



PYTHON

2



Python Programming

- 담당교수 : 최희식
- eMail : choihs3054@seoultech.ac.kr

이번 주 학습

■ 학습목표

- 알고리즘의 개념과 특징을 설명할 수 있다.
- 알고리즘의 구조와 표현 방법을 익혀 다양한 문제에 적용할 수 있다.
- 파이썬 변수에 대해 이해하고 활용할 수 있다.

■ 학습목차

- 알고리즘의 이해
- 알고리즘 표현과 구조
- 변수(variable)

1

이번
차시에서는

- 알고리즘의 이해



컴퓨터 알고리즘 개념과 특징

■ 컴퓨터 알고리즘이란?

- 컴퓨터 알고리즘은 컴퓨터로 해결 가능한 문제를 다루는 부분으로 '현재 위치에서 가장 빠른 길 찾기' 라면 컴퓨터로 해결할 수 있게끔 만들 수 있다.
- 하지만 '착한 사람 찾기'라고 한다면 기준이 모호하기 때문에 프로그램으로는 해결 가능한 문제가 아니므로 컴퓨터 알고리즘을 적용하기에는 적합하지 않다.
- 전기 밥솥의 예를 생각해 보자.



밥의 종류

상태

현미

백미

잡곡밥

보온

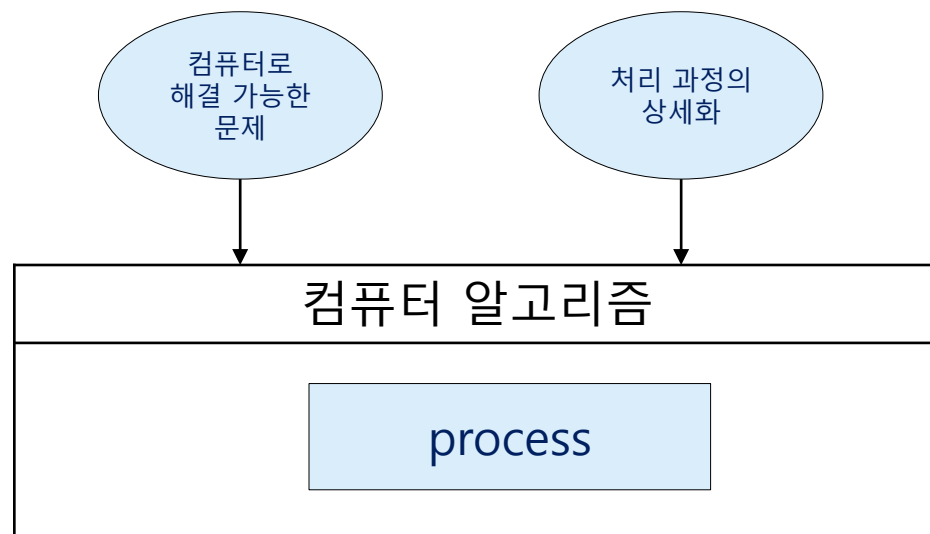
예약

취사

컴퓨터 알고리즘 개념과 특징

■ 컴퓨터 알고리즘이란?

- 현미 버튼을 누르면 현미에 맞게 취사가 된다.
- 밥솥에서 밥을 짓는 과정이지만 이것은 컴퓨터 알고리즘이 아니다.
- 취사를 누르면 밥솥 안에서 어떻게 처리해야 하는지 상세하게 나타내야 한다.
- 현미밥이 되기까지의 모든 과정을 알려주지 않으면 컴퓨터는 이해하지 못하기 때문이다.



컴퓨터 알고리즘 특징

■ 컴퓨터 알고리즘은 다음과 같은 특징이 있다.

- 0개 이상의 입력이 존재한다.
- 1개 이상의 출력이 반드시 존재한다.
- 유한 시간 내에 종료한다.
- 명확한 단계를 거쳐야 한다.
- 효율적인 복잡도를 가져야 한다.(문제를 해결하기 위한 각 과정은 수행 가능해야 한다.)

전기 밥솥을 알고리즘으로 수정

- 앞에서 살펴 보았던 전기밥솥을 입력, 처리, 출력 3단계인 알고리즘으로 표현하면 다음과 같다.
- 입력 : 원하는 밥의 종류를 선택
- 처리 : 눈에 보이지 않지만 프로그램으로 절차화 되어야 하는 내용인 시간, 온도, 압력 등을 적절히 조정해야 밥이 지어지는 과정이므로 처리가 된다.
- 출력 : 원하는 상태의 밥이 만들어지는 것으로 사용자에게 '취사 완료 알림'을 음성으로 들려줌으로 출력이 된다.

입력	처리	출력
원하는 밥의 종류	시간, 온도, 압력 등을 적절히 조정	원하는 상태의 밥이 완료 취사 완료 알림

알고리즘의 표현과 구조

■ 알고리즘의 표현

- 알고리즘은 순서도, 글, 의사 코드로 나타낼 수 있다.
 - 1) 순서도는 기호와 흐름선으로 수행과정을 한눈에 알아보기 쉽고 학습 단계에서 알고리즘의 구조를 학습하는 데 도움이 된다.
 - 복잡하고 대규모 알고리즘을 표현할 때는 적합하지 않다.
 - 2) 글은 우리가 사용하는 언어로 편리한 면은 있지만 모호성이 존재하므로 컴퓨터 알고리즘으로 표현하는 데는 한계가 있다.
 - 3) 의사코드는 코드를 흉내 낸 표현법으로 대규모 알고리즘에 표현할 때 적합하지만 실제 프로그래밍 언어가 아니므로 실행할 수는 없다.

알고리즘의 표현과 구조


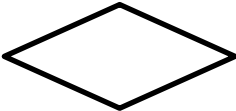




■ 의사 코드로 표현하기

- 의사 코드는 Pseudocode(슈도코드)라고 불리우며 알고리즘을 어떻게 짤지 계획할 때 사용되는 비공식적인 도구이다.
- 의사 코드는 특정 프로그래밍 언어로 표현한 것이 아니고 프로그래밍 언어를 흉내만 냈기 때문에 실제 프로그램 코드 작성하기 전에 주로 사용한다.
- 의사 코드는 대략적인 알고리즘의 구조와 흐름을 파악하기 쉬우므로, 프로그래밍 언어와 인간의 언어가 섞여 있는 형태로 작성되어, 기술적인 부분보다는 프로그램의 흐름을 이해하기 쉽게 표현하는데 사용되므로 실제로 실행할 수는 없다.

의사 코드1	의사 코드2	의사 코드3
두 수를 입력	A, B에 두 수를 입력한다.	input A,B
두 수 더하기	A, B 두 수를 더한다.	$C=A+B$
2의 결과 출력	2의 결과를 출력한다.	output C

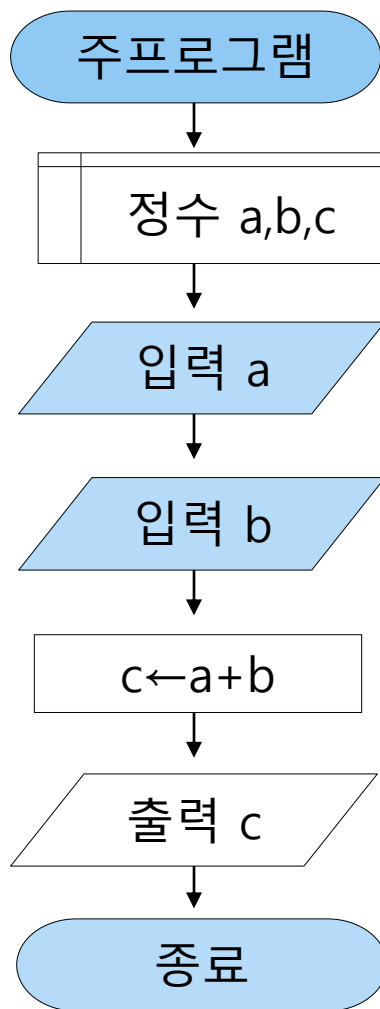
알고리즘의 표현과 구조

■ 순서도 표현하기

종류	설명	종류	설명
	순서도의 시작과 끝		조건에 따라 참과 거짓 판단
	변수의 초기화 선언		입출력되는 데이터를 기술
	처리하여야 할 작업을 명시		순서도 기호와 연결, 실행의 흐름을 표시

알고리즘의 표현과 구조

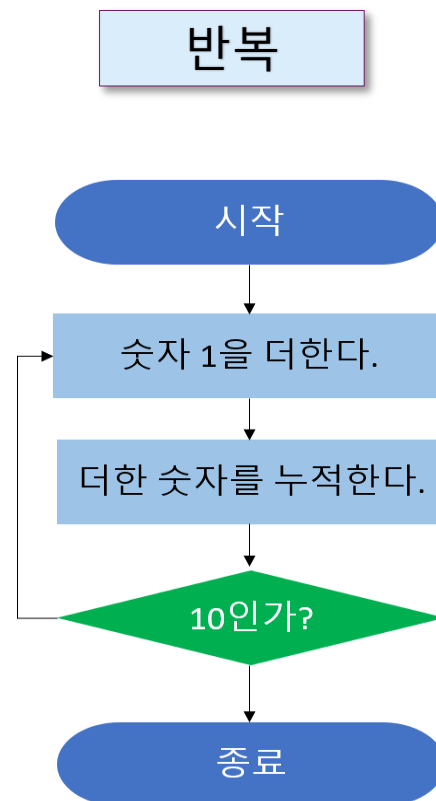
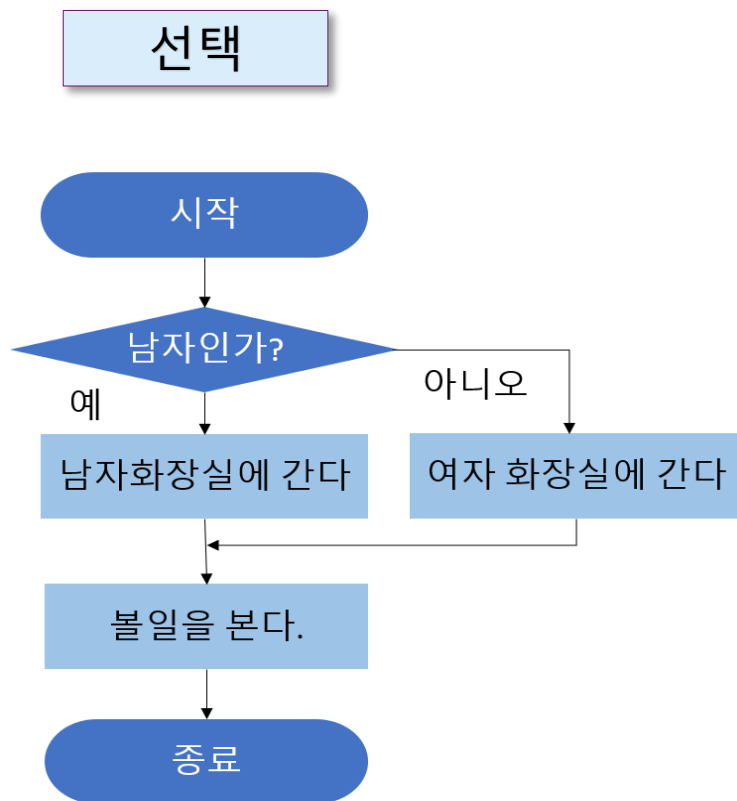
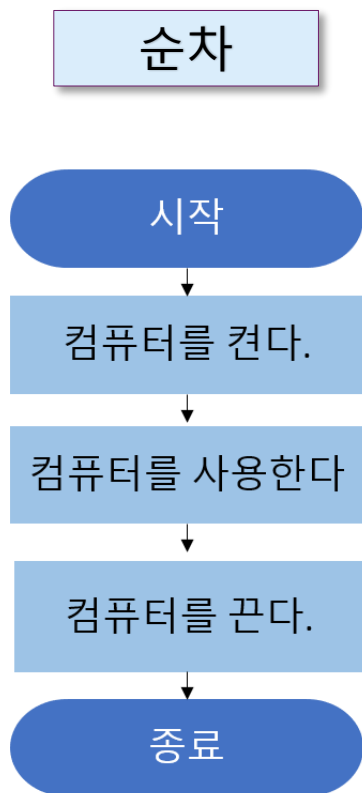
■ 순서도 표현하기



알고리즘의 표현과 구조

■ 알고리즘의 구조

- 알고리즘은 크게 순차 구조, 선택 구조, 반복 구조로 구성된다.



감사합니다.



2

이번
차시에서는

- 알고리즘 절차
- 디버깅
- 알고리즘 작성 실습



알고리즘 절차

- 프로그램 개발 과정은 문제분석→알고리즘설계→코딩→테스팅→유지/보수의 과정을 거칩니다.
- 컴퓨터를 이용해 주어진 문제를 해결하고자 할 때, 무작정 코딩부터 하려고 덤벼드는 것은 결코 좋은 태도가 아닙니다.
- 최근 프로그램 개발전에 요구되는 알고리즘을 통해 문제를 분석하고, 절차를 수립하고, 그에 따른 검증작업을 거치는 과정입니다.
- ⇨ 즉, 알고리즘으로 문제를 분석하고 해결하기 위해, 과정을 찾아가는 것이 일반적이고 정확한 처리 절차라고 볼 수 있습니다.

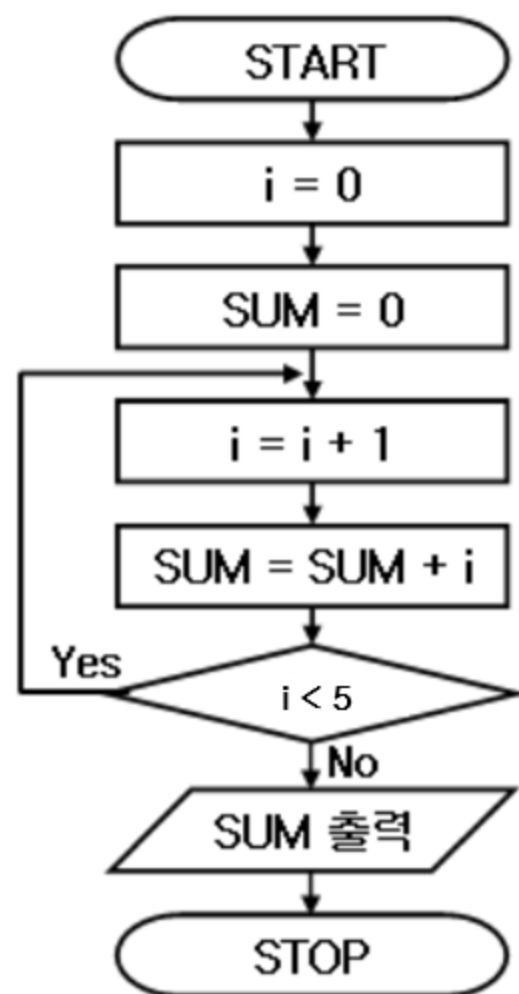
알고리즘 문제 해결 과정

- ① 문제를 명확하게 파악합니다.
- ② 처리조건(사용되는 변수 설명 포함)을 꼼꼼하게 살펴봅니다.
- ③ 알고리즘이 어떤 원리인지를 우선 파악합니다.
- ④ 각 변수들의 특징과, 변수들 간의 관계를 유추합니다.
- ⑤ 처리해야 할 과정을 하나 씩 완성해 나갑니다.
- ⑥ 완성한 처리 과정을 다시 디버깅해 보고,
원하는 결과가 나오지 않으면 다른 처리(계산)방법을 모색하여
해를 찾습니다.
- ⑦ 원하는 결과가 나올 때까지 ⑥번 과정을 반복합니다

디버깅

- 디버깅은 컴퓨터에서 발생한 오류를 찾기 위해 소스 코드를 한 줄 한 줄 따라가면서 변수 값의 변화를 검사하는 것입니다.
- 디버깅을 거치지 않은 순서도는 정확하다고 말할 수 없으므로 디버깅 결과와 계산한 결과 값을 비교하여 반드시 오류를 찾아 문제를 해결하도록 해야 합니다.

디버깅 방법



i	SUM	i < 5	SUM 출력
0	0		
1	1	Yes	
2	3	Yes	
3	6	Yes	
4	10	Yes	
5	15	No	15

$$1 + 2 + 3 + 4 + 5 = 15$$

디버깅 방법

- ① 디버깅 표에 순서도에서 사용된 모든 변수를 쓴다.
- ② 디버깅 표에 순서도에서 사용된 모든 조건식을 쓴다.
- ③ 디버깅 표에 순서도에서 사용된 모든 출력문을 쓴다.
- ④ 디버깅 범위를 줄인다.
- ⑤ 순서도를 따라가며 디버깅을 한다.
- ⑥ 디버깅 결과값과 내가 계산한 결과값을 비교한다.
- ⑦ ⑥의 결과 서로 같으면 종료하고, 다르면 오류를 찾아 수정을 한다.

[실습] 순서도 기본 모형

① 순차 구조(=직선형)

처음 시작부터 마지막 종료 명령까지 단계적으로 진행되는 순서로 수행되는 형태

[실습] $a=10$, $b=20$ 을 할당한 후, 두 변수에 대한 합을 출력하시오.

[실습] 순서도 기본 모형

② 선택 구조(=분기형)

조건에 따라 실행내용이나 순서를 달리하는 형태

[실습] a, b 변수를 선언한 후, a, b 값을 입력 받아서

a가 b보다 크면 a변수를 출력하고,

b가 a보다 크면 b변수를 출력하시오.

[실습] 순서도 기본 모형

③ 반복형

조건을 만족할 때까지 일정한 내용을 반복해서 수행하는 형태

[실습] 변수num을 선언한 후, 1-100까지 출력한 후,
1-100까지 누적인 합도 출력하시오.

[실습] 순서도 기본 모형

변수 num을 선언한 후, 1-100까지 짝수만 출력 하시오.

감사합니다.



3

이번
차시에서는

- 변수



변수

■ 변수(variable)란?

- 값을 저장할 때 사용하는 식별자
- 언제든지 새로운 값에 의해 값이 변할 수 있음

변수이름 = 데이터 값(상수)

■ 변수 식별자 이름 규칙

- ① 파이썬 키워드 식별자는 식별자로 사용할 수 없다.
- ② 영문자는 대소문자를 구별한다. (ex. Hap과 hap은 다른 식별자로 인식)
- ③ 식별자 구성은 영문자, 숫자, 밑줄문자(_)로 이루어진다. 단, 숫자로 시작하면 안 된다.
- ④ 첫 글자는 반드시 영문자나 밑줄(_)로 시작해야 한다.
- ⑤ 식별자 중간에 공백을 포함할 수 없다.
- ⑥ 식별자 생성에 길이 제한은 없다.

변수

■ 사용 가능한 식별자

식별자	설명
name	영문 알파벳 문자로 시작
_number	밑줄 문자로 시작할 수 있음
number_of_pens	중간에 밑줄 문자를 넣을 수 있음
Code3	맨 처음이 아니라면 숫자도 넣을 수 있음

■ 사용 불가능한 식별자

식별자	설명
1st_class	숫자로 시작할 수 없음
my car	중간에 공백이 들어갔음
def	def는 파이썬 키워드 임
num1#	특수기호 #이 사용되었음

[예제] 변수에 데이터를 기억한 후, 출력

```
>>> name="홍길동"
```

```
>>> print("나의 이름은", name, "이다")
```

나의 이름은 홍길동 이다.

[실습] 변수에 이름, 나이를 기억한 후, 출력하기

>>> _____

>>> _____

나의 이름은 홍길동이고,
나이는 20세이다.

변수

■숫자 천 단위 콤마

- 숫자 데이터에 세 자리마다 콤마(,) 단위를 주어 숫자를 쉽게 표현하기 문자열 포매팅format() 함수를 이용한다.

{형식규칙:,.format(값)}

```
>>> name="홍길동"
>>> kibon=2000000
>>> bonus=500000
>>> salary=kibon+bonus
>>> print("{0}씨 월급은 기본급 {1:,.} 보너스 {2:,.}, 총급여액 {3:,.} 이다.".format(name, kibon, bonus, salary))
홍길동씨 월급은 기본급 2,000,000 보너스 500,000, 총급여액 2,500,000 이다.
```

변수

■ 정수형 숫자 자리 형식 지정

- 형식규칙에 콜론(:) 앞에 숫자는 변수에 표시된 출력 순서를 나타낸다.
- *n*은 정수형 전체 출력 자리수를 의미한다.
- d은 정수형 유형 숫자를 의미하며, 실수 숫자 사용시 오류가 난다.

{순서:*nd*}.format(변수)

```
>>> num1, num2, num3=100, 200, 300
```

```
>>> print("{0:d} {1:5d} {2:05d}".format(num1, num2, num3))
```

```
100  200 00300
```


변수

■ 실수형 숫자 자리 형식 지정

- 형식규칙에 콜론(:) 앞에 숫자는 변수에 표시된 출력 순서를 나타낸다.
- 0,의 의미는 정수형 숫자로 3자리마다 ,(콤마)를 부여한다. 이때 0은 생략할 수 있다.
- .숫자f는 실수형 소숫점 자리를 나타낼 때 사용한다.

```
{0:0,.1f}.format(값)
```

```
>>> num1=45000000
```

```
>>> num2=100000.41375
```

```
>>> print("num1 변수: {0:0,.2f} {1:0,.2f}".format(num1, num2))
```

```
num1 변수: 45,000,000.00 100,000.41
```

변수

■ f-string 포맷

- f-string 포맷은 아주 간결하고 출력할 때 편리하게 사용할 수 있는 포매팅이다.
- f-string 포맷을 사용하기 위해서는 print() 함수에 f문자와 {변수}를 사용하면 된다.

f “표시할 문자열 {변수}”

```
>>> name="홍길동"
>>> age=20
>>> print(f"이름은 {name}이고, 나이는 {age} 세이다.")
이름은 홍길동이고, 나이는 20 세이다.
```

변수

■ f-string 포맷

- 변수를 지정할 때 2개 이상의 변수 num1, num2, num3에 하나의 값으로 각각 동일하게 지정할 수 있다.

```
>>> num1=num2=num3=10
```

```
>>> sum=num1+num2+num3
```

```
>>> print(f"{num1}+{num2}+{num3}의 합계는 {sum} 이다.")
```

10+10+10의 합계는 30 이다.

감사합니다.

