

4. 再帰呼び出しを可視化してみよう

Python Tutorで再帰呼び出しを試してみよう

- 以下の階乗を計算する関数を Python Tutorで実行してみましよう

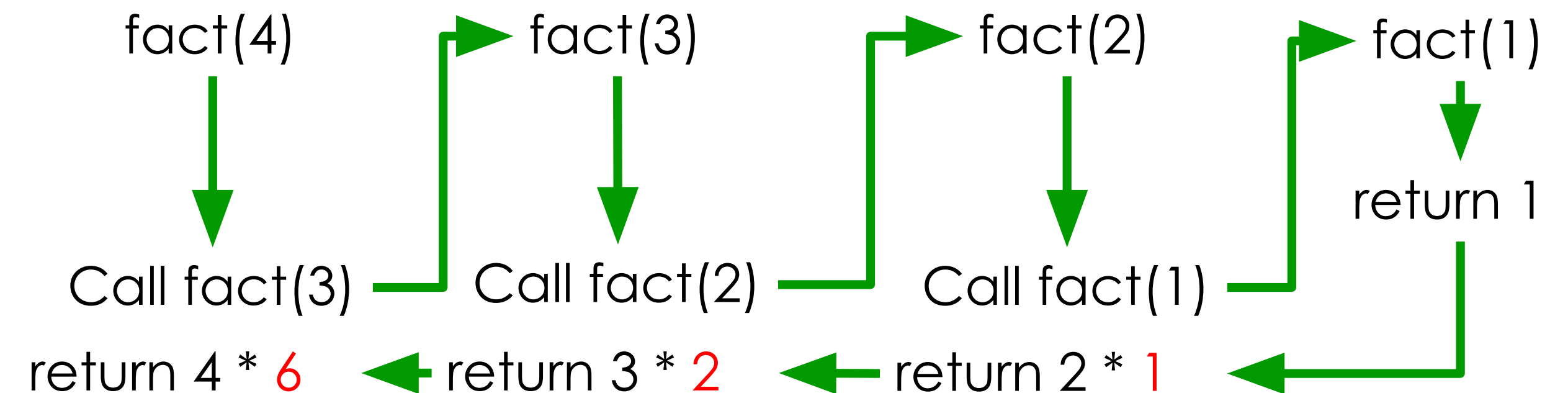
```
Write code in Python 3.6
```

```
1 def fact(n):
2     if n <= 1:
3         return 1
4     return n * fact(n-1)
5
6 fact(4)
```

(再掲) 再帰呼出しの例

- 例えば、`fact(4)` を呼び出すと、以下のような手順で実行されて結果が求まります

```
def fact(n):  
    if n <= 1:  
        return 1  
    return n * fact(n - 1)
```



Python Tutorで再帰呼び出しを試してみよう

- 実行の手順が確認できましたか？

The screenshot displays the Python Tutor interface for Python 3.6. The code being executed is a factorial function:

```
1 def fact(n):  
2     if n <= 1:  
3         return 1  
4     return n * fact(n-1)  
5  
6 fact(4)
```

The execution is at Step 15 of 18. The code editor shows the function definition and the call to `fact(4)`. The `fact` function is being called with `n=4`. The `fact` variable in the Global frame points to the `function fact(n)` object. The `fact` frame shows `n=4`. The `fact` frame shows `n=3`. The `fact` frame shows `n=2`. The `fact` frame shows `n=1` and the `Return value` is 1.

Navigation buttons: << First, < Prev, Next >, Last >>. Step 15 of 18.

フィボナッチ数列を Python Tutorで試してみよう

- 以下のフィボナッチ数列を求めるプログラムを Python Tutor で実行してみてください
 - フィボナッチ数列については、2-2計算量のスライドを見直して復習してください
- 関数 fib(n) が呼び出される順番は、以下のようになるはずです
 - fib(5)->fib(4)->fib(3)->fib(2)->fib(1)->fib(0)->fib(1)->fib(2)->fib(1)->fib(0)
->fib(3)->fib(2)->fib(1)->fib(0)->fib(1)

```
Write code in Python 3.6
1 def fib(n):
2     if n <= 0:
3         return 0
4     if n == 1:
5         return 1
6     return fib(n-1) + fib(n-2)
7
8
9 fib(5)
10
```