

프로그래밍언어론 학습활동: 다양한 프로그래밍 언어의 Factorial 함수 비교

1871056 한지수

1. Bash (Functional Language)

```
1 factorial ()
2 {
3     local num=$1;
4     if [ $num = 0 ]; then
5         echo 1
6         return ;
7     fi;
8     echo $(( $num * $(factorial $(( $num - 1 )) ) ))
9 }
10
11 for ((n = 0; n <= 10; n++))
12 do
13     echo "$n! = " $(factorial $((n)))
14 done
```

Execute Mode, Version, Inputs & Arguments

5.0.011

Stdin Inputs

☐ Interactive

CommandLine Arguments

Result

CPU Time: 0.00 sec(s), Memory: 3472 kilobyte(s) executed in 0.697 sec(s)

```
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

2. Ruby (Functional Language)

```
1 #!/usr/bin/env ruby
2 def factorial(n)
3     if n == 0
4         1
5     else
6         n * factorial(n - 1)
7     end
8 end
9
10 0.upto(10) do |n|
11     print(n, "! = ", factorial(n), "\n")
12 end
```

Execute Mode, Version, Inputs & Arguments

2.6.5

Stdin Inputs

☐ Interactive

CommandLine Arguments

Result

CPU Time: 0.12 sec(s), Memory: 13528 kilobyte(s) executed in 0.969 sec(s)

```
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

3. Python 2 (Functional Language)

```
1 def factorial(n):
2     if n == 0:
3         return 1
4     else:
5         return n * factorial(n - 1)
6
7 for n in range(10 + 1):
8     print "%d! = %d" % (n, factorial(n))
```




Execute Mode, Version, Inputs & Arguments

2.7.16

Stdin Inputs

☐ Interactive

CommandLine Arguments

Execute   

Result

CPU Time: 0.06 sec(s), Memory: 8144 kilobyte(s) executed in 0.721 sec(s)

```
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

4. PHP (Functional Language)

```
1 <?php
2 function factorial($n)
3 {
4     if ($n == 0) {
5         return 1;
6     }
7     return $n * factorial($n - 1);
8 }
9
10
11 for ($n = 0; $n <= 10; $n++) {
12     echo $n . "! = " . factorial($n) . "\n";
13 }
14 ?>
```

Execute Mode, Version, Inputs & Arguments

7.3.10

Stdin Inputs

☐ Interactive

CommandLine Arguments

Execute   

Result

CPU Time: 0.01 sec(s), Memory: 19892 kilobyte(s) executed in 0.741 sec(s)

```
0! = 1
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800
```

5. Java (Object-oriented Language)

```
1 public class FactorialExample{
2     public static void main(String args[]){
3         int i,fact=1;
4         int number=10;
5         for(i=1;i<=number;i++){
6             fact=fact*i;
7             System.out.println(i+"! = "+fact);
8         }
9     }
10 }
```

Execute Mode, Version, Inputs & Arguments

JDK 11.0.4

Stdin Inputs

Interactive

CommandLine Arguments

Execute

Result

CPU Time: 0.20 sec(s), Memory: 35072 kilobyte(s) compiled and executed in 0.84 sec(s)

1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
6! = 720
7! = 5040
8! = 40320
9! = 362880
10! = 3628800

추가.CPU/Memory 시간 비교

Language	CPU (sec)	Memory (kilobyte(s))
1. Bash	0.00	3472
2. Ruby	0.12	13528
3. Python 2	0.06	8144
4. PHP	0.01	19892
5. Java	0.20	35072

Factorial 함수 자체가 가벼운 함수이다보니 Functional Language 이 Object-oriented Language 보다 CPU 와 Memory 에서 효율적이다.