table>
  <tr>
    <td>1</td><td>Anna</td><td>Babacka</td>
  </tr>
  <tr>
    <td>2</td><td>Matylda</td><td>Babacka</td>
  </tr>
  <tr>
    <td>3</td><td>Tomasz</td><td>Cabacki</td>
  </tr>
</table>
```

Dostęp do bazy danych c.d.

Widok w przeglądarce

1	Anna	Abacka
2	Matylda	Babacka
3	Tomasz	Cabacki

Formularze w HTML

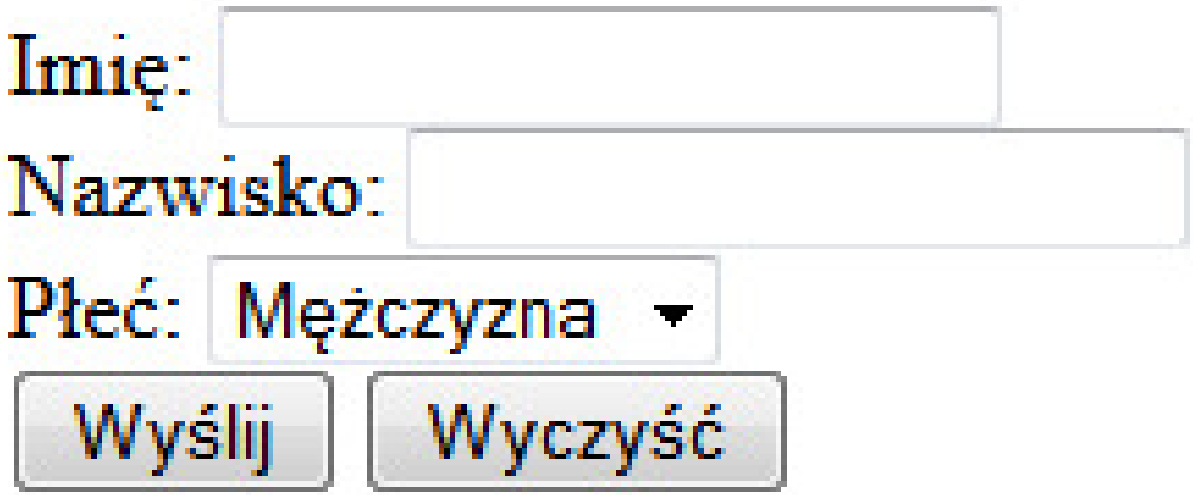
1. Znacznik `<form>`
2. Elementy formularza - znaczniki `<input>`
3. Typ pola określany przez atrybut `type`
   1. Pole tekstowe `<input type=''text''>`
   2. Pole jednokrotnego wyboru (radio) `<input type=''radio''>`
   3. Pole wielokrotnego zaznaczenia (checkbox) `<input type=''checkbox''>`
   4. Pole listy rozwijanej `<select><option>...</option>...<option>...</option></select`
   5. Pole hasła (nie wyświetla wpisywanych znaków) `<input type=''password''>`
   6. Przycisk wysyłania `<input type=''submit''>`
   7. Przycisk czyszczenia `<input type=''reset''>`

Przykładowy formularz

Pytanie o imię, nazwisko i płeć

```
1  <form>
2  Imie: <input type="text" name="imie" /><br />
3  Nazwisko: <input type="text" name="nazwisko" /><br
       />
4  Plec:
5  <select type="radios" name="plec" />
6    <option>Mezczyzna</option>
7    <option>Kobieta</option>
8  </select><br />
9  <input type="submit" value="Wyslij" />
10 <input type="reset" value="Wyczysc" />
11 </form>
```

Widok w przeglądarce



Co się dzieje z danymi?

1. Dane z formularza wysyłane są do serwera przy po kliknięciu w przycisk `submit`
2. Dwie metody
   1. **GET** - dane zakodowane w adresie URL
   2. **POST** - dane zapisane treści żądania do serwera
3. Konieczne jest podanie skryptu, który zajmie się danymi otrzymanymi przez serwer
4. Atrybut **action**
5. Ścieżka dostępu i nazwa pliku, który przetwarza dane

Wysyłanie danych metodą POST

```
1 <form action="witaj.php" method="POST">
2   Imie: <input type="text" name="imie" /><br />
3   Wiek: <input type="text" name="wiek" /><br />
4   <input type="submit" value="Wyslij" />
5 </form>
```

Dostęp do danych POST

witaj.php

```
1  <html>
2    <head>
3    </head>
4    <body>
5      <p>Witaj <?php echo $_POST["imie"]; ?>!</p>
6  <p>
7    Twoje nazwisko to <?php echo $_POST["nazwisko"];
         ?>.
8  </p>
9    </body>
10 </html>
```

Widok w przeglądarce po wysłaniu

Witaj Tomasz! Twoje nazwisko to Cabacki.

Wysyłanie danych metodą GET

```
1 <form action="witaj.php" method="GET">
2   Imie: <input type="text" name="imie" /><br />
3   Wiek: <input type="text" name="wiek" /><br />
4   <input type="submit" value="Wyslij" />
5 </form>
```

Adres URL wysłany do serwera

http://mojastrona.com/welcome.php?imie=Tomasz&wiek=32

Dostęp do danych GET

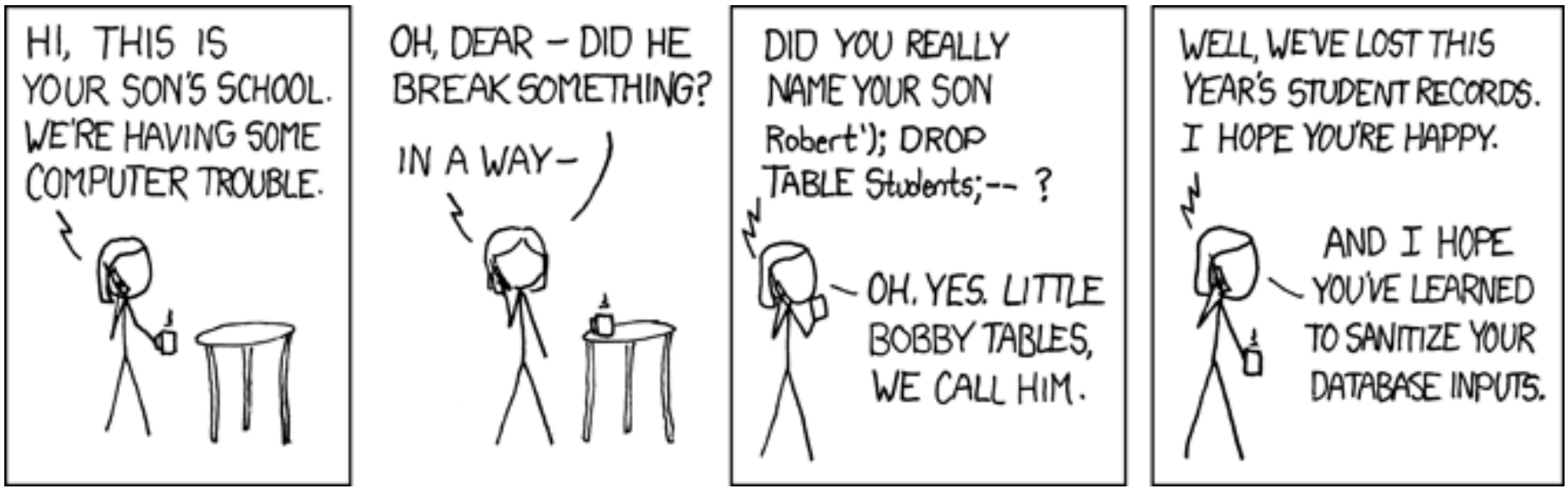
witaj.php

```
1 <html>
2   <head>
3   </head>
4   <body>
5     <p>Witaj <?php echo $_GET["imie'"]; ?>!</p>
6     <p>Masz <?php echo $_GET["wiek"]; ?> lat.</p>
7   </body>
8 </html>
```

Widok w przeglądarce po wysłaniu

Witaj Tomasz! Masz 32 lat.

SQL injection ;)



Rysunek: http://xkcd.com/327/