PHP (Personal Home Page → Hypertext Preprocessor)

- **Goal**: create dynamic and interactive web pages (creating server side web development)
- **Father of PHP**: Rasmus Lerdorf (in 1994)
- **Current Version:** PHP 7 (released in 2015 → PHP 8.0 released in 2020)
- **Characteristics of PHP:**
  - It is *widely – used, open source scripting language* (server side)
  - PHP scripts *runs on the server*
  - It is interpreted language
  - It is faster than other server side scripting language (JSP, ASP)
  - The results *returns to the client in HTML format*
  - The *PHP file have the extension of .php*
  - It is sued to create, open, read, write, delete and close the files on the server
  - *PHP runs in various platforms*
  - It is *compatible to run in all the servers*
  - It supports all the main protocols (HTTP, IMAP, FTP)
  - It is easily embedded inside of HTML files
  - It is dynamically typed language (loosely typed language)
  - It is easy to install and set-up
- **Suitable Scenario:**
  1. It is used to *create, open, read, write, delete and close the files on the server*
  2. It is used to process form data
  3. It have predefined functions to encrypt the data
  4. It is used to *interact with the database*
  5. It is used to control the user-access
  6. PHP can send and receive cookies and sessions
- **Applications**:
  - Currently social networks run in PHP servers (Facebook and yahoo)
  - It is a powerful tool for managing content management system (WordPress)
- **Content of PHP file:**
  1. HTML

2. CSS
3. Jscript
4. PHP script
5. Ajax
6. XML ad JSON
7. jQuery

❖ **Disadvantages of PHP:**

1. Inconsistency (function names and parameter orders)
2. Security Concerns (vulnerability)
3. Performance (not good like compiled languages)
4. Lack of modern features
5. Scalability challenges
6. Not suitable for large scale applications
7. Limited OOP support

❖ **PHP Datatypes:** **var_dump($variable_name)**

1. int
2. String
3. Float
4. Boolean
5. Array
6. Object
7. Resource (reference) → Storing reference to functions

   Ex: database call
8. Null

❖ **Scope of Variables:**

1. Local
2. Global → Accessing global variable → $GLOBALS[‘a’]
3. Static

❖ **Syntax for including PHP code:**

```
<?php
        ……. PHP actions ----------
?>
```

- **Processing Form Data:**
  1. $_POST[“name”]
  2. $_GET[“name”]
  3. $_SERVER[“REQUEST_METHOD”]==”POST”
- **htmlspecialchars($form_data)** → convert into html predefined character into entities.
- **Form data processing:**

```
function process($data)
{
        $data=trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
}
```

- Retain; value=”<?php echo $n;?>”
- empty($data)→ empty data
- **$_SERVER** → it is a ***global variable used to store the filename of currently executing script***.
- **Retain in the same page**: **$_SERVER[“PHP_SELF”]** → submitted the data to the same page
- Removing Cross site scripting (slash in the input) → **htmlspecialchars($_SERVER[“PHP_SELF”])**
- **Database Connection**

  Methods:

  1. affected_rows;
  2. close()
  3. mysqli_num_rows($result) → number of rows in $result
  4. fetch_assoc() / mysqli_fetch_assoc() → fetches a result row as associative array. (accessing by using column name)

     Example:

```
$row = $result->fetch_assoc()
```

5. stat() / mysqli_stat() function returns the current system status.
6. commit() /mysqli_commit() [$conn→ commit()] → it is used to commit the current transcation
7. autocommit(false) [$conn-> autocommit(false)]
8. rollback() / mysqli_rollback()
9. close()
10. connect() / mysqli_connect

Example:

$conn = new connect(“localhost”,”root”,”password”,”DB”)

$conn = mysqli_connect(“localhost”,”root”,”password”,”DB”)

11. connect_errno / mysqli_connect_errno() → it returns the error code from the last connection error, if any.
12. connect_error / mysqli_connect_error() it returns the error description from the last connection error, if any
13. filed_count / mysqli_field_count()
14. query() / mysqli_query()

Sample program for select Query:

```
<html>
<body>
<?php
   $conn = new mysqli("localhost","root","","karthi");
   if ($conn->connect_error)
   {
      die("Connection failure: " . $conn->connect_error);
   }
   else
   {
         echo "Connection success";
         $sql = "SELECT * FROM arun";
         if ($result = $conn->query($sql))
         {
               echo "Returned rows are: " . $result -> num_rows;
```

```
                while($row = $result->fetch_assoc())
                {
                        echo "id: " . $row["name"]. " - Name: "          .
$row["password"]. "<br>";
                }
        }
        else
        {
                echo "No records has been found";
        }
    }
?>
</body>
</html>
```

❖ **Example program for insert query:**

```
<html>
<body>
    <?php
        $conn = new mysqli("localhost","root","","karthi");
        if($conn->connect_error)
                die("Unable to connect".$conn->connect_errno);
        else
        {
                $q = "insert into arun values(\"puvanesh\", 342234)";
                $r = $conn->query($q);
                echo ("Number of rows ".$r);
        }
    ?>
<body>
</html>
```

❖ s