

# 소프트웨어와 문제해결

# 실습 1

문자열과 횟수를 매개변수로 전달받아 횟수만큼 문자열을 반복하여 출력하는 `print_str()` 함수를 선언해보자. 문자열과 횟수를 각각 입력받아 `print_str()` 함수를 호출하여 실행해보자.


문자열 : 파이썬

횟수 : 3

파이썬

파이썬

파이썬

 `def print_str(st, cnt)` 형태로 함수를 선언하고, 문자열과 횟수를 입력받아 `print_str(s, c)` 형태로 함수를 호출한다.

# 실습 2

두 개의 숫자를 전달받아 큰 수를 반환하는 maxnum() 함수를 선언해보자. 두 개의 숫자를 입력받아 함수를 호출하고 결과를 출력해보자.

숫자1 : 3

숫자2 : 5

큰 수 = 5




def maxnum(m, n) 형태로 함수를 선언하고, m과 n을 비교하여 큰 수를 반환한다.

# 실습 3

가로와 세로 길이를 매개변수로 전달받아 사각형의 넓이를 구하여 반환하는 `rectangle_area()` 함수를 선언해보자. 가로와 세로 길이를 입력받아 함수를 호출하고 결과를 반환받아 출력해보자.

```
가로 : 3  
세로 : 5  
가로 3 세로 5 인 사각형의 넓이 = 15
```

 `def rectangle_area(col, row)` 형태로 함수를 선언하고, 사각형의 넓이는 `가로*세로`로 계산한다.

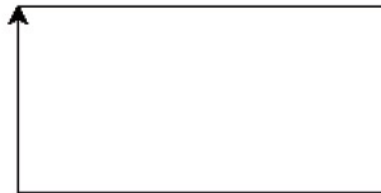
# 실습 4


실습 3번 문제 '에, 가로와 세로 길이를 매개변수로 전달받아 터틀 스크린에 사각형을 그리는 `rectangle_draw()` 함수를 추가하여 선언해보자. 사각형을 그리는 방법은 '5장의 Basic Coding 12번 문제'의 사각형 그리기를 참고하고, 가로와 세로 길이를 입력받아 `rectangle_draw()` 함수를 호출해보자.

가로 : 200

세로 : 100

가로 200 세로 100 인 사각형의 넓이 = 20000



 `def rectangle_draw(col, row)` 형태로 함수를 선언하고, `turtle.forward(col)`, `turtle.right(90)`, `turtle.forward(row)`, `turtle.right(90)` 순으로 사각형을 그린다.

# 실습 5

‘실습 4번 문제’를 참고하여, 사각형의 넓이를 사각형 위에 출력해보자.

가로 : 200

세로 : 100

가로200세로100인 사각형의 넓이 =20000



tmpstr = "가로" + str(c) + ... 형식으로 출력할 내용을 문자열로 만든 후, turtle.write(tmpstr)로 출력한다.

# 실습 6


자연수를 매개변수로 전달받아 1부터 전달된 자연수까지의 합계를 구하여 반환하는 `one2n_sum1()` 함수를 선언해보자. 자연수를 입력받아 함수를 호출하고 결과를 반환받아 출력해보자. 단, 입력된 수가 1보다 작은 수이면 '입력된 수가 1보다 작습니다.'라는 문자열을 출력한다.

자연수 : -1

입력된 수가 1보다 작습니다.

자연수 : 10

1 -- 10 = 55


 `def one2n_sum1(n)` 형태로 함수를 선언하며, 1부터 `n`까지의 합계를 구한 후, `return (s)`로 합계를 반환한다.

# 실습 7

임의의 두 정수를 매개변수로 전달받아 작은 수부터 큰 수까지의 합계를 구하여 반환하는 `m2n_sum()` 함수를 선언해보자. 정수를 입력받아 함수를 호출하고 결과를 반환받아 출력해보자.

```
정수1 : 3  
정수2 : 5  
3 -- 5 = 12
```

```
정수1 : 5  
정수2 : 3  
5 -- 3 = 12
```

 `def m2n_sum(m, n)` 형태로 함수를 선언하며, `m`부터 `n`까지의 합계를 구하여 결과를 반환한다.



# 실습 8

정수 두 개를 입력받아 두 수의 덧셈, 뺄셈, 곱셈, 나눗셈 계산을 하는 `calc(num1, num2, act="+")` 함수를 선언해보자. 매개변수 `act`는 디폴트 매개변수이며 “+”로 기본값이 설정되고 해당 위치의 인수를 생략하면 디폴트 매개변수로 선언된 “+”가 사용된다. 만약 매개변수 `act`에 “-”, “\*”, “/”가 전달되면 해당 문자열이 전달된다. `act`에 해당하는 연산을 한 후 결과를 반환한다. 아래의 결과 중 7은 `calc(n1, n2)`로 호출, 10은 `calc(n1, n2, "*")`로 호출, “잘못된 연산기호입니다.”는 `calc(n1, n2, "^")`로 호출한 결과이다.


정수1 : 5

정수2 : 2

7

10

잘못된 연산기호입니다.

 `def calc(num1, num2, act="+")` 형태로 선언하고, 두 정수를 입력받은 후 `calc(n1, n2)`, `calc(n1, n2, "*")`, `calc(n1, n2, "^")` 형식으로 각각 호출한다.