Web Database Application

CIT414 (2 Units)

Dr. B.L. Muhammad-Bello

Objectives

At the end of this unit, you should be able to:

- Describe the PHP date() and time() functions.
- Explain how conditional functions work.
- Use default parameters in functions

The PHP Date() Function

- The PHP date() function formats a timestamp to a more readable date and time.
- **Syntax:** date(format,timestamp)
  - Format: Required. Specifies how to format the date/time. It uses letters to represent date and time formats
  - Timestamp: Optional. Specifies a timestamp. The default is the current date and time (as a timestamp)
- The following characters are recognized in the *format* parameter string:
  - d - The day of the month (01-31)
  - j - The day of the month without leading zero (1 – 31)
  - D - The textual representation of a day, three letters (Mon through Sun)
  - l – The full textual representation of a day (Monday though Sunday)
  - S - English ordinal suffix for the day of the month (st, nd, rd, and th. Works with j)
  - m - The current month, as a number (01-12)
  - n – numeric representation of a month without leading zero (1 – 12)
  - F – A full textual representation of a month (January through December)
  - Y - The current year in four digits

What is a Timestamp

- A timestamp is the number of seconds since January 1, 1970 at 00:00:00 UTC/GMT. This is also known as the Unix Timestamp.
  - For example, the date and time " February 14, 2007 16:48:12 " in the GMT time zone is represented by the UNIX timestamp **1171471692** , because February 14, 2007 16:48:12 is exactly 1,171,471,692 seconds after midnight on January 1, 1970.
- A timestamp is simply an integer. Thus
  - it's easy for a computer to store it
  - it's easy to manipulate dates and times
- For example, to add one day to a timestamp, you just add the number of seconds in a day (which happens to be **86,400** seconds) to the value.

PHP Date - Adding a Timestamp

- The second parameter in the date() function specifies a timestamp. This parameter is optional and if unspecified, the current time is used.
  - `echo time();  // Displays current Date and Time as an Integer`
- The mktime() function is used to create a timestamp, it returns the Unix timestamp for a specified date.
- Syntax:
  `mktime(hour,minute,second,month,day,year,is_dst)`
- Example: Using the mktime() function to create a timestamp for tomorrow.

-
  ```
  <?php $tomorrow =
  mktime(0,0,0,date("m"),date(“d")+1,date("Y"));
      echo "Tomorrow is ".date("Y/m/d", $tomorrow );
  ```

mktime() and time() function

- **mktime()** is useful for doing date arithmetic and validation, as it will automatically calculate the correct value for out-of-range input.
- mktime() assumes that the arguments you pass are in your computer' s time zone — it converts the supplied time to UTC so that it can be returned as a timestamp.
- For example, each of the following lines produces the string "Jan-01-1998".

  -
    ```
    <?php
    echo date("M-d-Y", mktime(0, 0, 0, 12, 32, 1997));
    echo date("M-d-Y", mktime(0, 0, 0, 13, 1, 1997));
    echo date("M-d-Y", mktime(0, 0, 0, 1, 1, 1998));
    echo date("M-d-Y", mktime(0, 0, 0, 1, 1, 98));
    ?>
    ```

1. What would this echo statement output?

   -
     ```
     echo "July 1, 2000 is on a " . date("l", mktim
     e(0, 0, 0, 7, 1, 2000));
     ```

time() function Example

- The **time()** function returns the current time measured in the number of seconds since the Unix Timestamp (January 1 1970 00:00:00 GMT).
- Example:

  -
    ```
    <?php
    $nextWeek =
    time()+(7*24*60*60); // 7 days; 24 hours; 60 min
    s; 60secs
    ```

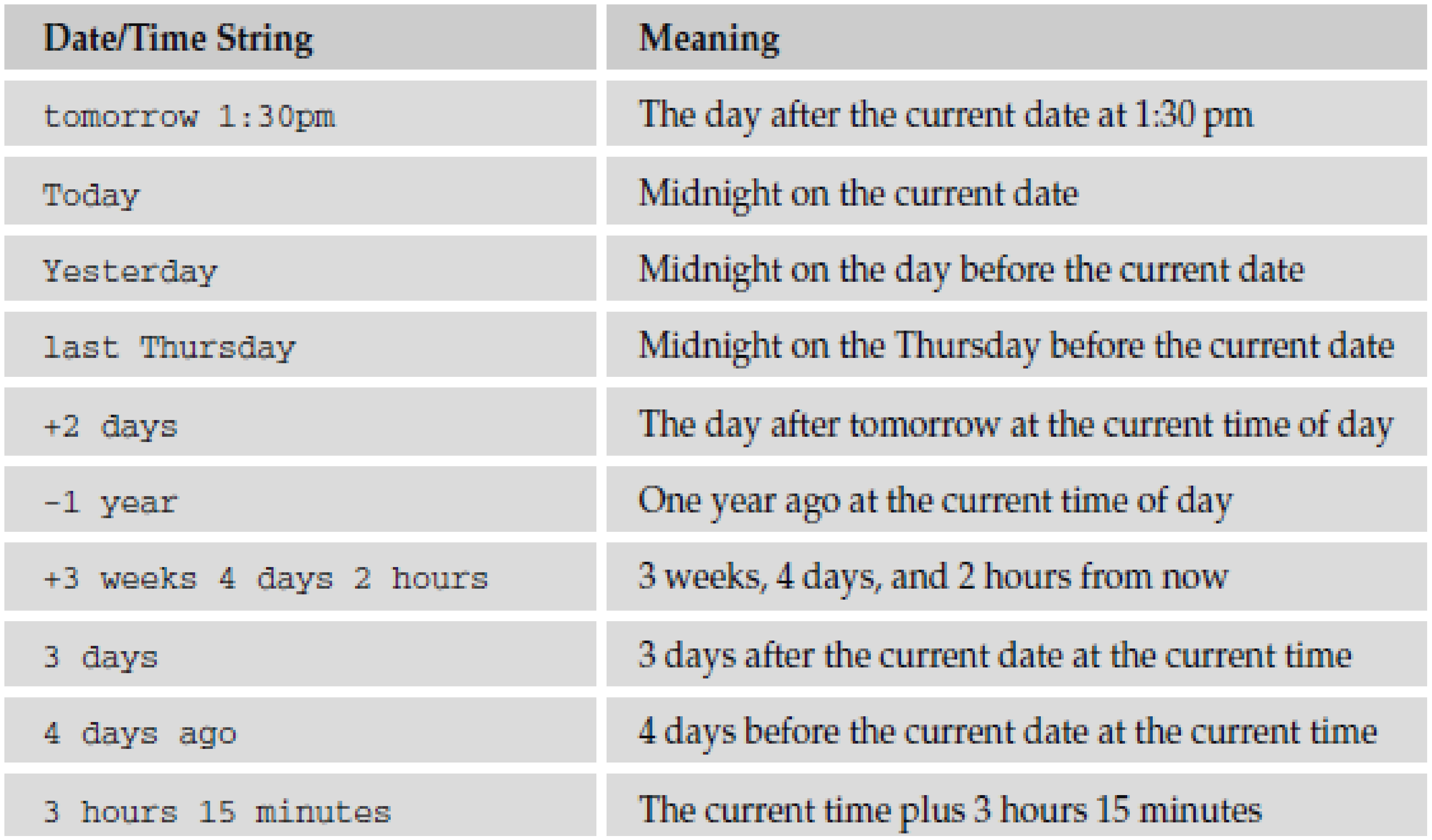
```
echo "Today is  ". date("d-m-Y", time()); echo
"</br>";
echo "Next week is ". date("d-m-Y", $nextWeek);
echo "</br>";     ?>
```

strtotime() function

- strtotime() expects a string representing a date, and attempts to convert the string into a timestamp:
  - `$timestamp = strtotime("15th September 2006 3:12pm");`
- strtotime() calculates relative dates (such as " tomorrow 1:30pm") based on the current date. If you want to calculate a relative date based on a different date, pass that date as a second argument to strtotime() , in timestamp format:
  - `$localTime = strtotime( "tomorrow 1:30pm", 0 ); // January 2nd 1970, 1:30:00 pm`
- You can pass in dates and times in practically any human - readable format you like.

Date/Time String	Meaning
6/18/99 3:12:28pm	3:12:28 pm on June 18th , 1999
15th Feb 04 9:30am	9:30 am on February 15th , 2004
February 15th 2004, 9:30am	9:30 am on February 15th , 2004

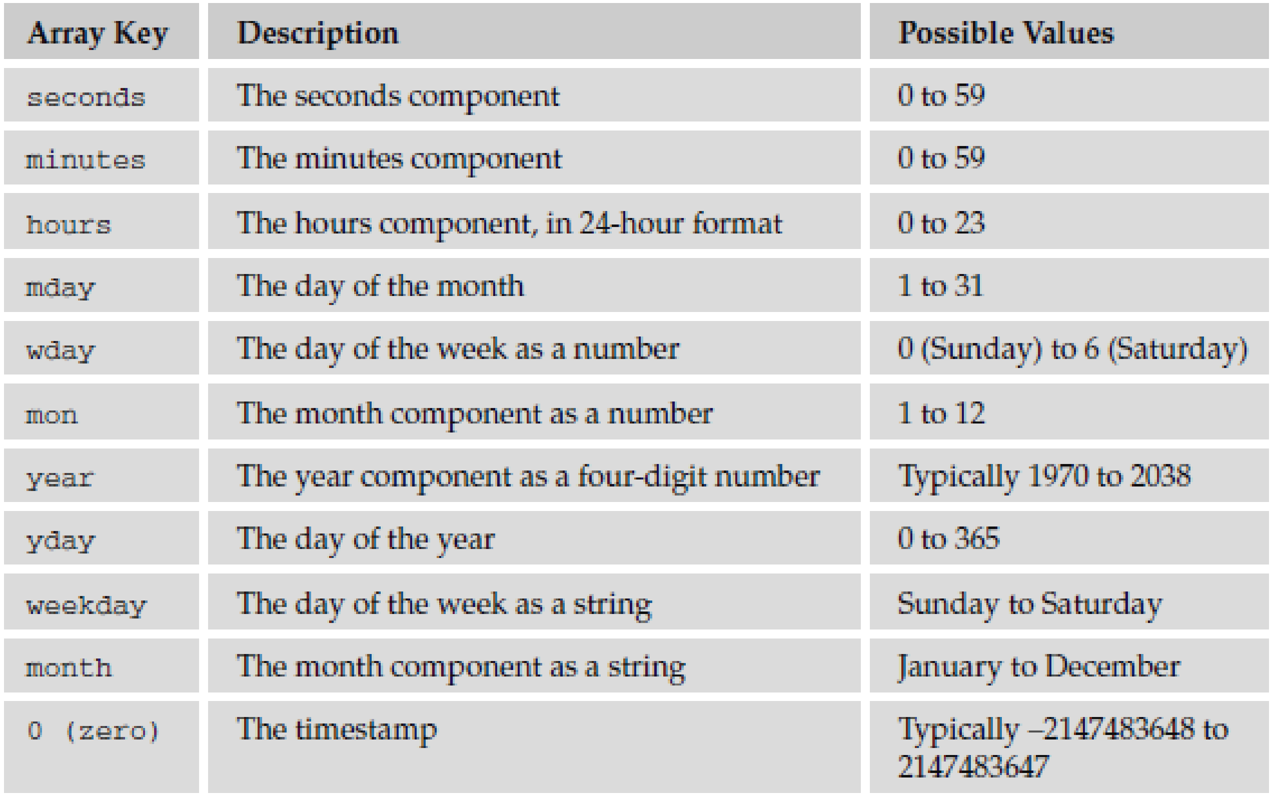
Other Examples



Date/Time String	Meaning
`tomorrow 1:30pm`	The day after the current date at 1:30 pm
`Today`	Midnight on the current date
`Yesterday`	Midnight on the day before the current date
`last Thursday`	Midnight on the Thursday before the current date
`+2 days`	The day after tomorrow at the current time of day
`-1 year`	One year ago at the current time of day
`+3 weeks 4 days 2 hours`	3 weeks, 4 days, and 2 hours from now
`3 days`	3 days after the current date at the current time
`4 days ago`	4 days before the current date at the current time
`3 hours 15 minutes`	The current time plus 3 hours 15 minutes

getdate() function

- getdate() accepts a timestamp and returns an associative array of date/time values corresponding to the supplied timestamp. The array contains the following keys:



Array Key	Description	Possible Values
`seconds`	The seconds component	0 to 59
`minutes`	The minutes component	0 to 59
`hours`	The hours component, in 24-hour format	0 to 23
`mday`	The day of the month	1 to 31
`wday`	The day of the week as a number	0 (Sunday) to 6 (Saturday)
`mon`	The month component as a number	1 to 12
`year`	The year component as a four-digit number	Typically 1970 to 2038
`yday`	The day of the year	0 to 365
`weekday`	The day of the week as a string	Sunday to Saturday
`month`	The month component as a string	January to December
`0 (zero)`	The timestamp	Typically –2147483648 to 2147483647

FUNCTIONS

- Generally speaking, a *function* — also called a *subroutine* in some other languages — is a self-contained block of code that performs a specific task.
- You define a function using a special syntax — and you can then call that function from elsewhere in your script.
- A function often accepts one or more *arguments* , which are values passed to the function by the code that calls it. The function can then read and work on those arguments.
- A function may also optionally return a value that can then be read by the calling code. In this way, the calling code can communicate with the function.

User-defined functions

- Functions can be defined using syntax such as the following:

-
  ```
  <?php
  function myfunc($para_1, $para_2, /* ...
  , */ $para_n)
  {
      echo "Example of a function.\n";
      return $returnVal;
  }
  ?>
  ```
- To tell PHP that you want your function to accept arguments, you specify one or more corresponding *parameters* when you define your function.
- A parameter is a variable that holds the value passed to it when the function is called.

PHP Functions - Example

- Example:

```
<?php
function writeMyName($fname)
  {
  echo $fname . " Obama.<br />";
  }
echo "My name is ";
writeMyName("Barack");
echo "My wife is ";
writeMyName(“Michelle");
echo "My daughter's name is ";
writeMyName(“Malia");
writeMyName("");
?>
```

2. Write a function that takes two parameters?

Use of default parameters in functions

- Example:

-
  ```
  <?php
  function makecoffee($type = "cappuccino")
  {
      return "Making a cup of $type.\n";
  }
  echo makecoffee();
  echo makecoffee(null);
  echo makecoffee("espresso");
  ?>
  ```
- What will the above example output?
  - Making a cup of cappuccino.
  - Making a cup of .
  - Making a cup of espresso.

Returning values

- Values are returned by using the optional return statement. Any type may be returned, including arrays and objects.
- This causes the function to end its execution immediately and pass control back to the line from which it was called.
- Example:

-
  ```
  <?php
  function square($num)
  {
      return $num * $num;
  }
  echo square(4);   // outputs '16'.
  ?>
  ```
- When the PHP engine encounters the return statement, it immediately exits the function and returns *value* back to the code that called the function, which can then use the value as required.

Conditional functions

- Functions need not be defined before they are referenced, *except* when a function is conditionally defined.
- When a function is defined in a conditional manner, its definition must be processed *prior* to being called.

Conditional functions

-
  ```
  <?php
  $makefoo = true;

  /* We can't call foo() from here since it doesn't exist yet, but we
  can call bar() */

  bar();
  if ($makefoo) {
    function foo()
    {
      echo "I don't exist until program execution reaches me.\n";
    }
  }

  // Now we can safely call foo() since $makefoo evaluated to true

  if ($makefoo) foo();

  function bar()
  {
    echo "I exist immediately upon program start.\n";
  }

  ?>
  ```

Functions within functions

- All functions and classes in PHP have the global scope - they can be called outside a function even if they were defined inside and vice versa.
- Example:

  -
    ```
    <?php
    function foo()
    {
      function bar()
      {
        echo "I don't exist until foo() is called.\n";
      }
    }

    /* We can't call bar() yet
       since it doesn't exist. */

    foo();

    /* Now we can call bar(),
       foo()'s processing has
       made it accessible. */

    bar();

    ?>
    ```

Class Exercise

1. Write a php file which would output:

    Today is 8th April 2019
    Tomorrow is 9th April 2019
    Next tomorrow is 04/10/2019
    The current date and time is : 08-Apr-2019 09:29:41 AM
    By this time last week it was: 15-Apr-2014 07:33:47 AM.
    In two weeks time, it would be: 22-Apr-2014 07:33:47 AM

You must use the date(), mkdate() and time() functions in your code.

2. Write a php function that will compute the sum of two numbers given as an input to it. Within your document, call your function with specified input numbers and output the result to the document.