



# 기본**API** 클래스



- 자바프로그램에서 **API**의 역할을 안다.
  - 프로그래밍시, **API**를 활용할 줄 안다.
  - 사용을 많이 하는 클래스들에 대한 활용 방법을 익힌다.
    - **String**
    - **Object**
    - **StringBuffer**
    - **Arrays**
    - **Wrapper**
    - **Date, Calendar**
-



## 생각해봅시다 :

- 자바 프로그래밍을 할 때와 **API**를 이용할 때를 구분하는 기준은 무엇인가?
  - **API**의 내용을 빠르게 효과적으로 찾아서 활용하는 방법은 없는가?
-



# 자바 **API** 도큐먼트 :

- **API(Application Programming Interface)** : 자바에서 라이브러리(library)라고 부르기도 하는데, 프로그램 개발에 자주 사용되는 클래스 및 인터페이스 모음.
- <JDK설치 경로>/jre/lib/rt.jar 라는 압축파일에 저장
- **API** 도큐먼트는  
<http://docs.oracle.com/javase/8/docs/api>
- **API** 세 개의 프레임으로 나뉘어짐
  1. 패키지 전체목록
  2. 패키지에 속하는 클래스와 인터페이스
  3. 중앙 프레임 - 상세 설명
- 상세 내용 중 메서드
  - 1) return값    2) 메서드명    3)param으로 입력할 값
  - return값에 대한 상세설명



## 주요 패키지들 :

- **java.lang** 패키지 : **import** 없이 바로 사용할 수 있는 **default** 패키지
    - **Object** : 자바 클래스(**api** 클래스, 사용자정의 클래스)의 최상위 클래스로 사용
    - **System** :
      - **.in** : 표준 입력 장치(키보드)로 부터 데이터를 입력받을 때 사용
      - **.out** : 표준 출력 장치(모니터)로 출력하기 위해 사용
      - **JVM**을 종료시킬 때 사용
      - 쓰레기 수집기(사용하지 않는 메모리 수집) 실행 요청할 때 사용.
-



## 주요 패키지들 :

- **Class** : 클래스를 메모리 로딩할 때 사용
    - **Class.forName("지정된 클래스")** 지정된 클래스를 메모리에 올림
  - **String** : 문자열을 지정하고 여러가지 정보를 얻을 때 사용
  - **StringBuffer, StringBuilder** : 문자열을 저장하고 내부 문자열을 조작할 때 사용
  - **Math** : 수학 함수
  - **Wrapper (Integer, Double..)** : 기본 type의 데이터를 갖는 객체를 만들 때 사용.  
**method()** 통해서 활용
-



## java.util.패키지 :

- Arrays : 배열을 조작(비교, 복사, 정렬, 찾기)
  - Calendar : 운영체제의 날짜와 시간
  - Date : 날짜와 시간 정보를 저장
  - Objects : 객체 비교, null 여부 등 조사할 때
  - StringTokenizer : 특정 문자로 구분된 문자열
  - Random : 난수를 얻을 때 사용
-



## 객체 해시코드(**hashCode()**):

- 객체를 식별할 하나의 정수값
  - `hashCode()` : 객체의 메모리번지를 이용해서 해시코드를 만듦
- 객체를 식별 프로세스..
  - `hashCode()` → `equals()` 리턴값 ➔ 동일 객체인식





## 객체 복제(clone()):

- 원본 객체의 필드값과 동일 값을 가지는 새로운 객체를 생성한 것을 말한다.
  - `Person p1 = new Person();`
  - `Person p2 = p1;`
  - `p2.setName("홍길동");` // p1에 영향을 미침
- 원복 객체의 필드값을 보호하고자 할 때, 활용
- 기본 구조
  - `try{`
    - `Object obj = clone();`
  - `}catch(CloneNotSupportedException e){}`



## 객체 소멸자 :

- 참조하지 않는 배열이나 객체는 쓰레기 수집기(**Garbage Collector**)가 힙영역에서 자동적으로 소멸시킨다.
- 쓰레기 수집기는 객체 소멸직전에 객체 소멸자(**finalize()**)를 실행시켜 소멸처리
- 클래스 참조 = **new** 객체();
- 참조 = **null**; // 객체를 쓰레기 만듦
- 명시적으로 쓰레기 수집기 실행 요청..
  - **System.gc();** ➔ 소멸시키는 내용을 **console**확인



## System 클래스 :

- 자바 프로그램은 JVM 위에서 실행된다.  
운영체제의 모든 기능을 자바 코드로 직접 접근하지는 못 한다.
- 프로그램 종료, 키보드 입력, 모니터 출력, 메모리 정리, 현재 시간읽기, 시스템 프로퍼티 읽기, 환경 변수 읽기 등으로 한정된 부분으로 가능하다.
  - `System.exit(5)` : JVM 종료 요청
    - 0:정상 종료, 그외 값은 비정상 종료
  - `System.gc()` : 쓰레기 수집기 실행
  - `System.currentTimeMillis()`: 현재시간 1/1000초



## 정리 및 확인하기(숙제- 정답 ):

---



**감사합니다 !**

---