Summary Report
Art of Readable Code
Chapter 03

KEY IDEA

Actively scrutinize your names by asking yourself, "What other meanings could someone interpret from this name?"

- **Prefer min and max for (Inclusive) Limits**
  - The clearest way to name a limit is to put max_ or min_ in front of the thing being limited.
- **Prefer first and last for Inclusive Ranges**
  - Here is another example where you can't tell if it's "up to" or "up to and including":
  - print integer_range(start=2, stop=4)
    # Does this print [2,3] or [2,3,4] (or something else)?
  - Although start is a reasonable parameter name, stop can be interpreted in multiple ways here.
  - For inclusive ranges likes these (where the range should include both end points), a good choice is first/last. For instance:
  - set.PrintKeys(first="Bart", last="Maggie")
  - Unlike stop, the word last is clearly inclusive.
  - In addition to first/last, the names min/max may also work for inclusive ranges, assuming they "sound right" in that context.
  - *In assigning names on a variable that sets ranges, it is better to use 'first/last' than 'start/end' it is usually confusing because we'll never know if the last value should be included or not.*
- **Prefer begin and end for Inclusive/Exclusive Ranges**
  - In practice, it's often more convenient to use inclusive/exclusive ranges.
- **Naming Booleans**
  - When picking a name for a boolean variable or a function that returns a boolean, be sure it's clear what true and false really mean**.**

- **Matching Expectations of Users**

    - Some names are misleading because the user has a preconceived idea of what the name means, even though you mean something else. In these cases, it's best to just "give in" and change the name so that it's not misleading.
    - The word "inherit" is familiar to most programmers, and it's understood that further modifications are made after inheritance. With class inheritance, you get all the methods and members of another class and then modify them or add more.

Summary

- The best names are ones that can't be misconstrued—the person reading your code will understand it the way you meant it, and no other way.
- The best names are resistant to misinterpretation.
- When it comes to defining an upper or lower limit for a value, max_ and min_ are good prefixes to use.
- For inclusive ranges, first and last are good.
- For inclusive/exclusive ranges, begin and end are best because they're the most idiomatic.
- When naming a boolean, use words like is and has to make it clear that it's a boolean. Avoid negated terms (e.g., disable_ssl).
- Beware of users' expectations about certain words. For example, users may expect get() or size() to be lightweight methods.