

파이썬

상명대학교 융합공과대학 지능·데이터융합학부

휴먼지능정보공학전공

상명대학교 바이오헬스혁신공유대학

dkim@smu.ac.kr

강의개요

- 강의소개 및 프로그래밍 개념
 - 프로그래밍과 컴퓨팅사고력 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, 딕셔너리
 - 코딩과 기초실습
- 실습예제
 - 코딩과 기초실습
- 파일읽고 쓰기
 - 코딩과 기초실습
- 객체지향 프로그래밍
 - 코딩과 기초실습
- 기말고사

프로그래밍 기본요소

- 자료형: 데이터(자료)를 표현하는 방법(형태가 있음)
 - 정수형, 실수형, 문자, 문자열(str)형, 리스트(list)형, 불(bool)형, 튜플(tuple), 집합(set)형, 사전(dict)형 등
- a=10
 - a(변수, 정수형) = (대입연산자) 10(데이터, 자료, 상수)

변수

- 변수: 데이터(자료)를 저장하는 공간 (이름, 형태가 있음)
 - 정수형, 실수형, 문자, 문자열(str)형, 리스트(list)형, 불(bool)형, 튜플(tuple), 집합(set)형, 사전(dict)형 등
- 데이터(자료): 프로그램을 운용할 수 있는 형태로 기호화 숫자화 한 자료(데이터)
 - 숫자, 문자
- 상수: 항상 같은 값을 가지는 수나 문자 데이터(자료) 자체
 - 숫자, 문자

자료형

- 자료형: 데이터(자료)를 표현하는 방법(형태가 있음)
 - 정수형, 실수형, 복소수형
 - 문자, 문자열(str)형
 - 불(bool)형
 - 리스트(list)형
 - 튜플(tuple)형
 - 집합(set)형
 - 사전(dict)형

자료형

- 자료형: 데이터(자료)를 표현하는 방법(형태가 있음)
 - 정수형, 실수형, 복소수형
 - 정수형: 가장 간단한 수치형 (소수점이 없음)
 - $a=20$
 - $b=20$
 - $c = a + b$
 - `print(c)` → 40

자료형

- 자료형: 데이터(자료)를 표현하는 방법(형태가 있음)
 - 정수형, 실수형, 복소수형
 - 실수형: 소수점이 있는 수치형
 - `a=20.34`
 - `b=20.54`
 - `c=2044.22e-2`
 - `d=a+b`
 - `e=b+c`
 - `print(c) → 20.4422`
 - `print(d) → 40.879999999999995`
 - `print(e) → 40.9822`
 - `print('% .4f' % d) → 40.8800` ## `print("%출력원하는 자리수f"%출력값)`
 - `print(round(d)) → 41`
 - `print(round(d, 1)) → 40.9`

자료형

- 자료형: 데이터(자료)를 표현하는 방법(형태가 있음)
 - 문자, 문자열(str)형 한 개의 문자, 여러 개의 문자열을 표현하고 처리
 - a = '상' → 문자
 - b = "상명" → 문자열
 - c = 상명대학교 → 오류
 - d = "우리는 '상명대학교' 학생입니다"
 - e = 100 + "100" → 오류
 - f = str(100) + "100" → 100100

자료형

- 자료형: 데이터(자료)를 표현하는 방법(형태가 있음)
 - 문자, 문자열(str)형: 한 개의 문자, 여러 개의 문자열을 표현하고 처리
 - `g = input("이름을 입력하세요")` → 상명대
 - `h = print("안녕하세요"+g+"님 반갑습니다")` → 안녕하세요 상명대님 반갑습니다.

자료형

- 자료형: 데이터(자료)를 표현하는 방법(형태가 있음)
 - 불(bool)형: 참과 거짓을 표현하고 처리
 - $a = \text{True} \rightarrow ##1$
 - $b = \text{False} \rightarrow ##0$
 - $c = a + b \rightarrow 1$
 - $d = a - b \rightarrow 1$
 - $e = a * b \rightarrow 1$
 - $f = a / b \rightarrow ##\text{오류}$ (나누기 연산)
 - $a = \text{bool}(1) \rightarrow \text{True}$
 - $b = \text{bool}(0) \rightarrow \text{False}$

자료형

- 자료형: 데이터(자료)를 표현하는 방법(형태가 있음)
 - 리스트(list)형: 순서가 있는 수정가능한 데이터의 집합
 - `a = [1,3,5]`
 - `b = [2,4,6]`
 - `c = a + b`
 - `print(c) → [1,3,5,2,4,6]`
 - `a = ['상', '명']`
 - `b = ['대', '학', '교']`
 - `c = a + b`
 - `print(c) → ['상', '명', '대', '학', '교']`
 - `a = ["상명"]`
 - `b = ["대학"]`
 - `c = ["교"]`
 - `print(a+b+c) → ["상명", "대학", "교"]`

자료형

- 자료형: 데이터(자료)를 표현하는 방법(형태가 있음)
 - 튜플(tuple)형: 순서가 있는 수정 불가능한 데이터의 집합
 - 데이터 접근, 삭제 가능, 추가 불가능
 - ()괄호로 작성되어지며, 내부 원소는 ,로 구분
 - 튜플 이름 = (요소1, 요소2, 요소3, ...)
 - 튜플 더하기 및 (정수)곱하기(반복) 연산 가능
 - `a = (1,2,3)`
 - `b = ('가','나','다')`
 - `c = a + b`
 - `print(c)` → `(1,2,3, '가', '나', '다')`

자료형

- 자료형: 데이터(자료)를 표현하는 방법(형태가 있음)
 - 집합(set)형: 순서가 없고 중복을 허용하지 않는 데이터의 집합
 - `a = set([1,2,3,4,5])`
 - `b = set([1,3,5,6])`
 - `c = a.intersection(b)`
 - `print(c) → {1,3,5}`
 - `d = a.union(b)`
 - `print(d) → {1, 2, 3, 4, 5, 6}`

자료형

- 자료형: 데이터(자료)를 표현하는 방법(형태가 있음)
 - 사전(dict)형: 쌍이 있는 수정 가능한 데이터의 집합
 - 접근, 추가, 삭제 가능
 - 중괄호({ })로 묶여 있으며 키와 값의 쌍으로 이루어지고 내부 원소는, 로 구분
 - 딕셔너리 이름 = {키1:값1, 키2:값2, 키3:값3, ...}
 - a = {1:"가", 2:"나", 3:"다", 4:"라", 5:"마"}
 - print(a) → {1: '가', 2: '나', 3: '다', 4: '라', 5: '마'}

프로그래밍 기본요소

- 연산자: 프로그램의 산술식이나 연산식을 표현하고 처리하기 위해 제공되는 기호
 - 대입연산, 산술연산, 증감연산, 관계연산, 논리연산

연산자

- 연산자: 프로그램의 산술식이나 연산식을 표현하고 처리하기 위해 제공되는 기호
 - 대입연산
 - 산술연산
 - 증감연산
 - 관계연산
 - 논리연산

연산자

- 연산자: 프로그램의 산술식이나 연산식을 표현하고 처리하기 위해 제공되는 기호
 - 대입연산 (=)
 - a = 10
 - b = “상명”
 - d = [1,2,3]
 - e = print(a)

연산자

- 연산자: 프로그램의 산술식이나 연산식을 표현하고 처리하기 위해 제공되는 기호
 - 산술연산 (+, -, *, /)
 - $a = 1 + 1 \rightarrow 2$
 - $b = a + 1 \rightarrow 3$
 - $a = 1 - 1 \rightarrow 0$
 - $b = a - 1 \rightarrow -1$
 - $a = 1 * 1 \rightarrow 1$
 - $b = a * 1 \rightarrow 1$
 - $a = 1 / 1 \rightarrow 1$
 - $b = a / 1 \rightarrow 1$

연산자

- 연산자: 프로그램의 산술식이나 연산식을 표현하고 처리하기 위해 제공되는 기호
 - 증감연산 ($+=$, $-=$)
 - $a = 1$
 - $a += 1$
 - $\text{print}(a) \rightarrow 2$
 - $a -= 1$
 - $\text{print}(a) \rightarrow 1$

연산자

- 연산자: 프로그램의 산술식이나 연산식을 표현하고 처리하기 위해 제공되는 기호
 - 관계연산 (> : 크다, >= : 크거나 같다, < : 작다, <= : 작거나 같다, == : 같다, != : 같지 않다)
 - a = 1
 - b = 1
 - print(a > b) → False
 - print(a >= b) → True
 - print(a < b) → False
 - print(a <= b) → True
 - print(a==b) → True
 - print(a != b) → False

연산자

- 연산자: 프로그램의 산술식이나 연산식을 표현하고 처리하기 위해 제공되는 기호
 - 논리연산 (and, or, not)
 - a = True
 - b = False
 - print(a and b) → False
 - print(a or b) → True
 - print(a and not b) → True

프로그래밍 기본요소

- 함수: 하나의 특별한 목적의 작업을 수행하기 위해 독립적으로 설계된 코드 모음
 - 내장함수(매개변수, 반환 값), 사용자정의함수(매개변수, 반환 값), 라이브러리(매개변수, 반환 값)
 - def 함수이름(매개변수):
실행하고자 하는 함수내용
(변수, 상수, 연산자, 대입문, 선언문, 반복문, 조건문, 분기문, 함수호출 등)
 - 함수이름(매개변수)

함수

- 함수형태
 - 기본형
 - 매개변수, 반환값(변수)이 없는 형태
 - 매개변수형
 - 매개변수만 있는 형태
 - 반환형
 - 반환값(변수)만 있는 형태, return
 - 매개변수, 반환형
 - 매개변수, 반환값(변수)가 있는 형태, return

```
def 함수이름 :
```

함수내용

```
def 함수이름(매개변수,매개변수,...) :
```

함수내용

```
def 함수이름 :
```

함수내용

return 반환값(변수)

```
def 함수이름(매개변수,매개변수,...) :
```

함수내용

return 반환값(변수)

함수

- 함수종류

- 내장함수: 파이썬에서 제공하는(포함되어 있는) 함수
 - 함수호출, 매개변수, 반환값(변수) 사용
 - import 키워드 사용하지 않음
 - 문자열출력함수: `print("출력")` → 출력
 - 입력함수: `mytext = input("문자를 입력하세요")` → 안녕하세요
 - 절대값함수: `a=abs(-10)` → 10
 - 문자열실행함수: `eval("1+2")` → 3
 - 16진수함수: `a = hex(3)` → 0x3
 - 정수변환함수: `a = int(3.4)` → 3
 - 배열길이반환함수: `a = len([1,2,3])` → 3
 - 최대값반환함수 `a= max([1,2,3])` → 3
 - 최솟값반환함수 `a= min([1,2,3])` → 1

함수

- 함수종류

- 내장함수: 파이썬에서 제공하는(포함되어 있는) 함수
 - 함수호출, 매개변수, 반환값(변수) 사용
 - import 키워드 사용하지 않음
 - 파일함수: `f = open("mytest.txt", "r")` → 파일
 - 리스트함수: `list((1,2,3))` → `[1,2,3]`
 - 제곱함수: `pow(2,4)` → 16
 - 반올림함수: `round(4.6)` → 5
 - 자료형반환함수: `type("abc")` → `<class 'str'>`

함수

- 함수종류

- 사용자정의함수: 사용자가 특정 작업을 수행하기 위해 만든 함수

- 함수 이름 앞에 def 키워드를 사용
 - 함수 이름 다음 () 안에 매개변수 사용
 - () 다음에 : 붙임
 - 함수내용 작성
 - return 키워드 사용하여 반환값(변수) 지정
 - 재귀함수: 함수내용 안에 자기자신 함수(호출)를 사용한 함수

```
def MyHello(count):
```

```
    if count == 0: # 종료 조건을 만듦. count가 0이면 다시 hello 함수를 호출하지 않고 끝냄
```

```
        return
```

```
    print("Hello World", count)
```

```
    count -= 1    # count를 1 감소시킨 뒤
```

```
    MyHello(count) # 다시 Myhello에 넣음
```

```
MyHello(2) # hello 함수 호출
```

→

```
Hello World, 2
```

```
Hello World, 1
```

함수

- 함수종류

- 외장함수: 특별한 기능이 있는 모듈(함수의 집합)안에 있는 특정 함수를 사용
 - 함수를 포함시키기 위해 import 키워드를 사용
 - 시간모듈

```
import time
```

```
time.localtime() → time.struct_time(tm_year=2020, tm_mon=4, tm_mday=19, tm_hour=20,  
tm_min=20, tm_sec=40, tm_wday=6, tm_yday=110, tm_isdst=0):
```

```
# tm_wday는 요일(월요일~일요일, 0~6), tm_yday는 1월 1일부터 경과한 일수, tm_isdst는  
서머타임 여부
```

```
import time
```

```
for i in range(10):
```

```
    print(i)
```

```
    time.sleep(2) → 2초 간격으로 0부터 9까지 숫자 출력
```

프로그래밍 설계

프로그래밍 명세

- 두 수를 입력 받아 더하고, 빼고, 곱하고, 나눈 결과를 출력하는 프로그램

프로그래밍 설계

프로그래밍 명세

- 두 수를 입력 받아 더하고, 빼고, 곱하고, 나눈 결과를 출력하는 프로그램

입력(Input)

- 두 수(값)
- 입력받는다

처리(Processing)

- 두 수(값)
- 더하고, 빼고, 곱하고, 나눈다

출력(Output)

- 처리 결과(값)
- 출력한다

프로그래밍 설계

프로그래밍 명세

- 두 수를 입력 받아 더하고, 빼고, 곱하고, 나눈 결과를 출력하는 프로그램

입력(Input)

- 두 수(값)
- 입력받는다

- 데이터: 두 수
- 변수(이름, 형태): a,b(정수)
- 자료형: 정수형
- 연산자: 대입연산
- 명령문: 순서
- 함수: 입력함수(내장함수,사용자함수)
- 이벤트: ~할 때

처리(Processing)

- 두 수(값)
- 더하고, 빼고, 곱하고, 나눈다

- 데이터: 두 수, 처리 결과값
- 변수(이름, 형태): a,b,result(정수형)
- 자료형: 정수형
- 연산자: 대입, 사칙(=, -, *, /)연산
- 명령문: 순서
- 함수: 연산함수(내장함수,사용자함수)
- 이벤트: ~할 때

출력(Output)

- 처리 결과(값)
- 출력한다

- 데이터: 처리 결과값
- 변수(이름, 형태): result(정수형)
- 자료형: 정수형, 실수형
- 연산자: 대입
- 명령문: 순서
- 함수: 출력함수(내장함수,사용자함수)
- 이벤트: ~ 할 때

프로그래밍 설계

프로그래밍 명세

- 두 수를 입력 받아 더하고, 빼고, 곱하고, 나눈 결과를 출력하는 프로그램



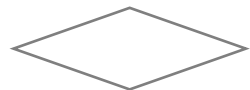
시작, 끝



초기값 선언(설정)



데이터 입,출력



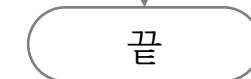
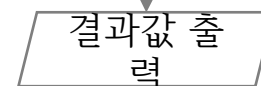
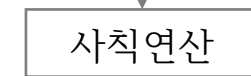
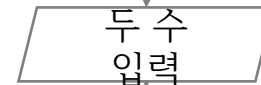
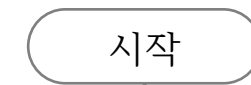
조건에 따른 판단(분기)



연산 및 처리



데이터 제어(흐름)



프로그래밍 설계 – 실습1

프로그래밍 명세

- 이름과 출생연도를 입력 받아 이름과 현재 나이를 출력하는 프로그램

입력(Input)

■

- 데이터:
- 변수(이름, 형태):
- 자료형:
- 연산자:
- 명령문:
- 함수:
- 이벤트:

처리(Processing)

■

- 데이터:
- 변수(이름, 형태):
- 자료형:
- 연산자:
- 명령문:
- 함수:
- 이벤트:

출력(Output)

■

- 데이터:
- 변수(이름, 형태):
- 자료형:
- 연산자:
- 명령문:
- 함수:
- 이벤트:

프로그래밍 설계 - 실습1

프로그래밍 명세

- 이름과 출생연도를 입력 받아 이름과 현재 나이를 출력하는 프로그램



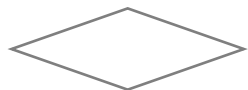
시작, 끝



초기값 선언(설정)



데이터 입,출력



조건에 따른 판단(분기)



연산 및 처리

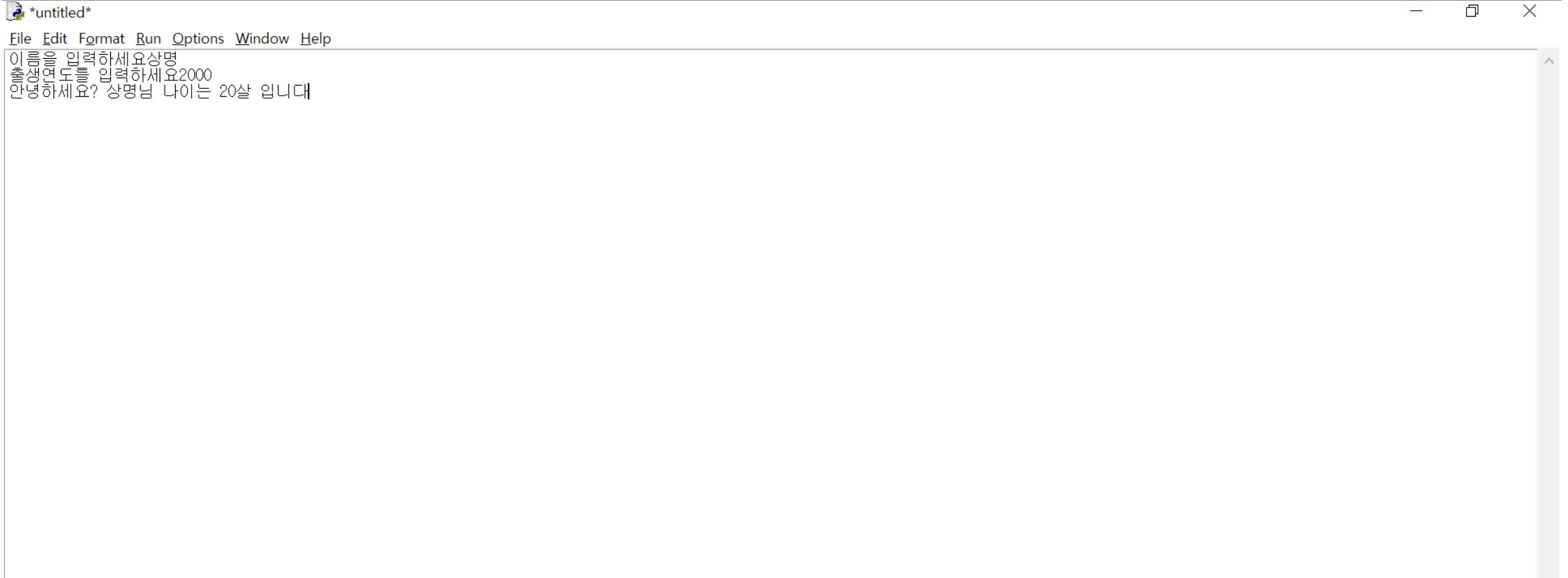


데이터 제어(흐름)

프로그래밍 맛보기

프로그래밍 명세

- 이름과 출생연도를 입력 받아 이름과 현재 나이를 출력하는 프로그램



예제맛보기

프로그래밍 명세

- 1)사용자에게 ‘묵’, ‘찌’, ‘빠’ 문자열을 입력받고 입력된 문자열이 각 ‘묵’, ‘찌’, ‘빠’ 인 경우에만 각 해당 문자열을 출력한다
- 2)다른 문자열을 입력된 경우 “재입력” 메시지를 출력해준다
- 3)사용자가 ‘묵’, ‘찌’, ‘빠’ 각 문자열을 입력 하지 않고 엔터를 누르면 프로그램은 종료된다

untitled

File Edit Format Run Options Window Help

```
묵 찌 빠 중 하나를 내주세요,종료하시려면 엔터를 눌러주세요묵
묵을 내셨습니다
묵 찌 빠 중 하나를 내주세요,종료하시려면 엔터를 눌러주세요찌
찌를 내셨습니다
묵 찌 빠 중 하나를 내주세요,종료하시려면 엔터를 눌러주세요빠
빠를 내셨습니다
묵 찌 빠 중 하나를 내주세요,종료하시려면 엔터를 눌러주세요가위
묵 찌 빠로 입력하세요
묵 찌 빠 중 하나를 내주세요,종료하시려면 엔터를 눌러주세요
종료합니다
>>> `
```

맺음말