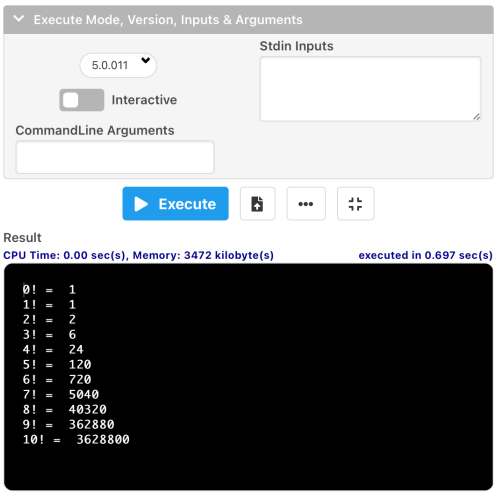
프로그래밍언어론 학습활동: 다양한 프로그래밍 언어의 Factorial 함수 비교

1871056 한지수

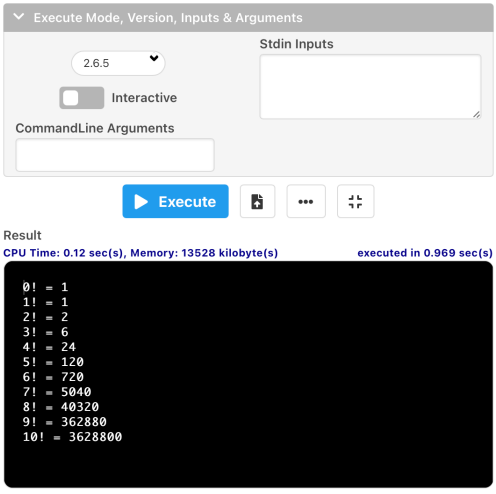
1. Bash (Functional Language)

```bash
factorial ()
{
    local num=$1;
    if [ $num = 0 ]; then
        echo 1
        return ;
    fi;
    echo $(( $num * $(factorial $(( $num - 1 )) ) ))
}

for ((n = 0; n <= 10; n++))
do
    echo "$n! = " $(factorial $(($n)))
done
```



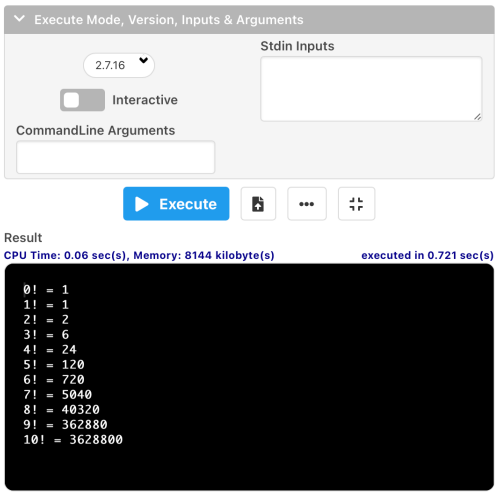
2. Ruby (Functional Language)

```ruby
#! /usr/bin/env ruby
def factorial(n)
    if n == 0
        1
    else
        n * factorial(n - 1)
    end
end

0.upto(10) do |n|
    print(n, "! = ", factorial(n), "\n")
end
```



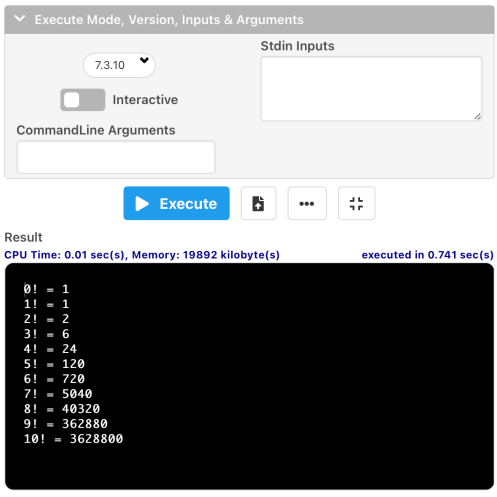
3. Python 2 (Functional Language)

```python
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n - 1)

for n in range(10 + 1):
    print "%d! = %d" % (n, factorial(n))
```



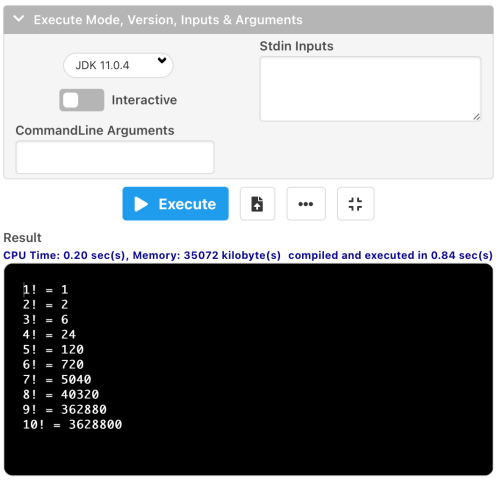
4. PHP (Functional Language)

```php
<?php
function factorial($n)
{
    if ($n == 0) {
        return 1;
    }

    return $n * factorial($n - 1);
}

for ($n = 0; $n <= 10; $n++) {
    echo $n . "! = " . factorial($n) . "\n";
}
?>
```



5. Java (Object-oriented Language)

```java
public class FactorialExample{
 public static void main(String args[]){
  int i,fact=1;
  int number=10;
  for(i=1;i<=number;i++){
      fact=fact*i;
      System.out.println(i+"! = "+fact);
  }
 }
}
```



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.