Naming

Importance of Naming

- Critical issue for software engineering
- Software Engineers make decisions about names constantly
- Naming impacts the readability and comprehension of software
- Good names reduce the cost of development
- Careful selection of names can convey the high-level meaning of a task to the developer

Words of Wisdom

- If you have a good name for a method, you do not need to look at the body - Fowler et. al.

- There are only two hard things in Computer Science: cache invalidation and naming things - Phil Karlton

- There are 2 hard problems in computer science: cache invalidation, naming things, and off-by-1 errors Leon Bambrick

What is a name?

- Technical term: *identifier*

- Formed of:
  - Lowercase letters: 'a'..'z'
  - Uppercase letters: 'A'..'Z'
  - Digits: '0'..'9'
  - Underscore: '_’

- First character: letter or underscore

Exceptions to the Rules

- Reserved Words: [Python Reserved Words]
- *contextual keywords - identifiers with special meaning*
- Starting with an underscore ‘_’ is an convention to indicate that an identifier is private (NOT enforced by the Python interpreter)

Things to Name

- variables
- fields
- parameters
- types
- functions
- methods
- classes

General Rules

- Names should not include *types* or any symbol for the types. I.e., the "Hungarian Notation" is not a good style
- variables, parameters, types, classes, fields, classes, files:
  - *entities*, i.e., *things*
  - therefore, use *nouns*
- functions/methods:
  - *actions*
  - therefore, use *verbs*
- This presentation focuses on function/method names

Method Naming Standards

Naming Standards

- There are many naming practices

- On the Naming of Methods: A Survey of Professional Developers
  - Survey of 1100+ developers
  - Wide support for these standards
  - Across multiple programming languages
  - Very slight differences based on experience

- Paper is referring to **methods**, but also applies to **free functions**

List of Naming Standards

- Naming Style
- Grammatical Structure
- Verb Phrase
- Dictionary Terms
- Full Words
- Idioms and Slang
- Abbreviations & Acronyms
- Prefix/Suffix
- Length

Naming Style

```
getFullName    getScriptState  call_with_default
garbage_collection check_static_allocation_size
getfullName getscriptstate
```

- Popular styles: camelCase and under_score
  - Python uses **under_score** for functions
- Little difference in cognitive load during comprehension
  - Pretty equivalent, except that camelCase appears to have a slight advantage

Grammatical Structure

```
registerManagedResource managedResourceRegister
```

- Function names with multiple words need to have a grammatically correct sentence structure
    - **verb phrase**
        - drawContentBorder
    - **verb phrase with a prepositional phrase**
        - performTestsFromZipFile
    - **noun phrase**
        - nextArea

Verb Phrase

```
manage_caching_sizes
computeProductBlockingSizes   get_cached_node
```

- Not only grammatically correct but contain a verb or a verb phrase

- Functions are actions

Dictionary Terms

```
findLength   abcdefg   cccc   aa2020
```

- Words used in the function name are actual dictionary terms
- Meaningful natural-language terms
- Not non-dictionary word, e.g., "2" instead of "to"
- Use of non-dictionary words make an otherwise high-quality identifier hard to understand

Full Words

```
startConnection  c
```

- Use full words as opposed to single-letter identifier names

- Research clearly shows improved comprehension with full-word identifiers instead of single-letter names

Idioms and Slang

```
/* personal name */ fido
/* idiom */ cutting_corners
/* slang */ curveball
```

- The name should not contain personal expressions, idioms, or slang
- Special case of *Dictionary Terms* and *Full Word* standards
- Using slang and idioms referred to as a *cute* practice
- Choose clarity over entertainment value

Abbreviations & Acronyms

```
getStr   pyConnection  get_algo
close_db_connection   open_GUI_interface
get_URL   get_FIFO   traverse_DOM_tree   repr
getProtoNameNode   get_QWE   sendAAAA
```

- If a name contains words that are abbreviations or acronyms, they should be well known or standard
- Standard ones used by the organization or domain
- **evil abbreviations** - unfamiliar abbreviations that are very hard for programmers to understand

Prefix/Suffix

```
gimpItemGetPath   swift_stdlib_u_char   m_count
```

- Method names should not contain a prefix or suffix that is a term from the system
- Does not apply to the C language or groups of related free functions
- Developers quickly learn to ignore the prefix/suffix, and they become markers of older code.
- Names should not contain scope or type information.

Length

```
Returnfalseifnosetterwasfoundandifreportnosetterfoundisfalse
```

- Long debate in the research literature and developer discussion
- Longer, more descriptive have a positive impact on code comprehension
- Fixed maximum number of words in the name, from 1 to 5
- Some disagreement in the results, as some developers stated that there should be no maximum length in all cases
- However, almost all agreed to around 5 in almost all cases
- One exception: Name for test-case methods

Summary

- Presented standards are widely accepted in the developer community
- Quality of identifiers has a strong impact on code comprehension
- Although survey was for methods, almost all the standards apply to free functions
- Will follow these naming conventions with PEP 8 standards in this course

PEP 8 Naming Rules

- Functions are **under_score()**
- Variables are **under_score** or **CAPITAL_UNDER_SCORE** for constants
- Class Names are **PascalCase** (referred to as **CapWords** in PEP 8)