

4. 再帰の動作(補足)



そもそも関数を呼び出すとは

```
def hello3():
    print("hello")
    print("hello")
    print("hello")

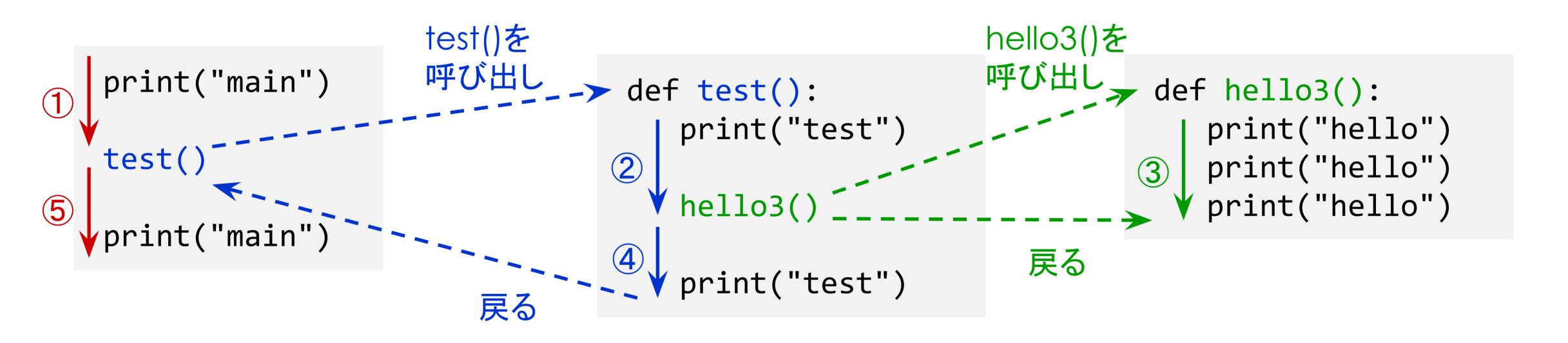
print("main")
hello3()
print("finished")
```

- プログラムは、原則上から下に向かって処理が実行される
- ・ 関数の呼び出しがあった場合は、現在の処理の流れを一 旦止めて、呼び出した関数内を上から順に実行する
- return が実行されるか、関数の最後まで処理が完了すると、元の処理の流れに戻る

```
print("main")
| B数呼び出し | print("hello")
| hello3() | print("hello")
| print("finished") | 元の処理の流れに戻る
```



関数を入れ子に呼び出すときは





再帰の考え方も同じ

```
def hello():
    print("hello")
    hello()
    print("hello")
```

```
hello()を hello()を movification print("main") 中で出し、def hello(): print("hello") print("hello") hello() print("main") print("hello") print("hello") print("hello")
```

この場合は、無限に呼び出し続けるので、終わらないプログラムになる



例:階乗(n!)の関数

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

    x = fact(n-1)

    return n * x</pre>
```



例:階乗(n!)の関数

