

단단한 취미 생활은 좋은 자료에서 시작된다. 검색창에 키워드를 던져 넣는 것만으로는 깊이가 잘 쌓이지 않는다. 스스로 체계와 기준을 세우고, 검증된 링크를 모아두면 학습 속도가 눈에 띄게 빨라진다. 아래 글은 사진, 요리, 코딩, 음악 네 가지 분야를 중심으로 실전에서 오래 써 온 링크모음 전략과 추천 주소모음을 함께 정리했다. 몇몇 링크는 유명하고 오래된 자료이고, 몇몇은 최근 업데이트가 활발한 곳들이다. 각 분야에서 자주 생기는 오해와 시행착오도 함께 짚었다. 주소모음은 단순한 북마크가 아니다. 자신만의 학습 도로망을 설계하는 일에 가깝다.

좋은 링크를 가려내는 간단한 기준

무엇을 저장하고 무엇을 버릴지에 대한 기준이 있으면 북마크가 쓰레기장이 되지 않는다. 한때 유행했던 글이나 광고성 페이지는 시간이 조금만 지나도 가치가 크게 떨어진다. 반대로, 원리와 방법을 중심으로 설명하는 자료, 그리고 문서화를 잘하는 커뮤니티는 오래 간다.

다음 항목을 염두에 두면 링크 품질을 빠르게 판별할 수 있다.

1. 최근 12개월 내 업데이트 여부와 유지 관리 주체가 명확한가
2. 개념의 정의, 예시, 한계까지 설명하는가, 아니면 요령만 나열하는가
3. 과장된 제목, 클릭 유도형 문구보다 근거와 데이터가 있는가
4. 댓글과 토론이 건장한가, 출처나 참고문헌을 제시하는가
5. 광고와 제휴 링크가 본문 이해를 방해하지 않는가

이 기준을 주소모음에 계속 적용하면, 한 해가 지나도 북마크의 체력이 유지된다.

사진, 장비보다 먼저 배우는 눈과 손

카메라가 좋아도 앵글과 빛을 모르면 사진이 밋밋하다. 반대로, 휴대폰 카메라로도 빛과 색, 구도만 잡히면 인상적인 결과가 나온다. 꾸준히 도움을 준 자원부터 정리한다.

사진 이론의 탄탄한 뼈대를 잡고 싶다면 Cambridge in Colour(<https://www.cambridgeincolour.com/>)가 여전히 유효하다. 노출 삼각형, 하이 다이내믹 레인지, 색 공간 같은 주제를 수식과 도표로 이해하기 쉽게 풀어준다. 디지털 카메라의 물리적 한계나 렌즈 왜곡처럼 장비와 원리가 만나는 지점도 정확하다.

실전 팁과 트렌드는 PetaPixel(<https://petapixel.com/>)과 DPReview(<https://www.dpreview.com/>)에서 확인한다. PetaPixel은 신제품 소식과 비하인드 생산 과정, 프로 작가 인터뷰를 빠르게 다루고, DPReview는 바디와 렌즈의 세부 테스트, 샘플 갤러리를 꼼꼼히 제공한다. 장비를 바꿔야 할지 망설일 때, 같은 조건에서 비교한 샘플 이미지는 말없이 많은 걸 알려준다.

국내 커뮤니티의 생생한 피드백도 학습에 큰 도움이 된다. SLR클럽(<https://www.slrclub.com/>)은 장비 정보가 빠르고, 실사용자가 공유하는 설정값과 문제 해결 노하우가 축적돼 있다. 다만 중고 거래와 광고 글이 섞여 있으니, 읽을 거리를 골라내는 눈이 필요하다. 촬영 포트폴리오에서 영감을 얻고 싶다면 500px(<https://500px.com/>)과 Flickr(<https://www.flickr.com/>)로 들어가 보자. 사진을 저장하는 용도보다, EXIF 데이터를 함께 열어보며 조리개, 셔터, ISO, 초점거리 같은 설정을 맥락과 함께 익히는 데 더 가치가 있다.

모바일 촬영은 전통적 사진 강좌와 결이 조금 다르다. Halide 블로그(<https://halide.cam/learn>)에는 아이폰 센서 특성, 딥퓨전 처리 방식, RAW 촬영의 실제 이점 같은 내용이 촘촘하다. 보정은 VSCO의 러닝 코너(<https://vSCO.co/learn>)나 Adobe의 Lightroom 튜토리얼(<https://helpx.adobe.com/kr/lightroom-cc/tutorials.html>)을 참고하자. 필터를 덮어씌우는 게 아니라, 화이트밸런스과 커브를 먼저 잡고, 채도와 채널별 HSL을 나중에 만지는 순서를 익히면 결과물이 고르게 좋아진다.

혼자 짚고 혼자 보정하면 종종 같은 실수를 반복한다. 작은 과제를 스스로 던져보면 좋다. 예를 들어 한 주 동안 역광 인물만 찍되, 후보정에서 하이라이트 복구 범위를 최소화해 노이즈를 줄이는 식이다. 이런 과제를 실행할 때, 촬영 전 체크리스트를 간단히 메모 앱에 저장해 두면 현장에서 허둥대지 않는다. 주소모음에 과제별 참고 링크를 묶어두면 반복 학습 루틴이 자연스럽게 만들어진다.

요리, 레시피를 넘어 원리로 접근하기

레시피는 출발점일 뿐이다. 응용을 하려면 재료의 작동 원리를 알아야 한다. 가열 방식, 물과 지방의 상호작용, 전분과 글루텐의 역할 같은 기본이 잡히면 냉장고 사정에 따라 조정이 가능해진다.

방법을 중심으로 가르치는 자료로 Serious Eats(<https://www.serious-eats.com/>)만 한 곳을 찾기 어렵다. 단순히 “10분 만에 파스타” 같은 문구 대신, 왜 소금물을 이 정도 농도로 잡는지, 팬의 크기가 소스 농도에 어떤 영향을 주는지를 데이터로 보여준다. 한국어로 체계적인 레시피를 찾는다면 만개의레시피(<https://www.10000recipe.com/>)가 유용하다. 재료 대체와 난이도 필터가 좋아 초보가 고르기 쉽다.

한식은 망설일 이유가 없다. Maangchi(<https://www.maangchi.com/>)는 영어권 자료이지만 조리 영상과 사진이 매우 직관적이라 언어 장벽이 낮다. 김치, 찌개, 전골처럼 기본기 위주로 따라 해보면 재료 손질과 간 맞추기의 감각이 생긴다. 한편, 식품 안전과 보관법은 식품안전나라(<https://www.foodsafetykorea.go.kr/>) 자료를 확인하는 편이 안전하다. 특히 여름철 비가열 조리식, 저온 조리, 진공 **무료넷플릭스** 포장 보관 등은 작은 실수로도 문제가 생긴다.

대체 재료가 필요할 때는 Cook's Thesaurus로 알려진 FoodSubs(<https://foodsubs.com/>)가 도움이 된다. 슈퍼마켓에 고수나 버터밀크가 없을 때, 향과 질감을 최대한 비슷하게 맞추는 대안을 찾아볼 수 있다. 제철 식재료는 농촌진흥청 농사로(<https://www.nongsaro.go.kr/>)에서 달력과 함께 확인 가능하다. 제철표와 마트 전단지 가격을 같이 보면서 메뉴를 짜면 식비와 신선도를 동시에 잡는다.

실전 팁 하나. 오븐이 없어도 집에서 베이킹을 하고 싶다면, 에어프라이어와 무쇠 팬으로 만들 수 있는 레시피를 모아보자. 굽기 온도와 시간을 미세하게 옮기기 어렵지만, 팬을 예열하고 반죽의 수분을 살짝 줄이면 만족스러운 결과가 나온다. 이런 변형 레시피는 커뮤니티 글에서 발견되는 경우가 많다. 그래서 요리 분야 주소모음에는 반드시 커뮤니티 글의 원문 링크와 날짜를 함께 저장해, 나중에 같은 조합을 재현할 때 혼란을 줄인다.

코딩, 문서와 문제 해결의 균형

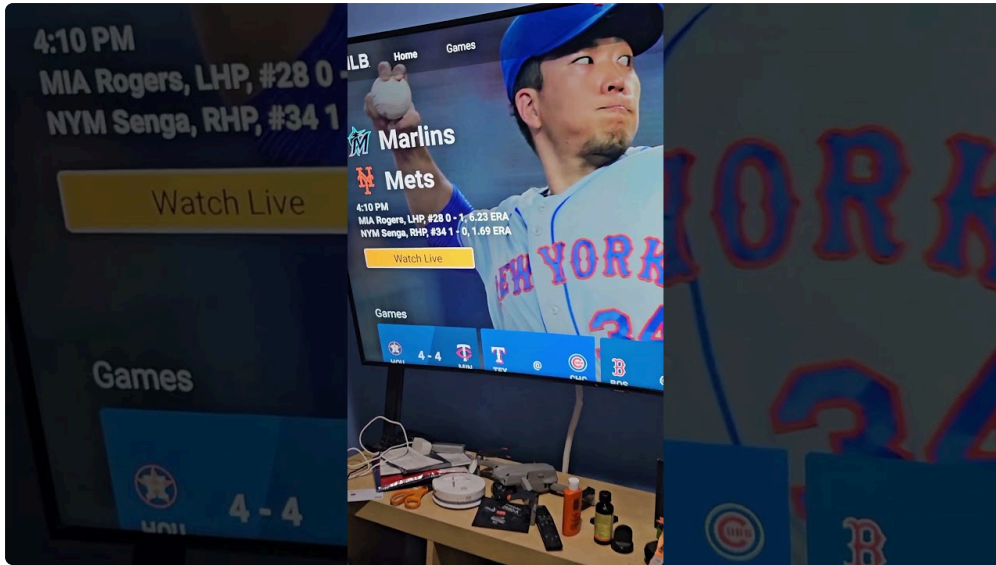
코딩 학습 링크모음은 보통 튜토리얼 위주로 채워진다. 하지만 시간을 오래 아끼는 링크는 공식 문서와 문제 풀이 플랫폼, 그리고 디버깅 흐름을 고도화해 주는 도구다. 언어와 분야에 따라 다르지만, 기본 축은 비슷하다.

웹 개발은 MDN Web Docs(<https://developer.mozilla.org/>)가 첫 번째 기준점이다. HTML, CSS, JavaScript의 정의와 브라우저 호환성 표, 샘플 코드가 잘 정리돼 있다. 이미 만들어진 튜토리얼만 복습하면 실력이 일정 수준에서 멈춘다. 표준 문서와 스펙의 뉘앙스를 파악하는 습관을 들이면, 라이브러리를 바꿔도 흔들리지 않는다. 자바스크립트 엔진 동작 원리를 더 좁혀 보고 싶다면 v8 블로그(<https://v8.dev/blog/>)가 해답을 준다. 최적화 포인트를 이해하면 성능 문제가 생겼을 때 어디를 건드릴지 감이 온다.

초중급 전반을 빠르게 끌어올리려면 freeCodeCamp(<https://www.freecodecamp.org/>)의 커리큘럼이 적합하다. 단계별 과제가 많아 루틴에 녹이기 좋고, 프로젝트 기반 진행이라 포트폴리오로 연결된다. CS50(<https://cs50.harvard.edu/>)은 강의가 탄탄하고, 문제 세트가 잘 설계돼 있다. 다만 자료의 양이 많아 중간에 속도를 잃기 쉽다. 내게 필요한 모듈을 선별해서 듣는 편이 효율적이다.

파이썬과 코틀린, 타입스크립트처럼 공식 문서의 품질이 높은 언어는 원문 문서에 시간을 투자하는 편이 결국 빠르다. 파이썬 공식 문서(<https://docs.python.org/3/>), 코틀린 문서(<https://kotlinlang.org/docs/home.html>), 타입스크립트 핸드북(<https://www.typescriptlang.org/docs/>) 모두 최신 예제와 모범 사례를 잘 담고 있다. 성능과 메모리, 동시성처럼 운영 환경을 건드리는 문제는 언어 문서보다 런타임과 플랫폼 문서에서 더 정확하다.

문제를 직접 풀어보는 루트를 놓치면, 읽는 공부도 금세 잊힌다. 알고리즘과 코딩 테스트 준비에는 백준 온라인 저지(<https://www.acmicpc.net/>), 프로그래머스(<https://programmers.co.kr/>), LeetCode(<https://leetcode.com/>)가 대표적이다. 백준은 문제 수가 방대하고, 온라인 실시간 랭킹으로 동기부여가 된다. 프로그래머스는 기업 실전 유형과 가까워 이력서와 연결되기 쉽다. LeetCode는 영어권 중심이지만 솔루션 토론의 질이 높아, 다양한 접근을 비교하기 좋다.



협업과 버전 관리 없이 성장하기는 어렵다. Git 문서는 Pro Git 한국어판(<https://git-scm.com/book/ko/v2>)이 입문자에게 가장 친절하다. 저장소 관리와 포크, 리베이스, 체리픽 같은 작업을 실제 흐름에 맞춰 설명한다. 깃허브(<https://github.com/>)와 깃랩(<https://gitlab.com/>)은 프로젝트를 시작하고 이슈를 관리하는 장으로 쓰자. 린트와 포맷터, 프리커밋 혹은 정리한 템플릿을 주소모음에 같이 붙여두면 새로운 프로젝트를 셋업할 때, 30분이 5분으로 준다.

실전 조언 하나. 디버깅 자료를 링크모음에 별도로 묶어라. 예를 들어 크롬 DevTools의 Performance 탭 사용법 문서, 파이썬 cProfile과 line_profiler 사용 예시, PostgreSQL EXPLAIN 분석 글, Node.js 이벤트 루프 시각화 자료 등을 한 폴더에 모으는 식이다. 문제가 생겼을 때 검색부터 다시 시작하는 시간을 크게 줄일 수 있다.

음악, 이론과 손의 리듬을 동시에

음악 학습은 귀와 손, 두 축에서 동시에 올라가야 느는 체감이 온다. 이론만 보면 악기에 손이 굳고, 손만 움직이면 곡이 복습을 벗어나지 못한다. 균형 잡힌 링크 주소모음이 중요한 이유다.

음악 이론 입문과 정리는 [musictheory.net](https://www.musictheory.net/)(<https://www.musictheory.net/>)이 가장 간결하다. 인터벌, 코드, 스케일, 진행법을 상호 참조하며 배울 수 있고, 연습 문제도 바로 풀 수 있다. 조금 더 설명이 많은 사이트를 원하면 Teoria(<https://www.teoria.com/>)를 추천한다. 코드 인버전과 카덴스, 보이싱처럼 작곡과 편곡에 바로 쓰이는 내용을 단계적으로 다룬다.

귀 훈련은 귀찮을수록 효과가 크다. ToneGym(<https://www.tonegym.co/>)은 유료 구간이 있지만, 무료 체험으로도 인터벌 감지와 리듬 인식의 약점을 찾는 데 충분하다. 완전 무료 루트로 가려면 Teoria의 Ear Training, [musictheory.net](https://www.musictheory.net/)의 Exercises를 번갈아 써도 된다. 중요한 건 매일 10분이라도 귀를 따로 훈련하는 습관이다. 2주 정도 지나면 코드 성격을 구별하는 능력이 눈에 띄게 오른다.

악기별로는 접근이 다르다. 기타는 JustinGuitar(<https://www.justinguitar.com/>)가 베이스, 리듬, 솔로까지 체계가 탄탄하다. 코드 전환 메트로놈 루틴과 스트러밍 패턴을 단계별로 제공하므로, 하루 20분만으로도 주 3회면 한 달 뒤에 곡이 한 곡 늘어난다. 피아노는 Pianote 블로그(<https://www.pianote.com/blog/>)가 동기부여와 실전 팁이 좋다. 무료 영상만으로도 코드 반주와 보이싱, 아르페지오 연습 루틴을 충분히 잡을 수 있다.

악보와 레퍼토리는 IMSLP(<https://imslp.org/>)가 압도적이다. 클래식 악보의 공용 저장소로, 퍼블릭 도메인 작품을 무료로 내려받을 수 있다. 편곡과 채보는 Muscores 커뮤니티(<https://musescore.org/>)에서 공유되는 파일들이 도움이 된다. 저작권 문제가 얽히기 쉬운 영역이므로, 상업적 사용이나 공개 공연에 앞서 각 작품의 라이선스를 반드시 확인한다.

디지털 음악 제작은 DAW 선택부터 막힌다. Mac을 쓰면 GarageBand가 가장 쉬운 출발점이다. Windows와 Linux에서는 Tracktion Waveform Free(<https://www.tracktion.com/products/waveform-free>)가 종합 제작에 적합하고, Audacity(<https://www.audacityteam.org/>)는 파형 편집에 최적화돼 있다. Reaper(<https://www.reaper.fm/>)는 평가판이 사실상 무기한이지만, 상업 사용이나 장기 사용은 라이선스를 구입하는 편이 윤리적이다. Ableton Live의 공식 튜토리얼(<https://www.ableton.com/en/help/learn-live/>)은 신디사이저와 샘플러의 실전 개념을 잡는 데 큰 도움이 된다.

작은 팁 하나. 메트로놈 대신 드럼 루프를 틀고 연습하면 시간 감각이 빨리 생긴다. 주소모음에 BPM별 루프를 60, 72, 90, 120처럼 네다섯 개만 링크해두고 번갈아 쓰면 루틴이 지루하지 않다.

합법, 안전, 지속 가능성에 대한 짧은 메모

“무료넷플릭스”라는 키워드로 검색하면 수상한 링크가 쏟아진다. 넷플릭스는 공식적으로 장기 무료 시청을 제공하지 않는다. 특별 프로모션이나 통신사 번들, 단기 체험이 있었던 시기가 있지만 지역과 시기에 따라 종료되거나 조건이 바뀐다. 합법적 근거 없이 유료 서비스를 우회하거나 크랙을 쓰는 링크모음은 결국 브라우저 보안 경고와 악성 코드 문제로 이어진다. 스트리밍 서비스를 대체하려는 목적이라면, 아래 같은 합법 경로를 링크모음에 정리하자. 지역별 제공 여부는 수시로 바뀌니, 접속 전 최신 공지를 확인하는 습관이 필요하다.

- 한국영상자료원 KOFA 공식 유튜브(<https://www.youtube.com/@koreanfilm>): 고전 한국영화를 합법적으로 무료 제공한다. 해설과 복원 정보까지 제공돼 영화 문해력을 키우기 좋다.
- EBS 온에듀(<https://www.ebs.co.kr/>)와 EBSi(<https://www.ebsi.co.kr/>): 다큐와 교육 콘텐츠의 품질이 높고, 많은 강좌가 무료로 열려 있다.
- K-MOOC(<https://www.kmooc.kr/>): 대학 강의를 무료로 수강할 수 있다. 인문, 과학, 예체능까지 폭이 넓다.
- Internet Archive(<https://archive.org/>): 퍼블릭 도메인 영화, 라디오, 문헌이 방대하다. 저작권 표시를 확인하고 이용 범위를 지키면 된다.
- 공공도서관 전자자료: 지역 도서관 회원이면 영상, 잡지, 오디오북 구독을 제공하는 경우가 많다. 도서관별 전자자료 포털 링크를 주소모음 상단에 고정해두면 유용하다.

무료로 노린다고 품질이 낮아질 필요는 없다. 중요한 건 출처의 합법성과 지속성이다.

주소모음이 학습 속도를 바꾸는 순간

링크를 저장하는 행위 자체가 목적이 되면 곤란하다. 학습 흐름에 맞춰 링크가 자동으로 불러와지는 환경을 만들면, 클릭 수가 줄고 몰입이 늘어난다. 몇 가지 방법을 제안한다.

먼저, 주제별이 아니라 루틴별로 폴더를 나누는 방식이 효율적이다. 예를 들어 사진 폴더 하위에 이론, 리뷰, 보정, 커뮤니티로 나누지 말고, 월요일 과제 루틴, 수요일 보정 루틴, 주말 촬영 루틴처럼 시간표 기준으로 그룹을 짠다. 각 루틴 폴더 안에 그날 필요한 링크만 담으면 시작할 때 망설임이 사라진다. 자투리 시간용 폴더를 따로 만들어 5분 안에 읽을 수 있는 짧은 글을 모아두면, 대기 시간에 생산성이 오른다.

둘째, 도구를 고른다. 브라우저 기본 북마크만으로도 충분하지만, 태그와 검색, 중복 관리가 필요한 시점이 온다. Raindrop.io(<https://raindrop.io/>)는 컬렉션과 태그, 스크린샷 저장이 편하고, Notion(<https://www.notion.so/>)은 링크를 데이터베이스로 관리해, 카테고리나 상태, 리마인더를 붙이기 좋다. 코드 스니펫과 함께 저장하려면 Obsidian(<https://obsidian.md/>)이 강력하다. 마크다운으로 오프라인 백업이 가능해, 서비스 중단 리스크를 줄여준다.

다. 읽기 전용 큐레이션이 필요하다면 Pocket(<https://getpocket.com/>)을 써라. 나중에 읽기 보관함을 정기적으로 비우는 루틴만 추가하면 효율이 유지된다.

셋째, 리마인더를 건다. 주소모음은 업데이트가 끊기면 가치가 빠르게 줄어든다. 달력에 분기별 점검 일정을 만들어, 1년 이상 업데이트 없는 페이지는 과감하게 보관함으로 옮긴다. 반대로 자주 쓰는 링크는 설명을 1줄씩 덧붙여둔다. “이 글에서 얻을 수 있는 것”을 정확히 적어두면, 몇 개월 뒤에도 혼선이 없다.

마지막으로, 공유를 시도한다. 링크모음은 혼자 쓸 때보다 팀이나 친구들과 문맥을 공유할 때 품질이 빨리 오른다. 맥락을 설명하기 위해 메모를 보완하게 되고, 불필요한 링크는 자연스럽게 정리된다. 협업이 가능한 북마크 도구를 쓰면 권한과 히스토리 관리가 쉬워진다.

사진을 위한 대표 링크, 이렇게 묶어본다

사진 분야 주소모음을 처음 만드는 사람이라면, 네 가지 축을 먼저 세워보자. 빛과 노출을 다루는 이론, 장비와 리뷰, 보정과 색, 커뮤니티와 포트폴리오다. 각 축에서 대표 링크를 두세 개만 고르고, 주제별로 반복 과제를 만든다. 예를 들면 다음과 같다.

Cambridge in Colour의 노출 가이드를 3회 정독하고, 실내, 그늘, 직사광 각각에서 동일 피사체를 노출 변수로 12컷씩 촬영한다. DPReview 샘플 갤러리에서 목표 바디의 고감도 샘플을 확인하고, 노이즈 억제와 디테일 손실의 균형을 눈에 익힌다. 보정은 Adobe Lightroom 튜토리얼에서 지역 대비와 컬러 그레이딩 파트를 선별해 실습한다. 마지막으로 500px의 유사 장르 상위 사진 20장을 분석해, 시선 유도과 색의 비율을 기록한다. 이 모든 링크를 월간 과제 폴더에 묶으면, 4주 루틴이 자연스럽게 돌아간다.

요리의 링크모음, 냉장고가 가이드가 되게

냉장고에 자주 있는 재료를 기준으로 링크를 묶으면 실전성이 급격히 올라간다. 예를 들어 달걀, 두부, 닭가슴살, 토마토 통조림이 항상 있다면, 이 네 재료로 가능한 조리법을 Serious Eats와 만개의레시피에서 3개씩만 골라 묶는다. 각 레시피에 “가열 방식 - 수분 처리 - 마무리 간”을 적고, 재료 대체 가능성은 FoodSubs 링크와 함께 적는다. 식품 안전나라의 보관 기준 링크를 같은 카드에 붙여두면, 남은 재료를 며칠 안에 처리해야 하는지도 한눈에 보인다. 냉장고가 곧 요리 교과서 목차가 된다.

코딩 학습 루틴, 디버깅 링크가 핵심

문제를 풀다 막히는 지점에서 링크가 빛난다. 그래서 코딩 주소모음의 첫 화면은 튜토리얼이 아니라 디버깅과 검색 전략이어야 한다. 예를 들어 자바스크립트 비동기 이슈가 생기면 MDN의 Promise와 async/await 문서, 크롬 DevTools의 Async 스택 트레이싱 문서, 그리고 V8 블로그의 마이크로태스크 큐 설명 글로 바로 들어간다. 파이썬에서 속도가 문제면 공식 문서의 프로파일링 챕터, PyPy와 CPython의 차이, C 확장 모듈 가이드 링크를 순서대로 검토한다. 이런 “문제별 진입” 링크모음은 고급자가 될수록 가치가 올라간다.

음악 루틴, 귀의 피드백을 링크로 고정

연습은 늘 좌절과 함께 온다. 그 좌절 지점을 링크가 덮어줘야 한다. 예를 들어 코드 톤을 정확히 구별하지 못할 때는 musictheory.net의 코드 퀴즈로 들어가고, 특정 BPM에서 리듬이 흔들릴 때는 90과 120 BPM 드럼 루프를 바로 재생한다. 보이싱이 단조로운가 싶으면 Teoria의 보이싱 튜토리얼과 피아노 보이싱 예제를 연결한다. 이어서 KOFA 유튜브 채널에서 사운드트랙 분석 영상을 보는 루틴을 붙이면, 이론과 감상의 균형이 맞춰진다.

북마크 정리, 꾸준함을 만드는 작은 습관

링크가 쌓이면 관리가 어렵다. 그럴수록 기본기를 지키는 체크리스트가 필요하다. 아래 다섯 가지만 지켜도 주소모음이 과체중이 되는 속도가 느려진다.

1. 새 링크를 저장할 때 1줄 요약과 태그 2개만 붙인다
2. 월 1회, 업데이트가 끊긴 링크를 보관함으로 이동한다
3. 루틴별 폴더 첫 화면은 최대 7개 링크로 제한한다
4. 읽기 전용 큐와 실습용 큐를 구분한다
5. 공유 폴더에는 맥락 설명과 대체 링크를 함께 둔다

체크리스트를 적용하면 링크모음이 늘어나도 체감 복잡도는 거의 증가하지 않는다. 결국 중요한 건, 필요한 순간에 필요한 링크만 눈앞에 나타나게 하는 설계다.

마지막으로, 주소모음은 취향의 지도다

사진, 요리, 코딩, 음악 네 분야를 두루 다뤘지만, 누구에게나 완벽히 맞는 지도가 하나로 정해져 있지는 않다. 어떤 사람에게는 공식 문서가 제일 잘 맞고, 다른 사람에게는 커뮤니티 토론이 직관을 깨운다. 그 차이를 존중하면서도, 합법성과 업데이트, 재현 가능한 실습이라는 세 가지 원칙만은 지키자. 그러면 링크 자체가 학습을 이끈다.

오늘 당장 할 수 있는 작은 시작을 제안한다. 각 분야에서 한 줄로 소개할 수 있을 만큼 신뢰하는 링크 3개씩만 골라, 루틴 폴더에 넣어 두자. 다음 주에는 그 12개의 링크로만 학습 루틴을 돌려본다. 끝나고 나면 무엇이 남고 무엇이 불필요했는지 감이 온다. 그 감을 따라 주소모음을 재배열하면, 취미 생활의 속도와 깊이가 한 번에 달라진다.