Talent Acceleration Program

Workbook

tap.kiitos.tech

Week 0: Preparation

It's not the will to win that matters – everyone has that. It's the will to prepare to win that matters.

Welcome to the *Preparation Module* for the **Talent Acceleration Program (TAP)**. In this module you'll learn all about how to set yourself up for success so that you can get the most out of the program!

It is divided into several sections:

1. Learning How To Learn
2. Web Development Fundamentals
3. TAP Quickstart

Good luck!

The TAP Team

Part 1: Learning How to Learn

In this section you'll be exposed to various ideas that will change your view of education and how you learn. Education itself is an ever-evolving science and it's good for you to know about the latest development in the field so you can learn how to study more effectively!

Learning goals:

- ❏ Understand the value of learning how to learn
- ❏ Become aware of common learning myths
- ❏ Learn proven study techniques

WHY LEARN HOW TO LEARN

After having been through school for years, you might be thinking: "Don't I already know how to learn?". Well, if you've ever thought learning was boring, difficult or frustrating for no real reason you've been taught wrong.

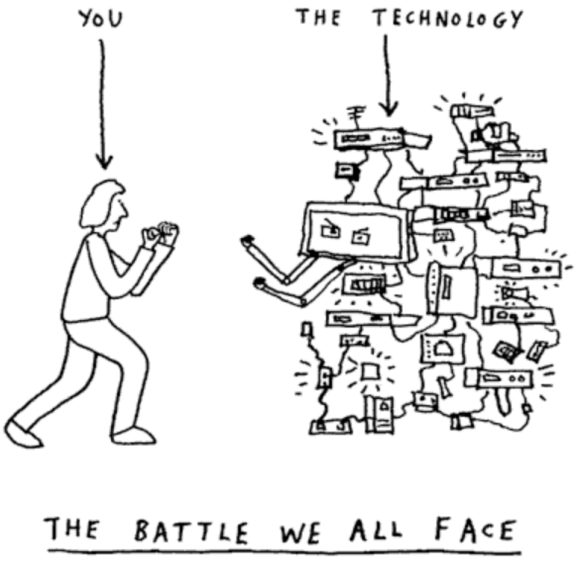
Most traditional schooling systems teach you *what* to learn, but almost never *how* to best learn it. Often there's an implicit belief that people will figure it out themselves. What it really does, however, is leave it up to chance whether or not you actually learn at the right speed (or at all)!

When it comes to software development, learning how to learn is indispensable as well. This is because the tools, frameworks and technologies are evolving at an increasingly faster pace.

This means that the things you learned in school might be obsolete next year. What a field we're in, right?

In order to better prepare yourself for this, it's therefore important to better understand how we learn: how our brains process information and what we can do to make best use of this.

This way we'll better be able to adapt to the fast-changing world of software development!



Learning Materials

- Do Schools Kill Creativity?
- Learning How To Learn

LEARNING MYTHS

As most time in education is spent on learning *content*, there have been many misconceptions about the actual learning *process*. In this section we'll highlight some to show you what you can start doing differently today!

MYTH 1: Multitasking gets more done

This myth is one of the most prevalent ones. On the surface it makes total sense: doing many things simultaneously allows you to finish all those tasks quicker, right?

Turns out this is wrong, especially with regards to studying. When you multitask, your brain has to switch context very quickly. This results in *incremental losses of energy*, *superficial attention* and *a state where you get easily distracted*. Instead of reducing you're actually increasing your working hours!

"Why can't I do homework and watch TV at the same time? All the other kids are multitasking."

Lesson: What you want to do instead is engage in monotasking: the act of focusing on one task at a time. By doing this you will be able to engage in "deep work", a state in which your brain is more inclined to absorb new information and be more creative.

This is especially true when it comes to software development. Have you ever wondered why most programmers put on headphones and like to work by themselves? They want to achieve deep work in order to do their work!

Learning Materials

- What Multitasking Does To Your Brain
- Forget Multitasking, Try Monotasking

MYTH 2: Memorization signifies learning

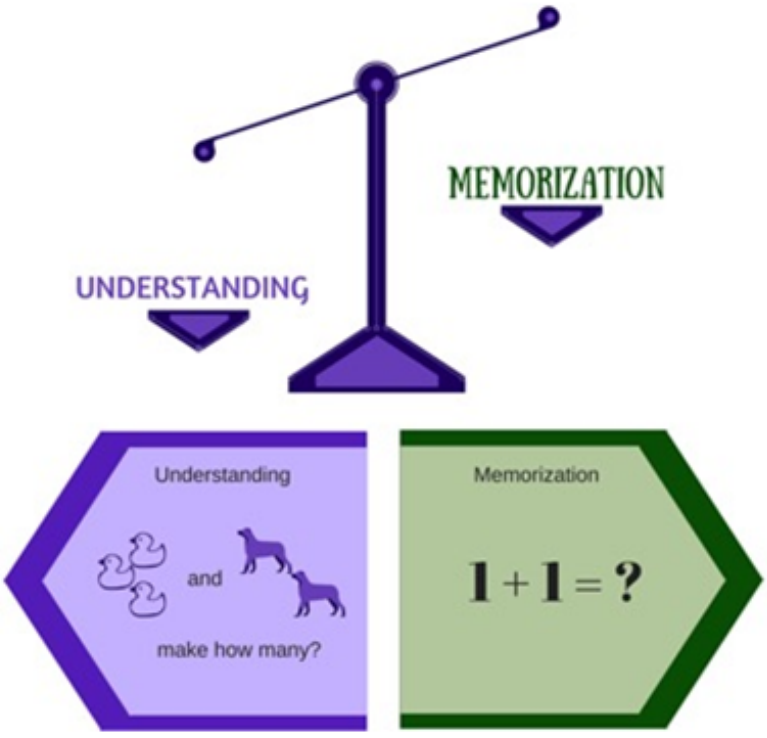
If you've ever crammed before an exam, you know all about memorization. Simply put, it's the act of *trying to recall facts without caring about why it*

exists in the first place. In this way you might be able to recite what you learned, but not to recognize how the underlying principles might apply to new situations.

Learning for understanding, however, shifts from looking to reproduce facts and knowledge to *forming relationships between those facts and knowledge, to explain those relationships and to analyze them*. In this way you'll be able to recognize patterns when encountering new situations.

Learning for understanding is how someone becomes a senior software developer. This is because:

- It allows them to think of better solutions to a problem, rather than the "only one they know of"
- It give them the ability to understand complex topics faster and more deeply



- It helps them to recognize patterns across various programming languages/frameworks (allowing them to understand how to use new ones quicker)

Lesson: In order to learn for understanding, an effective way is to ask questions about the material while you're learning it. This doesn't necessarily mean being more active in the classroom; it can be as simple as asking yourself "why" when you take in new information you don't quite understand yet, and then trying to do research to answer your own question. In essence, this is what curiosity is!

Learning Materials

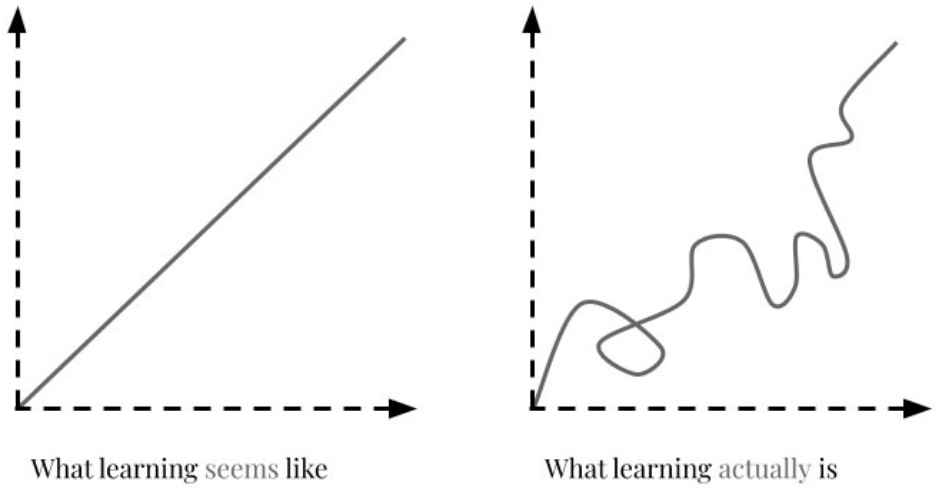
- Memorization vs. Understanding
- How to Develop Understanding

MYTH 3: Learning is linear

When learning to pick up a skill it's quite common to have expectations. Most of the time we want to grow steadily and thus expect our growth to go linearly. This expectation is formed by various influences, such as:

- Social media showing you examples of excessively talented people
- Comparing yourself to friends who learn faster than you
- An unconscious urge to prove that you're good enough

Truth is that *the brain doesn't work that way*. It needs time to make sense of the complexity you are exposing it to. And learning software development as a skill is very complex.



As the graph shows, learning is actually nonlinear. At first you will make some quick gains, but then you'll get stuck. This is called a *learning plateau*: a period of time where it seems you're not improving at all. What's actually happening is that *your brain is taking its time to make sense out of the complexity*.

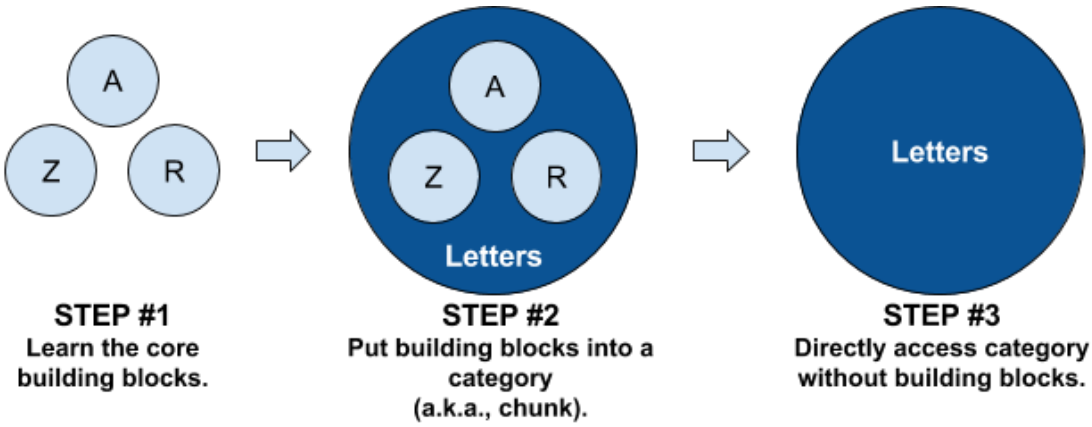
Learning Materials

- How to Break Through Any Learning Plateau and Never Stop Growing

LESSON: In order to make the most out of your learning, you should *change your mindset*: developing your abilities will take all the time it needs. Instead of focusing on how fast you think you should, you should focus on asking questions, doing research, discussing with peers and other activities that engage you in the learning process.

Lesson: Another way to make learning more efficient is to split new information into smaller “chunks”. Let’s say you’re learning a new programming language (i.e. JavaScript). Instead of getting overwhelmed by the complexity of the task...

1. First, you make it more manageable by looking at each concept separately
2. Then you categorize them by type (and in the process think about how concepts relate to each other)
3. Finally, you apply your knowledge without the need to think about the separate building blocks anymore!



As a professional software developer, you’ll feel like you’re on a plateau most of the time. This is completely normal! While working, you’re unconsciously learning to recognize the patterns between different software systems. Trust the process and you’ll get to senior level much more smoothly (and way less stressfully).

Learning Materials

- Chunking

These were just a few of the many learning myths that are still prevalent in our learning culture today. If you're interested in finding out more, check out the following...

Learning Materials

- 5 Learning Myths - Debunked

STUDY TECHNIQUES

As a professional software developer you'll be engaged in learning and studying all the time: learning a new tool or technology, becoming familiar with various codebases or reading documentation are an integral part of your daily work.

It's clear that it would be massively beneficial for you to become more effective at learning and retaining new information. The following are several scientifically proven techniques that will help you make the most out of your time learning!

Technique #1: Test Yourself

Everyone hates exams. The main reason why is because you're forced to recall

information on demand. This takes a lot of effort and you'd much rather be scrolling through Instagram, looking at funny cat videos!

However, there's a purpose behind tests: *you have to actively recall information and reproduce it in order to measure your understanding.*

What this does is strengthen the neural pathways in your brain, thereby retaining the information quicker and for much longer.

In order to make use of technique, however, you don't need to do an exam. Instead, you can get started by asking yourself questions about the material you've just gone through right after you finish (without looking at the answers!). This is known as self-testing.

Try it out right now using the following questions to reflect on what you just learned:

- What is "self-testing"?
- What is "active recall"?

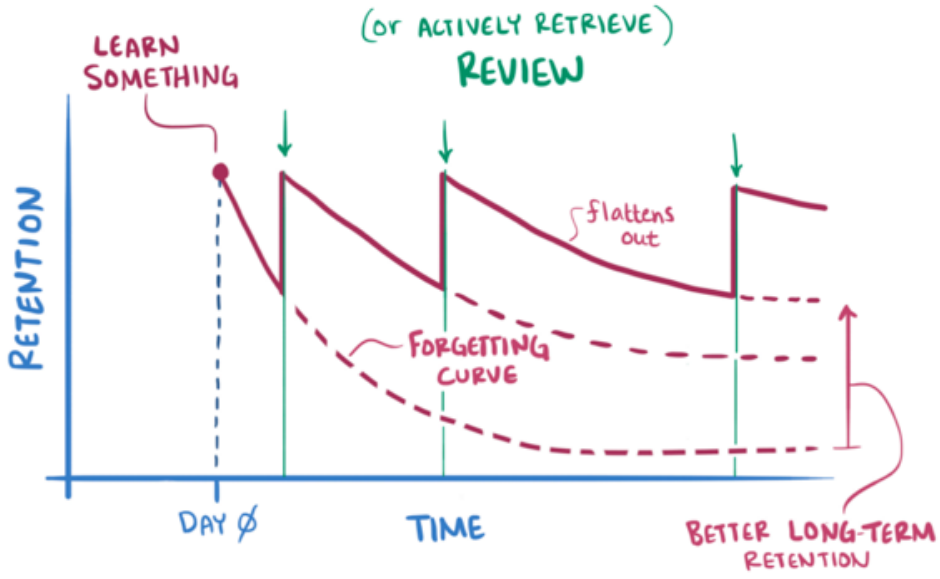
Learning Materials

- How "Self-Testing" Can Help You Learn Anything Faster
- 5 Practical Ways of Using Active Recall

Technique #2: Spaced Repetition

We are able to learn new things all the time, no matter what age we are. This is because of an attribute of our brains called *neuroplasticity*. It refers to the brain's ability to change and adapt as a result of experience and continued exposure.

It works as follows: you learn something new today and then review it again tomorrow. And then two days later. And then 4 days later.



This means you can't just read or watch something once and then know it. You have to *repeat exposure to the new information over time*. This technique is known as spaced repetition.

This is how you build your skill as a software developer: every day you're going back to the code, thinking about it and making changes. Over time you'll find that you develop your understanding gradually, until it becomes second nature!

Learning Materials

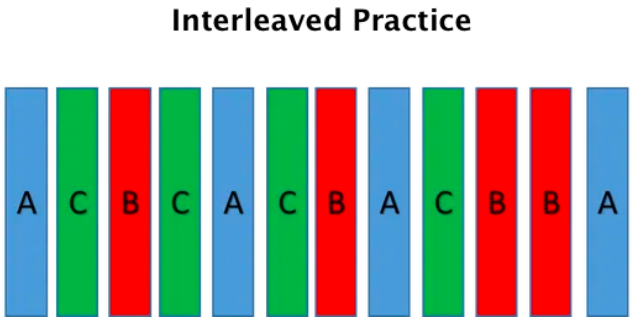
- Neuroplasticity
- Spaced Repetition In Learning Theory

Technique #3: Interleaving

Studying the same topic for hours might make you feel bored very quickly. This is your brain signalling that you need to do something else. Listen to it!

The moment you're bored you actually can't absorb any new information any more. That's when it's helpful to do something else.

Imagine going to the gym to lift weights. Do you think exercising for 8 hours straight will help you get fit faster?



The technique of interleaving also applies to your work. No real software developer can write code for 8 hours straight. If they did, it would create a burnout in no-time.

During the course of TAP you will be exposed to a range of skills that help you develop as a professional. All of these skills are linked - and it is part of your responsibility in TAP to continuously make connections between them.

The program will help you break things down into manageable constituent elements. Some of the information you may have been exposed to, and others may be entirely new to you.

How you piece these together will help you to create memory associations, assisting you in remembering skills and knowledge in the long-term. By learning in this style you will promote your critical thinking abilities. It will also help you in being a well-rounded employee - your success in your future job is not only dependent on the technical skills you bring to the table.

Practicing in this way may not be easy at first - particularly because you likely have been exposed to a different form of learning and studying in those old ways has become a habit.

Learning Materials

- Study Strategies: Interleaving