

/* elice */

파이썬 기초

Python으로의 초대



수강 목표

컴퓨터가 우리에게 말할 수 있는 방법을 배운다.

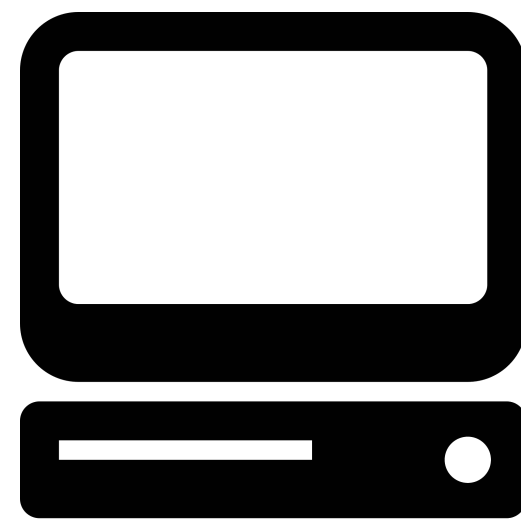
우리가 다루게 될 기본 자료들의 특징을 이해한다.

자료를 그릇에 담아 활용하는 법을 배운다.

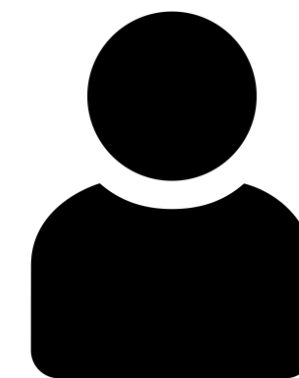
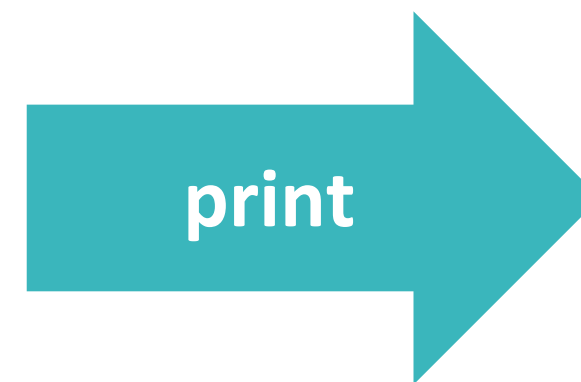
출력 - print

출력

Python에선 우리가 원하는 정보나 자료를
컴퓨터가 **출력**하게 할 수 있습니다!



컴퓨터



사용자

print

```
print('출력할 내용')
```

```
print('나의 꿈은 파이썬 정복!')
```

실행 결과

나의 꿈은 파이썬 정복!

print

Q. 하나가 아닌, 여러 자료를 출력하고
싶은데, 어떻게 하면 좋을까?

실행 결과

3 Hello!

print

A. ,(콤마)를 이용해서 여러 자료를 출력할 수 있다!

```
print(3, 'Hello!')
```

실행 결과

3 Hello!

print

Q. print로 삼행시를 짓고 싶은데...
어떻게 하면 좋을까?

실행 결과

비행기에 타신 승객 여러분
행복한 여행 되십시오
기내식은 바밤바

print

A. print를 여러 번 사용하자!

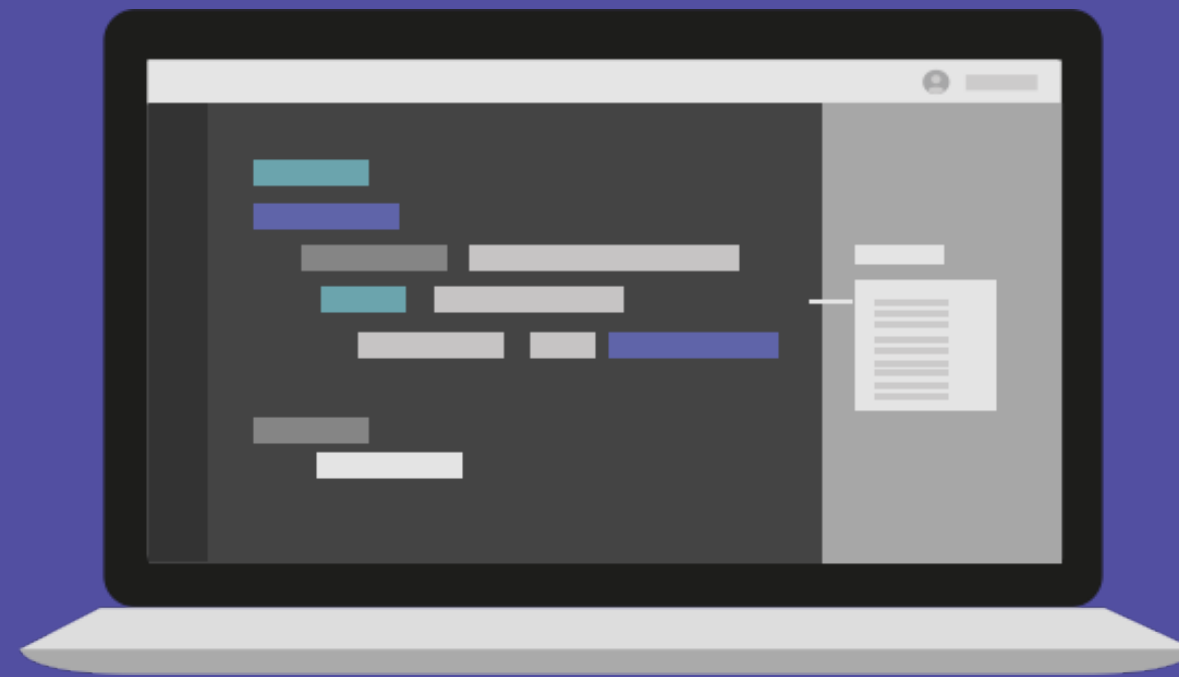
```
print('비행기에 타신 승객 여러분')
```

```
print('행복한 여행 되십시오')
```

```
print('기내식은 바밤바')
```

[실습1]

삼행시 짓기



기본 자료형

1) 숫자형 (Number)

숫자로 이루어진 자료형 정수나 실수 등을 다룰 수 있다.

숫자끼리의 연산이 가능하다.

3 # 정수(integer)

3.14 # 실수(float)

3+4j # 복소수

2) 문자열 (String)

문자나 문자들을 **늘어놓은 것**
큰 따옴표(“”)와 작은 따옴표(“)로 구분

‘Hello!’

‘3.14’ # 작은 따옴표 OK

“3.14” # 큰 따옴표 OK

3) 리스트 (List)

여러 자료를 보관하는 자료형

다른 종류의 자료를 함께 담을 수 있습니다.

자료 안에 순서가 있습니다.

```
[] # 빈 리스트
```

```
['a', 'b']
```

```
['a', 2] # 다른 자료형을 함께!
```

```
[['a', 'b'], 'c', 234]
```

cf.) 주석 (Comment)

주석은 컴퓨터가 무시합니다.

한 줄 주석은 #, 여러 줄 주석은 `““““ ””””`, `“ ”`로 합니다.

주석 처리한 말들은

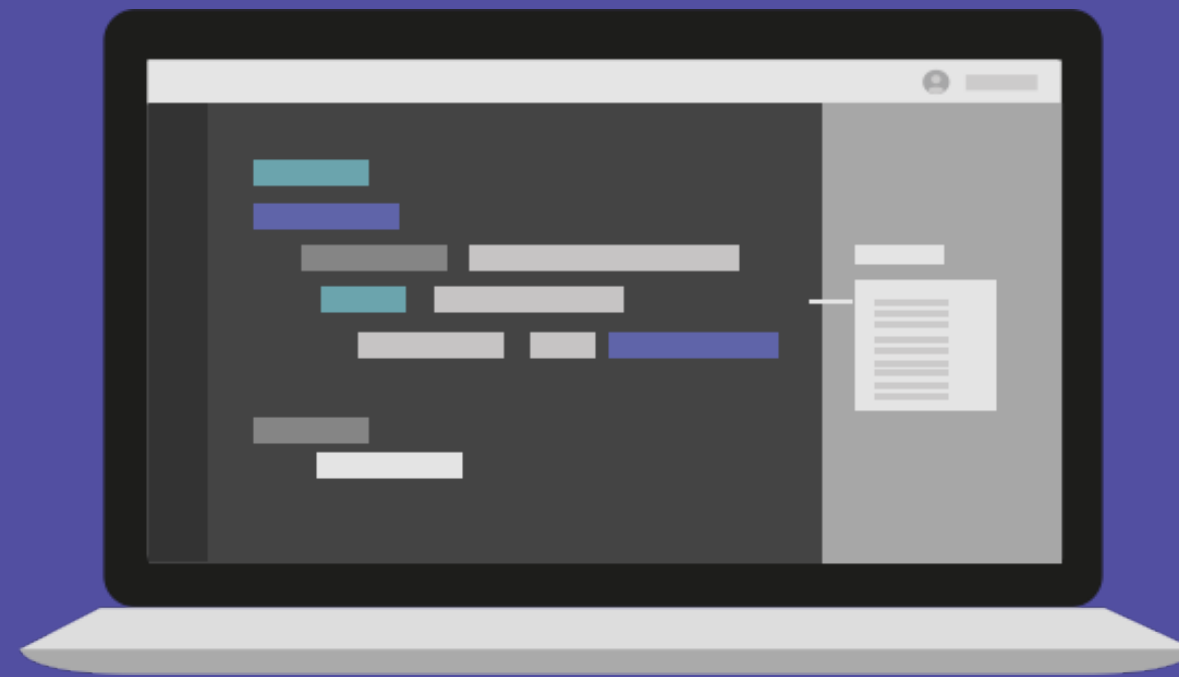
““““

컴퓨터가
실행하지
않아요!

””””

[실습2]

신비한 자료사전 : 자료형

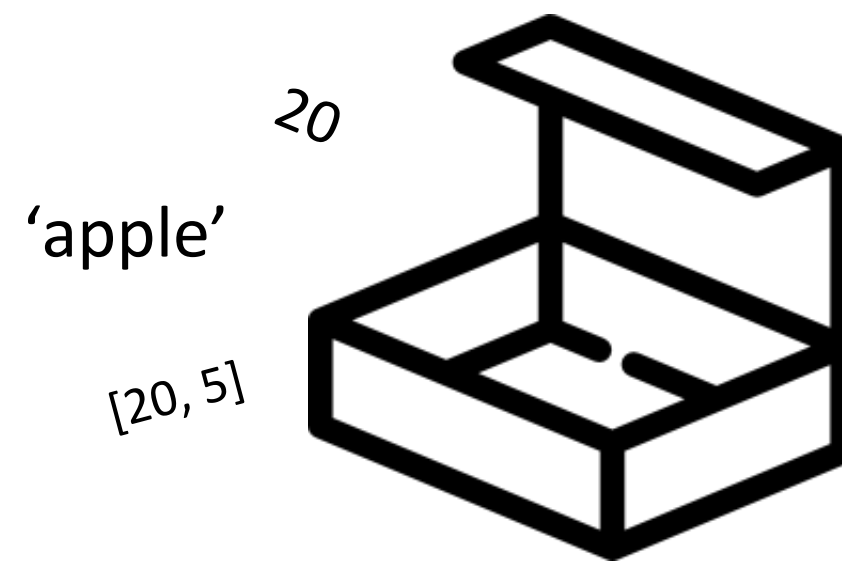


변수

변수

자료를 ‘그릇’에 담아서 보관, 사용하면 편리합니다.

이를 **변수(variable)**라 합니다.



변수 사용방법

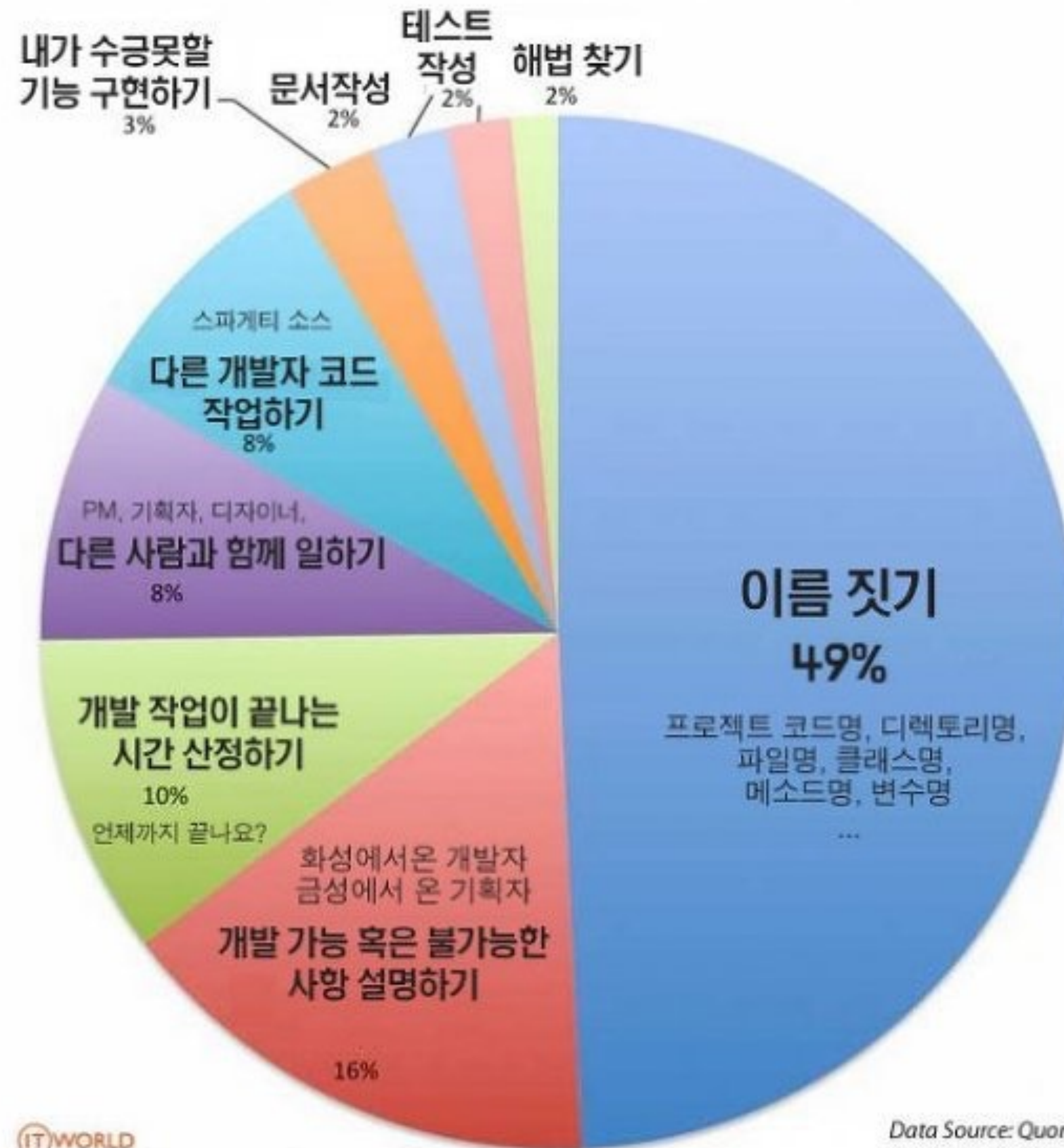
변수 이름 = 자료

```
num = 10 # 숫자
```

```
name = 'Michael' # 문자열
```

```
grade = ['A+', 'B+', 'A0'] # 리스트
```

프로그래머가 가장 힘들어하는 일은?



변수 사용방법

변수 이름 = 자료

```
num = 10 # 숫자
```

```
name = 'Michael' # 문자열
```

```
grade = ['A+', 'B+', 'A0'] # 리스트
```

참고

= : assign (대입 연산자)

== : equal (등호)

변수 이름 짓는 방법

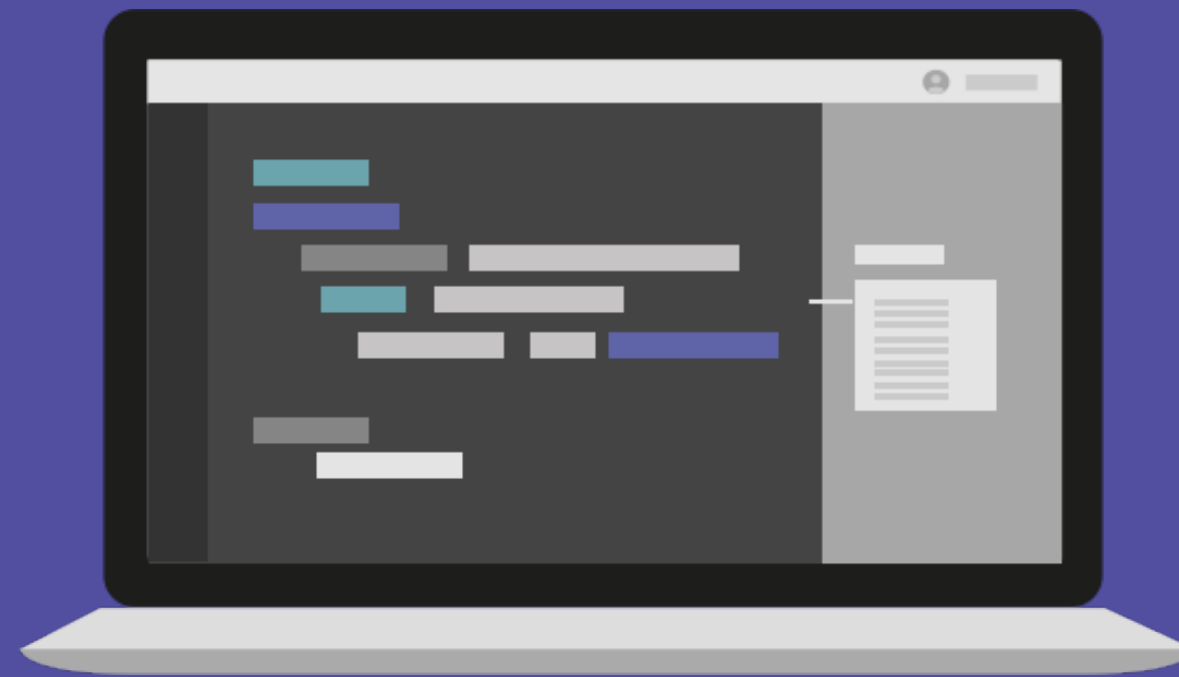
숫자, 알파벳, 한글, 언더바(_) 등을
사용할 수 있다.

변수 이름 짓는 방법

1. 변수이름이 **숫자로 시작**하면 안 된다.
2. 파이썬 문법에서 사용되는 **예약어**
(이미 사용되는 단어 : for, while, if ...) 사용 금지
3. **공백 문자**(' ')와 **연산자**(+, -, % 등) 사용 금지

[실습3]

그릇의 달인 : 변수



자료형의 연산

수학의 사칙연산

+

더하기

-

빼기

×

곱하기

÷

나누기

숫자형 자료의 사칙연산

+

더하기

-

빼기

*

곱하기

/

나누기

숫자형 자료의 사칙연산

```
print(3 + 5)    # 8
```

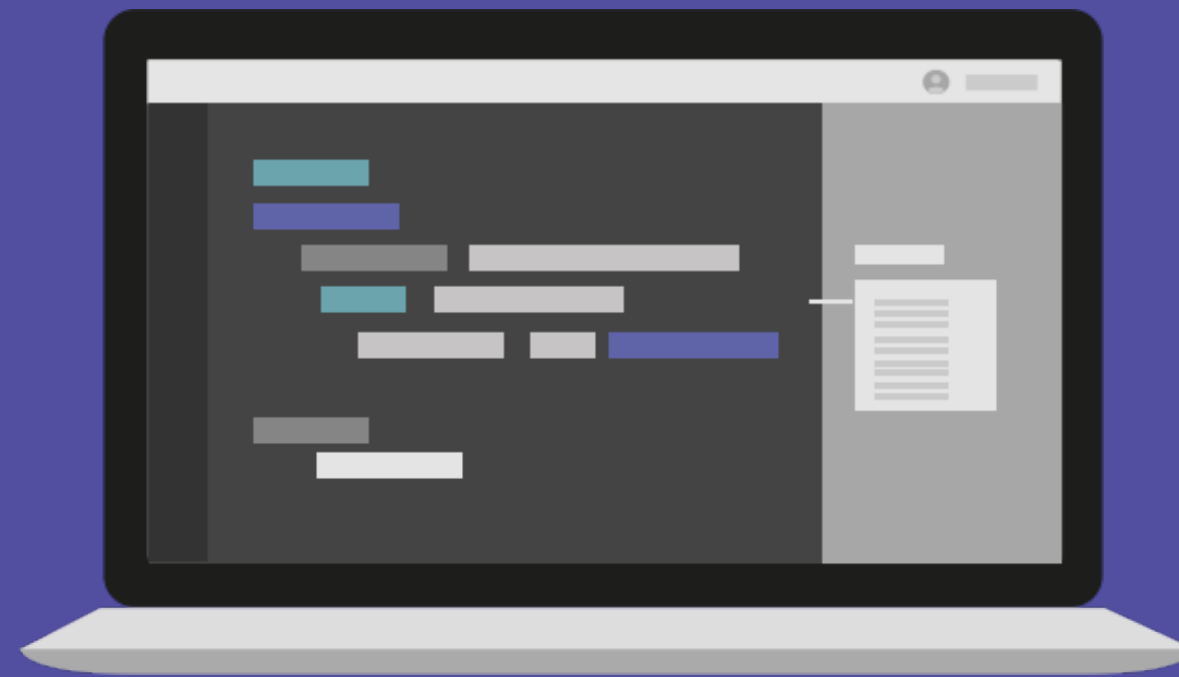
```
print(3 - 5)    # -2
```

```
print(3 * 5)    # 15
```

```
print(3 / 5)    # 0.6
```

[실습4]

알쏭달쏭 수수께끼 : 사칙연산



숫자형 자료의 특수연산

//

몫 연산자

%

나머지 연산자

**

제곱 연산자

숫자형 자료의 특수연산

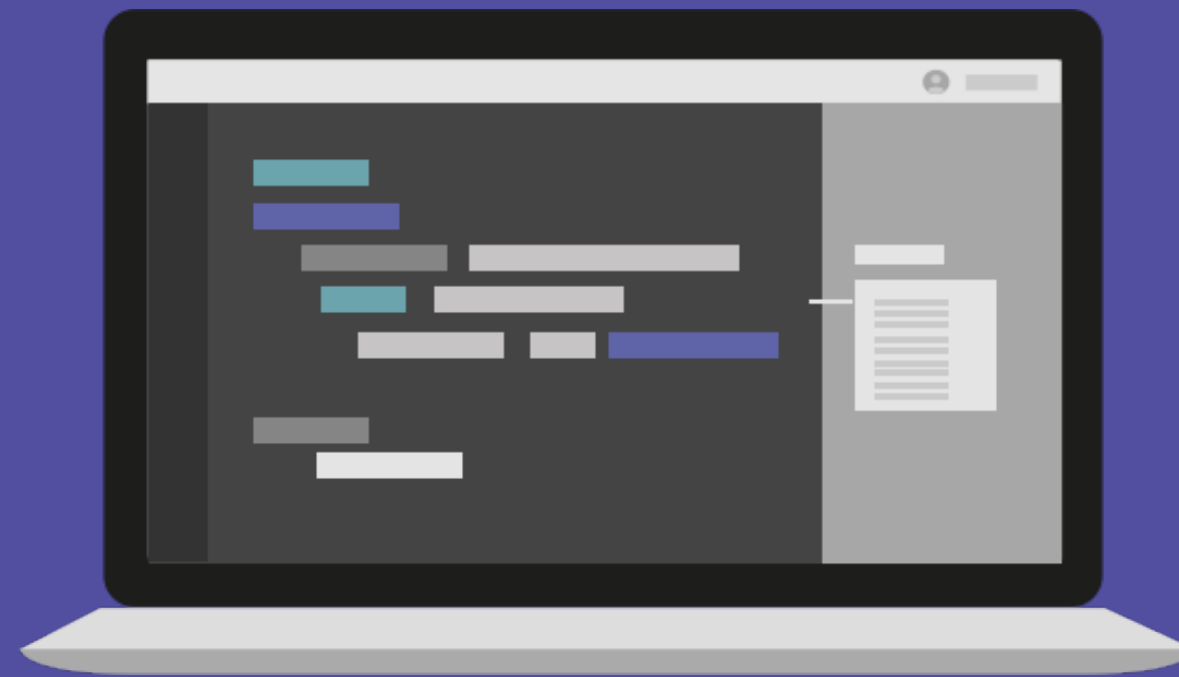
```
print(13 // 5)    # 2
```

```
print(13 % 5)     # 3
```

```
print(2 ** 4)     # 16
```


[실습5]

거꾸로 계산기 : 특수연산



문자형 자료의 연산

+

이어 붙이기

with 문자열

*

반복하기

with 숫자

문자형 자료의 연산

```
print('안녕' + '하세요')
```

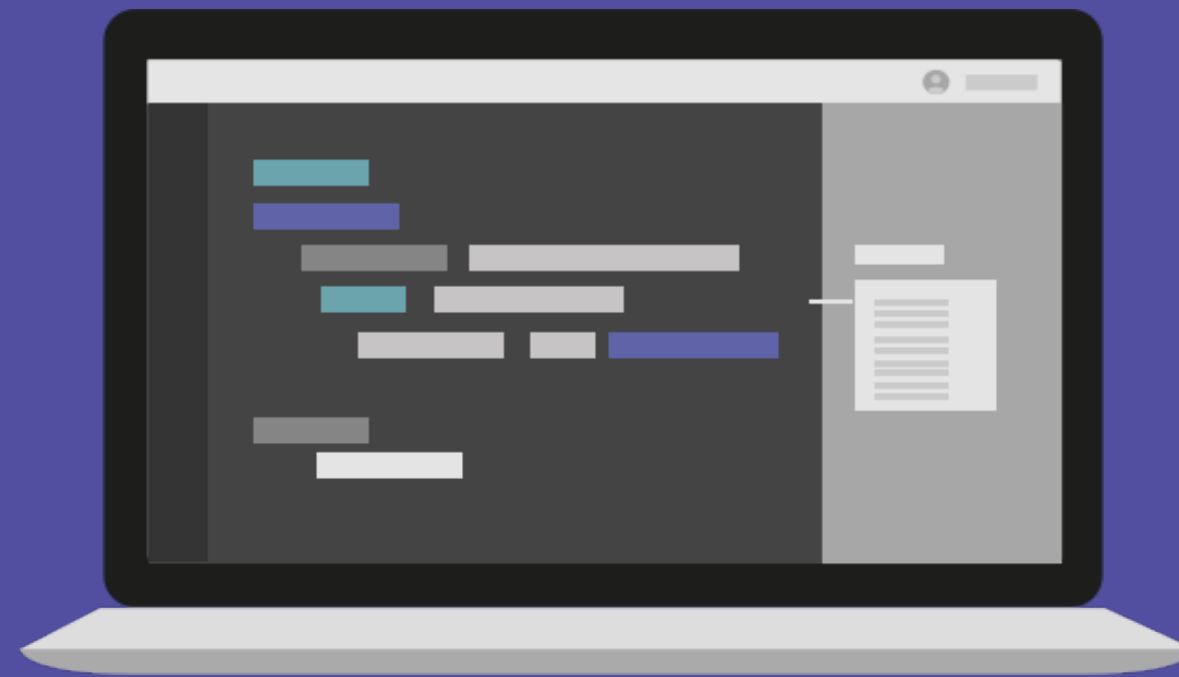
```
# 안녕하세요
```

```
print('안녕' * 3)
```

```
# 안녕안녕안녕
```

[실습6]

얼쑤! 자진모리 장단 : 문자열연산



문자열/리스트 접근하기

질문!

‘rescue’와 ‘secure’은 다른 문자열입니다

[1, 2, 3]과 [3, 2, 1]은 다른 리스트입니다

Why?

원소의 배치 순서가 다르기 때문이죠!

인덱스

문자열과 리스트 자료형은 여러 원소로 이루어져 있고
각각의 위치를 0부터 순서대로 매길 수 있습니다.

= 인덱스

'R e a d y'

0 1 2 3 4

[2, 4, 6, 8]

0 1 2 3

인덱싱

Index를 이용해서 리스트나 문자열의
특정 위치의 원소를 가져오는 방법

string/list [index]

```
# alpha에서 인덱스 1인 원소 'e'를 출력  
alpha = 'Ready'  
print(alpha[1])  
>>> e
```


슬라이싱

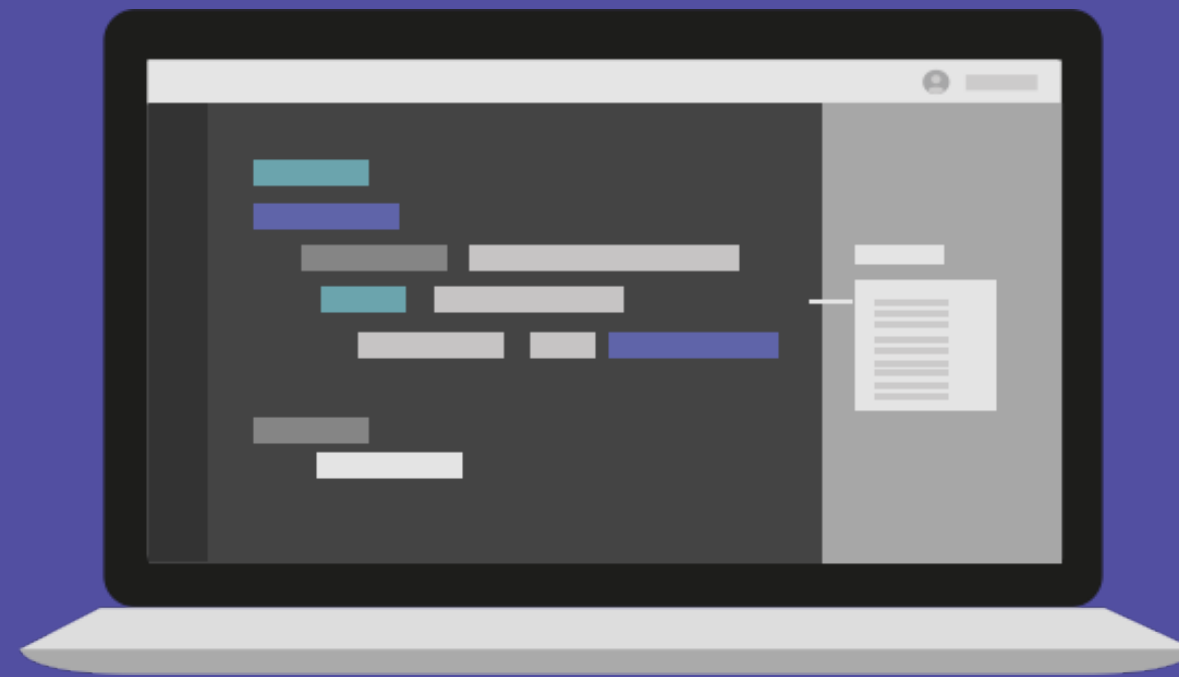
Index를 이용해서 리스트나 문자열의
일부분을 잘라서 가져오는 방법

string/list [a(시작 인덱스):b(종료 인덱스)]

```
# beta에서 2번째 원소 이상, 5번째 원소 미만을 가져온다.  
beta = [2, 4, 6, 8, 10, 12, 14]  
print(beta[2:5])  
>>> [6, 8, 10]
```

[실습기]

보석 홈치기 : 인덱싱/슬라이싱



/* elice */

문의 및 연락처

academy.elice.io

contact@elice.io

facebook.com/elice.io

medium.com/elice