

# 파이썬

상명대학교 융합공과대학 지능·데이터융합학부

휴먼지능정보공학전공

상명대학교 바이오헬스혁신공유대학

dkim@smu.ac.kr

# 강의개요

- 강의소개 및 프로그래밍 개념
  - 프로그래밍과 컴퓨팅사고력 소개
  - 프로그래밍 맛보기
- 변수, 자료형, 연산, 함수
  - 코딩과 기초실습
- 조건문, 연산자
  - 코딩과 기초실습
- 반복문
  - 코딩과 기초실습
- 함수, 매개변수
  - 코딩과 기초실습
- 중간고사

# 강의개요

- 자료형, 리스트
  - 코딩과 기초실습
- 자료형, 튜플
  - 코딩과 기초실습
- 자료형, 딕셔너리
  - 코딩과 기초실습
- 실습예제
  - 코딩과 기초실습
- 파일읽고 쓰기
  - 코딩과 기초실습
- 객체지향 프로그래밍
  - 코딩과 기초실습
- 기말고사

# 프로그래밍 문법

- 변수: 데이터를 저장하는 공간
  - 구별되게 고유한 이름을 붙임
- 자료형(타입) : 데이터(자료)를 표현하는 방법(형태(타입)가 있음)
  - 수치형(정수, 실수, 복소수), 문자열(따옴표)
- 연산자: 프로그램의 산술식이나 연산식을 표현하고 처리하기 위해 제공되는 기호
  - 대입연산, 산술연산, 복합대입연산
- 자료형(타입)변환: 데이터(자료)를 표현하는 방법을 변환
  - 문자열연산, 정수형 변환 `int()` 함수

# 프로그래밍 기본요소

- 연산자: 프로그램의 산술식이나 연산식을 표현하고 처리하기 위해 제공되는 기호
  - 대입연산, 산술연산, 증감연산, 관계연산, 논리연산, 비교연산, 삼항연산
- 명령문: 프로그램이 특정 작업을 수행하도록 프로그래밍 언어로 작성하는 명령
  - 선언문, 대입문, 함수(호출)문, 반복문(for, while), 조건문(if~else, if~elif, switch case), 분기문(break, continue)
- 함수: 하나의 특별한 목적의 작업을 수행하기 위해 독립적으로 설계된 코드 모음
  - 내장함수(매개변수, 반환 값), 사용자정의함수(매개변수, 반환 값), 라이브러리(매개변수, 반환 값)
  - def 함수이름(매개변수) :  
    함수내용  
function 함수이름(매개변수) { 함수내용 }

# 조건문

- 조건문: 조건식이 참인 경우 실행하는 문장(명령문)과 거짓인 경우 실행하는 문장(명령문)을 구분하여 실행하도록 만든 문장

if 조건식 1 :

조건식 1이 참인 경우 실행할 문장 1 (명령문, 함수)

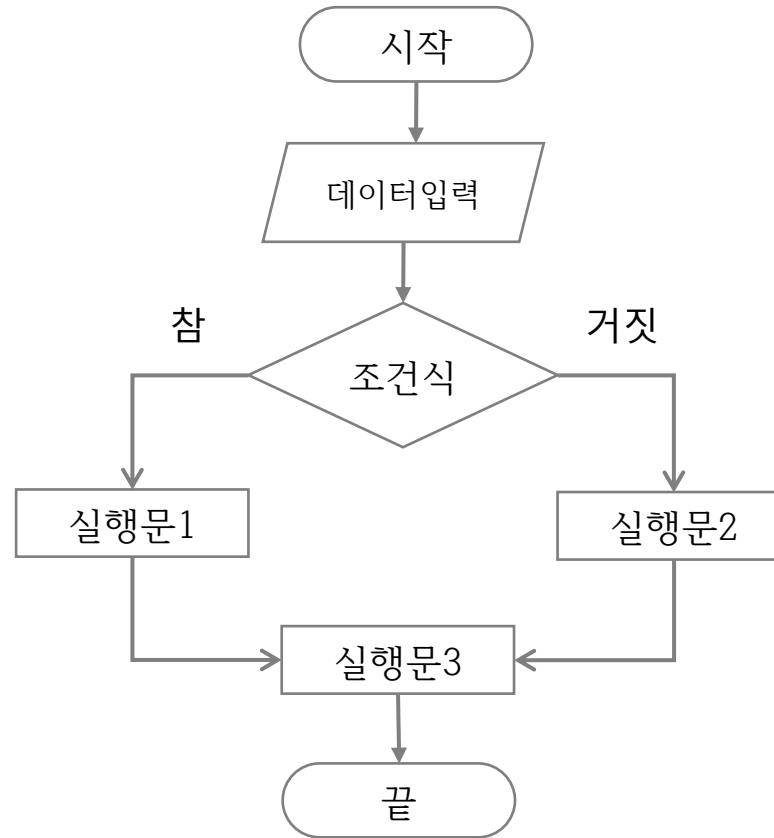
else :

조건식 1이 거짓인 경우 실행할 문장 2 (명령문, 함수)

if 조건문이 종료되고 실행할 문장 3 (명령문, 함수)

# 조건문

- 조건문: 조건식이 참인 경우 실행하는 문장(명령문)과 거짓인 경우 실행하는 문장(명령문)을 구분하여 실행하도록 만든 문장



# 조건문

- 조건식: 참과 거짓을 구분할 수 있는 식
  - 관계연산자, 논리연산자, 포함연산자 사용
    - 대입연산자: `a=0` (조건식이 아님, 항상 참)
    - 관계연산자:
      - `a == b`: a, b가 같으면 참(True) , ex) `a=1, b=1, print(a==b) → True`
      - `!=` : a, b가 같지않으면 참(True)
      - `a < b` : a가 작으면 참(True)
      - `a > b` : a가 크면 참(True)
      - `a <= b` : a가 작거나 같으면 참(True)
      - `a >= b` : a가 크거나 같으면 참(True)
    - 논리연산자:
      - `a or b` : a와 b 둘 중에 하나만 참이면 참, ex) `a=true, b=true, print(a or b) → True`
      - `a and b` : a와 b 둘 다(모두) 참이어야 참 ex) `a=true, b=true, print(a and b) → True`
      - `not a` : a가 거짓이면 참



# 조건문

- 조건식: 참과 거짓을 구분할 수 있는 식

- 포함연산자:

- in : 문자열, 리스트, 튜플, 집합 등의 자료형에서 찾고자 하는 값(문자)이 포함되어 있으면 참  
a="상명", b="상명대학교", print(a in b) ## → True  
a=10, b=[10,20,30,40], print(a in b) ## → True  
a=50, b=(10,20,30,40), print(a in b) ## → False  
a="상명", b={"상명대학교", "융합공과대학", "경영경제대학", "계당교양교육원"}, print(a in b) ## → False
    - not in : 문자열, 리스트, 튜플, 집합 등의 자료형에서 찾고자 하는 값(문자)이 포함되어 있지 않으면 참  
a="상명", b="상명대학교", print(a not in b) ## → False  
a=10, b=[10,20,30,40], print(a not in b) ## → False  
a=50, b=(10,20,30,40), print(a not in b) ## → True  
a="상명", b={"상명대학교", "융합공과대학", "경영경제대학", "계당교양교육원"}, print(a not in b) ## → True

# 조건문

- 조건문: 조건식이 참인 경우 실행하는 문장(명령문)과 거짓인 경우 실행하는 문장(명령문)을 구분하여 실행하도록 만든 문장

if 조건식 1 :

조건식 1이 참인 경우 실행할 문장 1(명령문, 함수)

elif 조건식 2:

조건식 2가 참인 경우 실행할 문장 2(명령문, 함수)

elif 조건식 3:

조건식 3이 참인 경우 실행할 문장 3(명령문, 함수)

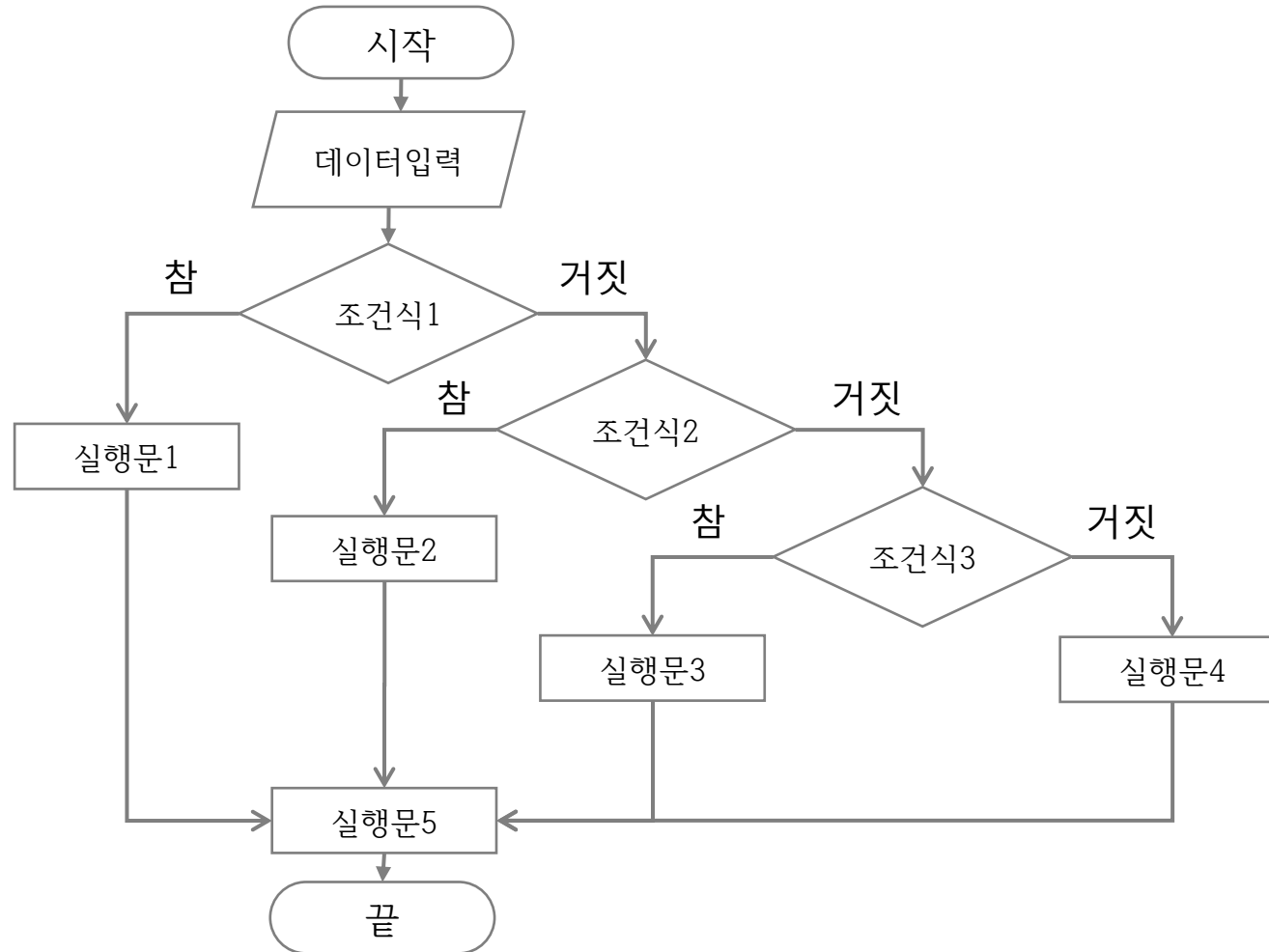
else :

조건식 1,2,3이 거짓인 경우 실행할 문장 4 (명령문, 함수)

if 조건문이 종료되고 실행할 문장5 (명령문, 함수)

# 조건문

- 조건문: 조건식이 참인 경우 실행하는 문장(명령문)과 거짓인 경우 실행하는 문장(명령문)을 구분하여 실행하도록 만든 문장



# 조건문

- 조건문: 조건식이 참인 경우 실행하는 문장(명령문)과 거짓인 경우 실행하는 문장(명령문)을 구분하여 실행하도록 만든 문장

if 조건식 1 :

조건식 1이 참인 경우 실행할 문장 1(명령문, 함수)

if 조건식 2:

조건식 1, 2가 참인 경우 실행할 문장 2(명령문, 함수)

else 조건식 3:

조건식 1이 참이고 조건식 2가 거짓인 경우 실행할 문장 3(명령문, 함수)

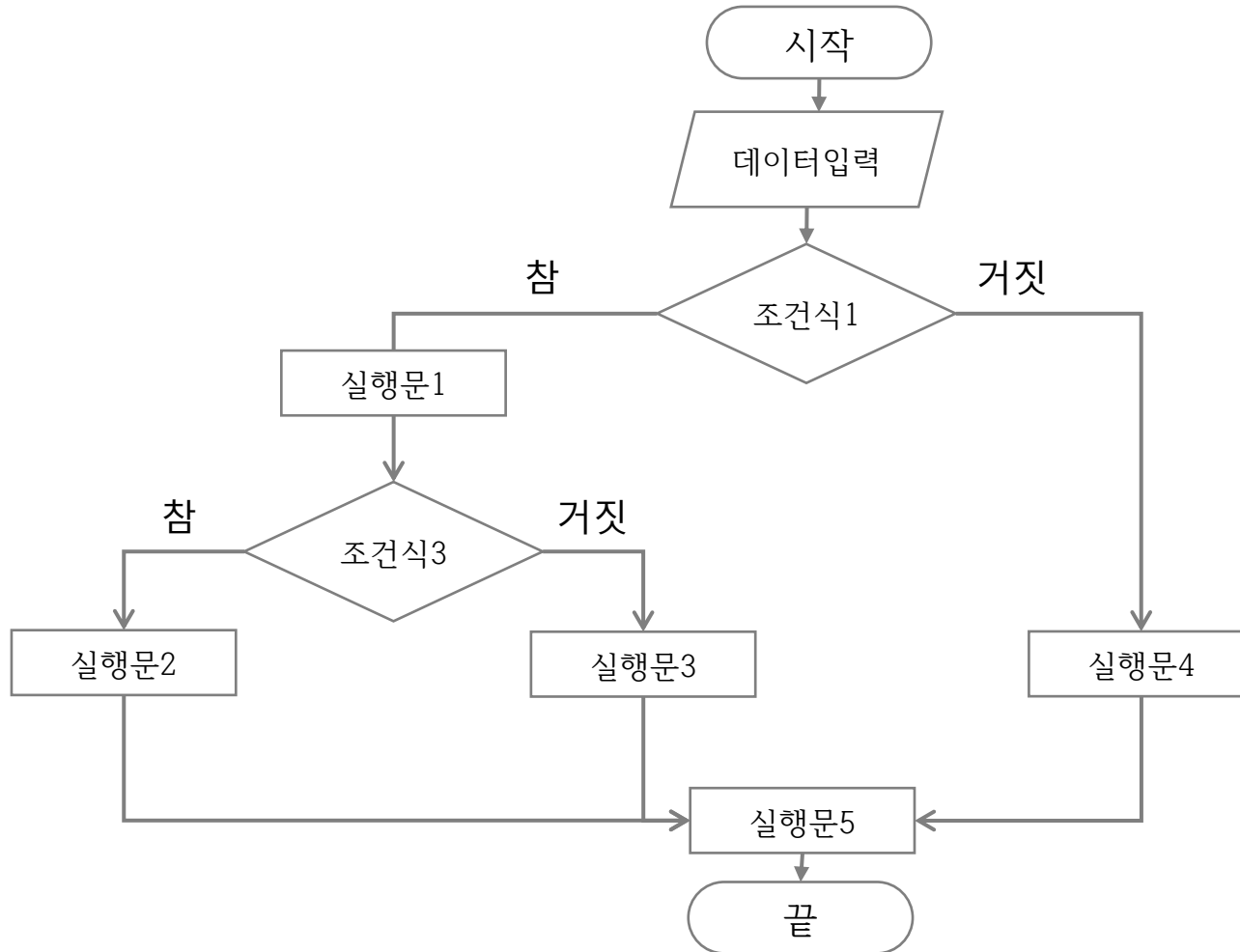
else :

조건식 1이 거짓인 경우 실행할 문장 4(명령문, 함수)

if 조건문이 종료되고 실행할 문장5 (명령문, 함수)

# 조건문

- 조건문: 조건식이 참인 경우 실행하는 문장(명령문)과 거짓인 경우 실행하는 문장(명령문)을 구분하여 실행하도록 만든 문장



# 프로그래밍 설계

## 프로그래밍 명세

- 두 수와 사칙 연산자를 입력 받아 사칙연산을 수행하고 결과를 출력하는 프로그램

# 프로그래밍 설계

## 프로그래밍 명세

- 두 수와 사칙 연산자를 입력 받아 사칙연산을 수행하고 결과를 출력하는 프로그램

입력(Input)

처리(Processing)

출력(Output)

▪

▪

▪

# 프로그래밍 설계

## 프로그래밍 명세

- 두 수와 사칙 연산자를 입력 받아 사칙연산을 수행하고 결과를 출력하는 프로그램

### 입력(Input)

- 두 수(값)
- 사칙 연산자
- 입력받는다

### 처리(Processing)

- 두 수(값)
- 사칙 연산한다

### 출력(Output)

- 처리 결과(값)
- 출력한다



# 프로그래밍 설계

## 프로그래밍 명세

- 두 수와 사칙 연산자를 입력 받아 사칙연산을 수행하고 결과를 출력하는 프로그램

### 입력(Input)

- 두 수(값)
- 사칙 연산자
- 입력받는다

- 데이터:
- 변수(이름, 형태):
- 자료형:
- 연산자:
- 명령문:
- 함수:
- 이벤트:

### 처리(Processing)

- 두 수(값)
- 사칙 연산한다

- 데이터:
- 변수(이름, 형태):
- 자료형:
- 연산자:
- 명령문:
- 함수:
- 이벤트:

### 출력(Output)

- 처리 결과(값)
- 출력한다

- 데이터:
- 변수(이름, 형태):
- 자료형:
- 연산자:
- 명령문:
- 함수:
- 이벤트:

# 프로그래밍 설계

## 프로그래밍 명세

- 두 수와 사칙 연산자를 입력 받아 사칙연산을 수행하고 결과를 출력하는 프로그램

### 입력(Input)

- 두 수(값)
- 사칙 연산자
- 입력받는다

- 데이터: 두 수
- 변수(이름, 형태): a,b(정수)
- 자료형: 정수형, 문자열
- 연산자: 대입연산, 비교연산
- 명령문: 순서, 조건문
- 함수: 입력함수(내장함수,사용자함수)
- 이벤트: ~할 때

### 처리(Processing)

- 두 수(값)
- 사칙 연산한다

- 데이터: 두 수, 처리 결과값
- 변수(이름, 형태): a,b,result(정수형)
- 자료형: 정수형, 문자열
- 연산자:대입,사칙연산,비교연산
- 명령문: 순서, 조건문
- 함수: 연산함수(내장함수,사용자함수)
- 이벤트: ~할 때

### 출력(Output)

- 처리 결과(값)
- 출력한다

- 데이터: 처리 결과값
- 변수(이름, 형태): result(정수형)
- 자료형: 정수형, 실수형
- 연산자: 대입연산, 비교연산
- 명령문: 순서, 조건문
- 함수: 출력함수(내장함수,사용자함수)
- 이벤트: ~ 할 때

# 프로그래밍 설계

## 프로그래밍 명세

- 두 수와 사칙 연산자를 입력 받아 사칙연산을 수행하고 결과를 출력하는 프로그램



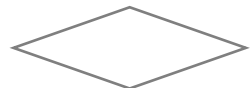
시작, 끝



초기값 선언(설정)



데이터 입,출력



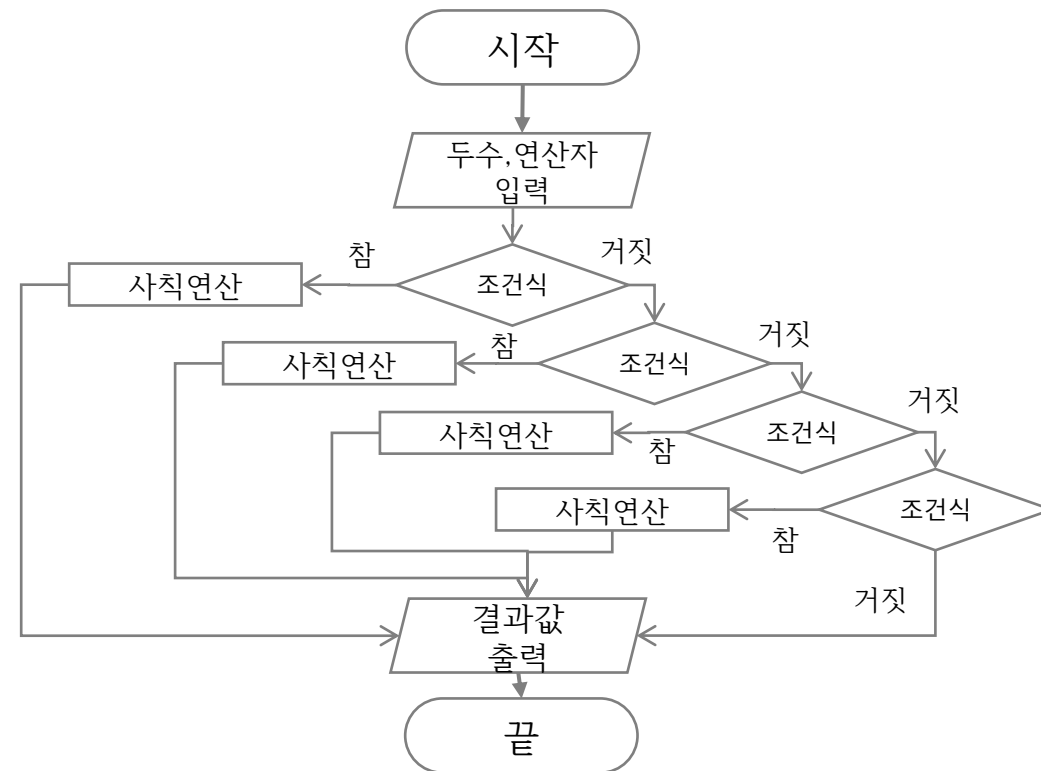
조건에 따른 판단(분기)



연산 및 처리



데이터 제어(흐름)



# 프로그래밍 설계 – 실습

## 프로그래밍 명세

- 1)사용자에게 ‘묵’, ‘찌’, ‘빠’ 문자열을 입력받고 입력된 문자열이 각 ‘묵’, ‘찌’, ‘빠’ 인 경우에만 각 해당 문자열을 출력한다
- 2)다른 문자열을 입력된 경우 “재입력” 메시지를 출력해준다
- 3)사용자가 ‘묵’, ‘찌’, ‘빠’ 각 문자열을 입력 하지 않고 엔터를 누르면 프로그램은 종료된다

### 입력(Input)

- ‘묵’, ‘찌’, ‘빠’ 문자열
- 엔터

- 데이터:
- 변수(이름, 형태):
- 자료형:
- 연산자:
- 명령문:
- 함수:

### 처리(Processing)

- ‘묵’, ‘찌’, ‘빠’ 문자열 비교 판단
- ‘묵’, ‘찌’, ‘빠’ 반복 입력
- 엔터 입력 비교 판단 (종료)

- 데이터:
- 변수(이름, 형태):
- 자료형:
- 연산자:
- 명령문:
- 함수:
- 이벤트:

### 출력(Output)

- ‘묵’, ‘찌’, ‘빠’ 문자열 출력
- ‘재입력하세요’ 문자열 출력

- 데이터:
- 변수(이름, 형태):
- 자료형:
- 연산자:
- 명령문:
- 함수:
- 이벤트:

# 프로그래밍 설계 – 실습

## 프로그래밍 명세

- 1)사용자에게 ‘묵’, ‘찌’, ‘빠’ 문자열을 입력받고 입력된 문자열이 각 ‘묵’, ‘찌’, ‘빠’ 인 경우에만 각 해당 문자열을 출력한다
- 2)다른 문자열을 입력된 경우 “재입력” 메시지를 출력해준다
- 3)사용자가 ‘묵’, ‘찌’, ‘빠’ 각 문자열을 입력 하지 않고 엔터를 누르면 프로그램은 종료된다



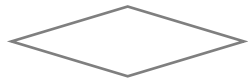
시작, 끝



초기값 선언(설정)



데이터 입,출력



조건에 따른 판단(분기)



연산 및 처리



데이터 제어(흐름)

# 프로그래밍 설계 - 실습

\*untitled\*

— □ ×

File Edit Format Run Options Window Help

```
목 짜 빠 중 하나를 내주세요,종료하시려면 엔터를 눌러주세요목
목을 내셨습니다
목 짜 빠 중 하나를 내주세요,종료하시려면 엔터를 눌러주세요짜
짜를 내셨습니다
목 짜 빠 중 하나를 내주세요,종료하시려면 엔터를 눌러주세요빠
빠를 내셨습니다
목 짜 빠 중 하나를 내주세요,종료하시려면 엔터를 눌러주세요가위
목 짜 빠로 입력하세요
목 짜 빠 중 하나를 내주세요,종료하시려면 엔터를 눌러주세요
종료합니다
>>> `
```

# 맺음말