**1. Describe the Autoglobals or Superglobals**

**Autoglobals** (also known as **superglobals**) are predefined variables in PHP that are always accessible, regardless of the scope—whether within a function, class, or script. These variables hold essential information related to the server, session, environment, user inputs, and more. You don't need to declare them or pass them explicitly. They are accessible anywhere in your PHP script.

The most commonly used superglobals include:

* $\_GET – Collects data sent via the HTTP GET method.
* $\_POST – Collects data sent via the HTTP POST method.
* $\_REQUEST – Contains combined data from $\_GET, $\_POST, and $\_COOKIE.
* $\_SESSION – Holds session variables that are available across multiple pages.
* $\_COOKIE – Holds cookie data sent from the client’s browser.
* $\_FILES – Contains information about uploaded files.
* $\_SERVER – Contains information about headers, paths, and script locations.
* $\_ENV – Contains environment variables.
* $\_GLOBALS – References all variables in global scope.

**2. Describe the Limitations of Using the GET Method to Submit Form Data**

The **GET method** sends form data as URL parameters. While it's useful in certain scenarios, it has limitations:

* **Data size limitation**: GET imposes a limit on the amount of data that can be sent (around 2,048 characters, though this may vary by browser).
* **URL visibility**: Data is appended to the URL, making it visible in the browser's address bar, which is a security risk, especially for sensitive information like passwords.
* **Limited security**: As the data is visible in the URL, it is not suitable for sending sensitive data (e.g., passwords or credit card numbers).
* **Caching**: Data sent via GET is stored in the browser's cache and server logs, and is also bookmarkable, which can cause issues for non-idempotent actions (e.g., form submissions that change server-side data).

Despite these limitations, GET is often used for **retrieving** (not modifying) data, such as search queries, since it allows users to bookmark and share URLs.

**3. Which Element of the $\_SERVER Auto-Global is Used to Refer to the Current Script?**

The $\_SERVER['PHP\_SELF'] element refers to the current script being executed. It returns the filename of the currently executing script relative to the root directory.

**Example:**

echo $\_SERVER['PHP\_SELF'];

If the script is located at http://example.com/form.php, this would output:

/form.php

**4. Which Auto-Global Is Used for Uploaded Files and What Are Their Element Names?**

The **$\_FILES** superglobal is used to handle file uploads in PHP. It contains information about the uploaded files, such as file name, type, size, temporary storage location, and potential errors.

Key elements of $\_FILES include:

* $\_FILES['file']['name'] – The original name of the uploaded file.
* $\_FILES['file']['type'] – The MIME type of the uploaded file (e.g., image/jpeg).
* $\_FILES['file']['size'] – The size of the uploaded file in bytes.
* $\_FILES['file']['tmp\_name'] – The temporary name of the file stored on the server.
* $\_FILES['file']['error'] – Any error associated with the file upload.

**Example:**

Explain

if ($\_FILES['uploaded\_file']['error'] == UPLOAD\_ERR\_OK) {

$tmpName = $\_FILES['uploaded\_file']['tmp\_name'];

$fileName = $\_FILES['uploaded\_file']['name'];

// Process file upload

}

**5. What Is the Difference Between GET and POST Methods When a Form is Submitted?**

**GET Method**:

* **Data transmission**: Appends form data to the URL as query parameters (e.g., http://example.com/form.php?name=John&age=25).
* **Data visibility**: Data is visible in the browser's address bar, making it less secure.
* **Data length**: Limited in size, typically up to 2,048 characters (may vary depending on the browser).
* **Caching**: GET requests are stored in browser history and can be bookmarked or cached.
* **Use case**: Suitable for **retrieving** data or performing **idempotent** actions, such as searching or browsing.

**POST Method**:

* **Data transmission**: Sends form data in the body of the HTTP request, making it invisible in the URL.
* **Data visibility**: Data is not visible in the URL, making it more secure for transmitting sensitive data.
* **Data length**: No significant size limit for POST data (depends on server configuration).
* **Caching**: POST requests are not cached by default, nor can they be bookmarked.
* **Use case**: Suitable for **modifying** or **submitting** data (e.g., form submissions, uploading files, user registration).

In summary, **GET** is mainly used for retrieving data, and **POST** is used for submitting data securely or when a large amount of data needs to be transferred.