

Uppgifter

Sammanställ minst 10 saker ni alla tycker gör kod lätt att förstå.

1. Bra namngivning +
2. Kommentarer +
3. Bra indentering +
4. Inte för långa rader
5. MVC-mönstret var tydligt +
6. Viktigt att hålla sig till mönstret +
7. Filstruktur
8. Ett flöde som är lätt att följa
9. Konsekvent med namngivning +
10. Hur man använder variabler och hur man deklarerar dem

Sammanställ 10 saker som gör kod svår att förstå.

1. Dålig namngivning
2. Inga kommentarer +
3. Avsaknad av indentering
4. För långa rader
5. Sekventiell kodning (spagethikod), avsaknad av objektorientering och mönster
6. Dålig filstruktur, alla filer ligger i roten
7. Otydliga flöden
8. Inkonsekvent namngivning
9. Att inte deklarera variabler och att inte göra det på ett ställe.
10. Avsaknad av måsvingar

Ge exempel på vad ni som grupp tycker är ett "fulhack".

Något som löser problemet för stunden, men inte i längden.

Icke-generiska lösningar.

Att skriva ett fulhack för att lösa problemet tillfälligt, för att komma vidare är inte ett problem. Det blir ett problem OM man inte löser det innan man lämnar ifrån sig koden.

Finns det något i PHP som gör det svårt att undvika dessa "fulhack"? Finns det något sätt runt det?

För att att hålla applikationen vid liv, måste man hela tiden se till att skicka med information när sidan laddas om. Stateless. Är det ett fulhack eller är det något som är accepterat?

Är det för att skrivbordsapplikationer är "finare" som hanteringen av webbapplikationers statelessa tillstånd anses vara fulhack. Skrivbordsapplikationer är normen?

Var ska måsvingen sitta? Samma rad eller raden under?

```
if () {  
}
```

Gruppen bestod av:

Juhani Aavanen, David Söderberg, Maria Nygren, Ulrika Falk

<http://www.php-fig.org/psr/psr-1/>

<http://pear.php.net/manual/en/standards.php>