CHƯƠNG 2: TỔNG QUAN VỀ NGÔN NGỮ PHP

2.1. GIỚI THIỆU VỀ PHP

2.2. CÚ PHÁP

2.3. CÁC KIỂU DỮ LIỆU

2.4. BIẾN VÀ HẰNG

2.5. PHÉP GÁN VÀ CÁC PHÉP TOÁN

2.6. TRUY CẬP ĐẾN FORM

2.7. CÁC CẤU TRÚC ĐIỀU KHIỂN

2.1. GIỚI THIỆU VỀ PHP

- **Php là gì?**
- **Đặc điểm của file php**
- **Lịch sử phát triển**
- **Download, cài đặt và cấu hình ứng dụng php**
- **Quá trình thông dịch trang php**

Php là gì?

- **PHP** được viết tắt của chữ **P**ersonal **H**ome **P**age
- Là ngôn ngữ kịch bản trình chủ (Server Script) chạy trên phía máy chủ (Server side) giống như các server script khác: asp, jsp, cold fusion, …
- Là kịch bản cho phép chúng ta xây dựng ứng dụng web trên mạng internet hay intranet tương tác với mọi cơ sở dữ liệu như: Informix, MySQL, PostgreSQL, Oracle, Sybase, SQL Server,…
- Là phần mềm mở, dùng cho mục đích tổng quát. Thích hợp với Web và có thể dễ dàng nhúng vào trang HTML

Đặc điểm của file php

- Các file PHP trả về kết quả cho trình duyệt là một trang thuần HTML
- Các file PHP có thể chứa văn bản (Text), các thẻ HTML (HTML tags) và các đoạn mã kịch bản (Script)
- Các file PHP có phần mở rộng là: .php, .php3, . Phpml
- Lưu ý rằng, từ phiên bản 4.0 trở về sau mới hỗ trợ session

Lịch sử phát triển

- Năm 1995, phiên bản đầu tiên ra đời có tên là PHP/FI được viết bởi nhà phát triển phần mềm Rasmus Lerdorf.
- PHP/FI, viết tắt từ "Personal Home Page/Forms Interpreter", bao gồm một số các chức năng cơ bản của PHP ngày nay.
- Năm 1997, phiên bản PHP/FI 2.0 ra đời nhưng chỉ được công bố dưới dạng các bản beta. Đến tháng 11 năm 1997 mới chính thức được công bố
- Năm 1998, phiên bản PHP 3.0 được chính thức công bố

Lịch sử phát triển

- Andi Gutmans và Zeev Suraski tiếp tục hoàn tất phần lõi nhằm cải tiến PHP 3.0.
- Tháng 05/2000, phiên bản PHP 4.0 với hàng loạt các tính năng mới bổ sung, đã chính thức được công bố
- 29/06/2003, phiên bản PHP 5 Beta 1 đã chính thức được công bố
- Tháng 10/2003, phiên bản Beta 2 ra mắt với sự xuất hiện của hai tính năng rất được chờ đợi: Iterators, Reflection nhưng namespace một tính năng gây tranh cãi khác đã bị loại khỏi mã nguồn

Lịch sử phát triển

- Ngày 21/12/2003: phiên bản PHP 5 Beta 3 đã được công bố
- Ngày 13/07/2004, phiên bản PHP 5 bản chính thức đã ra mắt sau một chuỗi khá dài các bản kiểm tra thử bao gồm Beta 4, RC 1, RC2, RC3
- Ngày 14/07/2005, phiên bản PHP 5.1 Beta 3 được PHP Team công bố đánh dấu sự chín muồi mới của PHP với sự có mặt của PDO
- Hiện nay, phiên bản tiếp theo của PHP đang được phát triển, PHP 6 bản sử dụng thử đã có thể được download tại địa chỉ http://snaps.php.net

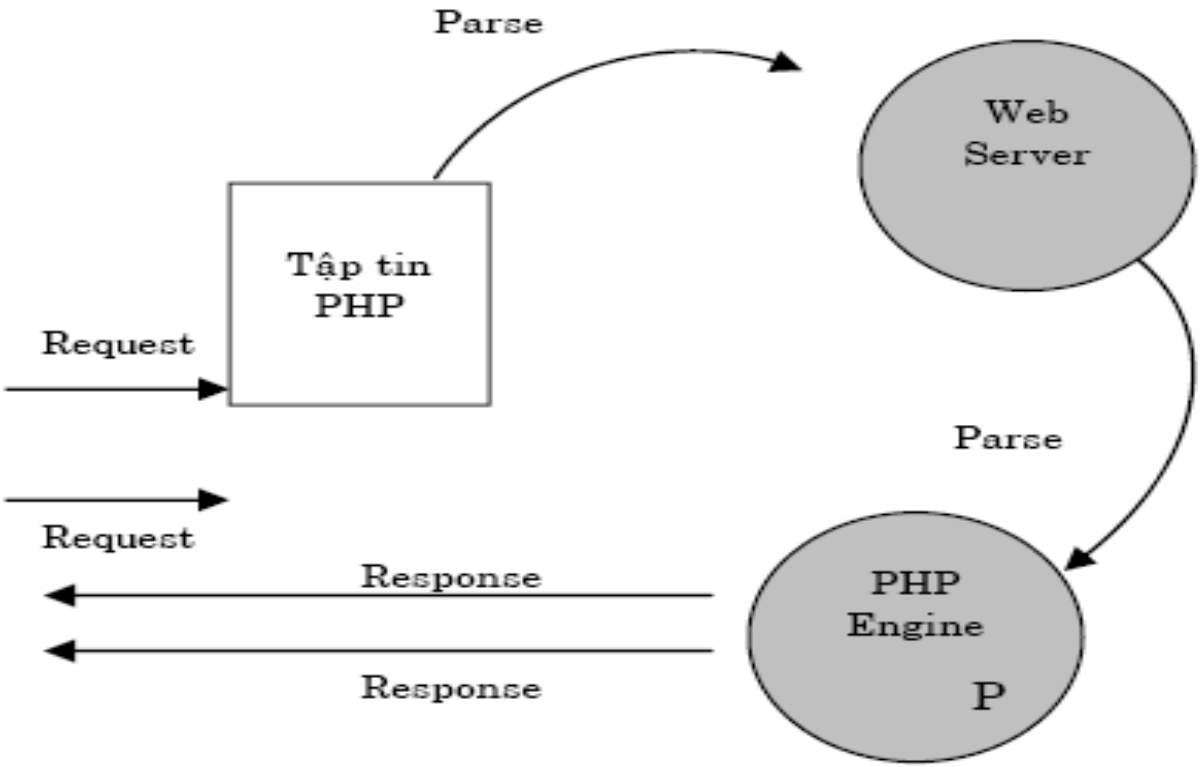
Download, cài đặt và cấu hình ứng dụng php

- Nếu máy chủ chưa được hỗ trợ PHP thì cần phải cài đặt nó.
- Download miễn phí tại: http://www.php.net/downloads.php
- Để truy cập được vào Web server có hỗ trợ PHP, cần:
  - Cài đặt Apache hoặc IIS trên máy chủ, cài PHP, MySQL
  - Hoặc thuê một Web hosting có hỗ trợ PHP và MySQL
- Có thể sử dụng một số phần mềm tích hợp sẵn Apache, php, MySQL. Chẳng hạn, như XAMPP download tại: www.apachefriends.org

Quá trình thông dịch trang php

- Php là kịch bản trình chủ được chạy trên nền php Engine, cùng với ứng dụng Web Server để quản lý chúng.
- Khi trang php được gọi, Web Server triệu gọi php Engine để thông dịch, dịch trang php và trả về kết quả cho người sử dụng là một trang thuần HTML
- Ta có mô hình như sau:

Quá trình thông dịch trang php



Hình 1-9: Quá trình thông dịch trang PHP

2.2. CÚ PHÁP

- Ta có thể nhúng các lệnh của php vào trang HTML
- Đoạn mã php luôn được bắt đầu và kết thúc bởi cặp thẻ theo cú pháp:

```
<?php
 các lệnh của php;
?>
```

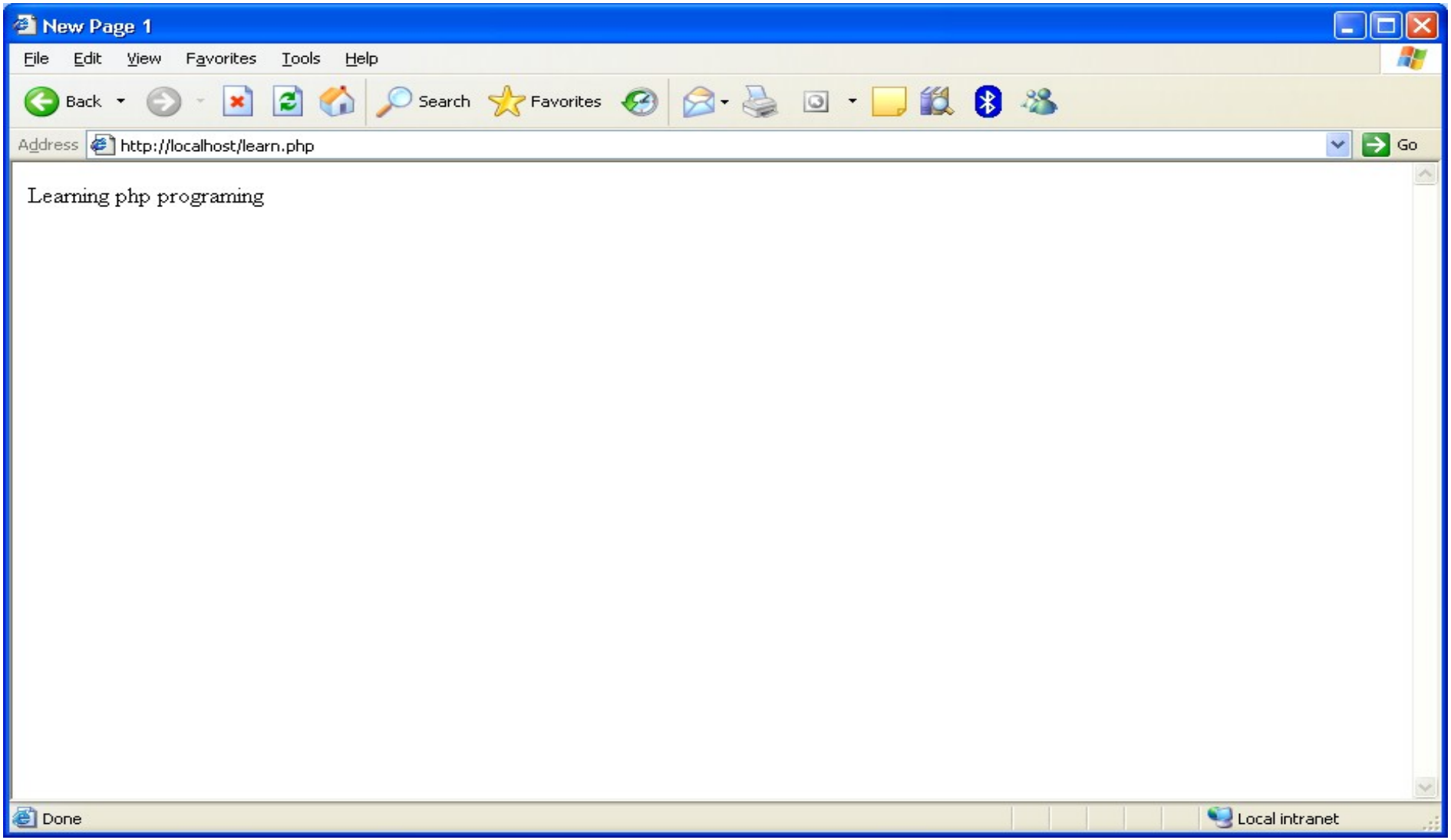
- Đoạn mã php có thể đặt bất kỳ đâu trong tài liệu
- Thông thường một trang php bao gồm các thẻ HTML như một trang HTML nhưng có thêm các đoạn mã php

2.2. CÚ PHÁP

Ví dụ: Ta có đoạn mã php hiển thị câu “Learning php programing” lên trình duyệt như sau:

```
<html>
<head>   </head>
<body>
<?php
    echo “Learning php programing";
?>
</body>
</html>
```

2.2. CÚ PHÁP


New Page 1
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/learn.php
Go
Learning php programing
Done
Local intranet

2.2. CÚ PHÁP

- Mỗi câu lệnh trong php được kết thúc bằng dấu (;). Dấu này là một toán tử dùng để phân biệt các cấu trúc với nhau
- Có hai câu lệnh cơ bản dùng để hiển thị các câu text ra browser là : ***echo*** và ***print***

2.2. CÚ PHÁP

Lưu ý:

- Các file php phải có phần mở rộng là .php. Nếu phần mở rộng là .html thì đoạn mã php sẽ không được thực thi
- Có thể viết các câu chú thích cho đoạn mã php. Có hai cách viết là: */* chú thích */* hoặc *// chú thích*
- Đoạn mã php cũng có thể đặt trong cặp thẻ:

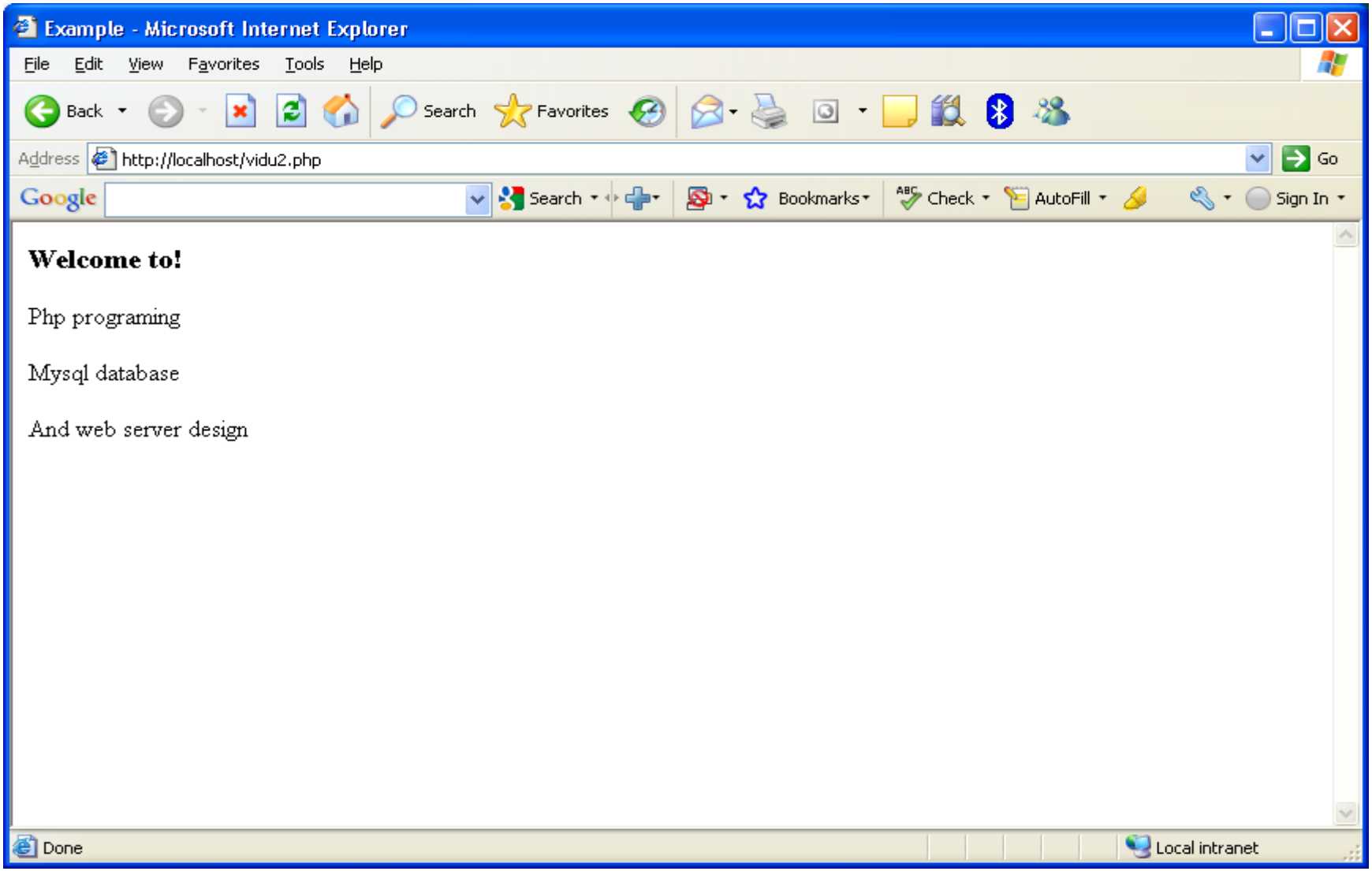
  <? Đoạn mã php ?>

2.2. CÚ PHÁP

Ví dụ: Ta có trang vidu2.php như sau:

```
<html>
<head>
<title>Example</title>
</head>
<h3> Welcome to!</h3>
<body>
<?php
    echo "<p>Php programing</p>";?>
<?php print "<p>Mysql database</p>"; ?>
<p>And web server design</p>
</body>
</html>
```

2.2. CÚ PHÁP


Example - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/vidu2.php
Go
Google Search Bookmarks Check AutoFill Sign In
Welcome to!
Php programing
Mysql database
And web server design
Done
Local intranet

2.3. CÁC KIỂU DỮ LIỆU

- Php hỗ trợ 5 kiểu dữ liệu như sau:
  - ***Integer**: sử dụng cho giá trị có kiểu dữ liệu là số nguyên*
  - ***Double**: sử dụng cho giá trị có kiểu dữ liệu là số thực*
  - ***String**: sử dụng cho các giá trị có kiểu dữ liệu là chuỗi và ký tự*
  - ***Array**: sử dụng cho các giá trị có kiểu dữ liệu là mảng*
  - ***Object**: sử dụng cho các giá trị có kiểu dữ liệu là đối tượng của lớp*

2.4. BIẾN VÀ HẰNG TRONG PHP

- **Biến**
- **Hằng**

Biến

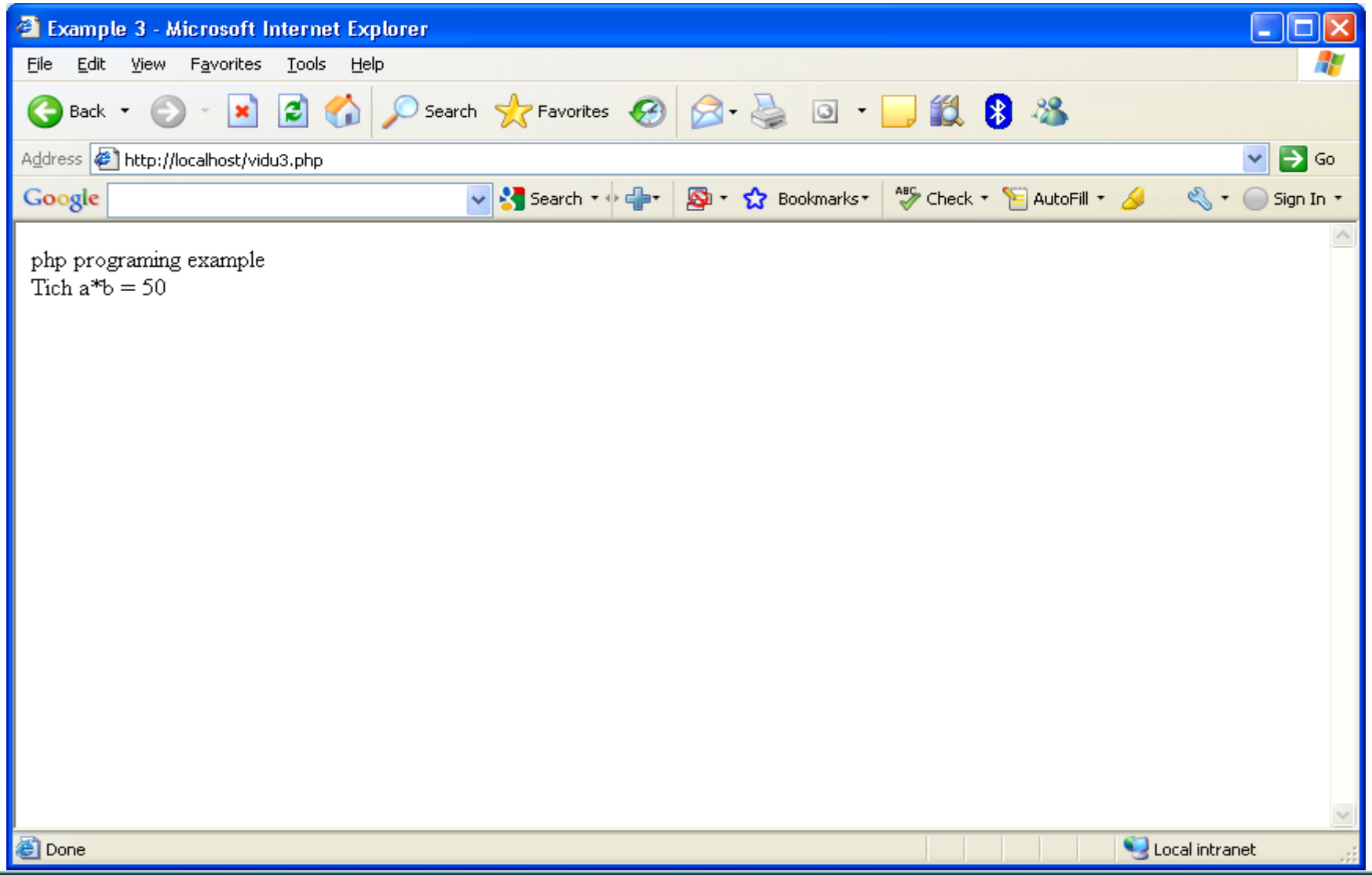
- Biến dùng để lưu giá trị như: xâu, số, ký tự, mảng,…
- Tất cả các biến trong php đều bắt đầu bằng ký hiệu **$**
- Biến được khai báo tự động khi sử dụng (gán giá trị) cho nó theo cú pháp: **$Tên_biến = Giá_trị;**
- Php là ngôn ngữ không định kiểu, nghĩa là không cần khai báo kiểu cho biến. Php sẽ chuyển kiểu của biến một cách tự động tùy thuộc vào giá trị của nó
- Lưu ý: Quy tắc đặt tên biến trong php giống như trong C, C++, …

Biến

Ví dụ: Ta có trang vidu3.php như sau:

```
<html>
<head>
<title>Example 3</title>
</head>
<?php $a = "php programing example"; echo "$a
    <br>";
    $a = 5;  $b = 10; $c = $a*$b;
    echo "Tich a*b = $c";
?>
</body>
</html>
```

Biến


Example 3 - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/vidu3.php
Go
Google Search Bookmarks Check AutoFill Sign In
php programing example
Tich a*b = 50
Done
Local intranet

Biến

- Phạm vi của biến:
  - Nếu biến được khai báo trong Script thì có phạm vi trong toàn Script
  - Nếu biến được khai báo trong một hàm nào đó thì chỉ có tác dụng trong hàm đó
- Kiểm tra/ loại bỏ biến:
  - Sử dụng hàm **isset(Tên_biến)** để kiểm tra biến đó có tồn tại hay không? Kết quả trae về kiểu boolean
  - Sử dụng hàm **unset(Tên_biến)** để loại bỏ biến đang tồn tại ra khỏi trạng thái thực thi
  - Sử dụng hàm **empty(Tên_biến)** để kiểm tra biến tồn tại và không rỗng

Biến

Ví dụ: Xét đoạn mã sau đây:

```
<?php
    $a = 10;
    echo  empty($a)."</br>";
    echo isset($a)."</br>";
    unset($a);
    echo isset($a)."</br>";
?>
```

Biến

- Ngoài ra, để kiểm tra kiểu dữ liệu của biến ta có thể sử dụng các hàm sau:
  - is_array()
  - is_double()
  - is_float()
  - is_long()
  - is_int()
  - is_string()
  - is_object()

Hằng

- Hằng trong php được khai báo giống như các ngôn ngữ C, C++
- Tên hằng thường được viết bằng chữ hoa
- Cú pháp khai báo hằng như sau:

define (“Tên_hằng”, giá_trị);

hoặc **define (Tên_hằng, giá trị);**

Ví dụ: Ta có khái báo hằng MAX = 100 như sau:

define (“MAX”, 100);

define (MAX, 100);

2.5 CÁC PHÉP TOÁN

- Phép gán
- Các phép toán số học
- Các phép toán quan hệ
- Các phép toán logic
- Phép toán trên chuỗi
- Các phép toán tự tăng giảm
- Biểu thức điều kiện

Phép gán

- Phép gán là phép toán cơ bản của mọi ngôn ngữ lập trình.
- Phép gán đơn: cú pháp: $Tên_biến = Giá_trị;
- Phép gán mở rộng: $Tên_biến **pt=** Giá_trị;
- Trong đó: **pt** có thể là: **+, -, *, /, %**

Các phép toán số học

- Phép toán số học một ngôi: - (đảo dấu)
- Phép toán số học hai ngôi:

Toán tử	Tên	Ví dụ
+	Cộng	$a + $b
-	Trừ	$a - $b
*	Nhân	$a * $b
/	Chia nguyên	$a / $b
%	Chia lấy dư	$a % $b

Các phép toán quan hệ

- Các phép toán quan hệ trả về kết quả là true hoặc false

Toán tử	Tên	Ví dụ
<	Bé thua	3 < 5
<=	Bé thua hoặc bằng	A <= b
>	Lớn hơn	$a > $b
>=	Lớn hơn hoặc bằng	$a>= $b
==	Bằng	123 = “123”
===	Bằng và cùng kiểu dữ liệu	123 = “123”
!=	Khác	123 != “123”
!==	Khác kiểu dữ liệu	123!==“123”
<>	Khác	123<> “123”

Các phép toán logic

- Giống như các phép toán quan hệ các phép toán logic trả về kết quả là true hoặc false

Toán tử	Tên	Ví dụ
&&	And (và)	$a && $b
||	Or (hoặc)	$a or $b
!	Not (phủ định)	!$b

Các phép toán tự tăng giảm

- Để tăng (hoặc giảm) giá trị của một biến lên (xuống) một đơn vị có thể sử dụng phép toán tự tăng ++ và tự giảm –
- Có hai cách viết phép toán tự tăng giảm:
  - ++&Tên_biến (hoặc --$Tên_biến)
  - &Tên_biến++ (hoặc $Tên_biến--)
- Lưu ý: cần phân biệt hai cách viết trên.

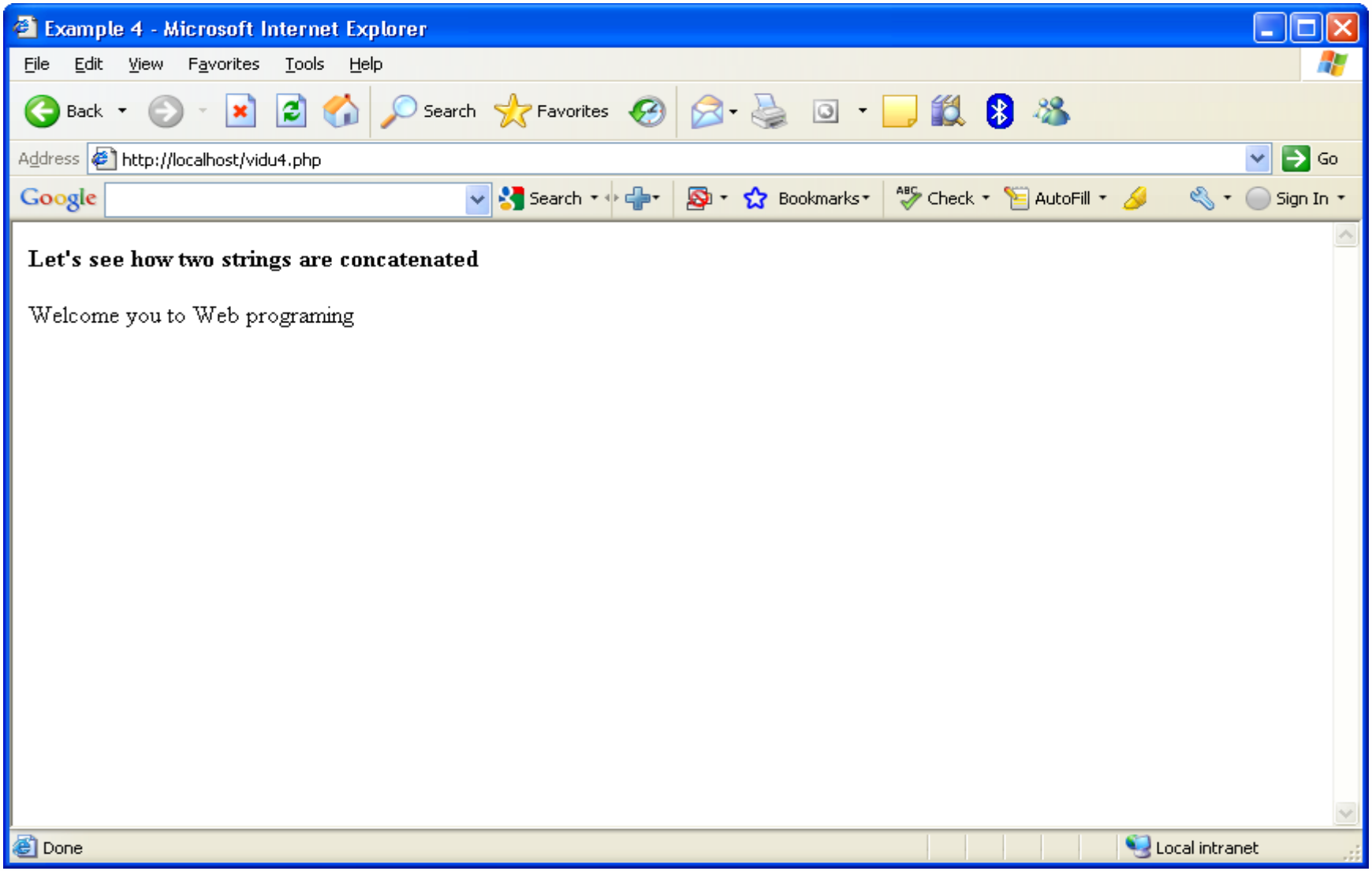
Phép toán về chuỗi

- **Phép cộng chuỗi:** Để cộng (ghép) hai chuỗi lại với nhau ta sử dụng dấu chấm (.)

Ví dụ: xét đoạn mã sau:

```
<html>
<head>
<title>Example 4</title>
</head>
<h4> Let's see how two strings are concatenated
</h4>
<?php $st1 = "Welcome you to "; $st2 = "Web
programing";
     echo $st1.$st2; ?>
</body> </html>
```

Phép toán về chuỗi



Biểu thức điều kiện

- Cú pháp:

  Biến = Giá trị 1> Giá trị 2? Giá trị 1: Giá trị 2;

Ví dụ:

```
<?php
        $a = ‘a’; $b = ‘b’;
        echo  $a > $b? $a: $b;
?>
```

2.6. TRUY CẬP ĐẾN FORM

- Form cùng với các thuộc tính của nó là nơi để người sử dụng nhập dữ liệu vào
- Để làm việc với các dữ liệu đó đòi hỏi phải kết nối đến form
- Khi làm việc với form thì các phần tử form trên trang html sẽ tự động trở thành biến trong đoạn mã php
- Để lấy giá trị từ các phần tử form ta sử dụng các hàm $_GET hoặc $_POST

Hàm $_GET

- Là hàm xây dựng sẵn dùng để lấy các giá trị từ form có sử dụng method = GET
- Thông tin khi truyền đi với phương thức GET sẽ được hiển thị trên Browser's address bar
- Mọi người có thể nhìn thấy thông tin và số ký tự tối đa là 100
- Cú pháp lấy giá trị từ các phần tử form

$_GET["Tên phần tử form"]

Hàm $_GET

Ví dụ: Ta có trang login.html như sau:

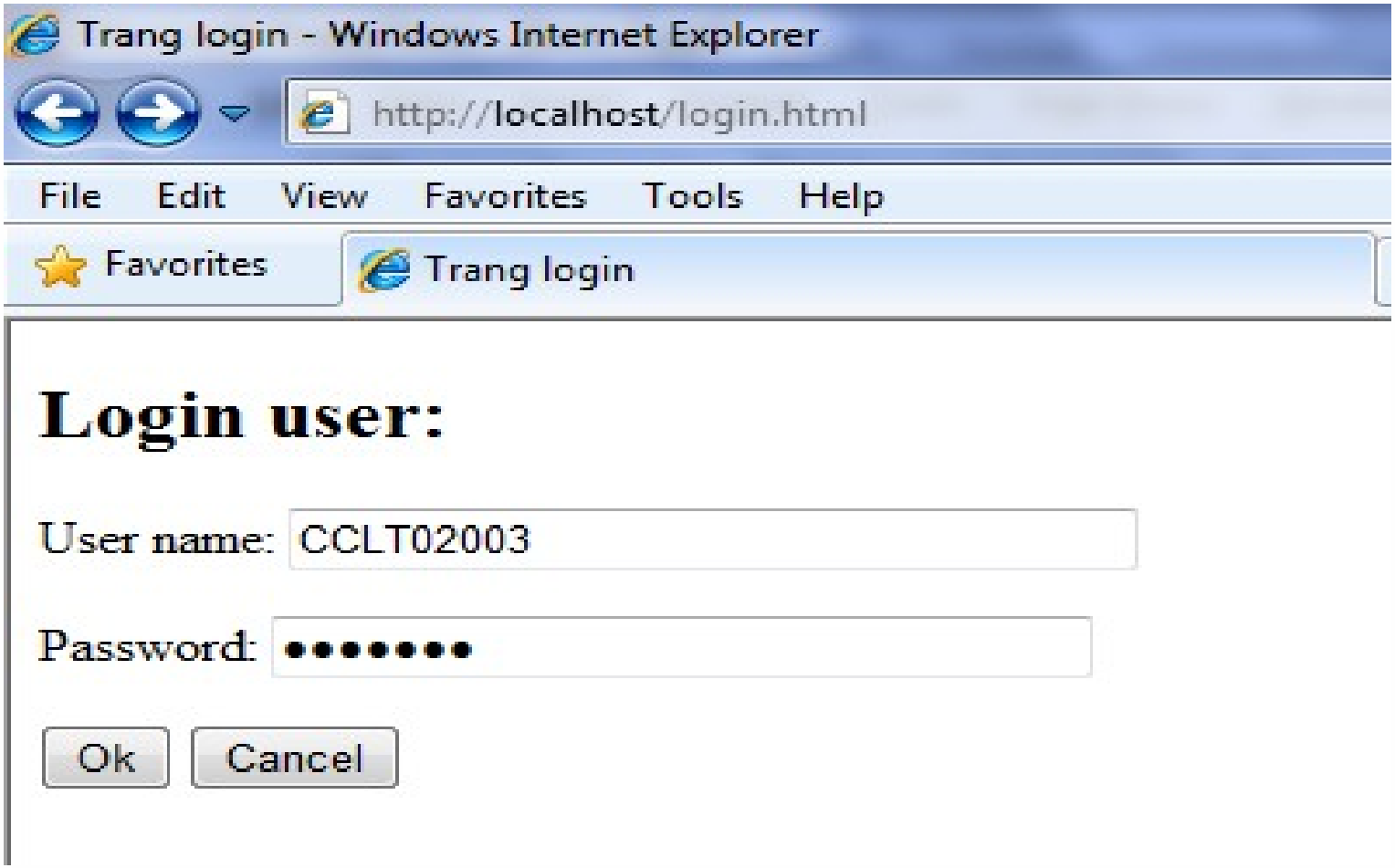
```
<h2> Login user: </h2>
<form name="f1“ method = “get” action =
    “display.php”>
<p>User name: <input type="text"
    name="username" size="35" maxlength="30"
    value=""></p>
<p>Password: <input type="password"
    name="password" size="35" value=""></p>
<p><input type="submit" name="submit"
    value="Ok" style="width: 50; height: 25">
<input type="reset" name="reset"
    value="Cancel" style="width:50;
    height:25"></p>
</form>
```

Hàm $_GET

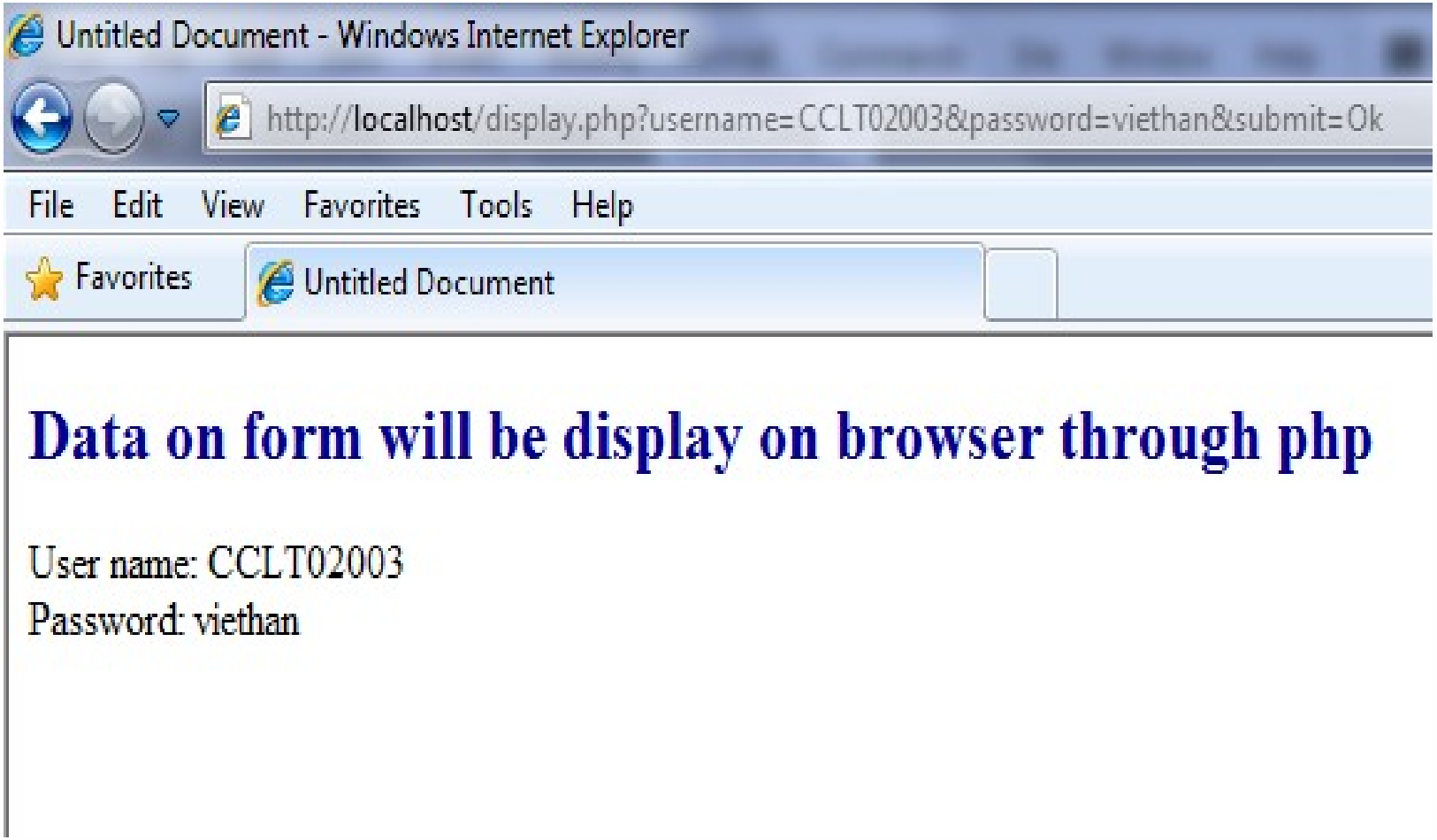
Trang display.php như sau:

```
<html>
<head>
</head>
<body>
<h2> Data on form will be display on browser
    through php</h2>
<?php
    echo "User name: ".
    $_GET["username"]."<br>";
    echo "Password: ".$_GET["password"]; ?>
</body>
</html>
```

Hàm $_GET


Trang login - Windows Internet Explorer
http://localhost/login.html
File Edit View Favorites Tools Help
Favorites
Trang login
Login user:
User name: CCLT02003
Password:
Ok
Cancel

Hàm $_GET


Untitled Document - Windows Internet Explorer
http://localhost/display.php?username=CCLT02003&password=viethan&submit=Ok
File Edit View Favorites Tools Help
Favorites
Untitled Document
Data on form will be display on browser through php
User name: CCLT02003
Password: viethan

Hàm $_POST

- Là hàm xây dựng sẵn dùng để lấy các giá trị từ form có sử dụng method = POST
- Thông tin khi truyền đi với phương thức POST sẽ không được hiển thị trên Browser's address bar
- Không thể nhìn thấy các thông tin (biến và giá trị) đang truyền.
- Cú pháp lấy giá trị từ các phần tử form

$_POST["Tên phần tử form"]

Ví dụ: Thiết kế form và trang php để giải quyết bài toán tìm nghiệm của phương trình bậc nhất, bậc hai

2.7 CÁC CẤU TRÚC ĐIỀU KHIỂN

- Cấu trúc rẽ nhánh
- Cấu trúc lặp

Cấu trúc rẽ nhánh

- **Cấu trúc if**:

Cú pháp: *if (điều kiện) câu lệnh php;*

Ví dụ:

```
<?php
    $a = 7; $b = 3;
    if ($a>$b) echo "Gia trị lớn nhất là:
    ".$a;
?>
```

Cấu trúc rẽ nhánh

Cấu trúc if … else:

Cú pháp: *if (điều kiện) công việc 1;*

Else công việc 2;

Ví dụ:
```
<?php
    $a = 7; $b = 3;
    if ($a>$b)
        echo "Gia trị lớn nhất là: ".$a;
    else
        echo "Gia trị lớn nhất là: ".$b;?>
```

Cấu trúc rẽ nhánh

Lưu ý:

- *Ta có thể sử dụng cấu trúc if lồng nhau khi có nhiều hơn 2 sự lựa chọn*
- *Nếu cần thực thi nhiều câu lệnh thì cần đặt nó trong cặp dấu ngoặc móc { }*

Cấu trúc rẽ nhánh

- **Cấu trúc switch**: sử dụng khi có nhiều sự lựa chọn

Cú pháp:

switch (*n*)
{case *label 1: code to be executed if n=label 1;* break;
 case *label 2: code to be executed if n=label 2;* break;
 …
 case *label n: code to be executed if n=label n;* break;
default: *code to be executed if n is different from label 1 to label n;*
}

Cấu trúc rẽ nhánh

Ví dụ:

```
<?php
$kt=insert;
switch($kt)
{
    case "edit":echo"sửa dữ liệu <br/>"; break;
    case "insert":echo"chèn dữ liệu<br/>";
    break;
    case "delete":echo"xoa dữ liệu<br/>"; break;
    case "save":echo"xóa dữ liệu<br/>"; break;
}
?>
```

Cấu trúc rẽ nhánh

Ví dụ:

```
<?php
$kt=insert;
switch($kt)
{
    case "edit":echo"sửa dữ liệu <br/>"; break;
    case "insert":echo"chèn dữ liệu<br/>";
    break;
    case "delete":echo"xoa dữ liệu<br/>"; break;
    case "save":echo"xóa dữ liệu<br/>"; break;
}
?>
```

Cấu trúc rẽ nhánh

Ví dụ:

```
<?php
$a=4;
switch($a)
{
    case $a%2==0: echo $a." là số chẵn";
    break;
    case $a%2!=0: echo $a." là số lẻ";
    break;
}
?>
```

Cấu trúc lặp

- **Cấu trúc While**:

Cú pháp:
```
while (condition)
{
        code to be executed;
}
```

- **Cấu trúc do … while**:

Cú pháp:
```
do
{
        code to be executed;
}
while (condition);
```

Cấu trúc rẽ nhánh

Ví dụ:

```
<?php
$a=0;
while($a<10)
{
    do   {echo $a;     $a++;    }
    while($a<5);
    echo"<br/>";  $a++;
}
?>
```

Cấu trúc lặp

- **Cấu trúc for**:

Cú pháp: for (*init; condition; increment*)
{
code to be executed;
}

- **Cấu trúc foreach**: sử dụng khi lặp trên mảng

Cú pháp: foreach ($*array* as $*value*)
{
code to be executed;
}

Cấu trúc rẽ nhánh

Ví dụ:

```
<?php
$tpho1 = array("HoChiMinh", "HaNoi", "HaiPhong",
     "DaNang");
$tpho2 = array("HCM" => "HoChiMinh", "HN" =>
     "HaNoi", "HP" => "HaiPhong", "DN" =>
     "DaNang");
foreach($tpho1 as $tp)
{
     echo $tp."<br/>";
}
foreach($tpho2 as $chiso=>$giatri)
echo $chiso.":".$giatri."<br/>";
?>
```