

Case Styles: Camel, Pascal, Snake, and Kebab Case

The most popular ways to combine words into a single string



Patrick Divine

[Follow](#)

Nov 4, 2018 · 3 min read ★

Photo by Oskar Yildiz on Unsplash

TLDR;

- camelCase
- PascalCase
- snake_case
- kebab-case

Removing spaces between words

In programming, we often remove the spaces between words because programs of different sorts reserve the space (‘ ’) character for special purposes. Because the space character is reserved, we cannot use it to represent a concept that we express in our human language with multiple words.

As an example, the concept of *user login count* is not referenced in our code as `user login count` often. We do not do the following:

```
user login count = 5
```

A typical language parse would treat each word as a separate concept. User, login, and count would each be treated as separate things. So, we do something like the following:

```
userLoginCount = 5
```

Now, the parser will see one concept, `userLoginCount`, and us programmers can easily see the representation.

The best way to combine words

There is no best way to combine words. In the above example, we removed spaces and capitalized each word following the first word. There are, however, a great number of algorithms for combining words, and a few very common ones.

The commonly used strategies for combining words are: camel case, pascal case, snake case, and kebab case. We'll go over those here.

Camel Case (camelCase)

"three camels standing on street" by Lombe Kabaso on Unsplash

Camel case combines words by capitalizing all words following the first word and removing the space, as follows:

Raw: `user login count`

Camel Case: `userLoginCount`

This is a very popular way to combine words to form a single concept. It is often used as a convention in variable declaration in many languages.

Pascal Case (PascalCase)

PHILOSOPHY - Blaise Pascal



Pascal case combines words by capitalizing all words (even the first word) and removing the space, as follows:

Raw: `user login count`

Pascal Case: `UserLoginCount`

This is also a very popular way to combine words to form a single concept. It is often used as a convention in declaring classes in many languages.

Snake Case (`snake_case`)

"brown snake" by David Clode on Unsplash

Snake case combines words by replacing each space with an underscore (`_`) and, in the all caps version, all letters are capitalized, as follows:

Raw: `user login count`

Snake Case: `user_login_count`

Snake Case (All Caps): `USER_LOGIN_COUNT`

This style, when capitalized, is often used as a convention in declaring constants in many languages. When lower cased, it is used conventionally in declaring database field names.

Kebab Case (kebab-case)

"barbecue on brown board" by pan xiaozhen on Unsplash

Kebab case combines words by replacing each space with a dash (-), as follows:

Raw: `user login count`

Kebab Case: `user-login-count`

This style is often used in URLs. For example, *www.blog.com/cool-article-1*. It is a nice, clean, human-readable way to combine the words.

Which is best?

There is no best method of combining words. The main thing is to be consistent with the convention used, and, if you're in a team, to come to an agreement on the convention together.

`thanksForReading!`

`ThanksForReading!`

`THANKS_FOR_READING_!`

`thanks-for-reading-!`

Cheers!

Programming

Camel Case

Pascal Case

Snake Case

Kebab Case