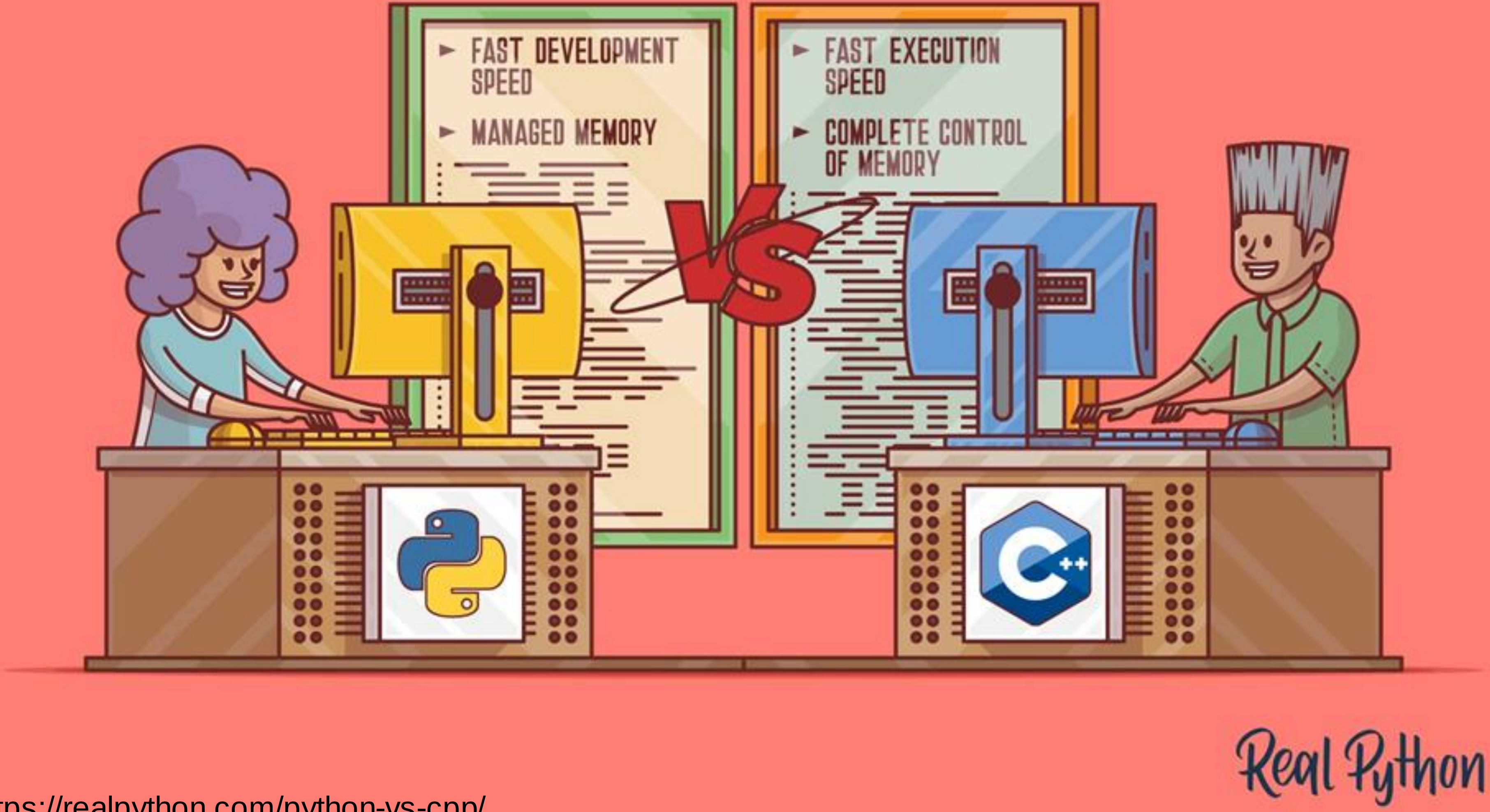
Py2Cy: A Genetic Improvement Tool To Speed Up Python

James Zhong, Max Hort, Federica Sarro



https://realpython.com/python-vs-cpp/

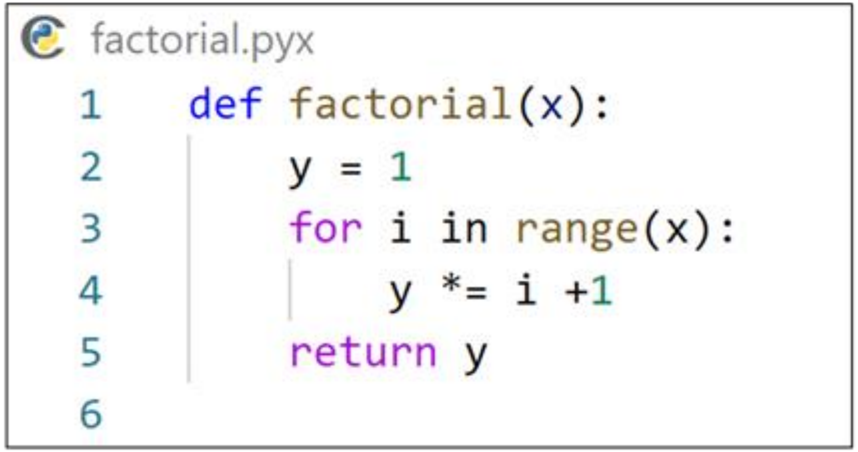
“Python extension designed to achieve **C-like runtime performance** with optional C-inspired syntax”

Cython Procedure

- Convert to .pyx file
- Add Cython constructs
- Compile .pyx file
- Test

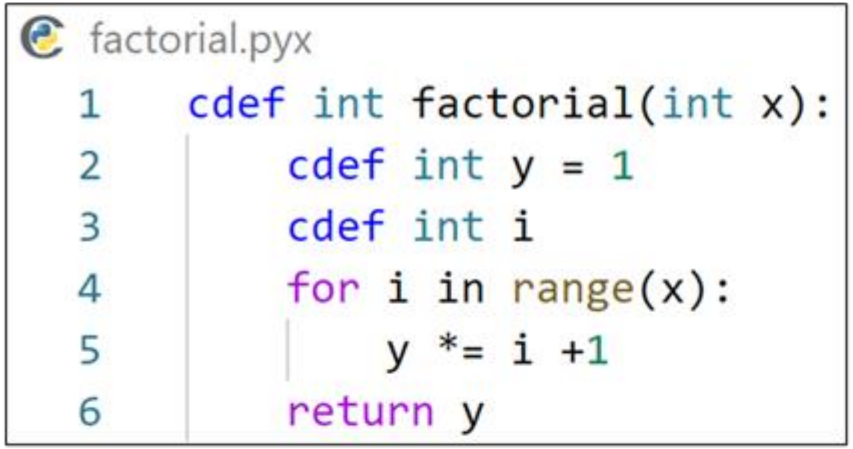
Cython Procedure

Add Cython constructs



factorial.pyx

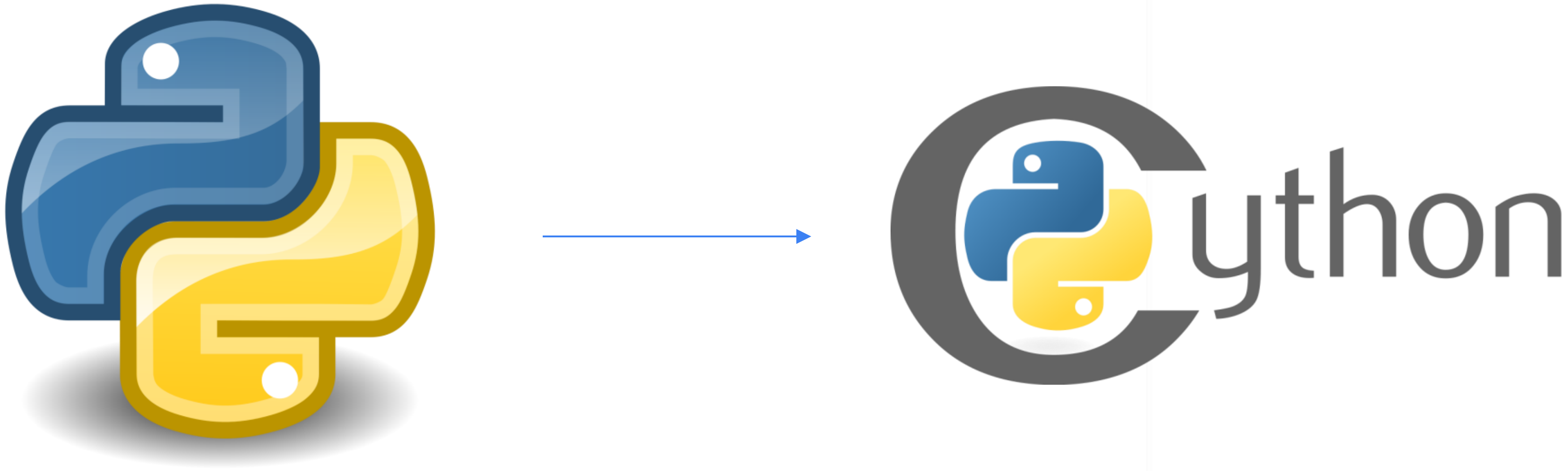
```python
def factorial(x):
    y = 1
    for i in range(x):
        y *= i +1
    return y

```



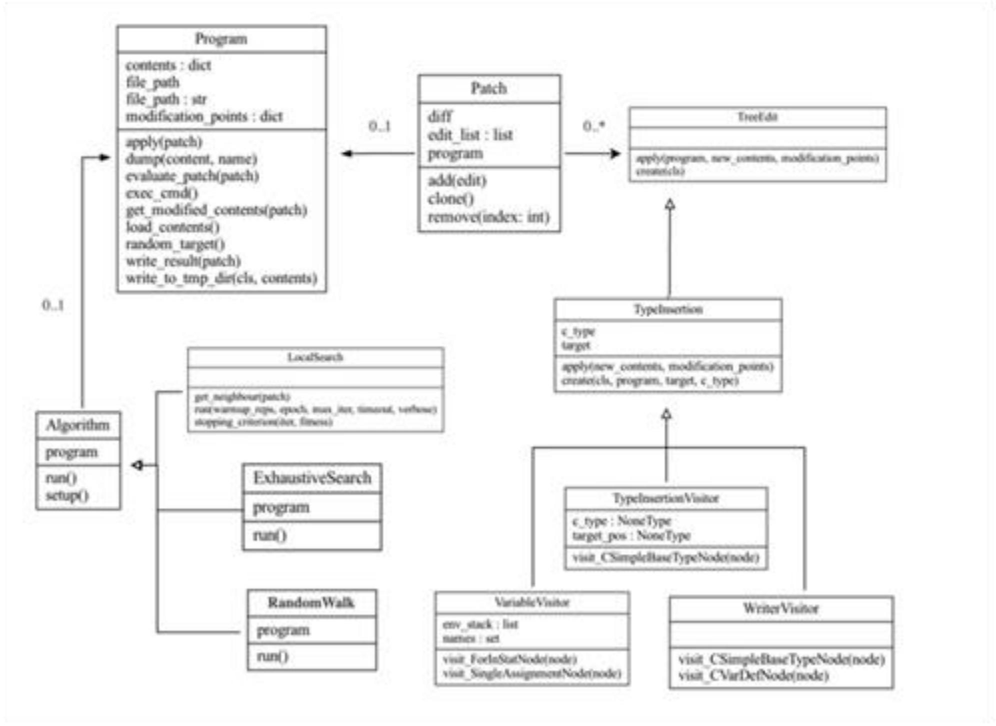
factorial.pyx

```cython
cdef int factorial(int x):
    cdef int y = 1
    cdef int i
    for i in range(x):
        y *= i +1
    return y
```

Py2Cy Framework



Py2Cy Framework

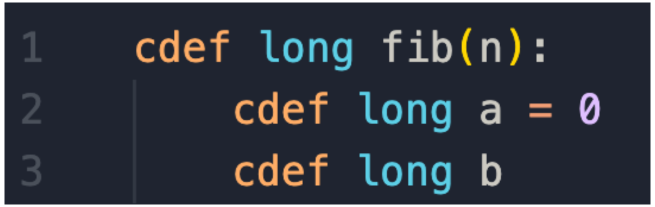


Py2Cy Pipeline

- Convert Python code to AST
- Create Patch
- Apply type Insertion
- Convert AST to Cython code
- Compilation
- Testing

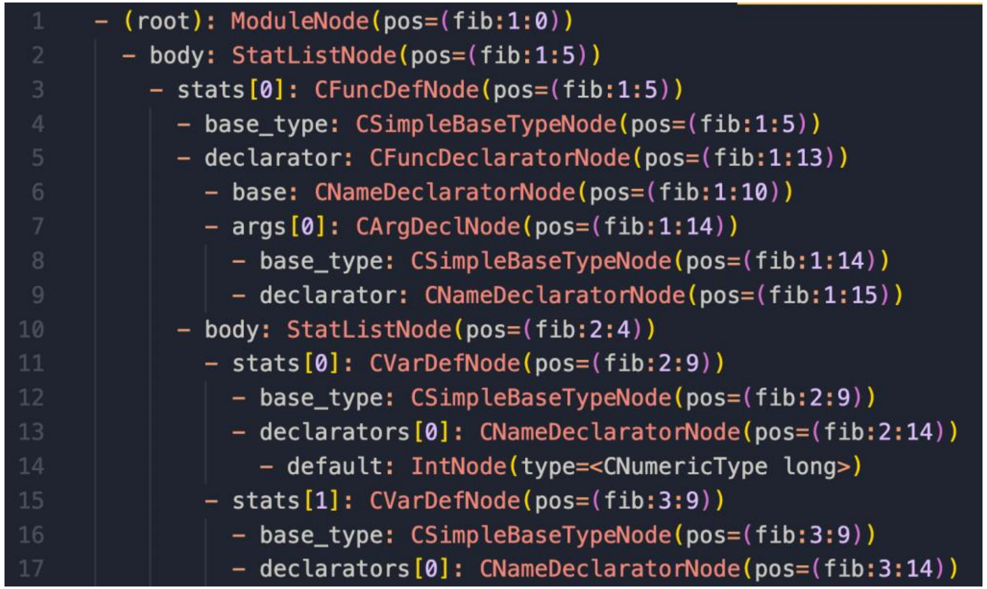
Cython AST

Cython Code



```
cdef long fib(n):
    cdef long a = 0
    cdef long b
```

Cython AST



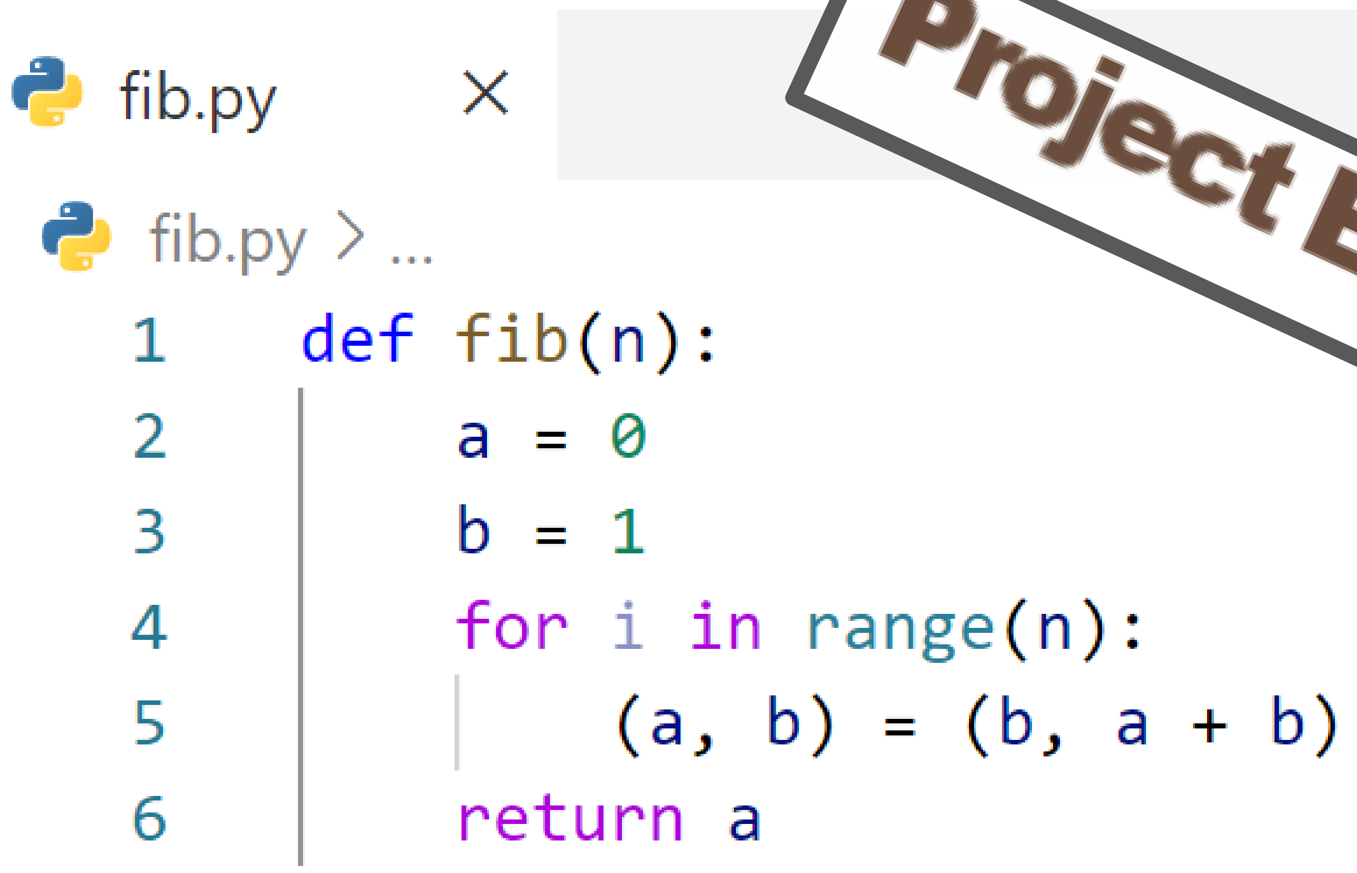
```
- (root): ModuleNode(pos=(fib:1:0))
  - body: StatListNode(pos=(fib:1:5))
    - stats[0]: CFuncDefNode(pos=(fib:1:5))
      - base_type: CSimpleBaseTypeNode(pos=(fib:1:5))
      - declarator: CFuncDeclaratorNode(pos=(fib:1:13))
        - base: CNameDeclaratorNode(pos=(fib:1:10))
        - args[0]: CArgDeclNode(pos=(fib:1:14))
          - base_type: CSimpleBaseTypeNode(pos=(fib:1:14))
          - declarator: CNameDeclaratorNode(pos=(fib:1:15))
      - body: StatListNode(pos=(fib:2:4))
        - stats[0]: CVarDefNode(pos=(fib:2:9))
          - base_type: CSimpleBaseTypeNode(pos=(fib:2:9))
          - declarators[0]: CNameDeclaratorNode(pos=(fib:2:14))
            - default: IntNode(type=<CNumericType long>)
        - stats[1]: CVarDefNode(pos=(fib:3:9))
          - base_type: CSimpleBaseTypeNode(pos=(fib:3:9))
          - declarators[0]: CNameDeclaratorNode(pos=(fib:3:14))
```

https://www.bryanwhitefield.com.au/wp-content/uploads/2020/03/Experiment.png

Program to Optimize

VERIFIABLE

EXISTING SOURCE CODE

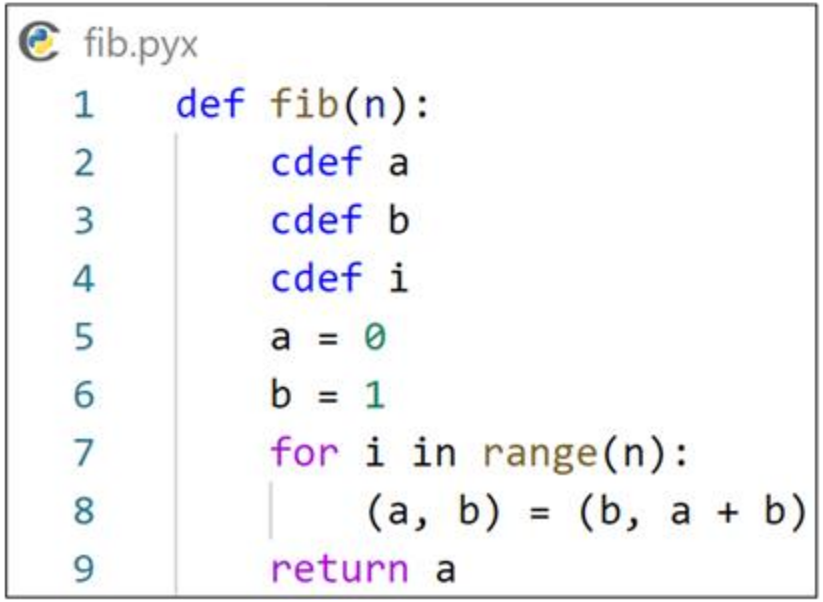


```python
def fib(n):
    a = 0
    b = 1
    for i in range(n):
        (a, b) = (b, a + b)
    return a
```

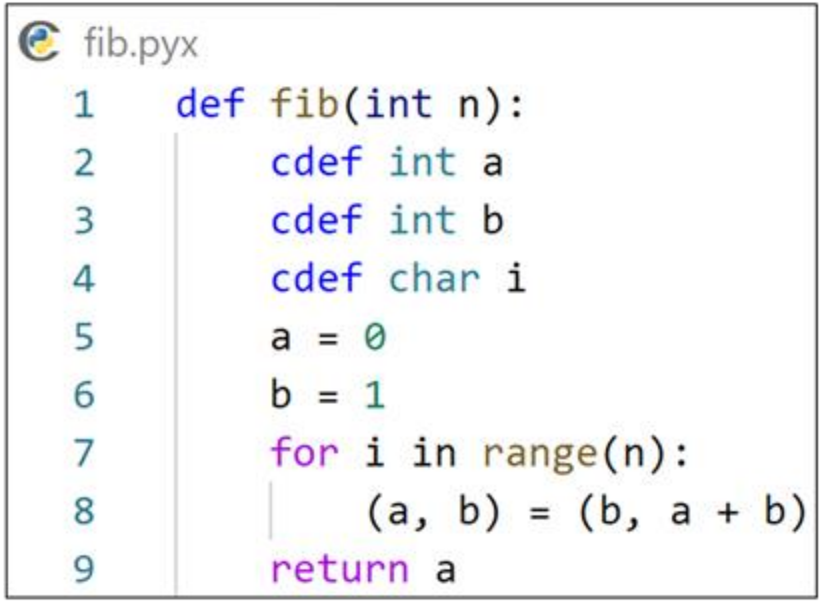
Project Euler .net

https://github.com/TheAlgorithms/Python/blob/master/project_euler/

Search Step 1

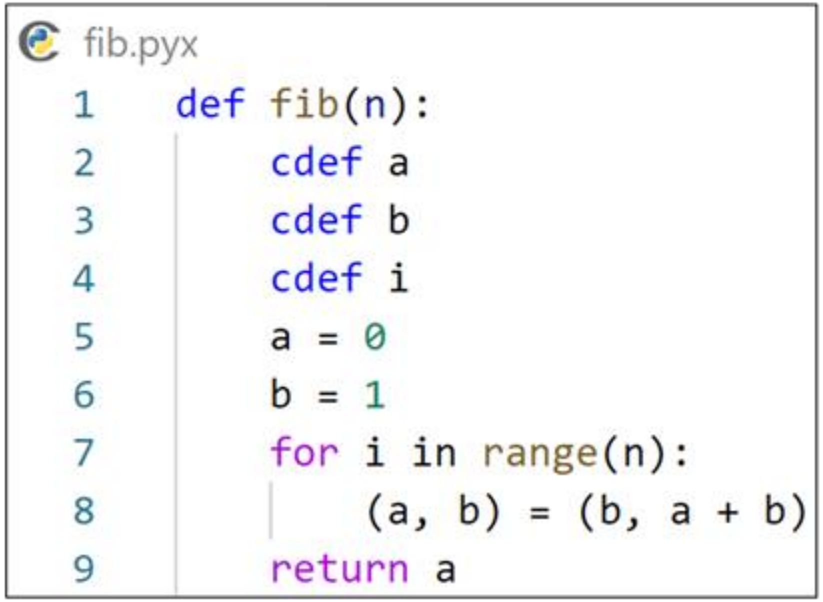

fib.pyx

```python
def fib(n):
    cdef a
    cdef b
    cdef i
    a = 0
    b = 1
    for i in range(n):
        (a, b) = (b, a + b)
    return a
```

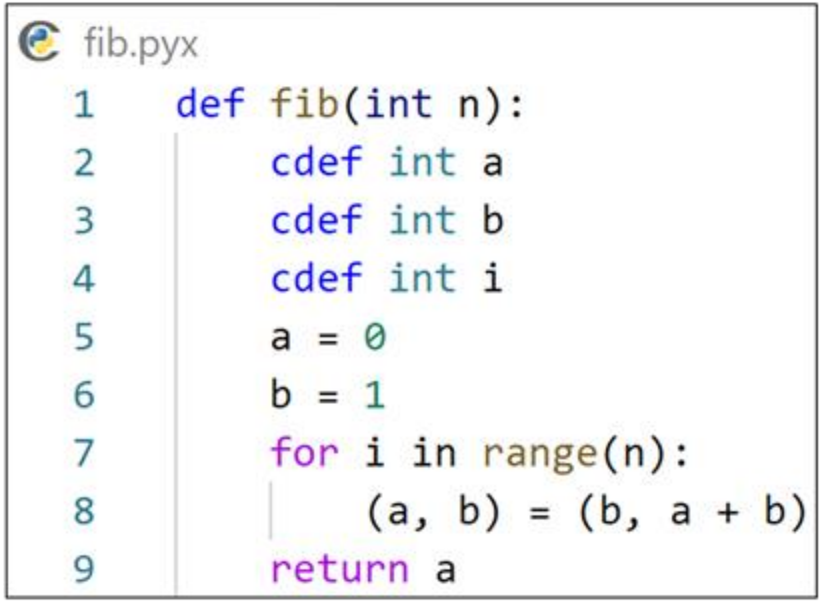

fib.pyx

```python
def fib(int n):
    cdef int a
    cdef int b
    cdef char i
    a = 0
    b = 1
    for i in range(n):
        (a, b) = (b, a + b)
    return a
```

Search Step 2

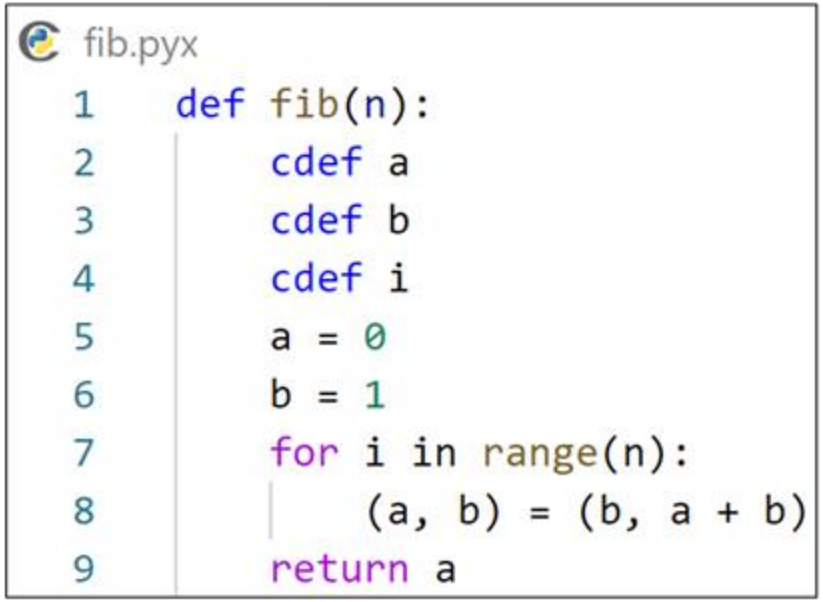

fib.pyx

```
def fib(n):
    cdef a
    cdef b
    cdef i
    a = 0
    b = 1
    for i in range(n):
        (a, b) = (b, a + b)
    return a
```


fib.pyx

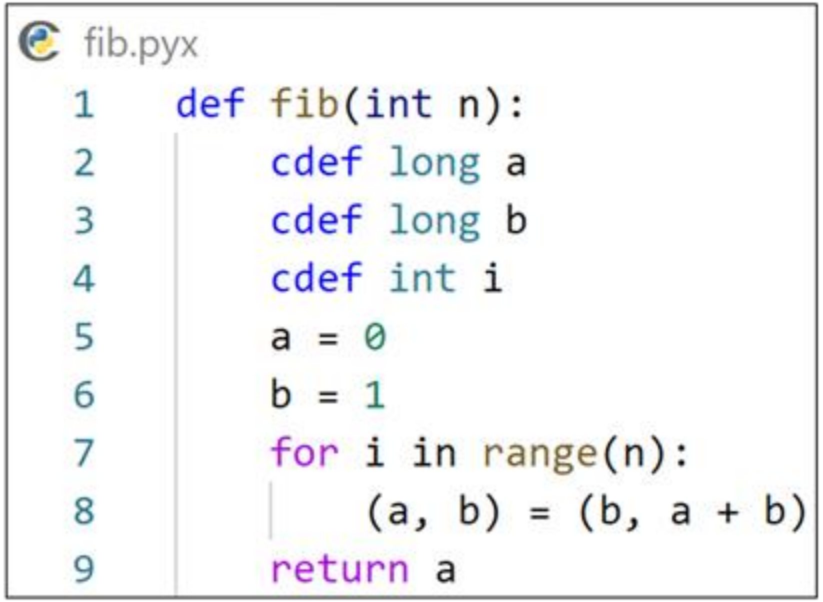
```
def fib(int n):
    cdef int a
    cdef int b
    cdef int i
    a = 0
    b = 1
    for i in range(n):
        (a, b) = (b, a + b)
    return a
```

Search Step 3



fib.pyx

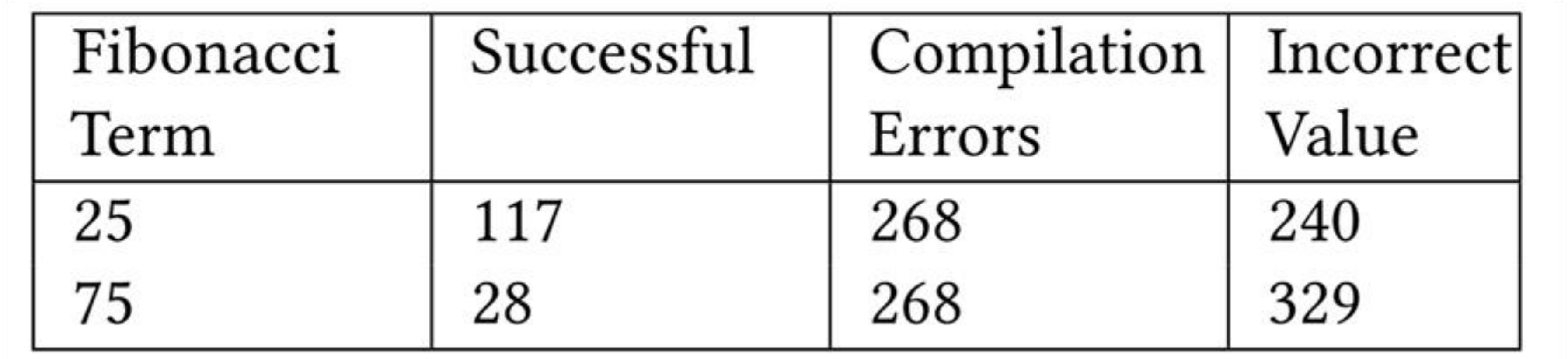
```
def fib(n):
    cdef a
    cdef b
    cdef i
    a = 0
    b = 1
    for i in range(n):
        (a, b) = (b, a + b)
    return a
```



fib.pyx

```
def fib(int n):
    cdef long a
    cdef long b
    cdef int i
    a = 0
    b = 1
    for i in range(n):
        (a, b) = (b, a + b)
    return a
```

Results - Compilations



Fibonacci Term	Successful	Compilation Errors	Incorrect Value
25	117	268	240
75	28	268	329

Table 2: Computational Search Results.

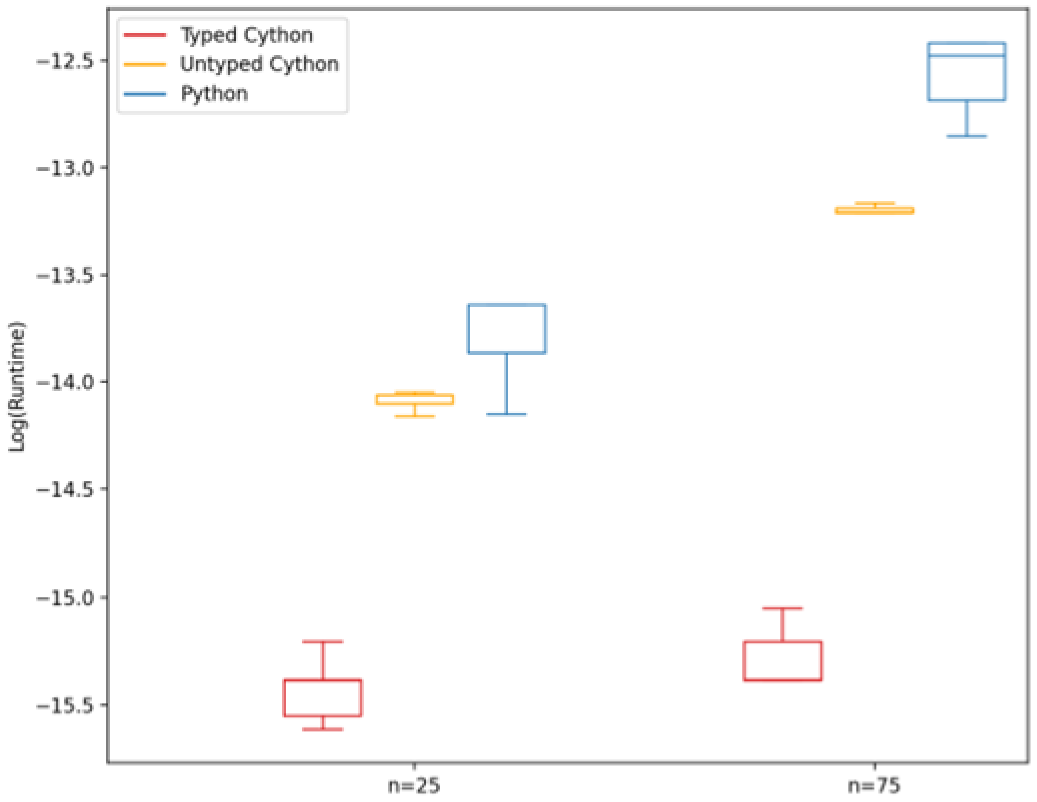


Figure 4.1: Run-times for computing Fibonacci numbers.

https://mentorphiledotcom.files.wordpress.com/2018/09/livedemo-1.png