〈 Comments

Libraries 〉

Metadata

Use metadata to give additional information about your code. A metadata annotation begins with the character `@`, followed by either a reference to a compile-time constant (such as `deprecated`) or a call to a constant constructor.

Four annotations are available to all Dart code: `@Deprecated`, `@deprecated`, `@override`, and `@pragma`. For examples of using `@override`, see Extending a class.

Here's an example of using the `@Deprecated` annotation:

```
class Television {
  /// Use [turnOn] to turn the power on instead.
  @Deprecated('Use turnOn instead')
  void activate() {
    turnOn();
  }

  /// Turns the TV's power on.
  void turnOn() {...}
  // ···
}
```

You can use `@deprecated` if you don't want to specify a message. However, we recommend always specifying a message with `@Deprecated`.

You can define your own metadata annotations. Here's an example of defining a `@Todo` annotation that takes two arguments:

```
class Todo {
  final String who;
  final String what;

  const Todo(this.who, this.what);
}
```

And here's an example of using that `@Todo` annotation:

```
@Todo('Dash', 'Implement this function')
void doSomething() {
  print('Do something');
}
```

Metadata can appear before a library, class, typedef, type parameter, constructor, factory, function, field, parameter, or variable declaration and before an import or export directive.

〈 Comments

Libraries 〉