

DEAL

writing bug-free Python code

a journey from tests to formal verification

or how to make a title so long that nobody cares to read it

orsinium.dev



dephell
flakehell
textdistance

```
def cat(left, right):  
    return left + right
```

```
def test_cat():  
    result = cat('abc', 'def')  
    assert result == 'abcdef'
```

```
@pytest.mark.parametrize('left, right, exp', [
    ('a', 'b', 'ab'),
    ('', '', ''),
    ('', 'b', 'b'),
    ('a', '', 'a'),
    ('text', 'check', 'textcheck'),
])
```

```
def test_cat(left, right, exp):
    result = cat(left, right)
    assert result == expected
```

```
@pytest.mark.parametrize('left, right', [
    ('a', 'b'),
    ('', ''),
    ('', 'b'),
    ('a', ''),
    ('text', 'check'),
])

def test_cat(left, right):
    result = cat(left, right)
    assert result.startswith(left)
    assert result.endswith(right)
    assert len(result) == len(left) + len(right)
```

```
import hypothesis
from hypothesis import strategies as s

@hypothesis.given(left=s.text(), right=s.text())
def test_cat(left, right):
    result = cat(left, right)
    assert result.startswith(left)
    assert result.endswith(right)
    assert len(result) == len(left) + len(right)
```

```
def cat(left: str, right: str) -> str:  
    return left + right
```



```
import hypothesis_auto

@hypothesis_auto.auto_pytest(cat)
def test_cat(test_case):
    result = test_case()
    left = test_case.parameters.kwargs['left']
    right = test_case.parameters.kwargs['right']
    assert result.startswith(left)
    assert result.endswith(right)
    assert len(result) == len(left) + len(right)
```

```
import deal
```

```
@deal.ensure(lambda _: _.result.startswith(_.left))
```

```
@deal.ensure(lambda _: _.result.endswith(_.right))
```

```
@deal.ensure(lambda _: len(_.result) == len(_.left) + len(_.right))
```

```
def cat(left: str, right: str) -> str:
```

```
    return left + right
```

```
test_cat = deal.cases(cat)
```

```
for case in deal.cases(cat):  
    case()
```

```
@deal.cases(cat)  
def test_div(case):  
    case()
```

```
test = deal.cases(div)  
atheris.Setup([], test)  
atheris.Fuzz()
```

```
> pygmentize example.py  
import deal
```

```
@deal.post(lambda result: 0 <= result <= 1)  
def sin(x):  
    return 2
```

```
sin(2)
```

```
> python3.9 example.py
```

```
Traceback (most recent call last):
```

```
  File "/home/gram/Documents/deal/example.py", line 7, in <module>  
    sin(2)
```

```
deal.PostContractError: expected 0 <= result <= 1 (where result=2)
```

```
> flake8 --show-source example.py --select DEAL
example.py:5:12: DEAL012 post contract error (2)
    return 2
           ^
```

```
> python3.9 -m deal lint example.py
example.py
5:11 DEAL012 post contract error (2)
    return 2
           ^
```

1931 Gödel's incompleteness theorems
1936 Halting problem
1949 The first proof of program correctness
1967 "Assigning Meanings to Programs"
1969 Hoare logic
1986 Eiffel, design by contract
2009 Dafny
2015 Z3
2018 deal
2019 Deal gets a linter
2021 deal-solver

```
@deal.ensure(lambda left, right, result: result.startswith(left))
@deal.ensure(lambda left, right, result: result.endswith(right))
def cat(left: str, right: str) -> str:
    if left + right == 'hello world':
        return left
    return left + right
```

```
> python3.9 -m deal prove example.py --skipped
```

```
example.py
```

```
cat
```

```
failed post-condition. Example: left='hello w', right='orld'.
```




CrossHair

chat

on gitter



Check

failing

downloads

46k

An analysis tool for Python that blurs the line between testing and type systems.

If you have a function with [type annotations](#) and add a contract [in a supported syntax](#), CrossHair will attempt to find counterexamples for you:

```
> crosshair check example.py
```

```
/home/gram/Documents/deal/example.py:4: error: false when calling cat('h', 'ello world') (which returns 'h')
```

```
@deal.pre(lambda _: _.right != 0)
def div(left, right):
    return left / right
```

```
@deal.post(lambda r: 0 <= r <= 1)
def sin(x):
    return math.sin(x)
```

```
@deal.inv(lambda obj: obj.likes >= 0)
class Video:
    likes = 1
    def like(self): self.likes += 1
    def dislike(self): self.likes -= 1
```

```
@deal.raises(ZeroDivisionError)
@deal.reason(
    ZeroDivisionError,
    lambda _: _.right == 0,
)
def div(left, right):
    return left / right
```

```
@deal.has( 'stdout' )  
def example():  
    print( 'oh hi' )
```

```
@deal.example(  
    lambda: double(3) == 6  
)  
def double(x):  
    return x * 2
```

```
import deal
import something_else

deal.module_load(deal.pure)

print(1)
```

```
python3 -m deal memtest
```

```
python3 -m deal decorate
```


github.com/life4/deal

