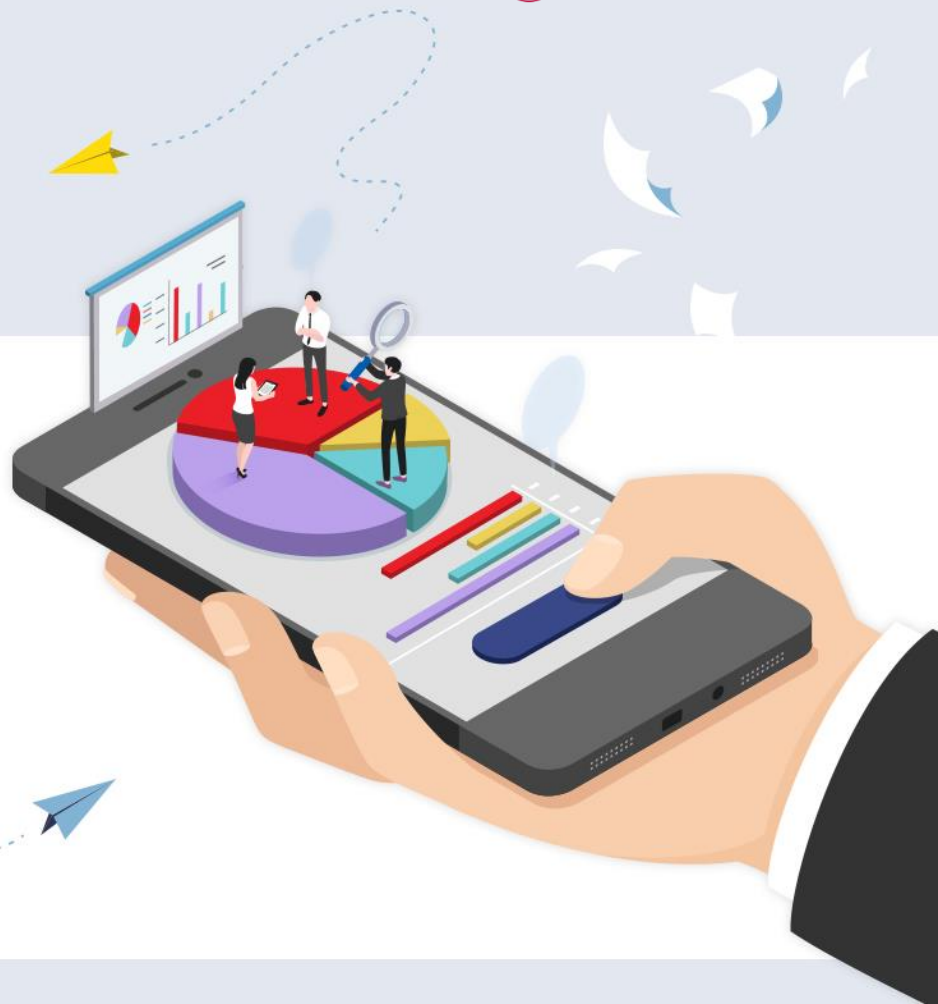




01

# 변수에 대한 이야기



변수  
(Variable)

쉽게 변하는 수



프로그램이 동작하면서 어떤 상황 혹은  
상태에 따라 변화하는 어떤 자료(데이터)를  
담아 두기 위해 사용하는 개념

“ 변수명 = 값 ”

- 특정 값을 저장하고
- 기억하는 목적

✓ 변수에는 숫자형(정수, 실수), 문자형(문자열형), 불, 리스트, 딕셔너리 등 거의 대부분의 무언가 할당 가능

✓ 컴퓨터에서 값을 저장하는 기억장치(메모리) 공간으로 변수를 사용하면 효율적인 프로그램을 작성 가능

## 특징

일시적으로 자료를 저장하는 공간임

변수에 저장된 값은 변할 수 있음

변수에는 숫자, 문자열 등 모든 자료형을  
저장할 수 있음

변수에는 다른 변수의 값도 저장할 수 있음

변수는 사용되기 전에 반드시 할당되어 있어야 함

## 예 변수 생성과 변수 사용

1

```
>>> x=100  
>>> print(x)  
100
```

```
>>> name='홍길동'  
>>> addr='서울시'  
>>> print(name,addr)  
홍길동 서울시
```

3

2

```
>>> x=100  
>>> x=200  
>>> print(x)  
200
```

```
>>> age=20  
>>> age=age+10  
>>> print('10년 후 나이', age)  
10년 후 나이 30
```

4



## 예 변수 생성과 변수 사용



파이썬에서 값이 할당되지 않은 변수를 사용하면  
오류가 발생함

```
>>> x=100
```

```
>>> name='홍길동'
```

```
>>> addr='서울시'
```

```
>>> print(name,addr)
```

```
>>> a=a+1
```

Traceback (most recent call last):

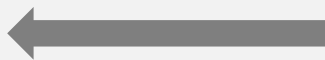
File "<pyshell#8>", line 1, in <module>

```
a=a+1
```

**NameError: name 'a' is not defined**

## 예 변수 생성과 변수 사용

할당 (Assignment)



**name = 'Kim'**

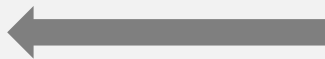
변수이름

값

- 파이썬에서의 변수
  - 값이 할당되는 순간 생성
- **프로그래밍** 언어에서 등호(=)
  - 오른쪽 값을 왼쪽에 저장(할당)한다는 의미
  - 왼쪽은 저장 공간을 의미
- **프로그램**에서 등호(=)
  - 대입연산자

## 예 변수 생성과 변수 사용

할당 (Assignment)



**name = 'Kim'**

변수이름

값

print() 함수

- 프로그래밍에서 함수 : 어떤 특별한 기능을 하도록 만들어진 코드의 묶음
- (...)안에 주어진 값들을 출력하는 기능



### 3 변수를 사용하는 이유



- ☑ 프로그램에서 데이터를 변수에 저장해놓으면  
필요할 때마다 꺼내 사용할 수 있음



## 예 변수를 사용하지 않은 소스 코드

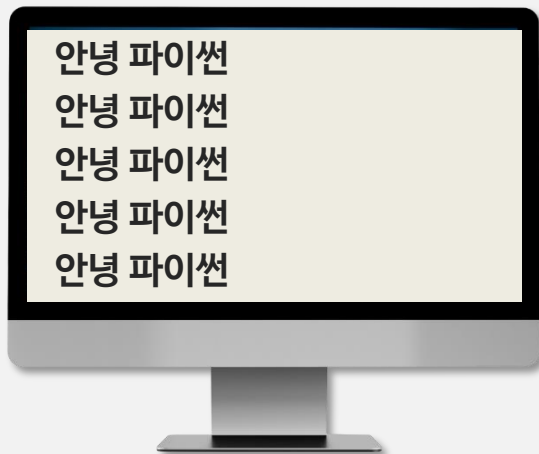
```
print('Hello python')  
print('Hello python')  
print('Hello python')  
print('Hello python')  
print('Hello python')
```

# Q

1. 영어로 된 문장을 한글로 바꾸고 싶다면?
2. 소스코드가 5줄부터 훨씬 더 많아진다면 ?

## 예 변수를 사용한 소스 코드

```
hello='안녕 파이썬'  
print(hello)  
print(hello)  
print(hello)  
print(hello)  
print(hello)
```



# A

변수는 데이터를 보관하고 변경이 가능하며,  
반복적으로 사용하는 데이터는 반드시 변수에  
저장되어야 함



02

## 변수 생성 및 규칙



1

영문자, 숫자, 언더바(\_)로 구성될 수 있으며  
첫 글자는 반드시 영문자/언더바(\_)로 시작함



언더바(\_)를 제외한 특수문자는 사용할 수 없음

예

- num(O), \_num(O), num1(O)
- 1num(X), num#(X)

## 2

공백이 들어가면 안 됨

예

- student\_num(O)
- student num(X)

## 3

대문자와 소문자를 구별함



Name과 name은 다른 변수!

## 4

파이썬에서 다른 용도로 사용되는 예약어는  
변수명으로 사용할 수 없음

예

```
>>> import keyword  
>>> print(keyword.kwlist)  
['False', 'None', 'True', 'and', 'as', 'assert', 'async',  
'await', 'break', 'class', 'continue', 'def', 'del', 'elif',  
'else', 'except', 'finally', 'for', 'from', 'global', 'if',  
'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or',  
'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

## 2 유효한 변수명과 유효하지 않은 변수명



### 유효한 변수명

age

\_count # 언더바(\_) 문자로 시작 가능

num3 # 맨 처음이 아니라면 숫자도 포함 가능

stu\_no # 언더바(\_) 문자가 중간 포함 가능

### 유효하지 않은 변수명



## 2 유효한 변수명과 유효하지 않은 변수명



### 유효한 변수명

2name # 숫자로 시작할 수 없음  
student no # 공백을 포함 할 수 없음  
money@ # @와 같은 특수문자 사용 안 됨  
if # 예약어 안 됨

### 유효하지 않은 변수명

Tip!

### 좋은 변수명 원칙

### 유효하지 않은 변수명

- 짧게 줄이는 것보다는 **의미 전달이 잘 되게** 하는 것이 좋음

예 전화번호를 저장하기 위한 변수로 pn보다는  
phone\_num가 좋음

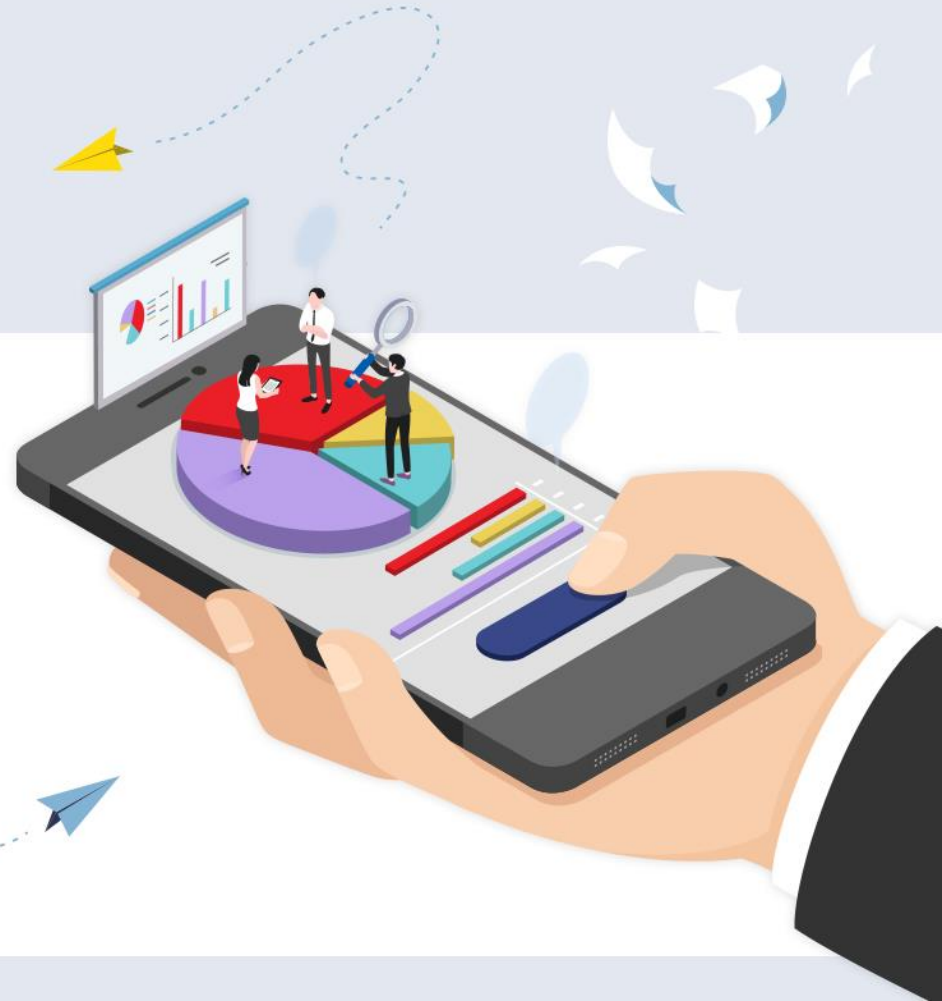
- 변수명은 **간단명료**한 것이 좋음
- 변수명은 전체 프로그램에서 **하나의 일관된 규칙으로**  
**작성**하는 것이 좋음

예 phoneNum와 phone\_num를 혼용해서 쓰지 않으며,  
파이썬은 **소문자와 언더바가 섞여있는 형식**을  
주로 사용함



03

# 다양한 타입의 변수



“ 변수에 들어 있는 자료의 값에 따라 ”  
변수의 형태가 결정됨

✓ 파이썬은 다른 프로그래밍 언어와는 달리 값을 할당하면  
그때 타입이 결정됨

✓ `type(변수이름)`을 통해 변수의 타입을 알아볼 수 있음

## 예 다양한 타입의 변수

```
>>> num=100  
>>> type(num)  
<class 'int'>
```

```
>>> name='홍길동'  
>>> type(name)  
<class 'str'>
```

정수형

문자열형

실수형

불형

```
>>> num=95.5  
>>> type(num)  
<class 'float'>
```

```
>>> result=True  
>>> type(result)  
<class 'bool'>
```



## 예 다양한 타입의 변수

기본 자료형의 종류

00

>>> name='홍길동'

자료형	의미	예시
int	integer, 정수	n=100
float	float, 부동 소수점	n=95.5
str	string, 문자열	n='Kim' n="Kim"
bool	boolean	n=True n=False



# 04 연산자





더하기



빼기



곱하기



나누기



나머지



제곱



몫





〈1/2〉

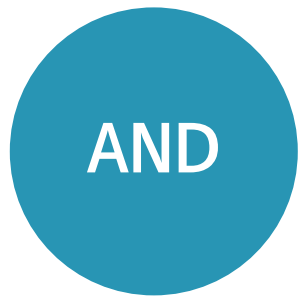
연산자	의미	예시
=	왼쪽변수에 오른쪽 값을 할당	$a=10$
+=	왼쪽변수에 오른쪽 값을 더하고 결과를 왼쪽변수에 할당	$a+=10$ ( $a=a+10$ )
-=	왼쪽변수에 오른쪽 값을 빼고 결과를 왼쪽변수에 할당	$a-=10$ ( $a=a-10$ )
*=	왼쪽변수에 오른쪽 값을 곱하고 결과를 왼쪽변수에 할당	$a*=10$ ( $a=a*10$ )
/=	왼쪽변수에 오른쪽 값을 나누고 결과를 왼쪽변수에 할당	$a/=10$ ( $a=a/10$ )

〈2/2〉

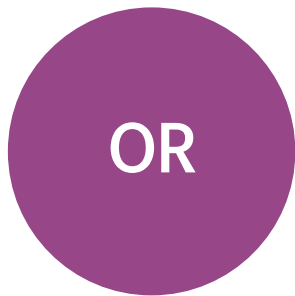
연산자	의미	예시
<code>%=</code>	왼쪽변수에 오른쪽 값을 나눈 나머지의 결과를 왼쪽변수에 할당	<code>a%=10</code> ( <code>a=a%10</code> )
<code>**=</code>	왼쪽변수에 오른쪽 값만큼 제곱을 하고 결과를 왼쪽변수에 할당	<code>a**=2</code> ( <code>a=a**2</code> )
<code>//=</code>	왼쪽 변수에서 오른쪽 값을 나눈 몫의 결과를 왼쪽변수에 할당	<code>a//=10</code> ( <code>a=a//10</code> )

연산자	의미
==	값이 동일하다
!=	값이 동일하지 않다
>	왼쪽 값이 오른쪽 값보다 크다
>=	왼쪽 값이 오른쪽 값보다 크거나 동일하다
<	왼쪽 값이 오른쪽 값보다 작다
<=	왼쪽 값이 오른쪽 값보다 작거나 동일하다

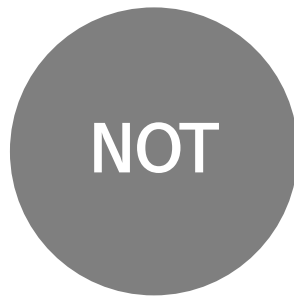
논리  연산  




둘 다 참일 때만 참



둘 중 하나만  
참이어도 참



논리 상태를  
반전

예

```
>>> a=10
```

```
>>> b=60
```

```
>>> a<50 and b>50 # 비교연산자와 논리연산자 함께 사용
```

```
True
```

```
>>> a>50 or b<70
```

```
True
```

```
>>> not a>100
```

```
True
```

## 문자열 더하기 +

```
>>> head='python'  
>>> tail=' is fun!'  
>>> head+tail  
'python is fun!'
```

## 문자열 곱하기 \*

```
>>> w='python '  
>>> w*5  
'python python python python python '
```

- 일반적으로 사용하는 숫자 곱하기의 의미와는 다름
- 문자열의 반복을 의미함
  - $w*5 = w$ 를 5번 반복하라!

## 문자열 곱하기 \*

```
>>> w='python '
```

```
>>> w*5
```

```
'python python python python python '
```

\*의 의미



# | 수고 하셨습니다.

본 강의 저작물은 저작권법 제25조 수업목적 저작물 이용 보상금제도에 의거하여  
한국복제전송저작권협회와 약정을 체결하고 적법하게 이용하고 있습니다.  
약정범위를 초과하는 사용은 저작권법에 저촉될 수 있으므로  
저작물의 재복제 및 수업 목적 외의 사용을 금지합니다.