Making your Python code write your Python code

Marcin Sobczyk

FOSDEM'19

Who am I?

Marcin Sobczyk

[Martseen Sopcheek]

Marcin Sobczyk

[Martseen Sopcheek]

msobczyk@redhat.com

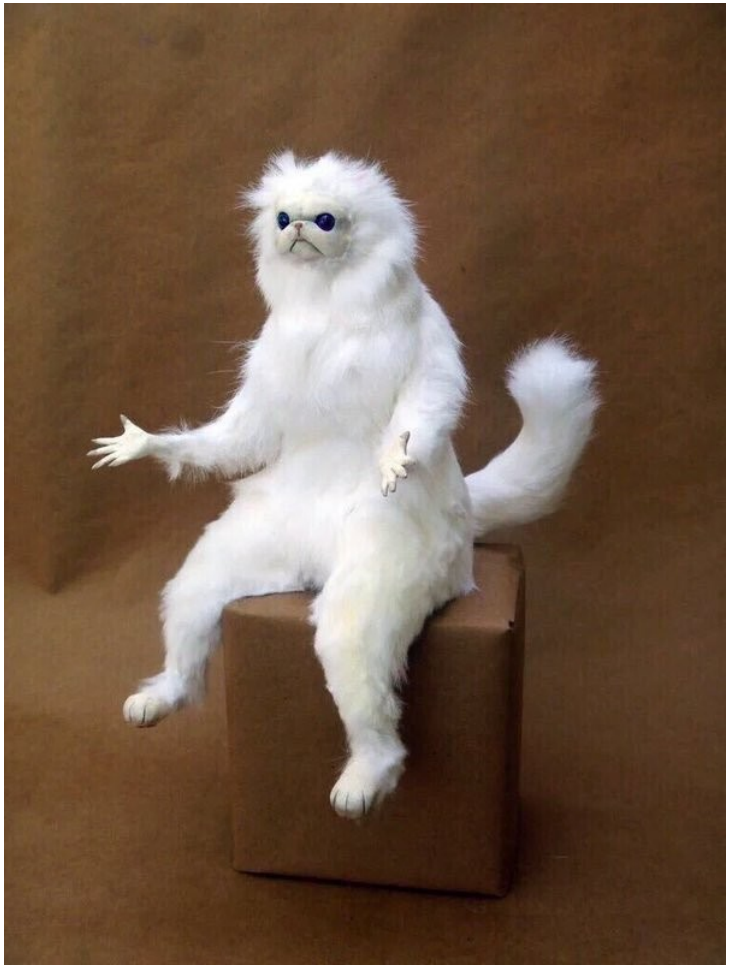

redhat


oVirt

Who is this talk for?

Abstract Syntax Trees

Why?



pytest

test_foobar.py:

```
from collections import namedtuple

FooBar = namedtuple(“FooBar”, “foo bar”)

def test_foobar():
    assert FooBar(1, 2) == FooBar(1, 3)
```

```
$ pytest test_foobar.py

> assert FooBar(1, 2) == FooBar(1, 3)
E assert FooBar(foo=1, bar=2) == FooBar(foo=1, bar=3)
E         At index 1 diff: 2 != 3
E         Use -v to get the full diff
```

- Analyze Python code

- Analyze Python code
- Transform Python code

- Analyze Python code
- Transform Python code
- Generate and execute bytecode

All of this… in runtime!

- Analyze Python code
- Transform Python code
- Generate and execute bytecode

benchmark.py:

```
@benchmark
def benchmark_me():
    print(“Hi, gonna sleep now...”)
    time.sleep(1)
    print(“I’m back!”)
```

```
$ ./benchmark.py

Hi, gonna sleep now...
---> Took 9.06-06 seconds to execute the LLoC
---> Took 1.00258 seconds to execute the LLoC
I’m back!
---> Took 1.33-05 seconds to execute the LLoC
```

How?

How?

- What are ASTs?

How?

- What are ASTs?
- What do they look like?

How?

- What are ASTs?
- What do they look like?
- How to create your own AST?

How?

- What are ASTs?
- What do they look like?
- How to create your own AST?
- How to make bytecode out of an AST?

How?

- What are ASTs?
- What do they look like?
- How to create your own AST?
- How to make bytecode out of an AST?
- Write the tool!

How?

How to draw an owl

1.

1. Draw some circles

2.

2. Draw the rest of the fucking owl

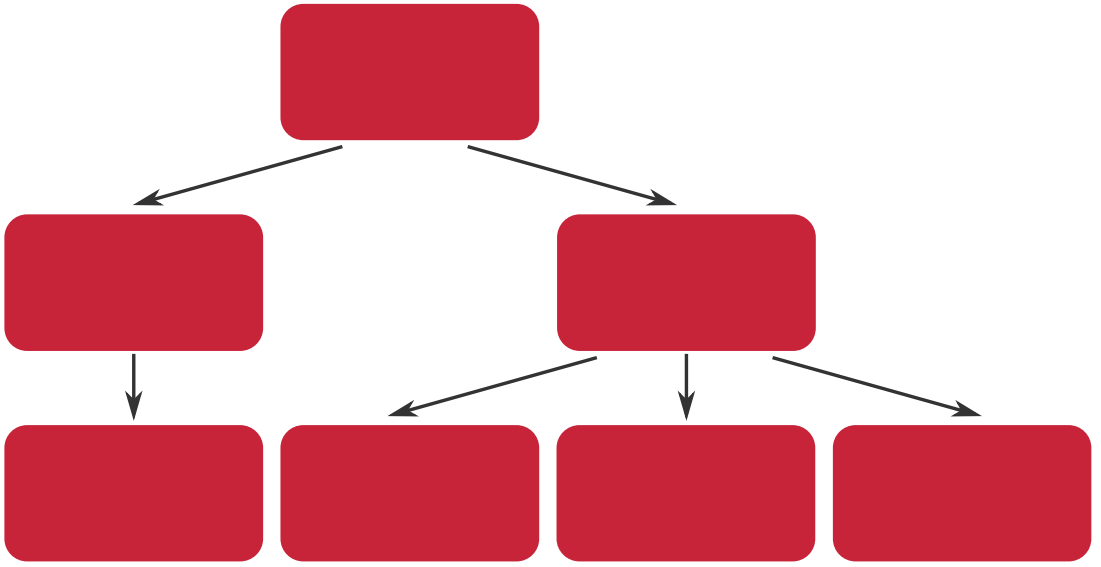
How?

- What are ASTs?
- What do they look like?
- How to create your own AST?
- How to make bytecode out of an AST?
- Write the tool!

What is an AST?

What is an AST?

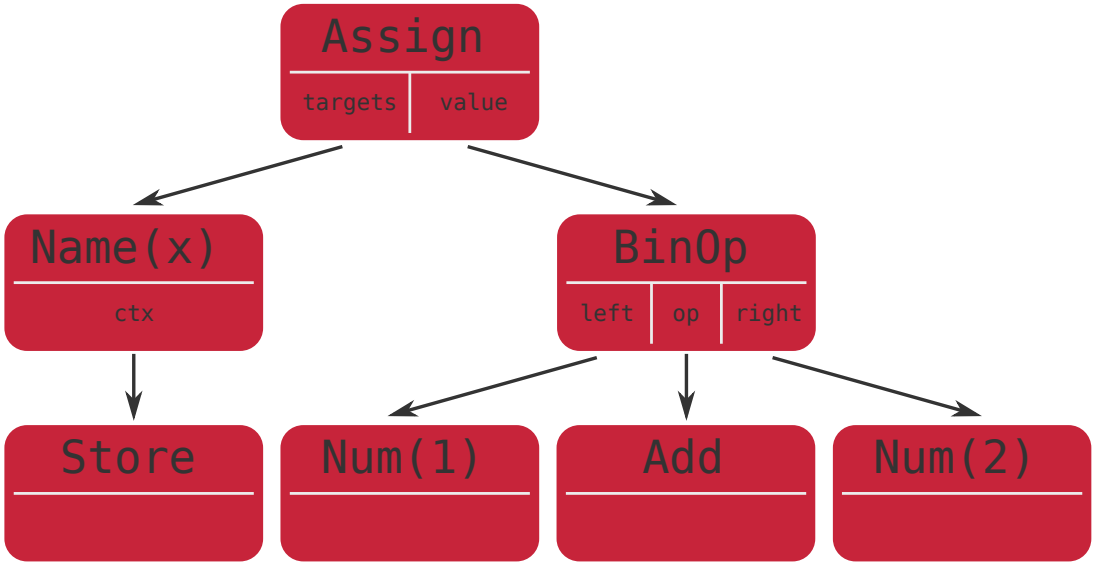
- Abstract
- Syntax
- Tree



What is an AST?

- Abstract
- Syntax
- Tree

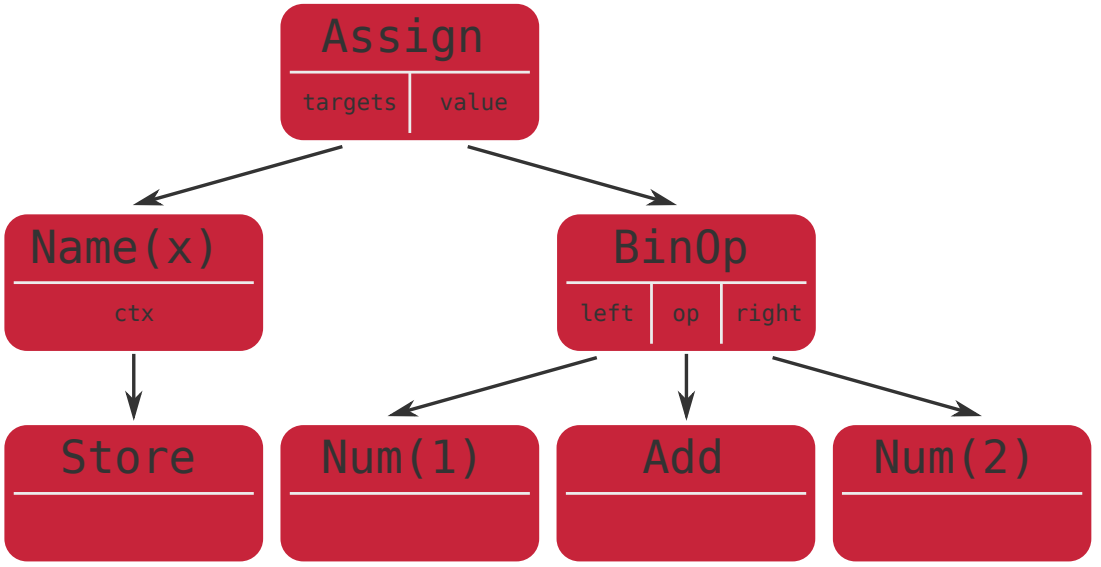
```
>>> x = 1 + 2
```



What is an AST?

- Abstract
- Syntax
- Tree

```
>>> x = 1 + 2
```



How?

- What are ASTs?
- What do they look like?
- How to create your own AST?
- How to make bytecode out of an AST?
- Write the tool!

How?

- ~~What are ASTs?~~
- What do they look like?
- How to create your own AST?
- How to make bytecode out of an AST?
- Write the tool!

How?

- ~~What are ASTs?~~
- What do they look like?
- How to create your own AST?
- How to make bytecode out of an AST?
- Write the tool!

What do they look like?

What do they look like?

- `ast.parse`
- `from astpretty import pprint as pp`

Let’s parse some code!

- `ast.parse`
- `from astpretty import pprint as pp`

Let’s parse some code!

- ~~`ast.parse`~~
- `from astpretty import pprint as pp`

Let’s analyze some ASTs!

- ~~`ast.parse`~~
- `from astpretty import pprint as pp`

```
>>> x = y + 1
```

How?

- ~~What are ASTs?~~
- What do they look like?
- How to create your own AST?
- How to make bytecode out of an AST?
- Write the tool!

How?

- ~~What are ASTs?~~
- ~~What do they look like?~~
- How to create your own AST?
- How to make bytecode out of an AST?
- Write the tool!

How?

- ~~What are ASTs?~~
- ~~What do they look like?~~
- How to create your own AST?
- How to make bytecode out of an AST?
- Write the tool!

Let’s create an AST for something like… this:

```
>>> 2 > 1
```

How?

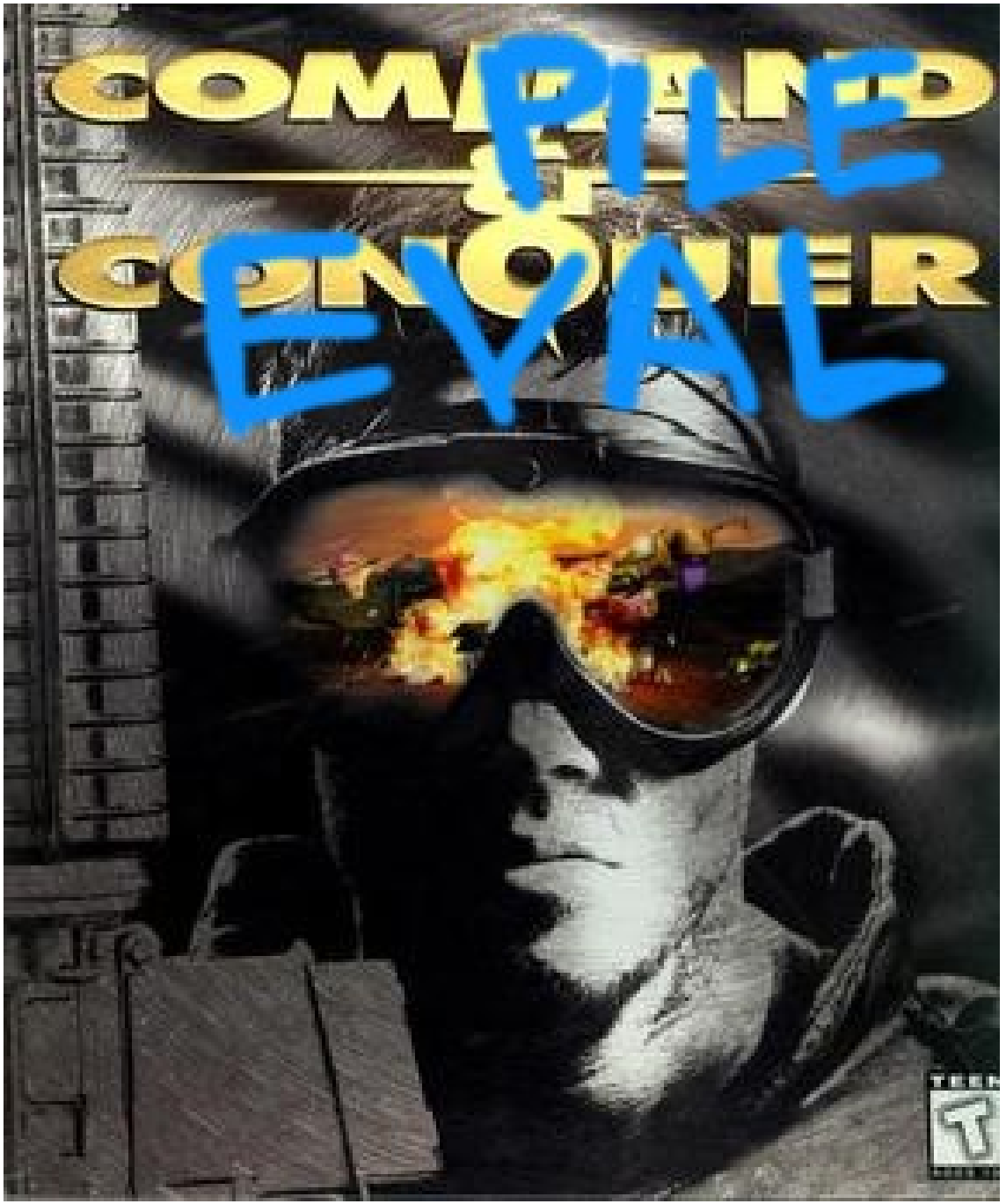
- ~~What are ASTs?~~
- ~~What do they look like?~~
- How to create your own AST?
- How to make bytecode out of an AST?
- Write the tool!

How?

- ~~What are ASTs?~~
- ~~What do they look like?~~
- ~~How to create your own AST?~~
- How to make bytecode out of an AST?
- Write the tool!

How?

- ~~What are ASTs?~~
- ~~What do they look like?~~
- ~~How to create your own AST?~~
- How to make bytecode out of an AST?
- Write the tool!


COMPILED
&
EVAL

How?

- ~~What are ASTs?~~
- ~~What do they look like?~~
- ~~How to create your own AST?~~
- How to make bytecode out of an AST?
- Write the tool!

How?

- ~~What are ASTs?~~
- ~~What do they look like?~~
- ~~How to create your own AST?~~
- ~~How to make bytecode out of an AST?~~
- Write the tool!

How?

- ~~What are ASTs?~~
- ~~What do they look like?~~
- ~~How to create your own AST?~~
- ~~How to make bytecode out of an AST?~~
- Write the tool!

But before we do...

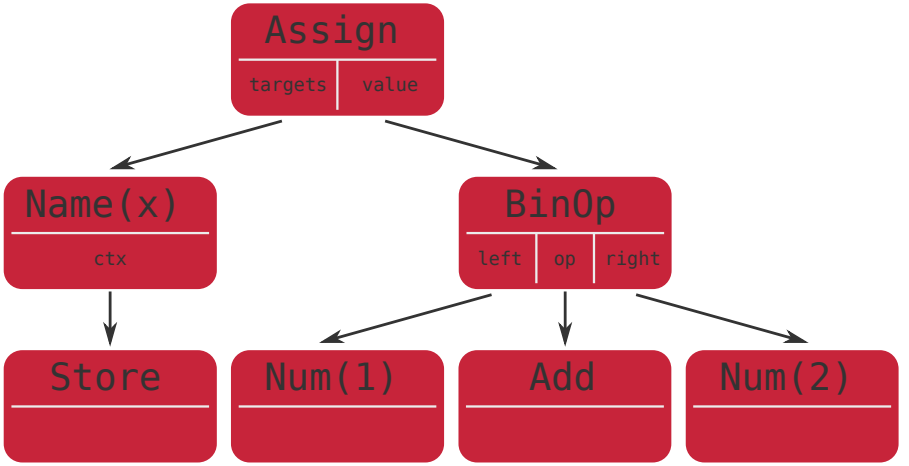
ast.NodeTransformer

ast.py:

```
class NodeTransformer:

    def visit(self, node):
        ...
```

my_transformer.py:

```
class MyTransformer(ast.NodeTransformer):

    def visit_Name(self, node):
        ...

    def visit_BinOp(self, node):
        ...
```

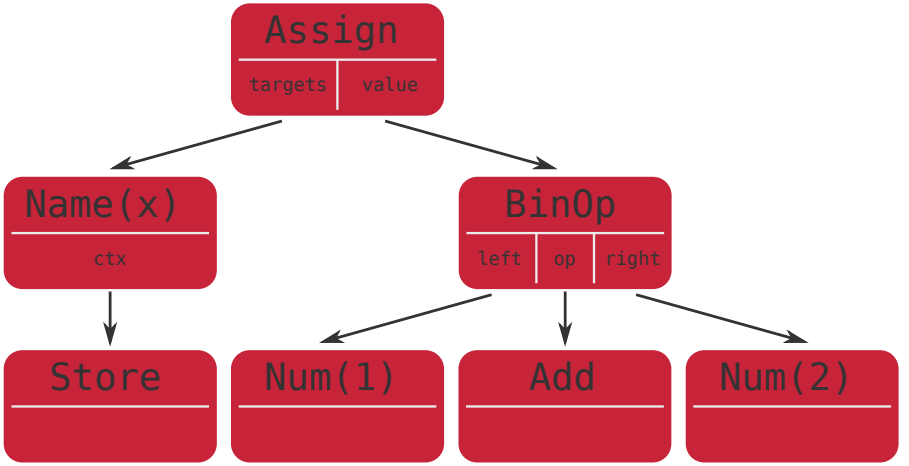


ast.py:

```
class NodeTransformer:

    def visit(self, node):
        ...
```

my_transformer.py:

```
class MyTransformer(ast.NodeTransformer):

    def visit_Name(self, node):
        ...

    def visit_BinOp(self, node):
        ...
```

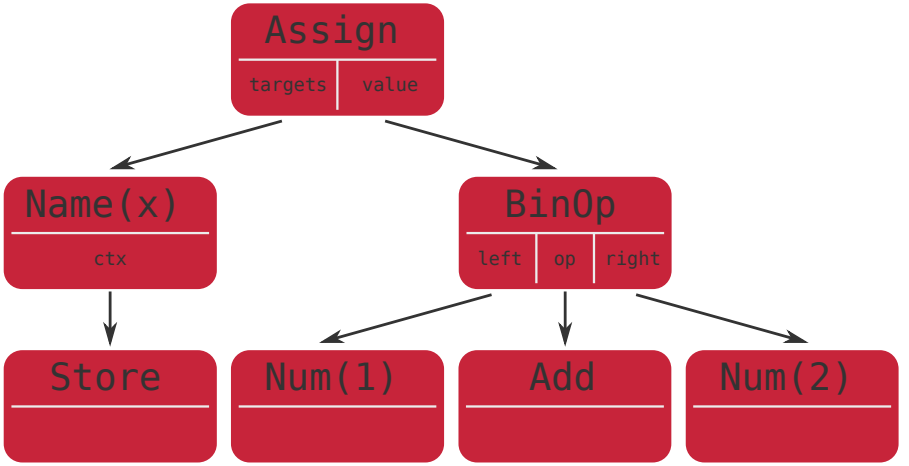


ast.py:

```
class NodeTransformer:

    def visit(self, node):
        ...
```

my_transformer.py:

```
class MyTransformer(ast.NodeTransformer):

    def visit_Name(self, node):
        ...

    def visit_BinOp(self, node):
        ...
```

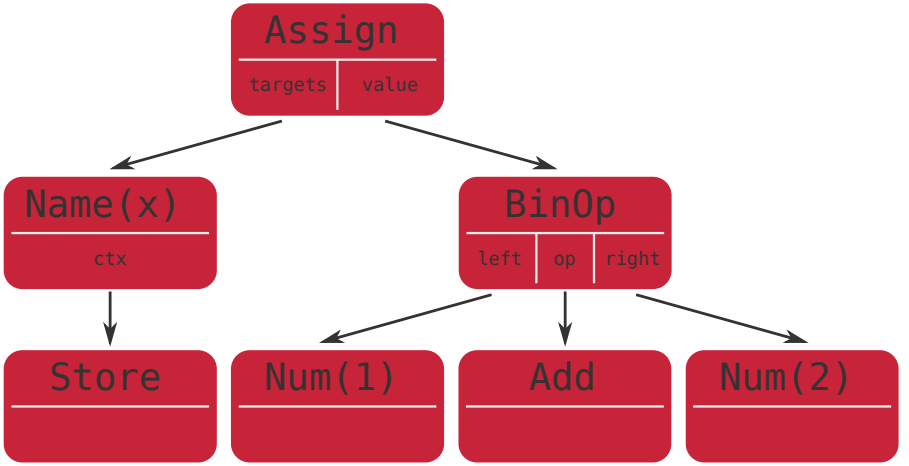


ast.py:

```
class NodeTransformer:

    def visit(self, node):
        ...
```

my_transformer.py:

```
class MyTransformer(ast.NodeTransformer):

    def visit_Name(self, node):
        ...

    def visit_BinOp(self, node):
        ...
```

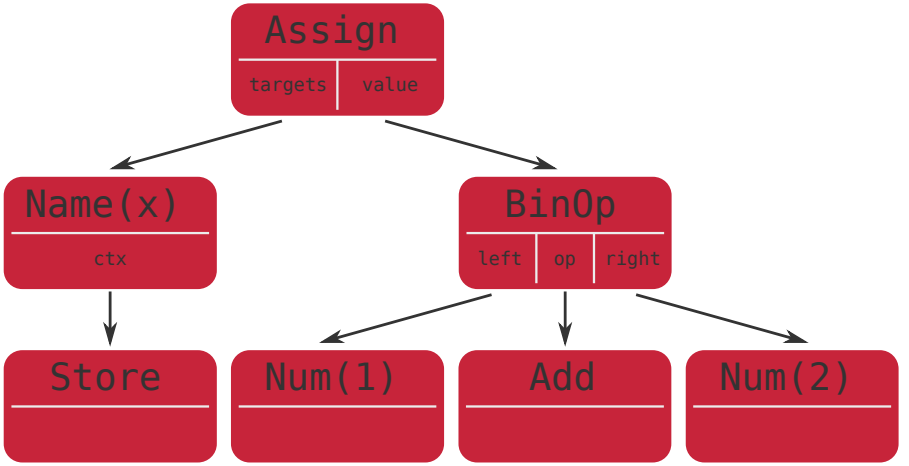


ast.py:

```
class NodeTransformer:

    def visit(self, node):
        ...
```

my_transformer.py:

```
class MyTransformer(ast.NodeTransformer):

    def visit_Name(self, node):
        ...

    def visit_BinOp(self, node):
        ...
```

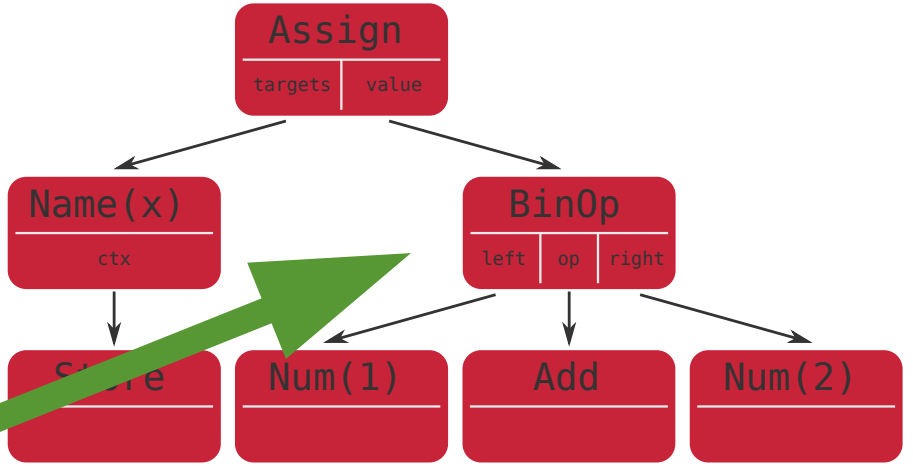


ast.py:

```
class NodeTransformer:

    def visit(self, node):
        ...
```

my_transformer.py:

```
class MyTransformer(ast.NodeTransformer):

    def visit_Name(self, node):
        ...

    def visit_BinOp(self, node):
        ...
```

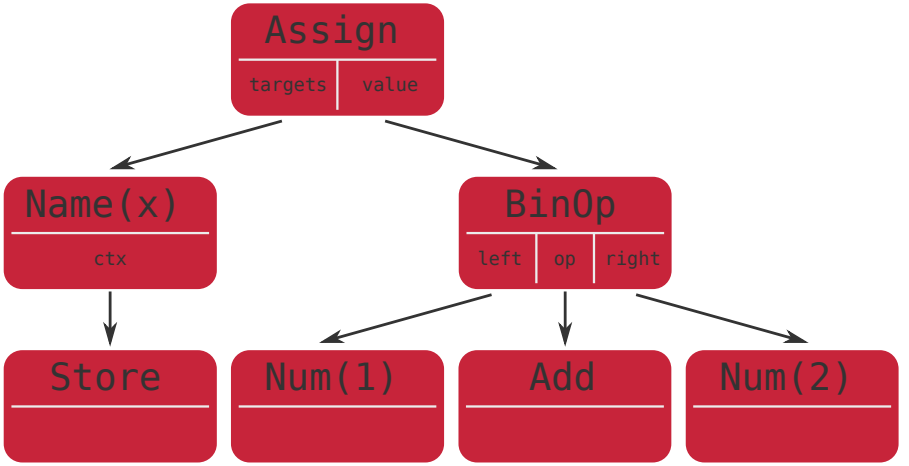


ast.py:

```
class NodeTransformer:

    def visit(self, node):
        ...
```

my_transformer.py:

```
class MyTransformer(ast.NodeTransformer):

    def visit_Name(self, node):
        return "kitten"

    def visit_BinOp(self, node):
        ...
```

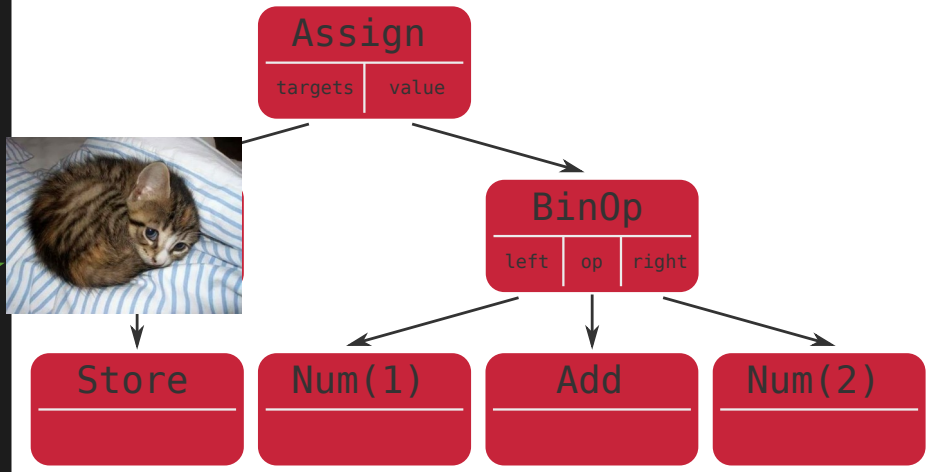


ast.py:

```
class NodeTransformer:

    def visit(self, node):
        ...
```

my_transformer.py:

```
class MyTransformer(ast.NodeTransformer):

    def visit_Name(self, node):
        return "kitten"

    def visit_BinOp(self, node):
        ...
```



Live ~~failure~~ demo time!

Thank you for your attention!

Making your Python code write your Python code

Marcin Sobczyk

FOSDEM'19