

# 1. java.util 패키지

---

## 1. Calendar

---

1. 기존에 날짜형식을 사용할 때는 Date클래스를 주로 사용했었는데 Date 클래스의 기능이 부족하고 더이상 사용할 수 없는 메소드나 속성들이 많아지면서 날짜형식은 Calendar클래스를 사용해서 표현하도록 권장되고 있다.
2. Calendar클래스는 추상클래스이기 때문에 인스턴스화가 불가능하여 직접 객체를 만들지 못하고 getInstance() 메소드를 통해서 인스턴스를 얻어서 사용한다.
3. GregorianCalendar 클래스가 Calendar를 상속받아 구현된 클래스. getInstance() 메소드 사용 시 GregorianCalendar 클래스의 객체를 얻을 수 있다.
4. Calendar 객체 얻는 방식
  - Calendar cal = Calendar.getInstance(); => 시스템의 현재 날짜와 시간을 가지고 있는 객체
5. Date <-> Calendar
  - Calendar를 Date타입으로 변환
    - Calendar cal = Calendar.getInstance();
    - Date d = new Date(cal.getTimeInMillis());
  - Date를 Calendar타입으로 변환
    - Date d = new Date();
    - Calendar cal = Calendar.getInstance();
    - cal.setTime(d);

## 2. SimpleDateFormat

---

1. Date나 Calendar를 사용해서 포맷에 맞춰 출력할 때는 SimpleDateFormat클래스를 사용한다.
2. SimpleDateFormat sdf = new SimpleDateFormat(형식화될 포맷);
3. sdf.format(Date);
4. 형식화될 포맷
  - y: 년도
  - M: 월
  - w: 년의 몇번째 주인지(1~53)
  - W: 월의 몇번째 주인지(1~5)

- d : 몇일인지(1~31)
- H : 시간(0~23)
- m : 분(0~59)
- s : 초(0~59)

yyyy-MM-dd HH:mm:ss(4자리 년도에 월일시분초 2자리로 표출), yy<sup>l</sup>

### 3. LocalDateTime, ZonedDateTime

---

1. LocalDateTime은 LocalDate, LocalTime 두개의 클래스의 기능을 합쳐놓은 클래스.
2. ZonedDateTime은 LocalDateTime에 타임존 개념을 추가한 클래스.
3. LocalDate, LocalTime을 이용해서 LocalDateTime 만들기
  - of(LocalDate, LocalTime) : 매개변수로 주어진 LocalDate, LocalTime을 이용해서 LocalDateTime을 생성
  - LocalDate.atTime(LocalTime) LocalTime.atDate(LocalDate)
  - LocalDate.atStartOfDay(); => 00시 00분 00초로 시간이 지정됨
4. LocalDateTime을 LocalDate, LocalTime으로 분리하는 방법
  - LocalDateTime.toLocalDate() : LocalDate로 변환
  - LocalDateTime.toLocalTime() : LocalTime으로 변환

```
LocalDateTime ldt = LocalDateTime.of(년도, 월, 날짜, 시간, 분, 초);
LocalDateTime ldt = LocalDateTime.now(); => 현재 년월일 시분초
```

6. ZonedDateTime을 만드는 방법
  - LocalDateTime의 atZone() 메소드로 Zone정보를 추가. Zone정보를 얻는 방식은 ZoneId.of("Zone명칭(Seoul/Asia, America/NewYork)")
  - ZonedDateTime.now().withZoneSameInstant(ZoneId);
7. ZoneOffset은 UTC(기준점 +-0)부터 얼마나 떨어져있는지를 표현. 서울 + 9.
8. OffsetDateTime은 ZoneId대신 ZoneOffset을 사용하는 클래스. 같은 존에 있는 PC가 서로 통신할 때는 LocalDateTime으로 충분하지만 다른 존에 있는 PC와 통신할 때는 OffsetDateTime이 필요하다.
9. ZonedDateTime, OffsetDateTime의 of 메소드
  - ZonedDateTime.of(date, time, zid);
  - OffsetDateTime.of(date, time, zOffset);
  - ZonedDateTime의 객체.toOffsetDateTime(); : ZonedDateTime의 객체를 OffsetDateTime 타입으로 변경

## 10. ZonedDateTime의 변환

- toLocalDate() : LocalDate 타입으로 변환
- toLocalTime()
- toLocalDateTime()