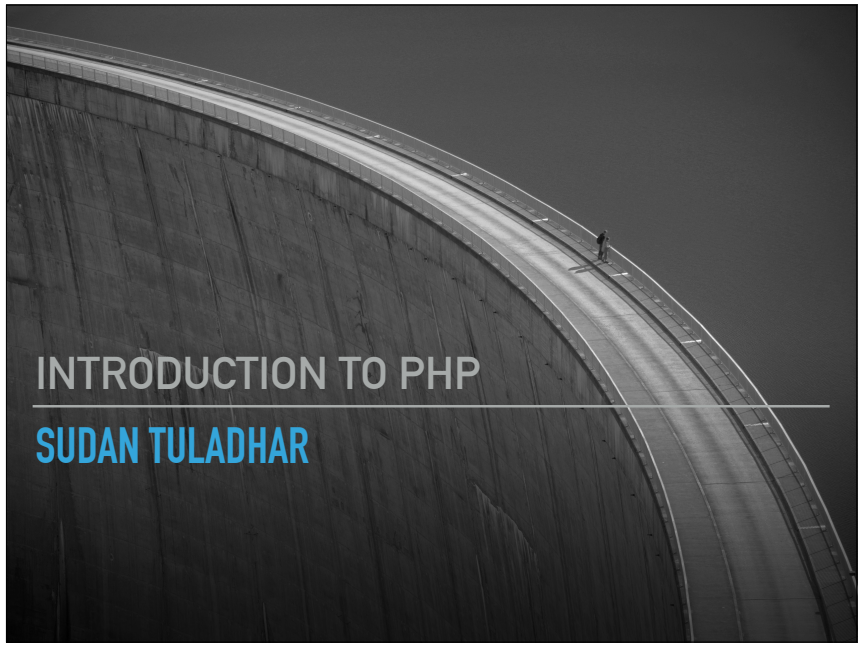

INTRODUCTION

- <?php phpinfo(); ?>
- case sensitive
- simple statements are terminated using semi-colons (;)
- white-space & line-breaks are ignored
- single line comments begin with // and #
- multi-line comments appear between /* and */

KEYWORDS

- reserved words that is used by language for core functionality
- cannot use keywords to name variables, constants, function, etc
- are case sensitive

VARIABLES

- begins with dollar ($) sign & are case-sensitive
  - $number = 100;
- should always begin with alphabet or _ sign
  - $key, $_POST, ~~$11street~~
- cannot contain spaces
  - ~~$first name~~

SUDAN TULADHAR

CONSTANTS

- identifiers for simple values
- define('HOSTNAME', 'localhost');
- once set, value of a constant cannot change

SUDAN TULADHAR

DATA TYPES

- integers - 100, 1979
- floating point numbers - 3.14, -7.1
- booleans - true | false
- strings
  - 'string value' | "string value"
- a variable can hold any data type

SUDAN TULADHAR

PRINTING TO HTML

- echo 'Hello World!!!';
- $var = 'Students!!!';
  echo 'Hello ' . $var;
  echo "Hello $var";
- dot operator (.) concatenates two string variables
- print()
- print_r(), var_dump()
- <?= $var; ?>

SUDAN TULADHAR

VARIOUS OPERATORS

- (arithmetic)
  - +, -, *, /, %
- (increment & decrement)
  - ++, --
- (comparison)
  - ==, ===, !=, <>, !==, >, >=, <, <=
- (logical)
  - &&, ||, !
- (assignment)
  - =, +=, -=, .=, *=, /=, %=

IF-STATEMENTS

- *if (expression) statement*
  - statement is executed if expression is evaluated to true
- specify alternative statement with else
  - *if(expr) stmt1*
    else stmt2
- multiple statements can be chained using *if-elseif-else*

FOR-LOOPS

- *for(start; condition; increment)*
  statements;
- *for($i=0; $i<10; $i++)*
  echo $i;

WHILE & DO-WHILE LOOPS

- *while(expression) statement*
  - statement is executed when expression evaluates to true
- *do*
  statement
  while(expression)

ARRAYS

- indexed or associative
- $arr = array();
  - $days = array('Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat');
  - $days = array(1 => 'Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat');
  - $dates = range(1,31) // array(1, 2, 3, ..., 31)
- $person = array('fname' => 'Ram', 'lname' => 'Bahadur');
- size of array: count($days)

ARRAYS

- array_keys(array), array_values(array)
- array_key_exists(key, array)
- extract(), compact()
  - $name = 'Ram'; $age = 10;
  - $arr = compact("name", "age")
    - array("name" => 'Ram', "age" => 10)
  - extract($arr)
    - $name = 'Ram'; $age = 10;
- in_array(find, array)
- array_merge($arr1, $arr2 [, $arr3])

ARRAYS

- traversing arrays with foreach loop
  - foreach($days as $day)
    echo $day;
  - foreach($person as $key=>$val)
    echo "{$key}: {$val}";

FUNCTIONS

- $val = function_name([parameter,...]);
  - $len = strlen("PHP");
- function function_name([parameter,...]) {
  statement list;
  }
  - function sum_of_numbers($num1, $num2) {
    return $num1 + $num2;
    }
- default parameters can be specified
  - function repeat($times = 5) { ... }

STRINGS

- trim(str), ltrim(str), rtrim(str)
- strtoupper(str), strtolower(str)
- rawurlencode(str), rawurldecode(str), urlencode(str), urldecode(str)
- strlen(str)
- substr(string, start [, length]), str_replace(old,new,string)
- explode(separator, string), implode(separator, array)
- strpos(haystack, needle), stripos() [compare with ===]
- parse_url(url) [returns array of URL components]
- isset(string), empty(string)

INCLUDES & REQUIRES

- <?php include “file.php”; ?>
- <?php include_once “file.php”; ?>
- <?php require “file.php”; ?>
- require is more stricter than include statements
- include - headers and footers
- require - loading libraries and functions