(continued from previous page)

```
    'dodo': Bird(
        'dodo',
        eats=['fruit'],
        fly=False,
        extinct=True
    )
}
```

You can add a `__rich_repr__` method to any class to enable the Rich formatting. This method should return an iterable of tuples. You could return a list of tuples, but it's easier to express with the `yield` keywords, making it a *generator*.

Each tuple specifies an element in the output.

- `yield value` will generate a positional argument.
- `yield name, value` will generate a keyword argument.
- `yield name, value, default` will generate a keyword argument *if* `value` is not equal to `default`.

If you use `None` as the `name`, then it will be treated as a positional argument as well, in order to support having `tuple` positional arguments.

You can also tell Rich to generate the *angular bracket* style of repr, which tend to be used where there is no easy way to recreate the object's constructor. To do this set the function attribute `"angular"` to `True` immediately after your `__rich_repr__` method. For example:

```
__rich_repr__.angular = True
```

This will change the output of the Rich repr example to the following:

```
{
    'gull': <Bird 'gull' eats=['fish', 'chips', 'ice cream', 'sausage rolls']>,
    'penguin': <Bird 'penguin' eats=['fish'] fly=False>,
    'dodo': <Bird 'dodo' eats=['fruit'] fly=False extinct=True>
}
```

Note that you can add `__rich_repr__` methods to third-party libraries *without* including Rich as a dependency. If Rich is not installed, then nothing will break. Hopefully more third-party libraries will adopt Rich repr methods in the future.

7.3.1 Typing

If you want to type the Rich repr method you can import and return `rich.repr.Result`, which will help catch logical errors:

```
import rich.repr


class Bird:
    def __init__(self, name, eats=None, fly=True, extinct=False):
        self.name = name
        self.eats = list(eats) if eats else []
        self.fly = fly
        self.extinct = extinct

```

(continues on next page)