



Keywords/Questions:

Notes:

Code Refactoring Principles

5aWVWUfad`YBd`UbWZ

#Z DVSTI[1fk

ZeWVkfZ`YadS`lIW`S`aYUS`iSk1

ZeWVkfZ`YUa_ _WfWea fZSf kag US`g`WHS`Vi ZSf WUZ bScf aX

Readability

fZWaWWeSTagf1

-When you come back in a year to try and understand your code, can you quickly understand what's going on?

Modularity

SZ ? aVg`Sdfk

-Is how easy it is to narrow down your code when you encounter errors

-If one particular part of your web site breaks down, is your code modular

Efficiency

enough that you would be able to narrow down on the exact section of code or code file that's responsible for the problems that's occurring

-Is how divided your code is to different parts.

Length

%Z 7XUWk

-How fast does your code run?

-It is less important compared to keeping your code modular & readable.

Code golf

&Z >WYZ- don't repeat yourself in your code so that you keep your code well

structured. If it will become less readable, don't reduce length.

Summary:

-Code golf:-game writing shortest code, codegolf.stackexchange.com

4 Code Refactoring Principles:-

1. Readability:- using comments -can you read and understand your code easily after a year

2. Modularity:- your code is divided to different smaller parts. -easy to track errors and mistakes.

3. Efficiency:- speed of your code -C is faster -not as important as the above 2

4. Length:- avoid repeating. Don't make it short, if it makes your code incomprehensible.

Code golf:- writing a code as short as possible competition.