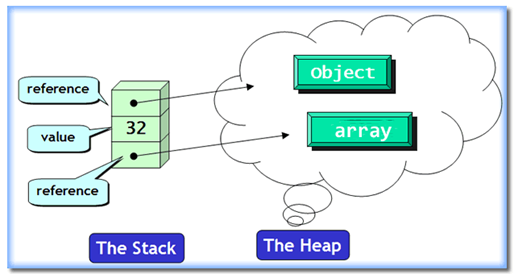
**Stack & Heap  
**

1. **Stack**- 스택은 메모리 크기가 정해져 있다. (정적 메모리 할당 / 불변 / int, Boolean, double 등)  
   - 장점 : 힙에 비해 액세스가 매우 빠름 / 변수를 명시적으로 할당, 해제 필요 없음  
   - 힙보다 빠른 이유는 힙은 스택의 주소를 이용하여 액세스를 하기 때문에 즉, 힙은 스택을 거쳐가야 하기 때문에 상대적으로 느린 것이다.  
   - 단점 : 메모리 크기를 변경할 수 없다.  
   테이블이(가) 표시된 사진

   자동 생성된 설명  
   - hel, lo는 사라지지 않는다. 따라서, 무분별한 +, concat은 메모리 효율을 떨어뜨린다.  
   - String 보다는 StringBuffer, StringBuilder 사용을 권장한다.
2. **Heap**- 메모리 크기가 정해져 있지 않다. (동적 메모리 할당 / 가변 / string, object, array 등)  
   - 장점 : 메모리 크기를 변경할 수 있다.(효율적)  
   - 단점 : 상대적으로 느린 액세스  
   테이블이(가) 표시된 사진

   자동 생성된 설명  
   - Stack과 달리 hel과 lo가 삭제된다. 메모리 효율이 좋다는 것을 알 수 있다.
3. **결론**  
   - 무조건적으로 힙이 스택보다 좋다는 것은 아니다.  
   - 각 역할에 맞게 사용하면 효율을 극대화시킬 수 있다.  
   - 결국 스택에는 주소가, 힙에는 값이 들어간다고 보면 된다.