

في هذا الريبورت كتبت عن كل شي اعرفه وطبقت عليه بالايام اللي فاتت.

الفرق بين السنجل كوتيشن والدبل كوتيشن .

- السنجل كوتيشن (' ') لا تفسر المتغيرات أو الرموز الخاصة (ماعداء الشرطة \)، وتتميز بأنها أسرع في التنفيذ
- ، لكن أبطأ قليلاً في التنفيذ \n و \t تفسر المتغيرات والرموز الخاصة مثل (" ") الدبل كوتيشن

2. الحلقات (Loops)

- while: تنفذ الكود طالما أن الشرط صحيح
- do while: تنفذ الكود مرة واحدة على الأقل ثم تتحقق من الشرط
- for: تُستخدم عندما يكون عدد التكرارات معروفاً
- foreach: (array) مثالية عند التعامل مع عناصر المصفوفة

3. المتغيرات العالمية والسوبر غلوبالز.

- \$GLOBALS: يمكن استخدامها داخل الدوال للوصول إلى مصفوفة تحتوي على جميع المتغيرات المعرفة بشكل عام
- مثال: \$GLOBALS['x'] = 5;
- \$_SERVER: والمجلدات IP تحتوي على معلومات عن السيرفر مثل اسم السيرفر، عنوان (localhost: مثال) اسم السيرفر: \$_SERVER['server_name'] - للسيرفر IP عنوان: \$_SERVER['server_addr'] - تعرض البيانات الموجودة في الرابط بعد علامة الاستفهام (؟): \$_SERVER['query_string'] -
- \$_REQUEST: ، لكن استخدامها غير مفضل في البيانات الحساسة و POST و GET تجمع بين
- \$_POST: تُستخدم في إرسال بيانات النموذج بطريقة آمنة (لا تظهر في الرابط)
- \$_GET: تُستخدم لنقل البيانات عبر الرابط وغالباً ما تُستخدم في عمليات الفلترة والبحث
- \$_COOKIE: لتخزين البيانات في المتصفح واستعادتها لاحقاً
- \$_SESSION: تُستخدم لحفظ البيانات على السيرفر وتبقى فعالة طوال الجلسة

4. التحقق من صحة البيانات (Validation)

<> عن طريق تحويل البيانات مثل HTML فنكشن تحمي من هجمات السكريبت و ال htmlspecialchars()

< > الى

- empty(): تتحقق إذا كان الحقل فارغاً
- preg_match(): تستخدم تعبيرات منتظمة للتحقق من التنسيق

مثال: التحقق من أن الاسم لا يحتوي على أرقام

```
if (preg_match("/^[a-zA-Z-']*$/", $name))
```

- filter_var(): أسهل وأدق طريقة للتحقق من صحة البريد الإلكتروني:

مثال: filter_var(\$email, FILTER_VALIDATE_EMAIL)

5. دوال مفيدة في PHP

- date(): تعرض التاريخ والوقت.
- include: تُستخدم لإدراج ملفات خارجية، ولا توقف التنفيذ عند الخطأ.
- require: ، لكنها توقف السكريبت إذا لم يتم العثور على الملف مثل include.
- include_once / require_once: تتأكد من عدم تضمين الملف أكثر من مرة.
- __construct(): دالة خاصة تُستخدم داخل الكلاسات وتُنفذ تلقائيًا عند إنشاء كائن جديد وظيفتها تسهيل و ترتيب الكود.

6. التعامل مع الملفات (Files)

- file_exists(): تتحقق من وجود الملف.
- file_put_contents(): لحفظ البيانات داخل ملف.
- dirname(): تُرجع اسم المجلد الأب لمسار معين.
- basename(): تُرجع اسم الملف من المسار.
- __FILE__: تُرجع المسار الكامل للملف الحالي.
- __DIR__: تُرجع مسار المجلد الحالي.

7. (Regular Expressions)

فنكشن تتأكد ان اليوزر دخل المعطيات مثل ما عطيته على سبيل المثال اذا تبينه يكتب اسمه بدون ارقام تسوي

```
If (preg_match("/^[a-zA-Z-']*$/", $name)) {  
  
}
```

8. PHP (PHP OOP) البرمجة الكائنية التوجه في

(What is OOP?)

، بحيث يتم بناء البرنامج باستخدام (Objects) هي أسلوب برمجي يعتمد على الكائنات (OOP) البرمجة كائنية التوجه في تنظيم الكود، وإعادة استخدامه، وتقسيمه إلى OOP كائنات تحتوي على بيانات (خصائص) ودوال (وظائف). تساعد وحدات صغيرة.

(Classes/Objects) الكلاسات والكائنات

أما الكائن فهو (Methods) ودوال (Properties) الكلاس هو قالب يُستخدم لإنشاء الكائنات، ويحتوي على خصائص نسخة من الكلاس يمكن استخدامها وتنفيذ وظائفه.

Constructor المُنشئ

تُستخدم لتهيئة القيم الأولية. هي دالة تُنفذ تلقائيًا عند إنشاء كائن جديد من الكلاس (`__construct()`).

Destructor المُدمر

تُنفذ تلقائيًا عند الانتهاء من استخدام الكائن أو عند خروجه من الذاكرة. تُستخدم عادة لتنظيف الموارد (`__destruct()`).

(Access Modifiers) محددات الوصول

تتحكم في صلاحية الوصول للخصائص والدوال داخل الكلاس:

- متاح من أي مكان: **public** -
- متاح فقط من نفس الكلاس أو كلاس وراث: **protected** -
- متاح فقط داخل نفس الكلاس: **private** -

(Inheritance) الوراثة

تساعد على إعادة `extends` تسمح بإنشاء كلاس جديد يرث خصائص ودوال من كلاس آخر باستخدام الكلمة المفتاحية `extends`. استخدام الكود.

(Constants) الثوابت

`const`: ثابت لا يمكن تغيير قيمته. يتم تعريفه داخل الكلاس باستخدام

مثال: `const VERSION = 1.0;`

(Abstract Classes) الكلاسات المجردة

يتم توريثها وإجبار الكلاسات الابنة على تنفيذ. كلاسات لا يمكن إنشاء كائن منها مباشرة، تحتوي على دوال بدون تنفيذ الدوال المجردة.

(Interfaces) الواجهات

ويلزم تنفيذ جميع الدوال `implements` تشبه الكلاسات المجردة لكن لا تحتوي على تنفيذ للدوال أبدًا. يتم استخدام الكلمة الموجودة في الواجهة

Interfaces make it easy to use a variety of different classes in the same way. When one or more classes use the same interface, it is referred to as "polymorphism".

Different between abstract and interface

Interface use **implement** keyword

While abstract use **abstract** keyword

Interface cannot have priorities while abstract can

All interface methods must be public, while abstract class methods is public or protected

Classes can implement an interface while inheriting from another class at the same time

Abstract cannot have multiable inheritance

اثنيهم مايقدرين يسوون كائن جديد

Traits

PHP only supports single inheritance: a child class can inherit only from one single parent.

So, what if a class needs to inherit multiple behaviors? OOP traits solve this problem.

(Static Methods/Properties) الدوال والخصائص الثابتة

::تُستخدم بدون إنشاء كائن من الكلاس. يتم الوصول لها باستخدام
مثال: ClassName::staticMethod();

Namespaces

. `namespace` تُعرف باستخدام الكلمة. تُستخدم لتنظيم الكود وتفادي تعارض الأسماء بين الكلاسات أو الدوال

Iterables

Iterator ، مثل المصفوفات والكائنات التي تطبق واجهة foreach تُشير إلى أي قيمة يمكن المرور عليها باستخدام