Marwadi
University

Marwadi University

01IT0701 – Advance Web Technology

Unit – 2
Advance PHP

Prof. Jaydeep K. Ratanpara
Computer Engineering Department

OUTLINE

- Regular Expression
- Web Scraping using cURL
- Mail function
- Web Services & APIs

REGULAR EXPRESSION

What is Regular Expression?

- PHP is an open-source language commonly used for website creation, it provides regular expression functions as an important tool.
- Many available modern languages and tools apply regexes on very large files and strings.
- "A sequence of characters that forms a search pattern."
- When you search for data in a text, you can use this search pattern to describe what you are searching for.
- It can be a single character, or a more complicated pattern.
- To perform all types of text search and text replace operations.

REGULAR EXPRESSION

- Commonly known as a regex (regexes) A sequence of characters describing a special search pattern in the form of text string.

- The exact sequence of characters are unpredictable, so the regex helps in fetching the required strings based on a pattern definition.

- A compact way of describing a string pattern that matches a particular amount of text.

REGULAR EXPRESSION

Regular Expression Syntax :

- Regular expressions are Strings composed of **Delimiters**, A **pattern** and **Optional modifiers**.

  ```
  $exp = "/MU/i";
  ```

- Here / is the delimiter, MU is the pattern that is being searched for, and i is a modifier that makes the search case-insensitive.
- Regular expressions are known to accomplish tasks such as validating email addresses, IP address etc.
- The delimiter Can be any character that is Not a letter, number, backslash or space.
- Most common delimiter → forward slash (/)
- When any pattern contains forward slashes it is convenient to choose other delimiters such as **#** or **~**.

REGULAR EXPRESSION

Advantages and uses of Regular expressions :

- Regular expressions help in validation of text strings.
- It offers a powerful tool for analyzing, searching a pattern and modifying the text data.
- It helps in searching specific string pattern and extracting matching results in a flexible manner.
- With the help of in-built regexes functions, easy and simple solutions are provided for identifying patterns.
- It effectively saves a lot of development time, which are in search of specific string pattern.

REGULAR EXPRESSION

Advantages and uses of Regular expressions :

- It helps in important user information validations like email address, phone numbers and IP address.
- It helps in highlighting special keywords in a file based on search result or input.
- It helps in identifying specific template tags and replacing those data with the actual data as per the requirement.
- Regexes are very useful for checking password strength and form validations.

REGULAR EXPRESSION

Two types of regular expression functions

POSIX Regular Expressions

- (This feature was DEPRECATED in PHP 5.3.0, and REMOVED in PHP 7.0.0). POSIX - Portable Operating System Interface for uniX
- https://www.php.net/manual/en/book.regex.php

PCRE - PERL Compatible Regular Expressions

- https://www.php.net/manual/en/book.pcre.php

REGULAR EXPRESSION

POSIX Regex Functions

- preg_replace — Replace regular expression
- preg — Regular expression match
- pregi_replace — Replace regular expression case insensitive
- pregi — Case insensitive regular expression match
- split — Split string into array by regular expression
- spliti — Split string into array by regular expression case insensitive
- sql_regcase — Make regular expression for case insensitive match

REGULAR EXPRESSION

PCRE Functions

- preg_filter — Perform a regular expression search and replace
- preg_grep — Return array entries that match the pattern
- preg_last_error — Returns the error code of the last PCRE regex execution
- preg_match_all — Perform a global regular expression match
- preg_match — Perform a regular expression match
- preg_quote — Quote regular expression characters
- preg_replace_callback_array — Perform a regular expression search and replace using callbacks
- preg_replace_callback — Perform a regular expression search and replace using a callback
- preg_replace — Perform a regular expression search & replace
- preg_split — Split string by a regular expression

REGULAR EXPRESSION

Regular expressions & corresponding to matching strings

REGULAR EXPRESSION	MATCHES
geeks	The string "*geeks*"
^geeks	The string which starts with "*geeks*"
geeks$	The string which have "*geeks*" at the end.
^geeks$	The string where "*geeks*" is alone on a string.
[abc]	a, b, or c
[a-z]	Any lowercase letter
[^A-Z]	Any letter which is NOT a uppercase letter
(gif|png)	Either "*gif*" or "*png*"
[a-z]+	One or more lowercase letters
^[a-zA-Z0-9]{1, }$	Any word with at least one number or one letter
([ax])([by])	ab, ay, xb, xy
[^A-Za-z0-9]	Any symbol other than a letter or other than number
([A-Z]{3}|[0-9]{5})	Matches three letters or five numbers

REGULAR EXPRESSION

Marwadi University

Operators in Regular Expression:

OPERATOR	DESCRIPTION
^	It denotes the start of string.
$	It denotes the end of string.
.	It denotes almost any single character.
()	It denotes a group of expressions.
[]	It finds a range of characters for example [xyz] means x, y or z .
[^]	It finds the items which are not in range for example [^abc] means NOT a, b or c.
– (dash)	It finds for character range within the given item range for example [a-z] means a through z.
| (pipe)	It is the logical OR for example x | y means x OR y.
?	It denotes zero or one of preceding character or item range.

REGULAR EXPRESSION

Operators in Regular Expression

OPERATOR	DESCRIPTION
*	It denotes zero or more of preceding character or item range.
+	It denotes one or more of preceding character or item range.
{n}	It denotes exactly n times of preceding character or item range for example n{2}.
{n, }	It denotes at least n times of preceding character or item range for example n{2, }.
{n, m}	It denotes at least n but not more than m times for example n{2, 4} means 2 to 4 of n.
\	It denotes the escape character.

REGULAR EXPRESSION

Special Character Classes in Regular Expressions

SPECIAL CHARACTER	MEANING
\n	It denotes a new line.
\r	It denotes a carriage return.
\t	It denotes a tab.
\v	It denotes a vertical tab.
\f	It denotes a form feed.
\xxx	It denotes octal character.
\xhh	It denotes hex character.

REGULAR EXPRESSION

Regular Expressions – **Notes**

By default, regular expressions are case sensitive.

There is a difference between strings inside single quotes and strings inside double quotes in PHP.

The former are treated literally, whereas for the strings inside double-quotes means the content of the variable is printed instead of just printing their names.

REGULAR EXPRESSION

Common Regular Expression Functions

Function	Description
preg_match()	To execute a pattern match on a string. Returns 1 if the pattern was found in the string and 0 if not
preg_match_all()	Returns the number of times the pattern was found in the string, which may also be 0
preg_replace()	Returns a new string where matched patterns have been replaced with another string
preg_split()	To perform a pattern match on a string followed by the splitting of results in to numeric array.

REGULAR EXPRESSION

1. preg_match( )

Syntax : preg_match (pattern, input, matches, flags, offset)

This function returns whether a match was found in a string.

Example: Use a regular expression to do a case-insensitive search for "MU" in a string:

```
<?php
$str = "Visit MU to know MU";
$pattern = "/MU/i";
echo preg_match($pattern, $str);
?>
```

Output

```
1
```

REGULAR EXPRESSION

preg_match( )

Parameter	Description
pattern	Required. Contains a regular expression indicating what to search for
input	Required. The string in which the search will be performed
matches	Optional. The variable used in this parameter will be populated with an array containing all of the matches that were found
flags	•Optional. A set of options that change how the matches array is structured: PREG_OFFSET_CAPTURE - When this option is enabled, each match, instead of being a string, will be an array where the **first element is a substring** containing the match and the **second element** is the position of the first character of the substring in the input. •PREG_UNMATCHED_AS_NULL - When this option is enabled, unmatched sub patterns will be returned as NULL instead of as an empty string.
offset	Optional. Defaults to 0. Indicates how far into the string to begin searching. The preg_match() function will not find matches that occur before the position given in this parameter

REGULAR EXPRESSION

preg_match( ) Example-2

```
<?php
$str = "Welcome to MU for the visit.";
$pattern = "/mu/i";
echo preg_match($pattern, $str);
echo "</br>";
preg_match($pattern, $str, $matches);
print_r($matches);
echo "</br>";
preg_match($pattern, $str, $matches, PREG_OFFSET_CAPTURE);
print_r($matches);
?>
```

Output:

```
1
Array ( [0] => MU )
Array ( [0] => Array ( [0] => MU [1] => 11 ) )
```

REGULAR EXPRESSION

2. preg_match_all()

Syntax : preg_match_all(pattern, input, matches, flags, offset)

This function will tell you how many matches were found for a pattern in a string.

Example: Use a regular expression to do a case-insensitive count of the number of occurrences of "MU" in a string:

```php
<?php
$str = "Visit MU to know MU and also to see MU.";
$pattern = "/MU/i";
echo preg_match_all($pattern, $str);
?>
// Outputs 3
```

REGULAR EXPRESSION

2. preg_match_all()

Parameter	Description
pattern	Required. Contains a regular expression indicating what to search for
input	Required. The string in which the search will be performed
matches	Optional. The variable used in this parameter will be populated with an array containing all of the matches that were found
flags	•Optional. A set of options that change how the matches array is structured. One of the following structures may be selected: PREG_PATTERN_ORDER - Default. Each element in the matches array is an array of matches from the same grouping in the regular expression, with index zero corresponding to matches of the whole expression and the remaining indices for sub pattern matches. •PREG_SET_ORDER - Each element in the matches array contains matches of all groupings for one of the found matches in the string. •Any number of the following options may be applied: PREG_OFFSET_CAPTURE - When this option is enabled, each match, instead of being a string, will be an array where the first element is a substring containing the match and the second element is the position of the first character of the substring in the input. •PREG_UNMATCHED_AS_NULL - When this option is enabled, unmatched subpatterns will be returned as NULL instead of as an empty string.
offset	Optional. Defaults to 0. Indicates how far into the string to begin searching. The preg_match() function will not find matches that occur before the position given in this parameter

REGULAR EXPRESSION

2. preg_match_all( ) – Example 2

```
<?php
$str = "The rain in SPAIN falls mainly on the plains.";
$pattern = "/ain/i";
if(preg_match_all($pattern, $str, $matches))
{
  print_r($matches);
}
?>
```

Output:

```
Array
(
 [0] => Array
   (
    [0] => ain
    [1] => AIN
    [2] => ain
    [3] => ain
   )
)
```

REGULAR EXPRESSION

2. preg_match_all( ) – Example 3

```
<?php
$str = "abc ABC";
$pattern = "/((a)b)(c)/i";
if(preg_match_all($pattern, $str, $matches, PREG_PATTERN_ORDER))
{
  print_r($matches);
}
?>
```

Output:

```
Array
(
    [0] => Array
        (
            [0] => abc
            [1] => ABC
        )
    [1] => Array
        (
            [0] => ab
            [1] => AB
        )
    [2] => Array
        (
            [0] => a
            [1] => A
        )
    [3] => Array
        (
            [0] => c
            [1] => C
        )
)
```

REGULAR EXPRESSION

3. preg_replace( )

This function will replace all of the matches of the pattern in a string with another string based on pattern/expression.

Syntax: preg_replace(*patterns, replacements, input, limit, count*)

Use a case-insensitive regular expression to replace MU with Marwadi University in a string:

```
<?php
$str = "Visit MU to know MU.";
$pattern = "/MU/i";
echo $str;
echo "<br/>";
echo preg_replace($pattern, "Marwadi University", $str);
?>
```

// Outputs

```
Visit MU to know MU.
Visit Marwadi University to know Marwadi University.
```

REGULAR EXPRESSION

Marwadi University

3. preg_replace( )

Parameter	Description
patterns	Required. Contains a regular expression or array of regular expressions.
replacements	Required. A replacement string or an array of replacement strings
input	Required. The string or array of strings in which replacements are being performed
limit	Optional. Defaults to -1, meaning unlimited. Sets a limit to how many replacements can be done in each string
count	Optional. After the function has executed, this variable will contain a number indicating how many replacements were performed

REGULAR EXPRESSION

4. preg_split( )

The preg_split() function breaks a string into an array using matches of a regular expression as separators.

Syntax: preg_split(pattern, string, limit, flags)

REGULAR EXPRESSION

4. preg_split( ) – Example

```
<?php
$date = "1970-01-01 00:00:00";
$pattern = "/[-\s:]/";
$components = preg_split($pattern, $date);
print_r($components);
?>
```

Output:

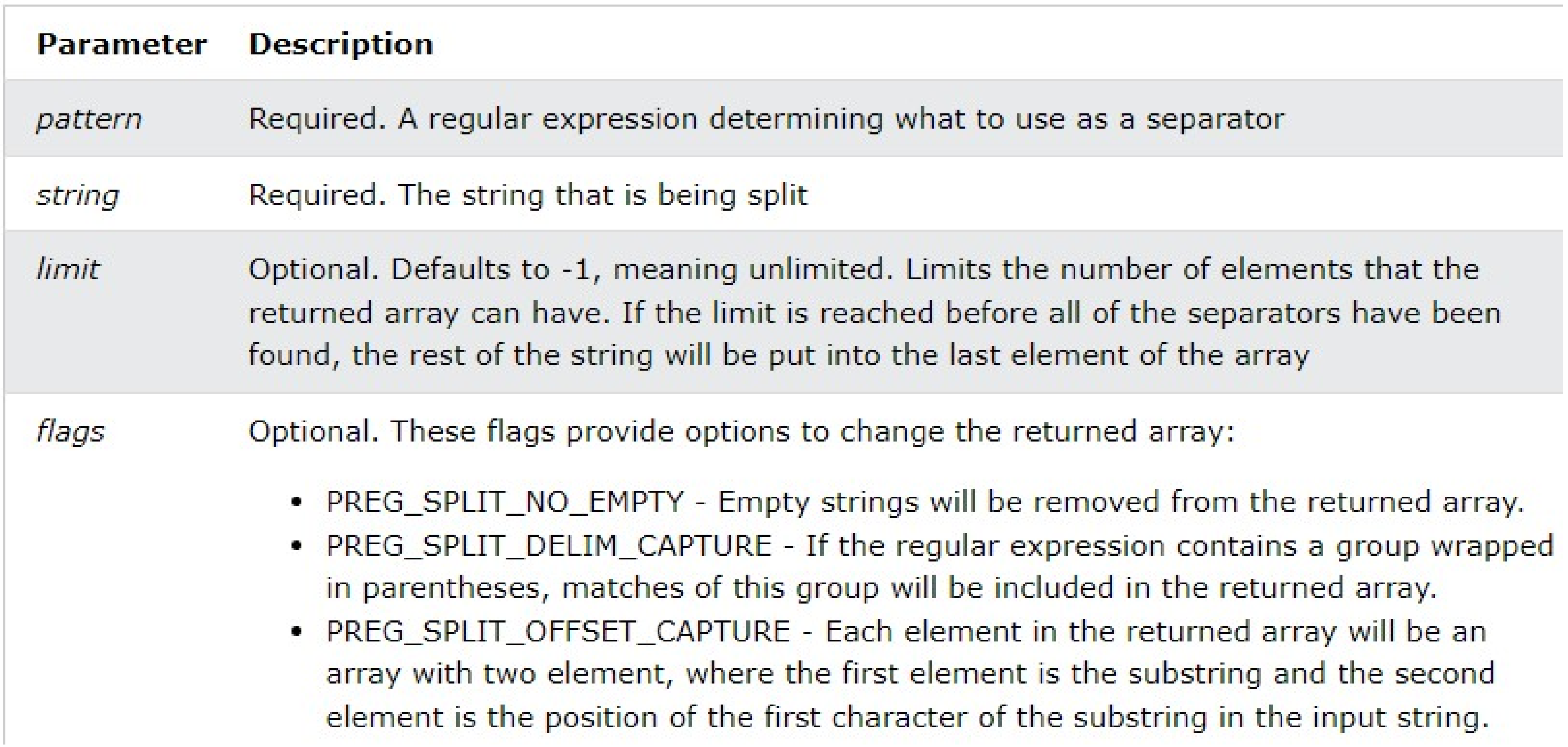
```
Array
(
    [0] => 1970
    [1] => 01
    [2] => 01
    [3] => 00
    [4] => 00
    [5] => 00
)
```

Note: The regular expression \s is a predefined character class. It indicates a single whitespace character.

REGULAR EXPRESSION

Marwadi University

4. preg_split( )



Parameter	Description
pattern	Required. A regular expression determining what to use as a separator
string	Required. The string that is being split
limit	Optional. Defaults to -1, meaning unlimited. Limits the number of elements that the returned array can have. If the limit is reached before all of the separators have been found, the rest of the string will be put into the last element of the array
flags	Optional. These flags provide options to change the returned array: • PREG_SPLIT_NO_EMPTY - Empty strings will be removed from the returned array. • PREG_SPLIT_DELIM_CAPTURE - If the regular expression contains a group wrapped in parentheses, matches of this group will be included in the returned array. • PREG_SPLIT_OFFSET_CAPTURE - Each element in the returned array will be an array with two element, where the first element is the substring and the second element is the position of the first character of the substring in the input string.

REGULAR EXPRESSION

4. preg_split( ) – Example – 2

Using the PREG_SPLIT_DELIM_CAPTURE flag:

```php
<?php
$date = "1970-01-01 00:00:00";
$pattern = "/([-\s:])/";
$components = preg_split($pattern, $date, -1, PREG_SPLIT_DELIM_CAPTURE);
print_r($components);
?>
```

Output:

```
Array
(
    [0] => 1970
    [1] => -
    [2] => 01
    [3] => -
    [4] => 01
    [5] =>
    [6] => 00
    [7] => :
    [8] => 00
    [9] => :
    [10] => 00
)
```

REGULAR EXPRESSION

5. preg_quote( )

preg_quote() takes string and puts a backslash in front of every character that is part of the regular expression syntax.

Syntax : preg_quote(input, delimiter)

```
<?php
   $keywords = '$40 for a g3/400';
   $keywords = preg_quote($keywords, '/');
   echo $keywords;
?>
```

Output:

```
\$40 for a g3\/400
```

REGULAR EXPRESSION

5. preg_quote( )

Use preg_quote() to safely use special characters in a regular expression:

The preg_quote() function adds a backslash to characters that have a special meaning in regular expressions so that searches for the literal characters can be done.

This function is useful when using user input in regular expressions.

Syntax: preg_quote(input, delimiter)

REGULAR EXPRESSION

5. preg_quote( ) – Example – 2

```
<?php
$search = preg_quote(“://”, ”/”);
echo $search;
echo “</br>”;
$input = 'https://www.w3schools.com/';
$pattern = "/$search/";
if(preg_match($pattern, $input))
{
  echo "The input is a URL.";
}
else
{
  echo "The input is not a URL.";
}
?>
```

Output:

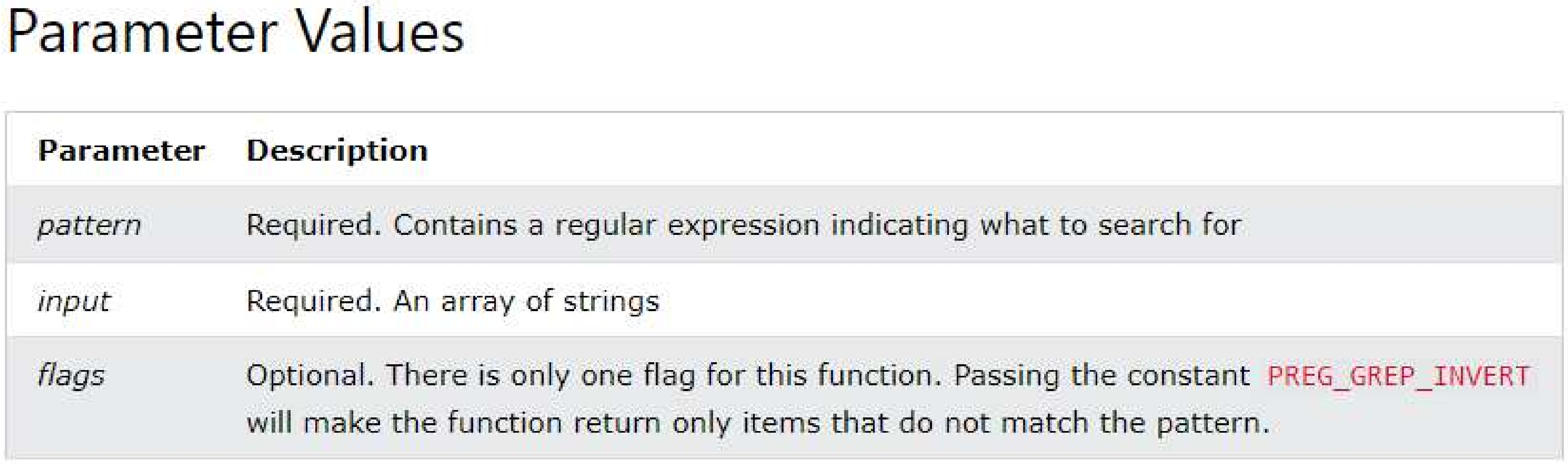
```
\:\/\/
The input is a URL.
```

REGULAR EXPRESSION

6. preg_grep()

The preg_grep( ) function returns an array containing only elements from the input that match the given pattern.

Syntax : preg_grep(pattern, input, flags)



Parameter Values

Parameter	**Description**
pattern	Required. Contains a regular expression indicating what to search for
input	Required. An array of strings
flags	Optional. There is only one flag for this function. Passing the constant `PREG_GREP_INVERT` will make the function return only items that do not match the pattern.

REGULAR EXPRESSION

6. preg_grep() – Example - Get items from an array which begin with "p":

```
<?php

$input = [ "Red", "Pink",  "Green", "Blue",  "Purple“ ];

$result = preg_grep("/^p/i", $input);
print_r($result);

?>
```

Output:

```
Array
(
    [1] => Pink
    [4] => Purple
)
```

METACHARACTERS

Meta characters

- There are two kinds of characters that are used in regular expressions these are: Regular characters and Meta characters.
- Regular characters are those characters which have a *'literal'* meaning
- Meta characters are those characters which have *'special'* meaning in regular expression.

METACHARACTERS

METACHARACTER	DESCRIPTION
.	It matches any single character other than a new line.
^	It matches the beginning of string.
$	It matches the string pattern at the end of the string.
*	It matches zero or more characters.
+	It matches preceding character appear atleast once.
\	It is used to escape meta characters in regex.
a-z	It matches lower case letters.
A-Z	It matches upper case letters.
0-9	It matches any number between 0 and 9.

META CHARACTERS

Notes:

Meta characters are very powerful in regular expressions pattern matching solutions. It handles a lot of complex pattern processing.

- Every character which is not a meta character is definitely a regular character.
- Every regular character matches the same character by itself.

METACHARACTERS

Following is the list of meta characters which can be used in PERL Style Regular Expressions.

- . a single character (dot)
- \s a whitespace character (space, tab, newline)
- \S non-whitespace character
- \d a digit (0-9)
- \D a non-digit
- \w a word character (a-z, A-Z, 0-9, _)
- \W a non-word character
- [aeiou] matches a single character in the given set
- [^aeiou] matches a single character outside the given set
- (foo|bar|baz) matches any of the alternatives specified

METACHARACTERS

QUANTIFIER	MEANING
a+	It matches the string containing at least one a.
a*	It matches the string containing zero or more a’s.
a?	It matches any string containing zero or one a’s.
a{x}	It matches letter ‘a’ exactly x times .
a{2, 3}	It matches any string containing the occurrence of two or three a’s.
a{2, }	It matches any string containing the occurrence of at least two a’s.

METACHARACTERS

QUANTIFIER	MEANING
a{2}	It matches any string containing the occurrence of exactly two a’s.
a{, y}	It matches any string containing the occurrence of not more than y number of a’s.
a$	It matches any string with ‘a’ at the end of it.
^a	It matches any string with ‘a’ at the beginning of it.
[^a-zA-Z]	It matches any string pattern not having characters from a to z and A to Z.
a.a	It matches any string pattern containing a, then any character and then another a.
^.{3}$	It matches any string having exactly three characters.

WEB SCRAPING USING cURL

What is PHP cURL?

- cURL stands for the client URL.
- PHP cURL is a library that is the most powerful extension of PHP.
- It allows the user to create the HTTP requests in PHP.
- cURL library is used to communicate with other servers with the help of a wide range of protocols.
- cURL allows the user to send and receive the data through the URL syntax.
- cURL makes it easy to communicate between different websites and domains.

WEB SCRAPING USING cURL

Use of cURL : Common cases for the use of cURL, and they include:

Downloading the content of a website

Downloading a file from a website

Auto form submission

Authentication

Use of cookies

WEB SCRAPING USING cURL

How Does the cURL work?

cURL works by sending a request to a web site, and this process includes the following four parts:

- Initialization
- Setting the options
- Execution with curl_exec
- Releasing the curl_handle

WEB SCRAPING USING cURL

Marwadi University

How Does the cURL work?

1. Initialization.

```
$handle = curl_init();
```

2. Setting the options. There are many options, for example, an option that defines the URL.

```
curl_setopt($handle, CURLOPT_URL, $url);
```

3. Execution with curl_exec().

```
$data = curl_exec($handle);
```

4. Releasing the cURL handle.

```
curl_close($handle);
```

1.Initialize a curl session

2.Set various options for the session

3.Execute and fetch/send data from/to server

4.Close the session

WEB SCRAPING USING cURL

Some basic cURL functions:

- The *curl_init()* function will initialize a new session and return a cURL handle.
- *curl_exec($ch)* function should be called after initialize a cURL session and all the options for the session are set. Its purpose is simply to execute the predefined CURL session (given by ch).
- *curl_setopt($ch, option, value)* set an option for a cURL session identified by the ch parameter. Option specifies which option is to set, and value specifies the value for the given option.
- *curl_setopt($ch, CURLOPT_RETURNTRANSFER, true)* return page contents. If set to false then no output will be returned.
- *curl_setopt($ch, CURLOPT_URL, $url)* pass URL as a parameter. This is your target server website address. This is the URL you want to get from the internet.
- *curl_exec($ch)* grab URL and pass it to the variable for showing output.
- *curl_close($ch)* close curl resource, and free up system resources.

WEB SCRAPING USING cURL

How Does the cURL work?

cURL works by sending a request to a web site, and this process includes the following four parts:

- Initialization
- Setting the options
- Execution with curl_exec
- Releasing the curl_handle

WEB SCRAPING USING cURL

1. **How to download the contents of a remote website to a local file?**

```
<?php

$handle=curl_init();
$url = "https://www.imdb.com/";
curl_setopt($handle,CURLOPT_RETURNTRANSFER,true);
curl_setopt($handle, CURLOPT_URL,$url);
$output =curl_exec($handle);
echo $output;
curl_close($handle);
?>
```

WEB SCRAPING USING cURL

2.How to download a file from a remote site using cURL?

```
<?php
$url = "https://www.mprnews.org/story/2015/05/15/books-short-stories";

$file = __DIR__ . DIRECTORY_SEPARATOR . "the_divine_comedy.html";
$handle = curl_init();

$fileHandle = fopen($file, "w");
curl_setopt_array($handle,array(CURLOPT_URL=>$url,CURLOPT_FILE => $fileHandle,));
$data = curl_exec($handle);

curl_close($handle);
fclose($fileHandle);
?>
```

WEB SCRAPING USING cURL

3. How to handle cookies with cURL?

```
<?php
$handle = curl_init();
$url="https://www.marwadiuniversity.ac.in/";
$file = __DIR__ . DIRECTORY_SEPARATOR . "cookie.txt";
curl_setopt_array($handle,array(CURLOPT_URL => $url,
CURLOPT_COOKIEFILE => $file,
CURLOPT_COOKIEJAR  => $file,
CURLOPT_RETURNTRANSFER=>true)
);

$data = curl_exec($handle);
curl_close($handle);
?>
```

MAIL FUNCTION

The mail( ) function allows you to send emails directly from a script.

For the mail functions to be available, PHP requires an installed and working email system. The program to be used is defined by the configuration settings in the php.ini file.

Syntax : mail(to, subject, message, headers);

Parameter Values

- **to :** Required. Specifies the receiver of the email
- **Subject** : Required. Specifies the subject of the email.
- **Message** : Required. Defines the message to be sent. Each line should be separated with a (\n). Lines should not exceed 70 characters.
- **headers**: Optional. Specifies additional headers, like From, Cc, and Bcc. The additional headers should be separated with a CRLF (\n).

MAIL FUNCTION

How To Send Mail From Localhost XAMPP Using Gmail

1: Open the XAMPP installation directory.

2: Navigate php.ini file from C:\xampp\php

3: Press ctrl + f to find the mail function

4: Search & pass the below-mentioned values.

› SMTP=smtp.gmail.com

› smtp_port=587

› sendmail_from = milopleInc@gmail.com /*Your gmail id*/

› sendmail_path = "\"C:\xampp\sendmail\sendmail.exe\" -t"

MAIL FUNCTION

How To Send Mail From Localhost XAMPP Using Gmail

5: Open sendmail.ini file from C:\xampp\sendmail.

6: Press ctrl + f & find sendmail.

7: Search & pass the below-mentioned values.

› smtp_server=smtp.gmail.com

› smtp_port=587

› error_logfile=error.log

› debug_logfile=debug.log

› auth_username=milopleInc@gmail.com /*Your Gmail id*/

› auth_password=********** /*Your Gmail password*

MAIL FUNCTION

How To Send Mail From Localhost XAMPP Using Gmail

8: Script to send mail

```
<?php
$to_email = "receipient@gmail.com";
$subject = "Test email to send from XAMPP";
$body = "Hi, This is test mail to check how to send mail from Localhost Using Gmail ";
$headers = "From: sender email";

if (mail($to_email, $subject, $body, $headers))
{
   echo "Email successfully sent to $to_email...";
}
else
{
   echo "Email sending failed!";
}
```

WEB SERVICES

- A Web Service is any service that is available over the Internet.
- Web services are open standard (XML, SOAP, HTTP, etc.) based web applications that interact with other web applications for the purpose of exchanging data.
- It uses a standardized XML messaging system and is not tied to any one operating system or programming language.
- A Web Service is seen as an application accessible to other applications over the Web.
- A Web Service is a software system identified by an URL, whose public interfaces and bindings are defined and described using XML.
- For example, a client invokes a web service by sending an XML message, then waits for a corresponding XML response. As all communication is in XML, web services are not tied to any one operating system or programming language — Java can talk with Perl; Windows applications can talk with Unix applications.

WEB SERVICES

Components of Web Services

- The basic web services platform is XML + HTTP. All the standard web services work using the following components –

- SOAP - (Simple Object Access Protocol)
- UDDI - (Universal Description, Discovery and Integration)
- WSDL - (Web Services Description Language)

WEB SERVICES

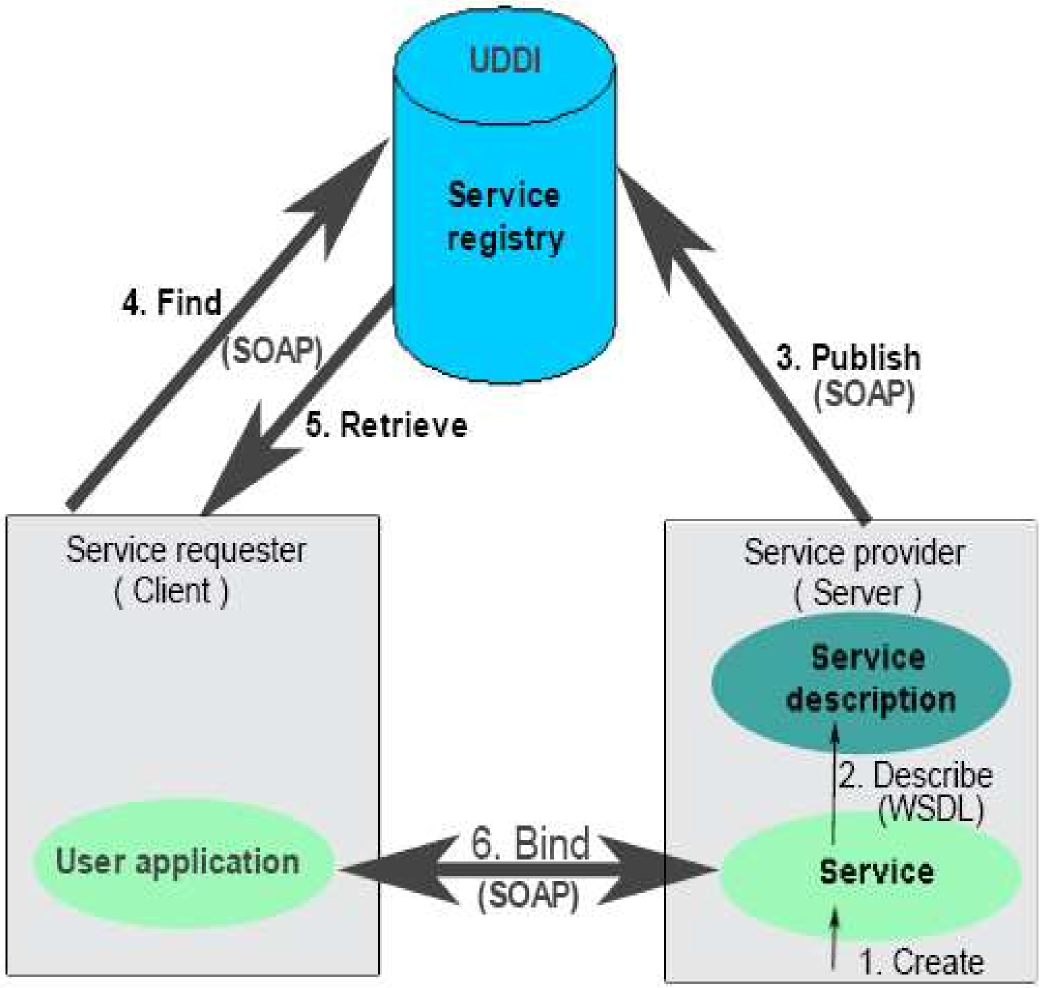
How Does a Web Service Work?

- A web service enables communication among various applications by using open standards such as HTML, XML, WSDL, and SOAP. A web service takes the help of –

- XML to tag the data
- SOAP to transfer a message
- WSDL to describe the availability of service.

WEB SERVICES ARCHITECTURE

Web Services Architecture



WEB SERVICES ARCHITECTURE

- The Architecture covers three primary entities, which are Service Provider, Service Registry, and Service Requester.
- Web Services are independent from programming languages, so any programming language can be used by the developer to create a service
- The service will then be described with WSDL for deploying.
- The service provider publishes the service in a Service Registry, which can be used for service requesters in public
- The Service Registry contains complete information about the service.
- The service requester will look for services, which it needs and retracts service information.
- Now, service requester can communicate with its corresponding service.

BASIC WEB SERVICES STANDARDS

- XML (eXtensible Markup Language)
- SOAP 1.2 (Simple Object Access Protocol)
- WSDL (Web Services Description Language)
- UDDI (Universal Description, Discovery and Integration)

BASIC WEB SERVICES STANDARDS

XML

- Developed by W3C
- XML stands for eXtensible Markup Language.
- XML was designed to store and transport data.
- XML was designed to be both human- and machine-readable.
- Its primary purpose is to facilitate the sharing of structured data across different information systems, particularly via the Internet.

XML Syntax Rules

XML documents must contain one root element that is the parent of all other elements:

```
<root>
          <child>
                      <subchild>.....</subchild>
         </child>
</root>
```

BASIC WEB SERVICES STANDARDS

XML Example

```
<bookstore>
       <book category="children">
                  <title>Harry Potter</title>
                  <author>J K. Rowling</author>
                   <year>2005</year>
                  <price>29.99</price>
      </book>
      <book category="web">
                  <title>Learning XML</title>
                  <author>Erik T. Ray</author>
                   <year>2003</year>
                   <price>39.95</price>
                  </book>
</bookstore>
```

BASIC WEB SERVICES STANDARDS

SOAP

- SOAP stands for Simple Object Access Protocol
- SOAP is an application communication protocol.
- SOAP provides away to communicate between applications running on different operating systems, with different technologies and programming languages.
- SOAP is a format for sending and receiving messages
- SOAP is platform independent
- SOAP is based on XML
- SOAP is a W3C recommendation

BASIC WEB SERVICES STANDARDS

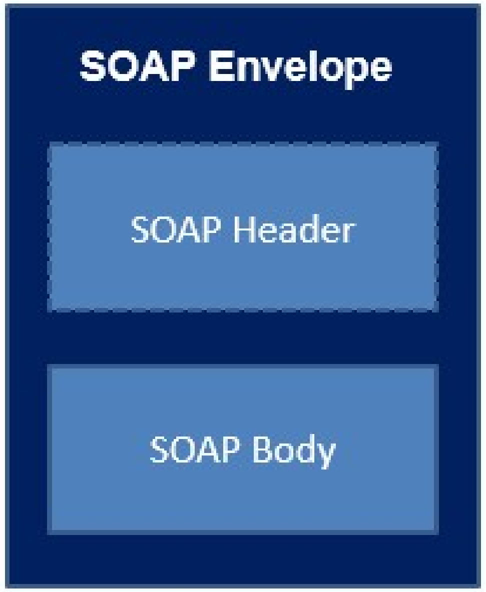
SOAP

```
<?xml version="1.0"?>
<soap:Envelope  xmlns:soap=http://www.w3.org/2001/12/soap-envelope
            soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

            <soap:Header>
                        … …
            </soap:Header>

            <soap:Body>
                        … …
            </soap:Body>

</soap:Envelope>
```



BASIC WEB SERVICES STANDARDS

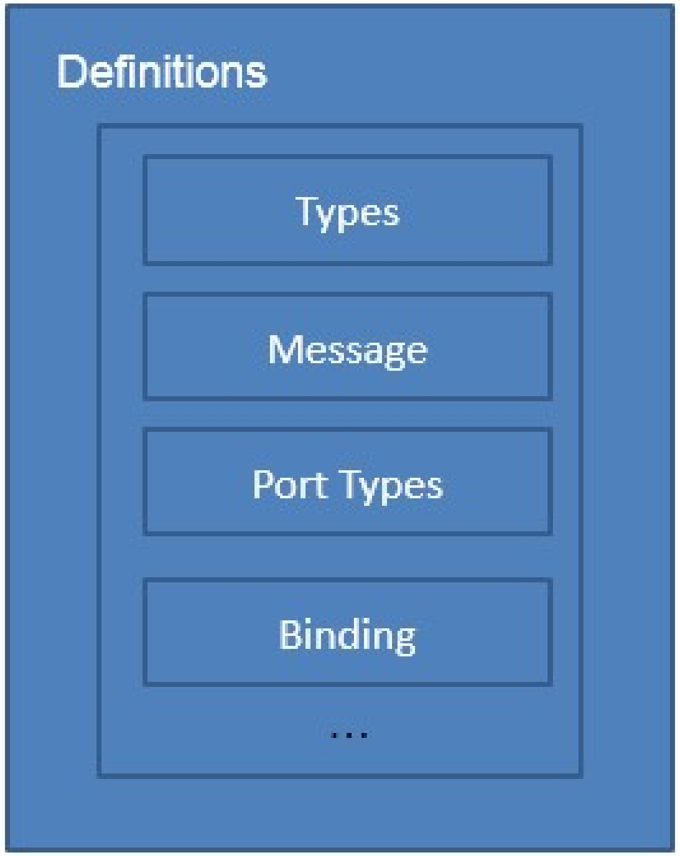
WSDL

- WSDL stands for Web Services Description Language
- It is an XML-based language for describing Web Services and how to access them.
- WSDL is a W3C recommendation from 26. June 2007
- It specifies the location of the service and the operations the services exposes.
- A WSDL document is simply a set of definitions.

BASIC WEB SERVICES STANDARDS

WSDL Structure

```
<definitions>
<types>
  data type definitions........
</types>
<message>
  definition of the data being communicated....
</message>
<portType>
  set of operations......
</portType>
<binding>
  protocol and data format specification....
</binding>
</definitions>
```



BASIC WEB SERVICES STANDARDS

UDDI

- WSDL stands for Web Services Description Language

- The Universal Description, Discovery, and Integration define a way to publish and discover information about Web services.
- The UDDI business registration is an XML file that describes a business entity and its Web services.
- Only authorized individuals can publish or change information in the registry
- Changes or deletions can only be made by the originator of the information
- Each instance of a registry can define its own user authentication mechanism

Marwadi University

Thank You!