Cheatsheet: Creating mocks in PHPUnit

Available in test cases

Name of the class to mock (= to subclass)

```
$mock = $this->getMock(
```

names of the methods to mock (empty = all, null = none)

all parameters but the first can be omitted

```
'Tx_Coffee_Cup',
['fill', 'stir'],
[$size, $color],
'MockedCup',
false
);
```

parameters for the constructor

name of the created class (empty = auto)

Why mock a method?
▷ To "disable" a method (to not write to the DB, or to not launch a cruise missile) and return null.
▷ To have the method return a particular return value.
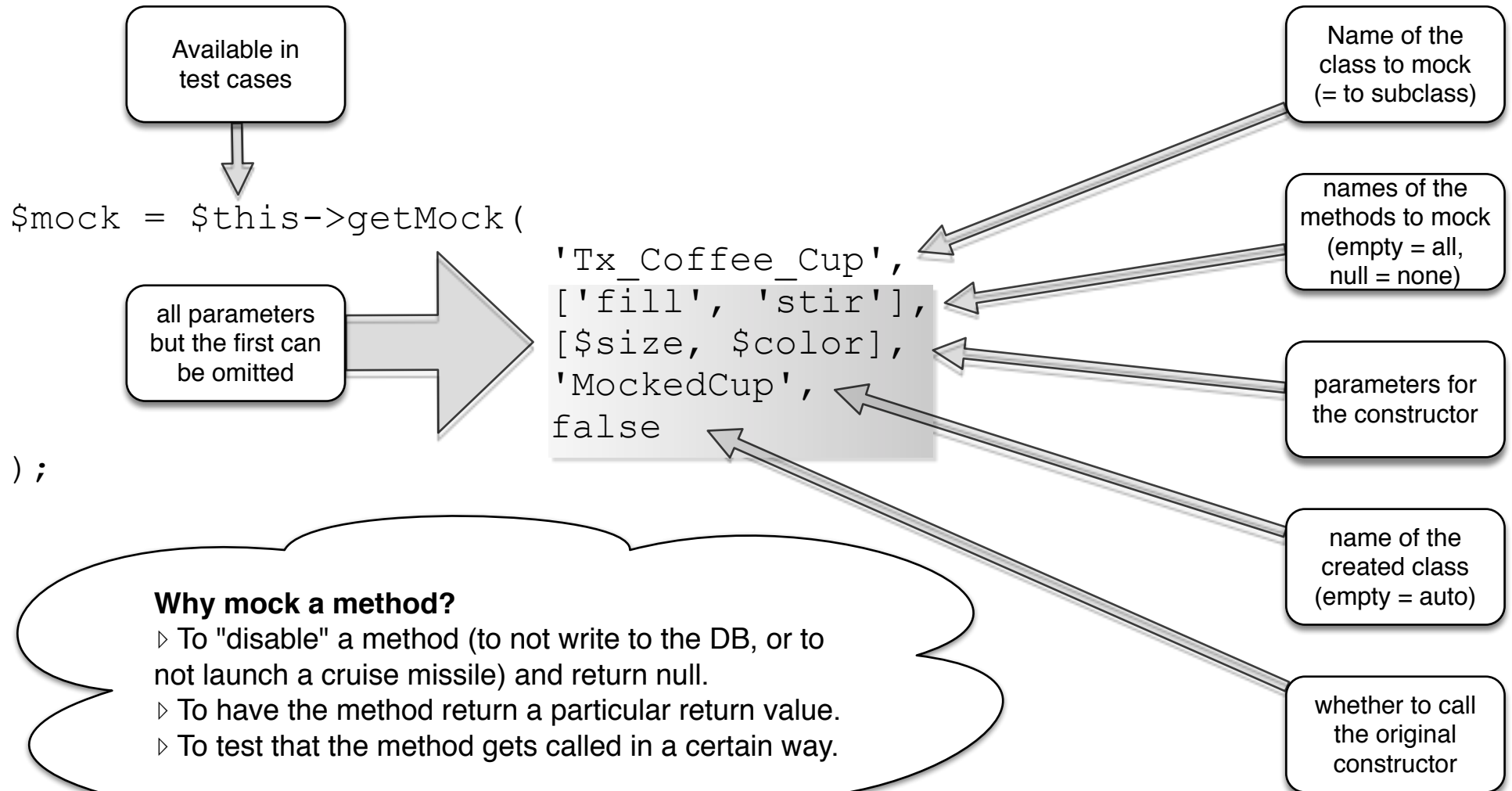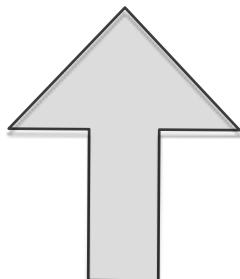▷ To test that the method gets called in a certain way.

whether to call the original constructor

Cheatsheet: Using mocks in PHPUnit

```php
$mock->expects(self::once())
    ->method('fill')
    ->with($coffee, 200)
    ->will(
        self::returnValue(true)
    );
```

how often:
never()
once()
any()
atLeastOnce()
atLeast(3)
exactly(4)

or which call:
at(0) for the first call
at(2) for the third call

name of the method, *must* be a mocked method

required parameters or self::anything(), omit for never()

return value

any of the last two parameters can be omitted