

Linking Code Readability, Structure, and Comprehension among Novices: It's Complicated

発表者：ダニエル

Eliane S. Wiese
University of Utah

Anna N. Rafferty
Carleton College

Armando Fox
University of California, Berkeley

背景

- 初心者のコード構造が貧弱だと、プログラミングに対する理解が浅いことが示唆される
- エキスパートレベルのコードが読みにくいと感じた初心者は、理解を妨げる場合は、適切なスタイルの実践を継続しない可能性がある
- 初心者が読みやすさとスタイルをどのように認識しているかを理解することは、より優れた指導とツールを設計するために重要

目的

- 初心者のスタイルと読みやすさの認識が、専門家の認識とどのように一致するかを調査する
- 初心者の読みやすさの好みが、実際の理解とコード作成行動にどのように関係するかを調べる

調査方法/ 研究デザイン

参加者: 大学のコンピューターサイエンス学部の学生 27 名

調査(survey)の構成:

- 自己申告による習熟度: 学生に Python のイディオム (リスト、セットなど)

コード理解に関する質問:

- 関数型コード (5 ~ 15 行)
- 特定の入力に対する出力を学生に判断してもらった

スタイルと読みやすさに関する質問:

- 同じ機能を持つ 4 つのコードブロックを示した。
- スタイルと読みやすさに基づいて学生に順位付けしてもらった。

調査方法/ 研究デザイン

Key Evaluations:

- Readability
- Style
- Comprehension

```
def func1(mat):  
    new_mat = []  
    for row in mat:  
        row_rest = row[1:]  
        new_mat.append(row_rest)  
    return new_mat
```

```
def func2(mat):  
    return [mat[i][1:] for i in range(len(mat))]
```

```
def func3(mat):  
    return [row[1:] for row in mat]
```

```
def remove_first_element(x):  
    return x[1:]
```

```
def func4(mat):  
    map(remove_first_element, mat)
```

#Recall that map() returns a list that results from applying map's first argument to every element of the collection given as its second argument

Optimal: Most stylistically elegant solution.

Naïve: Avoided idioms, used longer code with primitive features.

Metrics (NOVICE VS. EXPERT PATTERNS)

1. For exclusive cases, writing a series of `if` statements rather than using `else if`.
2. Conjoining conditions using nested `if` statements rather than the `&&` operator.
3. Repeating code inside an `if` block and its corresponding `else` block.
4. Including extraneous cases, with a general solution that covers those special cases.
5. Returning a boolean value by checking a condition with an operator (e.g., `==`, `>`, `<`) by using an `if` statement and explicitly returning `true` or `false` rather than simply returning the condition.
6. The previous pattern for conditions with method calls instead of operators.
7. Using a `while` loop when a `for` loop is more appropriate.

Research Question and 調査結果

RQ1: Do novice programmers find common novice patterns to be more readable than expert patterns?

TABLE I
SURVEY RESULTS: SELECTION OF EXPERT CODE AS MOST READABLE AND BEST STYLED, AND COMPREHENSION ACCURACY

Topic	Agreed Expert Code Was:			z-test	McNemar		Comprehension	
	Most Readable	Best Styled	Both				Good Style	Poor Style
Boolean Returns with Operator	39%	71%	36%	$p < .0001^*$	$p < .0001^*$	$\chi^2 = 62$	91%	93%
Boolean Returns without Operator	58%	86%	52%	$p < .0001^*$	$p < .0001^*$	$\chi^2 = 47$	95%	94%
Special Cases with General Solution	71%	94%	70%	$p = .0008^*$	$p < .0001^*$	$\chi^2 = 50$	72%	81%
Repeating Code within <code>if</code> and <code>else</code>	24%	25%	10%	$p < .0001^*$	$p = .80$	$\chi^2 = 0.1$	86%	86%
For vs. While loop	84%	81%	74%	$p = .92$	$p = .31$	$\chi^2 = 1.4$	67%	76%
<code>if</code> statements vs. <code>else if</code> (exclusive cases)	86%	92%	82%	$p = .99$	$p = .020$	$\chi^2 = 6.1$	92%	86%
Nested <code>if</code> statements vs. <code>&&</code>	90%	89%	84%	$p > .99$	$p = .86$	$\chi^2 = 0.2$	75%	77%

Research Question and 調査結果

RQ1: Do novice programmers find common novice patterns to be more readable than expert patterns?

7つのトピックのうち3つについては、大多数の学生が初心者向けコードの方が専門家向けコードよりも読みやすいと選択した。

TABLE I

SURVEY RESULTS: SELECTION OF EXPERT CODE AS MOST READABLE AND BEST STYLED, AND COMPREHENSION ACCURACY

Topic	Agreed Expert Code Was:			z-test	McNemar		Comprehension	
	Most Readable	Best Styled	Both				Good Style	Poor Style
Boolean Returns with Operator	39%	71%	36%	$p < .0001^*$	$p < .0001^*$	$\chi^2 = 62$	91%	93%
Boolean Returns without Operator	58%	86%	52%	$p < .0001^*$	$p < .0001^*$	$\chi^2 = 47$	95%	94%
Special Cases with General Solution	71%	94%	70%	$p = .0008^*$	$p < .0001^*$	$\chi^2 = 50$	72%	81%
Repeating Code within if and else	24%	25%	10%	$p < .0001^*$	$p = .80$	$\chi^2 = 0.1$	86%	86%
For vs. While loop	84%	81%	74%	$p = .92$	$p = .31$	$\chi^2 = 1.4$	67%	76%
if statements vs. else if (exclusive cases)	86%	92%	82%	$p = .99$	$p = .020$	$\chi^2 = 6.1$	92%	86%
Nested if statements vs. &&	90%	89%	84%	$p > .99$	$p = .86$	$\chi^2 = 0.2$	75%	77%

Research Question and 調查結果

RQ2: Do novice programmers believe that what is most readable to them also matches experts' style preferences?

TABLE I
SURVEY RESULTS: SELECTION OF EXPERT CODE AS MOST READABLE AND BEST STYLED, AND COMPREHENSION ACCURACY

Topic	Agreed Expert Code Was:			z-test	McNemar		Comprehension	
	Most Readable	Best Styled	Both				Good Style	Poor Style
Boolean Returns with Operator	39%	71%	36%	$p < .0001^*$	$p < .0001^*$	$\chi^2 = 62$	91%	93%
Boolean Returns without Operator	58%	86%	52%	$p < .0001^*$	$p < .0001^*$	$\chi^2 = 47$	95%	94%
Special Cases with General Solution	71%	94%	70%	$p = .0008^*$	$p < .0001^*$	$\chi^2 = 50$	72%	81%
Repeating Code within <code>if</code> and <code>else</code>	24%	25%	10%	$p < .0001^*$	$p = .80$	$\chi^2 = 0.1$	86%	86%
For vs. While loop	84%	81%	74%	$p = .92$	$p = .31$	$\chi^2 = 1.4$	67%	76%
<code>if</code> statements vs. <code>else if</code> (exclusive cases)	86%	92%	82%	$p = .99$	$p = .020$	$\chi^2 = 6.1$	92%	86%
Nested <code>if</code> statements vs. <code>&&</code>	90%	89%	84%	$p > .99$	$p = .86$	$\chi^2 = 0.2$	75%	77%

考察

| Identified topics where students prefer or use novice patterns.

| **Coding style is multi-faceted:** Students may recognize good style even if they are unable to produce it themselves.

| many novices **recognized expert preferences** even when they didn't find the code more readable.