

파이썬 기본 문법

변수

- 변수는 데이터를 저장하기 위한 저장 공간의 이름
- 프로그램 내에서 데이터를 저장, 수정, 재사용하기 위해 사용

변수 선언

- 파이썬에서는 변수를 선언할 때 특별한 키워드가 필요 없으며, 값을 할당하는 순간 변수가 생성
- 파이썬은 변수의 데이터 타입을 자동으로 인식하며, 데이터 타입은 변수에 할당된 값에 따라 결정
- 할당 연산자(=)를 사용하여 변수에 값을 할당

Python

 복사

```
number = 10 name = "Alice" print(number) print(name)
```

데이터 타입

- 기본 타입: 정수(int), 부동소수점(float), 문자열(str), 불리언(bool)

```
age = 30 # 정수 temperature = 36.5 # 부동소수점 is_valid = True # 불리언
```

데이터 타입 범주	데이터 타입
텍스트 타입	<code>str</code>
숫자 타입	<code>int</code> , <code>float</code> , <code>complex</code>
시퀀스 타입	<code>list</code> , <code>tuple</code> , <code>range</code>
매핑 타입	<code>dict</code>
집합 타입	<code>set</code> , <code>frozenset</code>
불리언 타입	<code>bool</code>
바이너리 타입	<code>bytes</code> , <code>bytearray</code> , <code>memoryview</code>
None 타입	<code>NoneType</code> (표현은 <code>None</code>)

정수형 (Integer, `int`)

- 정수형 데이터 타입은 양수, 음수, 0을 포함한 모든 정수를 표현

```
age = 25 temperature = -5 print(age) print(temperature)
```

부동소수점수 (Floating-point, `float`)

- 부동소수점수는 실수를 표현하며, 소수점을 포함한 수를 다룰 때 사용

```
height = 175.5 weight = 65.3 print(height) print(weight)
```

문자열 (String, `str`)

- 문자열은 텍스트를 표현하기 위해 사용
- 파이썬에서 문자열은 작은따옴표(' ') 또는 큰따옴표(" ")로 묶어서 표현

```
name = "John" greeting = 'Hello, World!' print(name) print(greeting)
```

불리언 (Boolean, `bool`)

- 불리언 타입은 참(`True`)과 거짓(`False`)의 두 가지 값만을 가지며, 조건문에서 많이 사용

```
is_student = True is_sunny = False print(is_student) print(is_sunny)
```

리스트, 튜플, 딕셔너리 등 다양한 타입은 뒤에서 학습하면서 다룰 예정

데이터 타입 확인

- `type()` 함수의 사용은 변수나 값의 데이터 타입을 확인하는 데 매우 유용
- 이 함수는 파이썬 내장 함수로, 인자로 주어진 객체의 데이터 타입을 반환
- `type()` 함수는 코드에서 예상치 못한 타입 관련 오류를 디버깅할 때 특히 유용

```
number = 42 print(type(number)) # 출력: <class 'int'> text = "Hello, World!" print(type(text)) # 출력: <class 'str'> flag = True print(type(flag)) # 출력: <class 'bool'>
```

데이터 타입 변환 함수

`str()`

- `str()` 함수는 파이썬에서 제공하는 내장 함수로, 정수, 실수, 리스트, 튜플 등 다양한 데이터 타입을 문자열 형태로 변환

```
number = 100 text = str(number) print(text) # 출력: "100"
```

int()

- `int()` 함수는 문자열 형태의 숫자나, 실수를 정수형으로 변환
- 문자열이 숫자를 나타내지 않는 경우 `ValueError` 오류를 발생

```
string_number = "123" real_number = int(string_number) print(real_number)
# 출력: 123 float_number = 3.14 integer_number = int(float_number) print(integer_number)
# 출력: 3
```

float()

- `float()` 함수는 문자열이나 정수 등을 실수형(float)으로 변환
- 문자열이 유효한 실수 형식이 아닐 경우 `ValueError` 오류를 발생

```
int_number = 10 float_from_int = float(int_number) print(float_from_int) #
출력: 10.0 string_number = "3.14159" float_from_string = float(string_number)
print(float_from_string) # 출력: 3.14159
```

오류 처리와 예외 처리

- 데이터 타입 변환 시 입력 값이 변환 가능한 형태가 아닐 경우 파이썬은 `ValueError` 를 발생
- 이러한 오류를 처리하기 위해 `try...except` 구문을 사용

```
try: user_input = "a123" number = int(user_input) except ValueError: print(
    ("입력 값이 정수로 변환될 수 없습니다."))
```

- `try...except` 구문은 파이썬에서 예외 처리를 위해 사용되는 구문
- 예외는 프로그램 실행 중 발생할 수 있는 오류의 일종으로, 예외적인 상황에서 프로그램이 중단되지 않고, 개발자가 지정한 방식으로 처리될 수 있도록 도와줌

기본 구조

```
try: # 실행할 코드 except ErrorType: # 에러가 발생했을 때 실행할 코드
```

변수 네이밍 규칙

기본 규칙

- 문자, 숫자, 밑줄 포함 가능: 변수 이름은 문자(`a-z`, `A-Z`), 밑줄(`_`), 숫자(`0-9`)를 포함 가능
- 숫자로 시작할 수 없음: 변수 이름은 숫자로 시작할 수 없습니다. 예를 들어, `1name` 은 유효하지 않지만, `name1` 은 유효
- 대소문자 구분: 파이썬은 대소문자를 구분합니다. 따라서, `Age`, `age`, `AGE` 는 모두 다른 변수로 인식

예약어 사용 금지

- 파이썬에서는 `if`, `for`, `class` 등의 예약어를 변수 이름으로 사용할 수 없음
- 이러한 단어들은 파이썬의 문법에 이미 특정 의미로 사용되고 있기 때문

권장하는 네이밍 컨벤션

- 스네이크 케이스: 파이썬에서 일반 변수와 함수 이름에는 소문자와 밑줄을 사용하는 스네이크 케이스를 권장합니다. 예: `my_name`, `student_age`
- 카멜 케이스: 클래스 이름에는 각 단어의 첫 글자를 대문자로 시작하는 카멜 케이스를 사용하는 것이 일반적입니다. 예: `MyClass`, `DataProcessor`

- 상수: 전역 상수는 모두 대문자와 밑줄을 사용합니다. 예: `MAX_SPEED`, `TOTAL`

```
# 좋은 예 username = "admin" user_id = 1001 # 나쁜 예 2nd_user = "bob" # 숫자로 시작할 수 없습니다. user-name = "alice" # 하이픈(-)은 변수명에 사용할 수 없습니다. global = "value" # 'global'은 파이썬의 예약어입니다.
```