

(continued from previous page)

```
def __rich_repr__(self) -> rich.repr.Result:
    yield self.name
    yield "eats", self.eats
    yield "fly", self.fly, True
    yield "extinct", self.extinct, False
```

### 7.3.2 Automatic Rich Repr

Rich can generate a rich repr automatically if the parameters are named the same as your attributes.

To automatically build a rich repr, use the `auto()` class decorator. The Bird example above follows the above rule, so we don't strictly need to implement our own `__rich_repr__`. The following code would generate the same repr:

```
import rich.repr

@rich.repr.auto
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

BIRDS = {
    "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)
}

from rich import print
print(BIRDS)
```

Note that the decorator will also create a `__repr__`, so you will get an auto-generated repr even if you don't print with Rich.

If you want to auto-generate the angular type of repr, then set `angular=True` on the decorator:

```
@rich.repr.auto(angular=True)
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
```

## 7.4 Example

See [repr.py](#) for the example code used in this page.