



# 파이썬기초

2주차 - 1교시  
변수의 이해





## 학습내용

- 변수의 이해
- 파이썬의 표준 입출력



## 학습목표

- 변수를 이해하고 변수선언 및 변수에 값을 저장할 수 있다.
- 파이썬의 표준 입출력을 이용해서 데이터의 출력과 입력을 할 수 있다.



## 생각해 봅시다

프로그램 내부에서 데이터를 저장하고  
필요할 때 사용할 수 있는 방법에 대해  
생각해봅시다.



01



# 변수의 이해



## 1 | 변수란?

### 01 변수(Variable)

#### 변수

데이터를 저장할 메모리공간을 생성하고 이름을  
정해두어 데이터를 저장하고 필요한 시점에 이름을  
이용해 **메모리에 접근할 수 있도록 하는 것**

→ 변수는 어떠한 값을 저장하는 메모리 공간

## 1 | 변수란?

### 01 변수(Variable)

#### 📖 메모리(Memory)

변수를 저장하는 **물리적 장소**이자  
변수에 값이 저장되는 공간

#### 메모리 주소

변수의 저장 위치로, 변수에 들어가는 값은  
반드시 어떤 특정한 메모리 주소를 갖게 됨



## 1 | 변수란?

### 02 변수의 이해

- **변수의 이름(변수명)**은 결국 메모리의 특정 위치에 이름을 붙이는 것
- 프로그램이 실행되는 동안 사용하기 위해 확보한 **메모리의 특정 저장공간**
  - 변수라는 명칭은 저장공간의 저장 값이 변경될 수 있다는 것을 의미함



- 프로그램에서 다양한 종류의 데이터를 표현하기 위해 변수를 사용

## 1 | 변수란?

### 03 변수에 데이터 저장

- 파이썬에서는 **대입 연산자(=)**를 사용하여 값을 저장
- 대입 연산자 우측에는 숫자나 문자 등 메모리에 저장하려는 값을 직접 적거나 수식을 사용
  - 대입 연산자 **좌측에는 변수만** 나와야 함





## 1 | 변수란?

### 04 파이썬에서의 변수

- 파이썬에서 변수의 데이터 형식은 값을 대입하는 순간마다 변경될 수 있는 **유연한 구조**



#### 대입 연산자의 왼쪽

: 무조건 **변수**만 올 수 있음



#### 대입 연산자의 오른쪽

: **무엇이든** 올 수 있음

→ 값, 변수, 수식, 함수 등

## 1 | 변수란?

### 04 파이썬에서의 변수

```
num = 10
```

→ 10 데이터를 메모리공간에 저장하고,  
10이 저장된 메모리공간의 주소를  
num이라고 하는 메모리 공간에 저장한다.



## 1 | 변수란?

### 05 변수의 명명 규칙

- 문자(A-z)와 숫자(0-9), \_(underscore)를 사용할 수 있음
  - ✓ `data = 0, _a1 = 2, _name = 'king'`
- 변수명은 대소문자가 구분됨
  - ✓ `Num`과 `num`은 다른 변수임
- 변수명은 의미 있는 단어로 사용하는 것이 좋음
  - ✓ `age, student_name, student_number`

## 1 | 변수란?

### 05 변수의 명명 규칙

- 변수명이 숫자로 시작하면 할 수 없음
  - ✓ 1num
- 변수명에는 공백이 포함될 수 없음
  - ✓ student name
- 예약어를 변수명으로 사용할 수 없음
  - ✓ for, if, True 등

## 1 | 변수란?

### 06 파이썬의 예약어

'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'



## 1 | 변수란?

### 07 변수의 활용

- 변수에 있던 기존 값은 없어지고 **새로 입력한 값**으로 변경됨

```
In [8]: num = 10
```

```
Out [8]: 10
```

```
In [9]: num = 100
```

```
In [10]: num
```

```
Out [10]: 100
```

## 1 | 변수란?

### 07 변수의 활용

- 변수에는 **계산 결과**를 넣을 수도 있음

```
In [11]: num1 = 10 + 100
```

```
In [12]: num1
```

```
Out [12]: 110
```

## 1 | 변수란?

### 07 변수의 활용

- 변수에는 **숫자와 변수의 연산**을 넣을 수도 있음

```
In [13]: num2 = num + 100
```

```
In [14]: num2
```

```
Out [14]: 200
```



02



# 파이썬의 표준 입출력



### 1 | 표준 입력 함수

#### 01 input() 함수

- 파이썬에서는 콘솔 창에서 입력을 받기 위해 표준 입력 함수인 input() 함수 사용
  - input() 함수의 반환 값은 항상 **문자열**임

### 1 | 표준 입력 함수

#### 02 input() 함수 활용 예제 1

In [\*]: name = input('이름을 입력하세요')

이름을 입력하세요. 손흥민

- 입력 폼이 나타나고 입력을 기다림
- 입력 후 엔터키를 누를 때 까지 기다림
- In [\*]는 현재 셀이 실행 중임을 나타냄
- 문자열 입력 후 변수 name을 출력

### 1 | 표준 입력 함수

#### 02 input() 함수 활용 예제 1

```
In [18]: name = input('이름을 입력하세요')
```

이름을 입력하세요손흥민

```
In [17]: name
```

```
Out [17]: '손흥민'
```

### 1 | 표준 입력 함수

#### 03 input() 함수 활용 예제 2

```
In [*]: num3 = input('숫자를 입력하세요')
```

숫자를 입력하세요. 100

- 100을 입력 후 엔터
- num3에 10을 더하는 숫자 연산을 실행

### 1 | 표준 입력 함수

#### 03 input() 함수 활용 예제 2

```
In [21]: num3 = input('숫자를 입력하세요')
```

숫자를 입력하세요100

```
In [22]: num3 + 10
```

-----  
-----  
TypeError

Traceback (most recent call

last)

Cell In[22], line 1

----> 1 num3 + 10

TypeError: can only concatenate str (not "int") to str

### 1 | 표준 입력 함수

#### 03 input() 함수 활용 예제 2

- input( ) 함수는 **문자열만을 반환**하기 때문에  
사용자가 숫자를 입력하더라도 숫자연산이 될 수 없음



숫자 연산이 필요하면  
형변환 함수를 이용해야 함

### 2 | 표준 출력 함수

#### 01 print() 함수

##### print() 함수를 이용한 데이터의 출력

print() 함수 안에 있는 콤마(,)를 이용해  
여러 값을 연결해 화면에 출력 가능

- ‘Hello’, ‘Python’ : 사이에 한 칸 띄고 화면에 출력됨  
→ “Hello Python“



문자형 간에 + 기호를 이용해 문자열을  
연결할 수 있음



### 2 | 표준 출력 함수

#### 02 print() 함수의 서식

서식	값의 예	설명
%d, %x, %o	10, 100, 1234	정수(10진수, 16진수, 8진수)
%f	0.5, 1.0, 3.14	실수(소수점이 붙은 수)
%c	"b", "한"	한 글자
%s	"안녕", "hello", "a"	두 글자 이상인 문자열

### 2 | 표준 출력 함수

#### 03 함수의 서식 예제

```
In [26]: print('%d'%10)
          print('%f'%3.14)
          print('%c'% 'A')
          print('%s'% '안녕하세요')
```

10

3.140000

A

안녕하세요

### 3 | 이스케이프 문자

#### 01 이스케이프 문자

이스케이프 문자	역할	설명
<code>\n</code>	새로운 줄로 이동	Enter를 누른 효과
<code>\t</code>	다음 탭으로 이동	Tab을 누른 효과
<code>\b</code>	뒤로 한 칸 이동	백스페이스를 누른 효과
<code>\\</code>	<code>\</code> 출력	
<code>\'</code>	<code>'</code> 출력	
<code>\*</code>	<code>*</code> 출력	

### 3 | 이스케이프 문자

#### 02 이스케이프 문자 사용 예제

```
In [49]: print("안녕하세요. 손흥민 입니다.")
          print("안녕하세요. Wn손흥민 입니다.")
          print("안녕하세요. Wt손흥민 입니다.")
          print("안녕하세요. Wb손흥민 입니다.")
          print("안녕하세요. WW손흥민 입니다.")
          print("안녕하세요. W"손흥민W" 입니다.")
          print("안녕하세요. W'손흥민W' 입니다.")
```

#### 결과(Console)

```
안녕하세요. 손흥민 입니다.
안녕하세요.
손흥민 입니다.
안녕하세요.  손흥민 입니다.
안녕하세요손흥민 입니다.
안녕하세요. W손흥민 입니다.
안녕하세요. "손흥민" 입니다.
안녕하세요. '손흥민' 입니다.
```

### 4 | format ( ) 함수

#### 01 {}를 사용해 함께 서식 지정

- 문자열을 출력할 때 **서식 지정자**를 이용하여 출력
- 문자열.format( ) 함수의 인자 값을 “**중괄호{}**”에 지정한 위치에 순서대로 값이 대입되어 출력

### 4 | format ( ) 함수

#### 02 예제 1

```
In [46]: print('{0}. 제목1'.format(123))  
print('{0} : {1} : {2}'.format(10, 20, 30))  
print('{0} : {2} : {1}'.format(10, 20, 30))  
print('{0:s} : {1:f} : {2:c}'.format('10', 10, 65))  
print('{0:s} : {1:f} : {2}'.format('10', 0.20, 'A'))  
print('이름은 {name}이고, 나이는 {age}살 입니다.  
      '.format(name="손흥민", age=20))
```

### 4 | format ( ) 함수

#### 02 예제 1

##### 결과(Console)

```
123. 제목1
10 : 20 : 30
10 : 30 : 20
10 : 10.000000 : A
10 : 0.200000 : A
이름은 손흥민이고, 나이는 20살 입니다.
```

### 4 | format ( ) 함수

#### 03 주석

```
In [54]: name = '박지성' # 이름을 저장하는 변수  
         age = 30        # 나이를 저장하는 변수
```





# 파이썬기초

NEXT

파이썬의 자료형 - 1

