

## 📖 황금키워드 선정 공식

### 1. 용어 정의

- **S** = 월간 검색수 (*monthly search volume*)
  - ▮ 검색 API/광고 키워드 툴에서 얻는 값.
- **D** = 누적 문서수 (*published docs count*)
  - ▮ 네이버 블로그 검색 API의 **total** 값.
- **C** = 경쟁도 (*competition ratio*)

$$C = \frac{D}{S}$$

- 값이 낮을수록 좋다.
- 필수 조건
  1.  $S > 0$
  2.  $D > 0 \leftarrow$  **D가 0인 경우는 없음**
  3.  $S > D$
  4.  $C \leq 1$  (즉, 문서 수가 검색 수보다 많아선 안 됨)

### 2. 단계별 컷라인

단계	블로그 단계 설명	문서수 상한 (D_max)	검색수 범위 (S_min ~ S_max)	공통조건
1단계	초기(방문자수 적음)	$\leq 500$	1 ~ 2,000	$S > D \wedge C \leq 1$
2단계	성장 단계	$\leq 1,000$	1 ~ 3,000	$S > D \wedge C \leq 1$
3단계	중급	$\leq 2,000$	1 ~ 10,000	$S > D \wedge C \leq 1$
4단계	상급(라이트)	$\leq 3,000$	1 ~ 15,000	$S > D \wedge C \leq 1$
5단계	상급	$\leq 3,000$	1 ~ 20,000	$S > D \wedge C \leq 1$

### 3. 채택 규칙

1. 선택한 단계의 컷라인(D\_max, S\_min, S\_max) 범위에 들어야 함.
2. 공통조건 충족:
  - $S > 0, D > 0$
  - $S > D$
  - $C = \frac{D}{S} \leq 1$

### 4. 정렬 규칙

1. 경쟁도  $C = D/S$  낮은 순 (오름차순)
2. **S** 큰 순 (내림차순)
3. **D** 작은 순 (오름차순)

▮ 즉, 문서 대비 검색 수 비율이 낮고 → 검색 수 자체는 많고 → 문서 수는 적은 순으로 정렬.

### 5. 의사코드

```
LIMITS = {
    1: {"D_max": 500, "S_min": 1, "S_max": 2000},
    2: {"D_max": 1000, "S_min": 1, "S_max": 3000},
    3: {"D_max": 2000, "S_min": 1, "S_max": 10000},
    4: {"D_max": 3000, "S_min": 1, "S_max": 15000},
    5: {"D_max": 3000, "S_min": 1, "S_max": 20000},
}

def accept(S, D, stage):
    if stage not in LIMITS:
        return False
    if S <= 0 or D <= 0 or not (S > D):
        return False
    L = LIMITS[stage]
    return L["S_min"] <= S <= L["S_max"] and D <= L["D_max"]

def rank(cands, stage):
    """
    cands: [(키워드, S, D), ...]
    반환: [{ 'kw': str, 'S': int, 'D': int, 'C': float }, ...]
    """
    filtered = []
    for kw, S, D in cands:
        if accept(S, D, stage):
            C = D / S
            filtered.append({"kw": kw, "S": S, "D": D, "C": C})
    return sorted(filtered, key=lambda r: (r["C"], -r["S"], r["D"]))
```

### 6. 쉬운 설명

- 왜  $C \leq 1$ ?
  - 문서 수가 검색 수를 초과하면 이미 과포화 상태라서 신규 진입 난이도가 높음.
- 왜  $S > D$ ?
  - 검색량이 문서량보다 많아야 내 글이 노출될 확률이 높음.
- 왜 단계별 컷라인?
  - 블로그 규모(문서 수·방문자)에 맞춰 무리하지 않는 키워드 경쟁 강도를 선택하기 위함.
- 정렬 기준
  - 경쟁도 낮은 게 1순위, 그 다음은 트래픽(검색량), 마지막으로 문서 수 적은 것.

원하면 이 규칙을 CSV·JSON·Python 모듈 형태로 저장해 줄 수도 있어.  
그렇게 하면 나중에 데이터 뽑아서 바로 필터+정렬 돌릴 수 있음.

바로 파일로도 만들어줄까?