

# Are dynamic types slower in Dart?

Asked 3 years, 3 months ago   Active 2 years, 7 months ago   Viewed 722 times



I have been wondering if dynamic types are slower in Dart.

6

Example given:



```
final dynamic example = "Example"
```



versus

```
final String example = "Example"
```

variables

dynamic

types

dart

Share Follow

edited May 17 '19 at 13:12

asked Sep 13 '18 at 23:29



TheUniqueProgrammer

435 ● 2 ● 4 ● 12

The first example is not a dynamic call – [Jonah Williams](#) Sep 14 '18 at 2:29

## 2 Answers

Active

Oldest

Votes



Yes, using `dynamic` typed variables in Dart is often slower than using variables typed with an actual type.

12



However, your example is not using `dynamic` as type, it is using type inference to infer the `String` type. That might cost a *little* extra at compile-time, but at run-time, your two code examples are completely identical. Both variables are typed as `String`.



A `dynamic` method invocation may be slower because the run-time system must add extra checks to ensure that the variable can do the things you are trying to do with it. If you have `int x = 2; print(x + 3);` the run-time system knows that `int` has a `+` operator, and even knows what it is. If you write `dynamic x = 2; print(x + 3);`, the run-time system must first check whether `x` has a `+` operator before it can call it, and find that operator's definition on the object before calling it. It might not always be slower, some cases optimize better than others, but it can never be faster.

**Join Stack Overflow** to learn, share knowledge, and build your career.

Sign up



`(stringOrList as List).length`. It may be slower depending on the compiler and the target platform.

Share Follow

edited Dec 3 '18 at 9:50

answered Sep 14 '18 at 6:01



Irn

45.7k ● 4 ● 71 ● 85



2



Well, in your first example (heh), `example` is inferred to be a type `String`, not `dynamic`, so how could it be slower? The style guide even recommends not adding redundant types to those variables that can be inferred correctly.

Share Follow

answered Sep 14 '18 at 1:42



Randal Schwartz

32.7k ● 4 ● 37 ● 60

What about leaving about return types from method signatures such as `Future get(resource)` vs. `Future<Response> get(resource)`? – [mLstudent33](#) Nov 22 '21 at 7:06