

한봄고등학교 [빅데이터 정보과]

정보처리산업기사 실기 과정평가형

2022. 11. 12(토)

Prepared by DaeKyeong Kim

Ph.D.

(주)엠아이티능력개발원



1

4장 : 프로그래밍 언어 활용

2

3

4

4장: 프로그래밍 언어 활용





1. Ubuntu 설치

학습목표

- 응용소프트웨어 개발에 사용되는 프로그래밍 언어의 기본문법을 활용하여 기본 응용소프트웨어를 구현할 수 있다.

눈높이 체크

- 구조적 프로그래밍 언어 활용하기에 대해 알고 계신가요?
- 객체지향 프로그래밍 언어 활용하기에 대해 알고 계신가요?
- 스크립트 활용하기에 대해 알고 계신가요?

31. 데이터 입·출력

1. 데이터 입·출력의 개요

데이터 입·출력은 키보드로부터 데이터를 입력받아 화면이나 파일로 출력하는 것을 의미한다. 이번 섹션에서는 C와 Java의 기본적인 프로그램 구조를 이해하고 다음과 같은 내용이 포함된 프로그램 코드를 읽고 해석하는 방법을 배운다.

- 헝가리안 표기법
- 자료형
- C언어의 입·출력 함수 : scanf, printf
- Java의 입·출력 함수 : Scanner, nextInt, print, printf, println
- 서식 지정자와 제어 문자
- 연산자와 우선순위

1. 데이터 입·출력의 개요

데이터 입·출력은 키보드로부터 데이터를 입력받아 화면이나 파일로 출력하는 것을 의미한다. 이번 섹션에서는 C와 Java의 기본적인 프로그램 구조를 이해하고 다음과 같은 내용이 포함된 프로그램 코드를 읽고 해석하는 방법을 배운다.

- 헝가리안 표기법
- 자료형
- C언어의 입·출력 함수 : scanf, printf
- Java의 입·출력 함수 : Scanner, nextInt, print, printf, println
- 서식 지정자와 제어 문자
- 연산자와 우선순위

2. C 문제

6과 4를 입력받았을 때 프로그램의 실행 결과를 확인하시오.

```
#include <stdio.h>
main()
{
    int i, j, k;
    scanf("%d %d", &i, &j);
    k = i + j;
    printf("%d\n", k);
}
```

3. Java 문제

4가 입력되었을 때 프로그램의 실행 결과를 확인하시오.

```
import java.util.Scanner;

public class Test {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        int a = scan.nextInt();
        System.out.printf("a * 3 = %d\n", a * 3);
        System.out.println("a / 2 = " + (a / 2));
        System.out.print("a - 1 = " + (a - 1));
        scan.close();
    }
}
```


1. 제어문의 개요

제어문은 서술된 순서에 따라 무조건 위에서 아래로 실행되는 프로그램의 순서를 변경할 때 사용하는 명령문이다.

이번 섹션에서는 C와 Java 각각 한 문제를 이용해 제어문이 포함된 코드를 읽고 해석하는 방법을 배운다. 이번 섹션을 공부하고 나면 배열과 if, switch, for, while 등의 제어문이 포함된 C나 Java 문제의 코드를 해석하여 답을 쓸 수 있다.

2. C 문제

다음 C언어로 구현된 프로그램의 실행 결과를 확인하시오.

```
#include <stdio.h>
main() {
    int score[] = { 86, 53, 95, 76, 61 };
    char grade;
    char str[] = "Rank";
    for (int i = 0; i < 5; i++) {
        [ SWITCH문 ]
    }
}
```

3. Java 문제

다음 Java로 구현된 프로그램의 실행 결과를 확인하시오.

```
public class Test {  
    public static void main(String[] args) {  
        String str = "agile";  
        int x[] = { 1, 2, 3, 4, 5 };  
        char y[] = new char[5];  
        int i = 0;  
        while (i < str.length()) {  
            y[i] = str.charAt(i);  
            i++;  
        }  
        for (int p : x) {  
            i--;  
            System.out.print(y[i]);  
            System.out.print(p + " ");  
        }  
    }  
}
```



33. 포인터

1. 포인터의 개요

앞에서 어떤 수나 문자를 저장하기 위해 변수를 사용했다. 사실 이 변수는 기억장소의 어느 위치에 대한 이름이며 그 위치는 주소로도 표현할 수 있다. 우리는 친구 홍길동의 집에 모이기 위해 “홍길동이네 집으로 와”라고 말하기도 하지만 홍길동의 집 주소인 “서울시 마포구 서교동 00번지로 와”라고 말하기도 한다. C언어에서는 변수의 주소를 포인터라고 하고, 포인터를 저장할 수 있는 변수를 포인터 변수라고 한다. 변수의 주소인 포인터는 출력할 수도 있고 포인터가 가리키는 곳에 값을 저장하거나 읽어 오는 등 다양한 조작이 가능하다. 이런 기능 때문에 C언어는 주소를 제어할 수 있는 기능이 있다고 말한다.

이번 섹션에서는 포인터 변수를 활용하는 코드를 읽고 해석하는 방법을 배운다. 이번 섹션을 공부하고 나면, 문자열 처리, 배열의 요소 지정 등 포인터 변수를 활용하는 C 문제의 코드를 해석하여 답을 쓸 수 있다. Java는 포인터를 다룰 수 없기 때문에 이번 섹션에서는 제외했다.

33. 포인터

2. C 문제

```
#include <stdio.h>
main() {
    int a = 50;
    int* b = &a;
    *b = *b + 20;
    printf("%d, %d\n", a, *b);
    char* s;
    s = "gilbut";
    for (int i = 0; i < 6; i += 2) {
        printf("%c, ", s[i]);
        printf("%c, ", *(s + i));
        printf("%s\n", s + i);
    }
}
```



34. 구조체

1. 구조체의 개요

2. C 문제

예제 다음 C언어로 구현된 프로그램의 실행 결과를 확인하시오.

```
#include <stdio.h>
```

```
struct jsu {  
    char nae[12];  
    int os, db, hab, hhab;  
};
```

```
int main( ) {
```

```
① struct jsu st[3] = { {"데이터1", 95, 88}, {"데이터2", 84, 91}, {"데이터3", 86, 75} };  
② struct jsu* p;  
③ p = &st[0];  
④ (p + 1)->hab = (p + 1)->os + (p + 2)->db;  
⑤ (p + 1)->hhab = (p + 1)->hab + p->os + p->db;  
⑥ printf("%d", (p + 1)->hab + (p + 1)->hhab);  
}
```




35. 사용자 정의 함수

1. 사용자 정의 함수의 개요

사용자 정의 함수는 말 그대로 사용자가 직접 만들어 사용하는 함수이다.

이번 섹션에서는 C 문제를 이용해 사용자 정의 함수로 자료를 처리하는 코드를 읽고 해석하는 방법을 배운다. 이번 섹션을 공부하고 나면, 함수 포인터, 재귀 함수가 포함된 C 문제의 코드를 해석하여 답을 쓸 수 있다.

35. 사용자 정의 함수

2. C 문제

다음 C언어로 구현된 프로그램의 실행 결과를 확인하시오.

```
#include <stdio.h>
int factorial(int n);
main() {
    int (*pf)(int);
    pf = factorial;
    printf("%d", pf(3));
}
int factorial(int n) {
    if (n <= 1)
        return 1;
    else
        return n * factorial(n - 1);
}
```

1. Java 클래스의 개요

클래스는 객체 생성을 위한 필드(속성)와 메소드(함수)를 정의하는 설계도로, Java는 아무리 작은 프로그램이라도 클래스를 만들어서 사용해야 한다.

이번 섹션에서는 Java의 클래스를 정의하고 객체를 생성한 후 사용하는 기본적인 방법과 클래스 간의 상속 관계가 적용된 코드를 읽고 해석하는 방법을 배운다. 이번 섹션을 공부하고 나면 Java에서 클래스 간의 상속 관계가 적용된 코드를 해석하여 답을 쓸 수 있다.

36. Java 클래스

1. Java 클래스의 개요

클래스를 만들어 사용하는 순서는 다음과 같다.

1. 클래스 이름을 정하고 객체 생성을 위한 필드와 메소드를 정의한다.
마치 자동차를 만들기 위한 설계도와 같다. 이때 사용하는 명령이 `class`이다.
2. 객체를 생성한다. 자동차 설계도로 자동차를 만들어야 사용할 수 있듯이 클래스를 이용해 객체를 생성해야 프로그램에서 사용할 수 있다. 이때 사용하는 명령이 `new`이다.
3. 생성된 객체들을 이용하여 프로그램을 코딩하면 된다.

36. Java 클래스

2. Java 문제

다음 Java로 구현된 프로그램의 실행 결과를 확인하시오.

```
class ClassA {  
    int a = 10;  
    int funcAdd(int x, int y) {  
        return x + y + a;  
    }  
}  
  
public class Test {  
    public static void main(String[] args) {  
        int x = 3, y = 6, r;  
        ClassA cal = new ClassA();  
        r = cal.funcAdd(x, y);  
        System.out.print(r);  
    }  
}
```

2. Java 문제

다음 Java로 구현된 프로그램의 실행 결과를 확인하시오.

```
class ClassA {
    ClassA() {
        System.out.print('A');
        this.prn(); }
    void prn() { System.out.print('B'); }}
class ClassB extends ClassA {
    ClassB() {
        super();
        System.out.print('D'); }
    void prn() { System.out.print('E'); }
    void prn(int x) { System.out.print(x); }
}
public class Test {
    public static void main(String[] args) {
        int x = 7;
        ClassB cal = new ClassB();
        cal.prn(x); }}
```

1. 추상 클래스와 형 변환의 개요

추상 메소드는 자식 클래스에서 재정의해야만 사용할 수 있는 메소드를 의미하며, 이런 메소드를 하나 이상 포함하는 클래스를 추상 클래스라고 한다. 그리고 이렇게 재정의한 메소드를 부모 클래스의 객체 변수를 통해 사용하려면 부모 클래스의 객체 변수를 선언할 때 자식 클래스의 생성자를 이용해야 하는데, 이것을 클래스의 형 변환이라고 한다.

이번 섹션에서는 Java 상속을 기반으로 하여 추상 클래스와 추상 메소드, 그리고 형 변환이 적용된 코드를 읽고 해석하는 방법을 배운다.

2. Java 문제

다음 Java로 구현된 프로그램의 실행 결과를 확인하시오.

```
abstract class Animal {  
    String a = " is animal";  
    abstract void look();  
    void show() { System.out.println("Zoo"); } }  
class Chicken extends Animal {  
    Chicken() { look(); }  
    void look() { System.out.println("Chicken" + a); }  
    void display() { System.out.println("two wings"); } }  
public class Test {  
    public static void main(String[] args) {  
        Animal a = new Chicken();  
        a.show(); } }
```

1. Python 개요

Python은 객체지향 기능을 지원하는 스크립트 언어로, 다른 언어에 비해 문법이 간단하다는 장점이 있다.

Python을 C, Java와 비교했을 때 기본 작성법의 차이점은 다음과 같다.

- 변수의 자료형에 대한 선언이 없다.
- 문자와 문자열을 작은따옴표(''), 큰따옴표(“)로 구분 없이 사용할 수 있으며, 3개의 작은따옴표('''), 3개의 큰따옴표(“””)로도 묶어 표현할 수 있다.
- 문장의 끝을 의미하는 세미콜론(;)을 사용할 필요가 없다.
- if나 for와 같이 코드 블록을 포함하는 명령문을 작성할 때 중괄호({ }) 대신 콜론(:)과 여백으로 구분한다.

1. Python 개요

이번 섹션에서는 C, Java에서 배웠던 자료형, 입·출력, 제어문이 Python에서는 어떻게 수행되는지 학습하고, Python에서만 기본적으로 제공하는 기능이 포함된 프로그램 코드를 읽고 해석하는 방법을 학습한다. 이번 섹션에서 배울 내용은 다음과 같다.

- 자료형 : 리스트(List), 세트(Set)
- 입·출력 함수 : input, print
- 슬라이스(Slice)
- Range

2. Python 문제

“xyz321”을 입력했을 때 그 실행 결과를 확인하시오.

```
# -*- coding:utf-8 -*-  
x = input('입력 :')  
a = [ 'abc123', 'def456', 'ghi789' ]  
a.append(x)  
a.remove('def456')  
print(a[1][-3:], a[2][: -3], sep = ',')  
for i in range(3, 6):  
    print(i, end = ' ')
```

2. Python 문제

다음 Python 프로그램의 실행 결과를 확인하시오.

```
a = {'apple', 'lemon', 'banana'}  
a.update( {'kiwi', 'banana'} )  
a.remove('lemon')  
a.add('apple')  
for i in a:  
    print(i)
```



39. 절차적 프로그래밍 언어


1. 절차적 프로그래밍 언어

절차적 프로그래밍 언어는 일련의 처리 절차를 정해진 문법에 따라 순서대로 기술해 나가는 언어

- 절차적 프로그래밍 언어는 프로그램이 실행되는 절차를 중요시함
- 절차적 프로그래밍 언어는 데이터를 중심으로 프로시저를 구현하며, 프로그램 전체가 유기적으로 연결되어 있음
- 절차적 프로그래밍 언어는 자연어에 가까운 단어와 문장으로 구성됨
- 절차적 프로그래밍 언어는 과학 계산이나 하드웨어 제어에 주로 사용됨


39. 절차적 프로그래밍 언어

2. 절차적 프로그래밍 언어 종류

언어	특징
 C	<ul style="list-style-type: none">• 1972년 미국 벨 연구소의 데니스 리치에 의해 개발되었음• 시스템 소프트웨어를 개발하기 편리하여 시스템 프로그래밍 언어로 널리 사용됨• 자료의 주소를 조작할 수 있는 포인터를 제공함• 고급 프로그래밍 언어이면서 저급 프로그램 언어의 특징을 모두 갖춤• UNIX의 일부가 C언어로 구현되었음• 컴파일러 방식의 언어• 이식성이 좋아 컴퓨터 기종에 관계없이 프로그램을 작성할 수 있음
ALGOL	<ul style="list-style-type: none">• 수치 계산이나 논리 연산을 위한 과학 기술 계산용 언어• PASCAL과 C언어의 모체가 되었음

39. 절차적 프로그래밍 언어

2. 절차적 프로그래밍 언어 종류

언어	특징
COBOL 	<ul style="list-style-type: none">• 사무 처리용 언어• 영어 문장 형식으로 구성되어 있어 이해와 사용이 쉬움• 4개의 DIVISION으로 구성 (Identification, Environment, Data, Procedure)
FORTRAN	<ul style="list-style-type: none">• 과학 기술 계산용 언어• 수학과 공학 분야의 공식이나 수식과 같은 형태로 프로그래밍 할 수 있음



40. 객체지향 프로그래밍 언어

1. 객체지향 프로그래밍 언어

객체지향 프로그래밍 언어는 현실 세계의 개체(Entity)를 기계의 부품처럼 하나의 객체로 만들어, 기계적인 부품들을 조립하여 제품을 만들 듯이 소프트웨어를 개발할 때도 객체들을 조립해서 프로그램을 작성할 수 있도록 한 프로그래밍 기법

- 프로시저보다는 명령과 데이터로 구성된 객체를 중심으로 하는 프로그래밍 기법으로, 한 프로그램을 다른 프로그램에서 이용할 수 있도록함

40. 객체지향 프로그래밍 언어

2. 객체지향 프로그래밍 언어 종류

언어	특징
Java	<ul style="list-style-type: none">• 분산 네트워크 환경에 적용이 가능하며, 멀티스레드 기능을 제공하므로 여러 작업을 동시에 처리할 수 있음• 운영체제 및 하드웨어에 독립적이며, 이식성이 강함• 캡슐화가 가능하고 재사용성 높음
C++	<ul style="list-style-type: none">• C언어에 객체지향 개념을 적용한 언어• 모든 문제를 객체로 모델링하여 표현함
Smalltalk	<ul style="list-style-type: none">• 1세대 객체지향 프로그래밍 언어 중 하나로 순수한 객체지향 프로그래밍 언어• 최초로 GUI를 제공한 언어

41. 스크립트 언어


1. 스크립트 언어

스크립트 언어는 HTML 문서 안에 직접 프로그래밍 언어를 삽입하여 사용하는 것으로, 기계어로 컴파일 되지 않고 별도의 번역기가 소스를 분석하여 동작하게 하는 언어

- 게시판 입력, 상품 검색, 회원 가입 등과 같은 데이터베이스 처리 작업을 수행하기 위해 주로 사용
- 스크립트 언어는 클라이언트의 웹 브라우저에서 해석되어 실행되는 클라이언트용 스크립트 언어와 서버에서 해석되어 실행된 후 결과만 클라이언트로 보내는 서버용 스크립트 언어가 있음
 - 서버용 스크립트 언어 : ASP, JSP, PHP, 파이썬
 - 클라이언트용 스크립트 언어 : 자바 스크립트(Java Script)

41. 스크립트 언어

2. 스크립트 언어 종류

 자바 스크립트 (Java Script)	<ul style="list-style-type: none">• 넷스케이프(Netscape)의 브렌던 아이크(Brendan Eich)가 개발한 클라이언트용 스크립트 언어• 웹 페이지의 동작을 제어하며, 변수 선언이 필요 없음• 서버에서 데이터를 전송할 때 아이디, 비밀번호, 수량 등의 입력 사항을 확인하기 위한 용도로 많이 사용됨
ASP (Active Server Page)	<ul style="list-style-type: none">• 서버 측에서 동적으로 수행되는 페이지를 만들기 위한 언어로 마이크로 소프트사에서 제작하였음• Windows 계열에서만 수행 가능한 프로그래밍 언어
JSP (Java Server Page)	Java로 만들어진 서버용 스크립트 언어로, 다양한 운영체제에서 사용이 가능함

41. 스크립트 언어

2. 스크립트 언어 종류

PHP(Professional Hypertext Preprocessor)	<ul style="list-style-type: none">• 서버용 스크립트 언어로, Linux, Unix, Windows 운영체제에서 사용 이 가능함• C, Java 등과 문법이 유사하므로 배우기 쉬워 웹 페이지 제작에 많이 사용
파이썬(Python)	객체지향 기능을 지원하는 대화형 인터프리터 언어로, 플랫폼에 독립적이고 문법이 간단하여 배우기 쉬움



42. 선언형 언어


1. 선언형 언어

선언형 언어는 명령형 언어와 반대되는 개념의 언어로, 명령형 언어가 문제를 해결하기 위한 방법을 기술한다면 선언형 언어는 프로그램이 수행해야 할 문제를 기술하는 언어

- 선언형 언어는 목표를 명시하고 알고리즘은 명시하지 않음
- 선언형 언어에는 함수형 언어와 논리형 언어 등이 있음

42. 선언형 언어

1. 선언형 언어

<u>함수형 언어</u> 	<ul style="list-style-type: none">• 수학적 함수를 조합하여 문제를 해결하는 언어로, 알려진 값을 함수에 적용하는 것을 기반으로함• 적용형 언어라고도함• 재귀호출이 자주 이용됨• 병렬 처리에 유리함• 종류 : LISP
논리형 언어	<ul style="list-style-type: none">• 기호 논리학에 기반을 둔 언어로, 논리 문장을 이용하여 프로그램을 표현하고 계산을 수행함• 선언적 언어라고도함• 반복문이나 선택문을 사용하지 않음• 비절차적 언어임• 종류 : PROLOG

42. 선언형 언어

2. 선언형 언어 종류

HTML	인터넷의 표준 문서인 하이퍼텍스트 문서를 만들기 위해 사용하는 언어로, 특별한 데이터 타입이 없는 단순한 텍스트이므로 호환성이 좋고 사용이 편리함
LISP	<ul style="list-style-type: none">• 인공지능 분야에 사용되는 언어• 기본 자료 구조가 연결 리스트 구조이며, 재귀(Recursion) 호출을 많이 사용함
PROLOG	논리학을 기초로 한 고급 언어로, 인공 지능 분야에서의 논리적인 추론이나 리스트 처리 등에 주로 사용
XML	<ul style="list-style-type: none">• 기존 HTML의 단점을 보완하여 웹에서 구조화된 폭넓고 다양한 문서들을 상호 교환할 수 있도록 설계된 언어• HTML에 사용자가 새로운 태그(Tag)를 정의할 수 있으며, 문서의 내용과 이를 표현하는 방식이 독립적임
Haskell	<ul style="list-style-type: none">• 함수형 프로그래밍 언어로 부작용(Side Effect)이 없음• 코드가 간결하고 에러 발생 가능성이 낮음

43. 라이브러리

1. 라이브러리

라이브러리는 프로그램을 효율적으로 개발할 수 있도록 자주 사용하는 함수나 데이터들을 미리 만들어 모아 놓은 집합체

- 자주 사용하는 함수들의 반복적인 코드 작성을 피하기 위해 미리 만들어 놓은 것으로, 필요할 때는 언제든지 호출하여 사용할 수 있음
- 라이브러리에는 표준 라이브러리와 외부 라이브러리가 있음
- **표준 라이브러리** : 프로그래밍 언어에 기본적으로 포함되어 있는 라이브러리로, 여러 종류의 모듈이나 패키지 형태
- **외부 라이브러리** : 개발자들이 필요한 기능들을 만들어 인터넷 등에 공유해 놓은 것으로, 외부 라이브러리를 다운받아 설치한 후 사용함




43. 라이브러리

2. C언어의 대표적인 표준 라이브러리

C언어는 라이브러리를 헤더 파일로 제공하는데, 각 헤더 파일에는 응용 프로그램 개발에 필요한 함수들이 정리되어 있음

- C언어에서 헤더 파일을 사용하려면 '#include <stdio.h>'와 같이 include문을 이용해 선언한 후 사용해야함

헤더 파일	기능
 stdio.h	<ul style="list-style-type: none">• 데이터의 입·출력에 사용되는 기능들을 제공함• 주요 함수 : printf, scanf, fprintf, fscanf, fclose, fopen 등
math.h	<ul style="list-style-type: none">• 수학 함수들을 제공함• 주요 함수 : sqrt, pow, abs 등
string.h	<ul style="list-style-type: none">• 문자열 처리에 사용되는 기능들을 제공함• 주요 함수 : strlen, strcpy, strcmp 등
stdlib.h	<ul style="list-style-type: none">• 자료형 변환, 난수 발생, 메모리 할당에 사용되는 기능들을 제공함• 주요 함수 : atoi, atof, srand, rand, malloc, free 등
time.h	<ul style="list-style-type: none">• 시간 처리에 사용되는 기능들을 제공함• 주요 함수 : time, clock 등

3. Java언어의 대표적인 표준 라이브러리

Java는 라이브러리를 패키지에 포함하여 제공하는데, 각 패키지에는 Java 응용 프로그램 개발에 필요한 메소드들이 클래스로 정리되어 있음

- Java에서 패키지를 사용하려면 'import java.util'과 같이 import문을 이용해 선언한 후 사용해야 함
- import로 선언된 패키지 안에 있는 클래스의 메소드를 사용할 때는 클래스와 메소드를 마침표(.)로 구분하여 'Math.abs()'와 같이 사용함

43. 라이브러리

3. Java언어의 대표적인 표준 라이브러리

패키지	기능
java.lang	<ul style="list-style-type: none">• Java에 기본적으로 필요한 인터페이스, 자료형, 예외처리 등에 관련된 기능 제공함• import문 없이도 사용할 수 있음• 주요 클래스 : String, System, Process, Runtime, Math, Error 등
java.util	<ul style="list-style-type: none">• 날짜 처리, 난수 발생, 복잡한 문자열 처리 등에 관련된 기능을 제공함• 주요 클래스 : Date, Calender, Random, StringTokenizer 등
java.io	<ul style="list-style-type: none">• 파일 입·출력과 관련된 기능 및 프로토콜을 제공함• 주요 클래스 : InputStream, OutputStream, Reader, Writer 등
java.net	<ul style="list-style-type: none">• 네트워크와 관련된 기능을 제공함• 주요 클래스 : Socket, URL, InetAddress 등
java.awt	<ul style="list-style-type: none">• 사용자 인터페이스(UI)와 관련된 기능을 제공함• 주요 클래스 : Frame, Panel, Dialog, Button, Checkbox 등

4. Python언어의 대표적인 표준 라이브러리


Python은 Java와 동일하게 라이브러리를 패키지로 제공하며, 패키지에는 응용 프로그램 개발에 필요한 클래스와 메소드들이 정의되어 있음

- Python에서 라이브러리를 사용하려면 'import random'과 같이 선언하고, 사용할 때는 마침표(.)로 구분하여 'random.choice()'와 같이 사용함

클래스	기능
내장 함수	<ul style="list-style-type: none">• Python에 기본적인 인터페이스로, import문이나 클래스명 없이도 사용할 수 있음• 주요 메소드 : abs(), slice(), pow(), print() 등
os	<ul style="list-style-type: none">• 운영체제와 상호 작용하기 위한 기능을 제공함• 주요 메소드 : getcwd(), chdir(), system() 등

43. 라이브러리

4. Python언어의 대표적인 표준 라이브러리

클래스	기능
re 	<ul style="list-style-type: none">• 고급 문자열 처리를 위한 기능을 제공함• 주요 메소드 : findall(), sub() 등
math	<ul style="list-style-type: none">• 복잡한 수학 연산을 위한 기능을 제공함• 주요 메소드 : cos(), log() 등
random	<ul style="list-style-type: none">• 무작위 선택을 위한 기능을 제공함• 주요 메소드 : choice(), sample(), random(), randrange() 등
statistics	<ul style="list-style-type: none">• 통계값 산출을 위한 기능을 제공함• 주요 메소드 : mean(), median(), variance() 등
datetime	<ul style="list-style-type: none">• 날짜와 시간 조작을 위한 기능을 제공함• 주요 메소드 : today(), date(), strftime() 등

Unit A

참고자료



문헌

1. <http://www.ncs.go.kr>
2. 2022 시나공 정보처리산업기사 실기, 김정준, 강윤석, 길벗알앤디 저 | 길벗 | 2022년 03월 14일

실습 자료

1. ...

감사합니다.

❖ Mobile: 010-9591-1401

❖ E-mail: onlooker2zip@naver.com / dkkim@lklab.org

