Identifier Readability

Experiment 2, Experimentation & Evaluation 2024

Abstract

This experiment investigates whether the readability of source code is influenced by the format of composed identifiers, specifically comparing camelCase to kebab-case. Inspired by prior research indicating that natural language is read faster with explicit word separators, we designed a controlled experiment to test if this concept applies to source code. Participants were tasked with identifying a target identifier from distractors written in both styles while tracking their response times and accuracy. Data was collected from at least 10 participants via a web application. The experiment results aim to determine if one style leads to faster recognition and improved readability in source code, contributing to best practices in software development.

1. Introduction

Studies in natural language processing show that people read faster when words are visually separated, such as "reading_this_text" vs. "readingthistext". This experiment explores if the same principle applies to source code identifiers, comparing camelCase (e.g., "moveSouth") with kebab-case (e.g., "move-south").

The research question is: **Does the use of explicit separators (kebab-case) improve the speed and accuracy of identifying source code identifiers compared to camelCase?** This is relevant for software development as enhanced readability can improve productivity and reduce errors.

We designed a controlled experiment where participants identify target identifiers from a list, tracking time, and accuracy. The goal is to determine if kebab-case provides a measurable advantage over camelCase, offering insights into optimal coding conventions.

Hypotheses:

Hypothesis 1: Participants will identify identifiers written in camelCase faster than those written in kebab-case.

Independent Variable: The identifier style (camelCase vs. kebab-case).
Dependent Variable: The time taken to identify the correct identifier.

Hypothesis 2: Participants with more programming experience will identify identifiers written in camelCase more accurately than those written in kebab-case.

Independent Variable: The identifier style (camelCase vs. kebab-case).
Dependent Variable: The time taken to identify the correct identifier.

Hypothesis 3: Participants with higher English proficiency will identify identifiers in both camelCase and kebab-case faster than those with lower English proficiency, with a greater performance difference observed for kebab-case identifiers.

Independent Variable: The identifier style (camelCase vs. kebab-case) and the participant's level of English proficiency.

Dependent Variable: The time taken to identify the correct identifier in camelCase vs. kebab-case.

2. Method

In the following subsections, describe everything that a reader would need to replicate your experiment in all important details.

2.1 Variables

Independent variable	Levels
Naming conventions	- kebab-case - camelCase

Dependent variable	Measurement Scale
Time in milliseconds	The time it took the person to choose the correct answer.

Control variable	Fixed Value
Tool for the experiment(framework)	Web framework that was used for developing an experiment.
Experiment page design	Constant design to mitigate potential distraction and ensure a consistent, ease-of-use environment for participants.

Blocking variable	Levels
Programming Experience	- Less than one year - One to three years

	- More than three years
Gender	- Male - Female
English Proficiency	- Native language - Second language (fluent or near fluent) - Foreign language (basic or moderate proficiency)
Age	- Age of participant

2.2 Design

Type of Study (check one):

□ **Observational Study**	□ **Quasi-Experiment**	✅ **Experiment**

Number of Factors (check one):

□ **Single-Factor Design**	✅ **Multi-Factor Design**	□ Other

Between vs. Within (check one): [for human subject studies]

□ **Between Group Design** (independent measures)	✅ **Within Subject Design** (repeated measures)	□ Other

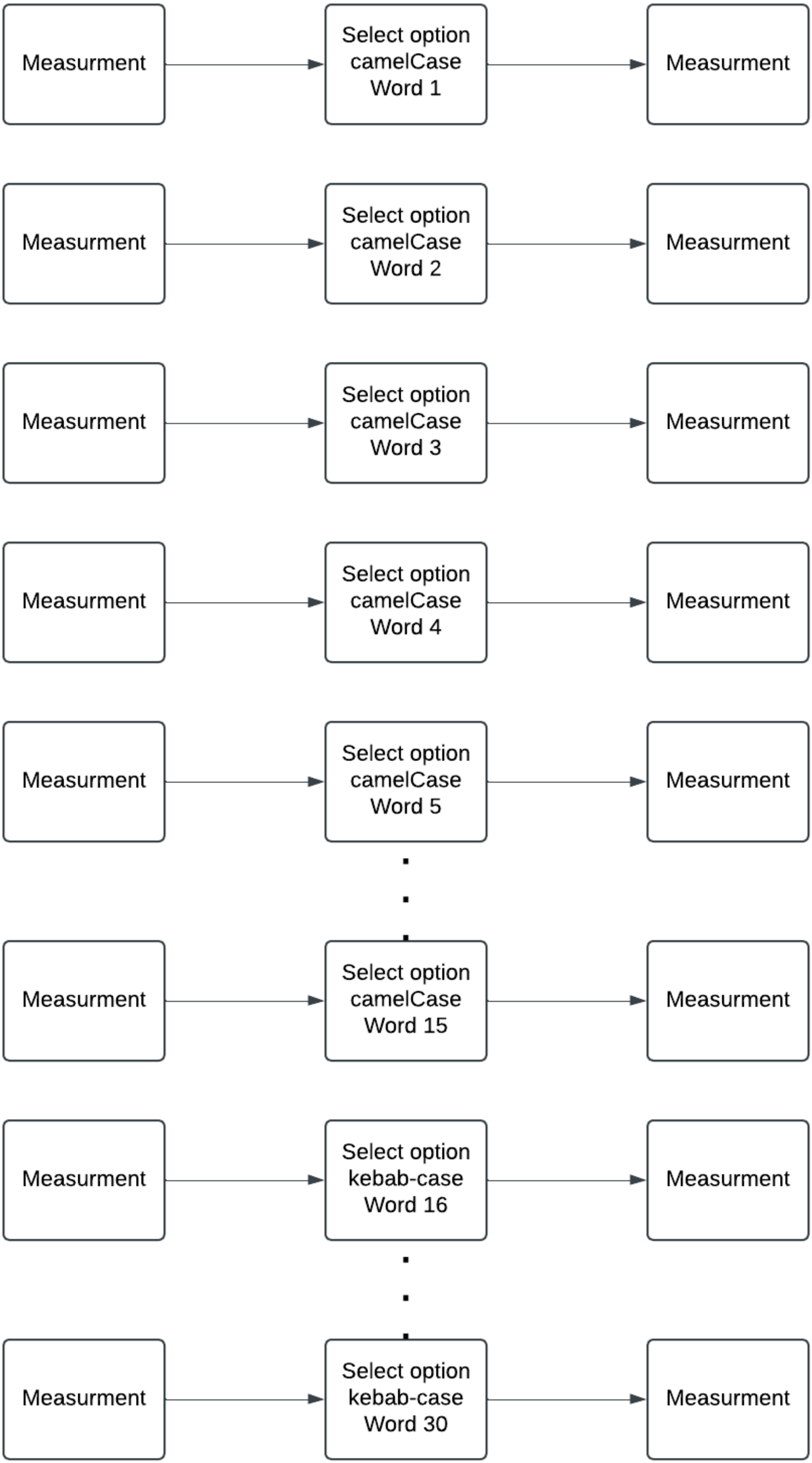
This is an **experiment** because we're testing how two different **naming conventions** — camelCase and kebab-case — affect how quickly and accurately participants can identify the correct identifier. The independent variable is the naming style, and the dependent variables are the time it takes to find the correct answer (measured in milliseconds) and whether they got it right or not.

We're using a **multi-factor design** since we're not just looking at the naming style but also considering factors like programming experience, English proficiency, and others like gender and age. These factors help us see if certain groups perform differently based on the naming convention. Also ensured controlled conditions by using the same web application, to ensure consistency in the results.

The experiment follows a **within-subject design**, meaning that each participant performs tasks with camelCase and kebab-case, which minimizes the variability. This makes it easier to compare how the same person performs with both styles.

Each participant completes 30 tasks: 15 for camelCase and 15 for kebab-case. For each task, they see four options, where only one is correct, and the other three are there to distract them. We record how long it takes them to choose and whether they pick the correct option. This is visible from the figure below, where the select option is our four options and they are repeated for each of the cases.

In summary, this experiment uses a multi-factor design and a within-subject approach to examine the effects of camelCase and kebab-case identifiers on participants. By measuring time and accuracy, while controlling for individual differences and experimental conditions, the study aims to provide clear insights into the impact of variable naming conventions.


Measurment
Select option camelCase Word 1
Measurment
Measurment
Select option camelCase Word 2
Measurment
Measurment
Select option camelCase Word 3
Measurment
Measurment
Select option camelCase Word 4
Measurment
Measurment
Select option camelCase Word 5
Measurment
Measurment
Select option camelCase Word 15
Measurment
Measurment
Select option kebab-case Word 16
Measurment
Measurment
Select option kebab-case Word 30
Measurment

2.3 Participants

The participants of the experiment primarily consist of students with Informatics degrees, with a smaller representation from other fields. While the majority are male, efforts were made to achieve a more balanced gender distribution. Participants are aged between 18 and 45, reflecting a diverse range of backgrounds and levels of experience.

All participants are either graduates or currently in the process of completing their degree, though not all have studied Informatics. Language proficiency was considered an important variable, as it was expected to influence the effectiveness of conducting the experiment, particularly for native speakers.

Recruitment was largely facilitated through an invitation link, allowing participants to easily join the experiment. The pool of experimenters includes classmates, friends, and some professors, forming a diverse and accessible group.

Unlike traditional experiments, we chose not to divide participants into control and experimental groups. Instead of allocating half of the participants to perform tasks using kebab-case and the other half using camelCase, we designed a single integrated test. This test involves all participants engaging in both kebab-case and camelCase tasks. Using this approach, we aimed to evaluate the performance of each participant across both naming conventions, ensuring that every individual experienced and contributed data for both cases. This decision allowed us to avoid potential biases that might arise from group allocation and provided a more comprehensive dataset for comparing the two naming conventions within the same participant population.

2.4 Apparatus and Materials

The experiment was conducted using a web application developed with the Nuxt framework. This application was hosted on Netlify hosting, and all interactions were performed via a standard web browser. Time measurements during the experiment were handled using a built-in function within the application, ensuring consistent and accurate timing.

2.5 Procedure

The experiment was conducted online and hosted on a web application. Participants accessed the experiment by clicking a provided link, which directed them to the application’s welcome page. On this page, participants were presented with a brief explanation of the experiment’s purpose and clear instructions on how to proceed.

Participants were required to select the correct kebab-case or camel-case format corresponding to a given main collocation displayed on the screen. Once they made their selection, they submitted their choice, and the system automatically recorded their response and time taken to complete the task using a built-in function.

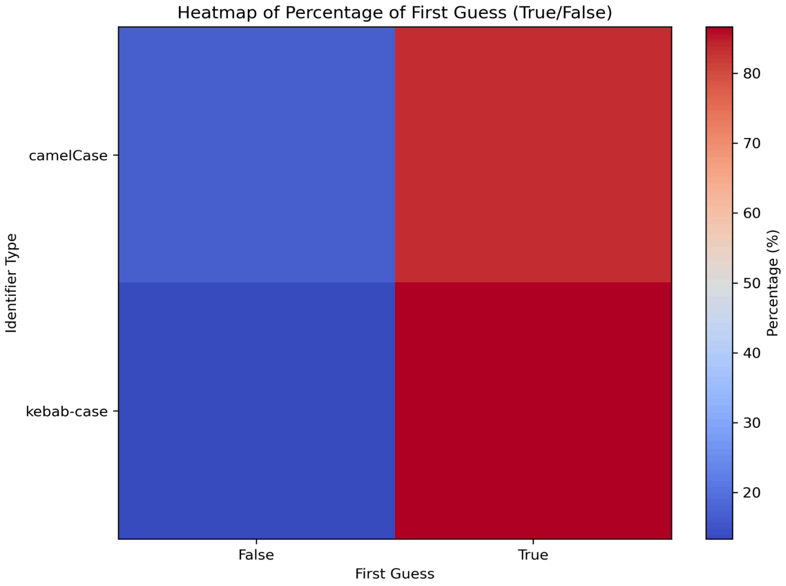
No further actions or interventions were required from the participants after reading the instructions and completing the task. The entire process was automated to ensure consistency and accuracy in data collection.

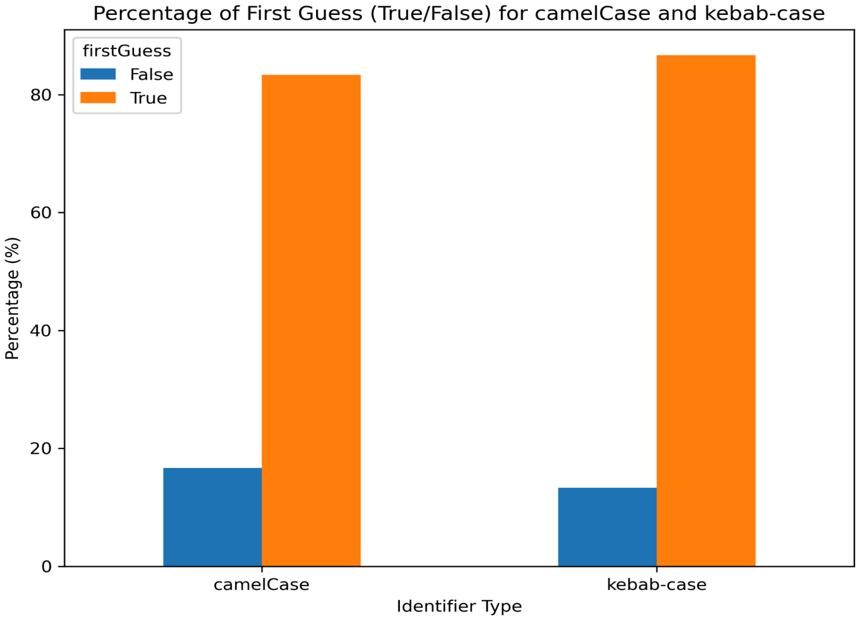
3. Results

3.1 Visual Overview

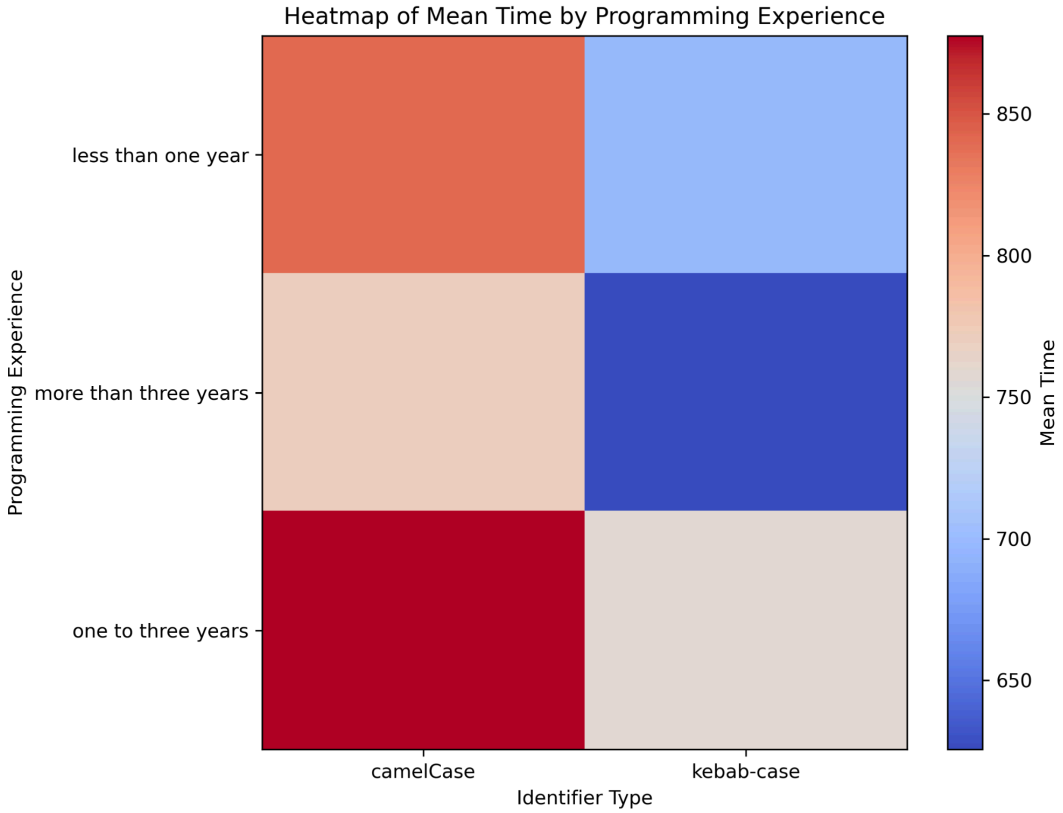
The following graphs are the Bar Chart and the Heat Map, they both show similar results, but the heat map allows us to see the results in more detail and how they change for each of the cases.

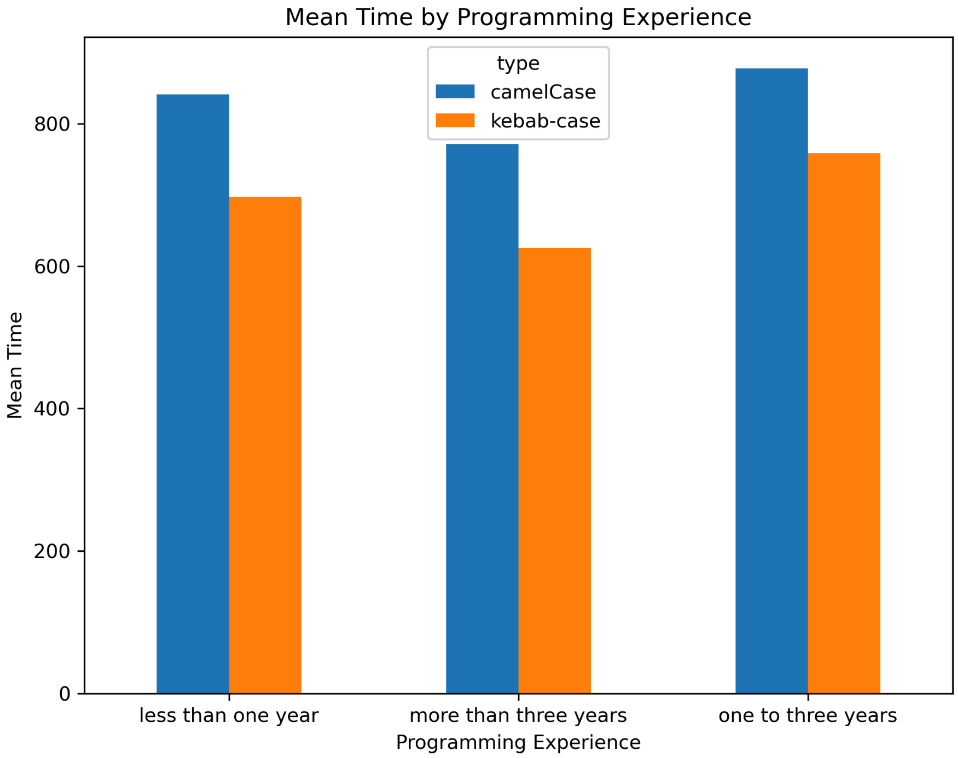
Percentage of the first guess for identifiers



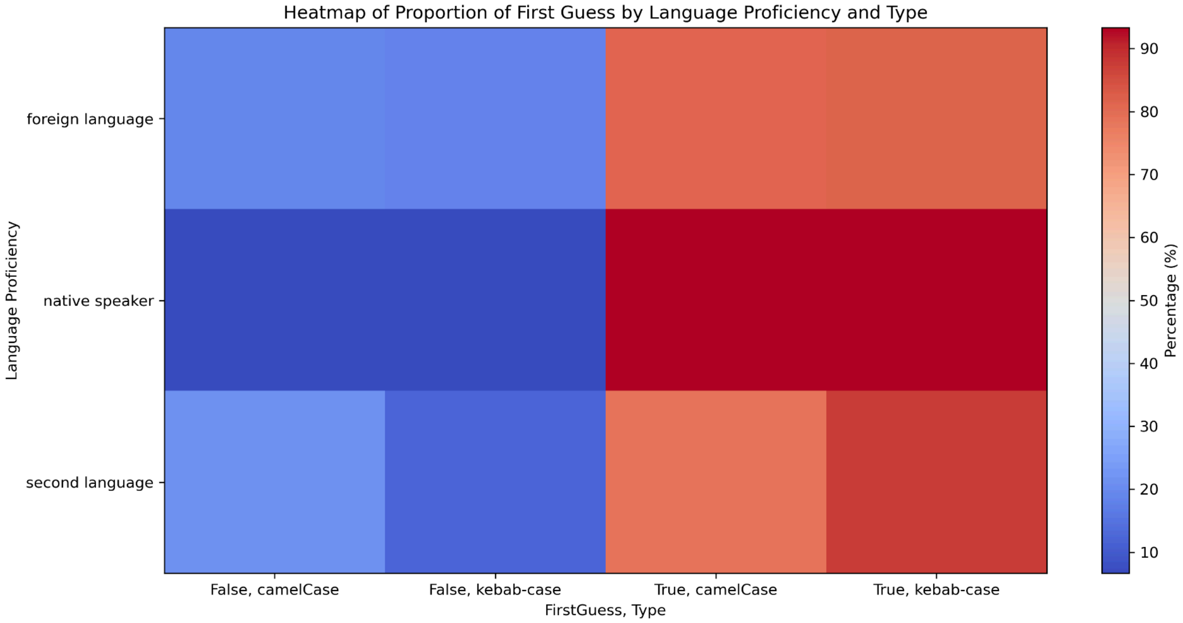


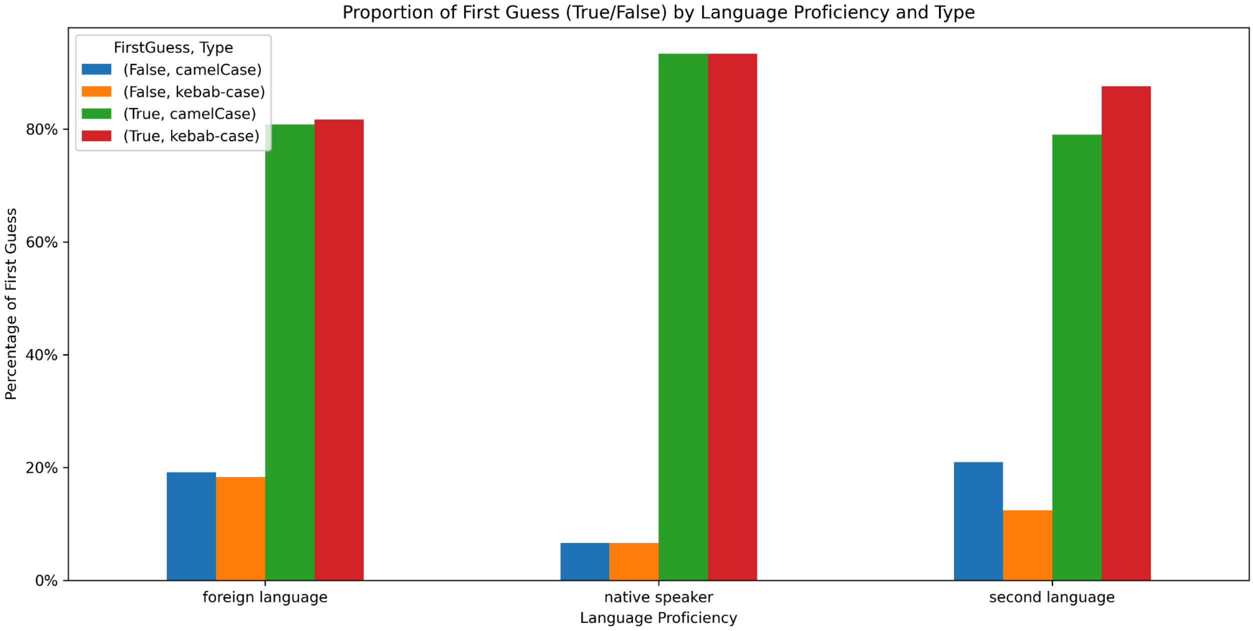
Mean time by Programming Experience





First guess by English Proficiency





3.2 Descriptive Statistics

The **five-number summary** is a statistical summary used to describe the distribution of the dataset. Here's an explanation of each part of the five-number summary:

- ➔ **Minimum**: The smallest value in the dataset. It represents the lower boundary of the data, giving the lowest point observed in the data for that group or condition.
- ➔ **First quartile**: Also known as the 25th percentile. This is the value below which 25% of the data falls. It represents the point at which a quarter of the data lies below it, giving a sense of the lower quarter of the data.
- ➔ **Median**: The middle value of the dataset when it's ordered from smallest to largest (or the average of the two middle values if there's an even number of observations). The median, or the 50th percentile, divides the dataset in half, with half of the data values below it and half above.
- ➔ **Third quartile**: Also known as the 75th percentile. This is the value below which 75% of the data falls. It represents the upper boundary for the lower 75% of the data, marking where the top 25% of the data begins.
- ➔ **Maximum**: The largest value in the dataset. It represents the highest point observed in the data for that group or condition.

Mean and standard deviation:

- ➔ **Mean** tells you the "average performance" of a group or condition. For example, if you calculate the mean time it took participants to complete tasks using camelCase vs. kebab-case, it represents the overall average completion time for each condition.
- ➔ S**tandard deviation** tells you how consistent the participants' performance was. A small SD indicates that most participants performed similarly, while a large SD means there were significant differences among participants.

Programming Experience

Programming experience	Minimum	Q1	Median	Q3	Maximum	Mean	SD
less than one year	50.0	302.50	683.5	1065.25	4878.0	769.11	635.34

more than three years	91.0	410.75	569.0	846.25	3888.0	698.32	456.44
one to three years	249.0	554.50	716.0	961.75	3330.0	817.91	420.93

Language Proficiency

Language Proficiency	Minimum	Q1	Median	Q3	Maximum	Mean	SD
foreign	50.0	379.00	571.0	815.25	2229.0	638.88	372.85
native	253.0	520.25	767.5	1167.75	4878.0	974.95	693.02
second	91.0	448.25	659.5	889.75	2276.0	711.61	366.47

Gender

Gender	Minimum	Q1	Median	Q3	Maximum	Mean	SD
female	50.0	371.00	577.5	781.25	4878.0	659.70	515.49
male	91.0	503.75	699.5	1002.00	3888.0	796.09	469.21

3.3 Inferential Statistics

The statistics provide a quantitative summary of the group's performance in terms of mean completion times.

Programming experience	Mean	**Language Proficiency**	Mean	**Gender**	Mean
less than one year	769.11	foreign	638.88	female	659.70

more than three years	698.32	native	974.95	male	796.09
one to three years	817.91	second	711.61		

The mean completion time for participants with "less than one year" of experience is **769.11 ms**, while for those with "more than three years," it is **698.32 ms**, and for participants with "one to three years," it is **817.91 ms**. This means participants with more programming experience tend to perform faster on average, with the "more than three years" group having the lowest mean completion time.

The mean completion time for participants classified as "foreign language" is **638.88 ms**, while for "native speakers," it is **974.95 ms**, and for "second language" speakers, it is **711.61 ms**. This means that participants with English as a foreign language might perform tasks more efficiently than second-language speakers and even native speakers.

Female participants have a mean completion time of **659.70 ms**, whereas male participants have an average of **796.09 ms**. This indicates that female participants, on average, completed tasks faster than their male counterparts.

4. Discussion

4.1 Compare Hypothesis to Results

Hypothesis 1 results do not support this hypothesis. Across all analyzed data, identifiers written in kebab-case consistently outperformed camelCase. **Participants identified kebab-case identifiers faster and more accurately** on their first attempt compared to camelCase identifiers. This trend was observed across all types of programming experience and language proficiency levels. These findings suggest that contrary to the initial hypothesis, the identifier style favoring kebab-case leads to better performance in terms of both speed and accuracy.

Hypothesis 2 results do not support this hypothesis. While participants with more programming experience performed better overall, as evidenced by their faster identification times compared to those with less experience, the data clearly shows that **participants with more programming experience identified kebab-cases faster than camelCase**.

This trend is evident in the graphs above, where participants with "more than three years" of programming experience consistently showed lower mean times for identifying kebab-case identifiers compared to camelCase. This indicates that even for experienced participants, kebab-case provides a significant advantage in terms of speed and accuracy.

Hypothesis 3 results support this hypothesis. **Participants with higher English proficiency did indeed identify identifiers in both camelCase and kebab-case faster** than those with lower English proficiency, as reflected in their percentage of first guesses. However, there was no significant difference in performance between camelCase and kebab-case identifiers for participants with higher English proficiency. Participants identified both styles with similar speed and accuracy, as demonstrated in the graphs provided in the report.

4.2 Limitations and Threats to Validity

A key **limitation** of this study is the use of isolated words as test material instead of real code, which makes it less realistic. Developers typically see identifiers in the context of full code, not as standalone words. Future studies could use real code snippets to better reflect real-world scenarios.

External Threats: The main external threat is the limited diversity of participants. Most participants had similar experience levels, so the results may not apply to developers with different skills or backgrounds. A more diverse participant pool could improve the generalizability of the findings.

Internal Threats: Internal threats include task order and learning effects, as participants might have improved over time, affecting the results. Another threat is the use of isolated words, which does not fully capture how developers recognize identifiers in actual coding. Future studies could randomize task orders and use real code snippets to address these issues.

4.3 Conclusions

1) **Identifier Style Performance**:
    - The percentage of correct first choices for **kebab-case** identifiers was significantly higher than for camelCase identifiers. This suggests that participants found kebab-case easier to identify accurately on the first attempt.

2) **Programming Experience**:
    - Participants with **more than three years** of programming experience had the lowest mean time for identifying correct collocations in both kebab-case and camelCase, as expected.
    - Interestingly, participants with **less than one year** of programming experience performed better (lower mean time) than those with **one to three years of programming experience**.

3) **Language Proficiency**:
   - **Native speakers** demonstrated the best first-choice accuracy for both kebab-case and camelCase identifiers, followed by participants for whom English is a **foreign language**.
   - Participants with English as a **second language** showed the lowest first-choice accuracy overall. However, within this group, the percentage of first-choice correctness for kebab-case was notably higher than for camelCase, indicating a preference or advantage for kebab-case among this demographic.

As a brief conclusion, **kebab-case** is more visible and easier to identify across a diverse range of participants. This was observed in terms of both accuracy and speed, with higher percentages of correct first choices compared to camelCase.

Appendix

A. Materials

Any documents you used for your informed consent (information sheets, consent) or as part of your apparatus (e.g., manual, hand-out), please include them here.

B. Reproduction Package (or: Raw Data)

Here is a link to github where you can find the project reproduction package:
https://github.com/NikBigBoss1

Also you need to change the connection of the database to the localhost and dump db with the collocations given it the assets/data/ directory.