Name :- Jat Kishanbhai Laxmanbhai

Roll No. :- 30

Subject Name :- Building Application Using PHP

Subject Code :- BCA-402

ક્રમાંક

Q-1 What is an array? Explain types of array and array function in detail.

→ • What is an array?

In PHP, an array is an ordered collection of elements, where each element has an associated index or key index. You can store various data types ~~of~~ within an array, including string, number, booleans, objects, and even other arrays (multidimensional arrays). Arrays offer a flexible and efficient way to manage multiple related data values under a single variable name.

• Types of arrays:

○ Indexed arrays:

- Use numerical indexes starting from 0 (zero-based) to access elements.
- Example: $fruits = array("Apple", "banana", "orange");
- Accessing elements: $firstfruit = $fruits[0]; // "Apple"

○ Associative arrays:

- Use named keys (strings) to access elements.
- Example: $person = array("name" => "John", "age" => 30, "city" => "New York");
- Accessing elements: $name = $person["name"]; // "John"

○ Mutidimensional arrays:

- Arrays containing other arrays, providing a

nested structure:

- Example: $employees = array(array("name" => "Alice", "department" => "Marketing"), array ("name" => "Bob", "department" => "Sales"));

- Accessing nested elements: $aliceDept = $employees[0] ["department"]; // "marketing"

- Array functions:

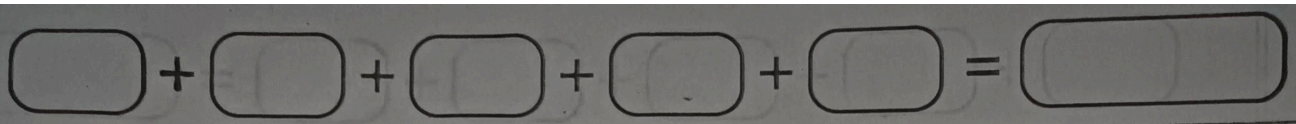
PHP offers a rich set of built-in array functions to ~~main~~ manipulate and work with arrays effectively.

Here are some commonly used functions:

- Creating arrays:
  - array(): Explicitly create an array.
  - []: Shortcut syntax for array creation.

- Accessing elements:
  - []: Indexed access using numerical indexes.
  - ["key"]: Associative access using named keys.

- Adding elements:
  - $array[] = $value: Append to the end.
  - $array[$key] = $value: Add or update using a specific key.
- Removing elements:
  - unset($array[$key]): Remove by key.
  - array_pop($array): Remove by & last element
  - array_shift($array): Remove first element.



- Iterating over arrays:
  - foreach loop: Flexible iteration with access to keys and values.
  - for loop: use with numerical indexes.
  - while loop: conditional iteration

- Other important functions:
  - count($array): Get the number of elements.
  - in_array($value, $array): check if a values exists.
  - array_merge($array1, $array2: Merge arrays
  - array_slice($array, $start, $length: Extract a portion of an array.
  - sort($array) or rsort($array): Sort arrays in ascending or descending order.

પ્રશ્ન પેટા પ્રશ્ન ક્રમાંક

□ + □ + □ + □ + □ = □

Q-2 What is exception? Explain type of errors in detail.

→ - An exception is an event that disrupts the normal flow of a program's execution

- it's a way of signaling that an error or unexpected condition has occurred.

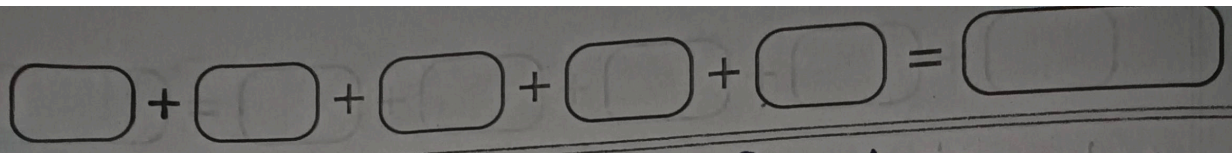
- Exceptions help prevent program crashes and allow for graceful error handling.

• Types of errors in PHP:

1. Parse errors:
   - Syntax errors in the code itself.
   - Prevent the code from executing at all.
   - Example: missing semicolons, mismatched parentheses, incorrect variable names.

2. Fatal errors:
   - Severe errors that halt the program execution abruptly.
   - Often indicate problems that cannot be recovered from
   - Example: accessing undefined variable, calling non-existed function, memory exhaustion

3. Warnings:
   - Indicate potential problems or deprecated features.
   - Don't stop execution, but can lead to unexpected behavior
   - Example: using undefined variables in loose



comparison, using deprecated functions.

4. Notices:
- Minor issuses that usually don't affect program execution.
- can be helpful for debugging or identifying potential problems.
- Examples: accessing array elements with non-integer keys, using unassigned variables in string contexts.

5. Exceptions:
- Objects that represent errors or exceptional conditions.
- can be thrown and caught using try... catch blocks for controlled error heunaling
- Examples: dividing by zero, accessing a file that doesn't exist, attemepting to connect to a ~~bata~~ database that's down.

પ્રશ્ન પેટા પ્રશ્ન ક્રમાંક

Q-3 Explain all MySQL functions in PHP in detail.

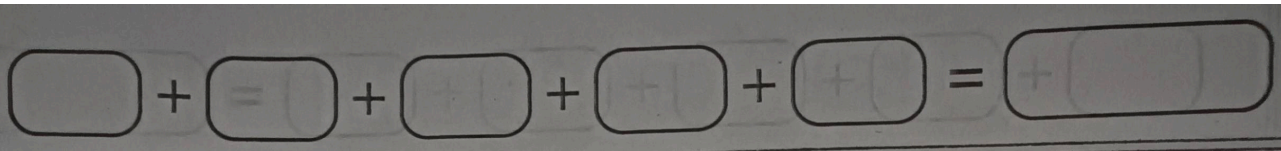
→ MySQL is officially deprecated since PHP 5.5, meaning it's no longer actively developed and might be removed in future PHP versions. MySQLi is the actively maintained and recommended extension. [They have superficial similarity]

Here are the core PHP functions for interacting with MySQL database, focusing on the recommended mysqli_* extension.

- **Connecting:**
- mysqli_connect(): Establishes a connection to a MySQL server.
- mysqli_real_connect(): Enhanced connection function with additional options.
- mysqli_select_db(): Selects a database to work with.

- **Querying:**
- mysqli_query(): Executes an SQL query.
- mysqli_prepare(): Prepares an SQL statement for execution, preventing SQL injection.
- mysqli_stmt_execute(): Executes a prepared statement.
- mysqli_stmt_bind_param(): Binds variable to prepared statement parameters.
- mysqli_stmt_fetch(): Fetches results from a prepared statement.

- **Fetching Results:**
- mysqli_fetch_assoc(): Fetches a row as a associative



array

- mysqli_fetch_array(): Fetches a row as an array with both associative and numeric indices.
- mysqli_fetch_row(): Fetches a row as a numeric array
- mysqli_fetch_object(): Fetches a row as an object.

• Result Information:

- mysqli_num_rows(): Returns the number of rows in a result set.
- mysqli_affected_rows(): Returns the number of rows affected by an INSERT, UPDATE, or DELETE query

• Error Handling:

- mysqli_error(): Returns the error message from the last MySQL operation.

• Closing Connections:

- mysqli_close(): Closes a MySQL connection.

પ્રશ્ન પેટા પ્રશ્ન ક્રમાંક

Q-4 Explain all components of joomla in detail.

→ Joomla! components : Building Blocks for Powerful web Applications

Joomla! is a renowned content management system (CMS) powered by PHP that empowers users to create and manage dynamic websites without extensive coding knowledge. Its modular structure relies on various components each serving a specific purpose and working together seamlessly to provide rich functionality.

Core Components :

- Com_content : The cornerstone of Joomla! enabling you to create, edit, publish, and organize diverse content types (articles, blog posts, news item.). It offers robust access control, versioning, and categorization capabilities.
- Com_Categories : Establishes a hierarchical structure for organizing content within com_content and other components that support categories, you can create nested categories to facilitate browsing and organization.

- Com_users : manages user accounts, enabling you to register, edit, and assign user to roles with varying permissions. This ensures granular control over website access and actions.

□ + □ + □ + □ + □ = □

- Com_menus: Constructs hierarchical menus to navigate your website's content. You can creates static or dynamic menus, positioning them across various locations in your templates.

- Com_templates: Contains themes that dictate the overall visual appearance and layout of your website. You can select from pre-built themes or create custom ones to align with your brand identity.

- Com_plugins: Extends Joomla's functionality through lightweight programs that inject specific behaviors or actions without alteri core files. Plugins enable diverse tasks like spam filtering, social media installing, updating and ~~uninstalling~~ seo and user registration.

- Com_installer: Facilitates installing, updating and uninstalling Joomla! extensions (components, modules, plugins) securely and conveniently.

By mastering these components and their interplay, you can leverage Joomla! to create robust, interactive, and feature-rich website that meet your divers needs.