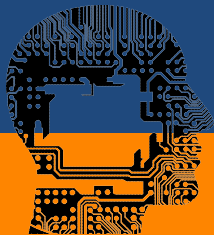




한국IT진흥부설

정보보호교육학원

아이섹



자바 프로그래밍 기초

API

By SoonGu Hong(Kokono)

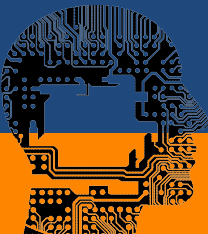


한국IT진흥부설

정보보호교육학원 아이섹

파이썬 기초

1. API란?



1-1. API란?



- API란 Application Programming Interface의 약자로 프로그램 개발에 자주 사용되는 클래스 및 인터페이스의 모음을 말합니다.
- 우리가 자주 사용한 String클래스나 System클래스도 모두 API에 속한 클래스들입니다.

1-2. Java API Document

OVERVIEW

MODULE

PACKAGE

CLASS

USE

TREE

DEPRECATED

INDEX

HELP

ALL CLASSES

Java® Platform, Standard Edition & Java Development Kit Version 11 API Specification

- API는 일반적으로 API Document라는 문서화된 파일을 가지고 있습니다.
- API를 개발한 개발자들이 해당 API의 사용법을 써놓은 문서라고 보시면 이해하기 쉬울 것입니다.
- 자바 API의 경우 웹문서로 내용을 제공하고 있습니다.

* java 8: <http://docs.oracle.com/javase/8/docs/api>

* java 11: <https://docs.oracle.com/en/java/javase/11/docs/api/index.html>



한국IT진흥부설

정보보호교육학원 아이섹

파이썬 기초

2. java.lang 패키지

2-1. java.lang 패키지

Module java.base

Package java.lang

Provides classes that are fundamental to the design of the Java programming language. The most important classes are `Object`, which is the root of the class hierarchy, and `Class`, instances of which represent classes at run time.

Frequently it is necessary to represent a value of primitive type as if it were an object. The wrapper classes `Boolean`, `Character`, `Integer`, `Long`, `Float`, and `Double` serve this purpose. An object of type `Double`, for example, contains a field whose type is `double`, representing that value in such a way that a reference to it can be stored in a variable of reference type. These classes also provide a number of methods for converting among primitive values, as well as supporting such standard methods as `equals` and `hashCode`. The `Void` class is a non-instantiable class that holds a reference to a `Class` object representing the type `void`.

The class `Math` provides commonly used mathematical functions such as sine, cosine, and square root. The classes `String`, `StringBuffer`, and `StringBuilder` similarly provide commonly used operations on character strings.

Classes `ClassLoader`, `Process`, `ProcessBuilder`, `Runtime`, `SecurityManager`, and `System` provide "system operations" that manage the dynamic loading of classes, creation of external processes, host environment inquiries such as the time of day, and enforcement of security policies.

Class `Throwable` encompasses objects that may be thrown by the `throw` statement. Subclasses of `Throwable` represent errors and exceptions.

- java.lang패키지는 자바 프로그램의 기본적인 클래스들을 담고 있는 패키지입니다.
- 그래서 import문 없이도 해당 패키지의 클래스들을 바로 사용할 수 있습니다.
- 우리가 import없이 사용했던 Math, String, System등이 lang패키지의 클래스들입니다.

2-2. java.lang 주요 클래스

- java.lang패키지의 주요 클래스들을 설명한 표입니다.

클래스	용도
Object	자바 최상위 클래스로 사용
System	표준 입,출력 제어 및 시스템 접근
Class	클래스를 동적 로딩, 리플렉션할 때 사용
String	문자열을 저장하고 제어하기 위해 사용
StringBuilder	문자열을 효율적으로 저장할 때 사용
Math	수학 함수를 이용할 때 사용
Byte, Short, Integer, Long, Float, Double, Boolean, Character	기본 타입 데이터를 객체로 만들 때 사용 문자열을 기본타입으로 변환할 때 사용 입력값 검사에 사용

2-3. Object 클래스

- 자바의 클래스는 명시적으로 extends를 쓰지 않으면 암묵적으로 Object를 상속받도록 구성되어있습니다.
- 즉, 자바의 모든 클래스는 Object클래스의 메서드를 오버라이딩하여 사용할 수 있습니다.

주요 메서드	용도
equals()	객체 비교
hashCode()	객체를 식별할 정수값 리턴
toString()	객체의 문자정보 리턴
clone()	원본 객체를 복사할 때 사용
finalize()	객체 소멸자 메서드

2-4. System 클래스

- 운영체제의 기능을 이용하게 해주는 클래스입니다.
- 프로그램 종료, 키보드 입력, 모니터 출력, 메모리 정리, 현재 시간 읽기, 프로퍼티 읽기 등의 기능을 제공합니다.
- 모든 필드와 메서드는 static으로 되어있습니다.

주요 메서드	용도
exit()	프로그램 종료
gc()	쓰레기 수집기 실행(garbage collector)
currentTimeMillis()	현재 시간 읽기(밀리초로 리턴)
getProperty()	시스템 프로퍼티 읽기
getEnv()	환경 변수 읽기

2-5. Class 클래스

- 자바는 클래스와 인터페이스의 메타 데이터를 Class 클래스로 관리합니다.
- 여기서 메타 데이터란 클래스명, 필드정보, 생성자정보, 메서드정보를 말합니다.

주요 메서드	용도
getClass(), forName()	Class객체 얻기
getDeclaredConstructors(), getDeclaredFields(), getDeclaredMethods()	리플렉션 메서드들. 생성자, 필드, 메서드 정보를 얻어옴
newInstance()	동적 객체 생성

2-6. String 클래스

- String 클래스는 문자열 저장 및 문자열 조작메서드들을 제공합니다.

주요 메서드	용도
charAt()	문자열의 특정 인덱스 문자 리턴
getBytes()	문자열을 바이트배열로 변환
indexOf()	특정 문자의 인덱스 리턴
length()	문자열의 길이 리턴
replace()	문자열을 대체
substring()	문자열 잘라내기
toLowerCase(), toUpperCase()	영문 대/소문자 변환
trim()	문자열 공백 제거
valueOf()	다른 타입 데이터를 문자열로 변환
split()	문자열을 분리하여 배열로 리턴

2-7. Wrapper 클래스

- Wrapper클래스는 자바의 기본타입 8가지를 객체로 포장한 클래스들을 말합니다.
- Byte, Short, Integer, Long, Float, Double, Boolean, Character 총 8가지 클래스를 말합니다.

주요 메서드	용도
Integer.parseInt("문자열")	문자열을 정수로 변환
Double.parseDouble("문자열")	문자열을 실수로 변환
Boolean.parseBoolean("문자열")	문자열을 논리로 변환

2-8. Math 클래스

- Math 클래스는 수학 관련 함수들을 제공합니다.
- 모든 메서드는 static으로 구성되어 있습니다.

주요 메서드	용도
abs(숫자)	절대값
ceil(숫자)	올림값
floor(숫자)	버림값
round(숫자)	반올림값
max(숫자1, 숫자2)	최대값
min(숫자1, 숫자2)	최소값
random()	랜덤값

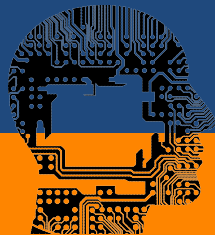


한국IT진흥부설

정보보호교육학원 아이섹

파이썬 기초

3. java.util패키지



3-1. java.util 주요 클래스

- java.util패키지는 프로그램 개발의 유틸리티성 클래스들을 포함하고 있습니다.
- 다음은 java.util 패키지의 주요 클래스들을 설명한 표입니다.

클래스	용도
Arrays	배열복사, 정렬, 비교 등의 기능을 지원
Random	난수 생성 기능을 지원
Date, Calendar	날짜 표현 기능을 지원
Collections	다양한 자료구조를 지원

3-2. Arrays 클래스

- Arrays 클래스는 배열을 다루는 다양한 기능들(복사, 정렬, 탐색 등)을 제공합니다.

주요 메서드	용도
copyOf(원본배열, 복사할 길이)	원본의 0번부터 길이만큼 복사
copyOfRange(원본배열, 시작 idx, 끝idx)	원본의 시작인덱스부터 끝인덱스까지 복사한 배열 리턴
equals(배열1, 배열2)	두 배열의 얕은 비교
deepEquals(배열1, 배열2)	두 배열의 깊은 비교(중첩 배열까지 비교)
sort(배열)	배열 항목을 오름차순으로 정렬
toString(배열)	배열 항목을 문자열로 리턴
fill(배열, 값)	전체 배열에 동일한 값을 저장
fill(배열, 시작idx, 끝idx, 값)	시작idx부터 끝idx까지 동일값 저장
binarySearch(배열, 찾는값)	이진 탐색 수행

3-3. Random 클래스

- Random 클래스는 난수를 생성하는 다양한 메서드들을 제공합니다.

주요 메서드	용도
nextBoolean()	랜덤 논리값 리턴
nextDouble()	$0.0 \leq \sim < 1.0$
nextInt()	int 허용범위안에서 랜덤 정수 리턴
nextInt(int n)	$0 \leq \sim < n$

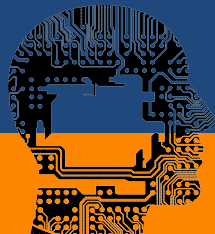


한국IT진흥부설

정보보호교육학원 아이섹

파이썬 기초

4. 날짜 관련 API



4-1. 날짜 관련 패키지 java.time

- java8 이전까지 활용되던 Date, Calendar 클래스의 불편성 때문에 java8 버전부터는 time 패키지가 추가되어 날짜 데이터를 편리하게 다룰 수 있게 되었습니다.
- now() 메서드로 현재 시간정보를 읽어올 수 있으며, of() 메서드로 특정 시간정보를 만들 수 있습니다.

클래스	용도
LocalDate	날짜 정보만을 저장, 제어
LocalTime	시간 정보만을 저장, 제어
LocalDateTime	위 2개 클래스를 결합한 형태

4-2. 날짜와 시간에 대한 정보 얻기

- **LocalDate**와 **LocalTime**에서는 각각 날짜와 시간 정보를 얻기 위한 메서드들을 제공하며 **LocalDateTime**은 모두 제공합니다.

주요 메서드	용도
<code>getYear()</code>	연도 정보 (숫자)
<code>getMonth()</code>	월 정보 (문자)
<code>getMonthValue()</code>	월 정보 (숫자)
<code>getDayOfYear()</code>	1년중 몇번째 일
<code>getDayOfMonth()</code>	일 정보(숫자)
<code>getDayOfWeek()</code>	요일 정보 (문자)
<code>getHour()</code>	시간 정보(숫자 0~23)
<code>getMinute()</code>	분 정보
<code>getSecond()</code>	초 정보

4-3. 날짜와 시간 조작하기

주요 메서드	용도
plusYears(long) / minusYears(long)	년 더하기/빼기
plusMonths(long) / minusMonths(long)	월 더하기/빼기
plusDays(long) / minusDays(long)	일 더하기/빼기
plusWeeks(long) / minusWeeks(long)	주 더하기/빼기
plusHours(long) / minusHours(long)	시 더하기/빼기
plusMinutes(long) / minusMinutes(long)	분 더하기/빼기
plusSeconds(long) / minusSeconds(long)	초 더하기/빼기

4-4. 날짜와 시간 비교하기

주요 메서드	용도
isAfter(other)	이후 날짜인지 비교
isBefore(other)	이전 날짜인지 비교
isEqual(other)	동일 날짜인지 비교

4-5. 날짜와 시간 포매팅

Module `java.base`

Package `java.time.format`

Class `DateTimeFormatter`

`java.lang.Object`

`java.time.format.DateTimeFormatter`

```
public final class DateTimeFormatter
```

```
extends Object
```

Formatter for printing and parsing date-time objects.

This class provides the main application entry point for printing and parsing and provides common implementations of `DateTimeFormatter`:

- Using predefined constants, such as `ISO_LOCAL_DATE`
- Using pattern letters, such as `uuuu-MMM-dd`
- Using localized styles, such as `long` or `medium`

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/time/format/DateTimeFormatter.html>

감사합니다
THANK YOU