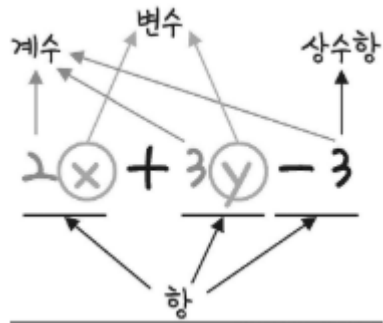


✓ 상수(constant)란


- 변하지 않고, 항상 일정한 값을 갖는 수



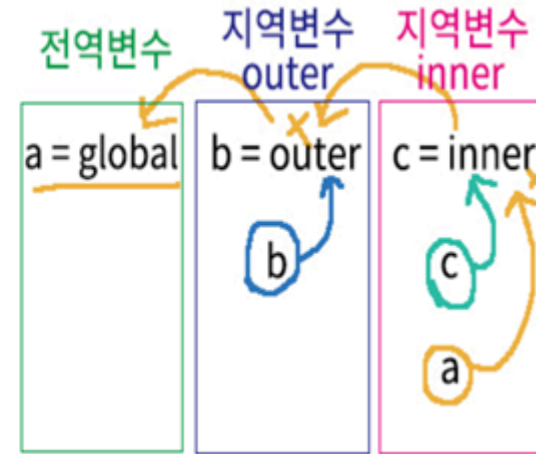
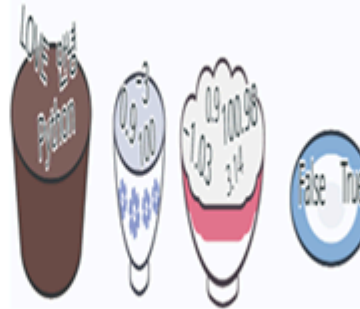
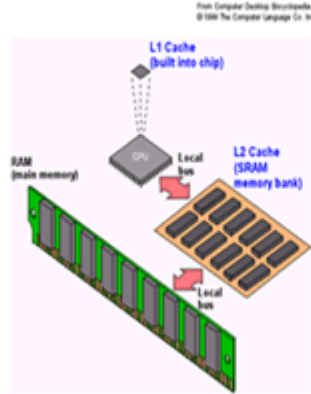
✓ 변수(variable)

'변수'란 어떤 값을 저장하기 위한 메모리 공간을 의미해요. '어떤 값을 담을 수 있는 상자'와 같죠. 이 상자 안에 무엇을 담느냐에 따라서 그 값이 변해요.

'물의 양'을 'water'라는 이름의 변수로 나타내 볼게요.



변수명 : water
변수값 : 12

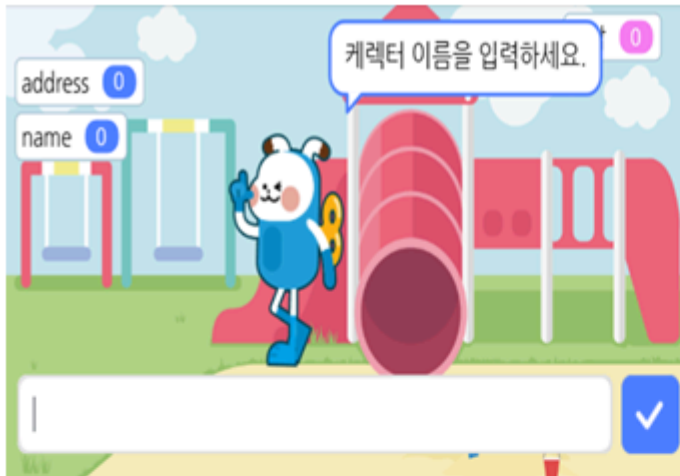


```
1 a=10 #변수 a
2 b=20
3 print(a+b)
```

⇒ 30

✓ 자기소개를 할 수 있는 캐릭터를 엔트리 블록 프로그램으로 만들어 보자!!!

- 우선은 자기소개를 생각해 보면, 자기소개에 필요한 자기 정보가 있어야 하는데, 자기 정보를 저장하기 위해서는 정보를 저장할 수 있는 변수가 필요
- 따라서 간단하게 캐릭터의 이름과 집 주소만을 정보로 입력받아 변수에 값을 저장하고, 저장된 변수의 값을 불러와서 자기소개를 할 수 있도록 만들어 보자



전체	? 변수	복 신호
	리스트	함수

변수 추가하기

변수 이름

name

모든 오브젝트에 사용 이 오브젝트에서 사용

☒ 일반 변수로 사용 (작품에 저장)

☐ 공유 변수로 사용 (서버에 저장)

☐ 실시간 변수로 사용 (서버에 저장)

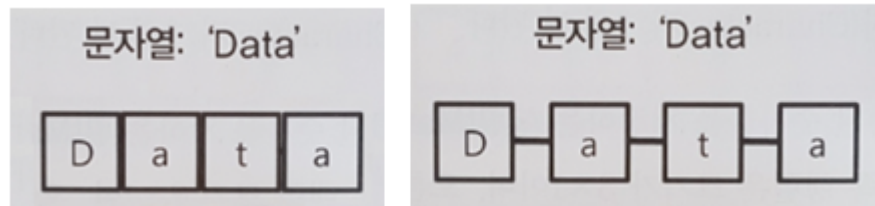
취소 변수 추가



✓ 리스트(list) - 데이터를 리스트하는 방법은 무엇일까요?

- 정보를 저장할 수 있는 변수 이외에 비슷한 정보들을 모아서 목록 형태를 만드는 것

- 여기서 리스트를 생각해 보면, 파일이 모여져 있고, 모여져 있는 목록이 리스트라고 생각할 수 있음
- 출석부, 번호표, 순위표 등등 일정한 정보들의 모임들이 필요한 경우에는 리스트를 만들어 놓고 사용
- 리스트는 일정한 정보들을 모아 놓은 후, 정보들을 꺼내어 쓰는 것을 말하는데, 이러한 리스트가 왜 필요한지 생각해 보자.
 - 우선 여러 가지의 자료를 다룰 때에는 같은 자료로 분류한 후, 모아서 사용하는 것이 편리할 수 있음
 - 예를 들어 회사에 직원이 30명이라고 가정한다면, 직원을 관리하기 위해서는 직원들의 자료가 모아져 있어야 함
 - 이 때 직원 한 명에 대한 정보를 저장하기 위해서는 한 개의 변수가 필요하게 되는데, 직원이 30명인 경우라면, 자료를 저장해야 하는 변수는 30개가 각각 다른 명칭으로 선언되어야 할 것임
 - 하지만 리스트를 사용하게 되면, 직원이라는 리스트 안에 30명의 자료를 넣어서 관리할 수 있고, 순번에 따라 직원의 정보를 불러들일 수 있어 관리가 편리해질 수 있는 장점이 있는 것이 리스트임



```

1 # 리스트 덧셈
2 a=["BlockDMask",333]
3 b=[1,2,3]
4
5 print(a+b)

```

→ ['BlockDMask', 333, 1, 2, 3]

```
1 a=[1,2,3]
2 b=a*3
3 print(b)
4
5 c=a*0
6 print(c)
```

```
⇒ [1, 2, 3, 1, 2, 3, 1, 2, 3]
   []
```

```
1 #리스트인덱싱
2 #길이가8인리스트
3 a=[101,102,'b','l','o',103,104,105]
4
5 #인자접근
6 print(f'a[0]:', a[0])
7 print(f'a[1]:', a[1])
8 print(f'a[2]:',a[2])
9
10 print(f'a[7]:',a[7])
11
12 #print(f'a[8]:') - error
13
14
15 #그럼 마이너스로접근가능할까요?
16 print(f'a[-1]:',a[-1])
17 print(f'a[-2]:',a[-2])
18 print(f'a[-8]:',a[-8])
19
20 #print(f'a[-9]:') - error
```

```
⇒ a[0]: 101
   a[1]: 102
   a[2]: b
   a[7]: 105
   a[-1]: 105
   a[-2]: 104
```

a[-8]: 101

✓ 친구를 소개하는 캐릭터를 엔트리 블록 프로그램으로 만들어 보자!!!

- 우선은 친구를 소개하는 방법을 생각해 보면, 친구를 소개하기 위해서는 친구를 소개할 수 있는 정보가 필요함

◦ 따라서 친구의 여러 가지 정보를 가질 수 있는 변수가 필요하며, 여러 명의 친구가 있음으로 친구들의 정보를 가진 여러 변수들이 필요함

- 하지만 이럴 때 사용하는 것이 리스트임

◦ 따라서 친구들의 이름을 저장할 수 있는 리스트와 친구들의 학교를 저장하는 리스트, 그리고 집 주소를 저장하는 리스트를 만들고, 리스트에 친구들의 정보를 저장하고, 출력하여 친구들을 소개하는 프로그램을 만듦

X: -180.8, Y: 33.9

name school address

대답 0

1번째 친구의 이름을 입력하십시오.

index 1

✓

블록 모양 소리 속성

전체 ? 변수 신호

리스트 함수

리스트 이름

리스트의 이름을 입력해 주세요.

모든 오브젝트에 사용 이 오브젝트에서 사용

☒ 일반 리스트로 사용 (작품에 저장)

☐ 공유 리스트로 사용 (서버에 저장)

☐ 실시간 리스트로 사용 (서버에 저장)

취소 리스트 추가

모든 오브젝트에서 사용되는 리스트 (3)

name

```

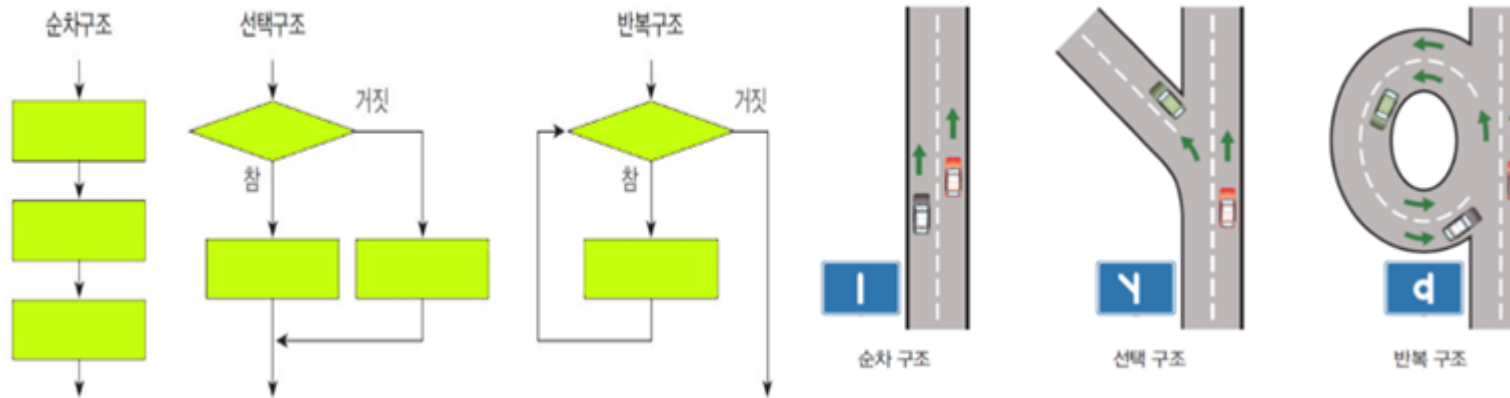
시작하기 버튼을 클릭했을 때
  index 1 을(를) 1 (으)로 정하기
  3 번 반복하기
    index 값 + 번째 친구의 이름을 입력하십시오. 을(를) 묻고 대답 기다리기
    대답 항목을 name 에 추가하기
    index 값 + 번째 친구가 다니는 학교를 입력하십시오. 을(를) 묻고 대답 기다리기
    대답 항목을 school 에 추가하기
    index 값 + 번째 친구의 집 주소를 입력하십시오. 을(를) 묻고 대답 기다리기
    대답 항목을 address 에 추가하기
    index 에 1 만큼 더하기
  
```

```

오브젝트를 클릭했을 때
  index 1 을(를) 1 (으)로 정하기
  3 번 반복하기
    index 값 + 번째 친구의 이름은 + name 의 index 값 번째 항목 + 입니다. 을(를) 2 초 동안 말하기
    index 값 + 번째 친구의 학교는 + name 의 index 값 번째 항목 + 입니다. 을(를) 2 초 동안 말하기
    index 값 + 번째 친구의 집은 + name 의 index 값 번째 항목 + 입니다. 을(를) 2 초 동안 말하기
  index 에 1 만큼 더하기
  
```

✓ 프로그램의 제어 구조

- 순차구조(sequence) - 명령들이 순차적으로 실행되는 구조
- 선택구조(selection) - 둘 중의 하나의 명령을 선택하여 실행되는 구조
- 반복구조(iteration) - 동일한 명령이 반복되면서 실행되는 구조



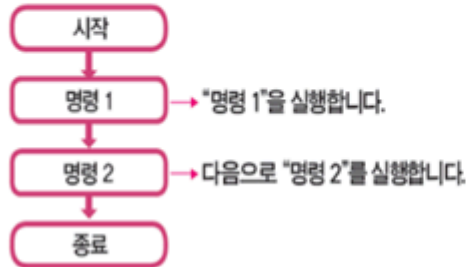


순차 구조란 무엇인가요?

순차 구조란 해야 할 순서대로 동작이나 명령을 나열한 것을 말합니다.



순서도에서는
오른쪽 그림과 같이
순차 구조를 표현하여,
화살표가 가리키는
순서대로 명령을
실행하면 됩니다.



순차구조.py - C:/Users/user/Desktop/파이썬/순차구조.py (3.8.5)

File Edit Format Run Options Window Help

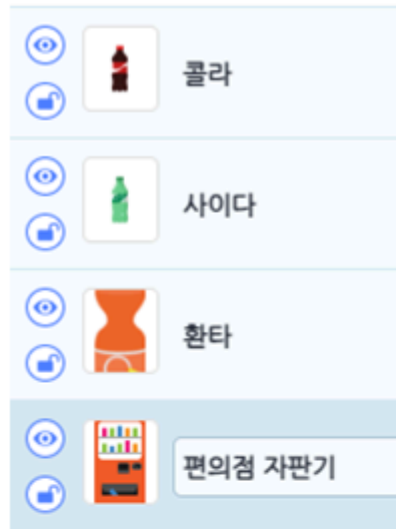
```
print("안녕!")
print("나는 파이썬이야")
print("만나서 반가워")
```

✓ 순차구조 - 자판기를 이용하는 프로그램

- 순차적 구조를 이용하여 자판기를 만들어보자!!!

◦ 자판기에 가서, 돈을 넣고, 원하는 음료수를 선택하면, 음료수가 자판기에서 나옴





```
1 #순차  
2 i = 1  
3 j = 2  
4 sum = i + j  
5 print(sum)
```

 3



반복 구조란 무엇인가요?

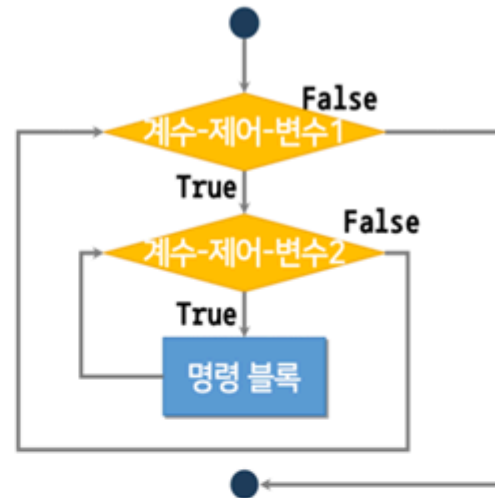
반복 구조란 여러 번 반복해야 하는 같은 동작을 횟수나 조건과 함께 표현한 것을 말합니다.

❶ 반복(횟수)만큼 다음 명령들을 반복합니다.

- "명령 1"을 실행합니다.
 - "명령 2"를 실행합니다.
- 횟수만큼 반복이 끝나면 "명령 3"을 실행합니다.



순서도에서는 왼쪽 그림과 같이 반복(횟수) 구조를 표현하며, 나타낸 반복 횟수만큼 명령(명령 1, 명령 2)을 실행합니다.



10 번 반복하기

계속 반복하기

참 이 될 때까지 반복하기

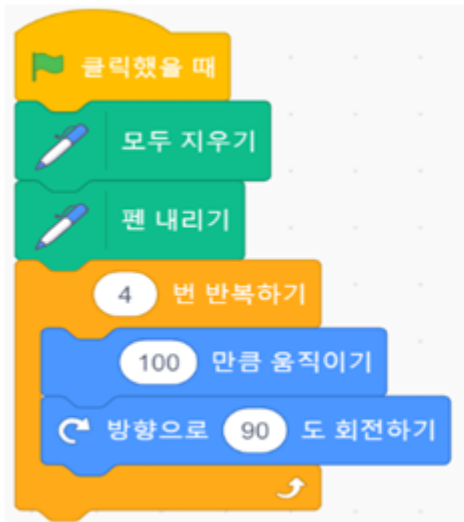
반복 중단하기

입력한 횟수만큼 감싸고 있는 블록들을 반복 실행합니다.

감싸고 있는 블록들을 계속해서 반복 실행합니다.

판단값에 따라 블록의 반복여부를 결정합니다.
이 될 때까지: 판단이 참이 될 때까지 감싸고 있는 블록들을 반복 실행합니다.
인 동안: 판단이 참인 동안 감싸고 있는 블록들을 반복 실행합니다.

이 블록을 감싸는 가장 가까운 반복 블록의 반복을 중단합니다.



```
반복횟수구조.py - C:/Users/user/Desktop/파이썬/반복횟수-  
File Edit Format Run Options Window Help  
import turtle as t  
  
for x in range(4):  
    t.forward(100)  
    t.right(90)
```

✓ 반복구조를 이용하여 사각형 그려보기



▽ ~될 때까지의 의미

sum 11

list1
1 1

