

학생 교과목 포트폴리오

20190720 이제호
컴퓨터정보공학과 PC

2020 학년도 1학기	전공	컴퓨터정보공학과	학부	컴퓨터공학부
과 목 명	파이썬프로그래밍(2019009-PC)			
강의실 과 강의시간	화:1(3-217),2(3-217),3(3-217)		학점	3
교과분류	이론/실습		시수	3

담당 교수	강환수 + 연구실 : 2호관-706 + 전 화 : 02-2610-1941 + E-MAIL : hskang@dongyang.ac.kr + 면담가능기간 : 화요일 13~16
-------	--

학과 교육목표				
과목 개요	2010년 이후 파이썬의 폭발적인 인기는 제4차 산업혁명 시대의 도래와도 밀접한 연관성이 있다. 컴퓨팅 사고력은 누구나 가져야할 역량이며, 인공지능, 빅데이터, 사물인터넷 등의 첨단 정보기술이 제4차 산업혁명 시대의 기술을 이끌고 있다. 제4차 산업혁명 시대를 주도하는 핵심 기술은 데이터과학과 머신러닝, 딥러닝이며, 이러한 분야에 적합한 언어인 파이썬은 매우 중요한 언어가 되었다. 본 교과목은 파이썬 프로그래밍의 기초적이고 체계적인 학습을 수행한다. 본 교과목을 통하여 데이터 처리 방법에 대한 효율적인 파이썬 프로그래밍 방법을 학습한다.			
학습목표 및 성취수준	1. 컴퓨팅 사고력의 중요성을 인지하고 4차 산업혁명에서 파이썬 언어의 필요성을 이해할 수 있다. 2. 기본적인 파이썬 문법을 이해하고 데이터 처리를 위한 자료구조를 이해하여 적용할 수 있다. 3. 문제 해결 방법을 위한 알고리즘을 이해하고 데이터 처리에 적용 할 수 있다. 4. 파이썬 프로그램을 이용하여 실무적인 코딩 작업을 할 수 있다.			
	도서명	저자	출판사	비고
주교재	파이썬으로 배우는 누구나 코딩	강환수, 신용현	홍릉과학출판사	
수업시 사용도구	파이썬 기본 도구, 파이참, 아나콘다와 주피터 노트북			
평가방법	중간고사 30%, 기말고사 40%, 과제물 및 퀴즈 10% 출석 20%(학교 규정, 학업성적 처리 지침에 따름)			
수강안내	1. 파이썬의 개발환경을 설치하고 활용할 수 있다. 2. 파이썬의 기본 자료형을 이해하고 조건과 반복 구문을 활용할 수 있다. 3. 파이썬의 주요 자료인 리스트, 튜플, 딕셔너리, 집합을 활용할 수 있다. 4. 파이썬의 표준 라이브러리와 외부 라이브러리를 이해하고 활용할 수 있다. 5. 파이썬으로 객체지향 프로그래밍을 수행할 수 있다.			

1 주차	[개강일(3/16)]
학습주제	교과목 소개 및 강의 계획 1장 파이썬 언어의 개요와 첫 프로그래밍
목표및 내용	<ul style="list-style-type: none"> · 파이썬 언어란 무엇인지 이해하고 이 언어가 인기 있는 이유를 설명할 수 있다. · 파이썬 개발 도구를 설치해 프로그램을 구현할 수 있다. · 파이썬의 특징과 활용 분야를 설명할 수 있다.
미리읽어오기	교재 1장, 파이썬 개발환경 설치 파이썬 IDLE
과제,시험,기타	도전 프로그래밍
2 주차	[2주]
학습주제	2장 파이썬 프로그래밍을 위한 기초 다지기
목표및 내용	<ul style="list-style-type: none"> · 파이썬의 재료인 문자열과 수에 대해 이해하고 코드로 구현할 수 있다. · 변수를 이해하고 다양한 대입 연산자를 활용할 수 있다. · 표준 입력으로 문자열을 입력받은 후 원하는 자료로 변환해 활용할 수 있다. · 파이썬 IDLE을 활용할 수 있다.
미리읽어오기	교재 2장 리터럴과 변수의 이해 아나콘다의 주피터 노트북
과제,시험,기타	도전 프로그래밍
3 주차	[3주]
학습주제	3장 일상에서 활용되는 문자열과 논리 연산
목표및 내용	<ul style="list-style-type: none"> · 문자열에서 문자나 부분 문자열을 반환하는 여러 방법을 구현할 수 있다. · 문자열 객체에 소속된 다양한 메소드를 이해하고 활용할 수 있다. · 논리 값을 이해하고 다양한 연산을 사용해 실생활에서의 표현에 활용할 수 있다. · 아나콘다의 주피터 노트북을 활용할 수 있다.
미리읽어오기	교재 3장 문자열과 논리연산 파이참(pycharm)
과제,시험,기타	도전 프로그래밍
4 주차	[4주]
학습주제	4장 일상생활과 비유되는 조건과 반복
목표및 내용	<ul style="list-style-type: none"> · 조건에 따라 하나를 결정하는 if문을 구현할 수 있다. · 반복을 수행하는 while문과 for문을 구현할 수 있다. · 임의의 수인 난수를 이해하고 반복을 제어하는 break문과 continue문을 활용할 수 있다. · 파이참(pycharm)을 활용할 수 있다.
미리읽어오기	교재 4장 조건과 반복
과제,시험,기타	도전 프로그래밍

5주차	[5주]
학습주제	5장 항목의 나열인 리스트와 튜플
목표및 내용	<ul style="list-style-type: none"> • 다양한 종류의 항목을 쉽게 나열하는 리스트를 구현할 수 있다. • 리스트에서 부분 참조 방법, 이를 이용한 수정, 리스트 연결, 삽입과 삭제 그리고 리스트 컴프리헨션 등을 구현할 수 있다. • 수정할 수 없는 다양한 종류의 항목 나열을 쉽게 처리하는 튜플을 구현할 수 있다.
미리읽어오기	교재 5장 배열과 리스트
과제,시험,기타	도전 프로그래밍
6주차	[6주]
학습주제	6장 키와 값의 나열인 딕셔너리와 중복을 불허하는 집합
목표및 내용	<ul style="list-style-type: none"> • 키와 값의 쌍인 항목을 관리하는 딕셔너리를 생성하고 수정하는 방법을 이해하고, 다양한 방법으로 딕셔너리를 구현할 수 있다. • 집합의 특징을 이해하고, 합집합 등과 같은 다양한 집합의 연산을 구현할 수 있다. • 내장 함수 zip()과 enumerate(), 시퀀스 간의 변환을 이해하고, 구현할 수 있다.
미리읽어오기	교재 6장 집합
과제,시험,기타	도전 프로그래밍
7주차	[7주]
학습주제	7장 특정 기능을 수행하는 사용자 정의 함수와 내장 함수
목표및 내용	<ul style="list-style-type: none"> • 함수의 내용과 필요성을 이해하고 함수를 직접 정의해 호출할 수 있다. • 인자의 기본 이해와 기본값 지정, 가변 인수와 키워드 인수를 활용할 수 있다. • 간편한 람다 함수와 표준 설치된 내장 함수를 사용할 수 있다.
미리읽어오기	교재 7장 함수의 정의와 호출
과제,시험,기타	도전 프로그래밍
8주차	[중간고사]
학습주제	- 직무수행능력평가 1차(중간고사)
목표및 내용	직무수행능력평가, 서술형 평가
미리읽어오기	교재 1장에서 7장 까지
과제,시험,기타	
9주차	[9주]
학습주제	8장 조건과 반복, 리스트와 튜플 기반의 미니 프로젝트 I
목표및 내용	8개의 미니 프로젝트를 스스로 생각하고 프로그래밍해 코딩 능력뿐 아니라 문제 해결 능력을 기를 수 있다.
미리읽어오기	교재 8장
과제,시험,기타	

10 주차	[10주]
학습주제	9장 라이브러리 활용을 위한 모듈과 패키지
목표및 내용	<ul style="list-style-type: none"> 표준 모듈을 이해하고 사용자 정의 모듈도 직접 구현해 사용할 수 있다. 표준 모듈인 turtle을 사용해 기본적인 도형을 그릴 수 있다. 썬드파티 모듈 numpy와 matplotlib 등을 설치해 활용할 수 있다.
미리읽어오기	교재 9장
과제,시험,기타	도전 프로그래밍
11 주차	[11주]
학습주제	10장 그래픽 사용자 인터페이스 Tkinter와 Pygame
목표및 내용	<ul style="list-style-type: none"> GUI를 이해하고 GUI 표준 모듈인 Tkinter를 사용해 필요한 위젯을 구성하고 윈도우를 생성할 수 있다. 이벤트 처리를 이해하고 Tkinter에서 이벤트 처리를 구현할 수 있다. 썬드파티 GUI 모듈인 pygame을 설치해 기본적인 윈도우를 구현할 수 있다.
미리읽어오기	교재 10장
과제,시험,기타	도전 프로그래밍
12 주차	[12주]
학습주제	11장 실행 오류 및 파일을 다루는 예외 처리와 파일 입출력
목표및 내용	<ul style="list-style-type: none"> 예외 처리의 필요성을 이해하고 try except 구문을 사용해 예외를 처리할 수 있다. 프로그램에서 파일을 생성하는 필요성을 이해하고 필요한 파일을 만들 수 있다. 이미 생성된 파일에서 내용을 읽어 처리할 수 있다
미리읽어오기	교재 11장
과제,시험,기타	도전 프로그래밍
13 주차	[13주]
학습주제	12장 일상생활의 사물 코딩인 객체지향 프로그래밍
목표및 내용	<ul style="list-style-type: none"> 객체와 클래스를 이해하고 필요한 클래스를 정의하고 객체를 만들어 활용할 수 있다. 클래스 속성과 인스턴스 속성, 정적 메소드와 클래스 메소드를 이해하고 정의할 수 있다. 상속을 이해하고 부모 클래스와 자식 클래스를 정의할 수 있다. 추상 메소드와 추상 클래스를 이해하고 정의할 수 있다
미리읽어오기	교재 12장
과제,시험,기타	도전 프로그래밍
14 주차	[14주]
학습주제	13장 GUI 모듈과 객체지향 기반의 미니 프로젝트 II
목표및 내용	학습한 파이썬 문법 구조와 프로그래밍 기법을 활용해 8개의 미니 프로젝트를 스스로 생각하고 프로그래밍해 코딩 능력뿐 아니라 문제 해결 능력을 키울 수 있다.
미리읽어오기	교재 1장
과제,시험,기타	

15 주차	[기말고사]
학습주제	직무수행능력평가 2차(기말고사)
목표및 내용	직무수행능력평가, 서술형평가
미리읽어오기	8장에서 13장까지
과제,시험,기타	
수업지원 안내	장애학생을 위한 별도의 수강 지원을 받을 수 있습니다. 언어가 문제가 되는 학생은 글로 된 과제 안내, 확대문자 시험지 제공 등의 지원을 드립니다.

목차

1. 파이썬 언어의 개요
2. 파이썬 기초
3. 파이썬의 문자열과 논리연산
4. 조건과 반복
5. 리스트와 튜플
6. 딕셔너리

제1장 파이썬 언어의 개요

●파이썬 언어란?

파이썬은 1991년 네덜란드의 귀도 반 로섬이 개발한 인터프리터 언어이다.파이썬의 사전적 의미는 '비단뱀'으로 그리스 신화에서 유래가 되었다. 현재 파이썬은 전 세계적으로 가장 많이 가르치는 프로그래밍 언어중 하나이다. 여담으로 구글의 소프트웨어 50%이상이 파이썬으로 개발되었다.

●파이썬 장점

- 문법이 쉽고 순서가 영어 구문과 유사하여 빠르게 배울 수 있음
- 풍부한 라이브러리로 개발 생산성이 매우 높음
- 멀티패러다임 프로그래밍
- 다양한 플랫폼에서 사용가능
- 쓰레딩 대신 단일 이벤트 루프를 사용해 소수 유닛에서 작업하는 비동기식 코드 작성에 유리
- 메모리 자동 관리
- 다른 언어로 작성된 모듈을 연결하는 Glue Language 로 이용
- 웹 프로그래밍, 수치연산 프로그래밍, 데이터베이스 프로그래밍등 다양한 분야 개발 가능

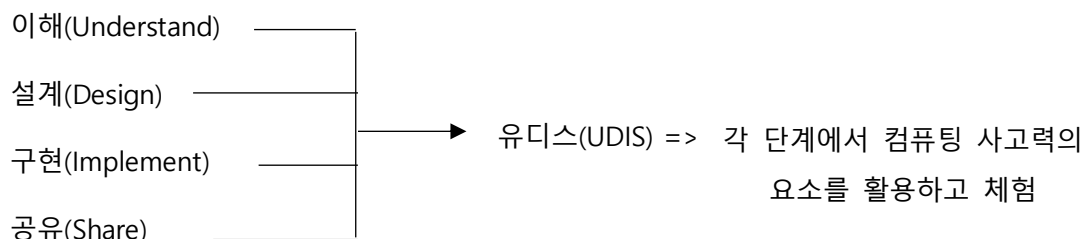
●파이썬 단점

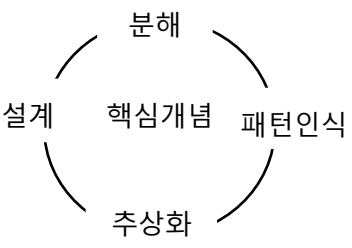
- 인터프리터 언어는 속도가 느림.
- 모바일 컴퓨팅과 브라우저 부재.

●컴퓨팅 사고력

컴퓨팅 사고력은 간단히 말해 컴퓨팅의 기본 개념과 원리를 기반으로 문제를 효율적으로 해결하는 사고 능력이라 할 수 있다.이러한 컴퓨팅 사고력을 향상시키는 방법 중 하나는 직접 프로그래밍하는 코딩교육이고,코딩 교육은 전문 프로그래머뿐만 아니라 문제해결능력, 창의.융합 사고능력 등을 향상시킬수있어 중요성이 커지고 있음.

●파이썬 프로그래밍 학습이유



코딩절차	컴퓨팅 사고력 요소 활용	내용
이해		-주어진 문제를 이해,파악 -분할: 문제를 좀 더쉬운 작은문제로 분해
설계		-패턴인식: 분해된 문제들의 유사성,패턴분석 -추상화 : 문제에 불필요부분제거,핵심만 -알고리즘 :언어에 맞게 입력 ,출력,변수구성
구현		-문제해결을 위해 파이썬 코드개발 -작성된 코드를 테스트하며 디버깅
공유	협업과 평가	-다른이의 프로그램과 비교 -교수의 피드백 및 평가

●활용 분야

대표적으로 흔히 아는 인공지능 구현 , 빅데이터 처리분야에 활용되고 그외에 머신러닝 , 자연어 처리 , 음성인식, 영상 처리, 사물 인터넷 , 웹개발 등등에 활용된다.

제2장 파이썬 기초

●문자열 표현

문자 하나 또는 문자가 모인 단어나 문장 또는 단락등을 문자열 이라한다.

■함수 print()는 문자열이나 숫자등의 자료를

콘솔에 출력

■함수 print()는 출력이후 다음줄로 이동후

다음출력 준비

```
>>print("Hello World!")
```

```
Hello World!
```

```
>>print('Hello Python!')
```

```
Hello Python!
```

●문자열 연산자 +,*

■ + : 문자열 연결 , * : 문자열 반복횟수

```
>>print('Hello World! ' + '좋은아침!')
```

```
Hello World! 좋은아침!
```

```
>>print('안녕! ' * 3)
```

```
안녕! 안녕! 안녕!
```

●주석처리

코드의 문법과 상관없는 파일이름이나 소스 설명들을 담게 해주는 것을 주석이라한다.

인터프리터는 주석을 무시한다.

-주석표시 : # , ' ' ' "

""주석처리된 문장""	
Print('#이후는 주석')	
Print(String : "파이썬")	
Print("숫자 : 1 5 3.14")	
String : "파이썬"	결과
숫자 : 1 5 3.14	

●정수와 실수

■15, 7, 12 와 같은 수는 정수 | 3.15 , 342.12 와 같은 수는 실수

●정수와 실수의 연산

■연산자 종류

+ : 더하기

- : 빼기

* : 곱하기

/ : 나누기(실수)

// : 나누기(정수)

% : 나머지

** : 지수승

■연산자 우선순위

1 순위 : **

2 , 4 순위 : + , -

3 순위 : * , / , % , //

●최근 결과값 저장장소 ()

```
>>>60
```

```
60
```

```
>>>_
```

```
60
```

→ 마지막 실행값 저장

●자료형 type()함수

-해당 매개값의 자료형을 알려줌

```
Type(3) -> <class 'int'>
```

```
Type(3.14) -> <class 'float'>
```

●함수 input()

함수 input()은 표준입력 o, ㄹ 문자열로 읽어 반환하는 함수

```
>>>input()
```

```
Java
```

```
'java'
```

```
>>>pl = input()
```

```
파이썬
```

```
>>>Print(pl)
```

```
파이썬
```

●진수 변환함수 bin() , oct() , hex()

```
data = 22
print('2진수: ',bin(data))
print('8진수 : 'oct(data))
print('10진수 : 'data)
print('16진수 : 'hex(data))
```

```
2진수 : 0b10110
```

```
8진수 | 0o26
```

```
10진수 : 22
```

결과

제3장 문자열과 논리연산

●문자열 str

문자열은 '문자의 나열' = 텍스트시퀀스. 문자열의 자료형은 class str이다.

■함수 len()으로 문자열의 길이 참조

```
>>>len('파이썬')
```

```
3
```

■문자열의 문자참조

문자열에 인덱스를 기술했을 때 참조가 가능

```
>>>'python'[0]
```

```
'p'
```

```
>>>'python'[3]
```

```
H
```

```
>>>'python'[-1]
```

```
N
```

■슬라이싱

- 0과 양수이용 : [start : end]로 start첨자에서 end-1까지 반환

```
>>>'python'[1:5]
```

```
'ytho'
```

-음수 이용

```
>>>'python'[-4 : -1]
```

```
'tho'
```

-0,양수,음수 혼합

```
>>>'python'[-6:4]
```

```
'pyth'
```

-start , end 구성 문자열 없는 경우

```
>>>'python'[5:-1]
```

```
' ' -> (빈 문자열)
```

-start와 end 공백처리 : 처음부터와 끝까지를 의미

```
>>>'python'[:3]
```

```
'pyt'
```

```
>>>'python'[1: ]
```

```
'ython'
```

■ 문자열의 최대 최소

-인자의 최대값과 최소값을 반환하는 함수. 문자열이면 코드값으로 최대 최소 판별

```
>>>min('python')
```

```
'h'
```

```
>>>max('python')
```

```
'y'
```

```
>>>min(2342)
```

```
'2'
```

```
>>>max(5432)
```

```
'5'
```

● 문자열 바꿔 반환하는 메소드 replace()

-replace(a,b) 에서 a의 값을 b값을 바꿔준다.

```
>>>str = '자바는 재밌다'
```

```
>>>str.replace('자바는','파이썬은')
```

```
'파이썬은 재밌다'
```

● 부분 문자열 출현 횟수 반환 메소드 count()

```
>>>str = '단순한 것이 복잡한 것보다 낫다'
```

```
>>>str.count('것')
```

1

●문자열 찾는 메소드 find() , index()

-find는 없는값일경우 -1 반환 index는 오류

```
>>>str = '자바 c 파이썬 코틀린'
```

```
>>>str.find('자바')
```

0

```
>>>str.find('c++')
```

-1

●문자열 나누는 split()

-split()은 문자열에서 공백을 기준으로 문자열을 나눠준다.

●format()메소드로 문자열 간결출력처리

-문자열 중간에 변수나 상수를 함께 출력할 때 format을 호출해야한다

```
>>>str = '{} + {} = {}'.format(3,4, 3 + 4)
```

```
>>>print(str)
```

3 + 4 = 7

●논리 연산자

&(and) : 두 수가 1이면 1, 하나라도 1이아니면 0을 반환

| (or) : 두 수중 하나가 1이면 1반환, 둘다 0 이면 0반환

^ (not) :두 수가 0 이거나 1이면 0반환

제4장 조건과 반복

●조건을 따라 선택하는 if

```
weather = '화창'
if weather == '화창':
    Print('어제 산 신발 신고가야지')
```

if 논리 표현식 :

문장1

문장2

●다중 택일 구조 elif 와 else

```
weather = '화창'
if weather == '화창':
    Print('어제 산 신발 신고가야지')
elif weather == '비':
    print('비가온다')
else:
    print('안나간다')
```

if 논리 표현식 :

문장1

문장2

elif 논리 표현식2:

문장1

else :

문장:

●반목문

반목 조건에 따른 반복

```
while 반복조건:
    반목몸체 문장들
else :
```

시퀀스 항목마다 반복

```
for 변수 in 시퀀스:
    반목몸체 문장들
else :
```

-else는 마지막에 반복이 마치고 실행되는 문장임

●내장함수 range()

```
for l in range(5):
```

```
    print( l , end= ' ')
```

(실행 결과)

```
0  1  2  3  4
```

●임의의 난수 , 반복제어 문

-파이썬은 모듈 random 의 함수 randint(시작 , 끝)으로 임의의 난수 발생가능

```
>>>import random
```

```
>>>random.randint(1,5)
```

```
4
```

```
>>>randomradint(1,5)
```

```
1
```


●반복제어 break 와 continue

while의 논리 표현식이 True라면 무한반복이 실행

break는 해당 반복문 for나 while을 else를 실행시키지 않고 무조건 종료한다.

continue는 이후 반복문체를 실행하지 않고 반복을 위해 논리 조건을 수행

제5장 리스트와 튜플

●관련 나열 항목 관리하는 리스트

-하나의 단위로 여러항목을 묶는 것

Ex) menu = ['coffee','beverage','ade']

coffee = ['에스프레소','아메리카노','카페라테','카페모카']

```
coffee = ['에스프레소','아메리카노','카페라테','카페모카']
print(coffee)
print(type(coffee))
num = 0
for s in coffee:
    num += 1['BTS','볼빨간사춘기','BTS','블랙핑크','태연','BTS']
print('%d. %s' %(num , s))
```

```
['에스프레소','아메리카노','카페라테','카페모카']
```

결과

```
<class 'list'>
```

```
1.에스프레소
```

```
2.아메리카노
```

```
3.카페라테
```

```
4.카페모카
```

●빈 리스트에 항목추가

```
>>>pl = list()
```

```
>>>pl.append('c++')
```

```
>>>pl.append('java')
```

```
>>>print(pl)
```

```
['c++','java']
```

●리스트의 항목 참조

-첨자는 척 요소가 0부터 시작

-마지막 요소 첨자는 역순으로 -1시작

```
>>>py = list('python')
```

```
>>>print('py[0],py[5])
```

```
p n
```

●리스트의 첨자로 항목수정

-리스트의 인덱스로 대입연산자를 통해 해당 항목을 수정할수있다

```
>>> top = ['BTS','볼빨간사춘기','BTS','블랙핑크','태연','BTS']
```

```
>>>top[1] = '장범준'
```

```
>>>top[3] = '잔나비'
```

```
>>>print(top)
```

```
['BTS','장범준','BTS','잔나비','태연','BTS']
```

●첨자3개로 리스트 일부분을 참조하는 슬라이싱

-리스트[start:stop:step]

```
>>> alp = list('abcdefghij')
```

```
>>>print(apl)
```

```
['a','b','c','b','d','e','f','g','h','i','j']
```

```
>>>len(apl)
```

```
>>>print(alp[::-1])
```

```
['j','l','h','g','f','e','d','c','b','a']
```

```
>>>print(alp[1:10:2])
```

```
['b','d','f','h','j']
```

```
>>>print('alp'[-3::-2])
```

```
['h','f','d','b']
```

●리스트의 슬라이스로 리스트 일부분 수정

```
>>>sports = ['풋살','족구','비치사커','야구','농구','배구']
```

```
>>>sports[0:3] = ['축구']
```

```
print(sports)
```

```
['축구','야구','농구','배구']
```

●리스트 메소드 insert , remove , pop

-리스트의 첨자위치로 insert(첨자,항목)을 이용해 항목삽입가능. , remove(항목), pop(첨자)로 삭제가능

```
>>>tool = ['이클립스','안드로이드','아톰']
```

```
>>>tool.remove('아톰')
```

```
>>>print(tool.pop(1))
```

```
안드로이드
```

```
>>>tool.insert(0,'파이참')
```

```
>>>tool.insert(0,'VS')
```

```
>>>print(tool)
```

```
['VS','파이참','이클립스']
```

●튜플

-튜플은 문자열 , 리스트와 같은 항목의 나열. 튜플은 리스트와 달리 수정불가.

튜플은 각각의 항목을 콤마로 구분하고 정수 실수 문자열 리스트 튜플등 제한없음

■튜플 생성

```
>>> ex = ()

>>> aaa = (15 , '파이썬')

>>> print(ex)

() ---> 빈 튜플을 나타냄

>>> print(aaa)

(15 , '파이썬')
```

■튜플 참조

-첨자 참조로 참조가능 , 하지만 첨자와 슬라이스로 수정은 불가

```
>>> tool = ('이클립스','안드로이드','파이참')

>>> tool[2]

('파이참')
```

■튜플 연결 , 반독연산자 , 순서정렬한수 sorted()

```
>>> tool = ('파이참','아톰', '이클립스')

>>> tool2 = ('VS')

>>> print(tool * 3)

('파이참','아톰', '이클립스','파이참','아톰', '이클립스','파이참','아톰', '이클립스')

>>> print(sorted(tool,reverse=True))    ---> true : 역순으로 정렬

['아톰','이클립스','파이참']

>>> print(tool + tool)

('파이참','아톰', '이클립스','VS')
```

제6장 딕셔너리

● 딕셔너리 개요

-딕셔너리는 키와 값을 가지고있다.딕셔너리를 사전이라 하면 키는 단어 , 값은 해설이라 볼수 있다. 첨자 참조로 해당키의 값을 참조할수 있고 키가 단순 문자열이면 "없이 기술할수 있다.

```
>>>example = { <key> : <values>,<key> : <values>,...} # 딕셔너리
```

```
>>>aaa = {} #빈 딕셔너리
```

```
>>>aaa['강좌명'] = '파이썬 기초';
```

```
{'강좌명' : '파이썬 기초'}
```

```
>>>print(aaa['강좌명'])
```

파이썬기초

■ 딕셔너리 메소드 values()

-값으로 구성된 리스트를 반환

```
>>> tool = dict(파이참 = '파이썬', phpstorm = 'php')
```

```
>>>print(tool.values())
```

```
dict_values(['파이썬','php'])
```

■ 키로 조회하는 딕셔너리 메소드 get()

```
>>>tool = {'파이참':'파이썬','phpstorm':'php'}
```

```
>>>tool.get('파이참')
```

'파이썬'

■ 임의의 항목을 삭제하는 딕셔너리 메소드 popitem()

-popitem()은 임의의 키,값의 튜플을 반환하고 삭제한다. 데이터가 하나도없을경우 오류가 발생

■문장 del로 딕셔너리 항목삭제

```
>>>tool = {'파이참':'파이썬','phpstorm':'php'}
```

```
>>>del tool['phpstorm'] ➔ 첨자참조를 쓰지않을경우 변수 자체를 제거해 참조불가능
```

```
>>>tool
```

```
{'파이참':'파이썬'}
```

■ 모든 한목제게 메소드 clear()

```
>>> tool = {'파이참': '파이썬', 'phpstorm': 'php'}
```

```
>>> tool.clear()
```

```
>>> tool
```

```
{}
```

■ 딕셔너리를 결합하는 메소드 update()

- 다른 딕셔너리와 합병함.

```
>>> tool = {'파이참': '파이썬', 'phpstorm': 'php'}
```

```
>>> tool2 = {'VS' : 'c'}
```

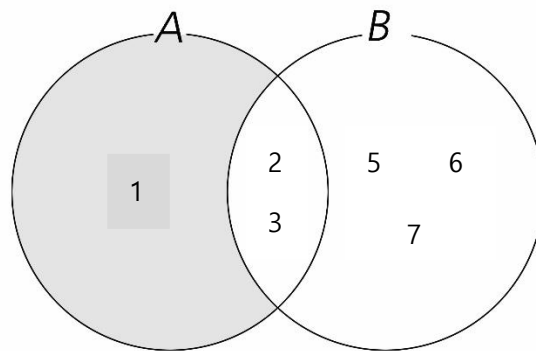
```
>>> tool.update(tool2)
```

```
>>> tool
```

```
{'파이참': '파이썬', 'phpstorm': 'php', 'VS' : 'c'}
```

● 집합

- 집합은 중복되는 요소가 없고 순서도 없는 원소의 모임이다.



$$A = \{1, 2, 3\} \quad B = \{2, 3, 5, 6, 7\}$$

- 파이썬에서의 집합

```
S1 = {1,2,3,4,5}
```

```
S2 = {'py','java','go'}
```

```
S3 = {(1,1),(2,3)}
```

■함수 set()

-집합은 내장함수 set()으로 생성가능

Set(원소로 구성된 리스트 , 튜플 , 문자열)

-인자가 없으면 빈 집합인 공집합 생성

-인자가 있으면 하나만 올수있다.

-리스트나 튜플을 인자로 사용할수 있다

```
>>>set([1,2,3])
```

```
{1,2,3}
```

```
>>>set((1,2,2))
```

```
{1,2}
```

-수정가능한 딕셔너리와 리스트는 집합의 원소로 사용불가

■집합 메소드 add()로 추가

```
>>>odd = {1,3,5,6}
```

```
>>>odd.add(8)
```

```
>>>print(odd)
```

```
{1,3,5,6,8}
```

■remove() , discard(),pop()으로 항목삭제

```
>>>odd = {1,3,5,6}
```

```
>>>odd.remove(3) ➔ odd.discard(3)과 동일 , 하지만 없는 원소값을 지정해도 오류는 안남.
```

```
>>>odd.remove(9) ➔ 오류발생
```

```
>>>print(odd)
```

```
{1,5,6}
```

```
>>>print(odd.pop()) ➔임의의 원소 삭제
```

```
1
```

```
>>>print(odd)
```

```
{5,6}
```

■clear() 로 모든원소삭제

```
>>>odd = {1,3,5,6}
```

```
>>>odd.clear()
```

```
>>>print(odd)
```

```
Set()
```

■합집합 연산자 | 와 union()

-원소를 합하는 합집합은 연산자 | 와 union()을 사용

```
>>> a = {4,5,6}
```

```
>>> b = {3,6,9}
```

```
>>> a | b
```

```
{3,4,5,6,9}
```

```
>>>a.union(b)
```

```
{3,4,5,6,9}
```

■교집합 연산자 & 와 intersection()

```
>>> a = {4,5,6}
```

```
>>> b = {3,6,9}
```

```
>>> a & b
```

```
{6}
```

■여집합 연산자 ^ 와 symmetric_difference()

-집합이지만 교집합이아닌 원소로 구성되는 여집합은 대칭 차집합이라고 부름

```
>>>a ^ b
```

```
{3,4,5,9}
```

```
>>>a.symmetric_diffenrence(b)
```

```
{3,4,5,9}
```


후기

파이썬으로 현재까지 배운 내용들을 정리하는 시간을 가지게 되었는데 처음에는 그저 이 많은 분량을 왜 요약 정리하라고 하셨을까 별로 도움이 되는 것 같지는 않은데 라는 생각을 가지고 글을 써내려갔다. 하지만 내 생각과는 다르게 이러한 행동이 무의미한 것은 아니였고 오히려 요약 정리함으로써 강의시간에 제대로 듣지 못하고 넘어갔던 부분을 요약정리하면서 되새기며 알게되고 예제 코딩들을 보면서 사용법을 익히게 되었다. 파이썬은 요즘 전세계적으로 대두되고있는 언어중에 하나이고, 인공지능 , 빅데이터 등등 여러분야에 사용되는 언어이므로 그 전망이 밝다고 볼 수가 있다.그래서 파이썬은 프로그래머라 한다면은 배우지 않을래야 않을수 없는 언어인 것 같다. 따라서 빠르게 변하고 있는 시대에 맞추어 나는 이 파이썬을 나중에 더 공부해볼 생각이고 , 이번에 처음으로 배운 파이썬은 기존 언어들과 달리 인터프리터 언어라서 조금 색다른 느낌도 들었다.

참고 서적

파이썬으로 배우는 누구나 코딩(강환수 , 신용현 지음)