المتغيرات العامة الفائقة(Superglobals) :

تُعتبر بمثابة العمود الفقري للبرمجة التفاعلية في لغةPHP ، وهي مصفوفات مدمجة ومُعرفة مسبقاً داخل محرك اللغة تتوفر تلقائياً في جميع نطاقات البرنامج (Global Scopes) دون الحاجة لإعلانها يدوياً. وتكمن الأهمية الوظيفية لهذه المتغيرات في قدرتها على كسر الحواجز البرمجية التقليدية، حيث يمكن الوصول إليها واستدعاء بياناتها من أي مكان داخل الكود، سواء كان ذلك داخل الدوال أو الفئات، مما يسهل عملية معالجة وتداول أنواع مختلفة من البيانات الديناميكية مثل مدخلات المستخدم عبر النماذج $_GET) و $_POST)، وبيانات الخادم($_SERVER) ، ومعلومات الجلسات المستمرة .($_SESSION) وبفضل طبيعتها كمصفوفات ترابطية، تتيح هذه الأدوات للمطورين بناء أنظمة تعليمية مرنة قادرة على استقبال وتخزين ونقل البيانات بين صفحات الموقع المختلفة بكفاءة عالية، مما يحول صفحات الويب من مجرد نصوص ثابتة إلى تطبيقات تفاعلية تستجيب لمدخلات الطلاب ومتغيرات بيئة التشغيل لحظياً.

قائمة المتغيرات العامة الفائقة في لغة:PHP

- **$GLOBALS**
- **$_SERVER**
- $_REQUEST
- $_GET
- $_POST
- $_SESSION
- $_COOKIE
- $_FILES
- **$_ENV**

– $_GET$جلب البيانات عبر معلمات الاستعلام(Query Parameters)

تكمن الخصوصية الوظيفية للمتغير العام الفائق **$_GET**في قدرته على تحويل روابط الصفحات الثابتة إلى أدوات ديناميكية قادرة على حمل البيانات ونقلها عبر بروتوكولHTTP ؛ حيث يتم إلحاق البيانات المرسلة مباشرة بنهاية رابط العنوان (URL) بعد علامة الاستفهام **?**على شكل أزواج من "المفاتيح والقيم–Key) " Value Pairs). وتبرز الأهمية التعليمية لهذا المتغير في العمليات التي لا تتطلب سرية عالية، مثل محركات البحث داخل المواقع، أو التنقل بين صفحات المناهج الدراسية، حيث يتيح للمستخدم إمكانية حفظ الرابط مضافاً إليه البيانات (Bookmark) أو مشاركته مع الآخرين بنفس الحالة التي كان عليها. وبالرغم من سهولة استخدامه، إلا أنه مقيد بحجم محدد من البيانات تفرضه المتصفحات على طول الرابط، كما أنه غير ملائم

نهائياً لنقل المعلومات الحساسة مثل كلمات المرور، نظراً لظهور كافة البيانات بشكل مكشوف وصريح في شريط العنوان وسجلات تاريخ المتصفح.

```
<form method="get" action="search.php">

    اسم الدرس:<br>
    <input type="text" name="lesson_name" /><br><br>


    رقم الصفحة:<br>
    <input type="text" name="page_no" /><br><br>


    <input type="submit" value="عرض المعلومات" />

</form>
```

اسم الدرس:

رقم الصفحة:

عرض المعلومات

أولاً: الجزء الخاص بـ) واجهة النموذج(HTML

هذا الجزء يمثل كيفية بناء المدخلات التي سيتفاعل معها الطالب في المتصفح:

1. **<form method="get" action="search.php">:**

- **method="get":** تحديد وسيلة نقل البيانات لتكون عبر رابط العنوان(URL) ، وهي الطريقة المثالية للتنقل بين الدروس.
- **action="search.php":** تحديد الملف البرمجي الذي سيستقبل هذه البيانات ويعالجها.

2. **name="lesson_name" وname="page_no":**

- هذه هي "**المفاتيح (Keys)** "البرمجية؛ فكل ما يكتبه الطالب داخل المربع سيتم تخزينه تحت هذا الاسم ليتمكن PHP من التعرف عليه لاحقاً.

3. **شكل الرابط الناتج في المتصفح**search.php?lesson_name=PHP&page_no=10 :

- **?:** علامة بدء إرسال البيانات.(Query String)
- **&:** أداة الربط المستخدمة للفصل بين اسم الدرس ورقم الصفحة في الرابط.

```
$_GET['lesson_name'];
```

```
$_GET['page_no'];
```

$_GET['lesson_name']

- مصفوفة فائقة (Superglobal) وظيفتها استدعاء القيمة المرتبطة بمفتاح "اسم الدرس" من رابط المتصفح(URL) ، والتي قام الطالب بإدخالها في النموذج.

$_GET['page_no']

- وظيفتها جلب القيمة المرتبطة بمفتاح "رقم الصفحة" من الرابط، لتمكن النظام من معرفة الصفحة المحددة التي يرغب الطالب في الانتقال إليها.

المسار الإجرائي (كيف يعمل الكود؟):

1. **المستخدم** :يقوم بكتابة "اسم الدرس" و "رقم الصفحة" في الخانات المخصصة، ثم يضغط على زر "**عرض المعلومات.**"
2. **المتصفح** :يجمع البيانات ويقوم بدمجها مباشرة في نهاية رابط العنوان (URL) بعد علامة الاستفهام ?لتظهر هكذا.search.php?lesson_name=PHP&page_no=10 :
3. **الخادم :(Server)** يستقبل الطلب ويقوم "بفك تشفير" الرابط، ثم يرسل هذه القيم إلى محرك.PHP
4. **محرك :PHP** يفتح المصفوفة الفائقة **$_GET**ويخزن بداخلها القيم القادمة من الرابط تحت المفاتيح lesson_name و page_no.
5. **النتيجة** :يتم تنفيذ الأوامر البرمجية المكتوبة بين وسمي <?php ... ?> لاسترجاع البيانات من المصفوفة، ثم عرضها للطالب في صفحة النتائج إن تطلب الأمر ذلك.

$_POSTنقل البيانات الآمن عبر النماذج(Forms)

يُعد المتغير العام الفائق **$_POST**الأداة الأساسية والأكثر أماناً لنقل البيانات الضخمة أو الحساسة من المتصفح إلى الخادم؛ حيث يتم إرسال البيانات ضمن "جسم الطلب (HTTP Request Body) "وليس كجزء من رابط العنوان .(URL) وتبرز الأهمية التعليمية لهذا المتغير في العمليات التي تتطلب خصوصية عالية وسرية تامة، مثل إدخال كلمات المرور، بيانات الدفع، أو رفع الملفات والواجبات الدراسية؛ حيث تظل البيانات مخفية تماماً عن أعين المتطفلين ولا تظهر في شريط العنوان أو سجلات تاريخ المتصفح Browser)

History).وخلافاً لـ $_GET؛ فإن $_POST غير مقيد بحجم محدد للبيانات، مما يجعله الخيار المثالي لإرسال النصوص الطويلة أو المدخلات المتعددة في النماذج المعقدة. وبالرغم من قوتها الأمنية، فإن هذه الطريقة لا تسمح للمستخدم بحفظ الرابط مضافاً إليه البيانات (Bookmark) أو مشاركته بنفس الحالة، حيث تتطلب البيانات إعادة إرسال فعلية في كل مرة يتم فيها تحميل الصفحة.

```
// HTML Form

<form method="post" action="process.php">
    <input type="text" name="username" />
    <input type="password" name="password" />
    <input type="submit" value="Submit" />
</form>

// process.php

<?php

echo "Username: " . $_POST['username'];
echo "Password: " . $_POST['password'];
```

// HTML Form

Submit

// process.php Username: Password:

أولاً: الجزء الخاص بـ(HTML Form)

هذا الجزء هو "الواجهة" التي يراها الطالب أو المستخدم في المتصفح:

1. **<form method="post" action="process.php">:**

- **method**="post": أهم جزء هنا؛ حيث حددنا أن وسيلة نقل البيانات ستكون مخفية (POST) لضمان السرية.
- action="process.php": يحدد "وجهة" البيانات؛ أي عند الضغط على زر الإرسال، اذهب إلى ملف اسمه process.php.

2. **<input type="text" name="username" />:**

- صندوق نصي لإدخال الاسم.
- **:**name="username"هذا هو "المفتاح (Key) "الذي سيتعرف عليه PHP لاحقاً لجلب القيمة.

3. **<input type="password" name="password" />:**

- صندوق لإدخال كلمة المرور (تظهر النجوم أو نقاط بدلاً من الحروف).
- **:**name="password"المفتاح الخاص بكلمة السر.

4. **<input type="submit" value="Submit" />:**

- زر الإرسال الذي يطلق عملية نقل البيانات إلى الصفحة المحددة في action.

ثانياً: الجزء الخاص بـ(process.php)

هذا الجزء هو "المحرك" الذي يعمل خلف الكواليس لاستقبال البيانات ومعالجتها:

1. **<?php ... ?>:** وسم بداية ونهاية كود PHP.

2. **$_POST['username']:**

- هنا استدعينا المتغير العام الفائق $_POST.
- أخبرناه أن يجلب القيمة المخزنة تحت اسم (['username']التي أدخلها المستخدم في الصندوق الأول.(

3. **$_POST['password']:**

- بنفس الطريقة، جلبنا القيمة المخزنة تحت اسم ['password'].

4. **echo "Username: " . ...:**

- دالة echoتقوم بطباعة النص على الشاشة.
- نقطة الربط .تدمج النص الثابت " Username:" مع القيمة القادمة من النموذج.

المسار الإجرائي (كيف يعمل الكود؟):

1. **المستخدم** :يملأ البيانات ويضغط.Submit
2. **الخادم :(Server)** يستقبل البيانات "مخفية" داخل جسم الطلب.(HTTP Body)
3. **محرك :PHP** يفتح المصفوفة الفائقة $_POSTويضع فيها القيم.
4. **النتيجة** :تظهر صفحة جديدة تعرض الاسم وكلمة المرور (وهذا مثال تعليمي فقط، ففي الواقع لا نقوم بطباعة كلمات المرور أبداً).

$_SESSIONإدارة الجلسات المستمرة وحفظ حالة المستخدم

يُعد المتغير العام الفائق $_SESSIONالأداة الجوهرية لتخزين بيانات المستخدم عبر صفحات الموقع المختلفة، حيث يكسر طبيعة بروتوكول" HTTP عدم الاحتفاظ بالحالة .(Stateless) "وتبرز الأهمية التعليمية لهذا المتغير في القدرة على "تذكر" هوية الطالب وخياراته أثناء تنقله بين صفحات المنصة التعليمية دون الحاجة لإعادة إدخال بياناته في كل صفحة. وخلافاً للمتغيرات الفائقة الأخرى التي تنتهي صلاحيتها بمجرد تحميل الصفحة، تظل بيانات الجلسة مخزنة في الخادم (Server) ومرتبطة بمعرف فريد للمتصفح، مما يجعلها وسيلة آمنة وفعالة لإدارة عمليات تسجيل الدخول، وتتبع تقدم الطالب في الاختبارات، والحفاظ على خصوصية بياناته بعيداً عن أيدي المستخدمين الآخرين.

```
session_start();

$_SESSION['user_id'] = "123";
```

أولاً: الجزء الخاص ببدء الجلسة وتخزين البيانات

في هذا الجزء، نقوم بفتح "الملف الخاص" بالمستخدم وحفظ معلومة بداخله:

1. **session_start();:**

- الدالة الأهم على الإطلاق؛ ويجب كتابتها في بداية السكريبت لتخبر الخادم بفتح أو استئناف جلسة حالية للمستخدم.

2. **$_SESSION['user_id'] = "123";:**

- **$_SESSION:** استدعاء المصفوفة الفائقة المسؤولة عن الجلسات.
- **['user_id']:** هو "المفتاح" الذي اخترناه لتمييز المعلومة (مثل الرقم الأكاديمي).
- **"123":** هي القيمة المراد حفظها لتبقى متاحة في كافة الصفحات القادمة.

ثانياً: الجزء الخاص باستدعاء البيانات (في صفحة أخرى)

هنا نوضح كيف يمكن للموقع أن "يتذكر" الطالب في صفحة جديدة تماماً:

1. **session_start();:**

- يجب تكرارها في كل صفحة نريد فيها الوصول للبيانات المخزنة؛ وبدونها لن يتمكن PHP من التعرف على المصفوفة.

2. **echo $_SESSION['user_id'];:**

- يقوم PHP بالبحث في ملفات الخادم عن القيمة المرتبطة بالمفتاح user_id لهذا المستخدم تحديداً ثم يقوم بطباعتها.

المسار الإجرائي (كيف تعمل الجلسة؟):

1. **المتصفح:** يطلب الصفحة لأول مرة، فيقوم الخادم بإنشاء معرف فريد (Session ID) وإرساله للمتصفح.
2. **الخادم (Server):** ينشئ ملفاً صغيراً على السيرفر يحمل نفس المعرف ويخزن فيه البيانات المكتوبة داخل $_SESSION.
3. **التنقل بين الصفحات:** في كل مرة ينتقل فيها الطالب لصفحة جديدة، يرسل المتصفح "المعرف" تلقائياً، فيعرف الخادم من هو الطالب ويجلب بياناته.
4. **النتيجة:** تظهر الصفحة وكأنها "تعرف" المستخدم شخصياً (مثل رسالة: "مرحباً بك يا أحمد") رغم انتقاله لصفحة جديدة.

- $_FILES معالجة ورفع الملفات إلى الخادم

تتمثل الخصوصية الوظيفية للمتغير العام الفائق **$_FILES** في قدرته الفريدة على التعامل مع البيانات "غير النصية"، حيث يُعد الأداة البرمجية المسؤولة عن نقل الملفات من جهاز المستخدم (الطالب) إلى الخادم. وتبرز الأهمية التعليمية لهذا المتغير في المنصات الأكاديمية عند تفعيل خصائص رفع الواجبات الدراسية، الأبحاث بصيغةPDF ، أو الصور التوضيحية؛ حيث يقوم PHP من خلال هذه المصفوفة بجمع كافة تفاصيل الملف المرفوع مثل (اسمه، نوعه، حجمه، ومكانه المؤقت). وخلافاً لـ $_POST و$_GET اللذين ينقلان نصوصاً بسيطة، فإن $_FILES يتطلب إعداداً خاصاً في نموذج HTML لضمان تجزئة الملف ونقله بشكل سليم تقنياً عبر بروتوكول.HTTP

```
<form method="post" enctype="multipart/form-data"
          action="upload.php">

    اسم الطالب:<br>
    <input type="text" name="student_name" /><br><br>

    اختر ملف الواجب (PDF):<br>
    <input type="file" name="homework" /><br><br>

    <input type="submit" value="إرسال الواجب" />

</form>
```

أولاً: الجزء الخاص بـ) واجهة النموذج(HTML

هذا الجزء هو "الواجهة" التي يتفاعل معها الطالب لرفع بياناته وأبحاثه الدراسية:

1. **<form method="post" enctype="multipart/form-data" action="upload.php">:**

- **method="post":** نستخدم هذه الطريقة لأنها الأكثر أماناً وقدرة على نقل البيانات الضخمة (مثل ملفات الأبحاث) التي لا يمكن إرسالها عبر الرابط.
- enctype="multipart/form-data": (العنصر الجوهري) وهي الخاصية المسؤولة عن إعلام المتصفح بضرورة تقسيم الملف إلى أجزاء رقمية صغيرة ليتمكن السيرفر من استقباله، وبدونها يفشل الرفع تماماً.
- **action="upload.php":** تحدد الصفحة البرمجية التي ستستقبل الملف وتقوم بمعالجته.

2. **<input type="text" name="student_name" />:**

- صندوق نصي مخصص لكتابة اسم الطالب.
- **name="student_name":** المفتاح الذي سيستخدمه PHP لاحقاً لجلب اسم الطالب عبر المصفوفة $_POST.

3. **<input type="file" name="homework" />:**

- **type="file":** لتحويل حقل الإدخال إلى زر استعراض (Browse) يسمح للطالب باختيار ملف PDF أو بحث من جهازه.
- **name="homework":** المفتاح البرمجي الذي سيستخدمه PHP للوصول لخصائص الملف عبر المصفوفة $_FILES.

```php
<?php

 $_POST['student_name'] . "<br><br>";

 "<b>تفاصيل الملف المستلم:</b><br>";

 $_FILES['homework']['name'] . "<br>";

 $_FILES['homework']['size'] . " بايت <br>";
"المكان المؤقت للملف: " . $_FILES['homework']['tmp_name'];

?>
```

ثانياً: الجزء الخاص بـ(upload.php)

هذا الجزء يمثل "المحرك" الذي يقرأ البيانات النصية وخصائص الملف في آن واحد:

1. **$_POST['student_name']:**
   - تُستخدم لجلب النص المكتوب في خانة الاسم (مثل: "أحمد علي").
2. **$_FILES['homework']['name']:**
   - وظيفتها جلب الاسم الأصلي للملف الذي قام الطالب برفعه (مثلاً: research_final.pdf).
3. **$_FILES['homework']['tmp_name']:**
   - تشير إلى "المسار المؤقت" الذي وضع فيه السيرفر الملف فور وصوله؛ ليكون جاهزاً للنقل النهائي إلى مجلد الواجبات.

اسم الطالب:

اختر ملف الواجب (PDF):

Choose File No file chosen

إرسال الواجب

المسار الإجرائي (كيف يعمل الكود؟):

1. **الطالب** :يكتب اسمه، يختار ملف البحث، ثم يضغط على زر "**إرسال الواجب.**"
2. **المتصفح** :يقوم بدمج الاسم النصي مع أجزاء الملف المشفرة ويرسلها داخل "جسم الطلب" بفضل خاصية enctype.
3. **الخادم :(Server)** يستقبل الطلب ويقوم بفرز البيانات؛ فيضع الاسم في مصفوفة **$_POST**، ويضع خصائص الملف في مصفوفة **$_FILES.**
4. **النتيجة** :تظهر للطالب رسالة تأكيد تحتوي على اسمه واسم الملف الذي تم رفعه بنجاح، مما يعزز الثقة في التعامل مع المنصة التعليمية.

– $_SERVER مصفوفة معلومات الخادم وبيئة التنفيذ

يُعد المتغير العام الفائق $_SERVER المستودع المعلوماتي الشامل في لغة PHP، حيث يقوم الخادم (Server) بإنشائه تلقائياً لتخزين بيانات حيوية حول مسارات الملفات، عناوين الشبكة، وبيئة التصفح. وتبرز الأهمية التعليمية لهذا المتغير في قدرته على تزويد النظام ببيانات أمنية وإدارية هامة؛ مثل تحديد عنوان IP الخاص بالطالب، ومعرفة اسم الملف الحالي الذي يتم تنفيذه، أو التحقق من نوع المتصفح المستخدم. وخلافاً للمتغيرات التي تنقل بيانات المستخدم ($_GET و $_POST)، فإن $_SERVER يتعامل مع "كواليس" عملية الاتصال بين المتصفح والخادم، مما يجعله أداة أساسية في حماية الموقع وتنظيم مسارات الروابط الداخلية.

أولاً: الجزء الخاص بـ (كود المعالجة)

هذا الجزء يوضح كيف نستخدم المفاتيح البرمجية داخل $_SERVER لاستخراج معلومات النظام (بدون طباعة):

1. **$_SERVER['PHP_SELF']:**
   - يعيد المسار الخاص بالملف الحالي الذي يتم تنفيذه الآن (مثلاً: /course/index.php).
2. **$_SERVER['SERVER_NAME']:**
   - يُستخدم للحصول على اسم المضيف أو "الدومين" الخاص بالموقع التعليمي (مثلاً: www.edu-tech.com).
3. **$_SERVER['HTTP_USER_AGENT']:**
   - يجلب تفاصيل نظام التشغيل ونوع المتصفح الذي يستخدمه الطالب للدخول على المنصة.
4. **$_SERVER['REMOTE_ADDR']:**
   - مفتاح جوهري لجلب عنوان الـ **IP** الخاص بجهاز الطالب، وهو ضروري لعمليات الحماية وتتبع الحضور الرقمي.