

Typedefs

A type alias—often called a *typedef* because it's declared with the keyword `typedef`—is a concise way to refer to a type. Here's an example of declaring and using a type alias named `IntList`:

```
typedef IntList = List<int>;
IntList il = [1, 2, 3];
```

dart

A type alias can have type parameters:

```
typedef ListMapper<X> = Map<X, List<X>>;
Map<String, List<String>> m1 = {}; // Verbose.
ListMapper<String> m2 = {}; // Same thing but shorter and clearer.
```

⬆ 版本提示

Before 2.13, typedefs were restricted to function types. Using the new typedefs requires a [language version](#) of at least 2.13.

We recommend using [inline function types](#) instead of typedefs for functions, in most situations. However, function typedefs can still be useful:

```
typedef Compare<T> = int Function(T a, T b);

int sort(int a, int b) => a - b;

void main() {
  assert(sort is Compare<int>); // True!
}
```

dart