파이썬

상명대학교 융합공과대학
지능•데이터융합학부
휴먼지능정보공학전공
dkim@smu.ac.kr

강의개요

- 강의소개 및 프로그래밍 개념
  - 강의 및 프로그래밍 소개
  - 프로그래밍 맛보기
- 변수, 자료형, 연산, 함수
  - 코딩과 기초실습
- 조건문, 연산자
  - 코딩과 기초실습
- 반복문
  - 코딩과 기초실습
- 함수, 매개변수
  - 코딩과 기초실습
- 중간고사

강의개요

- 자료형, 리스트
  - 코딩과 기초실습
- 자료형, 튜플
  - 코딩과 기초실습
- 자료형, 딕션너리
  - 코딩과 기초실습
- 실습예제
  - 코딩과 기초실습
- 파일읽고 쓰기
  - 코딩과 기초실습
- 객체지향 프로그래밍
  - 코딩과 기초실습
- 기말고사

문자열

- 문자열: 자료구조 형태중 하나로 문자 데이터의 나열
  - a="상명대"
    - >>> print(a) → 상명대
  - 0부터 시작하는 인덱스로 접근가능 (리스트와 유사)
    - >>> print(a)
    - >>> print(a[0])
    - >>> print(a[1:2])
    - >>> print(a[:1])
    - →
    - 상명대
    - 상
    - 명
    - 상명
  - 연결할 경우 더하기(+) 사용, 곱하기(*)는 문자열을 반복

문자열

- 문자열: 자료구조 형태중 하나로 문자 데이터의 나열
  - 문자열 함수(대문자/소문자 변환)
  - b="Sangmyung university"
    - upper( ) 함수 : 소문자를 대문자로 변경
      - print(b.upper()) → SANGMYUNG UNIVERSITY
    - lower( ) 함수 : 대문자를 소문자로 변경
      - print(b.lower()) →sangmyung university
    - swapcase( ) 함수 : 대소문자를 상호 변환
      - print(b.swapcase()) → sANGMYUNG UNIVERSITY
    - title( ) 함수 : 각 단어의 제일 앞 글자만 대문자로 변환
      - print(b.title()) → Sangmyung University

문자열

- 문자열: 자료구조 형태 중 하나로 문자 데이터의 나열
  - 문자열 함수(대문자/소문자 변환)
  - b="Sangmyung university"
    - upper( ) 함수 : 소문자를 대문자로 변경
      - print(b.upper()) → SANGMYUNG UNIVERSITY
    - lower( ) 함수 : 대문자를 소문자로 변경
      - print(b.lower()) →sangmyung university
    - swapcase( ) 함수 : 대소문자를 상호 변환
      - print(b.swapcase()) → sANGMYUNG UNIVERSITY
    - title( ) 함수 : 각 단어의 제일 앞 글자만 대문자로 변환
      - print(b.title()) → Sangmyung University

문자열

- 문자열: 자료구조 형태중 하나로 문자 데이터의 나열
  - 문자열 함수(문자열 찾기)
  - c="Sangmyung university! 2021 "
    - count( ) : 찾을 문자열이 몇 개 들었는지 개수를 반환
      - print(c.count("21"))→ 1
    - find( ) : 찾을 문자열이 몇 번째 위치하는지 반환
      - find(찾을문자, 찾기시작할위치)
      - print(c.find("21")) → 24
    - rfind( ) : 오른쪽부터 탐색
      - print(c.rfind("21"),1) → 24

문자열

- 문자열: 자료구조 형태중 하나로 문자 데이터의 나열
  - 문자열 함수(문자열 찾기)
  - c="Sangmyung university! 2021 "
    - index( ) : find() 함수와 동일한 용도, 찾을 문자열이 없다면 오류가 발생함
      - print(c.index("2021")) → 22
    - startswith( ) : 문자열로 시작하면 True를, 그렇지 않으면 False를 반환
      - startswith(시작하는문자, 시작지점)
      - print(c.startswith("21",c.find("21"))) → True
      - print(c.startswith("21",c.find("21")+10)) → False
    - endswith( ) : 문자열로 끝나면 True를 반환
      - endswith(끝나는문자, 문자열의시작, 문자열의끝)
      - print(c.endswith("2021",c.find("2021")))→ False
      - print(c.endswith("2021 ",c.find("2021 "))) → True

문자열

- 문자열: 자료구조 형태중 하나로 문자 데이터의 나열
  - 문자열 함수(제거, 변경)
  - d="*The Best, Sangmyung university! 2021 !“
    - strip(): 문자열 양쪽에 있는 해당 문자를 삭제
      - print(d.strip("*!")) → The Best, Sangmyung university! 2021
    - rstrip(): 문자열 오른쪽에 있는 해당 문자를 삭제
      - print(d.rstrip("!")) → *The Best, Sangmyung university! 2021
    - lstrip(): 문자열 왼쪽에 있는 해당 문자를 삭제
      - print(d.lstrip("*")) → The Best, Sangmyung university! 2021 !
    - replace(): 문자열에서 해당 문자를 특정 문자로 변경
      - d = d.strip("*!")
      - d = d.replace("university","University")
      - print(d) → The Best, Sangmyung University! 2021
      - d = d.strip()
      - print(d) → The Best, Sangmyung University! 2021

문자열

- 문자열: 자료구조 형태중 하나로 문자 데이터의 나열
  - 문자열 구성파악 (True, False 반환)
  - e="SangmyungUniversity2021"
    - isdigit( ) : 전체가 숫자로만 구성되어 있는가
      - print(e.isdigit()) → False
    - isalpha( ) : 전체가 알파벳으로만 구성되어 있는가
      - print(e.isalpha()) → False
    - isalnum( ) : 전체가 알파벳와 숫자가 섞여서 구성되어 있는가
      - print(e.isalnum()) →True
    - islower( ) : 전체가 소문자로만 구성되어 있는가
      - print(e.islower()) → False
    - isupper( ) : 전체가 대문자로만 구성되어 있는가
      - print(e.isupper()) → False
    - isspace( ) : 전체가 공백문자로만 구성되어 있는가
      - print(e.isspace()) →False

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - 접근, 수정, 삭제, 추가가 가능
  - [] 대괄호로 작성되어지며, 내부 원소는 ,로 구분
  - 리스트 이름 = [요소1, 요소2, 요소3,…]
  - a = [1,2,3,4,5]
    - >>> print(a) → [1,2,3,4,5]
  - 0부터 시작하는 인덱스로 접근가능
    - >>> print(a[0]) → 1, print(a[1])→ 2, print(a[2])→ 3, print(a[4])→ 5, print(a[5])→ IndexError: list index out of range
  - 인덱스를 이용하여 수정가능
    - >>> a[0] = “가”
    - >>> print(a) → [“가”, 12,3,4,5]

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - a = []
  - b = [1, 2, 3]
  - c = [‘Red’, ‘Green’, ‘Blue’]
  - d = [1, 2, ‘도’, ‘개’]
  - e = [1, 2, [‘도’, ‘개’,’걸’]]

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - b = [1, 2, 3]
  - print(b)
  - → [1, 2, 3]
  - print(b[1])
  - → 2
  - print(b[1]+b[2])
  - → 5
  - print(b[-1])
  - → 3

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - e = [1, 2, [‘도’, ‘개’,’걸’]]
  - print(e[2][1])
  - → [‘개’]
  - print(e[?][?])
  - → [‘걸’]

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - e = [1, 2, [‘도’, ‘개’,’걸’]]
  - print(e[:2])
  - → 2
  - print(e[2][:2])
  - → [‘도’, ‘개’,’걸’]

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - b = [1, 2, 3]
  - bb = [4, 5, 6]
  - print(b + bb)
  - → [1, 2, 3, 4, 5, 6]
  - print(b * 2)
  - → [1, 2, 3, 1, 2, 3]

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - b = [1, 2, 3]
  - bb = [4, 5, 6]
  - print(len(b))
  - → 3
  - print(len(b) * bb)
  - → [4, 5, 6, 4, 5, 6, 4, 5, 6]

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - b = [1, 2, 3]
  - b[2] = 100
  - print(b)
  - → [1, 2, 100]
  - print(del a[0])
  - → [2, 100]
  - b.append(200)
  - print(b)
  - → [1, 2, 3, 200]
  - b.append([‘상’, ‘명’,’대’])
  - → [1, 2, 3, 200, ‘상’, ‘명’,’대’]

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - bb = [3, 2, 1]
  - bb.sort()
  - print(bb)
  - → [1, 2, 3]
  - bbb = [1, 2, 3, ‘상’, ‘명’,’대’]
  - bbb.sort()
  - print(bbb)
  - → ?
  - bbb.reverse()
  - print(bbb)
  - → [’대’,’명’,’상’,3,2,1]

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - b = [1, 2, 3]
  - b.insert(0,4)
  - print(b)
  - → [4, 1, 2, 3]
  - b.remove(1)
  - print(b)
  - → [4, 2, 3]

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - b = [1, 2, 3]
  - print(b.pop())
  - → 3
  - print(b)
  - → [1, 2]
  - bb = [3, 2, 1]
  - print(bb.pop(1))
  - → [3, 1]

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - b = [1, 2, 3]
  - b.append(1)
  - print(b)
  - → [1, 2, 3, 1]
  - b.count(1)
  - → 2

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - b = [1, 2, 3]
  - b.extend([4,5])
  - print(b)
  - → [1, 2, 3, 4, 5]

리스트

- 리스트(배열): 자료구조 형태 중 하나로 순서가 있는 수정 가능한 데이터의 집합
  - b = [1, 2, 3]
  - b.extend([4,5])
  - print(b)
  - → [1, 2, 3, 4, 5]

프로그래밍 설계

프로그래밍 명세

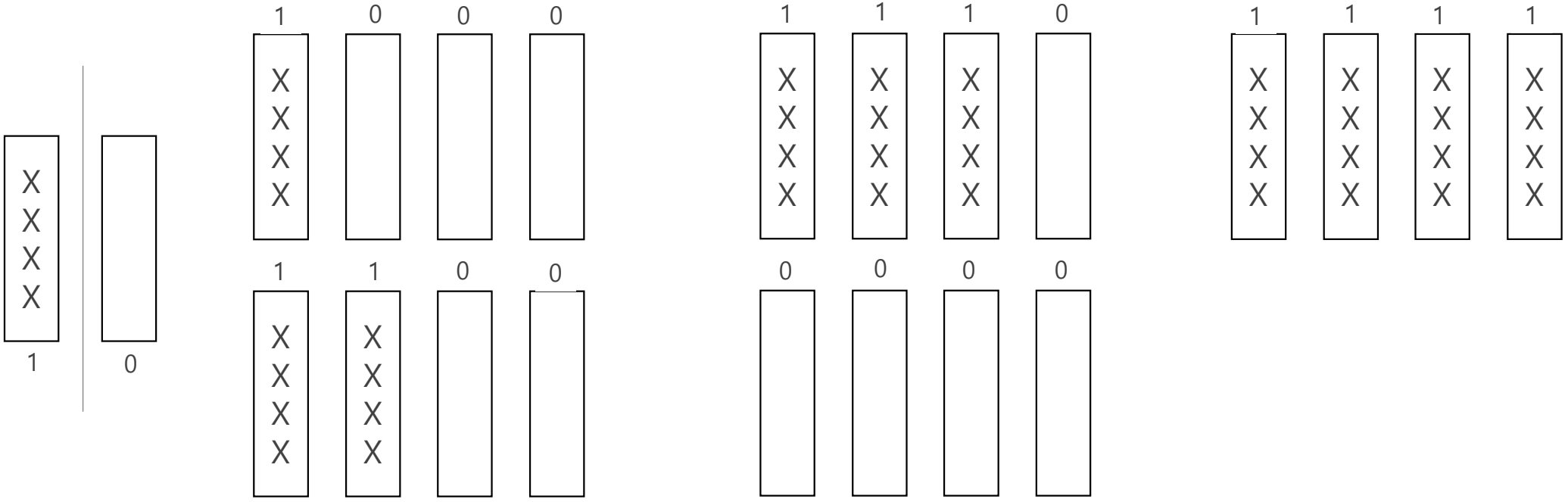
- 보물찾기 프로그램
- 다음 데이터 중 “보물”의 위치를 찾아보세요
- treaser_box = [1,2, ”보”, “물”, “보 물”, “1”,”2”,”물보”, “보물 “, “ 보물”, [‘보’, ‘물’], “보물”]
- 다음 데이터를 정렬해보고 다시 “보물 " 의 위치를 찾아보세요

입력(Input)	처리(Processing)	출력(Output)
▪ 데이터: ▪ 변수(이름, 형태): ▪ 자료형: ▪ 연산자: ▪ 명령문: ▪ 함수:	▪ 데이터: ▪ 변수(이름, 형태): ▪ 자료형: ▪ 연산자: ▪ 명령문: ▪ 함수: ▪ 이벤트:	▪ 데이터: ▪ 변수(이름, 형태): ▪ 자료형: ▪ 연산자: ▪ 명령문: ▪ 함수: ▪ 이벤트:

프로그래밍 설계

프로그래밍 명세 - 윷놀이

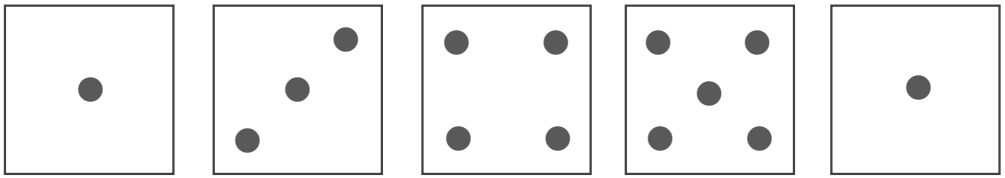
- 1) 각각 4개의 윷을 한번에 던지면 무작위로 '도 개 걸 윷 모' 가 나온다
- 2) 사용자가 종료하기 전까지 연속해서 윷을 던진다
- 3) 종료하면 사용자가 던진 횟수, 나온 윷의 횟수를 출력한다

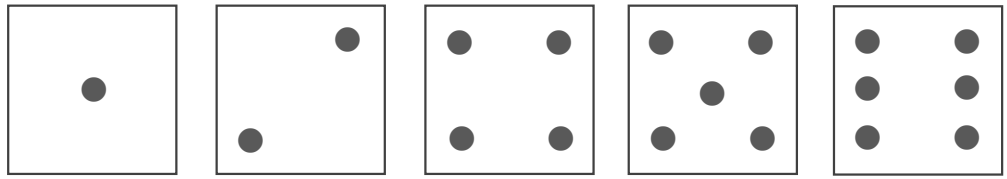


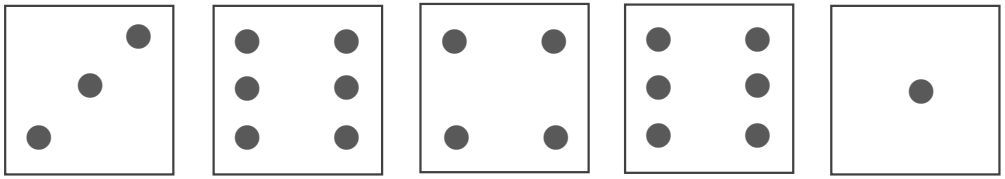
프로그래밍 설계

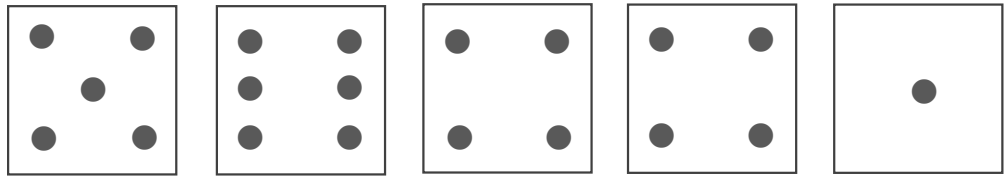
프로그래밍 명세 - 주사위합 맞추기

- 1) 사용자가 무작위로 주사위를 5회 던진다.
- 2) 컴퓨터도 무작위로 주사위를 5회 던진다.
- 3) 두 플레이어 중 주사위의 합이 20이 되면 (20을 맞추면) 종료되고 결과(승패)를 출력한다





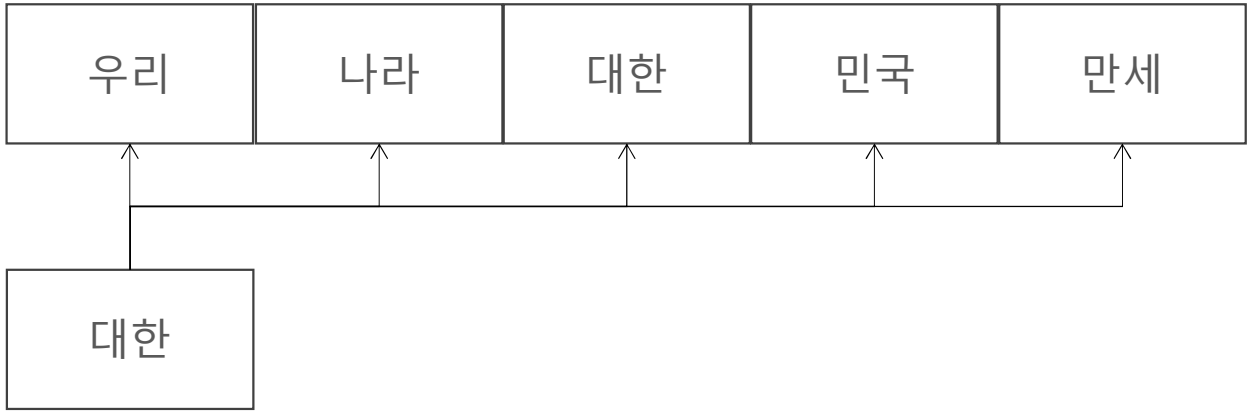




프로그래밍 설계

프로그래밍 명세 - 친구찾기

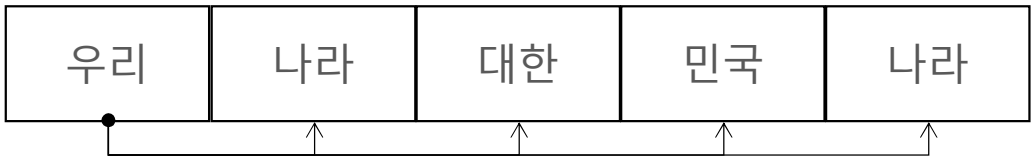
- 1) 사용자로부터 먼저 5명의 친구 이름 입력 받는다
- 2) 사용자에게 그 중 찾고 싶은 친구 이름을 입력 받는다
- 3) 찾고 싶은 친구이름이 등록되어 있으면 찾은 이름을 출력하고 없으면 “못찾았다” 문자열 출력하고 종료한다

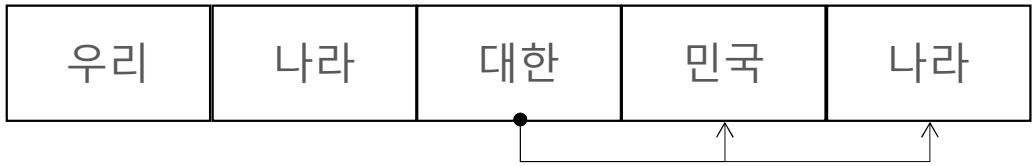


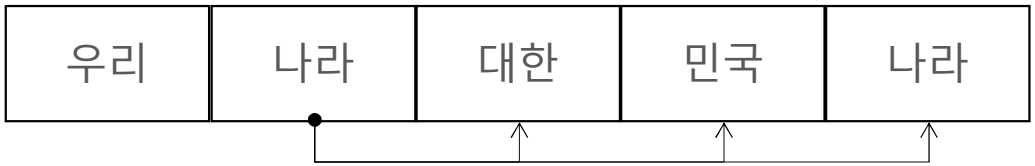
프로그래밍 설계

프로그래밍 명세 - 같은 이름 친구찾기

- 1) 사용자로부터 먼저 5명의 친구 이름 입력 받는다(같은 이름 추가)
- 2) 등록된 친구 중 같은 이름이 있으면 같은 이름을 출력하고 없으면 “같은 이름 없다” 문자열 출력하고 종료한다







프로그래밍 설계

프로그래밍 명세 - 메신저 친구 프로필

- 1) 친구 리스트 출력
- 2) 친구 찾기
- 3) 친구 추가
- 4) 친구 삭제
- 5) 이름변경
- 6) 종료

맺음말