

<프로그래밍의 3대 요소>

1. 변수(Variable) : 데이터를 저장하는 기억공간
2. 자료형(DataType) : 변수에 저장된 자료의 종류
int, float, str, bool, complex : 기본자료형
list, tuple, set, dict : 군집자료형
3. 할당, 대입(=) : 변수에 데이터를 저장(대입)하는 것
 - * Python에서 모든 자료형은 (객체)로 취급한다.
 - * Python에서 모든 자료형은 (class)로 만들어졌다.

<Python에서 객체생성과정>

1. 생성(만들)하고자 하는 객체를 설계 해야 한다
 - 직접 설계를 할 수도 있다.(무엇으로 class 로)
 - 이미 설계된 class를 활용 할 수도 있다.
2. 생성자 함수를 이용해서 객체를 생성한다. -> **`__init(self)`**
 - 생성자 함수 : 메모리에 객체를 생성하는 함수, 객체를 초기화를 할 수 있다.
특징(기본적으로 생략되어 있다.)
3. 생성된 객체를 조작하기 위해서는 객체가 가지고 있는 함수를 이용하면 된다.

A. 정수 10이라는 객체를 생성과정(제공되는 class를 활용)

1. 생성(만들)하고자 하는 객체를 설계 해야 한다

class **int** :-> class : 객체를 설계하는 도구, 모델링 도구
int(자료형, type) -> 기본자료형
멤버변수
생성자 함수(객체를 생성하는 함수, `__init__(self)` 함수)
일반 함수

2. 생성자 함수를 이용해서 객체를 생성한다.

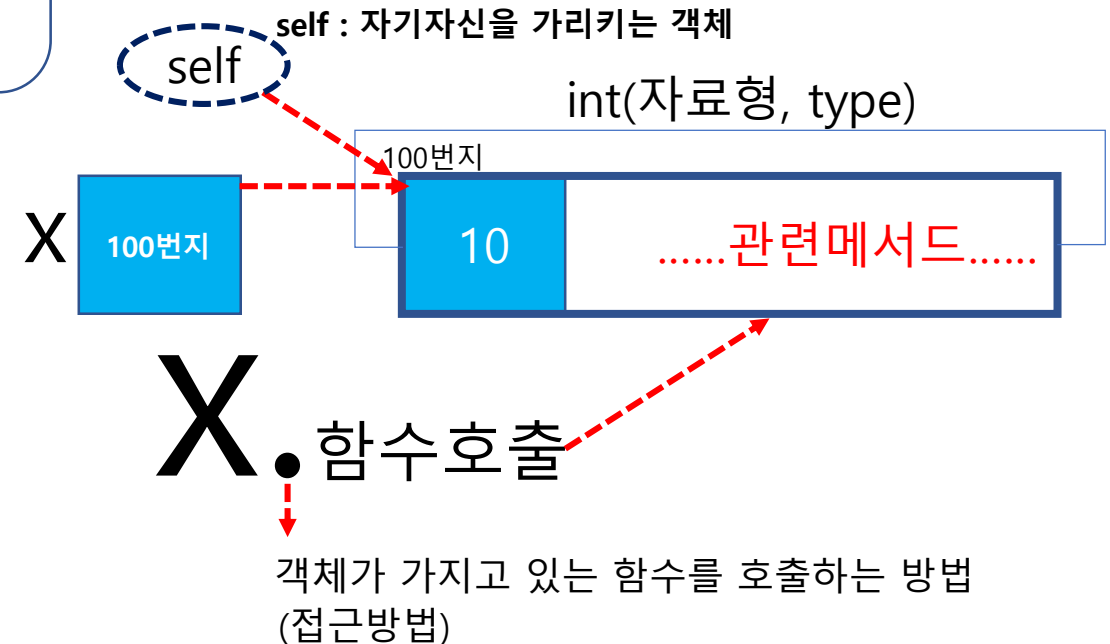
x=int(10) or x=10

3. 생성된 객체를 조작하기 위해서는 객체가 가지고 있는 함수를 이용하면 된다.

X ● 함수호출(Ctrl + Space Bar)

```
*Python 3.7.3 Shell*
File Edit Shell Debug Optio
Python 3.7.3 (v3
l)] on win32
Type "help", "co
>>> x=int(10)
>>> type(x)
<class 'int'>
>>> x.bit_length|
```

x=int(10)



B. 볼(컬러, 스피드)객체를 생성과정(제공되지 않는 class를 활용, 직접 만들어야 한다.)

```
class Ball:
```

```
    color=""  
    speed=0
```

```
    def setSpeed(self, value):  
        self.speed=value
```

class : 객체를 설계하는 도구, 모델링 도구
Ball(자료형, type) -> 사용자정의자료형

.7.3 (v3.7.3:ef4ec6ed12, Mar 25 2019, 21:26:53)

C:/Users/2017-학생-

076/AppData/Local/Programs/Python/Python37-
32/ballClass.py

Ball01 Color: Red

Ball01 Color: 10

Ball02 Color: Blue

Ball02 Color: 20

>>>

결과

```
ball01=Ball();
```

```
ball01.color="Red"
```

```
ball01.setSpeed(10)
```

```
ball02=Ball();
```

```
ball02.color="Blue"
```

```
ball02.setSpeed(20)
```

```
print("Ball01 Color: %s" %ball01.color)
```

```
print("Ball01 Color: %s" %ball01.speed)
```

```
print("Ball02 Color: %s" %ball02.color)
```

```
print("Ball02 Color: %s" %ball02.speed)
```

```
ball01=Ball();
```

self

Ball(자료형, type)

ball01

100번지

100번지

""

0

setSpeed(self, value)

color

speed

ball01.

↓ 객체가 가지고 있는 멤버들을 접근하는 방법
(멤버종류> 멤버 변수, 멤버 함수)

C. 객체를 생성할 때 사용되는 함수(생성자 함수)

```
class Ball:  
    color=""  
    speed=0
```

```
def __init__(self):  
    print("생성자 함수가 호출")
```

```
def setSpeed(self, value):  
    self.speed=value
```

```
ball01=Ball();  
ball01.color="Red"  
ball01.setSpeed(10)
```

객체를 생성하기 위해서 생성자 함수를 호출하는 부분

`__init__(self)` : 생성자 함수

- 객체를 생성할 때 가장 먼저 호출되는 함수
- 객체의 초기화를 담당
- 객체의 속성을 정의 할 때도 사용한다.



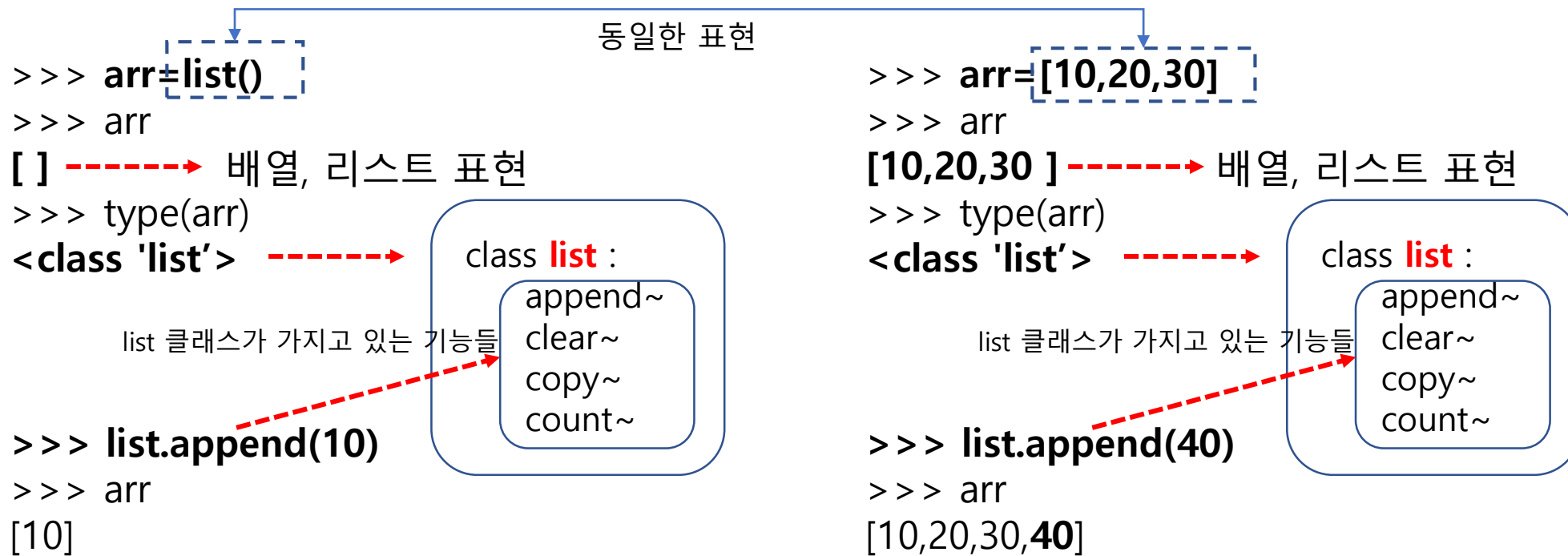
```
class Ball:  
    color=""  
    speed=0
```

```
def __init__(self,color, speed):  
    self. color=color  
    self.speed=speed
```

```
def setSpeed(self, value):  
    self.speed=value
```

```
ball02=Ball("Blue", 20);
```

D. 심화 학습(군집 데이터 Array & List)



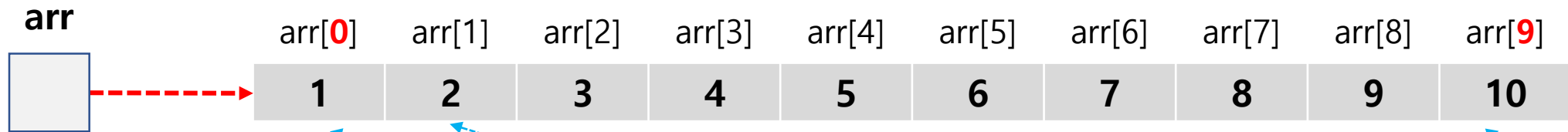
문제1. list 에 1~10까지의 수를 저장하고 내림차순으로 정렬하여 출력 하세요.

[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]  [10, 9, 8, 7, 6, 5, 4, 3, 2, 1]

문제2. list에 ['H', 'e', 'l', 'l', 'o', 'p', 'y', 't', 'h', 'o', 'n'] 를 저장하고 'o' 의 개수를 구하여 출력 하세요.

A. list 군집자료형의 표현 과 반복 문(for)의 이해

Q) arr=[1,2,3,4,5,6,7,8,9,10]



간단한 반복 문 for

for **i** in **군집자료** :
실행 해야 될 문장

```
for i in arr:  
    print(i)
```

← 10번 실행됨

1
2
3
4
5
6
7
8
9
10

Q. 동일한 명령을 4번 반복해서 실행해봅시다. : range(범위)함수 활용하기

```
for i in range(4):  
    print("CS101 is fantastic!")
```



들여쓰기를 잊지 마세요!

```
for i in range(4):  
    print("CS101 is great!")  
    print("I love programming!")
```

클래스(class) : 1.새로운 데이터 타입을 만드는 도구, 모델링 도구
2.현실 세계에 있는 사물을(객체를) 컴퓨터 프로그래밍으로 표현하기 위하여 그 객체의 상태정보(멤버변수)와 행위정보(함수,메서드)를 추출 하여 설계하는 도구

클래스(class)의 구성 = 멤버변수+ 멤버 메서드(함수) +생성자 메서드(함수)

Q. 자동차 객체를 정의(자동차 클래스를 설계해보자)

```
class Car:
```

```
# 생성자함수(객체생성+객체의 초기화)
```

```
def __init__(self, name, color) :
```

```
    self.name=name
```

```
    self.color=color
```

```
# 함수(메서드)
```

```
def show_info(self):
```

```
    print("이름:" ,self.name, "/ 색상:", self.color)
```

```
car1=Car("소나타", "빨간색") # 객체생성하기(인스턴스 만들기) , car2=Car("BMW", "흰색")
```

```
car1.show_info() --> 이름: 소나타 / 색상: 빨간색
```

```
car1=Car("소나타", "빨간색")
```

