

---

# Python 기초 강의

2 Weeks



Cyber-Physical Systems Security

순천향대학교 정보보호학과

김준원([ruskonert@gmail.com](mailto:ruskonert@gmail.com))

2018.03.26

---

# Agenda

❖숫자 자료형

❖문자열 자료형

❖리스트 & 튜플 자료형

❖변수(Variable)

❖Q & A

---

# 숫자

정수형

실수형

8진수와 16진수

관련 연산자

## ❖숫자 형태로 이루어진 자료형

- ✓ 크게 정수, 실수, 8진수, 16진수로 나눔
- ✓ 8진수 사용시 '0o' 사용
- ✓ 16진수 사용시 '0x' 사용

항목	사용 예
정수	12, -34, 5, 0
실수	123.45, -1234.5, 3.4e10
8진수	0o34, 0o25
16진수	0x2A, 0xFF

## ❖ 정수형 (Integer)

```
>>> a = 120
>>> a = -12
```

## ❖ 기수(Numeral System)

✓ 8진수

```
>>> a = 0o15
```

✓ 16진수

```
>>> a = 0x15
```

## ❖ 실수형 (Floating-point)

## ✓ 소수점을 포함한 방식

```
>>> a = 1.5
```

## ✓ 지수 표현 방식

```
>>> a = 5.5E-6
```

❖ 사칙연산

```
>>> a = 3
>>> b = 4
>>> a + b
7
```

```
>>> a = 3
>>> b = 4
>>> a * b
12
```

```
>>> a = 3
>>> b = 4
>>> a / b
0.75
```

◆\*\* 연산자

✓ **제공 값 반환**

```
>>> a = 3
>>> b = 4
>>> a ** b
81
```

```
>>> a = 3
>>> b = 0
>>> a ** b
1
```

◆ \*\* 연산자

✓ 나머지 값 반환

```
>>> a = 7
>>> b = 3
>>> a % b
1
```

◆ // 연산자

✓ 소수점 아랫자리를 버림

```
>>> a = 7
>>> b = 4
>>> a / b
1.75
>>> a // b
1
>>> -a / b
-1.75
>>> a // b
-2
```

- ❖ 연산자(Operator): 둘 이상의 요소에 대응하는 계산 기호  
일반적으로 쓰이는 연산자들은 ++, --, +, /, -, %, == 등이 있음

---

# 문자열

문자열 만들기

문자열 연산

문자열 인덱싱 & 슬라이싱

문자열 Formatting

문자열 관련 함수 (생략)

“CPSS Team”  
“123”  
“a”

“123”  
“a”

- ```
'''hello world'''
```



## ❖ 문자열에 작은 따옴표(') 포함시키기

- ✓ 큰따옴표를 사용하여 둘러싸야 하며 작은 따옴표로 묶으면 문법 오류 발생

```
>>> message = "CPSS's the club of SCH University"  
CPSS's best club of SCH University  
>>> message = 'CPSS's the club of SCH University'  
SyntaxError: invalid syntax
```

## ❖ 문자열에 큰 따옴표(") 포함시키기

- ✓ 작은 따옴표를 사용하여 둘러싸야 하며 큰 따옴표로 묶으면 문법 오류 발생

```
>>> message = 'He says, "Hello world".'  
He says, "Hello world".  
>>> message = "He says, "Hello world"."  
SyntaxError: invalid syntax
```



```
>>> head = "Python"
>>> head * 3
'PythonPythonPython'
```

✓ 인덱싱(번호) : 문자열에서 각각의 문자를 가리키는 번호

✓ 0번째부터 시작하며 각각의 문자 위치에 순서대로 대응

## Cyber-physical Systems Security

## ❖ 문자열 인덱싱 번호 활용

- ✓ 인덱스 번호가 음수라면 뒤에서부터 시작하는 것을 의미함

```
>>> a = "Life is too short, You need Python"
>>> a[-1]
'o'
>>> a[-3]
'h'
```

## ❖ 문자열 슬라이싱

- ✓ 인덱싱 번호를 활용하여 문자열 뽑아내기

```
>>> a = "Life is too short, You need Python"
>>> b = a[0] + a[1] + a[2] + a[3]
>>> b
'Life'
>>> a[0:4]
'Life'
```

```
>>> a = "Life is too short, You need Python"
>>> a[0:4]
'Life'
>>> a[0:5]
'Life '
>>> a[5:]
' is too short, You need Python'
>>> a[:17]
'Life is too short'
```

# 문자열 Formatting

1010 00 0100110110101010  
1010 00 0100110110101010  
011001010  
0110101010

## ❖ 문자열을 출력 시 사용되는 형식

### ✓ 숫자 바로 대입

```
>>> "I eat %d apples." % 3  
'I eat 3 apples.'
```

### ✓ 문자열 바로 대입

```
>>> "I eat %s apples." % "three"  
'I eat three apples.'
```

### ✓ 숫자 나타내는 변수 대입

```
>>> num = 3  
>>> "I eat %d apples." % num  
'I eat 3 apples.'
```

### ✓ 2개 이상의 값 넣기

```
>>> num = 3  
>>> name = "Jane"  
>>> "%s eats %d apples." % (name, num)  
'Jane eats three apples.'
```

### ✓ 정렬 및 공백

```
>>> "%10sJane!" % "hi,"  
'          hi,Jane!'
```

```
>>> "%-10s Jane!" % "hi,"  
'hi,          Jane!'
```

### ✓ 소수점 표현하기

```
>>> "%0.5f" % 3.123456  
'3.12345'  
>>> "%10.4f" % 3.123456  
'          3.1234'
```

---

# 리스트 & 튜플

리스트 사용하기

리스트 연산자

리스트 수정 & 삭제

리스트 함수 (생략)

튜플 사용하기



## ❖리스트 더하기

```
>>> even = [2,4,6,8]
>>> odd = [1,3,5,7]
>>> even + odd
[2,4,6,8,1,3,5,7]
```

## ❖리스트 곱하기 (반복)

```
>>> even = [2,4,6,8]
>>> even * 2
[2,4,6,8,2,4,6,8]
```

**◆ 활용**

```
>>> even = [2,4,6,8, "Hello"]
>>> even[3] * 2
16
>>> even[4] + 13
TypeError: number coercion failed
```



## ❖리스트 값 수정하기

```
>>> a = [1, 2, 3]
>>> a[1] = 4
>>> a
[1, 4, 3]
```

```
>>> a = [1, 2, 3]
>>> a[1:2] = ['a', 'b', 'c']
>>> a
[1, 'a', 'b', 'c', 3]
```

## ❖리스트 요소 삭제하기

```
>>> a = [1, 2, 3, 4]
>>> a[1:3] = []
>>> a
[1, 4]
```

```
>>> a = [1, 2, 3, 4]
>>> del a[1]
>>> a
[1, 3, 4]
```

## ❖ 튜플(tuple)

- ✓ 리스트와 비슷하나 값의 생성, 삭제, 수정이 불가능
- ✓ 값을 상수처럼 사용할 때 많이 사용함
- ✓ 어떤 요소들을 지우거나 변경하려고 시도 시 에러 발생
- ✓ 괄호()를 사용하거나 어떠한 것으로도 묶지 않고 심표로만 사용해 구분시켜 생성

```
>>> a1 = ()
>>> a2 = (1,)
>>> a3 = (1,2,3)
>>> a4 = 1, 2, 3
```

---

# 변수

변수의 정의

변수를 만드는 여러가지 방법

메모리 상에서 변수 없애기

# 변수의 정의

1010 00 010011011010  
1010 00 010011011010  
011001010  
01101010

## ❖ 변수 (Variable)

- ✓ 이론적 용어 : 변하는 수
- ✓ 프로그래밍적 용어: 객체를 가리키는 것 또는 어떠한 값을 담는 공간을 지칭함

## ❖ 타 언어와의 차이점

- ✓ 사용법 : 변수명 = 변수에 저장할 값
  - ◆ Java나 C언어와 달리 변수의 자료형을 쓸 필요가 없으며 저장된 값에 따라 스스로 판단함
- ✓ `>>> a = 1` 이라고 하면 1이라는 값을 가지는 정수 자료형(=객체)이 자동으로 메모리에 생성
  - ◆ C언어와 달리 1은 상수가 아니라 정수형 객체

## ❖ 변수를 만드는 여러 가지 방법

- ✓ 튜플을 이용한 방법 (상수형처럼 사용)

```
>>> a, b = ('py', 'life')    >>> a, b = 'py', 'life'
```

- ## ✓ 리스트를 이용한 방법

```
>>> a, b = ['py', 'life']
```

- ✓ 여러 개의 변수에 값을 대입하기

```
>>> a = b = 'python'
```

- ## ➤ 두 변수의 값을 변경하기

```
>>> a = 3
>>> b = 5
>>> a, b = b, a
>>> a
5
>>> b
3
```

궁금하신 점이나 질문이 따로 있다면?

개인 이메일 [ruskonert@gmail.com](mailto:ruskonert@gmail.com)

동아리 이메일 [support@cpss.network](mailto:support@cpss.network)

소스 코드 참고 [github.com/CPSSOpenSource](https://github.com/CPSSOpenSource)

3주차 강의부터 본격적으로 소스 코드를 작성합니다. (사용할 IDE: PyCharm)

노트북을 지참하시는 것을 강력히 권장합니다.

강의를 들어 주셔서 감사합니다.