



# 변수와 데이터타입

소프트웨어개발센터 박경규

# 목차

- 변수(Variable)
- 변수 코딩실습
- 숫자(Number)
- 숫자형 코딩실습
- 문자열(String)
- 문자열 코딩실습

# 학습목표

- 프로그래밍에서 중요한 변수(Variable)에 대해 이해하고 설명할 수 있다.
- 파이썬의 정수, 실수, 문자열 데이터타입의 특징 및 사용법을 학습합니다.

# 변수(Variable)

- 파이썬에서는 정수, 실수, 문자열, 데이터 구조 등 모든 것이 객체(Object)로 구현되어 있습니다.
- 객체는 데이터가 담긴 박스 또는 컨테이너와 같습니다.
- 변수(Variable)는 데이터가 담긴 객체에 이름(symbolic name)을 붙이는 것입니다.
- 변수에 값을 할당하기 위해 =을 사용합니다.  
=는 '같다'라는 의미가 아니고 할당(assign)한다는 의미입니다.
- 예로, 파이썬 코드 `x = 7` 은  
정수객체 7을 생성하고,  
생성된 객체의 참조를 변수 x에 할당하는 것입니다.

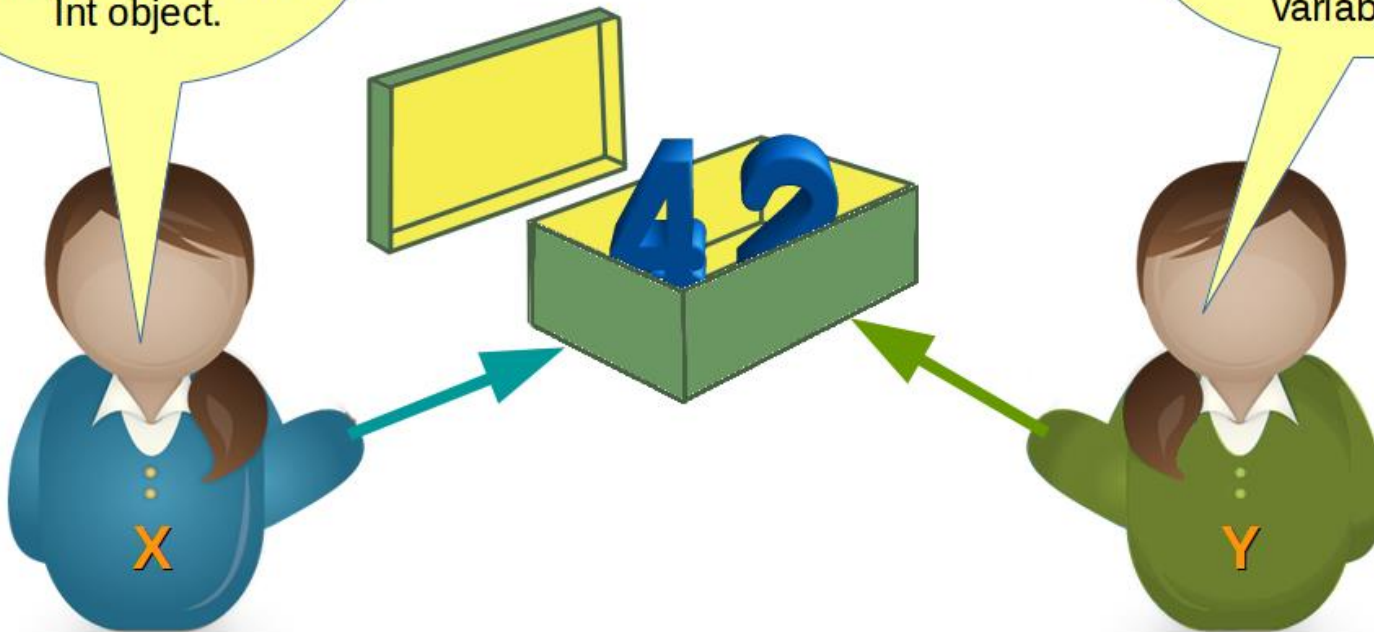
# 변수(Variable)

```
x = 42
```

```
y = x
```

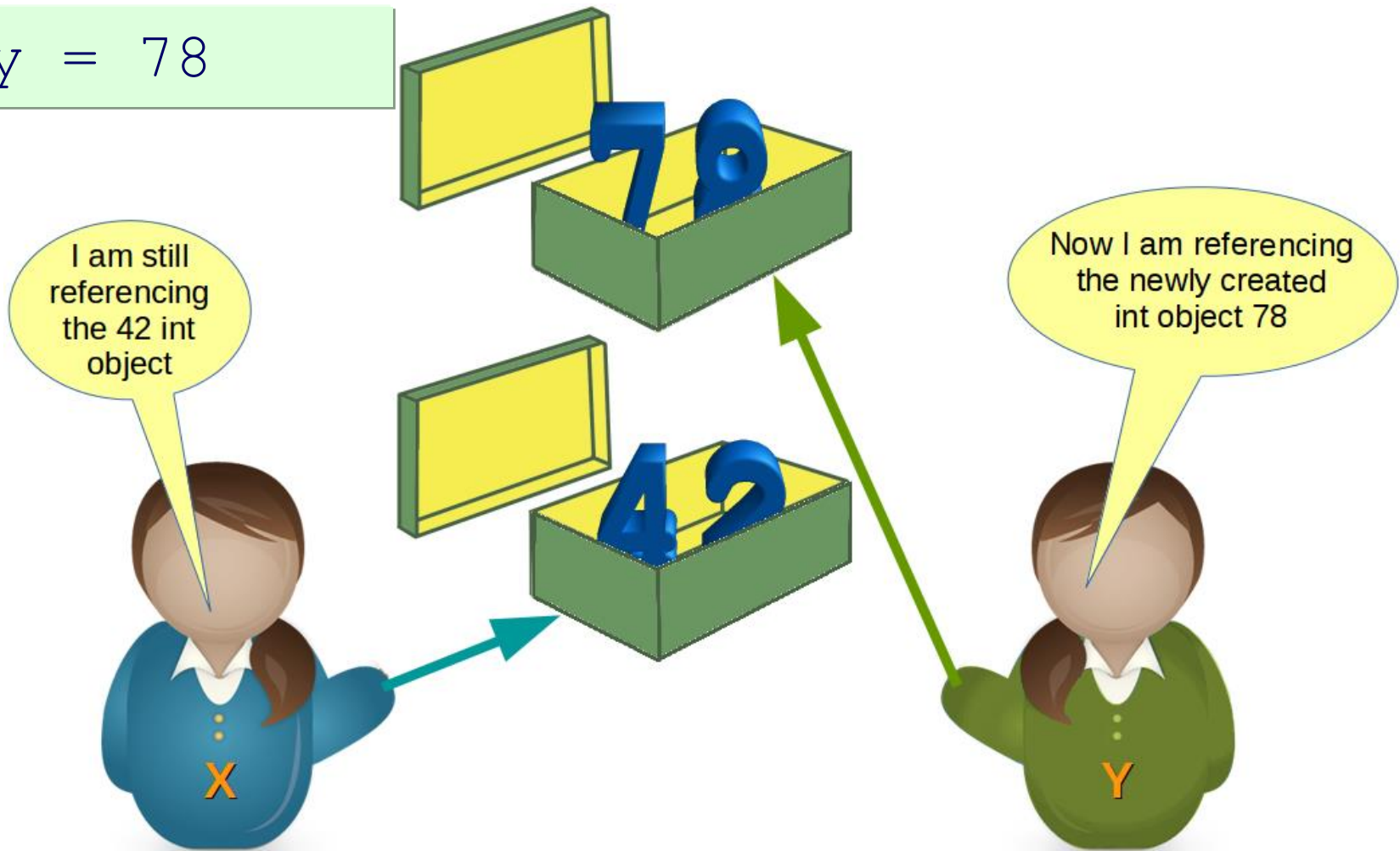
I am a Python variable. My name is X and I can point to an arbitrary object. In this case to an Int object.

My name is Y and I reference the same object as the variable X.



# 변수(Variable)

```
y = 78
```



# 변수(Variable)

```
x = "Text"
```

Now, I am  
referencing  
a string  
object!



I am still  
referencing the  
integer object 78

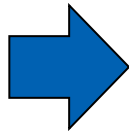


# 변수 코딩 실습

- 아나콘다 Spyder를 실행합니다.  
윈도우 시작메뉴 > Anaconda3 > Spyder
- IPython 콘솔에서 실습코드를 한 라인씩 입력하세요.
- print() 함수로 변수에 할당한 값을 출력해서 확인하세요.

## 실습코드

```
x = 123  
y = x  
print(x)  
print(y)
```



```
IPython console  
Console 1/A  
In [1]: x = 123  
In [2]: y = x  
In [3]: print(x)  
123  
In [4]: print(y)  
123
```



# 변수 코딩 실습

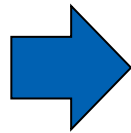
- id() 함수로 변수의 ID 를 확인할 수 있습니다.

## 실습코드

```
id(x)  
id(y)
```

```
y = 0.456  
print(y)  
id(y)
```

```
x = "파이썬 코딩"  
print(x)  
id(x)
```



```
In [5]: id(x)  
Out[5]: 140705996471872  
  
In [6]: id(y)  
Out[6]: 140705996471872  
  
In [7]: y = 0.456  
  
In [8]: print(y)  
0.456  
  
In [9]: id(y)  
Out[9]: 1704768216424  
  
In [10]: x = "파이썬 코딩"  
  
In [11]: print(x)  
파이썬 코딩  
  
In [12]: id(x)  
Out[12]: 1704771155424
```

# 숫자(Number)

- 파이썬 기본 데이터타입에 숫자형과 문자열 데이터타입이 있습니다.
- 숫자형에는 정수형(Integer)과 부동 소수점 숫자형(Float)이 있습니다.
- 정수(int) : 123, -123, +123
- 부동소수점수(float) : 3.14, 1.2e5
- 숫자형 데이터타입 연산자

연산자(Operator)	설명	예(a = 10, b = 5)
+	덧셈 연산자	$a + b = 15$
-	뺄셈 연산자	$a - b = 5$
*	곱셈 연산자	$a * b = 50$
/	나눗셈 연산자	$b / a = 0.5$
%	나머지 연산자	$b \% a = 0$
**	거듭제곱 연산자	$a ** b = 100000$

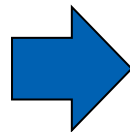
# 숫자형 코딩 실습

- IPython 콘솔에서 실습코드를 한 라인씩 입력하세요.
- `type()` 함수로 데이터타입을 확인하세요.

## 실습코드

```
i = 123  
print(i)  
type(i)
```

```
f = 3.14  
print(f)  
type(f)
```



```
IPython console  
Console 1/A  
In [1]: i = 123  
In [2]: print(i)  
123  
In [3]: type(i)  
Out[3]: int  
In [4]: f = 3.14  
In [5]: print(f)  
3.14  
In [6]: type(f)  
Out[6]: float
```

# 문자열(String)

- 텍스트 처리는 프로그램의 주요 작업이며, 대표적 예가 검색엔진입니다.
- 일반적으로 프로그래머는 숫자 계산 처리보다는 문자열 처리를 많이 하고 있으며, 수학적인 능력보다 논리적인 생각이 중요합니다.
- String은 문자들의 시퀀스(Sequence of Character)로 문자열입니다.
- 작은 따옴표(' ')를 이용하여 문자열을 지정합니다. 예)'Python is powerful'  
모든 공백문자(space), 탭(tab)은 입력한 그대로 유지됩니다.
- 큰 따옴표(" ")로 둘러싸인 문자열은 작은 따옴표로 둘러싸인 문자열과 완전히 동일하게 취급됩니다. 예)"Python is powerful"
- 따옴표 세 개('' ''')로 여러 줄에 걸친 문자열을 표현합니다.

예) '''This is the first line.

This is the second line.'''

# 문자열(String)

- 데이터 타입 변환 : `str()` 함수를 사용하여 데이터타입을 문자열로 변환  
`str(12.3) → '12.3'`
- 문자열 결합 : `+` 연산자를 사용하여 문자열 또는 문자열 변수를 결합  
`'Python is ' + 'powerful' → 'Python is powerful'`  
`a = 'Hello '`  
`b = 'Danny.'`  
`a + b → 'Hello Danny.'`
- 문자열 길이 : `len()`  
`s = 'Python is powerful'`  
`len(s) → 18`

# 문자열(String)

- 문자 추출 : 문자열 이름 뒤에 대괄호([])와 오프셋을 지정

`s[0]` → **'P'**

- 문자열 슬라이스 : 문자열 이름 뒤에 대괄호와 시작 오프셋(start offset), 끝 오프셋(end offset)을 지정

`s[0:6]` → **'Python'**

`s[-18: -12]` → **'Python'**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
P	y	t	h	o	n		i	s		p	o	w	e	r	f	u	l
-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

# 문자열(String)

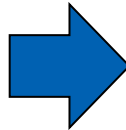
- 문자열 나누기 : `str.split([sep])`  
`fruits = 'Apple Banana Orange'`  
`fruits.split(' ') → ['Apple', 'Banana', 'Orange']`
- 문자열 바꾸기 : `str.replace(old, new')`  
`fruits.replace(' ', ', ') → 'Apple, Banana, Orange'`
- 문자열 찾기 : `str.find(search_str)`  
`fruits.find('Banana') → 6`  
`fruits.find('Lemon') → -1`
- 문자열 대소문자 변환 : `str.lower()`, `str.upper()`  
`fruits.lower() → 'apple banana orange'`  
`fruits.upper() → 'APPLE BANANA ORANGE'`

# 문자열 코딩 실습

- IPython 콘솔에서 실습코드를 한 라인씩 입력하세요.
- print() 함수로 문자열 출력을 확인하세요.

## 실습코드

```
s = 'Python is powerful'
print(s)
print(s[0])
print(s[0:6])
print(s[-18: -12])
```



```
IPython console
Console 1/A x

In [1]: s = 'Python is powerful'

In [2]: print(s)
Python is powerful

In [3]: print(s[0])
P

In [4]: print(s[0:6])
Python

In [5]: print(s[-18: -12])
Python
```

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
P	y	t	h	o	n		i	s		p	o	w	e	r	f	u	l
-18	-17	-16	-15	-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1



문자열

코딩실습

### 실습코드

fruits = 'Apple Banana Orange'

fruits.split(' ')

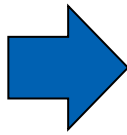
fruits.replace(' ', ', ')

fruits.find('Banana')

fruits.find('Lemon')

fruits.lower()

fruits.upper()



```
IPython console
Console 1/A x

In [2]: fruits.split(' ')
Out[2]: ['Apple', 'Banana', 'Orange']

In [3]: fruits.replace(' ', ', ')
Out[3]: 'Apple, Banana, Orange'

In [4]: fruits.find('Banana')
Out[4]: 6

In [5]: fruits.find('Lemon')
Out[5]: -1

In [6]: fruits.lower()
Out[6]: 'apple banana orange'

In [7]: fruits.upper()
Out[7]: 'APPLE BANANA ORANGE'

In [8]:
```



[danny.park@kt.com](mailto:danny.park@kt.com)  
[kgpark88@gmail.com](mailto:kgpark88@gmail.com)