

 x 

The main title of the slide, showing the logos for CODINGO and POSCO separated by a large black 'x' symbol. The CODINGO logo is the same as in the header, and the POSCO logo is in a blue, lowercase, sans-serif font.

 스마트 팩토리 3기

The subtitle of the slide, featuring the text "K-Digital Training" in a multi-colored font (K is green, D is orange, i is blue, g is purple, i is green, t is orange, a is blue, l is purple, T is green, r is orange, a is blue, i is purple, n is green, g is orange) followed by the Korean text "스마트 팩토리 3기" in a black, sans-serif font.

함수

함수

- 식별자에 괄호가 붙은 것
- 재사용 가능한 프로그램 조각
- 자주 사용하는 기능을 코드 블록으로 만듦
- 파이썬에 기본으로 포함되어 있는 함수도 있고 프로그래머가 직접 만들 수도 있음

```
# 내장 함수  
a = "Strawberry Moon"  
print(len(a)) # 15  
print(a.count("r")) # 3
```

함수 관련 용어

- 호출하다 : 함수를 사용하다
- 매개변수 : 함수 괄호 내부에 데이터를 넣는 것
- 리턴 값 : 함수를 호출해서 나오는 결과

함수의 기본적인 사용법

```
def 함수 이름 (매개변수1, 매개변수2, ...):  
    수행문 1  
    수행문 2  
    return 반환값
```

def : 함수 선언을 위한 키워드
매개변수 : 함수에 입력할 값

함수 관련 용어

- 수학에서의 함수로 설명
다음과 같은 함수를 정의함

$$f(x) = x^2 + 2x + 1$$

- $x = 3$ 을 대입한 $f(3)$ -> 함수 호출,
3 -> 매개 변수(함수에 3이라는 값 전달)
- $f(3) = 16$ -> 리턴 값! (함수를 호출해서 나오는 결과)

함수 기본 구조

- 입력값X, 결과값X → 그저 함수 내부의 일만 수행

```
# 함수 정의
def 함수명():
    문장

# 함수 호출
함수명()
```

```
# 함수 기본 구조

def say_hello():
    print("hello")
    print("hello")
    print("hello")

say_hello()
```

매개변수를 받는 함수 구조

- 입력값O, 결과값X → 입력 값을 이용해 함수 내부의 일을 수행
 - 매개변수 개수 일치하게

```
# 함수 정의
def 함수명(매개변수1, 매개변수2, ...):
    문장

# 함수 호출
함수명(인수1, 인수2, ...)

# 주의
매개변수: 함수 정의부에 전달하는 값
인수: 함수 호출부에 전달하는 값
```

```
# 매개변수를 받는 함수 구조

def say_hello(n): # f(x)
    for i in range(n):
        print("hello")

say_hello(3) # f(3)
```

결과값이 있는 함수 (return)

- 함수 내부의 수행 결과를 return에 담아서 내보냄

```
# 입력값이 없는 결과값 있는 함수
def 함수명():
    문장
    ...
    return 결과값
```

```
# 입력값도 있는 결과값 있는 함수
def 함수명(매개변수):
    문장
    ...
    return 결과값
```

```
# 결과 값이 있는 함수 (return)
# 1) 입력값이 있는 함수
def add(x, y):
    result = x + y
    return result
```

add(2, 5)

출력??

변수 범위

- 전역 변수: 프로그램 전체에서 사용되는 변수
- 지역 변수: 한정된 지역(ex. 함수)에서만 사용되는 변수

```
x = 10          # 전역 변수
def func():
    # 같은 이름의 변수 선언시
    x = 20      # x는 func의 지역 변수
    print(x)    # func의 지역 변수 출력

func()
print(x)        # 전역 변수 출력
```

재귀 함수

재귀 함수

- 자기 자신을 호출하는 함수
- 종료 조건이 있어야 함

```
def hello():
    print("Hello, World!")
    hello() # 자기 자신 호출

hello()
```

[illegible]

RecursionError: maximum recursion depth exceeded while pickling an object

팩토리얼 구하기

- 팩토리얼 : 그 수보다 작거나 같은 모든 양의 정수의 곱
- Ex) $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

```
# 반복문
n = 5
factorial = 1

for i in range(1, n+1):
    factorial *= i

print(factorial)
```

```
# 재귀 함수
def factorial(n):
    if n == 1:
        return 1
    return n * factorial(n-1)

print(factorial(5))
```