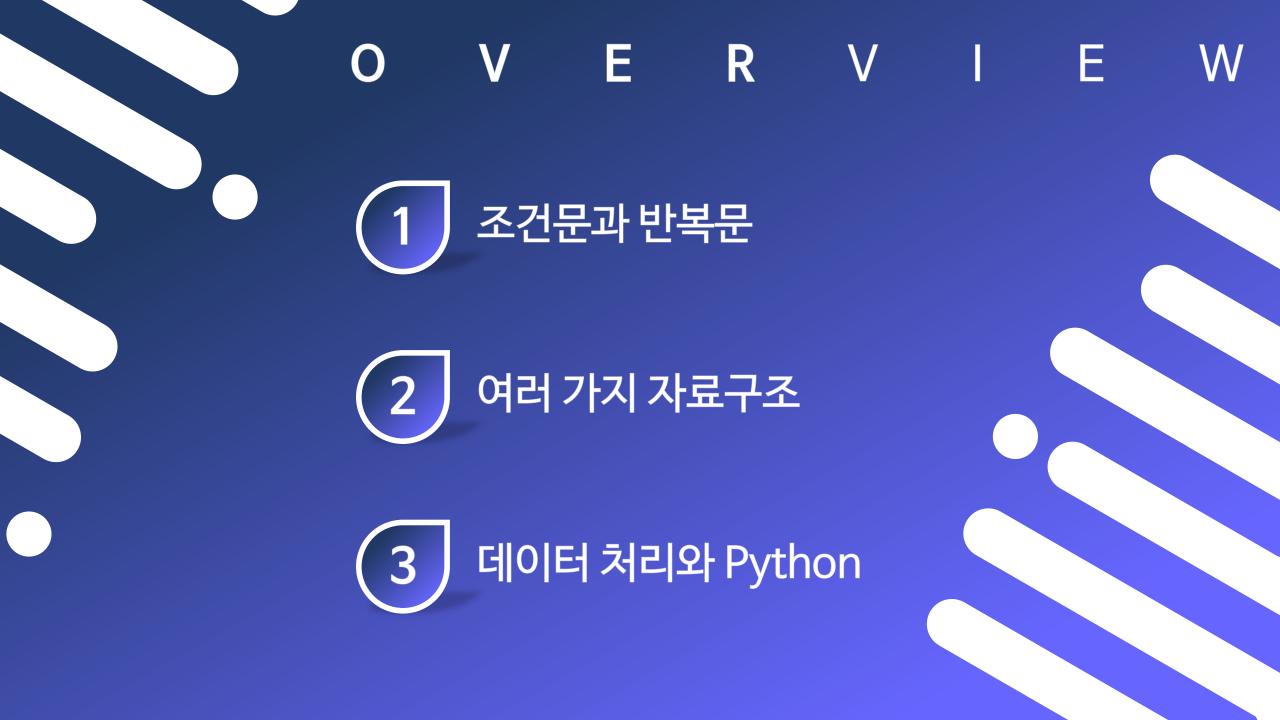


Python 실습 I 기초 문법과 데이터 처리

TA 유용재 yooy@dm.snu.ac.kr

2021학년도 봄학기 〈산업컴퓨팅개론〉 과제 안내

- 매 Python 실습마다 100점 만점의 과제 부여
- 총 5회 실습, 5차례 과제 배부, 과제당 1~2문항
- 자동 채점되므로 입출력 조건에 각별히 유의
- Grading 반영 비율 25%
- Delay Penalty는 마감 이후 당일 제출 20%, 익일 제출 50%, 이후 제출 100%







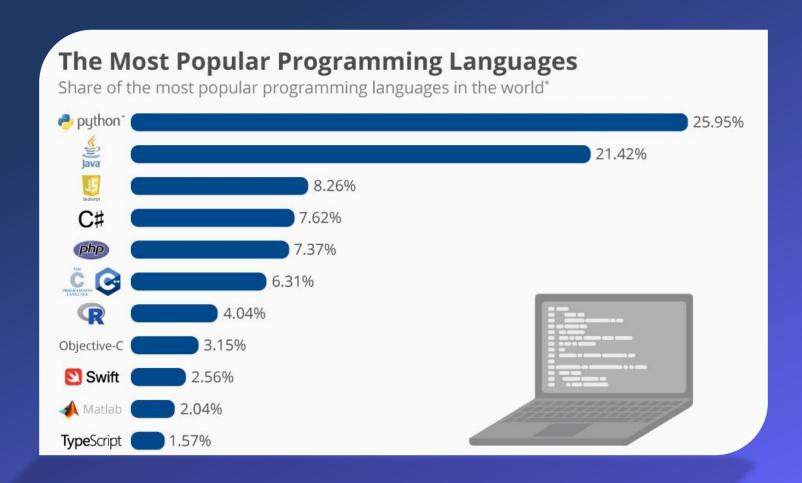
 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 다른 프로그래밍 언어에 비해 **Python이 가지는 강점**은?





- ✓ 간편하게 다룰 수 있는 자료형
- ✓ 강력한 라이브러리 기능
- ✓ 다수의 사용자, 높은 범용성
- ✓ 배우기 쉬운 직관적 문법

Theme 2 여러 가지 자료구조

데이터 처리와 Python

Theme 3

Python 첫걸음, 주석 작성과 기본 연산





Python에서 주석 작성하기

THIS IS COMMENT

◀ 주석 처리되었으므로 코드 실행에 영향을 주지 못함



기본 연산자

2 + 6

5 * 1

4 * 7

9 / 3

 Python 실습 I

 기초 문법과 데이터 처리

 Theme 1

 조건문과 반복문

 Theme 2

 여러 가지 자료구조

데이터 처리와 Python

Theme 3

○ 상황에 따라 사용할 수 있는 **다양한 연산자**





몫과 나머지를 구하는 연산자

3 % 2 10 // 3 나머지를 구하기 위한 % 연산자 몫을 구하기 위한 // 연산자



거듭제곱 값을 구하는 연산자

2 ** 5

◀ 거듭제곱 값을 구하기 위한 ** 연산자

Theme 2 여러 가지 자료구조

데이터 처리와 Python

Theme 3

🔍 **변수**를 만들어 값을 저장하고, **입력과 출력**을 수행하기





변수의 선언과 활용 및 출력

num = 3num = num + 2

- ◀ num이라는 변수에 처음으로 3을 저장
- ◀ num에 2를 더하고 다시 num에 덮어씌우기
- print('three plus two equals to', num) ◀ print 함수를 이용해 문자와 변수를 출력



사용자로부터 입력받기

food = input('I want to eat...:') ◀ 입력 칸에 적은 값은 food에 저장됨

 Theme 1

 조건문과 반복문

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 기초 문법의 핵심, **조건문과 반복문**이란?

Conditional Statements

특정 조건이 성립하는 경우에만 동작이 이루어지도록 하는 구문

* if - elif - else

Repetitive Statements

원하는 동작을 여러 차례 반복하게끔 만드는 구문

* while / for

Theme 2 여러 가지 자료구조

Theme 3 데이터 처리와 Python

🔾 Python에서 **조건문**의 일반적인 문법

```
if (gudok >= 100000000):
    print('get this DIAMOND button!')
elif (gudok >= 10000000):
    print('get this GOLD button!')
elif (gudok >= 1000000):
    print('get this SILVER button!')
else:
    print('cheer up!')
```



 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 조건문을 활용한 예제, **잘못된 부분을 모두 찿으면?**





구독자 수가 정확히 777명이 되면 경품 증정!

```
gudok = input('당신의 구독자 수는?')

if (gudok = 777):
  print('축하합니다! 경품 당첨! 39동에서 수령하세요.')
else
  print('아쉽습니다. 다음 기회에...')
```

 Theme 2

 여러 가지 자료구조

데이터 처리와 Python

Theme 3

○ 입출력에서 반드시 유념! Data Type Conversion





입력받은 문자를 숫자로 변환

gudok = input('당신의 구독자 수는?') gudok = int(gudok)

◀ 문자형이었던 gudok이 정수형으로 바뀌게 됨



정수형과 실수형 데이터의 변환

num = 3 * 4
num = float(num)
print(num)

◀ 정수였던 num이 실수로 바뀌게 됨

 Theme 2

 여러 가지 자료구조

데이터 처리와 Python

Theme 3

◯ Python에서 **반복문**의 일반적인 문법

```
num = 1
while (num < 10):
    print(num)
    num = num + 1</pre>
```

- ✓ 반복문을 구성하는 요소는?
- ✓ 조건문과 반복문의 공통점은?
- ✓ 반복문에서 주의해야 할 점은?

◯ 보다 직관적으로 반복문을 쓰기 위한 전략, while (True)





퀴즈의 정답을 맞힐 때까지 무한정 입력받기

```
while (True):
    ans = input('산업공학과는 서울대학교 관악캠퍼스 몇 동에 있을까요?')
    if (ans == '39'):
        print('정답! 짝짝짝')
    else:
        print('세상에, 다시 풀어보세요...')
```

 Theme 2

 여러 가지 자료구조

데이터 처리와 Python

Theme 3

○ 반복문 제어의 핵심, 동작을 멈춰주는 break



-

반복을 멈추고 코드 밖으로

```
while (True):
    ans = input('산업공학과는 서울대학교 관악캠퍼스 몇 동에 있을까요?')
    if (ans == '39'):
        print('정답! 짝짝짝')
        break
    else:
        print('세상에, 다시 풀어보세요...')
```

Exercise 1

실시간 Zoom 수업에서 출제 및 풀이 예정



2 여러 가지 자료구조

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 변수만 알면 끝? 자료구조가 필요한 순간





올해 서울대학교 신입생 3,000명 수능 성적 좀 정리하고 싶은데… student1_score = 298

student2_score = 290

student3_score = 295

student4_score = 300

student5_score = 289

student6_score = 292

student7_score = 279

student8_score = 288

•

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 가장 간편하면서도 강력한 자료구조 **list**



list

- 다수의 데이터를 간편하게 다룰 수 있게 하는 자료구조
- 여러 원소들을 대괄호로 한데 묶어 정의
- 원소를 자유롭게 추가하거나 삭제할 수 있음
- 리스트에 어떤 원소가 존재하는지 여부를 쉽게 검색할 수 있음
- 리스트 내의 첫 번째 원소에 붙는 index는 0임에 유의

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 리스트를 **새로 정의**하고, **원소를 추가**하기





list를 생성하고 원소를 추가

dooly = ['도우너', '고길동', '마이콜'] ◀ 새로운 리스트 dooly가 정의됨 dooly.append('희동이') ◀ dooly의 맨 마지막 원소로 희동이가 추가됨



원하는 위치에 원소를 추가하기

dooly = ['도우너', '고길동', '마이콜'] dooly.insert(2, '희동이') ◀ index 2 위치에 희동이가 추가됨

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 리스트의 특정 위치 원소를 **확인**하고, 또 **삭제**하기





list에서의 indexing

dooly = ['도우너', '고길동', '마이콜'] print(dooly[1])

◀ index 1 위치에 존재하는 원소가 출력됨



위치를 아는 list의 특정 원소 없애기

dooly = ['도우너', '고길동', '마이콜'] del dooly[2]

◀ index 2 위치에 존재하는 원소가 삭제됨

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 리스트의 강점 하나, **원소가 어디 있는지 몰라도**···



위치를 모르지만 값을 아는 list의 특정 원소 없애기

dooly = ['도우너', '고길동', '마이콜'] dooly.remove('마이콜')

◀ 맨 처음으로 등장하는 마이콜이 삭제됨

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 문자열도 리스트처럼! 문자열에서의 indexing



강서구, 진관동, 서울시, 서대문구, 대학동, 강남구, 가리봉동

- 글자 '강'으로 시작하는 행정구역은 모두 몇 개?
- 글자 '구'로 끝나는 행정구역은 모두 몇 개?

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 주어진 리스트에서 **특정 원소가 존재하는지 찿기**



ب

list에서의 원소 검색

```
dooly = ['도우너', '고길동', '마이콜', '희동이']
if ('고길동' in dooly):
print('Welcome to Dooly World!')
```

◀ 원소 존재 여부를 찿아주는 in

list에서의 원소 index 찿기

```
dooly = ['도우너', '고길동', '마이콜', '희동이']
dooly.index('고길동')
```

◀ 원소 위치를 반환하는 index

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 앞으로의 코딩에서 유용하게 쓰일, **list의 길이 파악**





list의 길이를 반환하기

song_name = '어떻게이별까지사랑하겠어널사랑하는거지' len(song_name)

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ list만으로 우리가 원하는 **모든 자료를 표현할 수 있나?**



dictionary

- key와 value가 한 쌍이 되어 존재하는 형태의 자료구조
- 각각의 원소를 구분짓게 해주는 key, 그에 대응하는 값인 value
- list가 대괄호라면, dictionary는 중괄호로 한데 묶어 정의

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ dictionary를 **새로 정의**하고, **원소를 추가**하기





새로운 dictionary를 생성

score = {'김철수' : 294, '이영희' : 295, '박민수' : 294}



dictionary에 새 원소 추가하기

score = {'김철수' : 294, '이영희' : 295, '박민수' : 294} score['최영수'] = 290

◀ 괄호에 유의

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

○ 결국 핵심은 key, dictionary에서 원소를 찿으려면?



<u>ب</u>

특정 key 값을 가지는 원소의 존재 여부 확인

```
score = {'김철수': 294, '이영희': 295, '박민수': 294} if ('이영희' in score): print('SERACH FINISHED')
```



특정 key 값에 대응하는 value 찾기

```
score = {'김철수' : 294, '이영희' : 295, '박민수' : 294} score['이영희']
```

◀ 상응하는 value를 찾음

 Theme 2

 여러 가지 자료구조

Theme 3 데이터 처리와 Python

Q Python에서 사용할 수 있는 기타 자료구조 정리

tuple

- 여러 개의 자료를 한데 묶어 저장할 수 있음
- 자료에 index가 존재한다는 점도 list와 동일
- 한 번 정의한 데이터를 재정의할 수 없음

set

- 여러 개의 자료를 순서 없이 저장
- 데이터는 서로 중복될 수 없음
- 수학에서의 집합과 그 개념이 유사

Exercise 2

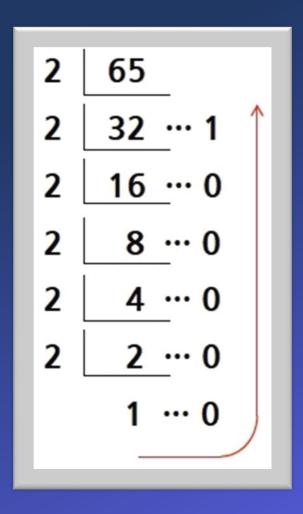
실시간 Zoom 수업에서 출제 및 풀이 예정



3 데이터 처리와 Python

○ 십진수가 주어졌을 때, 이를 **이진수로 어떻게 바꾸나?**





- 십진수를 더 이상 나눌 수 없을 때까지 2로 계속 나눔
- 나머지를 역순으로 읽으면 이진수

조건문과 반복문

🔾 주어진 십진수를 **이진수로 변환하기 위한 Python 코드는?**



십진수가 주어졌을 때 이를 이진수로 변환

```
num = input('십진수를 입력하세요: ')
num = int(num)
binary = []

while (num > 0):
  remainder = num % 2
  binary.insert(0, remainder)
  num = num // 2

print(binary)
```

- ◀ input으로 받은 값은 문자로 취급됨에 유의
- ◀ 이진수 출력을 위해 비어 있는 list를 생성
- ◀ 더 이상 나누어질 수 없을 때까지 반복
- ◀ 나머지가 역순으로 삽입되도록 insert
- ◀ 몫만을 남겨야 함에 유의

조건문과 반복문

○ 이진수가 주어졌을 때, 이를 **십진수로 어떻게 바꾸나?**

맨 마지막 자리는 1, 그 왼쪽 자리는 2, 그 왼쪽 자리는 4…

이진수 10101

차례로 더하면 이진수를 십진수로 변환할 수 있음

조건문과 반복문

\bigcirc 주어진 이진수를 **십진수로 변환하기 위한 Python 코드는?** $\qquad \checkmark$



이진수가 주어졌을 때 이를 십진수로 변환

Assignment #1

실시간 Zoom 수업에서 출제 예정

Input & Output

과제 1 유의 사항

- 규칙을 벗어나는 입력 값은 주어지지 않으므로, 별도의 예외 처리를 하실 필요가 없습니다.
- list의 원소를 모두 지울 정도로 큰 입력 값은 주어지지 않습니다.
- eTL 〈과제 1 제출〉 과제함에 03/31(수) 오후 8시까지 제출해주세요.
- 파일명은 HW1_202012345.ipynb의 형식으로 제출해 주시고, 압축하지 마시기를 바랍니다.
- 제출하신 파일은 자동 채점되며, 맞힌 Test Case 개수에 따라 점수가 달라집니다.
- Test Case에서 list의 원소는 예시와 다를 수 있습니다.
- 모든 종류의 부정 행위는 학칙에 따라 처리됩니다.



Any Questions?

I A

TA TODAY 유용재 yooy@dm.snu.ac.kr 안영훈 younghoon.ahn@dm.snu.ac.kr