

IPA 주관 인공지능센터 기본(fundamental) 과정

- GitHub link: [here](#)
- E-Mail: windkyle7@gmail.com

Python Type and Identifier Part. 2

호모지니어스(Homogeneous) 와 헤테로지니어스(Heterogeneous)

호모지니어스는 동질의, 헤테로지니어스는 서로 다른 종류들로 이루어진 이라는 의미를 가지고 있다.

즉, 문자열은 모두 동일한 타입인 문자로 구성된 sequence 타입이므로 호모지니어스라고 할 수 있으며, 리스트와 튜플은 서로 다른 타입을 가질 수 있으므로 헤테로지니어스라고 할 수 있다.

문자열

str 역시 sequence이기 때문에 인덱싱과 슬라이싱이 가능하다.

```
In [1]: a = '안녕하세요'
```

```
In [2]: a[:2]
```

```
Out[2]: '안녕'
```

```
In [3]: a[0]
```

```
Out[3]: '안'
```

문자열은 서로 비교할 때 가장 앞에 있는 문자의 크기순으로 비교한다.

```
In [4]: 'ABC' > 'abc'
```

```
Out[4]: False
```

```
In [5]: 'z1Ab' > 'a3Bc'
```

```
Out[5]: True
```

```
In [6]: 'z' > 'Z'
```

```
Out[6]: True
```

```
In [7]: 'z' > 'A'
```

```
Out[7]: True
```

```
In [8]: 'z' > 'Abcd'
```

```
Out[8]: True
```

멤버십 연산자 in

멤버십 연산자는 컨테이너 안에 해당 요소가 있으면 True, 없으면 False를 반환한다.

```
In [9]: a = 'Hello'
```

```
In [10]: 'H' in a
```

```
Out[10]: True
```

```
In [11]: 'b' in a
```

```
Out[11]: False
```

```
In [12]: b = [1, 2, 3, 4, 5]
```

```
In [13]: 3 in b
```

```
Out[13]: True
```

```
In [14]: 10 in b
```

```
Out[14]: False
```

```
In [15]: 'H' in b
```

```
Out[15]: False
```

```
In [16]: c = ['1', '2', '3']
```

```
In [17]: 1 in c
```

```
Out[17]: False
```

```
In [18]: '1' in c
```

```
Out[18]: True
```

```
In [19]: ['1'] in c
```

```
Out[19]: False
```

```
In [20]: d = [['4', '6'], '1', 3, '2']
```

```
In [21]: ['4'] in d
```

```
Out[21]: False
```

```
In [22]: ['4', '6'] in d
```

```
Out[22]: True
```

```
In [23]: 3 in d
```

```
Out[23]: True
```

비교 연산자 `is`

파이썬에서는 논리 연산자 외에 `is` 연산자를 사용하여 비교할 수 있다. 그러나 `is` 연산자는 값(value) 비교가 아닌 래퍼런스(reference) 비교를 한다는 점을 명심해야 한다.

```
In [24]: 1 is 3
```

```
Out[24]: False
```

```
In [25]: a = '1'
b = 1
```

```
In [26]: a is b
```

```
Out[26]: False
```

```
In [27]: a = 10
b = 10
```

```
In [28]: a is b
```

```
Out[28]: True
```

```
In [29]: id(a)
```

```
Out[29]: 10910688
```

```
In [30]: id(b)
```

```
Out[30]: 10910688
```

```
In [31]: a = -5
b = -5
```

```
In [32]: a is b
```

```
Out[32]: True
```

```
In [33]: id(a), id(b)
```

```
Out[33]: (10910208, 10910208)
```

```
In [34]: a = -6
b = -6
```

```
In [35]: a is b
```

```
Out[35]: False
```

```
In [36]: id(a), id(b)
```

```
Out[36]: (140032041441392, 140032041441136)
```

```
In [37]: a = 256
b = 256
```

```
In [38]: a is b
```

```
Out[38]: True
```

```
In [39]: id(a), id(b)
```

```
Out[39]: (10918560, 10918560)
```

```
In [40]: a = 257
```

```
b = 257
```

```
In [41]: a is b
```

```
Out[41]: False
```

```
In [42]: id(a), id(b)
```

```
Out[42]: (140032041441488, 140032041441712)
```

위의 결과에서 살펴볼 수 있듯, 파이썬에서는 -5부터 256까지의 범위의 값은 메모리에 캐시하여 동일한 주소를 참조하도록 하고있다.