기초컴퓨터프로그래밍(파이썬)

2장. 기본자료형과 입출력

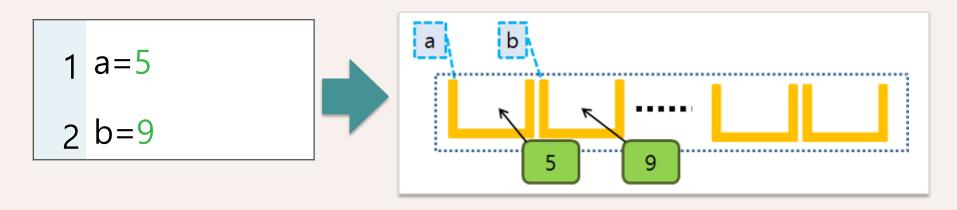
INDEX

- 01 변수와 자료형
- 02 숫자형 자료
- 03 문자열 자료
- 04 문자열 사용 실습
- 05 입력
- 06 출력
- 07 입출력 실습

01

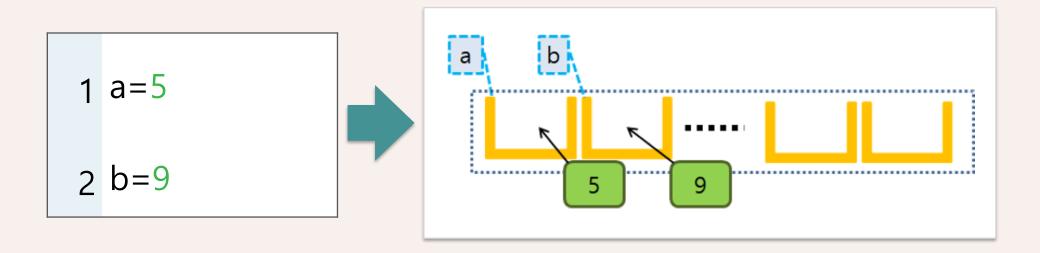
변수와 자료형

- 변수: 프로그램이 사용하는 데이터를 임시로 저장하는 메모리 공간
 - Colab 코드 셀에서 변수명을 적고 그냥 값을 할당하면 됨



01

변수와 자료형



| = | 오른쪽의 값을 왼쪽 변수에 할당 |
|----|-------------------|
| == | 왼쪽과 오른쪽의 값이 같음 |

- 변수 이름 명명 규칙
 - 변수명은 문자와 숫자, 밑줄 기호(_)만으로 구성되어야 함
 - 변수명은 공백을 포함해서는 안됨
 - 변수명의 첫 글자는 숫자로 시작할 수 없음 (반드시 문자 또는 밑줄 기호로 시작해야 함)
 - 대문자와 소문자는 구별됨 (AA와 Aa, aA, aa는 모두 다른 변수)
 - 예약어는 사용할 수 없음 (False, True, if, for 등)
 - 변수 이름은 변수의 역할을 설명할 수 있어야

■ 파이썬 변수의 자료형

기본 자료형

- •숫자형
 - 정수형
 - 실수형
 - 복소수형
- ●문자열

연속 자료형

- ●리스트
- ●튜플
- ●딕셔너리
- ●집합

02 숫자형 자료

| 정수형 | 실수형 | 복소수형 |
|--------------|-----------------------|-------------|
| 123, -123, 0 | 123.23, -12.23, 5.4e5 | 12+3j, 3-5j |

실행 예

$$_{3} c = 0$$

$$1 x = 123.23$$

$$y = -12.23$$

$$z = 5.4e5$$

$$_{4}$$
 w = 5.4E-5

$$1 a = 12+3j$$

$$2 b = 3-5J$$

산술 연산자(arithmetic operators)

| 덧셈 | 뺄셈 | 곱셈 | 나눗셈 | 나머지 | 몫 | 지수승 |
|----|----|----|-----|-----|----|-----|
| + | - | * | / | % | // | ** |

- 연산자 간 우선 순위에 주의



```
1 a=3
2 b=2
3 a-b*2
```

```
1 a=5
2 b=2
3 a/b # 결과: 2.5
4 a%b # 결과: 1
5 a//b # 결과: 2
```

■ 복합 연산자

- 변수의 현재 값에 연산을 수행한 결과 를 다시 해당 변수에 저장하는 경우 사용 가능

| 복합 연산자 | 동작 |
|--------|--------|
| a+=b | a=a+b |
| a-=b | a=a-b |
| a*=b | a=a*b |
| a/=b | a=a/b |
| a//=b | a=a//b |
| a%=b | a=a%b |
| a**=b | a=a**b |

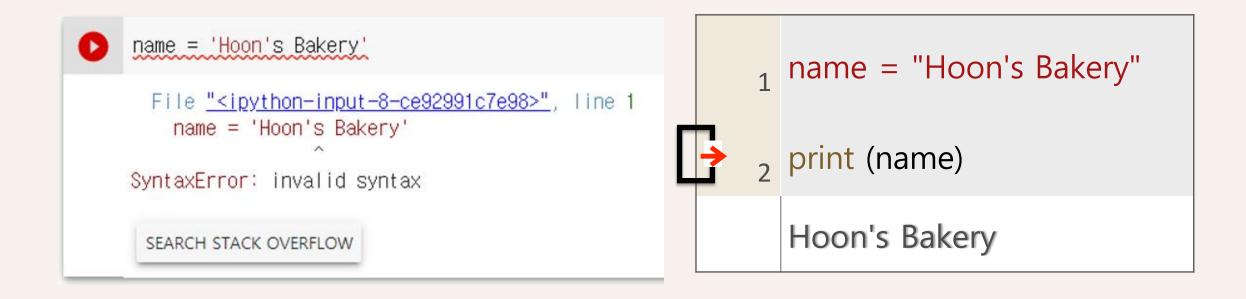
03 문자열자료

- 문자 자료형
 - 알파벳, 한글 등의 문자, 0~9의 숫자, 기호

■ 문자열 정의 방법

```
"박보검" # 큰따옴표로 감싸기
'컴퓨터공학과' # 작은따옴표로 감싸기
"""김보검""" # 큰따옴표로 3개로 감싸기
""컴퓨터'" # 작은따옴표로 3개로 감싸기
```

- 큰따옴표와 작은따옴표는 사용상 차이가 없음
- 한 가지 따옴표가 문자열 내에서 사용될 경우를 대비하여 두 가지 지원





▶ 출력하려는 내용에 큰 따옴표가 포함된 경우



전체를 작은 따옴표로 감싸기

- msg = '"We think Washington has gone crazy," says Andranik'
- print (msg)



"We think Washington has gone crazy," says Andranik

- 문자열 저장의 실제 모습
 - 메모리 한 자리마다 하나의 문자 저장
 - 각 문자에는 순서 번호가 부여

1 a = '1\$ 환율은 1100원입니다.'

- 문자열 인덱싱(indexing)
 - 부여된 순서 번호로 각 문자 접근 가능

```
    번호
    0
    1
    2
    3
    4
    5
    6
    7
    8
    9
    10
    11
    12
    13
    14
    15

    문자
    1
    $
    *
    *
    *
    *
    1
    1
    0
    0
    원
    입
    니
    다
    .
```



```
1 a = '1$ 환율은 1100원입니다.'
```

print (a[1], a[3], a[5])

\$ 환 은

■ 문자열 인덱싱(indexing)

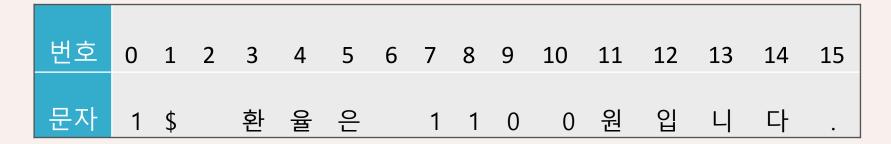
- 음수 인덱스도 가능

```
번호 0 -15 -14 -13 -12 -11 -10 -9 -8 -7 -6 -5 -4 -3 -2 -1
문자 1 $ 환 율 은 1 1 0 0 원 입 니 다 .
```

! 실행 예

```
1 a = '1$ 환율은 1100원입니다.'
2 print (a[-0], a[-4], a[-5])
1 입 원
```

- 문자열 슬라이싱(slicing)
 - 여러 개의 문자를 한꺼번에 추출
 - s[시작번호 : 끝 번호+1] (음수 인덱스도 사용 가능)
 - s의 문자 중 시작번호부터 끝번호까지 가져옴



실행 예

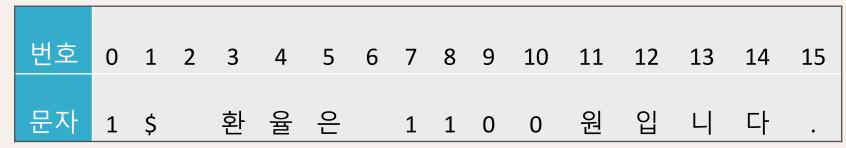
1 a = '1\$ 환율은 1100원입니다.'

print (a[3:5], a[7:11])



환율 1100

- 문자열 슬라이싱(slicing)
 - 맨 처음부터 가져오거나 맨 끝까지 가져올 때는 시작번호나 끝 번호 생략 가능



① 실행 예 3 print (a[-4:], a[:6]) 다 입니다. 1\$ 환율은

■ 문자열 전용 함수

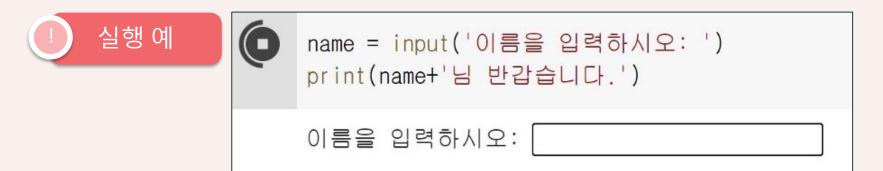
| 함수 이름 | 설명 |
|-------------------------|-------------------------------------------------------|
| count() | 문자열에 포함된 특정 문자의 개수 반환 |
| find() | 문자열에서 찾는 문자가 처음으로 나타나는 위치 |
| join() | 문자열 문자 사이사이에 지정한 문자 삽입 |
| split() | 문자열을 특정 문자 기준으로 분리 |
| upper() | 문자열을 모두 대문자로 변환 |
| lower() | 문자열을 모두 소문자로 변환 |
| replace(A, B [, count]) | 문자열에서 A를 찾아서 B로 교체 (count 옵션: count번째까지의 A만 B로 교체) |

04 문자열 사용 실습



05 입력

- input(): 사용자에게서 값을 입력받는 함수
 - 함수 내의 문자열은 사용자에게 입력가이드로 화면상에 표시



```
1 name = input('이름을 입력하시오: ')

print(name+'님 반갑습니다.')
이름을 입력하시오: 유재석
유재석님 반갑습니다.
```

■ input() 사용시 유의할 점

- 모든 입력이 문자열로 저장



- 1 eng = input('영어 점수를 입력하시오: ')
- 2 math = input('수학 점수를 입력하시오: ')
- 3 totalScore = eng+math
- 4 print('총점 =', totalScore)

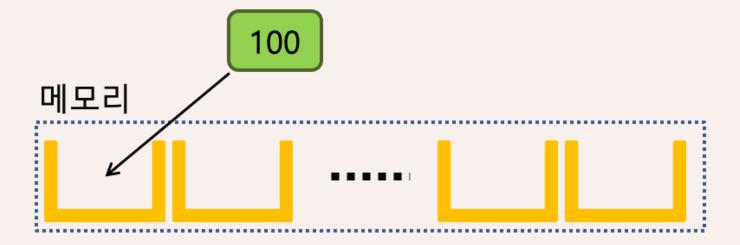


→ 영어 점수를 입력하시오: 80

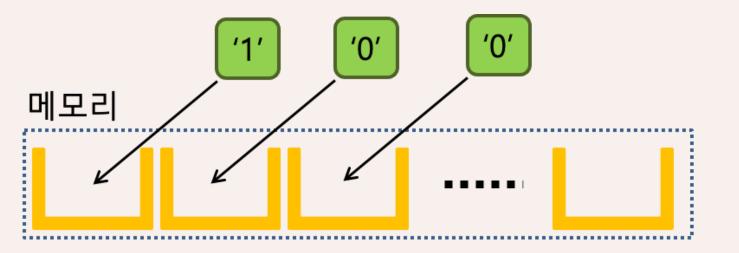
수학 점수를 입력하시오: 90

총점 = 8090

- 수 100 vs. 문자열 '100'
 - 수 100은 개수나 양 등을 나타내는 값



- 문자열 100은 '1', '0', '0' 세 개로 구성



■ int(): 문자열을 수로 변환

! 실행 예

- 1 eng = int(input('영어 점수를 입력하시오: '))
- 2 math = int(input('수학 점수를 입력하시오: '))
- 3 totalScore = eng+math
- 4 print('총점 =', totalScore)



영어 점수를 입력하시오: 80

수학 점수를 입력하시오: 90

총점 = 170

06 출력

- print(): 문자열 및 변수값 출력을 위한 함수
 - 문자열 중간 중간에 변수의 값이 출력되어야 하는 경우엔 작성이 번거로움 유재석님의체중은 67kg이고,키는 175.4cm입니다.
 - 변수값을 중간에 넣어 문자열을 만든 후 출력하는 문자열 포맷팅 방법 사용
 - 형식지정자를 활용한 문자열 포맷팅
 - f-strings를 활용한 문자열 포맷팅

- 형식지정자 사용한 문자열 포맷팅
 - 변수의 값을 문자열 내에 삽입하는 방법
 - 형식지정자 + 문자열 뒤에 '%'를 넣고 변수명

- ! 실행 예
- 1 FirstName = "재석"; LastName = "유"; job1 = "개그맨"; job2 = "MC"
- 2 print("저는 %s이자 %s인 %c%s 입니다." % (job1, job2, LastName, FirstName))
- → 저는 개그맨이자 MC인 유재석 입니다.
- 형식지정자
 - 위치와 자료형 선언

| 형식지정자 | 자료형 |
|-------|----------|
| %s | 문자열 |
| %c | 문자 1개 |
| %d | 정수 |
| %f | 실수 |
| %% | % 기호의 출력 |

- 형식지정자 사용한 문자열 포맷팅
 - 여러 개의 변수를 삽입하는 방법
 - 형식지정자들로 변수값 삽입 위치를 정한 뒤 문자열 뒤에 '%'를 넣고 괄호 안에 변수를 출력 순서대로 나열

■ 형식지정자 사용한 문자열 포맷팅

실행 예

- 1 name = "유재석" #문자열
- 2 weight = 67 # 정수형
- 3 height = 175.4 #실수형
- 4 print("%s 님의 체중은 %dkg이고, 키는 %fcm입니다." % (name, weight, height))



유재석 님의 체중은 67kg이고, 키는 175.4cm입니다.

- 형식지정자 포맷팅의 단점
 - 변수가 여럿인 경우 코드를 보고 출력될 내용을 한눈에 예상하기 어려움
- 실행 예
- 1 FirstName = "재석"; LastName = "유"; job1 = "개그맨"; job2 = "MC"
- 2 print("저는 %s이자 %s인 %c%s 입니다." % (job1, job2, LastName, FirstName))



□→ 저는 개그맨이자 MC인 유재석 입니다.

■ f-strings 사용한 문자열 포맷팅

- 문자열 시작 전에 f 또는 F 입력
- 변수를 넣고 싶은 곳에 중괄호로 감싼 변



- 1 FirstName = "재석"; LastName = "유"; job1 = "개그맨"; job2 = "MC"
- 2 print(f"저는 {job1}이자 {job2}인 {LastName}{FirstName} 입니다.")
- → 저는 개그맨이자 MC인 유재석 입니다.
 - 3 name = "유재석"; weight = 67; height = 175.4
 - 4 print(F"{name}님의 체중은 {weight}kg이고, 키는 {height}cm입니다.")

유재석님의 체중은 67kg이고, 키는 175.4cm입니다.

- print()의 end 옵션
 - print()는 출력 후 줄을 자동으로 바꿈
 - 이는 end 옵션의 기본값이 줄바꿈을 뜻하는 특수문자('\n')이기 때문
 - 줄바꿈을 원하지 않는다면 end 옵션 값을 변경 (end=")



- 1 print("안녕하세요.")
- 2 print("유재석")
- 3 print("입니다.")
- 안녕하세요.
 - 유재석
 - 입니다.
 - 1 print("안녕하세요.")
 - 2 print("유재석",end=")
 - 3 print("입니다.")
- 안녕하세요.

유재석입니다.