

파이썬프로그래밍기초

1강. 컴퓨터의 이해

주요용어

- * 데이터: 어떤 현상이나 사실에 대한 설명 또는 설명의 집합
- * 정보: 문제 또는 질문을 해결하기 위해 사용할 수 있는 데이터와 데이터의 집합
- * 프로그램: 컴퓨터가 어떠한 작업을 자동으로 처리할 수 있도록 처리 방법 및 순서를 컴퓨터가 이해할 수 있는 언어 형태로 기술한 것
- * 알고리즘: 문제를 풀기 위한 단계별 절차를 체계적 명령의 형태로 기술한 것

정리하기

- * 데이터란 관찰이나 측정을 통해 얻은 수치, 문자와 같은 변할 수 있는 질적 또는 양적 값을 뜻한다.
- * 정보란 데이터를 처리하여 실제 문제해결에 도움이 될 수 데이터를 정리한 결과물을 뜻한다.
데이터 처리에는 정렬, 합산, 군집화 등의 작업이 동반된다.
- * 데이터는 현상으로부터 관찰 및 측정을 통해 얻어지며, 정보는 데이터를 처리 및 가공하여 추출된다.
- * 컴퓨터란 입력된 데이터를 자동으로 처리하는 기계로, 데이터 처리 과정을 순서화하여 전체 과정을 자동화할 수 있는 장치를 말한다.
- * 컴퓨터는 데이터를 외부로부터 받아들이는 입력기능, 내부 장치를 조정하는 제어기능, 입력된 데이터를 저장하는 저장·기억기능, 계산을 수행하는 연산기능, 내부의 데이터를 사용자에게 전달하는 출력기능을 갖추고 있다.
- * 디지털 컴퓨터는 0과 1이라는 신호만으로 동작하기 때문에, 데이터 및 명령문 또한 0과 1로 표현되어야 한다. 즉, 수나 문자, 명령어 모두 0과 1로 표현된다.
- * 프로그램이란 컴퓨터가 어떠한 작업을 자동으로 처리할 수 있도록, 처리 방법 및 순서를 컴퓨터가 이해할 수 있는 언어로 기술해 놓은 것을 뜻한다. 프로그램을 제작하는 사람을 프로그래머라고 하며, 이 제작 과정은 프로그래밍이라 부른다.
- * 알고리즘은 문제를 풀기 위한 단계별 절차를 수학적으로 기술한 것을 뜻한다. 알고리즘에는 입력과 출력이 있어야 하며, 명확성과 유한성, 효율성이 보장되어야 한다.
- * 프로그램은 프로그래밍 언어를 이용하여 알고리즘을 구현한 결과이며, 프로그래밍 언어로 작성된 소스코드를 컴파일러나 인터프리터를 통해 컴퓨터에서 실행될 수 있는 프로그램으로 생성된다.

2강. 컴퓨터의 구성

주요용어

- * 폰노이만 구조: CPU, 메모리, 프로그램 구조를 갖는 범용 컴퓨터 구조
- * 하드웨어: 컴퓨터를 구성하고 있는 물리적으로 존재하는 부품
- * 소프트웨어: 다양한 장치들을 동작시켜 특정 작업을 해결하는 프로그램

정리하기

- * 컴퓨터는 물리적 장치인 하드웨어와 논리적 장치인 소프트웨어로 구성된다.
- * 일반적으로 하드웨어는 사람이 정보처리를 하는 역할과 유사하게 입력장치, 연산장치, 기억장치, 출력장치, 제어장치로 구분된다.
- * 현대의 컴퓨터는 정보와 정보처리를 위한 프로그램을 내부의 메모리에 저장하고 이를 처리하는 폰 노이만 구조에 기반한다.
- * 입력장치는 외부 데이터를 컴퓨터로 전달하는 장치이며 출력 장치는 입력장치를 통해 중앙처리장치로 들어온 데이터를 처리하고, 그 결과를 표시해 주는 장치이다.
- * 중앙처리장치는 연산, 비교, 판단을 수행하는 연산장치와 명령어를 해석하고 실행하는 제어장치로 구성된다.
- * 주기억장치는 컴퓨터에서 처리할 데이터와 프로그램을 기억하고, 처리결과를 기억하는 장치이며 보조기

억장치는 주기억장치를 보완하여 컴퓨터 전원이 차단되었다가 다시 컴퓨터를 켜올 때, 주기억장치에 데이터를 불러 올 수 있도록 영구적으로 저장하는 역할을 한다.

- * 시스템 소프트웨어는 응용 소프트웨어를 실행하기 위한 환경을 제공하고 하드웨어를 제어·관리할 수 있도록 설계된 컴퓨터 소프트웨어이다.
- * 시스템 소프트웨어 종류에는 운영체제는 사용자가 컴퓨터를 효율적으로 운영·관리·사용할 수 있도록 하드웨어를 제어하는 운영체제, 고급언어로 작성된 프로그램을 컴퓨터가 이해할 수 있는 기계어로 번역해주는 컴파일러, 그리고 사용자가 컴퓨터를 효율적이고 편리하게 사용할 수 있도록 지원해 유틸리티 등이 있다.
- * 운영체제는 중앙처리장치, 주기억장치, 보조기억장치와 같은 장치로 구성되는 컴퓨터의 하드웨어를 작동시켜 사용자가 원하는 작업을 처리할 수 있는 환경을 제공하고 컴퓨터와 사용자 사이에 중계역할을 한다.

3강. 파이썬의 이해

주요용어

- * 프로그래밍 패러다임: 프로그래밍을 생성하는 접근 방식
- * 라이브러리: 자주 사용되는 프로그래밍 구성 요소를 미리 작성하여 패키징한 프로그램 모음
- * 주피터 노트북: 오픈소스 웹 기반 대화형 개발 및 실행 환경

정리하기

- * 파이썬은 히도 판로섬(Guido van Rossum)에 의해 1991년 분산 운영 체제(아메바)의 시스템 관리를 위한 셸 스크립팅 언어로 개발되었다.
- * 파이썬은 명령형 프로그래밍, 절차적 프로그래밍, 객체지향 프로그래밍, 함수형 프로그래밍 패러다임을 지원한다.
- * 파이썬은 교육적인 목적뿐만 아니라 응용 프로그램과 웹, 백 엔드 개발, 사물 인터넷 등 다양한 분야에 활용된다.
- * 파이썬은 파이썬 확장제안(PEP)를 통해 새로운 파이썬의 기능, 파이썬 프로세스, 환경에 대해 커뮤니티에 설계 문서나 정보를 제공하는 등 많은 개발자의 의견을 수용하고 토론하며 발전한 언어이다.
- * 파이썬은 플랫폼에 독립적인 인터프리터식 객체지향, 동적 타이핑(dynamically typed) 대화형 언어이다.
- * 파이썬 소스 코드는 파이썬 인터프리팅 과정에서 CPython에 의해 바이트 코드로 변환되고, 파이썬 가상머신(PVM)이 한 라인씩 기계어 코드로 변환하여 실행된다.
- * 파이썬 개발 환경에는 구문 강조, 자동 완성, 스마트 들여쓰기 등의 기능을 지원하는 IDLE, 주피터 노트북, 구글 코랩 등이 있다.
- * 구글 코랩(Colab)은 주피터 노트북과 구글 드라이브를 결합한 서비스로 데이터 분석 및 딥러닝 연산 등 고성능 컴퓨팅 리소스 활용 가능하다.

4강. 파이썬 시작하기

주요용어

- * 함수: 특정 작업을 수행하는 코드의 집합으로 함수의 이름만으로 실행할 수 있는 단위
- * 변수: 처리할 데이터와 처리된 결과를 임시적 저장하는 공간
- * 산술연산자: 피연산자(operand)에 대해 지정된 산술 연산을 지시하는 기호
- * 내장함수: 별도의 라이브러리 없이 파이썬 인터프리터에서 기본적으로 지원하는 함수

정리하기

- * 파이썬은 히도 판로섬(Guido van Rossum)에 의해 1991년 분산 운영 체제(아메바)의 시스템 관리를 위한 셸 스크립팅 언어로 개발되었다.
- * 피연산자와 기본연산자(+, -, *, /, **)를 이용한 표현식은 파이썬 인터프리터에 의해 자동 계산된다.
- * 특정 작업을 수행하는 코드의 집합으로 함수의 이름만으로 실행할 수 있는 단위를 함수라고 한다.

- * 파이썬은 들여쓰기에 의존적 언어이며 코드의 논리적 집합인 블록을 표현하는데 사용된다.
- * 주석은 개발 속도 향상시키고 유지보수를 용이하게 하는 중요한 설명이다. 주석은 #과 ""를 사용한다.
- * 처리할 데이터와 처리된 결과의 임시적 저장에 변수가 사용되며 할당연산자의 왼쪽에 정의한다.
- * 식별자는 프로그램 내부에 정의된 객체(변수, 함수 등)의 이름이다.
- * 프로그래밍 언어에서만 사용되는 산술연산자로 정수 나눗셈 연산자(//)와 모듈로 연산자(%)가 있다.
- * 연산자 우선순위는 표현식에 사용된 여러 연산자의 연산 순서를 결정한다.
- * 파이썬 내장함수는 별도의 모듈이나 패키지없이 파이썬 인터프리터에서 기본적으로 지원하는 함수이다.

5강. 순차 구조

주요용어

- * 제어구조: 프로그램의 실행 흐름을 결정하는 구조
- * 순차구조: 실행의 흐름을 주어진 명령의 위치적 흐름에 따라 수행하는 구조
- * 프로그래밍 에러: 설계 미숙, 결함 또는 문법 오류로 프로그램이 의도한 대로 결과를 생성하지 못하는 문제 상황
- * 데이터 타입 변환: 한 데이터의 타입을 다른 데이터 타입으로 전환하는 과정

정리하기

- * 파이썬은 구조적 프로그래밍 패러다임을 지원하며 goto 문을 사용하지 않고 프로그램을 구성할 수 있는 순차, 선택, 반복 구조를 제공한다.
- * 순차 구조는 실행의 흐름을 주어진 명령의 위치적 흐름에 따라 수행하는 구조이다.
- * 선택 구조는 특정 영역 내의 명령문에 대한 실행 여부를 프로그램 실행 과정 중 조건에 따라 결정하는 구조이다.
- * 반복 구조는 특정 영역의 명령문을 조건에 따라 여러 번 재실행하는 구조이다.
- * input 함수는 사용자로부터 데이터를 입력 받아, 문자 데이터 타입으로 반환한다.
- * 프로그래밍 에러는 설계 미숙, 결함 또는 문법 오류로 프로그램이 의도한 대로 결과를 생성하지 못하는 현상이며 구문 오류, 실행 오류, 의미 오류로 구분된다.
- * 데이터 타입을 다른 데이터 타입으로 전환하기 위해 str, int, float 함수를 사용한다.
- * print 함수는 여러 개의 데이터를 단일 print 함수로 출력 가능하며, 콤마(,)로 파라미터를 구분하여 입력한다.

6강. 선택 구조

주요용어

- * 선택 구조: 특정 영역 내의 명령문에 대한 실행 여부를 판단에 따라 결정하는 구조
- * 불리언 타입: 논리값인 참(True)과 거짓(False)의 값만 표현할 수 있는 데이터 타입
- * 불리언식 : 비교연산자를 사용하여 결과가 불리언 타입으로 생성되는 표현식
- * 논리연산자: 두 개의 논리값(불리언식)을 연산하여 참 또는 거짓을 결과로 얻는 연산자

정리하기

- * 선택 구조란 특정 영역 내의 명령문에 대한 실행 여부를 조건에 따라 판단하여 결정하는 구조이다.
- * 선택구조의 실행 여부를 표현하는 조건은 참(True)과 거짓(False)으로 결정되는 불리언식을 사용하며, 불리언식은 비교연산자를 사용하여 결과가 불리언 타입으로 생성되는 표현식이다.
- * 불리언 타입은 논리값인 참(True)과 거짓(False)의 값만 표현할 수 있는 데이터 타입이다.
- * 논리연산자는 두 개의 논리값(불리언식)을 연산하여 참 또는 거짓을 결과로 얻는 연산자이다.
- * 단락평가란 첫 번째 논리값 만으로 전체 연산 결과가 판별 가능할 때 두 번째 논리값은 확인(평가)

하지 않는 기법이다.

- * if, elif, else 명령어를 사용하여 이분 또는 다분 선택구조로 확장할 수 있다.
- * 중첩 선택구조는 한 선택구조 내부에 또 다른 선택구조를 내포하는 구조이다.

7강. 반복 구조

주요용어

- * 조건 제어 반복 : 참과 거짓 조건에 의해 반복 여부가 결정되는 반복구조
- * 계수 제어 반복 : 특정 계수만큼 반복 횟수가 정해진 반복 구조
- * 인덱스 연산자 : 시퀀스 타입의 원소에 접근하는 연산자
- * 중첩 반복 구조 : 반복구조 내에 다른 반복 구조를 내포한 형식

정리하기

- * 반복 구조란 동일한 명령문에 대해서 사용자가 지정한 횟수만큼 또는 조건을 만족하는 동안 동일한 코드를 반복하여 수행할 때 사용하는 구조이다.
- * while 문은 조건이 참(True)인 동안 while문 블록 내부의 명령문을 수행하며 조건이 거짓(False)이 될 때 while문을 빠져나오는 조건 제어 반복 구조이다.
- * 리스트는 순서화된 값의 집합체를 저장할 수 있는 시퀀스 데이터 타입의 일종이다.
- * for 문은 리스트 등과 같은 시퀀스의 첫 번째 요소부터 마지막 요소까지 하나씩 변수에 대입하여 블록의 명령문을 수행하며 마지막 요소까지 수행한 후 종료하는 계수 제어 반복이다.
- * 중첩 반복 구조는 하나의 외부 반복 구조에 여러 개의 내부 반복 구조가 중첩되어 사용되는 구조를 말하며, 외부 반복 구조가 매 반복될 때마다 내부 반복 구조에 재진입하여 새롭게 실행된다.

8강. 체크포인트 - 강의 참조 바랍니다.

9강. 함수

주요용어

- * 동시 할당 : 복수의 변수에 값을 동시에 할당하는 명령
- * 스코프: 프로그램에서 변수가 참조될 수 있는 영역
- * 기본 매개변수: 함수 호출 시 매개변수가 전달되지 않을 경우 기본값이 전달되는 매개변수
- * 가변 매개변수: 함수 호출 시 매개변수를 사용자가 원하는 개수 만큼 지정할 수 있는 매개변수

정리하기

- * 함수란 특정 작업을 함수의 이름 만으로 수행하는 명령문의 집합을 말한다.
- * 반환값의 유무에 따라 반환값이 없는 함수와 반환값이 있는 함수로 구분된다.
- * 반환값이 없는 함수는 함수 실행 후 어떤 값도 남기지 않는 함수이다.
반면 반환값이 있는 함수는 함수 내부에서 return 명령어를 통해 값을 남긴다.
- * 동시 할당이란 복수 개의 변수에 값을 동시에 할당하는 연산이다.
이를 통해 단일 명령문으로 변수의 값을 맞바꿈이 가능하다.
- * 값의 전달에 의한 호출이란 함수 호출 시 함수 내부에는 매개변수의 값이 전달된다.
- * 변수의 스코프는 프로그램에서 변수가 참조될 수 있는 영역이다.
변수의 스코프에 따라 변수는 전역변수와 지역변수로 구분된다.
- * 기본 매개변수란 함수 호출 시 매개변수가 전달되지 않을 경우 기본값이 전달되는 매개변수를 말한다.
- * 함수 호출 시 매개변수를 사용자가 원하는 개수 만큼 지정할 수 있는 매개변수를 가변 매개변수라고 하며, 함수 정의 시 매개변수 이름 앞에 *를 사용한다.

10강. 객체지향

주요용어

- * 객체지향: 객체와 객체 사이의 상호작용으로 프로그램을 구성하는 프로그래밍 패러다임
- * 클래스: 실세계의 객체에 대한 데이터와 연산을 표현한 단위
- * 초기자: 객체의 상태를 초기화하는 특수 메소드
- * 객체 멤버 접근 연산자: 객체의 데이터 필드 접근 및 메소드 호출에 사용되는 연산자

정리하기

- * 객체와 객체 사이의 상호작용으로 프로그램을 구성하는 프로그래밍 패러다임을 객체지향이라고 한다.
- * 객체지향 패러다임의 특징에는 추상화, 캡슐화, 상속, 다형성이 있다.
- * 클래스는 실세계의 사물에 대한 상태(데이터)와 연산(메소드)을 표현한 단위이다.
- * 클래스는 객체의 행동을 정의하는 메소드와 객체의 상태를 초기화하는 특수 메소드로 구성된다.
- * 모든 메소드의 첫 번째 매개변수는 self이다.
- * 클래스의 생성자(constructor)호출을 통해 클래스의 인스턴스인 객체가 생성된다. 클래스의 생성자는 클래스의 이름과 동일하며 초기자의 매개변수를 사용한다.
- * 객체의 데이터 필드 접근 및 메소드 호출에 객체 멤버 접근 연산자(.)를 사용한다.
- * 객체의 데이터 필드의 직접 변경을 방지하기 위해 사용자의 직접적 접근을 차단하기 위해 데이터 은닉을 사용한다.
- * private 데이터 필드는 클래스 내부에서만 접근 가능하며 앞 두 밑줄(__)로 정의한다.
- * private 데이터 필드에 대해 접근하기 위한 접근자와 변경자 메소드가 정의되어야 한다.

11강. 모듈

주요용어

- * 모듈: 함수, 상수 또는 클래스를 모아 놓은 집합체
- * 네임스페이스: 특정 객체를 이름에 따라 구분할 수 있는 범위
- * 난수: 특정한 배열 순서나 규칙적인 의미가 없는, 임의의 수를 의미

정리하기

- * 모듈은 함수, 상수 또는 클래스를 모아 놓은 집합체이다.
- * 모듈, 패키지, 라이브러리는 구분되는 개념이며 패키지는 하위 패키지 및 모듈의 집합, 라이브러리는 패키지 및 모듈의 집합으로 구성되는 상위 개념이다.
- * 모듈 등록 시 import 구문과 from import 구문을 사용한다. import 구문을 통해 등록된 모듈은 사용시 모듈이름 또는 별칭을 변수/함수/클래스 앞에 붙여야 한다.
- * import는 파이썬 모듈을 프로그램 내부에서 사용할 수 있게 네임스페이스에 추가하는 기능을 수행한다.
- * dir 함수는 네임스페이스에 등록되어 있는 모든 이름들을 리스트로 반환한다.
- * help 함수는 대화형 도움말 시스템 호출 또는 클래스나 메소드의 사용방법을 반환한다.
- * 네임스페이스는 특정 객체를 이름에 따라 구분할 수 있는 범위를 나타낸다.
- * del 명령어는 네임스페이스 내 등록된 모듈의 멤버 식별자를 제거한다.
- * math 모듈은 수학적 계산 문제를 해결하기 위한 수학 함수 및 상수의 집합으로 구성된다.
- * random은 난수 관련된 기능을 제공하는 모듈이다.
- * time은 에포크 시간을 얻어 다양한 형식으로 표시하는 기능 제공하는 모듈이다.

12강. 파일

주요용어

- * 파일 이름: 파일 경로를 내포하는 파일의 고유 식별자
- * 파일 포인터: 파일 내부에서 작업 위치를 나타내는 포인터
- * 디렉터리: 키와 값의 쌍을 저장하는 시퀀스

정리하기

- * 컴퓨터에 의해 처리될 또는 처리된 데이터와 정보가 임시적으로 연속된 바이트의 형태로 저장된 상태이다.
- * 파일은 연속된 바이트와 파일의 시작, 파일 포인터, 파일의 끝(EoF)으로 구성된다.
- * 파일은 데이터가 저장되는 방식에 따라 텍스트 파일과 바이너리 파일로 구분된다.
- * 파이썬은 인터프리터에서 파일의 시작, 파일 포인터, 파일의 끝을 활용하여 데이터 읽기, 쓰기를 위한 함수를 제공한다.
- * 파일은 파일 객체 생성, 읽기/쓰기/추가하기 작업, 파일 객체 삭제 과정을 통해 처리된다.
- * 파일의 이름은 파일의 고유 식별자 역할 및 저장장치 내부에서 파일의 위치를 표현하는 파일경로를 내포한다.
- * 디렉터리는 키와 값의 쌍(pair)을 저장하는 시퀀스이다. 컴퓨터에 의해 처리될 또는 처리된 데이터와 정보가 임시적으로 연속된 바이트의 형태로 저장된 상태이다.
- * 파일은 연속된 바이트와 파일의 시작, 파일 포인터, 파일의 끝(EoF)으로 구성된다.
- * 파일은 데이터가 저장되는 방식에 따라 텍스트 파일과 바이너리 파일로 구분된다.
- * 파이썬은 인터프리터에서 파일의 시작, 파일 포인터, 파일의 끝을 활용하여 데이터 읽기, 쓰기를 위한 함수를 제공한다.
- * 파일은 파일 객체 생성, 읽기/쓰기/추가하기 작업, 파일 객체 삭제 과정을 통해 처리된다.
- * 파일의 이름은 파일의 고유 식별자 역할 및 저장장치 내부에서 파일의 위치를 표현하는 파일경로를 내포한다.
- * 파일 디렉터리는 키와 값의 쌍(pair)을 저장하는 시퀀스이다.

13강. 실전 프로그래밍 1

주요용어

- * 소프트웨어 개발 라이프사이클 : 소프트웨어 개발 계획 수립부터 폐기까지 신뢰도 높은 소프트웨어 결과물 구현을 위한 체계화된 소프트웨어 개발 체계
- * 배치 프로세스 : 일괄적으로 모아놓은 대량의 데이터를 지정한 처리방법에 따라 처리하는 과정, 프로그램
- * 이벤트 : 프로그램에 의해 감지되고 처리될수 있는 동작이나 사건

정리하기

- * 소프트웨어 개발 라이프 사이클이란 개발 계획 수립부터 폐기까지 전체적 흐름을 체계화하여 고품질의 소프트웨어 개발을 위한 체계를 말한다.
- * 소프트웨어 개발 라이프 사이클은 대규모 소프트웨어 개발시 자원을 투입하여 초기에 예상한대로의 결과물 구현을 목적으로 한다
- * 소프트웨어 개발 라이프 사이클은 계획, 분석, 설계, 구현, 테스트, 유지보수의 순으로 진행된다.
- * 배치프로세스란 일괄적으로 모인 대량의 데이터를 특정시간에 지정한 처리방법에 따라 처리하는 과정이나 프로그램을 말한다.
- * 게임은 입력장치,타이머,프로그램 등에서 발생하는 이벤트를 처리하는 이벤트 기반 프로그램이다,
- * 이벤트란 프로그램에 의해 감지되고 처리될수 있는 동작이나 사건이다.
- * 게임은 게임초기화후 ,이벤트 검사,이벤트 처리,게임 업데이트,화면생성,화면 업데이트 과정을 무한히 반복하는 게임루프로 구성된다.
- * 틱택토 2 player 3x3판 O X 가로 세로 대각선 일직선에 먼저 놓이도록 하는 게임

14강. 실전 프로그래밍 2

주요용어

- * 2차원 리스트 : 1차원 리스트의 행 인덱스와 내포된 1차원 리스트의 열 인덱스로 구성된 리스트

정리하기

- * 리스트 : 단일 식별자로 순서화된 값의 집합체를 저장할수 있는 데이터 타입
- * 2차원 리스트는 행 인덱스라는 1차원 리스트를 가리키고 있으며 행 인덱스의 각 원소가 열 인덱스라는 또 다른 1차원 리스트를 포함하는 구조이다
- * 2차원 리스트의 순회는 각 차원을 순차적 순회하기 위한 2개의 반복 구조가 중첩된 구조로 이루어진다

15강 체크포인트2 - 강의 참조 바랍니다.