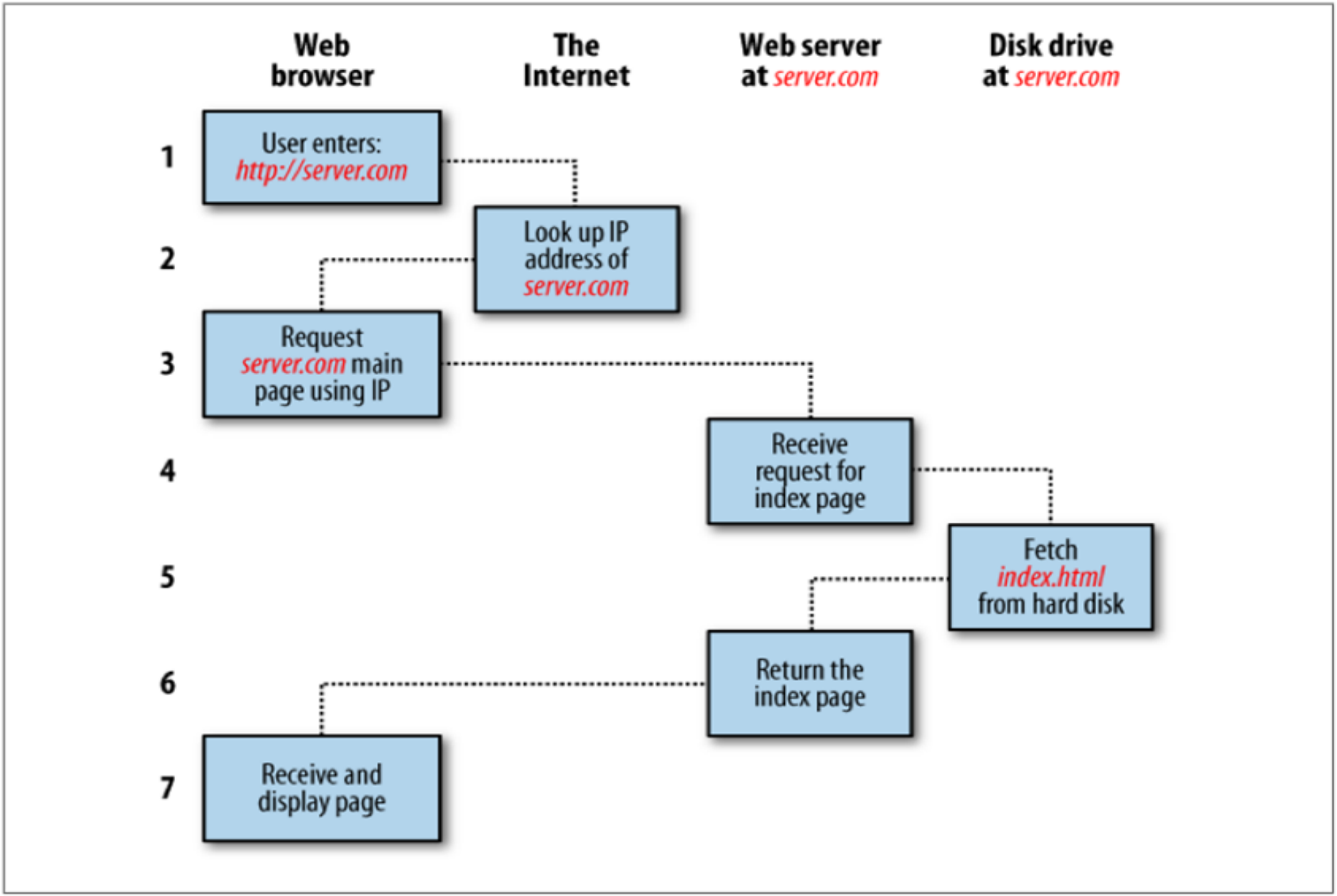
Variables and Data Type in PHP

H S Rana

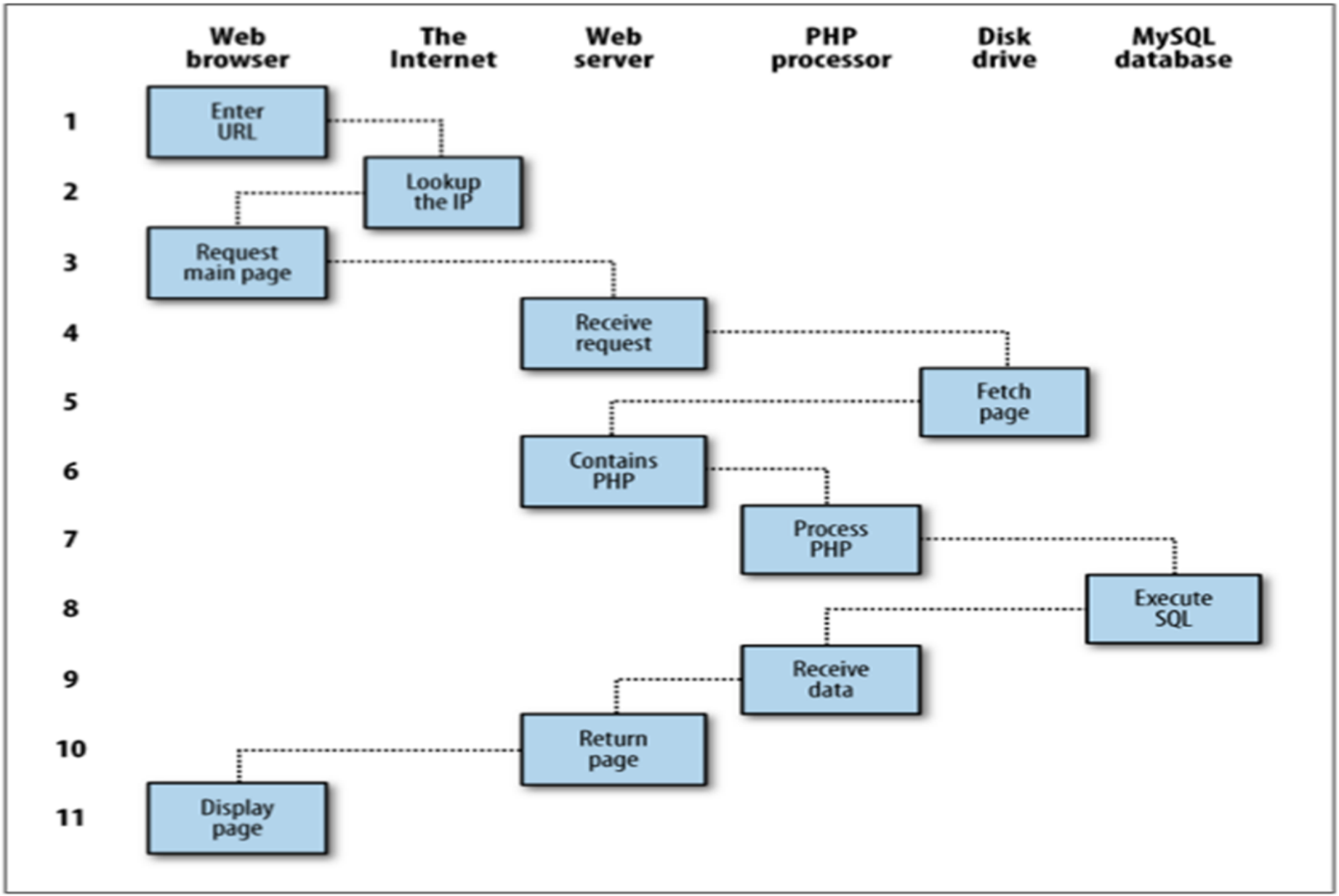
CIT,UPES

January 27, 2014

Recall...(The Request/Response Procedure)



Recall...(Request/Response Cycle for Dynamic web pages)



Internet

HTML JavaScript

AJAX CSS

Cookies

HTTP Request

Response GET

POST

PHP MySql

Templates

Recall...(What is PHP)

- PHP stands for "PHP Hypertext Preprocessor"
- an HTML-embedded server-side scripting language
- used to make web pages dynamic
  - process form information
  - authenticate users
  - provide different content depending on context
  - interface with other services: database, e-mail, etc
  - generates HTML and/or client-side scripts sent to client browsers

Recall...(Why to use)

- many other options: ASP.NET, ColdFusion, JSP...

Recall...(Why to use)

- many other options: ASP.NET, ColdFusion, JSP...
- PHP is:

Recall...(Why to use)

- many other options: ASP.NET, ColdFusion, JSP...
- PHP is:
  - free and open source: anyone can run a PHP-enabled server

Recall...(Why to use)

- many other options: ASP.NET, ColdFusion, JSP...
- PHP is:
  - free and open source: anyone can run a PHP-enabled server
  - compatible: supported by most popular web servers

Recall...(Why to use)

- many other options: ASP.NET, ColdFusion, JSP...
- PHP is:
  - free and open source: anyone can run a PHP-enabled server
  - compatible: supported by most popular web servers
  - simple: lots of built-in functionality; familiar syntax

PHP Files

- generally have .php or .phtml extensions
- generally contain both HTML and PHP
- when a client views the source, only HTML is visible
- all PHP script blocks start with <?php and end with ?>

Storing Data in Variable

- A variable is simply a container that's used to store both numeric and non-numeric information.

Storing Data in Variable

- A variable is simply a container that's used to store both numeric and non-numeric information.
- And just as with any container, you can move it from place to place, add stuff to it, or empty it out on the floor in a pile and fill it with something completely different.

Storing Data in Variable

- A variable is simply a container that's used to store both numeric and non-numeric information.
- And just as with any container, you can move it from place to place, add stuff to it, or empty it out on the floor in a pile and fill it with something completely different.
- it's a good idea to label every container.

Storing Data in Variable

- A variable is simply a container that's used to store both numeric and non-numeric information.
- And just as with any container, you can move it from place to place, add stuff to it, or empty it out on the floor in a pile and fill it with something completely different.
- it's a good idea to label every container.
- PHP has some simple rules for naming variables. Every variable name must be preceded with a dollar ($) symbol and must begin with a letter or underscore character

Storing Data in Variable

- PHP has some simple rules for naming variables. Every variable name must be preceded with a dollar ($) symbol and must begin with a letter or underscore character

Storing Data in Variable

- PHP has some simple rules for naming variables. Every variable name must be preceded with a dollar ($) symbol and must begin with a letter or underscore character
- Common punctuation characters, such as commas, quotation marks, or periods, are not permitted in variable names; neither are spaces.

Storing Data in Variable

- PHP has some simple rules for naming variables. Every variable name must be preceded with a dollar ($) symbol and must begin with a letter or underscore character
- Common punctuation characters, such as commas, quotation marks, or periods, are not permitted in variable names; neither are spaces.
- $root, $_num, and $query2 are all valid variable names

Storing Data in Variable

- PHP has some simple rules for naming variables. Every variable name must be preceded with a dollar ($) symbol and must begin with a letter or underscore character
- Common punctuation characters, such as commas, quotation marks, or periods, are not permitted in variable names; neither are spaces.
- $root, $_num, and $query2 are all valid variable names
- $58%, $1day, and email are all invalid variable names.

Assigning Value to the Variables

Syntax for assigning Value to variable as follows

1

```
VariableName=Value
$x=5
```

Question???

Question Is it possible for a variable's name itself to be a variable?

Answer Yes

Example

Example

```
<?php
// set a variable
$attribute = 'price';

// create a new variable
// its name comes dynamically
// from the value of $attribute
${$attribute} = 678;

// output: 678
echo $price;
?>
```

Destroying

To destroy a variable, pass the variable to PHP's function unset() ,

Destroying

To destroy a variable, pass the variable to PHP's function unset() ,

```
<?php
// assign value to variable
$car = 'Porsche';

// print variable value
// output: 'Before unset(), my car is a Porsche'
echo "Before unset(), my car is a $car";

// destroy variable
unset($car);

// print variable value
// this will generate an 'undefined variable' error
// output: 'After unset(), my car is a '
echo "After unset(), my car is a $car";
?>
```

Inspecting Variable Contents

PHP offers the var_dump() function, which accepts a variable and X-rays it for you.

Inspecting Variable Contents

PHP offers the var_dump() function, which accepts a variable and X-rays it for you.

```
<?php
// define variables
$name = 'Fiona';
$age = 28;

// display variable contents
var_dump($name);
var_dump($age);
?>
```

Data Types in PHP

- PHP supports eight primitive data types
- There are four scalar types
  - boolean
  - integer
  - floating-point number
  - string
- There are two structured types
  - array
  - object
- There are two special data types
  - resource
  - NULL
- The programmer does not specify the type of a variable
- a variable's type is determined from the context of its usage

Understanding PHP's Data Types

- Booleans are the simplest of all PHP data types. Like a switch that has only two states, on and off, it consists of a single value that may be set to either 1 (true) or 0 (false).
- PHP also supports two numeric data types: integers and floating-point values. Floating- point values (also known as floats or doubles) are decimal or fractional numbers, while integers are round numbers. Both may be less than, greater than, or equal to zero.
- For non-numeric data, PHP offers the string data type, which can hold letters, numbers, and special characters. String values must be enclosed in either single quotes or double quotes.
- You may also encounter the NULL data type, which is a "special" data type first introduced in PHP 4. NULLs are used to represent "empty" variables in PHP; a variable of type NULL is a variable without any data.

Booleans

- The boolean data type admits two values
  - true (case-insensitive)
  - false (case-insensitive)
- Example usage
  - $itIsRainingToday = true;
  - $thePrinterIsBusy = True;
  - $theQueueIsEmpty = FALSE;

Integers

- Integers can be specified in decimal, hexadecimal or octal notation, optionally preceded by a sign
  - In octal notation, the number must have a leading 0
  - In hexadecimal notation, the number must have a leading 0x.
- Examples
  - $a = 1234; # decimal number
  - $a = -123; # a negative number
  - $a = 0123; # octal number (equivalent to 83 decimal)
  - $a = 0x1B; # hexadecimal number (equivalent to 27 decimal)
- The maximum size of an integer is platform-dependent, but usually it's 32 bits signed – about 2,000,000,000
- PHP does not support unsigned integers.

Floating Point Numbers

- These can be specified using any of these forms:
  - $a = 1.234;
  - $a = 1.2e3;
  - $a = 7E-10;
- The maximum size of a float is platform-dependent, although most support a maximum of about 1.8e308 with a precision of roughly 14 decimal digits

String

- A string literal can be specified in three different ways:
  - single quoted
  - double quoted
  - heredoc syntax

Single-quoted Strings

- In single-quoted strings, single-quotes and backslashes must be escaped with a preceding backslash
- Example usage
  - echo 'this is a simple string';
  - echo 'You can embed newlines in strings, just like this.'; echo 'Rahul said "I\'ll be back" when leaving the Dehradun';
  - echo 'Are you sure you want to delete C:*.*?';

Double-quoted Strings

- In double-quoted strings,
  - variables are interpreted to their values, and
  - various characters can be escaped
    - \n linefeed
    - \r carriage return
    - \t horizontal tab
    - \\ backslash
    - \$ dollar sign
    - \" double quote
    - \[0-7]{1,3} a character in octal notation
    - \x[0-9A-Fa-f]{1,2} a character in hexadecimal notation

Heredoc Strings

- Heredoc strings are like double-quoted strings without the double quotes
- A heredoc string is delimited as follows
  - The string is preceded by <<< followed by a label
  - The string ends with the 2nd occurrence of the same label
  - Example usage
    - $str = <<<EOD Example of string spanning multiple lines using heredoc syntax. EOD;