Cognitive Capital

My simple system to turn information into actionable knowledge and thrive in the information age

Sylvain Kerkour

Continuous Learning

My simple system to turn information into actionable knowledge and thrive in the information age

Sylvain Kerkour

v2023.1

Contents

+++ title = “Continuous Learning - Introduction” date = 2021-07-26T6:00:00Z type = “page” url = “/continuous-learning/introduction” access = “paid_members” +++

Chapter 1

Introduction

1.1 The information age

Internet, computers and smartphone made the amount of information available to the average Human explode, to the point that our epoch is called the Information age: an economy mainly based on information technologies.

We now have access to so much information that it's even problematic: we no longer have the time to process all this information which leads us to be overloaded and even burnout.

This is why you need a system to turn all this information into something actionable, something that can help you achieve your goals.

1.2 Data vs. Information vs. Knowledge

Before continuing, you first need to understand what knowledge is, and what it's not.

Data are the raw facts that we gather from the surrounding world: `42°C in Paris at 2021-07-26T6:`
.

Information is data refined for consumption by Humans: `42°C in Paris on Tuesday 26 of July`

Finally, **knowledge** is what you use to make a decision, most of the time by making connections between the pieces of information: `Paris is hot in July. I should think about wearing light c`

In one sentence: **Information is refined data whereas knowledge is useful information**.

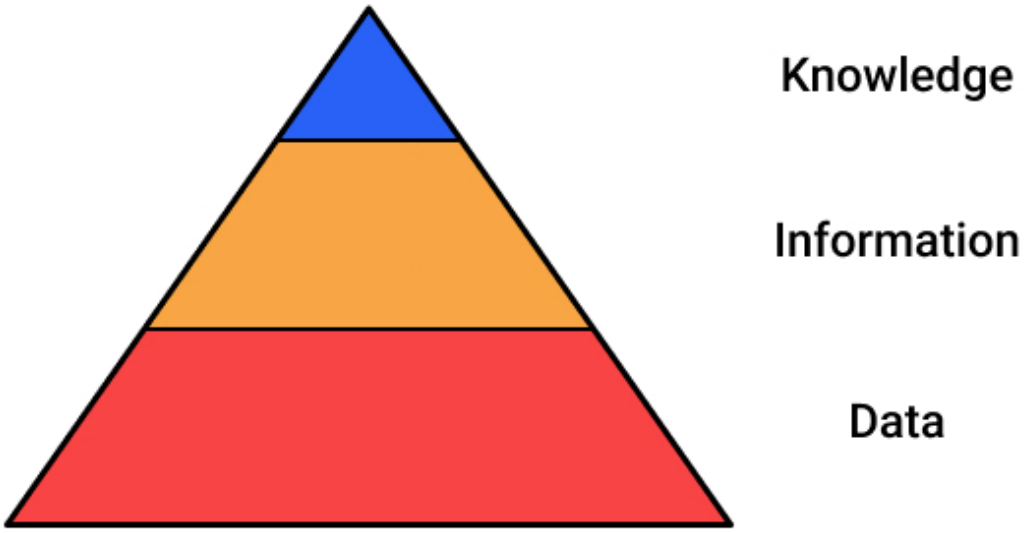

Figure 1.1: Data vs Information vs Knowledge

1.3 The problem with how we approach education

Most societies currently approach education as a stage in our lives, approximately between 3 and 25 years.

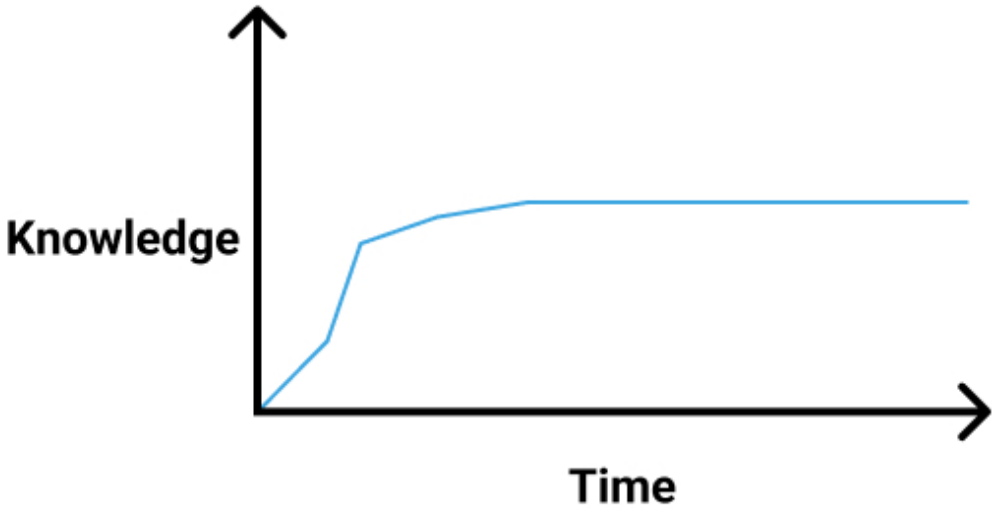

Figure 1.2: Knowledge over time

And then what? The world stops going forward, and you no longer need to learn anything?

Of course not, if you want to stay relevant, whether it be in your job, as an entrepreneur, or simply as a citizen, you need to continue to learn all your life.

And this is even before talking about the superior education in some countries where if you

don't come from a very rich family, you have to get a massive student debt that will basically turn you into a modern slave for the best years of your life.

So yeah, while schools are fighting and cheating to improve their ranking and spending huge amounts of money and energy on bureaucratic tasks we

Just to be clear for the rest of the book: even if I attended some prestigious schools such as 42, I barely graduated from high school and have not bothered to get any other piece of paper stating my supposed skills. In other words, I have no degree and still managed to be a not-so-bad programmer.

Here is how.

1.3.1 Ahead of Time vs. Just in Time

The first distinction to make is Ahead of Time vs. Just in Time learning.

Engineers and scientific people love Ahead of Time learning. Basically, it's when you learn things that you don't need to know now, mostly because it's seem interesting. You learn things, and then you find useful applications for this knowledge.

On the other hand, Just in Time learning is when you spend time learning new things only when you need it, or you need to complete a project. Just in Time learning is goal-driven: You first define your goals, and then you acquire the knowledge to achieve them.

I do believe that Just in Time is better. Indeed, there is just too many things to learn, and you don't have an infinite amount of time, so it's better to start from your goals and then learn what you need to achieve your goals. Ahead of Time learning can lead to **severe procrastination**, which in turn can lead to unhappiness.

But, as we will see, you should not limit your information and knowledge acquisition to dedicated learning sessions (e.g. when reading a book or following a course). It should be something that happens all the time: - Taking notes when reading a book - Saving the important ideas of the articles that you read online - Saving the important things that someone may say in a chat

1.4 Neurplasticity

Your brain is composed of billions of neurons and trillions of synapses, connecting the neurons together and forming very complex networks that scientists are trying to unravel the mysteries and engineers are mimicking to create Artificial Intelligence software, but you may already know that.

However, do you know how to use this phenomenal organ to optimize your learning process?

Basically, when you are learning something or practicing some activity, new connections between your neurons are created. This phenomenon is called **neuroplasticity**: the ability of your brain to create, strengthen or dismantle connections between your neurons.

That's why when you learn something for the first time, you tend to forget it quickly: your neural connections are not strong enough.

1.5 The forgetting curve

This phenomenon was studied In the late 19th century by the German psychologist Hermann Ebbinghaus carried out many experiments to better understand learning and forgetting.

His most famous finding certainly was the **forgetting curve**: a curve describing the loss of information that one has learned.

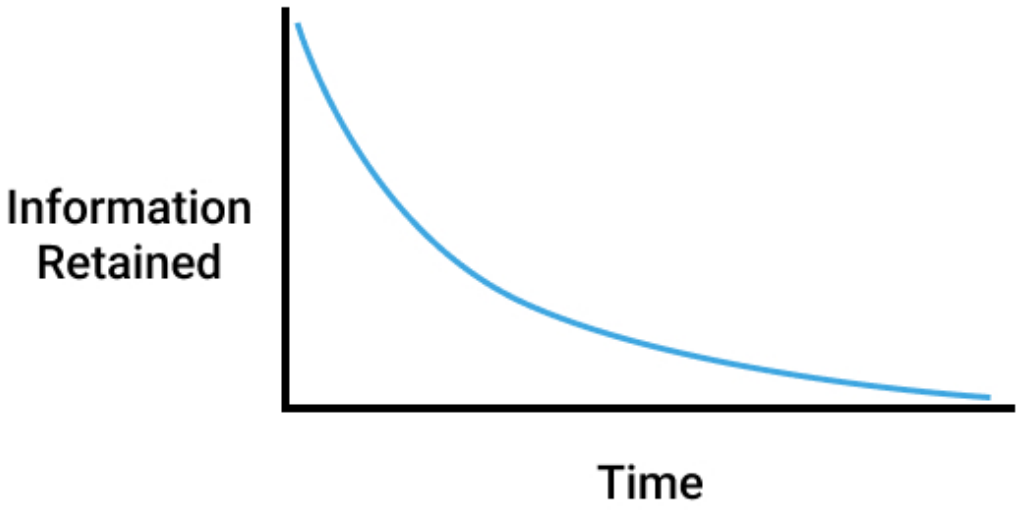

Figure 1.3: Ebbinghaus' forgetting curve

His conclusion was that to maximize retention you need to periodically refresh your knowledge.

Active recall of information not only refreshes the forgetting curve, but makes it decay more slowly afterward.

This, as you may have guessed, means that your neural network encoding the information becomes stronger and stronger.

This is where **spaced repetition** comes into play: when you want to learn something new, you need to periodically refresh your knowledge to maximize retention.

1.6 Spaced repetition in practice

Please, tell me: **How to practice spaced repetition**?

I personally use Quizlet, but any flashcard application will do it.

The idea is that every week, you create a new deck with what you want to learn and read the deck at a fixed interval.

For this to be as effective and easy as possible, you have to make it a habit and do it every day. Maybe before lunch, maybe before sleeping, just do it every single day. It doesn't need to be for 1 hour, 15-20 minutes is enough.

1.7 Thriving thanks to Diminishing Returns

Efficient knowledge assimilation (the act of turning information into something valuable and actionable) is the key to thrive in the information age.

Whether you are a software engineer, a lawyer, a manager, or an entrepreneur, your value lies in how efficiently you can learn new things to stay relevant in your field in an ever-accelerating world.

But there are two problems.

The first one is that today, there is simply too much information available and it's hard to isolate the signal from the noise.

The second problem is that knowledge (*"white collar"*) jobs are evolving faster than the capacity of most people to learn new things. For example, today, you are a software engineer developing backend applications, and you may want to move into Machine Learning with all the recent and exciting advancements, such as the Open Source / Model Stable Diffusion.

But I have good news for you.

Learning follows the law of diminishing returns: you need to spend exponentially more effort to squeeze the last bits of performance when learning a specific topic.

And this is where the hackers start to see how to use this tendency to their own advantage: most people are afraid of learning a new topic, because they think they will never be able to be as knowledgeable as the most famous experts.

By spending two weeks studying a topic really hard, you can assimilate the most important principles of this topic and become better than maybe 90% of the people on this planet (I admit that I took this number out of my hat, but from experience, I think it's pretty accurate).

Of course, you won't become a Ph.D. in the field in two weeks, but anyway, being too much an expert has many drawbacks, such as overthinking and analysis paralysis.

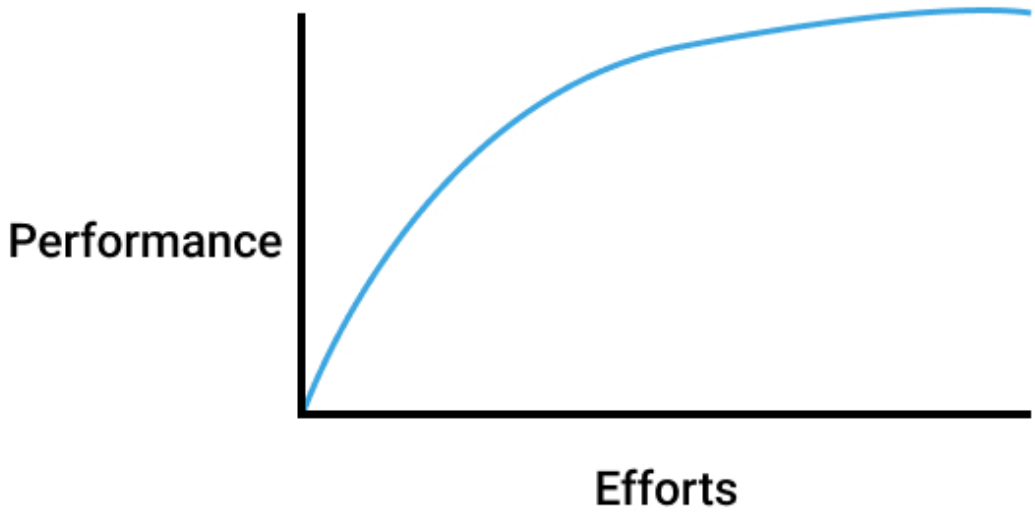

Figure 1.4: Diminishing returns

So, do you want to change your job or start a new startup in a field where you have no expertise?

Study hard the big principles for 2 weeks. Then do 1 applied project per week for 3 weeks to learn the current practices and trends. That's all.

In a little bit more than 1 month, you will be more effective in this field than most of your peers/competitors, even if they studied the topic in university 15 years ago… You don't need to be an expert to **start doing**.

1.8 The system

For that, I've developed a 3-stages system: **Acquisition** (processing), **Organization** (storing), **Retrieval** (search and recalling)

```
Acquisition -> Organization -> Retrieval
```

It's what I call the **knowledge assimilation pipeline**: a system to learn complex topics from mixed media sources.

Put another way, a method to reliably **turn information into knowledge**.

Let's start with the second and third stages of the system: Building and using a knowledge base.

+++ title = "Continuous Learning - The Knowledge Base" date = 2021-07-26T6:00:00Z type = "page" url = "/continuous-learning/knowledge-base" access = "paid_members" +++

Chapter 2

The Knowledge Base

A knowledge base should be seen as your personal Google: the place where you look for answers when you have questions about something you have already read, heard or seen.

The most important thing to understand about your knowledge base is that it is a central repository for all the things that you learned in the past.

Central means that you **SHOULD NOT** have to look to another place for a specific piece of information. In an old chat, in your bookmarks, in your email inbox or in the highlight of your kindle device for example.

Websites can go offline for a lot of reasons (company going bankrupt, technical issues, hacks…) that's why bookmarks are not a good way to preserve knowledge. Instead, you should copy the parts of interest into your knowledge base, and then add the link to the webpage at the top or the bottom of the note.

For example, I have in my knowledge base a folder called `wallet` where I put all my transport and event tickets. I don't need to search in my `Downloads` folder or my inbox when I'm about to board a train.

2.1 Organizing your knowledge base

You can't organize ideas only in a hierarchical way, using folders and subfolders.

For example, where belongs a note about using python for data science: In the `programming/python` folder, or in the `datascience` folder?

Instead, the trick is to group related notes in the same folder (ex: `courses` to group notes about online courses) and then add **tags** to your notes.

Tags allow notes to belong to multiple categories at the same time.

For example let's take the following quote by Edsger W. Dijkstra.

```
Simplicity is a great virtue but it requires hard work to achieve it and education to apprec
```

You can put the note in the `quote` folder and add the following tags: `#author/dijkstra #quote #minimalis`

2.2 Choosing the right tool

Now that you understand the requirements for a good knowledge base, let's do a quick review of the best apps available today.

Most people think that good knowledge base software requires both computer/tablet and smartphone apps, but this is wrong. As we will see in the next chapter, you don't necessarily need to be able to append notes to your knowledge base from a phone. On the other hand, searching is preferable.

As we have seen, a pre-requisite for our knowledge base software is to support tags. Another pre-requisite is that it should works even without internet connectivity: You should be able to access your entire knowledge base, and update it even if there is an internet outage, or if the service hosting it (if any) is down.

2.2.1 Evernote

Evernote has fallen in disgrace. Don't use it. Slow, buggy, taking data hostage… I've read many reviews of long-time users who now regret being blocked with it.

2.2.2 Notion

Its biggest strenght is an ecosystem of tools that can be connected to your Notion workspace such as a blogging engine, a form collector, and many others.

While notion is all the rage these days, I believe it will follow the same fate as Evernote, but in a different way.

Notion is now sold as a knowledge management tool for **organizations**. A very competitive space, and thus, will certainly end up bloated and not a good fit for a personal knowledge management tool.

And this is before talking about the pricing strategy of this kind of B2B startups, which is optimized for companies, not individuals.

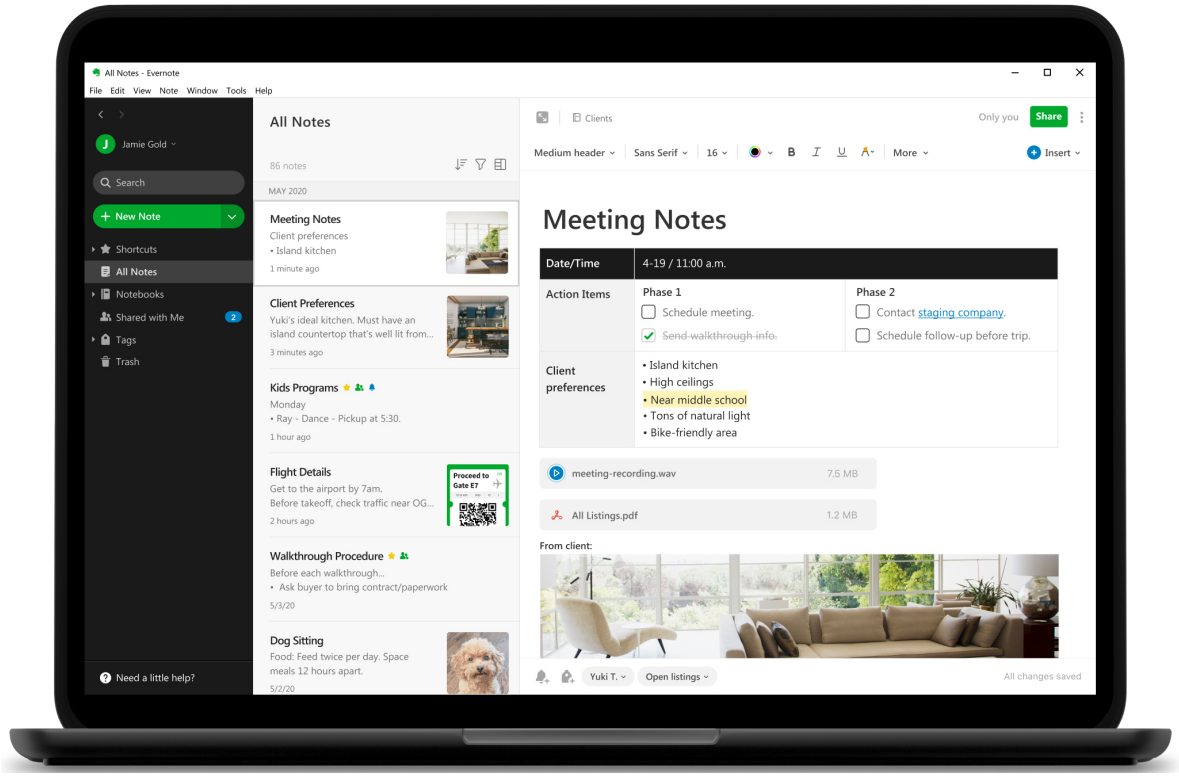

Figure 2.1: Evernote

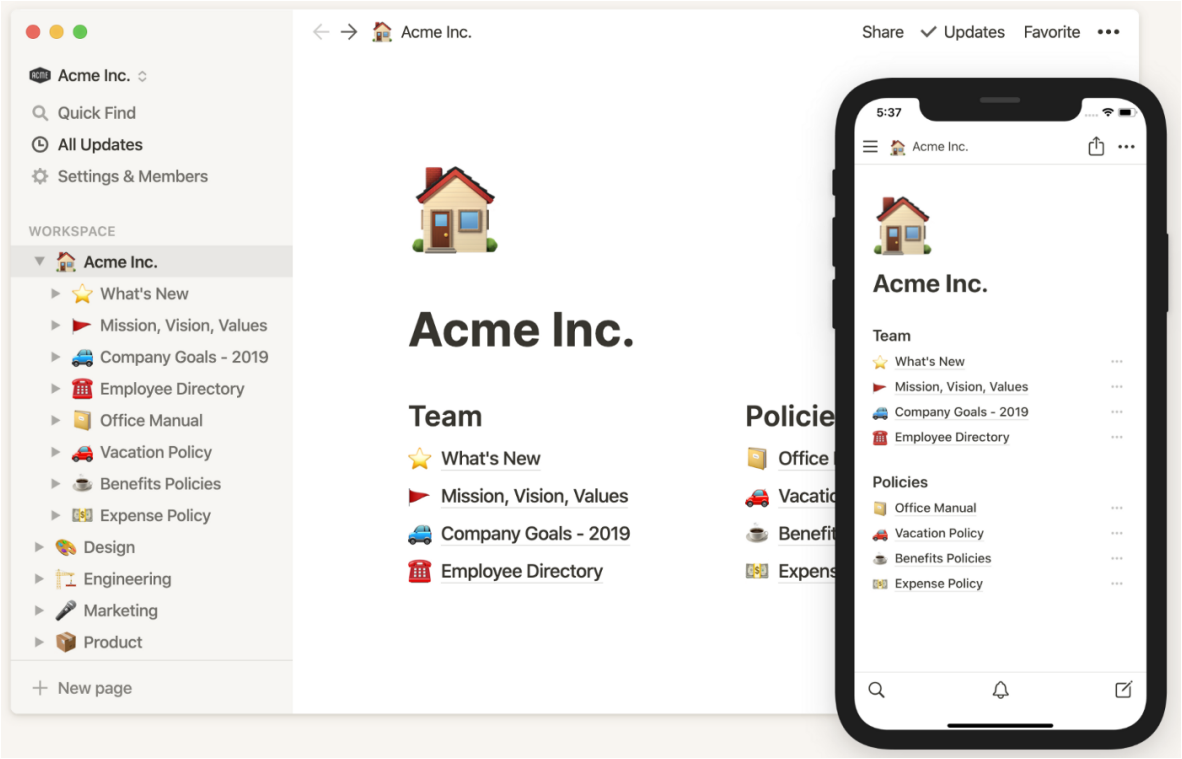

Figure 2.2: Notion

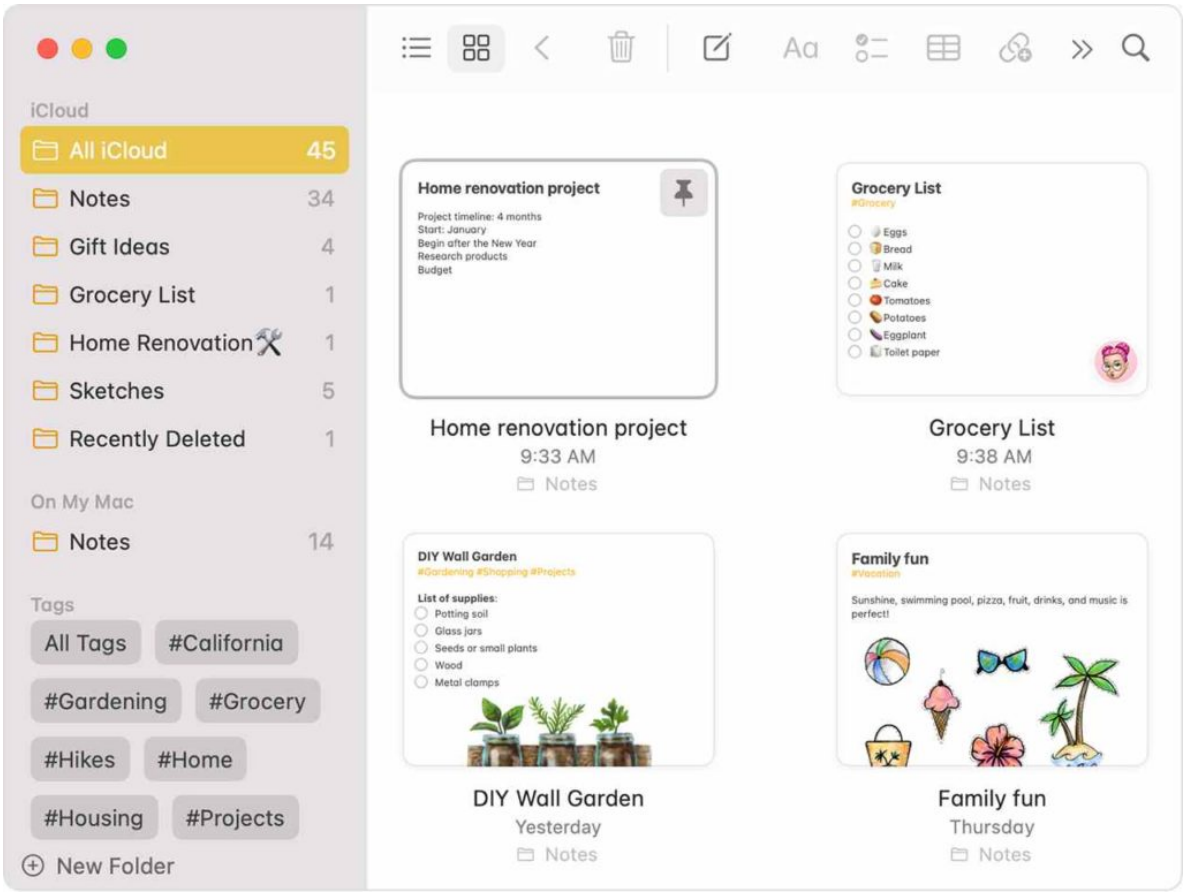

Figure 2.3: Apple Notes

2.2.3 Apple notes

While Apple Notes had too few features in the past to make it a good knowledge base, it became a serious contender with the recent support for Tags.

The 2 principal drawbacks of Apple Notes are:

1. The lack of Markdown support, which makes it not interoperable with other apps and software.
2. When you consolidate all your digital life under a single account, if that company goes rogue, or if your account is banned due to the mistake of an automated system, even for a reason completely unrelated to your knowledge base, you will lose everything.

2.2.4 Bear app

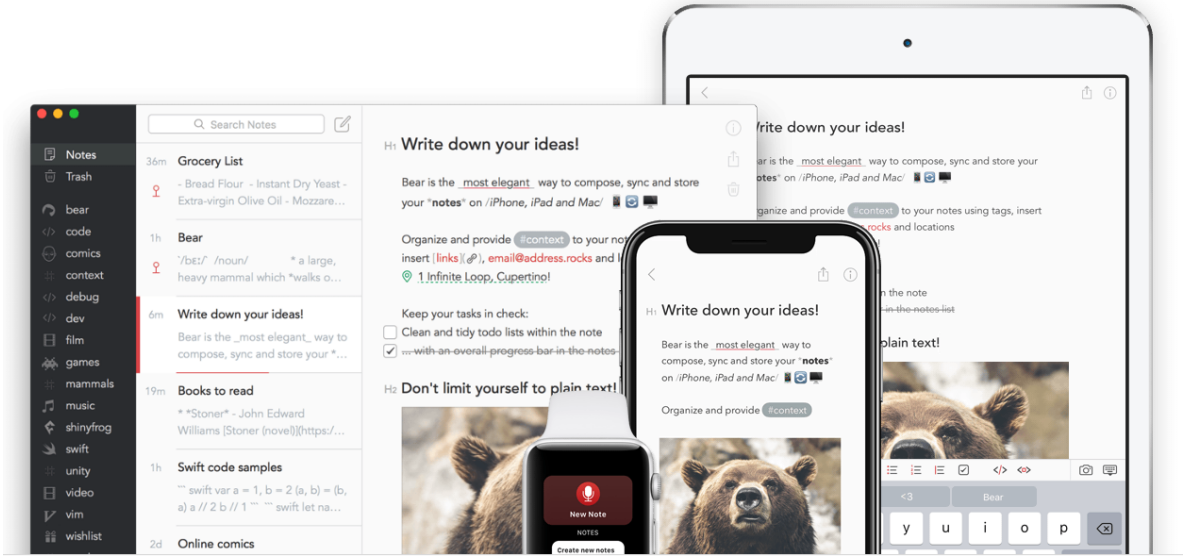

Figure 2.4: Bear

Bear is a note apps for the Apple devices that support Markdown and tags. The app is beautiful and optimized for productivity.

It also has advanced features such as the ability to export to HTML, PDF, DOCX, MD, JPG, and encrypted notes.

The biggest problem with Bear is that it stores files on iCloud and not your filesystem. Thus your iCloud account remains a single point of failure.

2.2.5 Markdown + Git

This is for the reasons mentioned above that I prefer to use Markdown + Git for my knowledge base. These technologies are highly interoperable, and can be used across many different

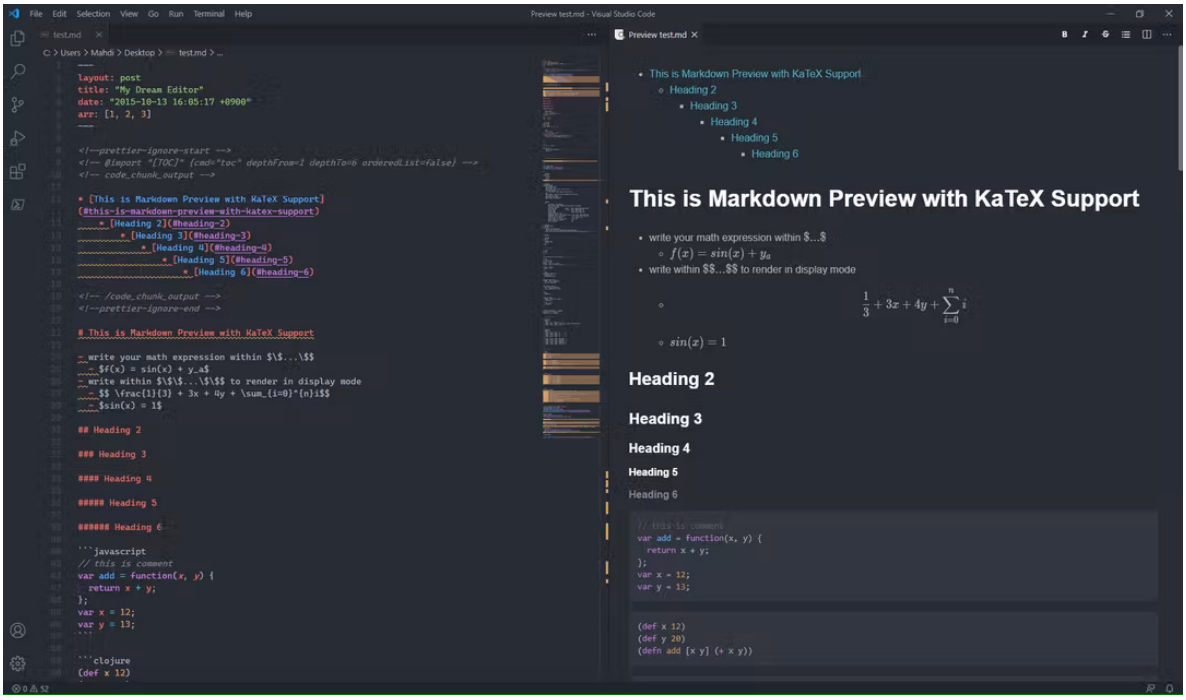

Figure 2.5: VS Code

providers, applications, and can even be hosted on your own server.

On a computer, you only need VS Code. You can have a very efficient workflow with the right keyboard shortcuts to commit and push to Git.

On iPhones and iPads, you can use the combo: Working Copy to sync your Git repository and iA Writer to search and edit your knowledge base, which supports tags.

2.3 Spaced repetition in practice

Once the information that you deemed as important is stored in your knowledge base, it's time to feed the essential parts to your brain using spaced repetition.

For that, I use a flashcard software called Quizlet.

You create card sets with the most important points that you want to remember (the non-important bits can be searched when needed) and study them every 3 to 6 days.

That's all, really.

2.3.1 Practice

The other way to reinforce your neural networks toward a specific piece of information is by practicing.

Fail, adjust, learn, repeat.

2.4 Summary

- Your knowledge should be centralized in your knowledge base. Not scattered across many applications.
- If you know how to use Git, use Markdown and Git for your knowledge base.
- Else, use Apple Notes or Bear if you are into the Apple ecosystem, or Notion otherwise.
- Use a flashcards or Anki cards software to practice spaced repetition.

+++ title = “Continuous Learning - Knowledge Acquisition” date = 2021-07-26T6:00:00Z type = “page” url = “/continuous-learning/acquisition” access = “paid_members” +++

Chapter 3

Knowledge Acquisition

Now that we have a centralized repository to store, search and learn from our knowledge, it's time to fill it.

Always remember that it's a selective process, and information will be lost, but that's on purpose: in order to turn information into knowledge, you need to trim the irrelevant "fat".

3.1 How to learn anything

My strategy to learn any thing is actually simple.

First I focus on the big and immutable ideas of the field, the "axioms", the things that didn't change for a long time (long time being relative to your field) and thus, according to the Lindy effect are going to stay relevant for a long time in the future.

Then, I apply what I studied with small to medium projects to learn the current practices and methodologies.

For exemple, if I would have to learn programming again today, I would start with computers artchitecture, operating systems and Linux, algorithms and design patterns. Then, I would practice by building small command line program to solve my real-world problems, such as a calculator, an app to display the schedule of my bus lines…

3.2 Filling your knowledge base

Now you know what to learn, it's time to see how to fill you knowledge base in practice.

The simplicity and effectiveness of this system rely on a 2-step process to fill your knowledge base.

First, you save what you want to remember. I say save, because it should not be noted into your knowledge base on the spot.

That's why you don't need a knowledge base application that supports mobile. Instead, I send notes and audios to myself in a chat application. Most chat applications nowadays support a "Send to myself" conversation to save notes: Signal, Telegram, Element...

The second step consists in re-reading or re-listening to your notes in your chat application, potentially augmenting them with additional research and transfering them in your knowledge base. Then, you can delete the messages in the chat app.

This 2-step process allows you not only to find potential mistakes but also to initiate spaced repetitions, as we saw in the introduction.

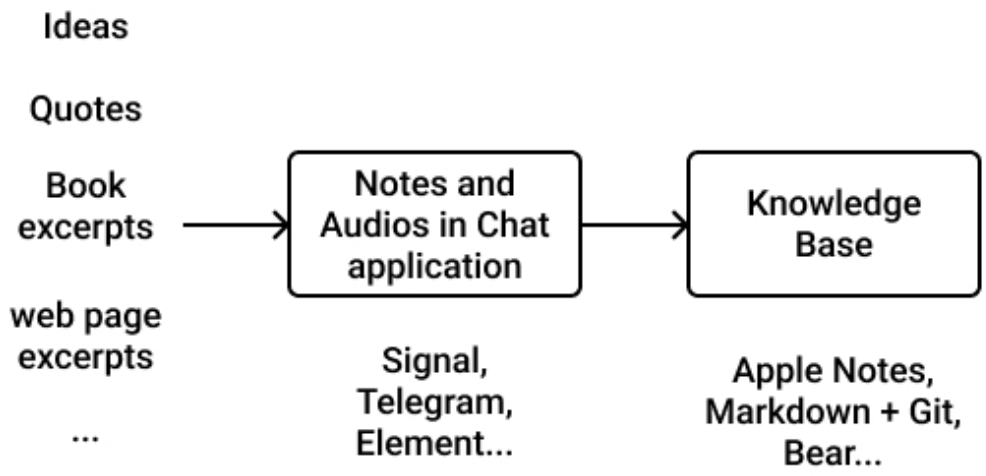

Figure 3.1: The Knowledge Acquisition Pipeline

Why a chat application? Because it acts as a Last-in, First-out queue, kind of like a todo list. Once your process a message and delete it, it's done, gone. Unlike bookmarks that will accumulate over time and clutter your digital space.

Also, you don't want to have to think about how to organize your knowledge base when having a new idea, like which tags to add. Focus on your idea, write it down, and organize your knowledge base later when poping the note from your Last-in, First-out queue.

Now, maybe you are like me, and you don't have a phone. You can replace the chat application with a small notebook and remove the pages once you transfer the notes to your knowledge base. Another alternative is to use a dictaphone.

Either way, the goal here is to be able to take notes on the spot with the less frinction as possible, and to process it later.

3.3 The Sources of Knowledge

Today, many medium exist to learn new things directly from experts: * Podcasts * Videos * Blog posts * Books * …

But not all sources of knowledge are equal.

I believe that text (with illustrations) was, is, and will remain the best tool at our disposal to transmit and store most knowledge.

There are many factors to explain this.

Text is easier to edit than audio and video. Thus if a mistake is found, it can be fixed quickly.

Text requires your full attention, while you can do something else when listening to a podcast.

Our brain is wired to recognize Human faces and be empathic. Thus, videos are good at transmitting and generating emotions. On the other hand, due to its raw form, text tends to activate only the rational thinking part of our brain.

Text mediums such as websites, books, and ebook readers can be optimized to provide a good focus for the reader. On the other hand, a video always has a background. Furthermore, video platforms tend to optimize for "engagement" and will use many tricks to grab your attention.

Text can be searched (e.g. `/ Ctrl + f` to search this page) while it's way harder to search for a precise passage in a video or podcast.

Text parts can be saved way more easily than parts of an audio or video.

Text has the best support for tools for people with disabilities.

The structure of a text is easy to appreciate, while it's most of the time way harder to rembmer how an educationnal video or podcast was structured.

Finally, text is lightweight. When you compare the knowledge / weight ratio of text to video or audio, text is the clear winner. As an example, the entire content of this website (blog posts + books + media) weighs a total of 35 MBytes on my disk, which means that it can be saved for very cheap and can be transfered in a few seconds.

My knowledge base (~4 years of notes, schemas, and screenshots) weighs ~101 MBytes.

On the other hands, videos containing the same information would weight hundreds of Giga-Bytes.

Text is and will always be the king when it comes to sharing and acquiring information and knowledge.

3.3.1 Paid vs. Free

Whether it be sharing expert knowledge or curating existing knowledge, creating content takes a lot of time. And authors and creators are like you: they need to pay the rent, the food, and the extras.

Furthermore, a lot (if not most) of the free content available on the internet today is produced to build an audience and ultimately make money out of it, not to help the readers / viewers achieve their goals. This is the role of the paid content.

Most content creators tell the **why** in their free content, and the **how** in their paid content.

This is why you should invest in paid content if you are serious about learning something. It will save you a lot of time, and thus money and frustration.

3.3.2 Books

I think I don't need to say anything about books, as it's the (approximately) universal way for Humans to share knowledge since thousands of years.

I just want to note that due to a lot of factors (e.g. customers perceiving a high number of pages as a proxy for good quality), a lot of books are, in my opinion, not curated enough: they contain too much noise to fill the pages and not enough signal.

That being said, book are not only a medium to share knowledge, but also to transmit emotions and inspire the readers.

Where to find books?

Goodreads of course, but also on social aggreagtor websites, where people regularly ask for recommendations. On Hacker News if you like tech and entrepreneurship, or reddit for example.

Finally, there are some specialised websites such as: Software Engineering Books.

3.4 Feedback

We are Humans, so we all make mistakes. Thus, it's a matter of time before you learn things that are wrong.

This is why you need **feedback** about what you learn. This is the purpose of exams and teachers in the traditional education system, but you may not necessarily be able to afford a private teacher all your life.

Thus, the best way to get feedback is to publish what you learn, for example, through blog posts.

+++ title = “Continuous Learning - Conclusion” date = 2021-07-26T6:00:00Z type = “page” url = “/continuous-learning/conclusion” access = “paid_members” +++

3.5 The future of knowledge bases

With the emergence of advanced and Open Source Machine Learning models such as the recent Stable Diffusion (open sourced last week as I’m writing this), we can expect that in the near future, we will be able to train an AI on our own knowledge base that will be able to answer advanced questions instead of having to rely on basic search and hashtags.

The biggest advantage of these AIs versus a traditional Google is that they will use information and data that you already vetted as it was saved into your knowledge base instead of the SEO-filled web.

3.6 A few more tips

One of the biggest barriers to knowledge assimilation is distractions.

While it’s easy to limit visual distractions by practicing isolation, it’s, in my experience, harder to limit auditive distractions.

My solution is to use “white-noise” sounds during my learning sessions. Those are songs that are repetitive, so your brain no longer pays attention after a few minutes and that cover the sounds from outside, from the road or your co-workers, for example.

3.6.1 Music

- https://www.youtube.com/watch?v=n44xjN3GjME
- https://www.youtube.com/watch?v=20ciZWpw5Z4
- https://www.lofi.cafe (or https://lofi.limo/, https://musicforprogramming.net/latest/, https://lofi.co)

3.7 Your turn

If not already done, download a knowledge base software now!

Then, start filling it with content from all your open tabs or bookmarks, until you no longer have open tabs, and that all your remaining bookmarks are day-to-day applications such as Banking, Netflix, or whatever.