Programming Style

Importance

Code should be readable so we can easily know what it'll do when we run it. We can make our code beautiful by indenting, commenting, and naming our variables with care. When writing code, always assume other people will read it. Be on your best behavior!

Indentation

Indentation is for creating logical seperation from nested blocks and orginaztion. Sometimes, saying nothing is better than getting it all out at once.

Comments

Comments should be written in plain english to explain any functions and outputs so readers can easily see the function of the code. Cut to the chase.

Case

- Classes: camel case with first letter of each word capitalized (like class ILikeChocolate).
- Interfaces: camel case with first letter capitalized (ILikeChocolate).
- Variable and methods: camel case with first letter lowercased (iLikeChocolate).
- Constants: all uppercases with underscores (NUMBER_OF_PEOPLE)

Naming conventions

Use descriptive variable names. Variables like x should only used for temporary variables. For naming methods, try to describe briefly what the method does in the method. Booleans should make it clear what the variable means if we set it to either `true` or `false`.

Consistency

If you like to whitespace after an equals sign like `(x + y = z)`, keep consistant. Don't do `(a+b+c=z)` and change up the spacing. This goes for whitespaces and brace layout. Make sure your cake looks beautiful all around.

Pseudocode

Pseudocode are the blueprints of your code. Contains logic without worrying about syntax. Kind of like sketching in a drawing, you're going from just the logic of your thinking, to the language of your robot. Slow and steady, piece by piece.

For example, to create a divider, we would create our pseudocode:

```
While we have user input,
(if input is valid)
        Get numerator
        Get denominator
        Calculate the quotient
        Print the quotient
(else)
        Ask for more input
```