

코딩의 표현 방법

- 파이썬 프로그래밍

Presented by

한국공학대학교 SW대학

-
- I. 코딩의 표현 방법
 - II. `print()` 함수를 사용한 다양한 출력
 - III. 변수의 선언과 사용
 - IV. 데이터 표현 단위와 진수 변환
 - V. 기본 데이터형
-

-
- I. 코딩의 표현 방법에 대하여 설명할 수 있다.
 - II. 컴퓨팅 사고와 표현하는 방법에 대하여 설명할 수 있다.
-

▶ 프로그래밍과 코딩

- 컴퓨터로 문제해결 시 먼저 알고리즘부터 개발
 - 알고리즘을 의사코드(pseudo code), 순서도(flowchart)를 이용하여 표현
 - 컴퓨터 프로그램을 줄여서 프로그램
- 컴퓨터 프로그램을 작성하여 최종 결과 도출
- 컴퓨터가 이해할 수 있는 규칙에 따라 프로그래밍 언어로 작성

▶ 알고리즘

- 문제 해결을 위한 방법이나 절차
- 알고리즘의 조건
 - 입력 : 외부에서 자료를 입력 받을 수 있어야 함
 - 출력 : 최소한 한 가지 이상의 결과가 나와야 함
 - 명확성 : 각각의 명령들은 무엇을 하는 것인지 명확하게 존재
 - 효과성 : 사용되는 모든 명령들은 실행 가능해야 함
 - 유한성 : 사용되는 모든 명령들을 거쳐 반드시 종료

▶ 자연어

- 알고리즘을 표현하기 위한 방법의 한가지
- 일상 생활에서 사용하는 말과 글로 표현
- 이해가 빠른 반면, 내용이 길어질 수 있음

▶ 의사코드(pseudo code)

- 알고리즘을 프로그래밍 언어와 유사한 형태로 풀어 써놓은 것
 - 정형화된 문법적 측면을 배제
 - 사고의 흐름을 간결하고 효과적으로 전달하는 표현
- **특정 프로그래밍 언어와 관계없이 논리의 명시를 위해 작성**
 - 프로그래밍 언어처럼 구체적인 구문을 사용하지 않음
 - 일반적으로 자연어와 유사하게 기술
- **코드를 흉내만 내는 코드로 컴퓨터에서 실행되지 않음**
- **알고리즘을 개략적으로 표현하는데 사용**
 - 알고리즘의 각 단계를 차례대로 작성

▶ 의사코드(pseudo code) 표현 예

- `sum = 0`
 - `sum`이라는 변수에 0을 넣음
- `sum = sum + 1`
 - `sum`이라는 변수에 `sum + 1` 값을 더한 결과값을 넣음
- `if (i == 5) print sum`
 - 만약 `i`가 5와 같으면 `sum`의 값을 출력
 - 같은 의미 : 만약, `i`가 5와 같으면 `sum` 출력

▶ 의사코드(pseudo code) 표현 예

- 삼각형의 면적 구하기
 - 공식 : 면적 = 밑변 × 높이 / 2
- 값이 밑변(bottom) : 10, 높이(height) : 20 일 때
 - 의사코드 표현 방법 1
 - ✓ 밑변에 10 저장(대입)
 - ✓ 높이에 20 저장(대입)
 - ✓ 면적에 밑변(10) × 높이(20) / 2 계산한 결과값 저장(대입)
 - ✓ 면적 값 출력
 - 의사코드 표현 방법 2
 - ✓ bottom = 10
 - ✓ height = 20
 - ✓ area = bottom x height / 2










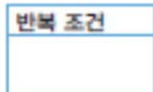


▶ 순서도(flowchart)

- 처리하려는 문제를 분석한 후 처리 순서를 단계화
- 표준 기호로 박스와 연결선으로 표현한 도표(diagram)
 - 프로그래밍 과정에서 프로그램의 설계도 역할
- 순서도를 바탕으로 명확하고 논리적인 프로그램 작성
- 일의 흐름을 나타내기 때문에 '흐름도' 라고도 함
- 컴퓨터의 이용을 전제로 하는 것이 아님

▶ 순서도(flowchart) 종류

- 개략 순서도
 - 전체를 작업 단위로 개괄적으로 작성
- 상세 순서도
 - 실행 가능하도록 줄 단위로 상세하게 작성
 - 상세 순서도를 작성 후 컴퓨터 프로그램 작성

▶ 순서도(flowchart) 기호

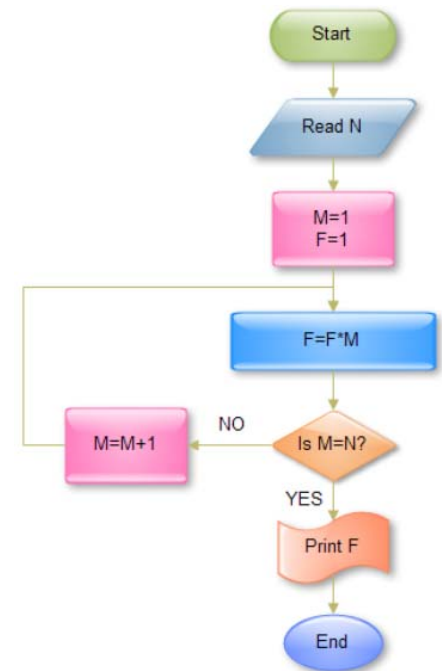
기호	명칭	사용 용도	기호	명칭	사용 용도
	처리	각종 연산, 데이터 이동 등의 처리		터미널	순서도의 시작과 끝 표시
	연결자	흐름이 다른 곳과 연결되는 입출구를 나타냄		천공카드	천공카드의 입출력
	입출력	데이터의 입력과 출력		서류	서류를 매체로 하는 입출력 표시
	흐름선	처리의 흐름과 기호를 연결하는 기능		수동입력	콘솔에 의한 입력
	준비	기억장소, 초기값 등 작업의 준비 과정 나타냄		반복	조건을 만족하면 반복
	미리 정의된 처리	미리 정의된 처리로 옮길 때 사용		디스플레이	결과를 모니터로 나타냄

▶ 순서도(flowchart) 필요한 이유

- 작업의 흐름을 도표화하여 쉽게 이해
- 간단한 도형 형태의 기호를 사용하여 간결한 도식화 가능
- 특정 프로그래밍 언어에 관계없이 알고리즘 표현 가능
- 프로그램 코딩의 기본 자료로서 활용 가능
- 알고리즘의 이해와 추적이 쉬움
- 문제 처리 과정의 논리적 파악으로 에러 검색이 쉬움

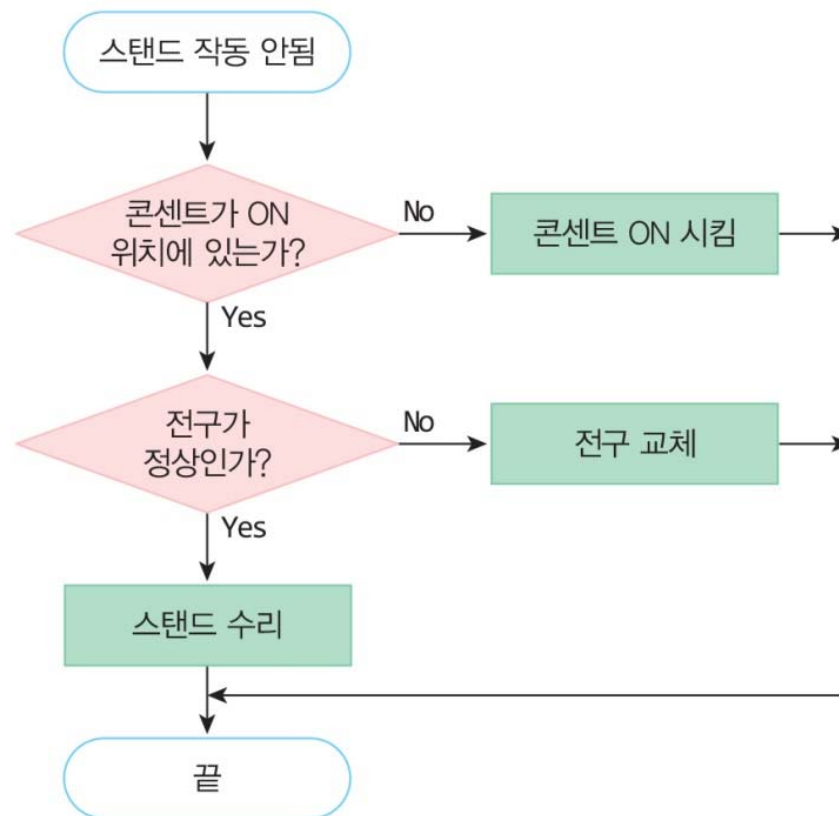
▶ 순서도(flowchart) 작성 규칙

- 전체의 흐름을 명확하게 알아볼 수 있도록 작성
- 표준 기호를 사용하여 간단하고 명료하게 표현
- 기호 내부에 처리할 내용을 간략하게 기입
- 원칙적으로 위에서 아래로, 왼쪽에서 오른쪽 방향
- 과정이 길거나 복잡하면 나누어서 작성, 연결자로 연결




▶ 순서도(flowchart) 작성 예

- 거실에 있는 스탠드를 켜는 순서도
 - 스탠드가 켜져있지 않다는 가정하고 작성



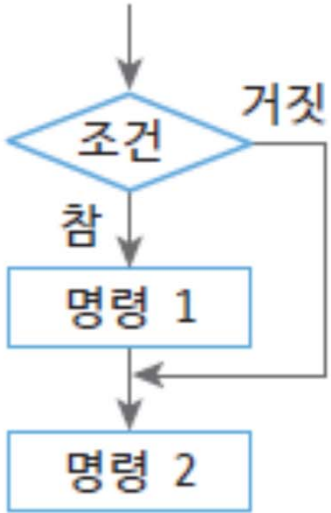
▶ 순서도(flowchart) 3가지 표현 방법

- 순차형(sequence)
 - 동작을 순서에 따라 차례로 실행 (시작부터 종료까지 진행)

순서도	설명
 <pre>graph TD; Start(()) --> Command1[명령 1]; Command1 --> Command2[명령 2]; Command2 --> End(())</pre>	<ul style="list-style-type: none">• “명령 1”을 실행합니다.• 다음으로 “명령 2”를 실행합니다.

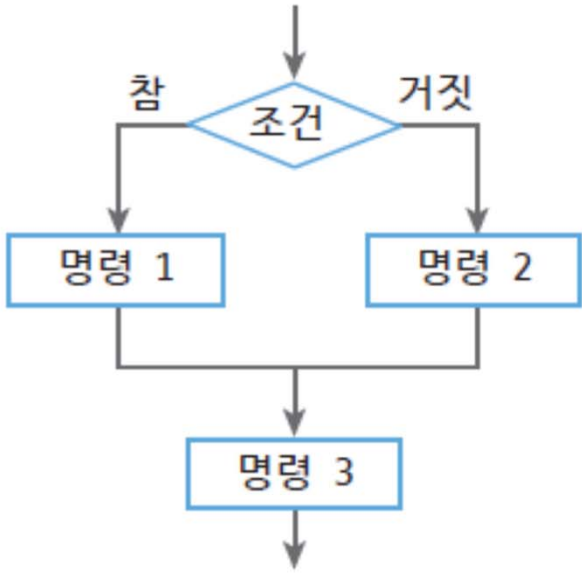
▶ 순서도(flowchart) 3가지 표현 방법

- 선택형(selection)
 - 조건에 맞는 동작 하나를 선택하여 실행

순서도	설명
	<ul style="list-style-type: none">• 조건을 검사합니다.<ul style="list-style-type: none">– 참이면 “명령 1”을 실행합니다.– 거짓이면 “명령 1을” 실행하지 않습니다.• “명령 2”를 실행합니다.

▶ 순서도(flowchart) 3가지 표현 방법

- 선택형(selection)
 - 조건에 맞는 동작 하나를 선택하여 실행

순서도	설명
 <pre>graph TD; Start(()) --> Condition{조건}; Condition -- 참 --> Command1[명령 1]; Condition -- 거짓 --> Command2[명령 2]; Command1 --> Command3[명령 3]; Command2 --> Command3; Command3 --> End(())</pre>	<ul style="list-style-type: none">• 조건을 검사합니다.<ul style="list-style-type: none">– 참이면 “명령 1”을 실행합니다.– 거짓이면 “명령 2”를 실행합니다.• “명령 3”을 실행합니다.

▶ 순서도(flowchart) 3가지 표현 방법

- 반복형(iteration) (= 루프(loop))
 - 정해진 조건에 도달할 때까지 반복적인 동작을 수행

순서도	설명
<pre>graph TD; Entry(()) --> LoopBox[반복 횟수]; LoopBox --> Cmd1[명령 1]; Cmd1 --> Cmd2[명령 2]; Cmd2 --> Cmd3[명령 3];</pre>	<ul style="list-style-type: none">• 반복 횟수만큼 다음 명령들을 반복합니다.<ul style="list-style-type: none">– “명령 1”을 실행합니다.– “명령 2”를 실행합니다.• 반복이 끝나면 “명령 3”을 실행합니다.

▶ 생활 속 사례

- 라면을 조리하기 위한 방법 표현
 - 알고리즘 작성 (라면봉지의 조리법)

1. 물 550cc(3컵 정도)를 끓인다.
2. 면과 분말, 건더기스프를 같이 넣는다.
3. 4~5분 더 끓인다.
4. 그릇에 담아 맛있게 먹는다.

▶ 생활 속 사례

- 라면을 조리하기 위한 방법 표현
 - 알고리즘 표현 - 자연어

물 550cc(3컵 정도)를 끓이다가 면과 분말, 건더기스프를 같이 넣은 후 4~5분 더 끓이다가 다 끓으면 그릇에 담아 맛있게 먹는다.

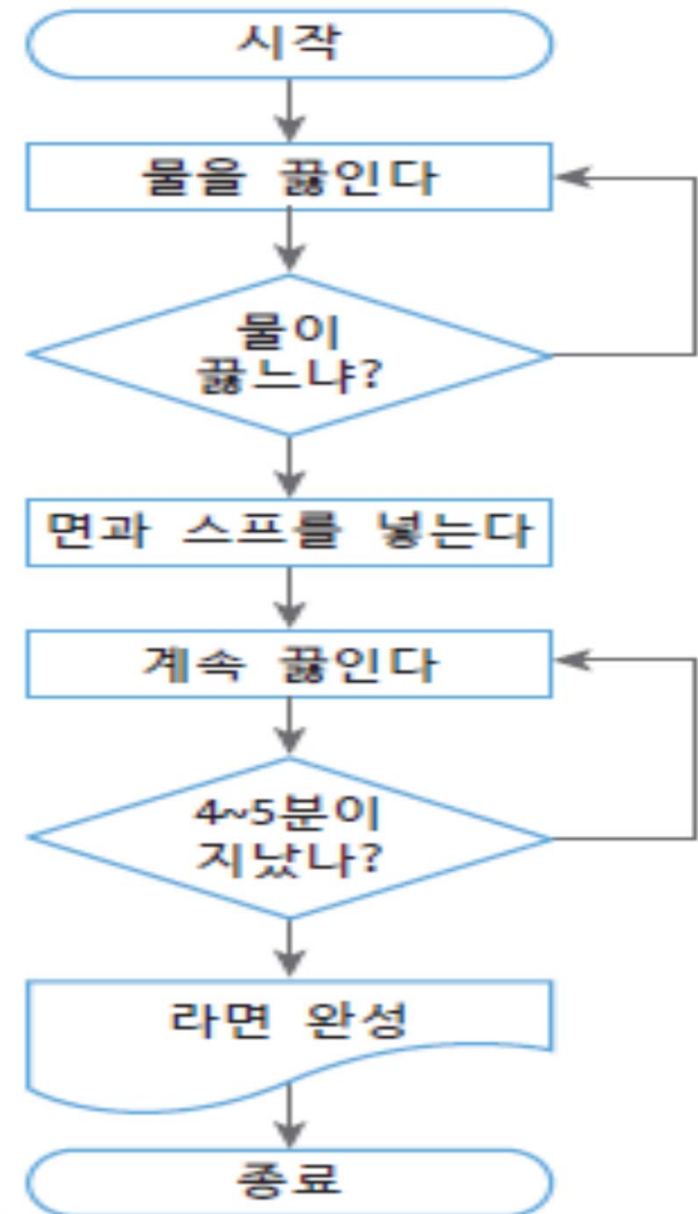
▶ 생활 속 사례

- 라면을 조리하기 위한 방법 표현
 - 알고리즘 표현 - 의사코드

```
do
물 550cc(3컵정도)를 끓인다.
if (면과 스프를 넣고 4~5분 지났나?)
break;
while( 물 != 끓는다 )
```

▶ 생활 속 사례

- 라면을 조리하기 위한 방법 표현
 - 알고리즘 표현 - 순서도



▶ 생활 속 사례

- 라면을 조리하기 위한 방법 표현 정리

컴퓨터 관점	인간 관점
프로그램	라면조리
프로그래머	아이
알고리즘	라면조리법
로직	라면 조리 순서
프로그래밍	라면 조리를 위한 행위
결과	라면 완성

Thank you