The new e-ra of learning.

felpapp.

Credits

Author:

Milica Sucevic

Slavo Popovic

Reviewers:

Nikola Stankovic

Una Sucevic

Designer:

Ana Quirosa

Table of Contents

Part I

VOCABULARY

INTRODUCTION

Part II

FELPAPP

STUDENTS

Part III

Part IV

MILESTONES

Part I

Vocabulary

```
>>> print("Hello, World!")
Hello, World!
```

The level of your technical knowledge might vary from being a total newbie to someone with some experience in the field. Whatever the case may be, if you've just wondered what that line of code meant, then you've come to the right place! **So don't panic - sit back, relax and enjoy your flight! :)**

The aim of FelpApp is to make you feel part of this journey and, most importantly, to make you feel confident in your abilities. Taking into consideration that you all come from different backgrounds, we aim to provide you with tools you can always rely on.
Once upon a time, we were all beginners. Throughout our journey, we used a variety of tools to assist us with multiple challenges until we figured out the right path to take. The good news is that you don't have to walk this path alone!

The tools and websites in this book will help you comprehend what is happening, which directions to take, and what some of these terms may mean. They are an excellent source of information that can be used as a reference at any time! Since the book is in digital format, you can always find the relevant links by searching back or by bookmarking them in your internet browser. What exactly is a browser? Discover more by clicking on the link below!

The tech terms Computer Dictionary

Basically, you can search for any technical term, and you will be presented with examples and background information. One day, when you come across the same term again, you will be able to recall what you searched for more easily.

Alternatively, you can use YouTube if you're more of a visual learner. To ensure you are never left in the dark, we will also offer book recommendations if that is what you prefer.
As well as this, we have a community where you can ask questions and discuss different topics with your teammates at any time!

Listed below are some other helpful links we found on this topic:

- 99 Tech Terms that you should know
- Glossary of IT Terms

Introduction

There are many educational institutions nowadays that offer Computer Science courses and programs. However, you can also learn it on your own with a little help from the Internet, the right guidance, and the community supporting you in your endeavour.

Self-education is beneficial for both professionals and beginners. Professionals can benefit from it when it comes to staying on top of the latest technology. On the other hand, beginners can benefit from it in terms of their personal interests and development. Learning can be adjusted to fit into other daily obligations by adjusting the style and dynamics.

According to the Stack Overflow research, half of the developers in the world lack higher education, and many lack any education at all to begin with. This proves that information technology can be learned independently. This is very encouraging especially for newbies, like yourself, who are planning on stepping into this field on their own to one day do this professionally as well.

Take a look at this video from Tyler Dewitt
(Online learning could change academia - for good)

Image by Freepik

Many start-up companies will give juniors a chance even if they lack relevant degrees. In order to be taken on board, they must demonstrate adequate knowledge, skillsets and a willingness to devote hours and hours of hard work. This is where we come in to help :)

Image by Freepik

NOW IS THE TIME TO DISCOVER HOW TO ACHIEVE YOUR GOALS!

Step 1: Can I do this?

Let me start by asking you: How do you feel about computer science in general? Do you see yourself participating in this?

Having the ability to code or build a server is a skill that anyone can learn at any time in their lives. Even though most people would agree to a certain extent with this statement, in our opinion, it is much more interesting and challenging to ask whether or not someone has the ability to succeed as an engineer or a programmer?

Yes... But, not because someone is more or less smart. For starters, not everyone has the capability to sit in front of a computer screen all day long, looking at the screens and composing code without tiring out after a while. Also when you are faced with greater challenges, you are more likely to feel frustrated and nervous. Which leads to one of the reasons why so many people who want to become programmers or engineers, in the end, fail to fulfil their dreams and eventually drop out.

Check this article out
Why Consistency and Persistency are 2 vital ingredientes to Succes?

Thus, questions such as:

- How do I become a programmer?
- What is the best programming language to learn first?
- Which career pay the most?
- When learning a new tool, how long does it take?

These are all the wrong questions for a number of reasons

Getting into a profession for the sake of how much it pays is the wrong motivation. Money is like energy. Investing more time and energy into developing yourself will yield more gains over time. Differently shaped.

Also, in terms of time... If you opt for something easy and fast, it will only show you're not patient enough. That you want to choose something that is easy and fast. It is therefore important to choose your questions wisely.

Although we cannot generalise everything, this is how it works in 95% of cases. It's understandable that these kinds of questions are on our minds. Those thoughts do get ingrained in all of us, and that's okay. However, they shouldn't be our primary questions... or mots importantly - OUR PRIMARY FOCUS.

Whenever we talk about technology, computers, and the Internet, we must take into account a few general terms. This is so that these things do not scare you, but rather appeal to you. As an overall strategy, it offers a great deal, but it requires a high level of discipline.

Image by Freepik

Next, and most importantly, you should develop the ability to think logically.

What are your strengths when it comes to solving logical problems?

Regardless of whether you can complete the most complex tasks of this type, you should be able to solve the simpler ones, and you should understand the underlying reason for the solution and the background to the more complex ones. If you find that you are inclined towards solving problems like this and that you have a natural inclination to do so, then you should begin studying programming and/or engineering.

Is it worth a try?

Absolutely YES! Invest a few months in programming or learning how to engineer something, and you will not regret it. Take my word for it!

In the end, you might decide this isn't for you. And that's okay too.

However, you will have a journey under your belt that you will be proud of for the rest of your life. And all the challenges that you'll face and OVERCOME... You'll gain a whole new perspective, and you'll open up many other potential paths.

Nonetheless, if you do:

1. Find that you get a sense of satisfaction from solving problems and bugs
2. Tick most of those boxes above
3. Are committed to continuous learning and making a difference in the world

Then this is the right place for you ;)

Image by Freepik

Step 2: Evolution through mindfullness

After deciding what you want to do and what path you wish to pursue, in addition to working in this field and for tech companies, there are also these things to consider:

- Taking part in exciting projects
- Using cutting-edge technologies
- Having a good income
- As well as frequent meetups, socialising, work trips, remote work, etc.

However, there is also another side to this as well, which involves significant responsibility, sacrifice, renunciation for the sake of progress and learning:

CONTINUOUSLY

In a time when technology is rapidly advancing and gaining momentum, your main focus in technology will always be on following the latest trends. It is therefore necessary for you to be prepared for **adaptation** and continual development.

It is imperative that an individual has a good grasp of the English language in order to easily master new technologies. The majority of tutorials, books, videos, docs, and references are in **English**. Almost all communication is done in English, and every problem or solution will most likely be found in English as well. The more so when you are a part of large tech communities.

A few tips to keep in mind when entering the world of computers, programming, or any other IT occupation are as follows:

1. Begin with the basics and work your way up.

2. Try not to compare your progress to others, we all progress at different rates. Regardless of how difficult the task is, persistence and perseverance are essential. Trust me, there will be some challenging days ahead! What sets you apart from others is your focus on the goal!

3. Learning a new skill set is like that - every problem has a solution, and you'll learn that as you gain more experience and knowledge, but also every problem is unique and should be approached differently. With time, you'll get there. There is a saying that says patience is a virtue.
   Additionally, competing with others can have a negative effect on you because you will have to go faster to catch up. This is not a solution. The only competition you have is yourself - strive to get better every day!

4. As you research the basics, at some point you have to decide where you want to improve, what you want to learn, and finally what you want to accomplish. Technology covers a wide range of fields - from the web, mobile apps, programs, and games to robotics, science, and the Internet of Things.

5. Start off small, not out of carelessness, but because you care about your progress. Write down a list of all the things you want to accomplish that day, even if it's just reading one page, watching a video tutorial, or writing 20 lines of code.

6. Set up a git account and commit significant changes every day. You'll keep track of your progress this way, and at the same time you'll be building your portfolio from the very beginning! (Git explained in greater detail later in this book)

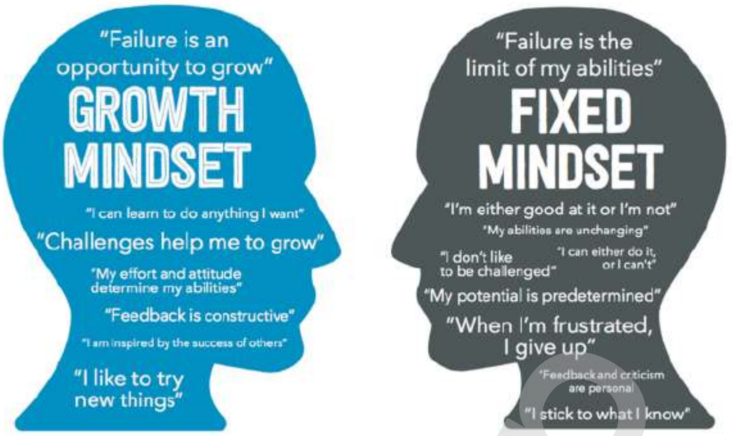

"Failure is an opportunity to grow"
GROWTH MINDSET
"I can learn to do anything I want"
"Challenges help me to grow"
"My effort and attitude determine my abilities"
"Feedback is constructive"
"I am inspired by the success of others"
"I like to try new things"
"Failure is the limit of my abilities"
FIXED MINDSET
"I'm either good at it or I'm not"
"My abilities are unchanging"
"I don't like to be challenged"
"I can either do it, or I can't"
"My potential is predetermined"
"When I'm frustrated, I give up"
"Feedback and criticism are personal
"I stick to what I know"

Photo Credits

The last few years have afforded us the opportunity to supervise talented junior engineers from whom we have learned a great deal. The experience prompted us to write this book and share it with young talents like you, with the hope, perhaps one day to even work together!

In this book, many of the points are applicable to other professions within technology so that you can become exceptional at anything you want, even if you don't program or do engineering. Keep reading this book, it will guide you in the right direction, and then pass it along to other people who may also find it useful.

Is it necessary to have a degree?

Identifying your personal goals is the first step toward answering this question. Are you looking for a traditional school education that offers a well-rounded curriculum? Then by all means, go for it. Earning a degree shows that you have put in the hours, sweat, and effort to learn something new.

If you're interested in entering the technology industry quickly, you don't necessarily need a degree in coding or computer engineering. **FelpApp** offers this type of platform that can help you gain the skills you need in a time-saving and cost-effective manner, making it easier for you to get started.

Occasionally, we may have to consider the fact that some companies continue to operate in a traditional manner. An applicant without a degree in Computer Science, for example, may not be considered for a coding position.

Occasionally, we may have to consider the fact that some companies continue to operate in a traditional manner. An applicant without a degree in Computer Science, for example, may not be considered for a coding position.

In recent years, this has become less and less common. Nowadays, employers have different expectations and hiring processes than they did in the past. Besides demonstrating your coding skills and engineering expertise, modern employers will look for portfolio examples from your resume.

Their interest is in seeing how you deal with different tools and challenges, as well as your solutions that demonstrate your technical acumen. In the end, it comes down to whether you are capable of doing the job, not whether you have a degree. Of course, they are also interested in how you will fit in with the rest of their organisation.

The alternative certification route has gained more respect over the last few years. Data we have on employers' perspectives indicates that nearly 90% of them believe self-taught developers and engineers are just as prepared, if not more so, than degree holders. Many talented, self-thinking individuals began their careers with nothing more than an online certification or a basic understanding of some common languages and tools.

Therefore, when others say you can't get a tech job without a degree, that's simply untrue. It is entirely up to you how hard you work and how many hours a day you devote to your personal development.

You might be interested in this article on
How To Become A Programmer: Basic Tips

Image by Freepik

Step 3: Start Line

The participation is extremely convenient and does not require any additional equipment besides a laptop with internet access.

Image by Freepik

Through the course of this book, we will be covering all the steps you will need to take in order to have a successful training program and to ensure that you are prepared to start your training.

During the first six months of the training, we will cover the fundamentals, tools, expectations, and everything else that will guide you from the very beginning to getting interviews and starting your career in junior positions.

Take a look at this article about becoming a developer:
With No Experience /Degree/ Any Age

Last but not least, I want to share with you a great inspirational video from Elon Musk explaining the importance of getting a degree in a relevant field in order to get a job, as well as focusing on the crucial elements at first that are necessary at the beginning of the process.

Education system as seen through the eyes of Elon Musk

Step 4: Pomodoro Technique

Want to become more productive? Put the pomodoro technique to the test!

There are times when we feel that no matter how long the day is, it is not long enough for us to accomplish all our daily tasks. With each effort we make, we end up investing greater and greater strength into completing all the tasks. Despite our best efforts, the result is nothing but stress and reduced productivity. Implementing the Pomodoro method will help increase productivity without causing too much resistance.

The Pomodoro (Italian tomato) technique was invented by Francesco Cirillo in the late eighties. His goal was to improve concentration and focus, get rid of procrastination, and have enough time to complete all tasks on time.

How does that relate to tomatoes?

The name of this time management technique is derived from no less than a kitchen timer shaped like a tomato. This is the most powerful tool and the only one you need, as it has worked for many who needed better productivity, so why not give it a try yourself?

With the Pomodoro Technique, you can accomplish the following:

- Calculate the time it will take you to complete a task, no matter how complex it is.
- Focus on a specific task for a longer period of time.
- Reduce the number of interruptions that distract you.
- Maintain a well-defined schedule for completing tasks.

With the Pomodoro technique, you will be able to achieve quality work in a shorter amount of time and within a time frame that you have defined.
Whenever you are concentrating on the execution of only ONE task, you tend to focus on that one task and finish it in that time interval rather than working on several tasks simultaneously.

-> Try out this awesome Pomodoro Timer

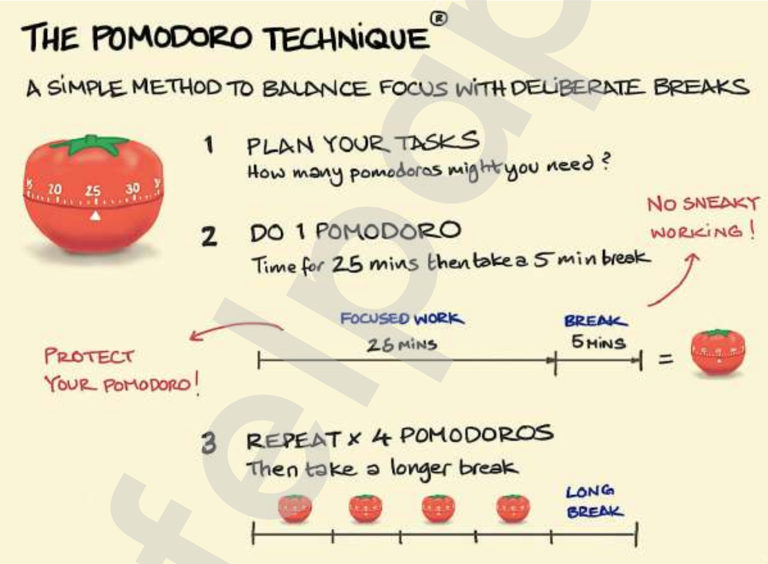
You will perform all duties in 25-minute intervals with breaks. Essentially, you can break down difficult and complicated tasks into smaller ones by making a list for the day. Group your “small” tasks that probably don’t take 25 minutes into one unit if you have several “small” tasks. A pomodoro is 25.

minutes of focused, active work on a specific activity without interruption. Pomodoros cannot be shared, so either you have completed the task or you haven't.

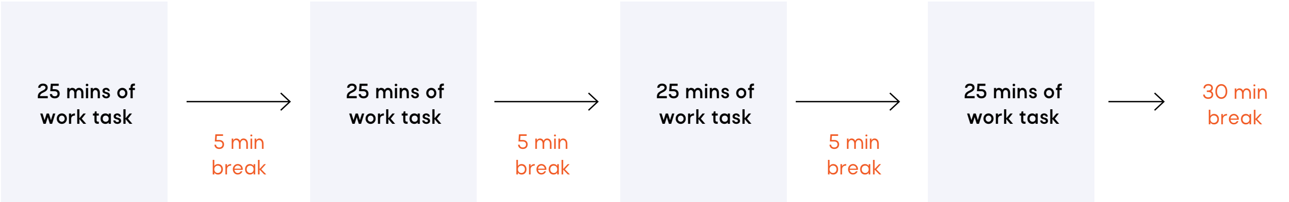
Steps are:

1. Create a list of tasks for today and set the timer for 25 minutes
2. Work on the first task during that time
3. Take a 5-minute break when the timer rings
4. After the break, set the timer again for 25 minutes and continue working.
5. After four pomodoros, you can take a 15 to 30 minute break.

At first, you might think it's unnecessary to take a break every 25 minutes. After trying it, you will realise you are more productive and can handle your work better when you take a short break.



Here's how it looks:


25 mins of work task → 5 min break → 25 mins of work task → 5 min break → 25 mins of work task → 5 min break → 25 mins of work task → 30 min break

During this time, there is no interruption to work in progress! The timer does not have a pause option. If you must stop working for any reason, take a break of three to five minutes and set a timer. Consequently, you "teach" .

yourself not to be interrupted and to devote yourself fully to completing the task.

Conversely, if you finish the task before the time expires, you can use the remaining time for an additional review. The timer can be reset if you have a lot of time left or if you do not have enough time to do the next activity. A short break will help you move from one activity to another.

If you are interested in reading more on Pomodoro Technique please read more here:

Pomodoro Technique:
Beat procrastination and improve your focus one pomodoro at a time

Step 5: Programmatic Learning

Programmatic learning: what is it?

In essence, programmatic learning consists of a mixture of learning intervention delivered over a period of time. The program actively supports and encourages continuous learning. As it focuses on established outcomes and objectives, with performance measured against set metrics and reviewed regularly, it is ideal for delivering complex targets.

Demands from businesses and markets continue to drive corporate learning requirements as well as for the individual to pursue tech careers. Although workplace training has been in flux for many years, it may never be the same in the post-Covid era. The transition to remote working has been relatively easy for businesses and the majority of them have migrated to remote work, which has increased investment and adoption of new technology.
Automation trends have existed for some time; new technologies have transformed how jobs and skills are applied. But COVID-19 has added a new dimension to it.

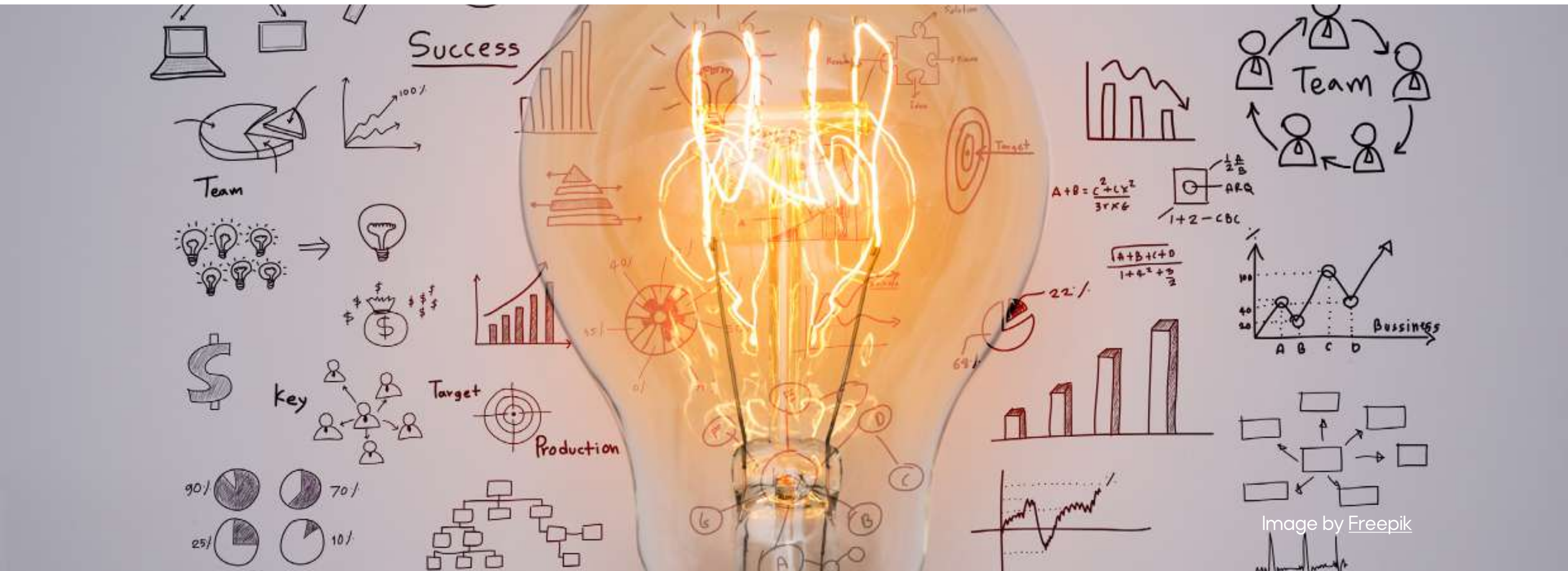


Image by Freepik

With work and everyday schedules becoming more flexible, we'll need to offer new learning and collaboration opportunities in the upcoming years. The use of digital communication channels and other technologies will need to be seriously readjusted in order to make learning more efficient. Future economic success will require a shift in attitudes and work cultures, not just skills development.

The challenges of social distancing and remote work can be addressed with tools available here at FelpApp. These modern programs offer students the chance to ask questions, test each other's knowledge, and receive immediate feedback through online platforms and discussion channels which is supporting continuous learning and progression. In the near future, many businesses will adopt this culture!

The world's economic recovery depends on digital skills, but the Learning & Development departments just aren't keeping up with the new trends. Currently, only half of companies are capable of providing employees with digital skills training on the job, even though new employees expect companies to invest in their training.

Providing mentorship and 1-to-1 training to subject matter experts is one way in which organisations can cultivate new talent. Based on our students' experience and the needs of companies, we design a program to fill skill gaps. Especially in the field of technology, where everything is headed in the near future.

Our goal is to produce a new generation of tailor-made professionals with targeted goals from the beginning of their career.

Image by Freepik

In these changing times, our students need support and upskilling. The world has become highly mutable, so now is the time to address its needs!

Read these articles on:

- **7 Reasons Why Continuous Learning is Important**
- **Continuous Learning: What It Is, Why It's Important, and How to Support It**

Step 6: The Power of the Unknown

Since forever, people have been afraid to say “I don’t know”. In the past, employment was determined mostly by one’s skill and knowledge in a given trade. To become a “professional”, an apprentice usually had to undertake intensive training. “I don’t know,” was an indication of incompetence. Nevertheless, work environments have changed.

The value of in-depth expertise has diminished in today’s workplace, where potential has taken precedence over expertise. When one learns something new, their capability and potential are more valuable than vast knowledge.

Embracing not knowing is one of the most important and challenging aspects of life and growth. Instead, having the ability to live in a state of wonder and possibility, like children, allows us to be creative, open-minded, and willing to explore.

How much more could we learn if we didn’t have to know everything all the time? What would it be like if we let go, and accepted and trusted, the unknown?

Image by Freepik

The knowledge of the world is limitless. There is no end goal here, **learning is a process.** When you reach a point where you feel confident and know something and then suddenly you face a challenge, don’t get discouraged. Keep going - You’ve got this! Growing requires curiosity and not knowing, as we become more engaged in the process as we learn and become curious.

The internet created a massive shift in the way people access information. The world is literally at our fingertips when it comes to information, data, and knowledge. Previously, things shared with closed circles were restricted to a selected group of people. The information is available now and it is there for you to take it, if you want it.

Ultimately, transition is inevitable, and no person or place stays the same forever.

Check these articles on:

- How to Embrace Change in Your Life
- Don't Know Much About Tech? Don't Panic. Here Are 5 Simple Solutions to Stay Informed and Innovative.

The following short video from Google discusses this topic in particular:

- Google enlists Marcus Rashford to ask life's searching questions in latest ad push

Step 7: Showcase your work

GitHub

Recruiters and hiring managers may ask you, "Can we check your GitHub?" during a job search, and the applicants may not realise how recent this practice is.

Here at FelpApp, you'll be working on a wide variety of projects, which you'll be able to show off right away! This is why it is so important to emphasise just how beneficial it will be for you to use it right from the start.

You can showcase your work and projects on your GitHub profile. There is a feature on Github that allows you to pin specific projects to highlight your contributions and work that you are the most proud of as well as to save gists.

-> If you are curious about gists,
please check official Github documentation on Creating Gists

Coding snippets and single files can be shared quickly and easily with GitHub gists. The contents of a gist will appear on your profile when it is pinned. In later chapters, we will talk about Git in general, including Github and Gitlab, as well as how to set Get Started and create your first Github account.
GitHub profiles are being used to assess developers now, so newbies like you might be wondering what exactly employers are looking for.

Please read this article on GitHub: Advantages and tips for a profile 10

Step 8: Mentorship, Study Buddy, Community

Not sure where to go with your career path? Interested in improving your skills and need guidance? Or you are just out of college and in search of a job? Perhaps you're seeking a promotion? If you are in any of these situations, you could benefit from a mentor or coach.

Choosing a mentor is important for a number of reasons:

- Experience: Mentors have already travelled the long path and can help you avoid mistakes as well as to advance as quickly as possible.

- Focus: You can identify your improvement areas with the help of a mentor.

- Confidence: Support and encouragement are provided by a mentor.

- Community: Introducing you to an online community that will encourage your growth and provide you with knowledge sharing opportunities.

We use a variety of learning methods today, from eLearning to videos to even learning experience labs. They may fall short, however, when faced with more complex challenges. There is a pandemic of digital transformation in learning, increased challenges in attracting and retaining talent, and a need for personalised, digitally supported programs to help deliver high-value transformational projects.

As a result, they all require mentorship, support, and connections with their peers in order to succeed. Learning needs to be integrated into the workflow. The training programme must provide students with opportunities to practice in a spaced manner, apply the knowledge and retain it as they learn it.

Read more on Every Developer Needs a Mentor

Virtual Study Budy & FelpApp Community

The concept of study buddies is not new - someone to motivate and hold you accountable - but did you know you can recruit your own study buddy at any time, from anywhere?

This is where Virtual Study Buddy comes into play - It is without a doubt that the popularity of “study with me” videos has exploded over the past few years as it has been increasing steadily. There is a sense of therapy, motivation, and visual appeal in these videos. A lot of them come with ambient sounds, high-quality music, or you can listen to no sound at all if that is what you prefer as long as the sounds are appealing to you.

If you feel like you might need a virtual study buddy whilst you are completing your daily tasks, you don’t have to do it alone. Through thousands of videos, live streams, and even Discord communities, like minded students share tips, notes, and ideas on the site. Here are some of the best “study with me” videos:

5 ‘Study with Me’ YouTube channels for working or studying

You can always play this if it helps you with your studying methods. It is ultimately up to you and what works best for you in your own setup and environment.

FelpApp’s community is also available via Discord, where you can connect with us and other members. If you have never used Discord before please read here what it is and how it works!

What is Discord?

Image by Freepik

Part II

Team

Slavo Popovic

Chairperson at FelpApp
📍 **Located in Miami, US**

Slavo Popovic is a self-taught Senior Web Software Engineer. Over the past decade, he has been involved with the industry. At the beginning of his career, Slavo worked as a PHP developer for websites. It was his responsibility to develop and maintain WordPress websites. Afterward, he transitions to JavaScript, which is where he is currently working. Throughout his career, Slavo was involved in multiple successful startups, as well as big systems and companies. His expertise includes JavaScript/TypeScript, React/React Native, NodeJS, MongoDB, SQL, AWS lambda, AWS and Firebase.

Team

Milica Sucevic

Chairperson at FelpApp
📍 **Located in London, UK**

Milica Sucevic is a self-taught DevOps Engineer. Her experience in this area spans over six years. Her first job was working as an IT Technician, and then eventually, she transitioned into Infrastructure Engineering and specialised in Cyber Security. Due to her interest in innovation, automation, and scalability, she has shifted her focus to DevOps Engineering. Milica's journey started as a Junior DevOps Engineer and led to her current position as a DevOps Team Leader. Among her specialties are Cloud Computing, Kubernetes, Docker, Terraform, Ansible and scripting.

Team

Nikola Stankovic

CTO at FelpApp
📍 Located in Miami, US

Nikola is a self-taught Full Stack Developer. For the past two years, he has been working as a JS/TS developer for Prototype.NEXT/MimiCom24. He specialises in Front End Technologies like React and NodeJS/Express for the backend. During his career Nikola has developed web apps, optimised web apps for SEO purposes, enhanced performance, fixed meta tags, and correctly structured their HTML tree.

Location

Currently US
& Europe

As of now, FelpApp is primarily targeting U.S. and European audiences.

Community

Image by Freepik

Welcome to Felpapp

Listed below is an overview of our community group chat that we use for information, knowledge sharing, mentorship guidance as well as study support and discussions on different topics related or unrelated to this course, which will help you make new connections and friendships with people with similar minds and ideas that will benefit your personal and professional growth.

Several years ago, there was a video about a business influencer and motivational speaker who was asked, “What would you do if you just lost your job? ”
First, he began by describing the community in which he lives. Among whom is he surrounded? He said: “If I am jobless for longer than four weeks, my connections are poor around me”. So make sure your community grows!

INFORMATION

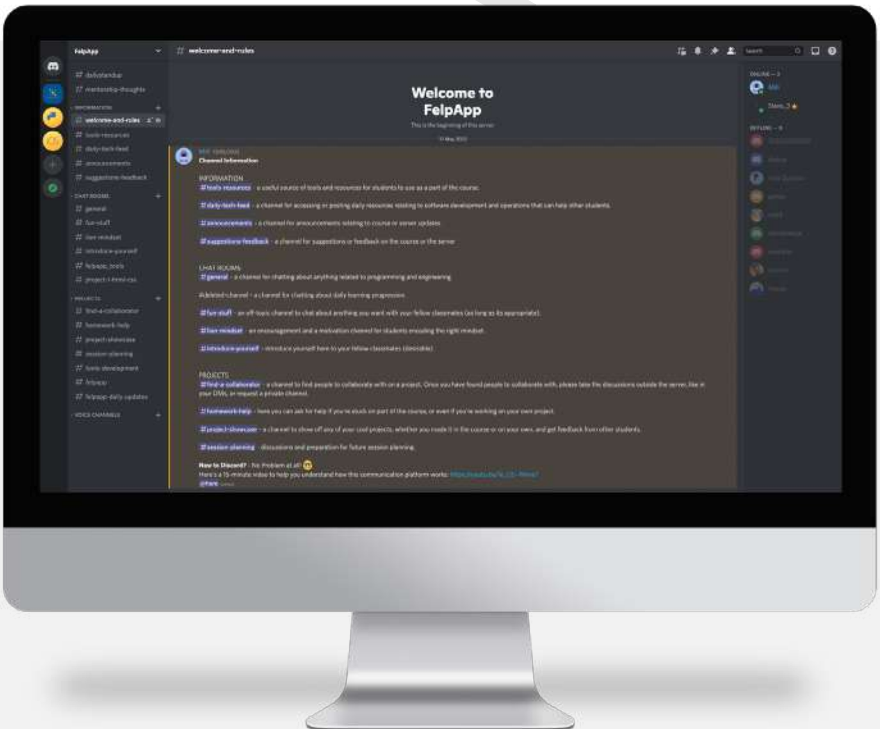
#tools-resources - a useful source of tools and resources for students to use as a part of the course.
#daily-tech-feed - a channel for accessing or posting daily resources relating to software development and operations that can help other students.
#announcements - a channel for announcements relating to the course or server updates.
#suggestions-feedback - a channel for suggestions or feedback on the course or the server

CHAT ROOMS

#general - a channel for chatting about software development and operations.
#fun-stuff - an off-topic channel to chat about anything you want with your fellow classmates (as long as it's appropriate).
#lion-mindset - an encouragement and a motivation channel for students encoding the right mindset.
#introduce-yourself - introduce yourself here to your fellow classmates (desirable).

PROJECTS

#find-a-collaborator - a channel to find people to collaborate with on a project. Once you have found people to collaborate with, please take the discussions outside the server, like in your DMs, or request a private channel.
#homework-help - here you can ask for help if you're stuck on part of the course, or even if you're working on your own project.
#project-showcase - a channel to show off any of your cool projects, whether you made it in the course or on your own, and get feedback from other students.
#session-planning - discussions and preparation for future session planning.



Welcome to FelpApp

This is the beginning of this server.

Channel Information

INFORMATION
#tools-resources - a useful source of tools and resources for students to use as a part of the course.

#daily-tech-feed - a channel for accessing or posting daily resources relating to software development and operations that can help other students.

#announcements - a channel for announcements relating to course or server updates

#suggestions-feedback - a channel for suggestions or feedback on the course or the server

CHAT ROOMS
#general - a channel for chatting about anything related to programming and engineering

#deleted-channel - a channel for chatting about daily learning progression

#fun-stuff - an off-topic channel to chat about anything you want with your fellow classmates (as long as it's appropriate).

#lion-mindset - an encouragement and a motivation channel for students encoding the right mindset.

#introduce-yourself - introduce yourself here to your fellow classmates (desirable).

PROJECTS
#find-a-collaborator - a channel to find people to collaborate with on a project. Once you have found people to collaborate with, please take the discussions outside the server, like in your DMs, or request a private channel.

#homework-help - here you can ask for help if you're stuck on part of the course, or even if you're working on your own project.

#project-showcase - a channel to show off any of your cool projects, whether you made it in the course or on your own, and get feedback from other students.

#session-planning - discussions and preparation for future session planning.

New to Discord? - No Problem at all! 😁
Here's a 15-minute video to help you understand how this communication platform works: https://youtu.be/Ie_CE--Mnvs?
@here

New to Discord?

No Problem at all!

- **Here's a 15-minute video to help you understand how this communication platform works:** https://youtu.be/Ie_CE--Mnvs?

- **Discord install link (iOS, Linux, Android, Windows):** https://discord.com/download

Daily Stand-Ups

A common practice in many sports, including football and rugby, is for teams to meet before games. In addition to keeping the team connected and focused during the game, the conversation serves as a strategic communication tool. The stand-up activity is similar for development and operations teams. In other words, it reinforces what we feel and is commonly called a daily argument. Team members stay informed of environmental events and the team's progression.

In FelpApp, a stand-up is a daily meeting of the core team: the product owner, developers, and engineers. For every team, the atmosphere of a meeting will be different, but it will cover at least three topics:

- **What did I do yesterday?**
- **What am I doing today?**
- **What problems are blocking me?**

The answers to these questions make progress visible and reveal team barriers. Having everyone share their progress also strengthens the team.
Team members are more likely to get excited about the project's progress when each of their accomplishments and plans are shared on a daily basis.

Image by Freepik

To make your daytime meetings more productive, follow these tips:

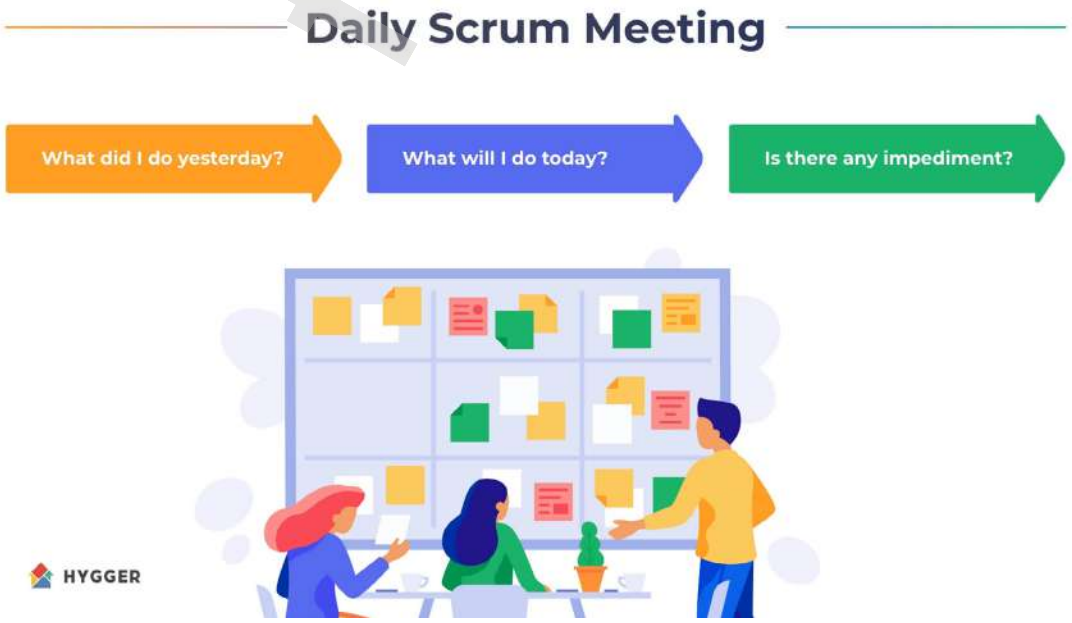
1. **Set a convenient time for everyone**
   Teams usually hold stand-up meetings between 9:00 and 10:00, most of the time remotely these days. This gives everyone time to get a good start on their day, without anyone having to arrive early. Choosing a time that works for all team members is important for geographically dispersed teams.

2. **Ensure the retrospective is backed up**
   In many agile cultures, stand-ups are an essential element, but that doesn't mean they can't be discussed during retrospectives. For some teams, catch-ups take place on a daily basis, but other teams might do it differently. Remember that stand-ups are also agile, so adjust it according to your team!

3. **Keep track of history reports**
   Analyse ongoing reports to learn more about the performance of team members and the progress of projects. Investigate the reasons for lagging performance among some team members.It may depend on how they progress. If there are delays or missing reports, request stand-up time.

4. **Prepare well**
   It is important for individuals to be prepared each day and know what to say. By doing so, the team will remain motivated and the energy of everyone will be maintained. Prepare yourself for standup by knowing what your tasks and blocks are and what you have recently achieved that you can share with your team.

5. **Put your best foot forward**
   Focus is essential to an effective stand-up. A team member will always rotate the spotlight, ensuring everyone has a voice and ownership of their duties. The duration of a stand-up should not exceed 15 minutes.



Students

Vision

Our vision is to enable students from around the globe to enhance their knowledge and skill sets from being absolute beginners to landing jobs in the technology field through a community-based learning platform. Here at FelpApp, we will provide you with the best tools and resources to help you make a career change transition as quickly and smoothly as possible.
With peer-to-peer mentoring, you will have real-world experience through mentorship from peers within the community, with whom you will be working on challenging projects that will enhance your professional development and growth.

During the training, you will learn about the experiences and stories of various software developers and engineers, as well as links to courses and materials. There is a lot of information on the Internet about these topics, so we'll cover everything we wish someone had taught us when we started out.
In today's fast-paced world, new starters should focus their attention as soon as possible on what is necessary. Unless you are 15 years old and intend to devote everything you have to computer science during college and high school. However, if you intend to retrain, or wish to start your own business in the near future, then you most likely won't have this luxury.

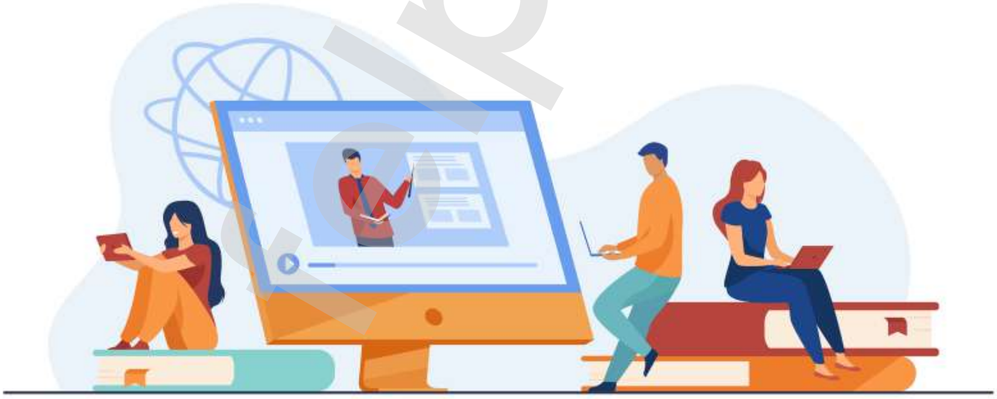


Image by Freepik

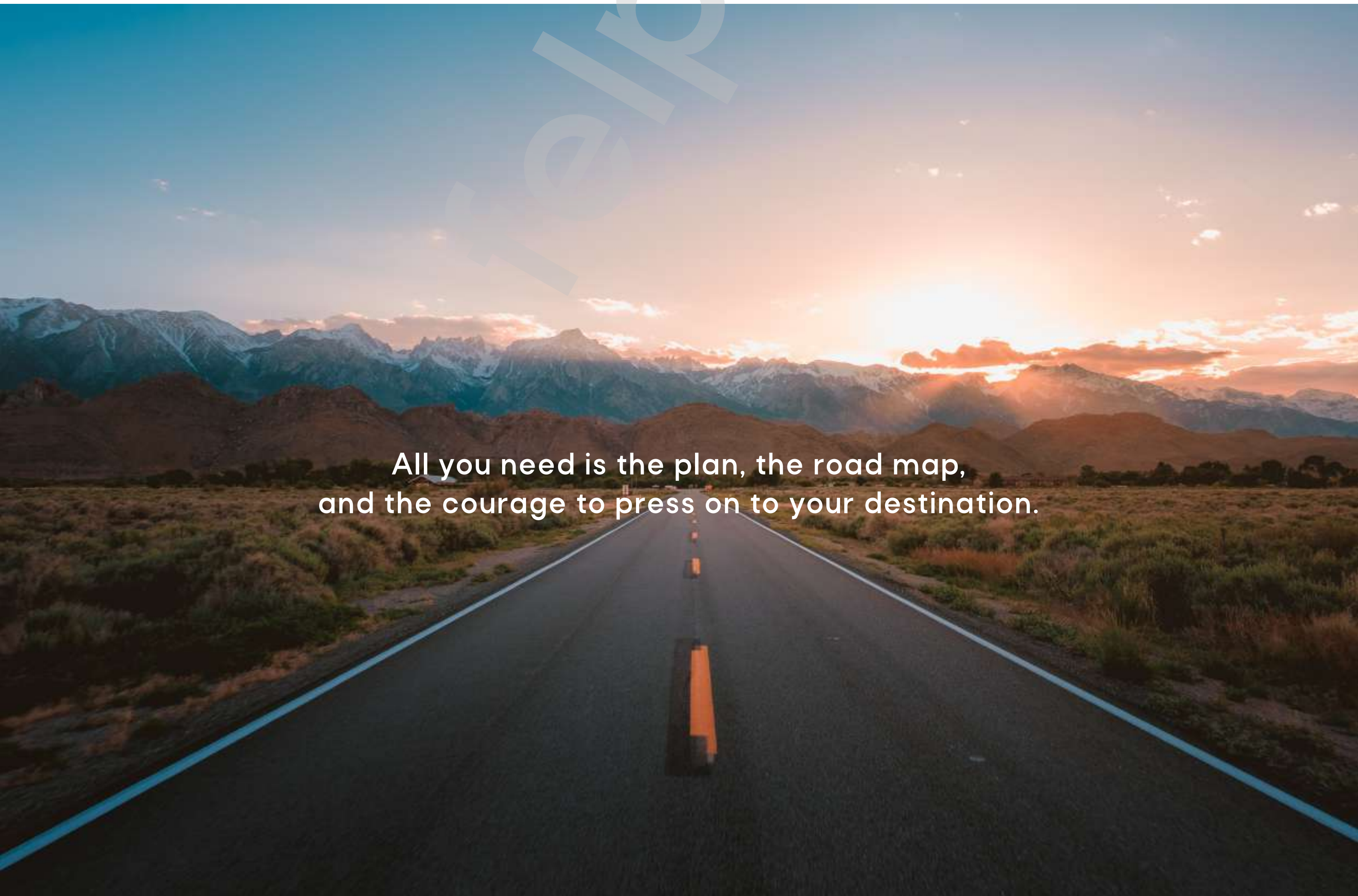
The process of transitioning into a new field can be confusing, disorienting, and time-consuming. Furthermore, you may end up reading things that are totally irrelevant to you or your future career, and the reality is that the world no longer needs half-hearted programmers. In the absence of a diploma from a world-renowned university, a portfolio is more valuable than a CV and diploma combined. Since most candidates for programming jobs do not even have GitHub accounts, you will be treated as a serious candidate if you can provide three relevant personal projects.

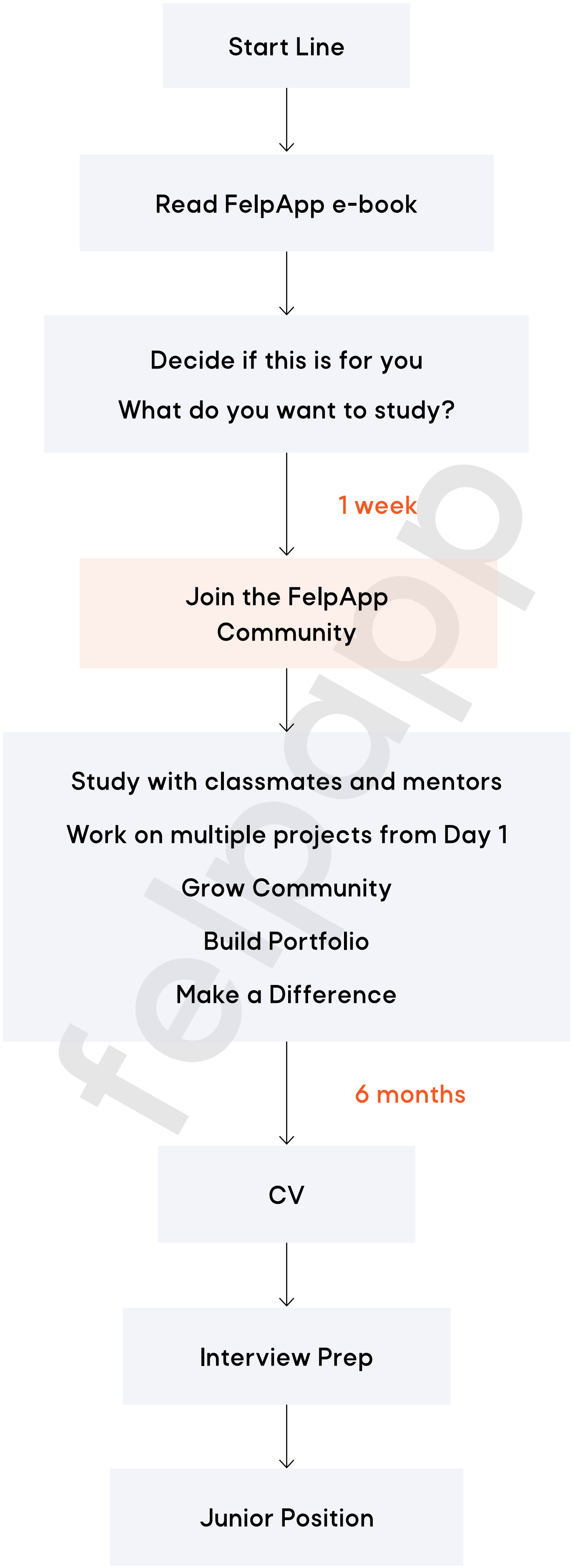
As soon as you sign up, you will be given a week to read this book, ask questions, do your own research, and really think about what excites you. There may not be an answer right away, but the good thing about this community is that you will always have our support, and if you have any doubts you can always share your ideas and concerns and you will always get guidance, but it is up to you to recognise what is the right conclusion for you.

When you've decided what direction you want to go in, envision yourself achieving that goal one day and stay committed to it. There is no guarantee that it will be easy or fast. Neither was it for us, but here we are. You have already taken the first step, which is always the hardest.

Our 6-month program prepares you for your first job interview, helps you build a resume tailored for tech jobs, and prepares you for real-life interview situations (with recruiters, managers, covering technical questions, etc.). Over the course of the program, you will also create a portfolio of work you can present to prospective employers. Now let's examine what happens after you sign up until you land a job.

Roadmap


All you need is the plan, the road map,
and the courage to press on to your destination.



Start Line

Read FelpApp e-book

Decide if this is for you

What do you want to study?

1 week

Join the FelpApp Community

Study with classmates and mentors

Work on multiple projects from Day 1

Grow Community

Build Portfolio

Make a Difference

6 months

CV

Interview Prep

Junior Position

Sorting Hat

The principles remain the same regardless of what cloud provider or a programming language you choose to learn if you are just starting out. As a SOLID FOUNDATION, they will always stand strong.

Building a solid foundation is the first thing you should focus on, as that is the only factor we can rely on in a world that is constantly changing. By mastering the basics, you will have the confidence to explore and be creative. This applies to both engineering and software development.

Therefore, when it comes to choosing the programming language, pick a language, get to know the basics, rules, understand the data structure, variables, commands, and syntax of the language itself. Let's take for example C++, Java, Python, Java Script, etc. The concept is very similar in each of these languages. There are a few things they all have in common:

- **Variables**
- **Mathematical**
- **Logical operations**
- **Input**
- **Output**
- **Conditions**
- **Loops**
- **Functions**

In short, once you've mastered one programming language, you're pretty much 2 weeks away from the other!

Check out this video about
How I would learn to code (If I could start over)

After theory comes practice, practice, and more practice. The next level is to participate in projects, which is exactly what we will be offering you here at **FelpApp**, building your first projects.

It is no different when it comes to cloud computing. As you become proficient with one cloud provider (AWS, Azure, GCP...), you will be able to apply the same principles to the other cloud providers as well.

This video discusses A
WS Vs Azure vs Google (Which Cloud Platform Should I Learn)

FelpApp will cover the following basic divisions (directions) in the Technology Sphere:

- **Front End Development**
- **DevOps Engineering**

These videos will give you an insight into:

- IT or Developer - WhWich is the right Career Choice for you?
- 5 most IN-DEMAND Jobs in 2022
- Top 10 Tech Jobs of 2021

There is also an interesting article on Coursera that we found **7 IT Career Paths and How to Get Started**

If you want to find out if a tech field is a good fit for you, here are some quizzes to try out:

- **What technical field is for me?**
  https://www.techskills.org/careers/quiz/

- **What is the best programming language for me?**
  Taking a test based on your interests is a great place to start, which will generate all the possibilities which are most relevant and suit your profile.
  https://www.bestprogramminglanguagefor.me/

Image by Freepik

Search Engine

Google is your best friend! Therefore, you should always keep asking questions and challenging yourself. Whenever you need someone's help, make sure you google it first! Make that a habit from day one!

In response to people asking us questions they could have Googled themselves, you can sometimes pass them this website called **Let Me Google That For You**.

The website's official title: *"For all those people that find it more convenient to bother you with their question than to google it for themselves."*

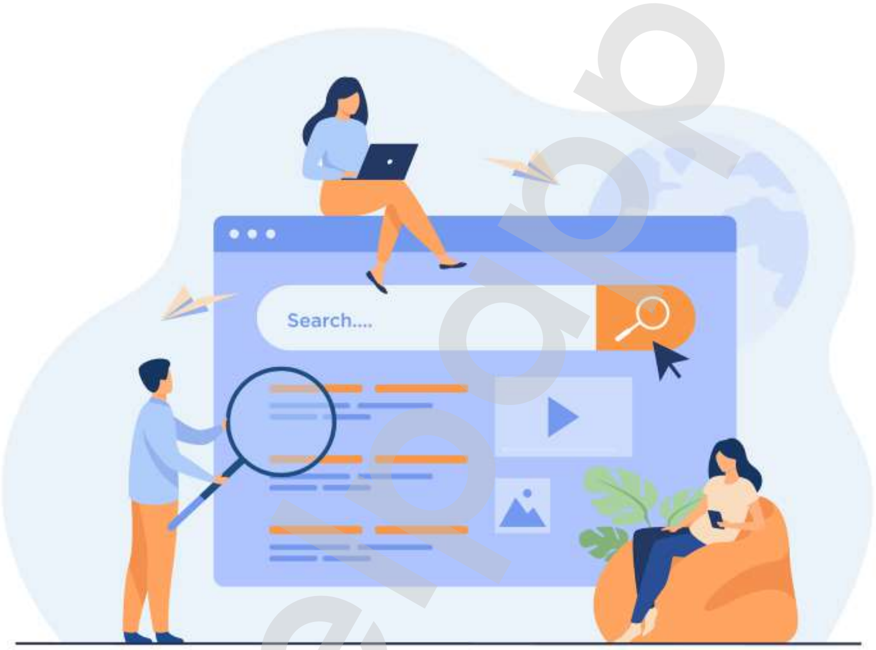


Image by Freepik

Be sure you are in control of your search, understand the problem itself before you start searching, know what resources/words might lead you to the answers you need, know where certain tools and technologies' documentation is located, etc.

A mastery of this skill set will take you far in life, and it is the most important one to learn. Nowadays, everything is available online, Google algorithms are so advanced that you can ask specific questions and get quick answers based on your search.

This is an interesting article about
Why Google is the most important Learning Tool ever invented?

Part III

What is a Digital Nomad?

A digital nomad is one of the new occupations created by the modern age and digitalisation, which did not even exist a few years ago.

Throughout this section, we will try to cover questions such as
who are digital nomads and what they do.

Have you ever dreamed of taking your work anywhere you want? Beach restaurants certainly seem more appealing than dark offices, don't they? It is exactly what every digital nomad desires:

- **Travelling around the world**
- **The ability to earn money through their online business.**

So, the key to turning your world into an exciting workplace is to do this!

Travelling while working is a privilege that digital nomads enjoy since they are not confined to a particular location. If you take the translation literally, a Digital Nomad is always on the move and can make money online through digital platforms. Although it sounds like a professional title, in this case, it is not about the job but the way it is done. To accomplish this, it is necessary to have a job that does not require your physical presence. The nature of the job is such that you can do it with a laptop and an internet connection.

Globally, they can work as freelancers or as full-time employees for various companies under various contracts. The majority of them stay (temporarily) in a certain area for a few months (or a year), depending on their travel visas.

Image by Freepik

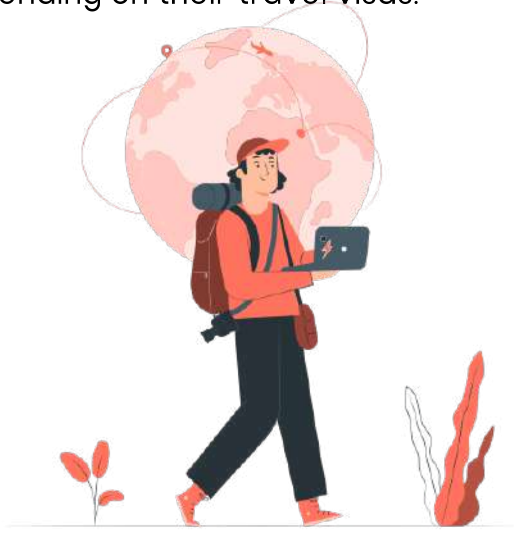


Image by Freepik

Nowadays, many international companies have branches in different countries, which makes it easier for employees to relocate.

The Visa Requirements and Taxes should be considered when deciding whether to go abroad. Please refer to the following link for more information: https://www.safeguardglobal.com/resources/blog/working-remotely-from-another-country

Exactly what do digital nomads do?

Digital nomads are most commonly engaged in modern IT jobs, which include:

- **Programming**
- **DevOps Engineering**
- **Grapic Design**
- **Web Development**
- **App Development**
- **The Internet marketing industry, specifically digital marketing and social media marketing**

The range of what a digital nomad does is very large and includes even more than the usual free nomadic jobs and other occupations. If you are drawn to one of these professions and want to explore the world, then you too can become a Digital Nomad!

I would like to become one. How can I do this?

For you to become one, there is no particular trick or way you have to follow. Our best recommendation would be to try to discover your passion, work on your skill sets, and discover your way of working remotely, which is something we will help you with!

All you need to do is determine what goals you want to achieve and what other passions you may have that will fit nicely into this lifestyle! Perhaps your biggest passion is travel, which will make working from abroad even more exciting!

What are the advantages and disadvantages?

As in any profession, you will always come across some advantages and disadvantages, but there are more advantages to this one (I promise!), and they are:

- **You are your own boss.**

- **Your job is in your suitcase,** and the office is all over the world.

- There is no established everyday life and the so-called working hours are from 9 am to 5 pm.

- A **good earning potential** and the **possibility of continuous improvement.**

- Despite frequent changes of residence, **you will find colleagues virtually everywhere when you live a nomadic lifestyle**. There are many freelancer communities who help each other, so you will have no problem adapting to the new environment. Additionally, meetup.com is available in major cities, encompassing coworking spaces where nomads and freelancers gather. There are many places that offer events such as workstations, vacations, trips for remote teams, conferences, etc. With fun and socialising, nomads share their experiences and knowledge.

- **Living costs** - Nomads often look for places with low living costs. Your earnings will be worth more in Eastern Europe than living in Western countries. As a result, you will have real savings faster, which you can invest wisely in the future.

- **The discovery of new cities, countries, cultures, and the opportunity for a fuller and more meaningful life.** There seems to be no better way to balance work and pleasure than by combining the two. It is possible to travel the world without limiting yourself to the number of vacation days if you know how to successfully solve all business challenges.

There are, however, some drawbacks to this way of life:

- **Issues pertaining to organisational matters,** such as visas, work permits, and paperwork.

- Due to the fact that you will be working a full-time job and may be required to work overtime, **you should choose a location that coincides with the time zone of your company.**

- Despite 2022, **some areas will not have access to the internet.** Because of this, no matter where you are in the world, you cannot guarantee a quality internet connection.

- The simple **distraction of being in a new environment** or having new experiences **can make it difficult to focus on our work.** Digital nomads need to find a rhythm with which to complete their tasks without excessive distraction.

- Although surrounded by new people, people who have replaced the office with some unfamiliar, exotic environment can often **feel nostalgic.**

References:

- How to Become a Digital Nomad
- Pros and Cons
- Remote Business Ideas

Videos:

- Working online and travel
- How to Become a Digital Nomad

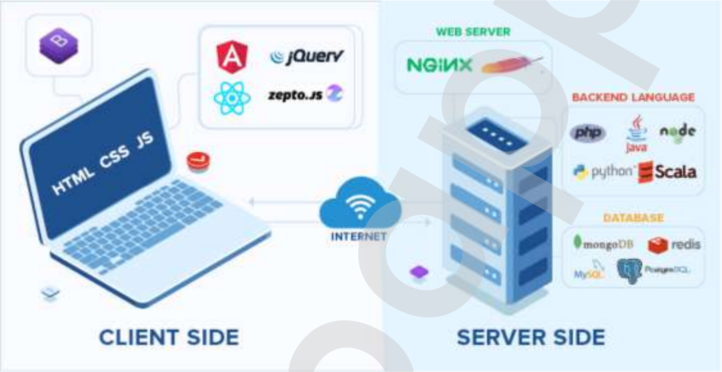
YouTube

Technical Roles

Front End Developer

On the market, there are two types of web developers, namely front-end and back-end developers - depending on whatever they specialise in client-side or server-side development. This version of e-book is primarily geared toward front-end developers. It combines and integrates the topics of web design, web programming, and search engine optimisation.

By the end of the course, students should be able to build an Internet-ready website that meets current/up-to-date requirements. An emphasis is also placed on tools, tricks, and skills that allow the site to appear on the first page of Google during searches through the search engines.



A few key topics will be covered in the e-book:

1) Users interface creation and manitenance:

Front-end development focuses on creating the user experience and interfaces that users interact with on a site or application. Due to this, front-end development is often called "client-side" development since it encompasses everything that happens on a client's device which includes creating static pages that contain elements like buttons, graphics, animations, photos, and navigation. Using the client-server model of communication, your browser acts as a client on the web. When you access websites using your web browser, what you see as text, images and videos is a representation of the conten located on a remote web server. This is what we refer to as a user interface. It is therefore essential to hire a front-end developer who can maintain the design to attract users.

2) Mobile/responsive websites:

Besides designing the appearance of a site, front-end developers are responsible for the visitor's experience i.e. how all components of the site behave, and the ability to develop mobile/responsive websites (cross-platform).

3) Improve the website by fixing bugs and errors:

The work of a front-end developer is often confused with the work of a web designer since front-end developers focus on aesthetics.

In terms of skills and responsibilities, designers and developers share some similarities, but the most striking difference lies in the difference between modelling and implementation.

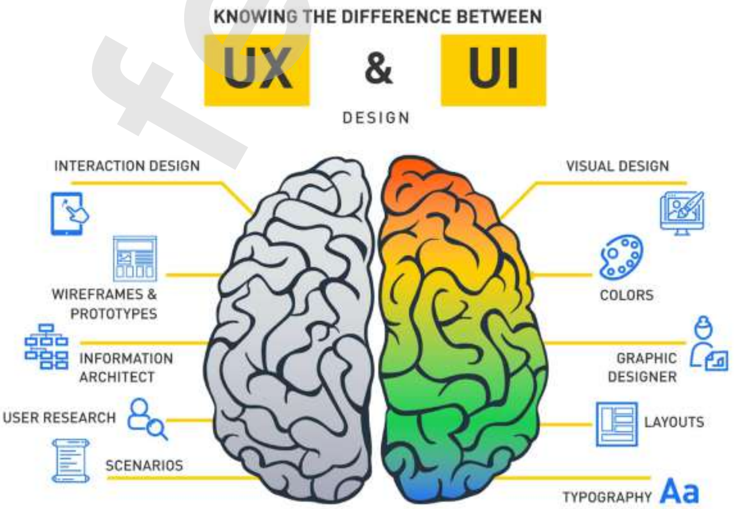
Due to their priamry focus on presentation, front-end developers require artistic vision to effectively present data. In most cases, this includes mastering HTML, CSS, and some JavaScript frameworks such as React, Angular and Vue.

An ideal front-end developer should also be familiar with fixing bugs and errors, as well as platform performance as security.

Despite de importance of all three technologies, front-end engineers choose the one that best fits their preferences. The interests of some developers lie more in designing, while those of others lie more in programming. The HTML and CSS technologies appeal to visual and verbal people, whereas JavaScript appeals to logical people. It's not a stretch to imagine that front-end development could be your future profession if you consider yourself creative.

There are some concepts and roles that are similar to Front-End Developers:

- **A user Experience (UX) designer** is researching and studying how people use websites, they make chnges to the site based on their knowledge and extensive testing.
- **A user Interface (UI) designer** is basicallly a visual designer who specialises in design. Despite not being involved in the implementation of the desing, they should be familiar with HTML and CSS. This will enable them to communicate their ideas clearly to the front-end developer.



Videos:
[What do Front-end Developers actually do]

References:
[Coursera: What Does a Front-End Developer Do?]
[Guru99: Who is a Front-End Developer?]

DevOps Engineer

In this day and age, even those without direct contact with the IT industry have heard of the term **DevOps**. Some people say that a DevOps engineer is a person who unites several IT disciplines and who is in charge of automation, system work, product scaling, service customisation, and publishing of the final product. According to an academic definition of DevOps, it is a set of practices designed to reduce the time between introducing a change to a system and implementing the change in production.

However, no one has yet discovered its exact definition, and there is a constant conflict in discussions when this question is raised. DevOps is a relatively new profession, or rather a term for a group of concepts (or a movement) that have gained much popularity in the tech world over the past decade or so. A general definition for this profession isn't available, but in the IT community, it is commonly stated that it evolved from two trends.

- Firstly, there is "Agile infrastructure", i.e. "Agile operations", since it refers to the use of agile and lean methodologies in business.

- Secondly, **development** (Dev) and **operations** (Ops) began to cooperate. In summary, DevOps refers to the person responsible for resolving technical issues during the entire software development process.

Consequently, DevOps has become increasingly popular in technology circles in recent years and is now an essential subject in any discussion of software development efficiency.
In modern business, there is an unspoken rule that big companies tend to have an advantage over small ones. To this present day, preference is given to those with defined processes and priorities that enable faster product delivery cycles.
Applied to technology companies, speed plays a much more crucial role, so it is no surprise that agile methodologies and automation of various processes are becoming increasingly important in IT companies to respond to customer needs and high demand more quickly.

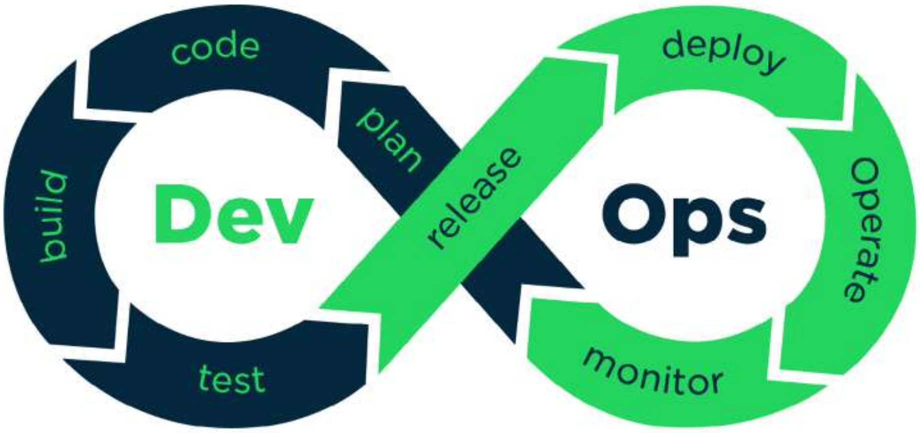


Photo Credit

A number of benefits can be gained from hiring DevOps engineers because they can:

- **Streamline repetitive internal processes by automating them**
  The DevOps engineers reduce the repetitive tasks required to revive a new system or update it. Essentially, they implement solutions that make development easier and faster for developers.

- **Provide assistance in designing a cyber-secure system**
  A DevOps engineer optimises the security of a company's infrastructure by designing cyber-secure systems, updates, and processes. Although all DevOps engineers think about security in their daily work, some (called SecDevOps engineers) are full-time security experts.

- **Enhance the integration and usability of IT systems for companies**
  It's the responsibility of the DevOps engineers to identify and fix infrastructure vulnerabilities. In particular, they help developers improve their workflow, find program flaws, and update designs based on feedback from users.
  By focusing on continuous integration (CI) and continuous delivery (CD), DevOps engineers constantly improve their IT infrastructure.

- **Utilise CI / CD practices**
  - In continuous integration, your code repository is continually updated with new code changes (basically, your code is compiled as you write it).
  - In continuous delivery, code changes are automatically and consistently prepared for implementation (instead of releasing updates all at once, new features are continuously delivered to users).

- **Improve the quality of developers' work and monitor the infrastructure by implementing DevOps tools and technologies**
  By benchmarking the IT infrastructure, DevOps engineers monitor its day-to-day operations. The system is then analysed to identify areas of inefficiency and mitigate potential problems.

- **Project management and planning**
  Their duties include scheduling and leading meetings, setting deadlines, reporting to team members, delegating work, evaluating work, training teammates, and advising on project decisions. As part of the process of designing the system, DevOps engineers also communicate with other teams/ members, management, and developers.

Similar concepts/roles to DevOps:
Platform Engineering, Cloud Engineering and Site Reliability Engineering.

The differences between each of these concepts can be found here:
https://iximiuz.com/en/posts/devops-sre-and-platform-engineering/

Technologies & Tools

In a world where so many new products and applications exist, how can you decide what tools to use? Studies that evaluated the effectiveness of digital tools in enhancing student learning identified three features that all digital tools must possess. These features must provide a quality learning experience. The framework has been designed to evaluate mobile apps, but can be applied to any digital application.

Adaptability - Firstly, the tool must be customisable to meet students' needs. Among the options may be allowing students to customise the way they study the content, such as reading articles or watching videos, or creating an avatar or profile as they progress through the content. Students feel more connected and invested in their work when a personal touch is added. The use of digital tools should be tailored to the individual, not one size fits all. In order to create an enjoyable and effective learning experience, students need to be able to curate it themselves.

Integrity - Furthermore, students should consider whether the tool can help them make connections between learning experiences and real-life situations. It is more likely that students will value and understand information and concepts when they can identify how it will benefit them in the future. The use of digital tools can assist students in connecting the dots when they are integrated into a lesson plan or assignment that simulates real-world situations.

Teamwork - Finally, consider using a digital tool that allows students to collaborate on creating and discussing content. There is a temptation to use technology for its own sake. We can, however, enhance students' overall learning experience by effectively using technology by selecting tools that are both effective and comprehensive.
Through receiving and providing feedback, students can ensure that they do not walk away with misconceptions or gaps in knowledge. As a result, students feel involved in an online class and can establish a sense of community. Building social connections enables students to become more invested in what they learn.

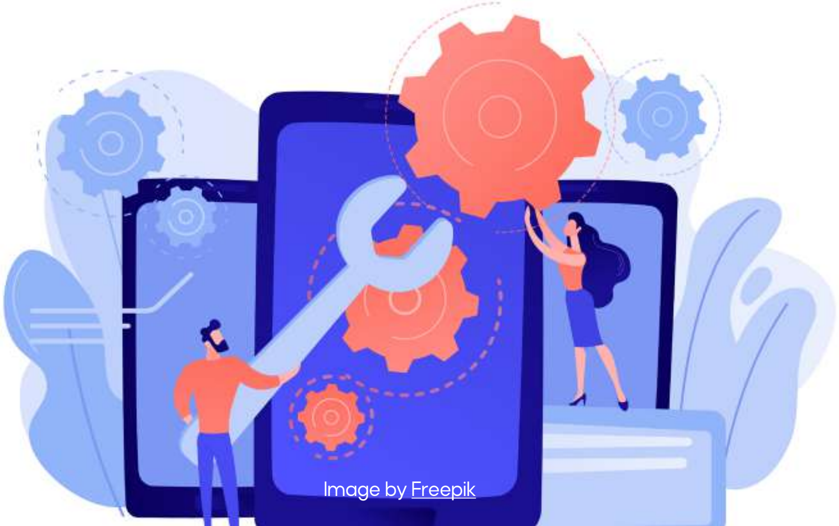


Image by Freepik

Today, Software Development and DevOps use a wide variety of tools and technologies. Throughout your 6 month training, you will be immersed in these tools and technologies.

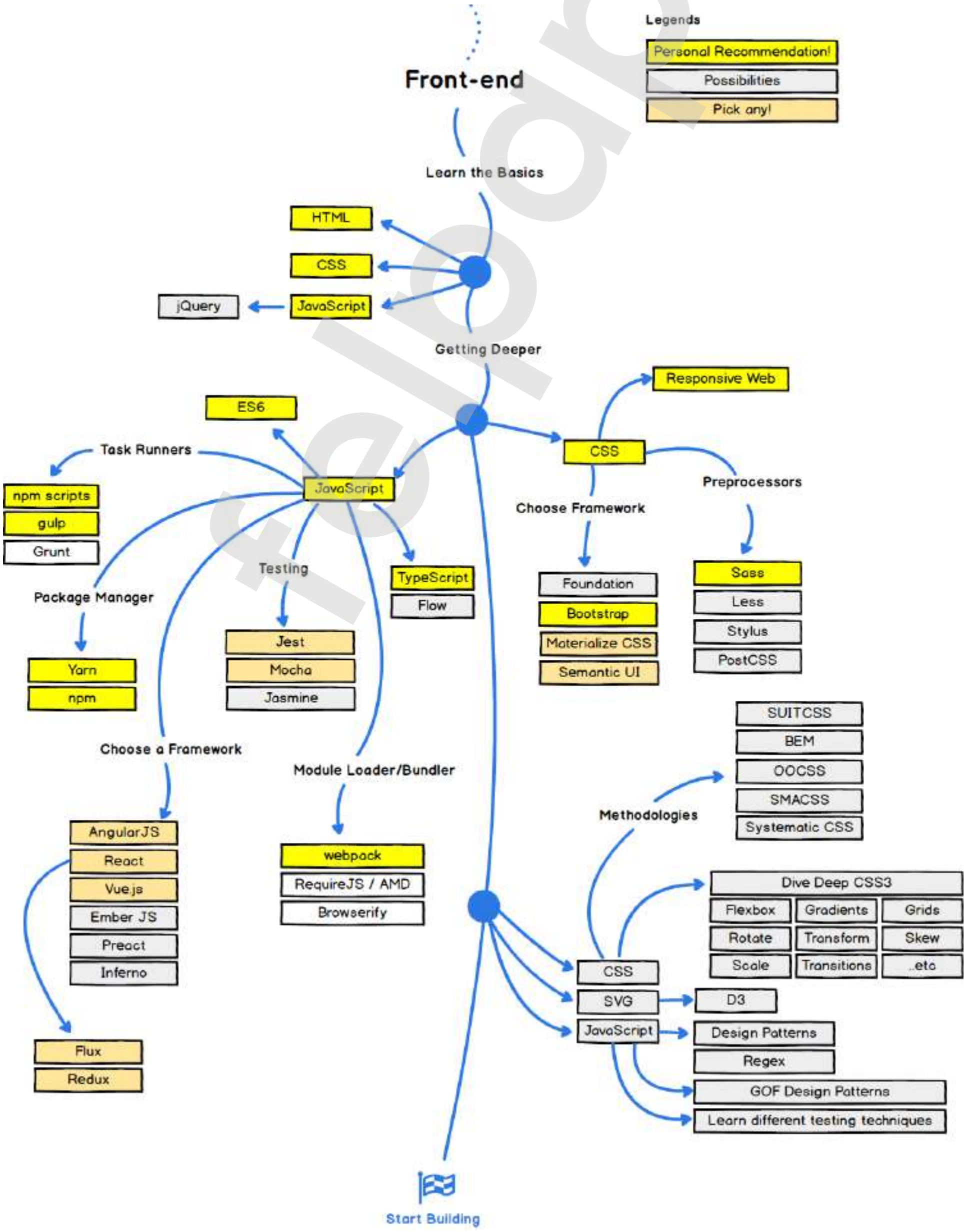
The following videos cover each of the Career Roadmaps in detail, with toolsets and technologies:

- The Fastest Path Into Software Engineering - Career Change 2022
- DevOps Roadmap 2022 - How to become a DevOps Engineer? What is DevOps?

Check out this website for different career roadmaps & toolsets:

- Front-End
- DevOps

1. Front-End Development



FRONT-END DEVELOPMENT | HTML

HTML may be a new concept to you. It is a simple component of WWW technology, yet a vital part of the Internet we know today. The existence of even the simplest website would not be possible without HTML code. Thus, in the following text, we will explain what HTML is.

CHALLENGE 1 | Day 0-30

How do you begin transferring a design to a website interface with HTML?

▶ FelpApp Video - Coming soon...

The first 30 days of your training should be dedicated to learning how websites operate inside browsers. Students often ask, “How can I turn my design into an interactive website?” This is where understanding the components of web browsers (such as Google Chrome, Safari, Spartan, Opera, etc) is key. Usually, they are built using a markup language (HTML), a style sheet language (CSS), and a programming language (JavaScript).

Image by Freepik

As a matter of fact, we will begin our discussion with HTML, which is an acronym for Hypertext Markup Language. As you may already know, HTML is not a programming language per se, but rather a simple language used to create the structure of web pages. Learning and understanding HTML is therefore a simple process, ans a solid understanding of HTML is the foundation of developing websites.

HTML has simple tags that are case-insensitive, which is why it is so easy to read. Due to its simplicity, people are able to understand and modify other people's codes effortlessly. As opposed to other programming languages, it does not cause errors if a developer forgets to close some tags or makes some mistakes in the code.

Today, HTML is mostly used in the construction of the "skeleton" of a website. Our interest hereis in what's "under the hood", in therms of the formatting of text, the addition of photos and links, as well as the presentation of audio and video files with tags. Typically, when an Internet user visits a website, they see a graphical interface, which makes it easier to search for the information that they're looking for on the website. Web developers are tagging text files so they can be used on a webpage to create font, colour, graphics, hyoperlinks effects, etc. To put it simply, HTML allows web developers to display all of that information on the screen in a seamless manner. Through the use of these components, information that is available online can be accessed easily and instantly by the internet users, making their search even more efficent.

Image by Freepik

Tags are therefore created to serve a particular purpose. To see HTML tags with examples, please click here: HTML Tags List With Examples

The tags in an HTML document are generally opened with <tag_name> and closed with </tag_name> (e.g. <body>some web page content</body>). The following are some basic HTML tags:

- **<html></html>** - the main tag in which all other tags are placed
- **<head></head>** - header tag in which the tags are:
- **<title>**Name of the website**</title>** - defines the name of the website
- **<meta name="description" content="Some description of the website" />** - there are many types of meta tags, and they are used to help internet search engines with indexed content, formatting of display on social networks, etc.
- **<body></body>** - contains all the elements and tags that the user sees when he opens the web page
- **<p>**denotes a paragraph - paragraph**</p>**
- **<a href=**"[URL]">Link**</a>**
- **<b>**bold text**</b>**
- **<i>**italics**</i>**
- **<img src=**"[IMAGE_URL]" **/>** - displays the image

To see examples of famous HTML-only (no CSS) websites please click on the this link: **10 famous HTML-only website examples**
HTML files can also be viewed with a web browser. If you are interested in how to do this please check this link out: **W3Schools HTML Editors**
Start by signing up at **FreeCodeCamp: (New) Responsive Web Design** for the exercises that will teach you the basics of HTML, together with the **FelpApp** platform and our community.

Stay on top of this challenge by checking out FelpApp for the latest videos and references.

Ideally, you should start slowly and spend at least 1 hour learning every day. The references below are there to support your studying, in the best possible way. The references below are there to help you study as effectively as possible. The content has been chosen to lay out your fundamentals from the beginning until the point you are ready to start working on your first project within the next 30 days.

🎥 Videos:

- Programming with Mosh: HTML Tutorial for Beginners: HTML Crash Course
- FreeCodeCamp: HTML Full Course - Build a Website Tutorial
- Harvard's CS50: HTTP, HTML, CSS - Intro to Computer Science

🏁 Get Started:

- Learn HTML – step by step study plan and resources
- FreeCodeCamp: (New) Responsive Web Design

📎 Documentation:

- DevDocs: HTML Reference (Syntax, Examples, Indexes)
- W3Schools - HTML Element Reference
- HTML For Beginners The Easy Way: Start Learning HTML & CSS Today

📖 Books:

- 10+ Best HTML/CSS Books for Beginners and Advanced Coders

💡 Cheat Sheet:

- HTML Interactive Cheat Sheet
- HTML Cheat Sheet

🔎 References:

- HTML Beginner Tutorial
- What Is HTML? Hypertext Markup Language Basics Explained

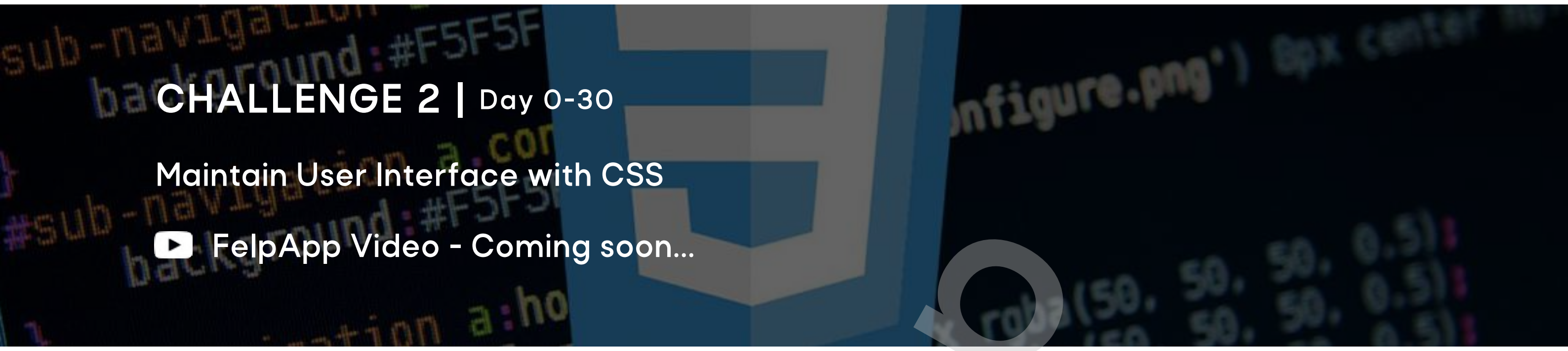
Web development editors:

1. **Choose your editor:** You can find the best HTML editors at this link
2. **How to install the editor on your computer:** ▶ FelpApp Video - Coming soon
3. **Utilise shortcuts to accelerate your development:** Here's a fun article on How keyboard shortcuts could revive America's economy

Function	Command
Search entire computer	cmd + space
Delete an entire line of text	cmd + backspace
Skip forward a word	option + ➡️
Skip back a word	option + ⬅️
Go to top of file/page	cmd + ⬆️
Go to bottom of file/page	cmd + ⬇️

Select all text	cmd + a
Stop a process in terminal	cmd + c
Switch apps	cmd + tab
Quit app	cmd + q
Open new tab	cmd + t
Close tab	cmd + w
Switch tab	cmd + tab
Open Incognito Window	cmd + shift + n
Switch between full screen windows	cmd + ⬅️ (or cmdl + ➡️)
Enter/exit full screen	cmd + ctrl + f
Find text in a file/page	cmd + f
Open console/inspector in browser	cmd + option + i
Create a new file	cmd + n
Save a file	cmd + s
Refresh a website	cmd + r
Select search bar in browser	cmd + l
Search all files in VSCode	cmd + shift + f
Toggle sidebar in VSCode	cmd + b
Open file tree in VSCode	cmd + shift + e
Open file in VSCode	cmd + p
Go to last tab	cmd + 9
Go to first tab	cmd + 1
Take a screenshot	cmd + 4
Take a screenshot that you can paste (e.g. into Messages or Slack) but does not save to the computer	cmd + shift + 4
Record screen	cmd + 5
Select all instances of a highlighted in VSCode	cmd + shift + d
Highlight word (cursor after word)	shift + option + --⬅️
Higlight word (cursor before word)	shift + option + --➡️
Open file navitagor in VSCode	cmd + shift + e

This role was originally taken over by HTML itself. As websites grew more complex over time, the code became more difficult to read and maintain. It became apparent that a new method was needed to solve the problem. In the mid-1990s, CSS, which was then known as CHSS, provided the solution.



CHALLENGE 2 | Day 0-30

Maintain User Interface with CSS

▶ FelpApp Video - Coming soon...

This language contains coding elements used for formatting web pages such as HTML or XML, and is composed of “Cascading Style Sheets,” which are also called CSS files.

Even though CSS differs greatly from HTML, it is unable to function without it. Despite their differences, HTML and CSS are interdependent. By creating page elements in HTML, and styling them in CSS, the desired appearance and function can be achieved. Since CSS is able to adapt to any screen resolution, it allows viewing websites on mobile devices of any size. The CSS styles can be placed in a separate (external) file (with the extension .css) or on the html/php page itself (internal). The CSS code is typically located in a separate/external file that is then included where necessary. This is also the preferred method of working with CSS. You can also use CSS directly in the HTML file (internally) where the structure is defined. An internal style can be defined in the page header (within the head tag, embedded), or in the HTML element itself (Inline).

For more on this please click on the following link Types of CSS: Inline, External and Internal Definitions and Differences Explained as well as this one from FreeCodeCamp Learn CSS in 5 minutes - A tutorial for beginners.

Also check out What Folder Structure should a Website have?

CSS is often referred to as the "design" of a website. In spite of the fact that it's not entirely true, it's not far off the mark either. By using CSS, you can change the colour of text, font style, spacing between paragraphs, and size and layout of columns of HTML elements on a web page, regardless of their order in the HTML file. You can, for example, make the element at the very end of the HTML structure the first visually, and move it up, down, left or right as necessary. We did not change the HTML page structure with CSS; it remained the same, but the visual representation, for the viewer, however, is different.

Image by Freepik

CSS saves time since you can use the same CSS sheet on multiple HTML pages, which is great for code recycling. In other words, if you want to have a website with an appealing appearance, you need to know CSS. Without CSS, modern websites would appear very simple and would be difficult to use. Check out this link to see the difference between HTML-only and HTML+CSS websites: [here]

Image by Freepik

CHALLENGE 3 | Day 30-60

Implement theory into practice, build a mobile/responsive website as a part of your portoflio using HTML & CSS!

▶ FelpApp Video - Coming soon...

Hopefully now you have a better understanding of HTML & CSS. The current versions that are recommended for use are HTML5 and CSS3. These standards resulted from years of development and often turbulent events in the tech community. There are still plenty of resources at FreeCodeCamp: (New) Responsive Web Design to help you continue to advance your HTML skills as well as learn CSS in combination with it.

Stay on top of this challenge by checking out FelpApp for the latest videos and references.

During the following days of this challenge, you will gain a basic understanding of how to build a website using HTML and CSS, and in truth, these are the only tools you need. Practising HTML & CSS for 30 days will prepare you for some front-end jobs.
Upon completion of Challenge 3, we will begin reviewing a few of the following items as a means of preparing you for some of the next challenges designed to advance your skill set.
With the combination of simple elements and respect for the clear rules that are woven into the foundation of HTML and CSS, we can achieve great results. There are, however, some limitations to these languages. Their descriptive nature restricts them from helping us with the details that need to be programmed. Because of this, most websites today use JavaScript and other programming languages in addition to HTML and CSS. It's not uncommon for HTML, CSS, and JavaScript to be used simultaneously on a website. As a result, they don't compete with one another but work together as a complementary team.

In light of this, we will cover JavaScript as our next topic.

🎥 Videos:

- FreeCodeCamp: CSS Tutorial - Zero to Hero (Complete Course)
- Harvard's CS50: HTTP, HTML, CSS: Intro to Computer Science

🏁 Get Started:

- FreeCodeCamp: (New) Responsive Web Design

📎 **Documentation:**

- DevDocs: CSS Reference (Syntax, Examples, Indexes)
- W3Schools: CSS Reference
- Mozilla: CSS - Cascading Style Sheets

📖 **Books:**

- Best Books To Learn CSS & CSS3 in 2022

💡 **Cheat Sheet:**

- CSS Interactive Cheat Sheet
- CSS Cheat Sheet

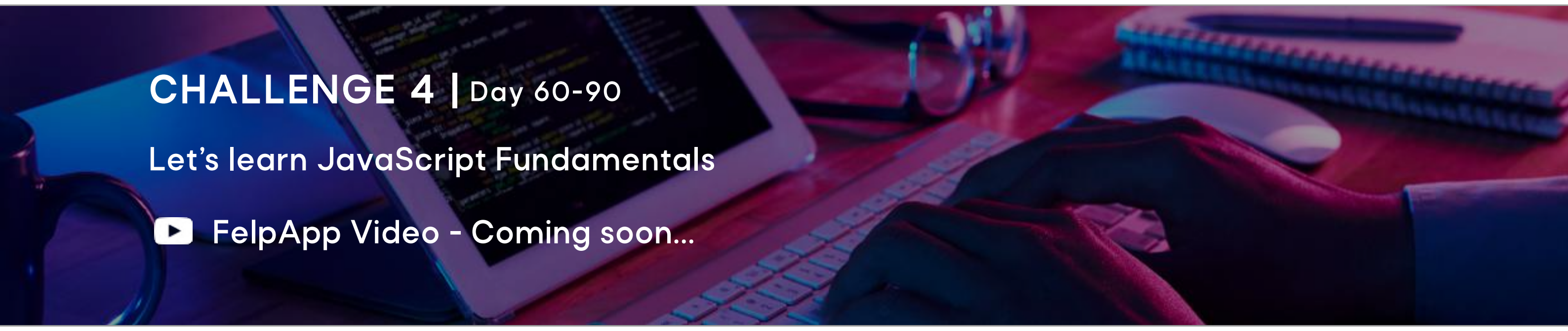
🔎 **References:**

- Mozilla - Styling the web - What is CSS?
- The Ultimate CSS Tutorial for Beginner Programmers

FRONT-END DEVELOPMENT | JavaScript

In our previous chapters, we saw that HTML (the basic code of web pages) allows us to design and edit page elements, but not their behaviour. It is true that HTML and CSS provide the structure and style to web pages, but JavaScript adds interactive elements that make them appealing to users. It is for this reason that JavaScript is integrated into the website to make it more dynamic and user-friendly. By incorporating JavaScript, we can also have more control over how the website behaves. HTML and CSS combined with JavaScript create dynamically generated HTML (DHTML).

Image by Freepik



CHALLENGE 4 | Day 60-90

Let’s learn JavaScript Fundamentals

▶ FelpApp Video - Coming soon...

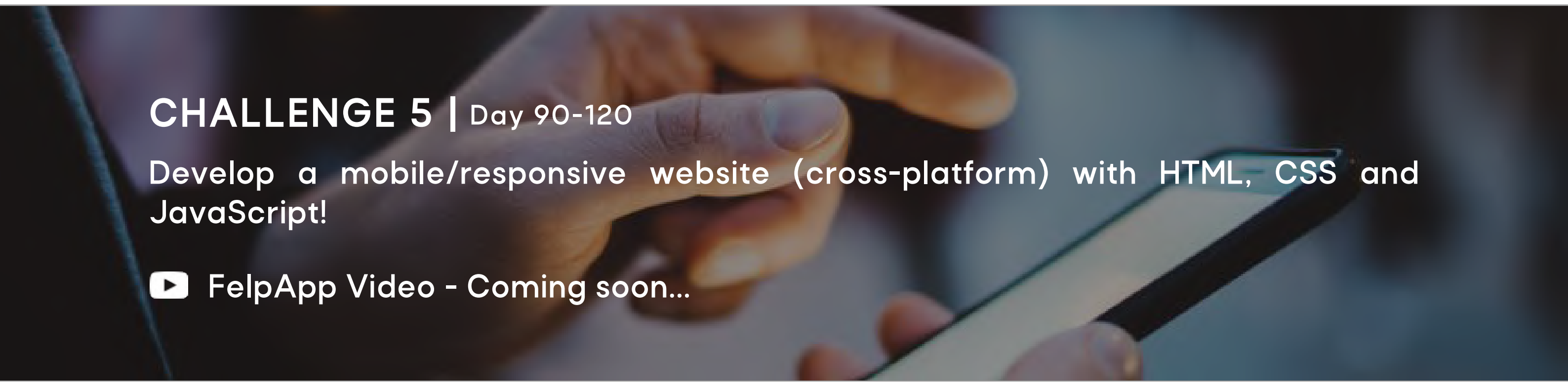
JavaScript is a **text-based programming language** used both on the client-side and server-side that allows you to make web pages interactive. Because it consists of a sequence of commands, the content can be read directly from an interpreter, without needing to be compiled beforehand. Which makes it one of the most popular scripting languages used today, and its wide support by most popular browsers (Mozilla Firefox, Opera, Internet Explorer, etc.) makes it particularly useful. So, instead of translating it into machine language (binary code - 0’s and 1’s), the commands are read directly from the code (source code or bytecode). Due to this feature, JavaScript is executed on the client side, i.e. on a computer running JavaScript content. The code can, however, only be executed by a client application. Since all the data is stored locally, the script will run even without an internet connection. As opposed to Java, which is purely server-oriented, this is a significant difference. By default, HTML allows users to send data directly to the server. A valid set of data must be entered first, and if such data is not present, the entire process must be repeated until valid data is entered. **This is one of the main reasons JavaScript was created**, which checks the credibility of data in the client’s browser and thus facilitates web development. Browsers such as those we mentioned earlier are used for this purpose. In such a case, the client decides whether to execute the code or not.

Furthermore, besides its popularity on the web, it has evolved into many frameworks and libraries on the front-end, such as **React**, Vue, Angular, Ember, and jQuery. There is no doubt that JavaScript reigns supreme on the web - It is the king of the web. Especially if you’re just getting started, you should learn JavaScript before learning any other programming language.

Due to its other specific features, JavaScript is commonly used to create web pages. In comparison with some other programming languages, it offers great advantages:

- **High level of interactivity**: Users can activate content by moving their mouse over it or pressing a button on the keyboard;
- **The ability to work locally:** Because the language focuses on the client, communication with the server is minimised;
- **Improved interface:** Facilitates complex page structuring and the execution of specific commands, such as dragging content;
- **Feedback is instant:** No reloading is necessary to see the change.

The time has come for you to learn your first programming language.
Isn’t that exciting?! :)



CHALLENGE 5 | Day 90-120

Develop a mobile/responsive website (cross-platform) with HTML, CSS and JavaScript!

▶ FelpApp Video - Coming soon...

The fun part of this module is that you can now combine your previous knowledge of HTML, CSS, and JS together to build unique websites. Therefore, our page is now ready for some interaction.

Interaction? - As you already know with HTML and CSS, we went through tasks of building a static web page. It is unfortunate that this web page itself is not able to do much as it is now. However, with the power of JavaScript(JS), we can enhance the experience by making interactions between the user and the web page. If, for example, one wants to click a button on a web page, that button has to provide you with some information on that page. It also has to populate it with all the new styles and text as a result of the button being clicked. This is what object-based programming is. Often characterised by the fact that the programmer defines not only the data type, but also the type of operations (functions) that can be applied to the data structure. Data structures are thus transformed into objects containing both data and functions. Additionally, developers can create relationships between objects. Objects can, for instance, acquire characteristics from other objects.

Our recommendations for practising and learning JavaScript are to start with **FreeCodeCamp: JavaScript Algorithms and Data Structures**, alongside with FelpApp, and ENKI. This challenge involves completing all exercises in the **Eloquent JS book** and **FreeCodeCamp: JavaScript Algorithms and Data Structures** training in order to take your skills to the next level. Although the dare will definitely be there, if you do this honestly and begin learning JavaScript with just HTML and CSS you will be able to apply for junior positions in the front-end world. During the next 30 days, try to devote a significant amount of time to learning and practising JavaScript. For you, this will lead to great success.

Stay on top of this challenge by checking out FelpApp for the latest videos and references.

In the next chapter, we will introduce you to one of the most popular JavaScript frameworks used today - ReactJS, which will assist you in expanding your skill set and discovering new areas where you will be able to contribute and expand.

🎥 **Videos:**

- Harvard’s CS50: Computational Thinking & Scratch - Intro to Computer Science
- Programming with Mosh: JavaScript Tutorial for Beginners: Learn JavaScript in 1 Hour
- FreeCodeCamp: Learn JavaScript - Full Course for Beginners

🏁 **Get Started:**

- FreeCodeCamp: JavaScript Algorithms and Data Structures
- Mozilla: JavaScript Basics
- Enki: Learn Coding/Programming

🖇 **Documentation:**

- DevDocs: JavaScript Reference
- JavaScript Official Documentation

📖 **Books:**

- 10 Best javascript books for 2022

💡 **Cheat Sheet:**

- JavaScript Interactive Cheat Sheet

🔎 **References:**

- freeCodeCamp: Learn JavaScript – Free JS Courses for Beginners
- Tutorialspoint: Javascript Tutorial
- Guru99: JavaScript Tutorial for Beginners: Learn Javascript Step by Step

CHALLENGE 6 | Day 120-150

A Deep dive into one of the most famous JavaScript's libraries.

▶ FelpApp Video - Coming soon...

Your hard work has paid off! Now that you've built your first website, let's make it more fun and accessible by making it more interactive. :)
With your understanding of Vanilla Javascript, HTML, and CSS, now it's time to explore Frameworks, learn why they're important, and which one to learn first.

First, let's answer the obvious question - What is a framework?

Web development relies heavily on frontend frameworks and libraries. It has become imperative for developers to make use of libraries in order to build high-performing responsive websites and web-based applications. A framework is a collection of tools that is used by many successful coding teams as their starting point. Scripting and programming are much faster and more efficient with these platforms because they enable libraries of bundled code, software modelling, APIs, and other features. As the digital world moves at an exponential rate, new libraries are released every year. Choosing the right library or framework for each project can be confusing. Please do not worry if you still do not understand what a framework means. In a nutshell: **Frameworks help you build your website quickly with pre-written code.**
Rather than writing all the code from scratch, for instance, if you wish to develop an image slider, you can import the framework's set of tools and get it up and running in minutes.
You're probably wondering how you can just add those tools to your development? This is an excellent question to ask! It is for this reason that you should learn the framework you intend to use. Front End Developers usually work with one or three popular frameworks: React, Angular, or Vue, though other frameworks and libraries such as JQuery and MeteorJS are also commonly used.

How do you decide which one to start learning first?

React is a good place to start. One of the main reasons why you should use React is because you will hear the term "With REACT, you are writing pure JavaScript". The React library is forcing you to learn and write JavaScript. It is the most popular framework and is used by most companies. Additionally, if you learn React, you can quickly move to mobile development since you can convert your knowledge to

React Native. Mobile apps can be built using React Native on iOS and Android. The ReactJS framework gives developers complete flexibility. You can use it to create SPAs, mobile or hybrid apps, even a TV application. ReactJS will ensure your app performance is optimised. You can add as many external libraries and tools as required, creating a massive, complicated web application. A major reason to choose React for frontend development is its strong community support. Since React is an open-source library, a large community of developers is making it better. Coders from around the world help people learn the technology in different ways.

Exactly what is React?

React (also React.js or ReactJS) is a free and open-source JavaScript library for building user interfaces based on UI components. Using this library, you can write and reuse UI components in your projects. By using this approach, you will be able to build your web applications quickly.

Perhaps you don't understand the concept of UI components? That's what we'll talk about next.

Apps and websites are built with user interface (UI) elements. They provide touchpoints for the user as they navigate around the user interface; think buttons, scrollbars, menu items, and checkboxes.

What is the best way to build UI elements using React?

Components are smaller building blocks that make up user interfaces. Using components, you can build self-contained, reusable code snippets. If you think of components as LEGO bricks, you can take these individual blocks and combine them to form larger structures. In order to update a piece of the UI, you can update specific components or bricks.

With modularity, you can easily add, update, and delete components without affecting the rest of our application as it grows. The great thing about React components is that they're just JavaScript. The following example shows how a JavaScript component can be written for React. A website can be divided into multiple UI elements. A number of UI elements can be developed once and reused later in the program, such as: Header, Footer, NavBar, Slider, Contact Forms, Carousel, Image Gallery, etc. As soon as you have your components ready, you can use them on different pages, which will speed up your development and keep your web app consistent. There might be a time in the future when you need to create a new page for your web app. Since you already have pre-built components, it will be super easy. Additionally, React allows you to control your data or insert data dynamically if some components have different data/content on other pages. Wow, that's awesome! Based on where our UI element resides in our app, we add data after building UI elements.

What is the purpose of doing that? You raise an excellent question. As an example, you build a slider component and want to use it on different pages, such as About and Home. You don't want the same images/slides on both pages, nor do you want to build two identical sliders with different data/images. On other pages, you want to pass additional data to the same slider.

What are the methods for passing data to UI elements and components?

You will hear the term 'props' a lot in React. 'Props' is the term for properties and the data you pass to components are referred to as 'props'. Understanding this is probably one of the most important aspects of React. Since this is a guide on becoming a great developer, we won't get into much detail, but we'd like you to spend some time understanding React's data flow.

There is, however, a difference between 'state' and 'props' that needs to be understood.

The parent's data is known as props. Data can only be transferred from parent to child, not vice versa, so child components cannot update or modify the data independently.

How do child components differ from parent components? Different UI elements can be contained within one UI element/component. Nav bars can, for example, be included in the Header. Here, the Header is the parent component, and the navigation bar is the child component. It is possible to transfer some data from the Header to the Nav Bar, but not the other way around. Hopefully you now have a clearer understanding of the parent and child components.

What is the reason for only being able to pass data from a parent to a child?

A simple, predictable app is one you can control completely. It's important not to rush here. The data flow in React must be understood and mastered. This period should be used to play and practise a lot of passing data and building components so that you become proficient. Please do not hesitate to contact the FelpApp team if you have any questions.

The Advanced Course: React & APIs

How do React hooks and lifecycles work?

The lifecycle of a React Component can be defined as a series of methods that are invoked at different stages of the component's lifecycle.

Imagine that a visitor on your website is reading a fantastic post that you wrote. However, you want to give them an opportunity to write some comments, and you want those comments displayed on that component. Additionally, you want to be able to answer the comment and make it visible on the component. In this way, you are giving the user a chance to interact with you. It might still be a mystery to you how an app's lifecycle works. Don't worry, we'll break it all down for you. Your app/component's lifecycle means you can update the data during user interactions. Initially, there is only text, then users can add a comment, and that comment appears on the screen; you can update the same component with your comment while visiting the app. Despite the fact that your element is loaded on the screen, we edited it multiple times after the first load. Updates during that period are referred to as the lifecycle.

A React application's State and Lifecycle methods are its foundation, and Hooks enable functional components to implement them efficiently. As hooks don't function inside classes, you can use state and other React features without writing a class.

In simple terms, what is a hook?

React state and lifecycle features can be accessed from function components using hooks.
During the life cycle of the app, React has written methods that let you control your data.
There are four main points you need to understand in depth:
* useState
* useEffect
* useCallback
* useContext
Furthermore, you need to understand: useRef, useMemo.
More hooks will be taught to you, but those six will be used on a daily basis.

Put in a lot of practice. Consider using APIs from big companies to build something. Create a Twitter Presenter app. Use the Twitter API to present tweets in real time on your web application. You have all the knowledge you need. By now you are familiar with how components are built and how data is passed to UI components. The only thing you need to do is call API, collect data from Twitter, and present it to your app; also, you need to update your app whenever a new Tweet is posted. More on this will be covered in our bonus challenge: Challenge 7. For more information, please see below.

Image by Freepik

🎥 Videos:

- FreeCodeCamp: React Course - Beginner's Tutorial for React JavaScript Library
- JavaScript Mastery: React JS Full Course 2022 | Build an App and Master React in 1 Hour

🏁 Get Started:

- FreeCodeCamp: Front End Development Libraries
- FreeCodeCamp: How to Get Started With React – A Beginner's Guide
- Medium: Beginners Guide to ReactJS
- React: Create a New React App
- Guru99: ReactJS Tutorial for Beginners: Learn with Step by Step Example
- Mozilla - Getting started with React

🖇 Documentation:

- DevDocs: React Reference
- Official React Documentation - A JavaScript library for building user interfaces
- W3Schools - React Introduction

📖 Books:

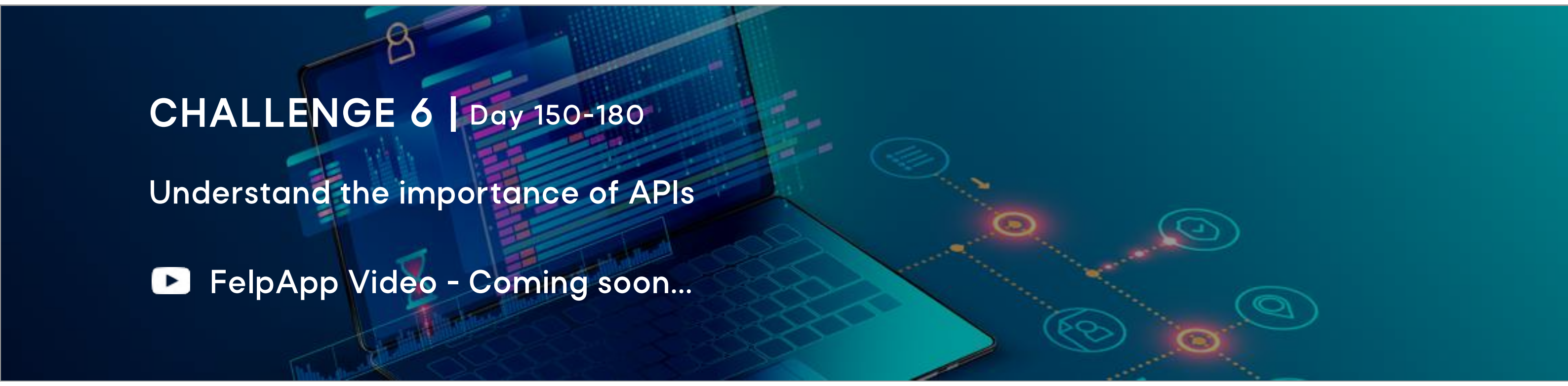
- 6 Best React.js Books for Beginners and Experienced Web Developers

💡 Cheat Sheet:

- React.js cheatsheet
- FreeCodeCamp: The React Cheatsheet for 2022

🔎 References:

- React Tutorial: Intro to React
- FreeCodeCamp: React for Beginners – A React.js Handbook for Front End Developers



CHALLENGE 6 | Day 150-180

Understand the importance of APIs

FelpApp Video - Coming soon...

APIs are one of the most important things you need to learn. It stands for Application Programming Interface.

What does this mean? Why is it so vital to know?

APIs are essential tools for businesses across all industries. APIs are important from a technical perspective since they enable one computer program to access the capabilities of another. In other words, they allow two different programs to communicate with each other.

Let's say you are building a crypto project and need a list of all the crypto prices. Rather than creating your own list of crypto prices, you can use someone else's system to present that information on your website. This information can be retrieved using their API and you can have it on your web app within a few hours or days. Sometimes you need to pay for those services, but a lot of them are free. Almost all big brands have APIs/services you can integrate into your web application. In order to be a successful web developer, you will have to know and understand this. They will need to be used or even built by you. Communication between different software and systems is standard. In the future, you may work for a company that makes partnerships with other companies. It will be necessary to integrate the software systems and communicate between them as part of those partnerships. In order to accomplish those things, you need to build or use APIs.

APIs - what are they?

Through APIs, two software components can communicate with each other using definitions and protocols. For example, your phone's weather app "talks" to the weather bureau's software system via APIs and displays daily weather updates based on the bureau's weather data. If you are developing software and want to integrate face recognition or payment processing, for instance, using APIs is the

eferred solution.
Becoming API-aware is essential if you want to define the future of business. In the next decade, APIs will play a crucial role in determining almost every aspect of how you conduct business.

Before learning APIs, you should be familiar with the following:

- Prerequisites: You must have basic computer literacy, knowledge of HTML and CSS, and a basic understanding of JavaScript.

- The first thing we need to do is understand what REST APIs are.

The REST API (also called RESTful API) is an application programming interface (API) that is compatible with REST architecture and can be used to interact with RESTful web services. The REST acronym stands for representational state transfer and was created by Roy Fielding.

- REST API operations: what are the most common?

In order to communicate using REST, HTTP verbs such as **POST, GET, PUT**, and **DELETE** are commonly used. These are known as CRUD operations (**C**reate, **R**ead, **U**pdate, **D**elete). Several other verbs are also used, but they are less common. REST is an easy to learn technology because these methods are quite self-explanatory.

- POST:

The POST verb is most commonly used to **create** new resources. Specifically, it's used to create subordinate resources that are subordinate to other resources (e.g., parents). As a result, when creating a new resource, you POST to the parent, and the service will assign the ID (new resource URI) and associate the new resource with the parent.

- GET:

HTTP GET is used to **read** (or retrieve) a resource representation. When GET returns a "happy" (or error-free) representation in XML or JSON, it returns 200 (OK). When there is an error, it most often returns a 404 (NOT FOUND) or 400 (BAD REQUEST).
A GET request (along with a HEAD request) can only be used to read data, not to change it. This makes them safe when used in this way. As a result, they can be called repeatedly or once without resulting in data loss or corruption. Furthermore, GET (and HEAD) are idempotent, meaning making multiple identical requests has the same result as making a single request.

- PUT:

The most common use of PUT is to **update** a resource, PUT-ing to the original URI with the newly-updated representation of the resource in the request body.

The client can choose the resource ID instead of the server when using PUT to create a resource. For example, if the PUT contains a nonexistent resource ID as the value of the URI. Many find the resource representation in the body of the request to be confusing and convoluted. Consequently, this creation method should be used sparingly, if at all.

Alternatively, use POST to create new resources and provide a client-defined ID in the body representation-presumably to a URI without the resource ID (see POST below).

A PUT that returns 200 (or 204 if there is no content in the body) indicates that the update has been successful. If the PUT was successful, the HTTP status code is 201. The body of a response is optional, but it consumes more bandwidth than an empty response. Since the resource ID has already been set by the client, it is unnecessary to return a link via the Location header in the creation case.

The PUT operation is not a safe operation because it modifies (or creates) a state on the server, but it is idempotent. If you create or update a resource using PUT and then make the same call again, the resource is still there and has the same state as before. If, for instance, calling PUT on a resource increments a counter within the resource, it is no longer idempotent. When that happens, it may suffice to demonstrate the call is not idempotent. PUT requests should be kept idempotent, and non-idempotent recommendations should be sent through POST.

- PATCH:

The patch is used to ****modify**** capabilities. The PATCH request should only contain the changes to the resource, not the entire resource.

In contrast to PUT, this request contains instructions describing how to modify a current resource on the server to produce a new version. In other words, the PATCH body should not only be an edited part of the resource, but also in a patch language such as JSON Patch or XML Patch.

PATCH is neither idempotent nor safe. It is possible to issue a PATCH request in a way that makes it idempotent. This helps prevent unintended outcomes from collisions between two PATCH requests in a similar time period. There is a higher risk of resource corruption when multiple PATCH requests occur because some patch formats operate from a known base point. This patch application should be used with a conditional request such that the client won't be able to access the resource if the resource has been updated since they last accessed it. The client can, for example, use a strong ETag in the If-Match header when submitting a PATCH request.

- DELETE:

DELETE is pretty easy to understand, and it is used to **delete** a resource identified by a URI. Upon successful deletion, return HTTP status 200 (OK) along with a response body, perhaps a representation of the deleted item (demands too much bandwidth), or a wrapped response (see Return Values below). It's either that or return HTTP status 204 (NO CONTENT) with no response body. Therefore, a 204 status with no body or a JSEND-style response and HTTP status 200 are recommended.

According to the HTTP specification, DELETE operations are idempotent. When you DELETE a resource, it is removed. Calling DELETE repeatedly on that resource ends up with the same result: it is gone. The DELETE call is no longer idempotent if it decreases a counter (within the resource). While considering the service idempotent, usage statistics and measurements may be updated as long as no resource data is changed. Requests for non-idempotent resources are best made using POST.

It is important to note, however, that DELETE idempotence should be viewed with caution. DELETE on a resource a second time will often return 404 (NOT FOUND) since it has been removed and is no longer available. As a result, some argue that DELETE operations are no longer idempotent. However, the resource's end-state remains the same. The 404 response is acceptable and provides a clear indication of the call's status.

This is a detailed explanation, but it is the most crucial part of learning throughout your learning journey. Throughout your education, you probably heard a lot about JSON, a format used to exchange data between systems.

Why Should You Care About JSON?

This is a text-based way to represent JavaScript object literals, arrays, and scalar data. In addition to being relatively easy to read and write, JSON is also very easy for software to parse and generate. It is frequently used to serialise structured data and exchange it over a network, usually between servers and web applications.

JSON: What does it mean?

JSON is an open data interchange format that is both human- and machine-readable, commonly known as JavaScript Object Notation. In spite of its name, JavaScript Object Notation is independent of any programming language and is a standard API output in a variety of applications.

JSON is a subset of JavaScript. The Representative State Transfer (REST) architecture uses the HTTP protocol in a simple and efficient manner. Interfaces that adhere to REST practices are often referred to as RESTful.

Are you ready to integrate your Web App with some popular APIs?

- Twitter
- Facebook
- Open Weather
- Yahoo Finance
- Free NBA
- Adopt Pet API

🎥 Videos:

- FreeCodeCamp: APIs for Beginners - How to use an API (Full Course / Tutorial)
- Udemy: Learn API Concepts for beginners

🏁 Get Started:

- Mozilla: Introduction to web APIs
- Snipcart: A Beginner's Guide to APIs: How to Integrate and Use Them

🖇️ Documentation:

- DevDocs: API Reference
- React Official Documentation: Top-Level API
- React Official Documentation: Hooks API Reference
- FreeCodeCamp: API

📖 Books:

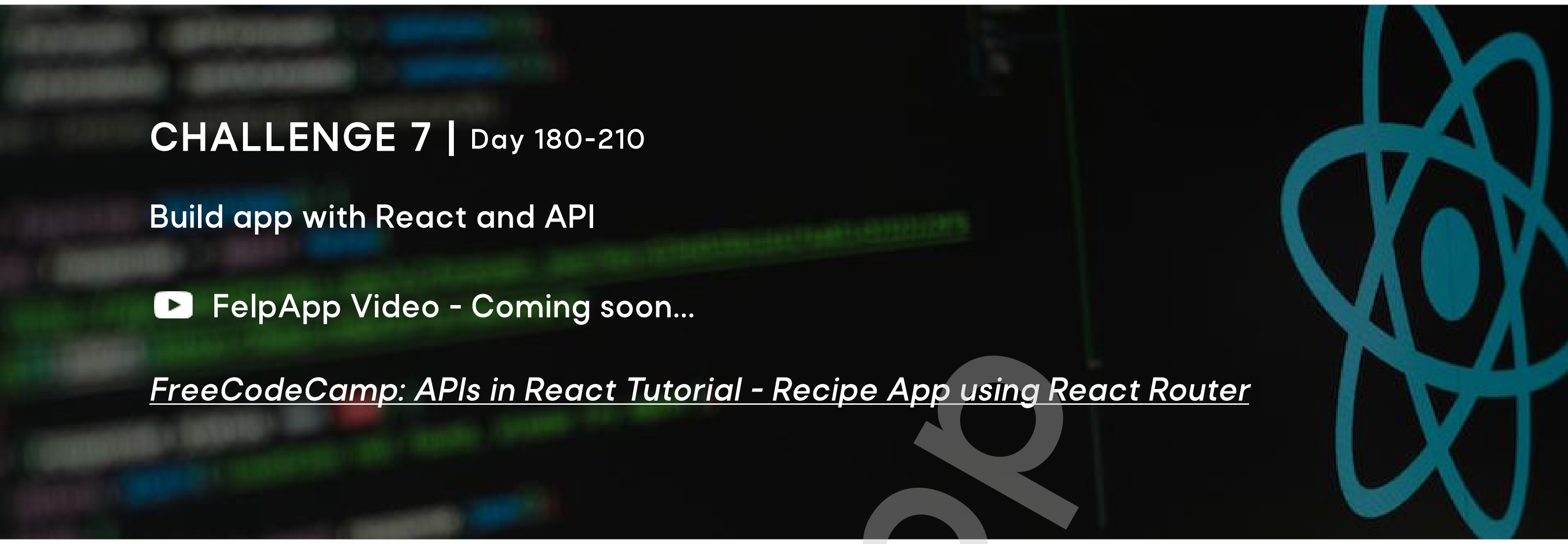
- 12 Books Every JavaScript Developer Should Read

💡 Cheat Sheet:

- API CheatSheet
- JavaScript CheatSheet

🔎 References:

- APIs for Beginners: How to use an API? A Complete Guide
- APIs for Dummies
- WebScrappingAPI: The Beginner's Guide to APIs: Everything You Need to Know
- Medium: What is an API | API for beginners?



CHALLENGE 7 | Day 180-210

Build app with React and API

▶ FelpApp Video - Coming soon...

FreeCodeCamp: APIs in React Tutorial - Recipe App using React Router

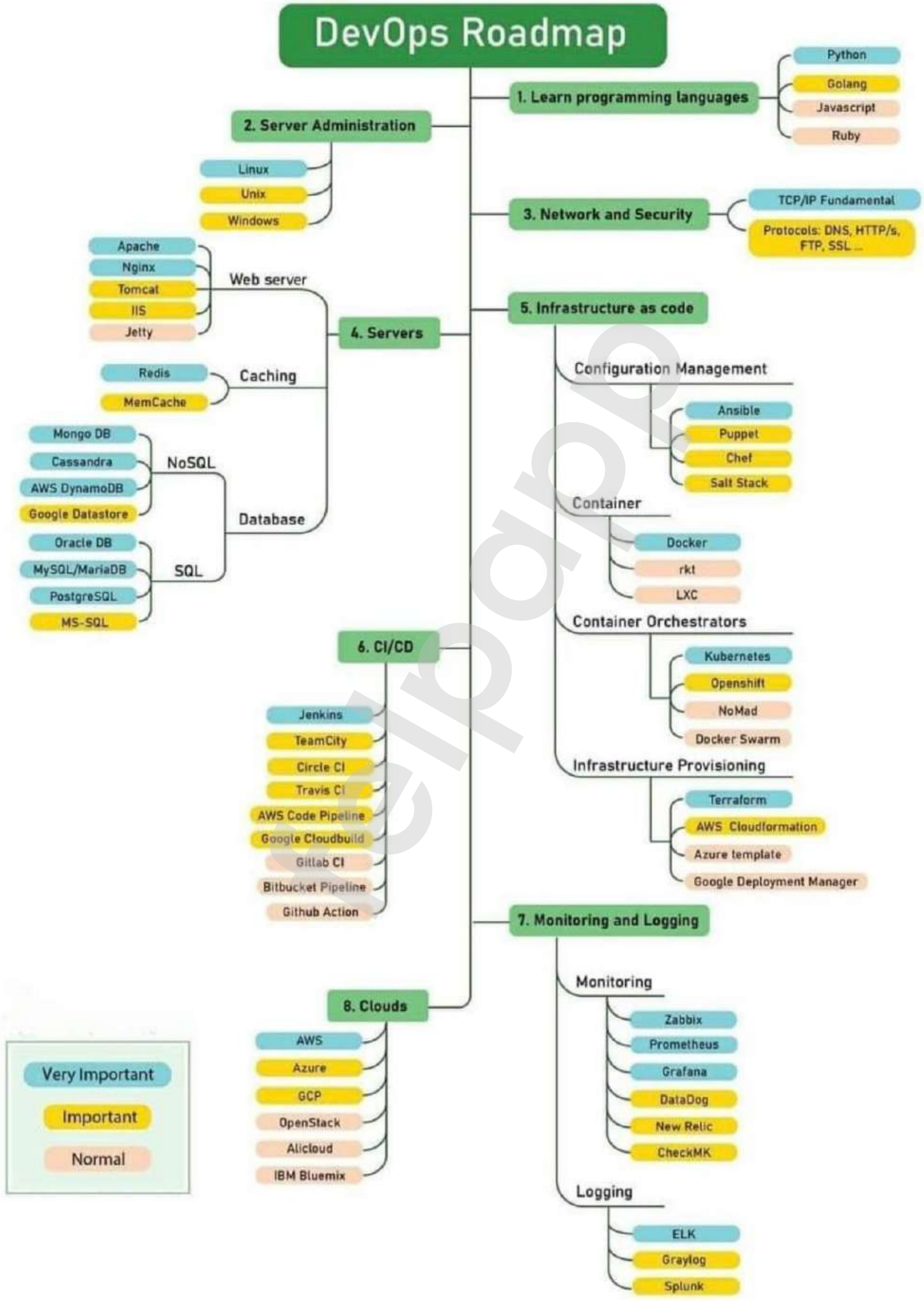
CHALLENGE 8 | Day 210-240

Build your personal project with React and NodeJS (Introduction)

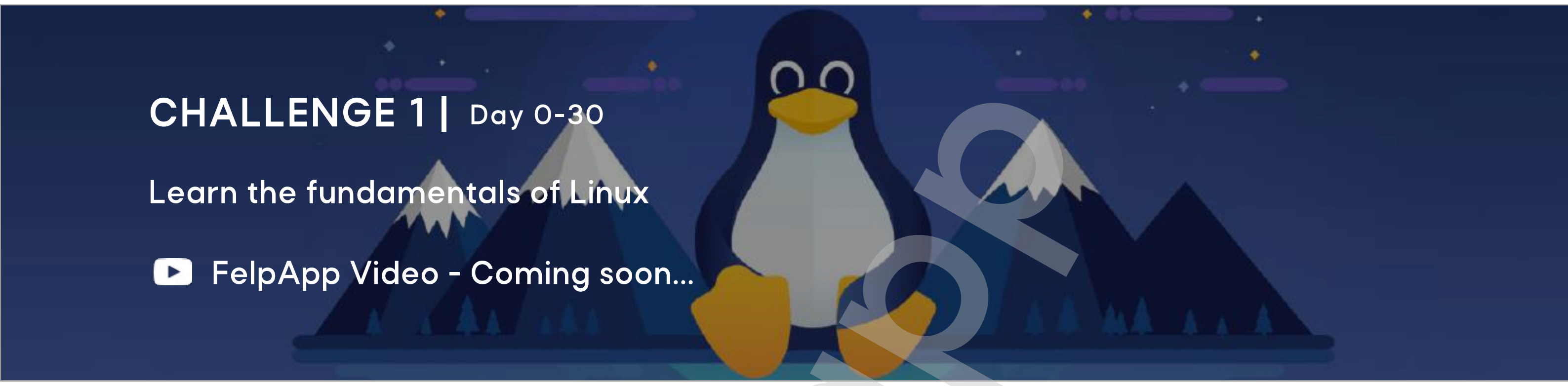
▶ FelpApp Video - Coming soon...

Simplilearn: NodeJS And ReactJS Full Course 2022 | NodeJS & ReactJS Tutorial For Beginners

2. DevOps


DevOps Roadmap
1. Learn programming languages
Python
Golang
Javascript
Ruby
2. Server Administration
Linux
Unix
Windows
3. Network and Security
TCP/IP Fundamental
Protocols: DNS, HTTP/s, FTP, SSL ...
4. Servers
Web server
Apache
Nginx
Tomcat
IIS
Jetty
Caching
Redis
MemCache
Database
NoSQL
Mongo DB
Cassandra
AWS DynamoDB
Google Datastore
SQL
Oracle DB
MySQL/MariaDB
PostgreSQL
MS-SQL
5. Infrastructure as code
Configuration Management
Ansible
Puppet
Chef
Salt Stack
Container
Docker
rkt
LXC
Container Orchestrators
Kubernetes
Openshift
NoMad
Docker Swarm
Infrastructure Provisioning
Terraform
AWS Cloudformation
Azure template
Google Deployment Manager
6. CI/CD
Jenkins
TeamCity
Circle CI
Travis CI
AWS Code Pipeline
Google Cloudbuild
Gitlab CI
Bitbucket Pipeline
Github Action
7. Monitoring and Logging
Monitoring
Zabbix
Prometheus
Grafana
DataDog
New Relic
CheckMK
Logging
ELK
Graylog
Splunk
8. Clouds
AWS
Azure
GCP
OpenStack
Alicloud
IBM Bluemix
Very Important
Important
Normal

Let's start by defining what an operating system is. Computers, mobile devices, and tablets all have operating systems that manage all their hardware resources. The operating system is responsible for managing the communication between the hardware and the software. As a result of its stability, security, and flexibility, Linux servers are widely used today and considered amongst the best. Linux is also fully open-source, which makes it a better choice over closed-source software such as Windows. This means it can be distributed and modified in an unrestricted manner.



CHALLENGE 1 | Day 0-30

Learn the fundamentals of Linux

▶ FelpApp Video - Coming soon...

What is a Linux distribution?

In the Linux world, distributions (or distros) are well-known names, such as Debian and Ubuntu, which distribute new releases along with all necessary software (network manager, package manager, desktop environment, and so on). Thus, even less experienced users can use the product immediately without learning Linux/ UNIX commands.

Linux distributions contain the Linux kernel, GNU tools, additional software, and a package manager as their main components. Besides this starter package, each distribution also offers regular kernel updates. Thus, Linux is actually a kernel, and Linux distributions are operating systems based on that kernel. This is why distributions are referred to as Linux-based systems. Furthermore, it provides regular desktop users with a display server and a desktop environment.

Stay on top of this challenge by checking out FelpApp for the latest videos and references.

Here are a few examples of Linux distributions (and links on how to build VMS with selected distros):

- Ubuntu: The most popular Linux distribution offers the largest collection of programs.

- Linux Mint: An easy-to-use distribution with an intuitive interface.

- Fedora: This Linux distribution is based on Red Hat Enterprise Linux, and it's recommended for professionals learning Linux

- CentOS: Based on Red Hat Linux, it is the commercial Linux distribution. Contrary to Fedora, it is designed for stability.

- ArchLinux: Users who prefer to do everything themselves can follow this distribution's installation guide.

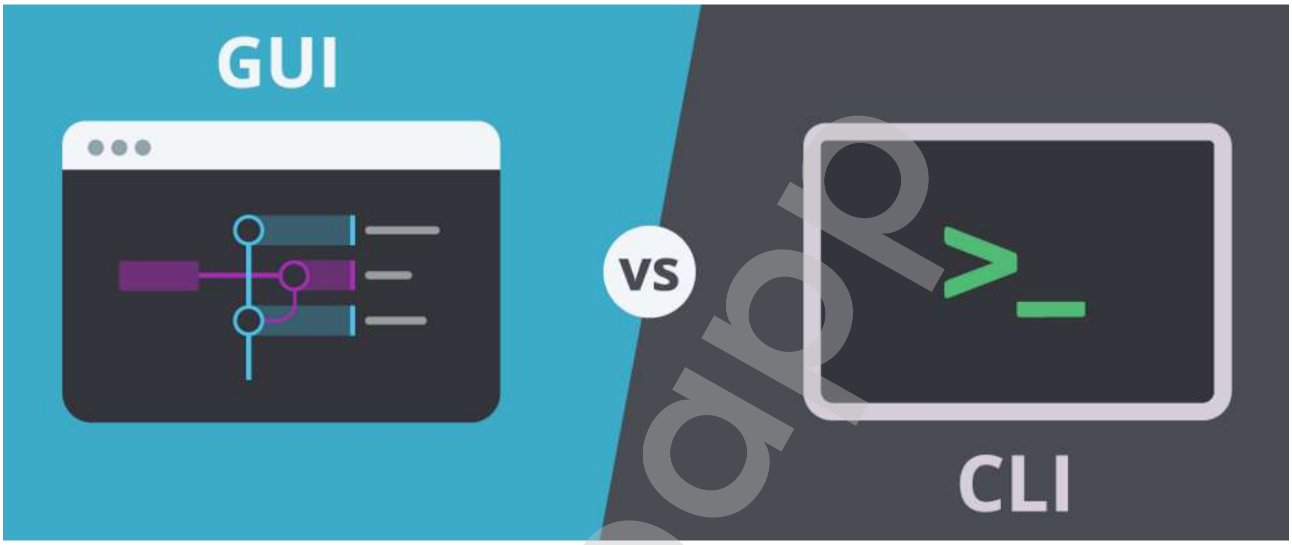


Photo Credit

What is GNU/Linux?

To be able to use Linux (without a graphical user interface (GUI)), you need a shell. Commands are usually entered using the Bash shell. Those commands and the shell are integral to Linux. However, as a matter of technicality, none of these commands originate from Linux, but rather from UNIX. Before Linux was created, the GNU project was one of the first free software projects. It all happened in 1983, when this project added to popular Unix tools like cat, grep, awk, and shell (bash) by creating its own compilers (GCC) and editors (Emacs). As UNIX was very costly back then, Linus Torvalds (credited as the inventor of Linux) created a UNIX-like kernel. For interacting with the Linux kernel, Torvalds used free and open-source GNU tools. As a result, Linux behaved like a UNIX operating system with the help of GNU tools. Due to this, Linux is commonly referred to as a UNIX-based system. Many people feel that GNU is unfairly overlooked because Linux is deeply integrated with GNU tools. When referring to all of these terms… In the end, what term would be the most appropriate to use? Linux, GNU/Linux, Linux distribution, Linux distro, Linux-based operating system or perhaps Unix-based operating system? Regardless of what you call it, today we learned about Linux distributions and why they are so popular. Moreover, we touched on the history of Linux and explained why you might hear the term GNU/Linux. Now that we have embraced the theory, let's take a look at some of the videos listed below. We have selected some of the best videos to help you learn Linux fundamentals quickly and easily. During the course of completing these videos, you will gain a lot of useful knowledge about Linux. Performing exercises and labs, creating your first Linux virtual machine, and writing your first command on Linux will help you gain a deeper understanding of the OS. Our next chapter will teach you how to turn these commands into Bash scripts. To support your studying, we have listed other resources below (book recommendations, cheat sheets, etc.)

🎥 **Videos:**

- FreeCodeCamp: DevOps Prerequisites Course - Getting Started with DevOps
- FreeCodeCamp: Linux Operating System - Crash Course for Beginners
- Intellipaat: Linux Administration Tutorial | Linux Tutorial | Linux Course
- Edureka: Linux Full Course In 5 Hours | Linux Tutorial For Beginners | Linux Training
- A Cloud Guru: LPI Linux Essentials Certification
- BONUS: Computer Networking Full Course - OSI Model with Real Life Examples

🏁 **Get Started:**

- Medium: Get Started With Linux: A Beginner's Guide

🖇 **Documentation:**

- The Linux Kernel documentation

📖 **Books:**

- 15 BEST Linux Books (2022 Update)

💡 **Cheat Sheet:**

- Linux Command Line Cheat Sheet
- Vim Cheat Sheet

🔎 **References:**

- javaTpoint - Linux/Unix Tutorial

DEVOPS | BASH

Scripting on Unix systems has largely been done with Bash. Anyone who wants to become reasonably competent in DevOps should have a working understanding of Shell Scripting. Even if you won't be writing scripts, understanding them is important, especially for analysing and modifying system behaviour.

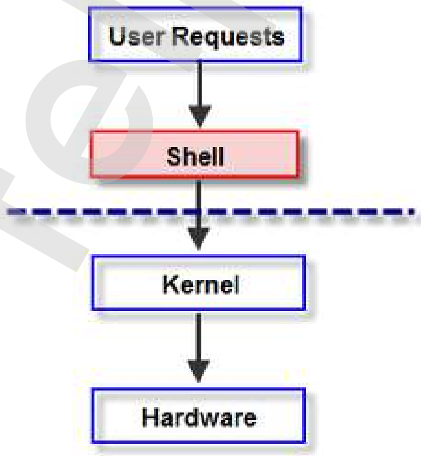
CHALLENGE 2 | Day 0-60

Introduction to Bash (Shell), Manipulation of Files and Directories using the Command Line or by writing simple scripts

▶ FelpApp Video - Coming soon...

Stay on top of this challenge by checking out FelpApp for the latest videos and references.

Earlier, we discussed Linux and GNU, in relation to that what are the following in Linux/GNU:



- Bash
  - In this case, it's a computer program that interprets orders. Brian Fox wrote Bash for the GNU Project as a free software alternative to Bourne shell. It is based on Unix and supports the POSIX standard. Most Linux distributions use it as the default shell since it was developed for the GNU project.

- Shell scripting:
  - An operating system’s components can be accessed by shell programs. There are many benefits to using shell scripts. The shell provides users (and other programs) with a means of getting inside the system. The shell defines the boundaries between inside and outside of the system.

  - Scripts should be written to accomplish tasks we want a computer to do for us. In essence, by scripting, we mean creating a file (a script that automates a task) using either the operating system’s shell (preferable) or a text editor (graphical or terminal), such as Emacs, Vi, Nano, among others. In most cases, they are saved with the extension .sh (or sometimes without it) and are run from the shell using the command: sh script name.sh. Scripts function exactly like shell commands.

  - The majority of code in this type of language is interpretable. In other words, unlike programs, shell scripts remain in their original form and are interpreted by commands, instead of being compiled (encoded), which is a permanent process of converting them into specific (special) code before execution (compilation). There is also the possibility of compiling scripts, but this is not a common practice. Which is why we use other languages like Python, JavaScript, Go etc. Learning one of these programming languages would be the next step up from Bash scripting.

  - More on this can be found here **Console vs Terminal vs Shell, and the difference between them?**

What are the characteristics of programs that use Shell Scripting?

Pros:

- ✓ The syntax is simple, does not take long to master.
- ✓ Scripts can be broken down into sections.
- ✓ There are only a few Shell-specific oper ators and options to learn.
- ✓ A shell script can be used to quickly prototype complex applications.
- ✓ Test and modify the application’s structure.
- ✓ Coding can be streamlined by identifying major (potential) issues .

Cons:

- ✗ Can be challenging when dealing with resource-intensive tasks (sorting, hashing, recursion)
- ✗ Whenever cross-platform portability is required.
- ✗ In complex applications, where the use of structured programming is necessary (typing variables, prototyping functions, etc.)

- ❌ Where the integrity of your system should be guaranteed in situations where security is of the utmost importance
- ❌ Integration with GUI

Here are some examples of common shell scripting text editors: **3 Best Scripting Editors for Linux**

🎥 Videos:

- FreeCodeCamp: Bash Tutorial: How to Use the Command Line in Linux, Windows, and Mac Terminal
- Edureka: Shell Scripting Tutorial | Shell Scripting Crash Course
- A Cloud Guru: The System Administrator's Guide to Bash Scripting
- Bash Scripting Full Course 3 Hours

🏁 Get Started:

- Rip Tutorial: Getting Started with Bash
- Medium: Bash Scripts (Part 1) Getting Started
- JavaTPoint: Bash Scripting Tutorial

📎 Documentation:

- Bash Documentation

📖 Books:

- Best Linux Bash Scripting Books for 2022

💡 Cheat Sheet:

- Bash Scripting Cheat Sheet

🔎 References:

- Ryans Tutorials - What is a Bash script?
- Linux Config - Bash Scripting Tutorials for Beginners

DEVOPS | Configuration Manager

The same tasks are probably repeated over and over again when you work in DevOps. Wouldn't it be nice if you could solve every problem on the first attempt, then automate your solution? With Ansible, you can accomplish this.

CHALLENGE 3 | Day 60-90

Learn how to execute shell commands with Ansible

▶ FelpApp Video - Coming soon...

According to its definition, Ansible is a configuration management tool that helps keep track of infrastructure components' configurations, documentation, and other data. It contains information about software versions, patches, and other features. The automation of complex tasks simplifies engineers' jobs, allowing them to focus on other more important duties. In turn, this increases an organisation's value. Consequently, it boosts productivity and saves time.

Ansible offers the following advantages over manual configuration:

- An outcome that can be predicted for repeatable actions
- More people can participate in the configuration process
- A clearly visible alterations in the infrastructure
- Code-based infrastructure definition and description

Server configuration involves managing users, software, services, firewalls, etc. These servers can be physical or virtual, running locally or virtually in the cloud. They can be Linux or Windows based, or if there are one or a thousand of them - Ansible does not care. As long as Ansible can communicate with the servers, it can also configure them, simple as that.

Ansible is of course not the first or the only configuration management tool.
Some of the alternatives that you will find are

- Puppet
- Chef
- Salt
- and of course shell scripts.

The following are some of the things that make Ansible unique:

- There is no need for agent software on configured servers
- The syntax is YAML, so it is relatively easy to learn
- It requires Python, which is installed by default on Linux systems

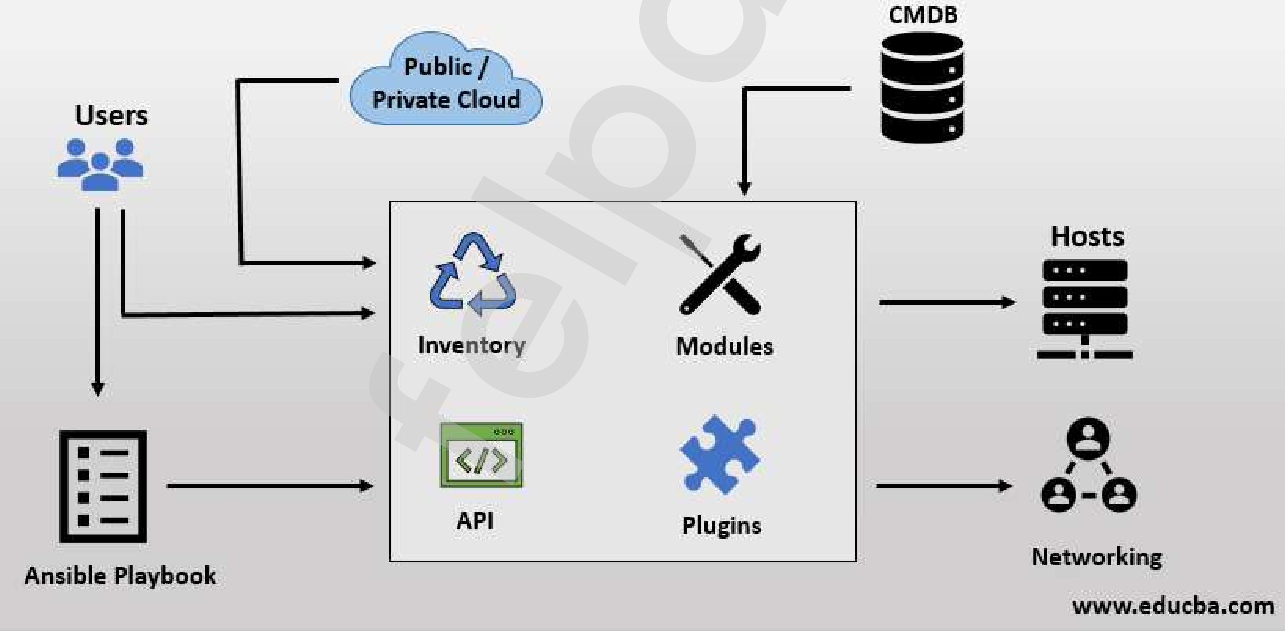
The guidance you will receive from FelpApp about Ansible as a beginner will help you obtain a job in your future role. As a part of your portfolio, you will also be given projects and tasks to work on so you can show off your work one day to potential employers.

Ansible Architecture

Ansible must meet a few basic requirements before it can be installed and configured:

- At least two cloud virtual machines are required
- A good internet connection

Ansible orchestration interacts with cloud and configuration management databases, as well as other services, using the Ansible orchestration engine. An illustration of this concept can be found in the diagram below.



www.educba.com

Ansible uses YAML to create playbooks, which define how tasks are processed. Executions can be performed synchronously or asynchronously. Your workflow is defined by your playbooks because they are processed in the order they are listed. Ansible connects to a number of subcomponents in the automation engine to push playbooks over SSH. The automation engine also allows users to run playbooks directly on hosts. It is important to note that there are many different components in the Ansible automation engine:

- **Inventory:**

The first is the inventory of hosts. Ansible's automation engine contains an inventory of all the IP addresses of all the hosts. SSH is used to communicate with Linux servers, and WinRM is used for communicating with Windows servers. In order to communicate with servers, it is necessary to define an inventory file, which can be named anything, although 'hosts' is a common naming convention. In the inventory, servers can be individually defined, grouped, and contained within groups.
Example: How to build your inventory

- **Modules**

Our focus has so far been on how Ansible connects to servers. The functionality of Ansible is defined by modules. The modules in Ansible contain functional code. Modules can be divided into two types: standard and extras.
Standard modules include copy, user, npm, ping, setup, cron, and hostname. The name of a module indicates what function it performs.
Ansible's Extras Modules can be installed via the Linux distribution's package manager or directly from GitHub.
Example: Extra Modules by Ansible

- **Tasks**

The command line can be used to execute modules, but this is not very common. A task is an atomic action with a start, duration, and end.
Example: Intro to Playbooks

- **Playbook**

The tasks mentioned in the previous paragraph aren't very meaningful on their own. With Ansible, you can organise tasks using a playbook. Therefore, a playbook describes the state of a system, performs its configuration, and orchestrates its execution.
The header of the playbook includes the title of the playbook, followed by the name of the server group or server we are configuring based on the inventory file.
Become indicates that all actions on the configured server will be performed through sudo elevation of rights (essentially becoming an admin to be able to execute the tasks).
If the playbook is executed on multiple configured servers, the tasks will be executed in parallel. Some tasks may fail due to errors, of course. The requests are executed one by one in the order in which they were defined. The configuration of the system will not be changed if the state does not agree with the system state. If there is an

error, you will have to correct it and restart the playbook.

Example: Working with Playbooks

- **Handlers**

It is necessary to establish a cause-and-effect relationship between actions during the system configuration process. A simple example would be to change the configuration file and restart the service. In these situations, Ansible uses the handler concept; handlers are tasks that require a trigger. In spite of being called by multiple tasks, the handler will only be executed once. Changing one file multiple times saves the service from multiple restarts, which is perfectly reasonable - change the file ten times, restart the service once.

Example: Handlers: running operations on change

Stay on top of this challenge by checking out FelpApp for the latest videos and references.

Visit the official website for documentation: **How Ansible Works by RedHat Ansible**

📽️ Videos:

- Edureka: Ansible Full Course | Ansible Training | Learn Ansible for Beginners
- Ansible 101 with Jeff Geerling
- Ansible Course for Beginners - Learn Ansible in 1 Hour

🏁 Get Started:

- RedHat Ansible: Get started - Ready to start automating?
- Run Your First Command and Playbook
- Medium: Getting Started With Ansible

🖇️ Documentation:

- Official Ansible Documentation
- Jeff Geerling's Ansible Github Repo

📖 Books:

- 15 Best New Ansible Books To Read In 2022

💡 Cheat Sheet:

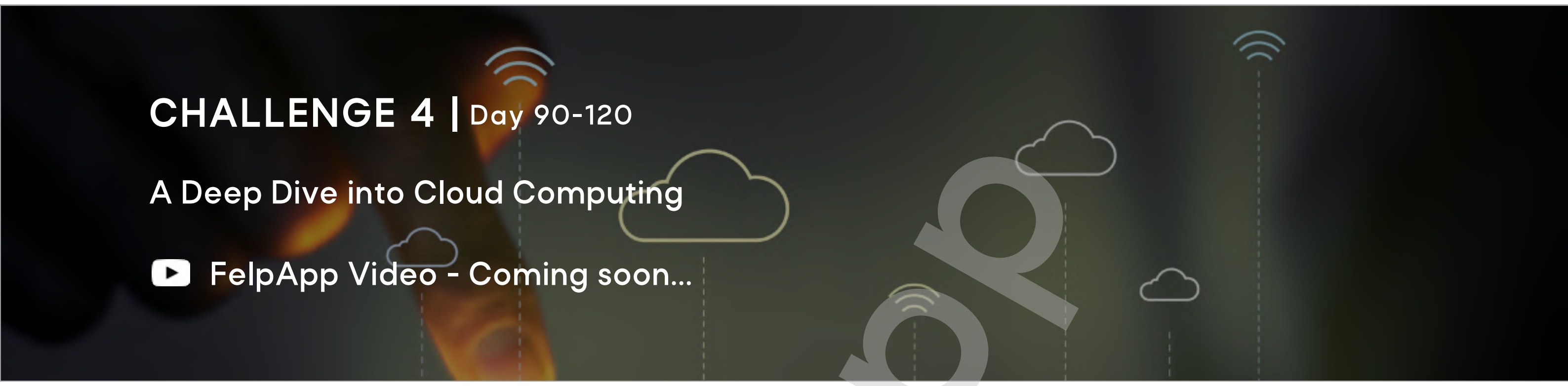
- Ansible Basic Cheat Sheet

🔎 References:

- Ansible Tutorial for Beginners: Playbook, Commands & Example
- Getting Started with Ansible Tutorial - Automate your Infrastructure

DEVOPS | Cloud Computing

This is a term used to describe all hosted services offered over the Internet. In addition to cloud computing, companies have the ability to use resources such as virtual machines and object storage for their applications, platforms, and the entire business. There are differences between cloud solutions in terms of the benefits they offer, the level of protection, the level of access to data, and the way that responsibilities are shared between the client and the provider.



CHALLENGE 4 | Day 90-120

A Deep Dive into Cloud Computing

▶ FelpApp Video - Coming soon...

Generally, cloud solutions fall into three categories:

- **Public Cloud:** A third party (cloud service provider) owns and manages the solutions, making their storage space and virtual servers available to customers. There are a variety of resources available in the cloud, including processing power, storage space, and specific applications. It is important to understand that these services may be free or charged per use, depending on the provider and the type of service. (Always check the pricing!)

  Therefore, all hardware and software equipment belongs to the cloud provider and they are responsible for maintaining the infrastructure. Users share resources on the Internet and can access them from anywhere in the world.

  In companies managing large projects involving multiple parties, the data can be updated instantly, which is a great benefit. Although this solution provides excellent scalability and flexibility, it is still a little more vulnerable (in terms of security) than the Private Cloud.

- **Private Cloud:** Private Cloud solutions are computing resources used exclusively by one company or organisation, which provides an additional layer of protection and security. Users have access to a system that is entirely theirs, and the company's IT department manages and controls this cloud. Some companies choose to house all of their infrastructure in their own physical location, while others partner with a cloud provider to handle the maintenance.

- **Hybrid Cloud:** A hybrid cloud combines the advantages of both public and private clouds, allowing data to be exchanged between them. By integrating certain public cloud services with an existing private cloud infrastructure, users can create a single entity, which includes both private and public clouds. Enhanced data security is provided as a result of this. This setup will also be virtually invisible to service users, as to where certain services operate within the infrastructure. Full mobility of services between private and public works is ensured, as well as unified management of the available infrastructure.

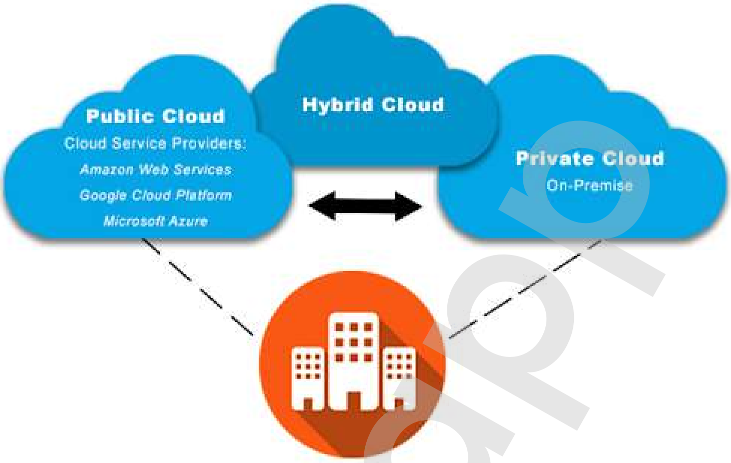


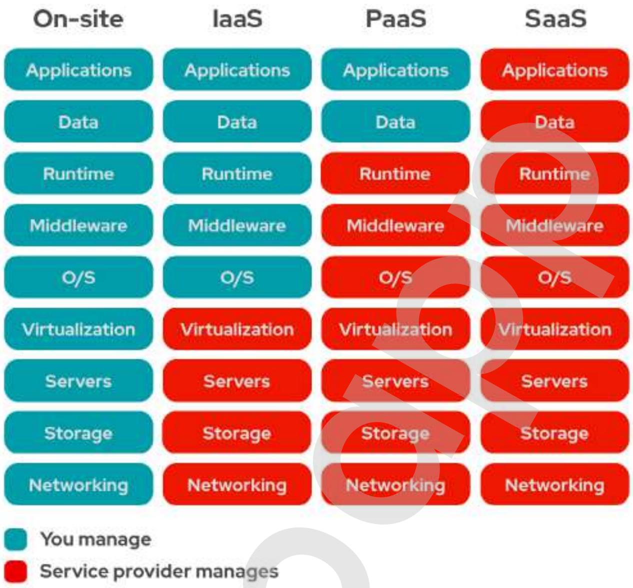
Photo: Logan Westberg on LinkedIn

In terms of cloud computing, there are several key features:

- **Easy Maintenance** - This is one of the best cloud features. Maintaining servers is easy, and downtime is usually minimal or nonexistent.
- **Rapid Elasticity and Scalability** - It is possible to quickly increase or decrease resources using cloud functionality. Certain functionalities can be automatically scaled depending on the settings.
- **Resilience** - As part of cloud computing, this means the ability of a service to quickly recover from any interruption. The resilience of a cloud is measured by how fast its servers, databases and network systems restart and recover from any loss or damage.
- **Availability** - This is another key feature of cloud computing. Due to the ability to access cloud services remotely, there are no geographical limitations or restrictions on their use.
- **On-demand Self-Service** - Selecting and running computer resources can be done independently by the user. Depending on how much and how often users use services, they are charged.

- Broad Network Access - You can access all services over the network with a variety of devices with different operating systems (mobile phones, tablets, laptops, workstations).
- Measured service - Cloud resources are checked and optimised with appropriate systems. The tool ensures that billing is being monitored and that resources are being utilised to the maximum.

Cloud computing can be divided into three types:



On-site	IaaS	PaaS	SaaS
🟦 Applications	🟦 Applications	🟦 Applications	🟥 Applications
🟦 Data	🟦 Data	🟦 Data	🟥 Data
🟦 Runtime	🟦 Runtime	🟥 Runtime	🟥 Runtime
🟦 Middleware	🟦 Middleware	🟥 Middleware	🟥 Middleware
🟦 O/S	🟦 O/S	🟥 O/S	🟥 O/S
🟦 Virtualization	🟥 Virtualization	🟥 Virtualization	🟥 Virtualization
🟦 Servers	🟥 Servers	🟥 Servers	🟥 Servers
🟦 Storage	🟥 Storage	🟥 Storage	🟥 Storage
🟦 Networking	🟥 Networking	🟥 Networking	🟥 Networking

🟦 You manage
🟥 Service provider manages

Photo: Red Hat

1. IaaS: Infrastructure as a service

An IaaS model is the fundamental layer and structure necessary for Cloud Computing. This is the most basic cloud computing model, which involves renting IT infrastructure (e.g. servers and virtual machines, storage space, networks, operating systems) from a third party (cloud provider). During the creation of a system, this layer ensures the proper functioning of the service and allows the platform to be implemented.

The system works on a pay-as-you-go basis, which means users only pay for the resources they actually use. By optimising costs and spending financial resources more rationally, companies are able to reduce costs. From infrastructure to end-user applications, all services are provided over the Internet by the provider. All these activities are coordinated by infrastructure architects and engineers, who organise and perform maintenance activities to ensure the service operates efficiently and effectively. This cloud service is particularly beneficial to small and medium-sized businesses with limited resources to invest in their own infrastructure. As a result, companies are able to acquire a reliable, stable, and highly available infrastructure, which was previously only available to large companies. A lot of the popularity of this service was due to the development of virtualisation technology.

IaaS Examples

2. PaaS: Platform as a service

Platform-as-a-Service (PaaS) is a cloud computing model that implies that a third party (i.e. cloud provider) provides users with all the necessary hardware and software equipment, which enables application development or collaboration through the platform via the Internet. In other words, PaaS is a tier above IaaS. A user can develop, test, and distribute their own applications on the infrastructure of the cloud service provider.

A provider is responsible for maintaining the infrastructure and owns the whole system. Typically this involves the execution environment, including servers, network infrastructure, data storage centres, operating systems, and programming languages. Some providers automatically adjust resource volume, so user admins don't have to make additional allocations. Therefore, whoever chooses the PaaS model can freely use the cloud, without increasing their hardware and software investments or incurring the costs typically associated with maintaining an on-site infrastructure. Applications and intermediary layers are under control of the user, while other infrastructure layers are under control of the cloud service provider. This is advantageous so software development teams are independent of geographical location, resources, and other team members.

PaaS Examples

3. SaaS: Software as a Service

Users now have the option to use cloud-based applications or better yet, rent or borrow online software rather than purchasing and installing it on their computers. Applications can be free or paid for. The most common model of paying for applications is a monthly or yearly subscription. The provider usually owns the network, services, operating systems, data storage systems, as well as specific software that is accessible via Internet browsers by a large number of users. A number of cloud-based applications are available to company employees. Essentially, the rented software is located on a platform where resources are shared. Clients can access applications via the Internet from a variety of devices (including phones, tablets, laptops, and desktop computers.). Cloud providers control the entire infrastructure, and companies control the access rights to rented software, for example, who can use it, from what location, for how long etc. It is not the customer's responsibility to invest in software, update hardware frequently, or maintain the system, as this will be undertaken by the provider. On such a system, personal data belongs to the company. In this sense, SaaS is similar to what people do when they use Gmail or Yahoo mail services, except that SaaS goes much further. SaaS is the fundamental idea behind centralised computing: entire businesses and thousands of employees will manage their computing tools as online rental products. Brilliant, isn't it?

SaaS Examples

Photo: Clinked

Can a beginner use cloud services? Is there anything they need to know?

Your only requirements will be a computer with an Internet browser, an Internet connection, and a cloud computing subscription.

As future DevOps engineers, you will often be working with cloud-based projects, therefore it will be important for you to familiarise yourself with cloud-based services to succeed in your new role. We will be focusing on three of the most popular public cloud services here at FelpApp:

The Cloud Provider you choose to learn and base your knowledge on will depend on your preference or prior experience (if you have any). In the future, you will see that all Cloud Providers will have similar principles.

Stay on top of this challenge by checking out FelpApp for the latest videos and references.

🎥 **Videos:**

1. My Great Learning - AWS Azure GCP
2. My Great Learning - Cloud Foundations
3. Cloud Computing Tutorial for Beginners | Cloud Computing Explained | Cloud Computing | Simplilearn
4. A Cloud Guru: Cloud Provider Comparisons: AWS vs Azure vs GCP
5. Cloud Provider Comparisons: AWS vs Azure vs GCP - Certifications

🏁 **Get Started:**

- Amazon Web Services (AWS)
- Google Cloud Platform (GCP)
- Microsoft Azure

🖇 **Documentation:**

- AWS Documentation
- GCP Documentation
- Azure Documentation

📖 **Books:**

- Best Cloud Computing Books for Beginners & Experienced Professionals

💡 **Cheat Sheet:**

- Cloud Computing - Summary Cheat Sheet

🔎 **References:**

- Guru99 - What is Cloud Computing? Examples, Types & Benefits
- Redhat - IaaS vs PaaS vs SaaS
- Geeks for Geeks - Difference between IAAS, PAAS and SAAS

Infraestructure as Code

Terraform is an open source infrastructure as code (IaC) software tool that makes it possible for DevOps engineers to automate the provisioning of the resources that an application requires. It supports a wide array of cloud platforms, which makes it distinctive. It is created by HashiCorp, it was first released in 2014 and is written in Golang.

Infrastructure deployment and management are not straightforward tasks. There are several steps involved in setting up an environment for a cloud application. Manual infrastructure management is not able to keep up with the rapid development cycles, resulting in bottlenecks and delays in delivery. Unless you write them all down and follow them carefully, all the time, you will make mistakes; we are all human. There is no easy way to share these steps. It is necessary to document many manual processes, and these documents can become out of date very quickly. Multiply that by the total number of environments for one application:

- Dev
- Test / QA
- Staging
- Production.
- Additionally, **you need to take their security into consideration.**

CHALLENGE 5 | Day 120-150

Learn the fundamentals of IaC and build your first infrastructure using Terraform

FelpApp Video - Coming soon...

nfrastructure management can be challenging, because we have to keep up with constantly changing domain expertise and learn the latest tools and workflows. The IaC reduces IT management complexity by managing applications' underlying infrastructure through code. This book provides you with an opportunity to learn about one of the approaches to infrastructure management. The end result should be you becoming more confident about Terraform and having a better understanding of possible approaches and challenges.

A Terraform infrastructure may include low-level components such as compute instances, storage, and networking, as well as high-level components such as DNS la entries and SaaS features. The system also lets users deal with complexity and code duplication (code recycling) with the help of shareable modules. Modules are used by DevOps engineers to group together the infrastructure resources they want to use. There is an input variable and an output variable in modules. Configurations can be made more quickly since modules can call each other.

👉 Find out more about this in this article **Terraform Modules** and their examples through code

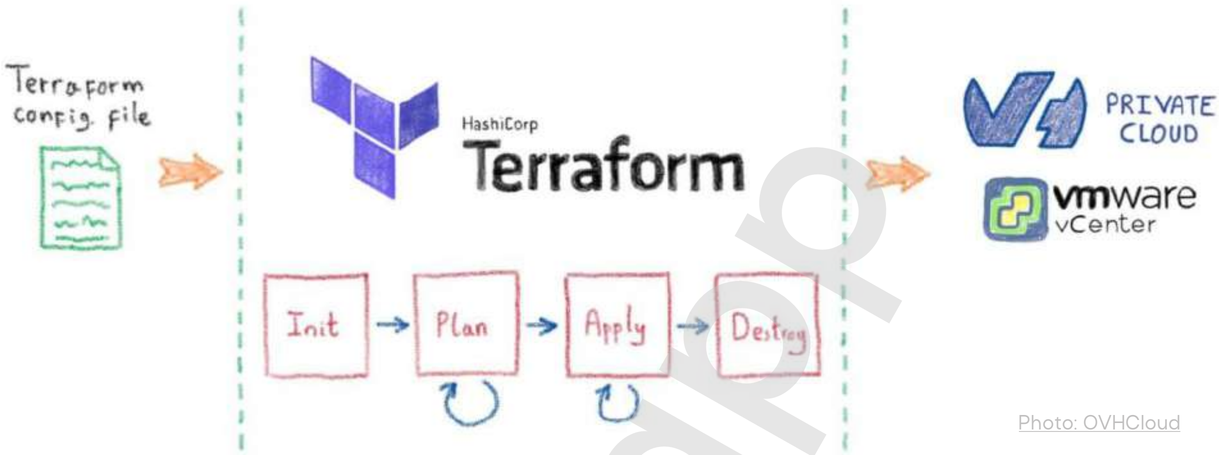


Photo: OVHCloud

Terraform uses a JSON-like configuration language known as HCL (HashiCorp Configuration Language) to define and enforce infrastructure configuration. In combination with HCL, DevOps team can run through the standard workflow:

1. Write: Terraform configuration files are used to define or modify the components required for a particular application or a whole data centre to run.
2. Plan: The second stage of the workflow is then accomplished using Terraform's execution plan, which describes how the desired state will be achieved. You will be able to see how much additional infrastructure charges will occur as a result of creating, resourcing, modifying, and deleting new resources.
3. Apply: In the final stage of the workflow, the user confirms the changes. In order to achieve the desired state, Terraform will perform the alterations in a specific sequence, respecting all resource dependencies.
   Please note: When you use Terraform, the state keeps track of all infrastructure changes and config drifts. At the time of Terraform's initial execution, a state file is created and updated with new changes. Depending on the environment, state files can be saved locally or in S3 buckets. Identifying resources and keeping track of infrastructure changes are always handled by referring to this state file.
4. Destroy: Delete all resources previously applied by Terraform

👉 You can find out more by watching this video **Terraform explained in 15 mins | Terraform Tutorial for Beginners**

The idea is to provision infrastructure and re-provision it across multiple cloud services and on-premises infrastructure. On this link of the official Terraform webpage, There are examples of building, modifying, and destroying infrastructure. (When you reach the webpage, click the relevant action on the left side).

👉 You may be interested in this article about **What is the use of Terraform in DevOps?**

It is important to note that these steps could be applied to a variety of cloud providers (AWS, Azure, GCP).

Alternative IaC tools: Ansible, Puppet, Chef, Saltstack

👉 Check out this article on 10 Best Terraform Alternatives in 2022: IT Automation Tools

There are several advantages to using Terraform:

- ✓ There are multiple providers supported, including AWS, Azure, Google Cloud Platform, and more.
- ✓ Rolling back is simple
- ✓ Provides configuration and orchestration management
- ✓ The language used is HCL, which is simple and easy enough to understand Managed state
- ✓ A client-only architecture eliminates the need for server-side configuration management.
- ✓ Automation of the infrastructure using CI/CD pipelines

The use of Terraform is enjoyable for many beginners and professionals, both of whom keep using it daily once they become familiar with it. Getting started can be a bumpy road, but there are more and more resources available to help you on your way. It is definitely worth exploring Terraform and trying it out.

Stay on top of this challenge by checking out FelpApp for the latest videos and references.

🎥 Videos:

1. FreeCodeCamp: Terraform Course - Automate your AWS cloud infrastructure
2. Complete Terraform Course - From BEGINNER to PRO! (Learn Infrastructure as Code)
3. Terraform Tutorial for Beginners + Labs: Complete Step by Step Guide
4. A Cloud Guru: Deploying to AWS with Terraform and Ansible

🏁 Get Started:

- Official HashiCorp Terraform Getting Started Labs and Tutorials

📎 Documentation:

- Official HashiCorp Terraform Documentation

📖 Books:

- 5 Best Books on Terraform 2022

💡 Cheat Sheet:

- Terraform Cheat Sheet

🔎 References:

- ITNEXT - Terraform for Beginners
- Hashicorp - Terraform Tutorials
- Geekflare - An introduction to Terraform for Beginners

For a better understanding of the terminology itself, let's distinguish what containers are. Containerization, also known as container virtualization or operating system-level virtualization, is a method for organising data in containers and sharing them over the Internet. This is where Docker is similar to Git in its principle of operation. The package is now available in almost every Linux distribution and is increasingly popular around the globe.

Its popularity can be attributed in part to Docker Hub, a web hub where container images are shared. Docker itself finances the development and maintenance of official images. Along with the hello-world image that tests Docker's installation, official images include httpd, python, php, node, haproxy, mariadb, mongo, postgres, redis, and numerous others. For each official image, detailed instructions are provided.

Docker Official Images

CHALLENGE 6 | Day 150-180

Understand the importance of Docker Containers

FelpApp Video - Coming soon...

Docker is consisted of:

- **Client and Server** - Docker is a client-server application. In Docker, the client interacts with a backend server or daemon. There are two ways to manage Docker: via the command line and via RESTful APIs.
- **Image** - The foundation of Docker. An image serves as a basis for launching containers. Images have a layered format, use the Union file system and are created by executing a specific set of commands. Images are considered to be the source code of containers. Their portability allows them to be stored, modified, and shared.
- **Registries** – images are stored in registries. Registries can be classified as either private or public. Docker Hub is the public image registry managed by Docker. In addition to public registries, Docker Hub also supports private registries, where users can save their images privately and download them at a later time.

- **Registries** – images are stored in registries. Registries can be classified as either private or public. Docker Hub is the public image registry managed by Docker. In addition to public registries, Docker Hub also supports private registries, where users can save their images privately and download them at a later time.
- **Container** - An image can launch a container, which can contain one or more processes. Containers based on Docker are similar to real containers, except they contain software rather than physical items. It doesn't matter where a container comes from or what's inside, Docker starts all containers the same way regardless of their origin.

Containers vs Virtual Machines

The container is built on top of the core of an operating system (OS) and consists of only the files that are most important for the operation of the system. In contrast, each virtual machine runs its own operating system. This is why containers are typically measured in megabytes, while virtual machines are measured in gigabytes. VMs are virtual machines in which physical resources such as CPU and RAM are managed by a "hypervisor". Docker uses the Docker engine for the same purpose. A benefit of using containers is the speed at which they start up. Containers start in seconds, whereas VMs take minutes. However, VMs are fully isolated from one another and are therefore more secure.

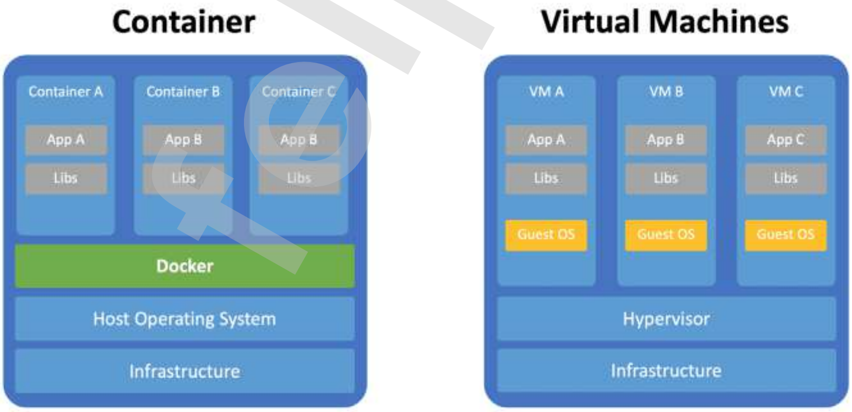
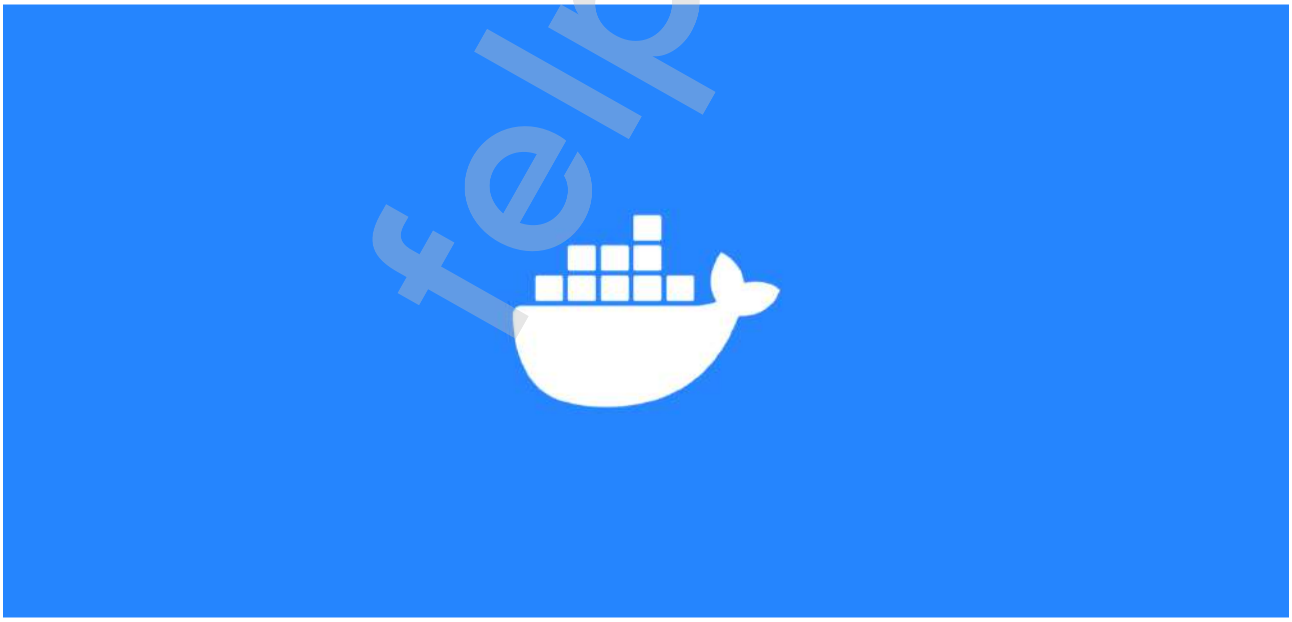


Photo Credits

In the world of development and operations, Docker is your best friend. Most people avoid it because they believe it is complicated and hard to work with, but it isn't like this at all.

Docker is a technology for creating software containers, which are packages of individual applications that include everything necessary to run them. It is possible to run multiple containers on one server simultaneously. As opposed to using virtual machines, setting up the server is simpler, as it only needs a basic operating system and application that enables containers, like Docker Engine in this case. A key benefit of Docker is that it facilitates the use of containers in a cloud environment, making it easy to use them in conjunction with DevOps applications such as Puppet, Chef, and Ansible, simplifying the tasks typically associated with these applications. Docker is especially useful for launching multiple development environments with different settings, for testing, or for allowing people to work on the same project with identical settings etc.

How many times did you hear developers or engineers complain about the same project setup on different computers? And always something went wrong.
Perhaps the database, app, or operating system had a problem, and a developer may have been unaware, or a devops specialist may not have known what the developer put in the dependency etc. As a result, they switch from computer to computer and spend everyone's time (including company's time) setting up. That's where Docker comes in. Docker is, essentially, a micro-operating system (though it acts as a service). Although it uses the kernel and some libraries of the host operating system, it is essentially independent and gives the impression of being on a remote server. Everything you need can be installed and set up within the container, and you can destroy everything without any remorse to the host operating system.



www.docker.com

Stay on top of this challenge by checking out FelpApp for the latest videos and references.

🎥 Videos:

1. Docker Tutorial for Beginners
2. TechWorld with Nana: Docker Tutorial for Beginners [FULL COURSE in 3 Hours]
3. What Is Docker? | What Is Docker And How It Works? | Docker Tutorial For Beginners | Simplilearn
4. Docker For Beginners: From Docker Desktop to Deployment

🏁 Get Started:

- Docker Orientation and Setup
- Docker Get Started

📎 Documentation:

- Docker Documentation

📖 Books:

- 5 Best Docker Books to Learn in 2022

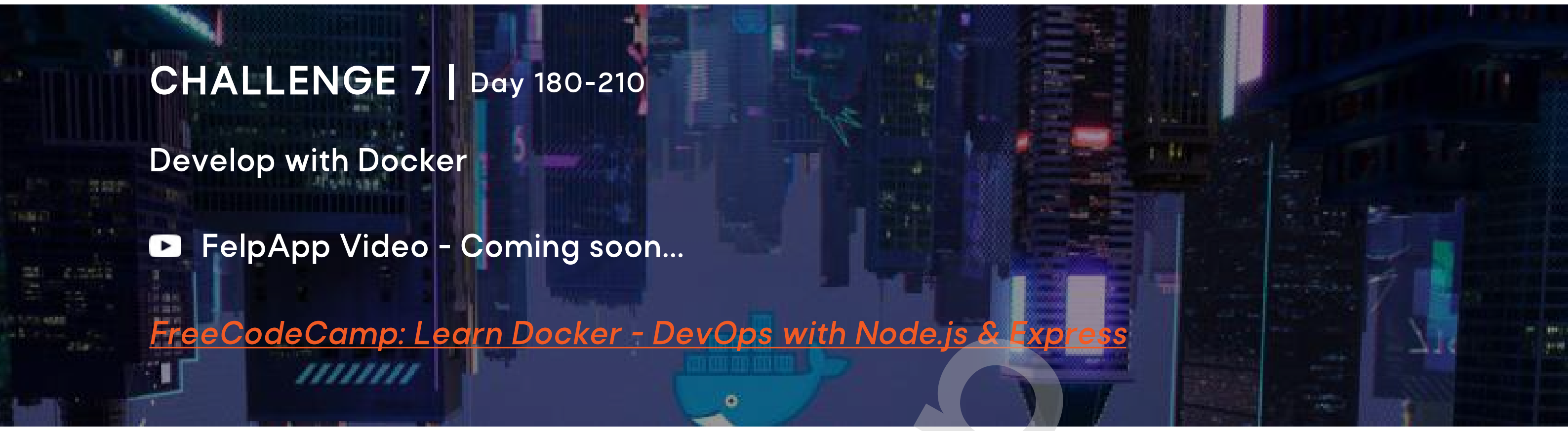
💡 Cheat Sheet:

- Docker CLI Cheat Sheet

🔎 References:

- Docker for Beginners
- Simplilearn Docker Tutorial: A step by step tutorial for Beginners
- Freecodecamp - Docker Simplified: A Hands-On Guide for Absolute Beginners

DEVOPS | Bonus Challenges



CHALLENGE 7 | Day 180-210

Develop with Docker

▶ FelpApp Video - Coming soon...

FreeCodeCamp: Learn Docker - DevOps with Node.js & Express

CHALLENGE 8 | Day 210-240

Build your personal project with Docker and Kubernetes (Introduction)

▶ FelpApp Video - Coming soon...

FreeCodeCamp: Kubernetes Course - Full Beginners Tutorial (Containerize Your Apps!)

2. Development & Operations Tools

DEVELOPMENT & OPERATIONS TOOLS | Containerisation

As part of this section, we will explore one of the most popular text editors out there, Visual Studio Code released by Microsoft. It is an integrated development environment (IDE) that allows users to edit various types of text and code. In comparison to its predecessors - Sublime Text and Notepad++ - it offers more functionality, but at the same time, it is free (there is no premium version like Sublime Text). Although Visual Studio Code has a great deal of power, it is easy to use and quick to master.

Various programming languages are available, including Go, Script, Python, Java, Dockerfile, SQL, Visual Basic, CSS, JSON, Ruby, PowerShell, Objective-C, JavaScript, Swift, C++, PHP. Each option features keyboard shortcuts designed for experienced programmers. This program offers a wide range of free extensions. Using these tools, you can manage databases, complete programming language syntax automatically, debug code, etc. The vast majority of extensions are open source. The "find and replace" feature is one of the coolest features of this program. With this tool, you can replace consecutive words in a project, allowing you to edit even more precisely. There is also the possibility of protecting code from deletion. Whatever the circumstance - a system crash or just closing the program - the last version will always be saved.

🎥 **Videos:**

Freecodecamp: Visual Studio Code Crash Course

🏁 **Get Started:**

Official Microsoft: Getting started with Visual Studio Code

🖇️ **Documentation:**

Official Microsoft Documentation for Visual Studio Code

📖 **Books:**

7 Best New Visual Studio Code books To Read In 2022

If you are already using GNU/Linux and have not yet had an opportunity to use Git, in this text we will run through some basic concepts of what Git is and how you can use it.
Originally created by Linus Torvalds, who is also the creator of Linux, Git was developed while he worked on building the Linux kernel. Since that time, Git has become accepted in many software projects and is a specific implementation of a version control tool, better known as distributed version control revision.

This tool is mainly used by Software developers and DevOps engineers for delivering software faster and more efficiently, while strengthening security and compliance.
It has the ability to track changes over a set of files. Changes can be monitored locally as well as in a shared work environment. This monitoring is advantageous because it gives us information about what has changed, when it has changed, who made the changes, and why they made them. This method allows the changes to eventually be merged into one history.

https://git-scm.com/

Git also allows one or more developers to work on the same project simultaneously and apply changes to the code in the files of a single project, which minimises the possibility of inconsistencies or conflicts due to these changes. Since it is a distributed system for version control, each developer has an entire repository that is stored locally, while the server enables cross-user collaboration and exchange. In other words, each computer has a copy of the data that is stored locally and that is synchronised with the server.

The main development line can be divided into branches using Git. These branches can be further changed and eventually merged with the main branch.
Changes to project files can be made remotely, for example on your device, and uploaded using Git to the repository. The development can be done on a remote repository such as GitHub or Gitlab, and only the changes can be pulled locally via Git Version Control's Pull and Deploy option.

Photo credits

🎥 Videos:

- Git and Github for Beginners
- Learn Git in 1 Hour

🏁 Get Started:

- Git on the command line

🖇️ Documentation:

- Official Git Docs
- Official Github Docs
- Official Gitlab Docs

📖 Books:

- 16 Best Git Books of 2022 | How to Learn Git

DEVELOPMENT & OPERATIONS TOOLS | Atlassian

As you become more and more involved in software development, engineering, etc, there is a good chance you will hear about Atlassian. Founded in Sydney, Australia in 2002, the company was reconstituted 12 years later, and registered in London as Atlassian Corporation PLC.

This company develops enterprise software, wahich is primarily designed for mid and large-sized businesses that require tools for software development, project management, and content management. Probably Atlassian's most famous tools are Jira and Confluence, which are team collaboration and wiki tools. As of now, there are probably over 200,000+ customers who use their tools to empower teams and drive their missions forward. Even more impressive is that 85 of the Fortune 100 companies use Atlassian products! $$$

Let's take a brief look at Jira and Confluence in this section, so that you are familiar with their concepts and use cases.

Atlassian JIRA is a release and project tracking tool that lets you develop software efficiently and effectively. Agile development teams use Jira to customise workflows, release software, and collaborate with other teams.

It is common for software development companies to use a variety of tools, such as tools for:

1. Tracking tasks
2. Tracking and issuing release notes for users
3. Reports
4. Managers, and so on

Jira software is a tool that can handle all of these tasks. With this software, you can plan software development, monitor the progress of the same, see what each member of your team is doing and at what stage they are with their projects, and plus other additional features that you could implement in your every day's work.

🎥 **Videos:**

- Jira Tutorial for Beginners: Jira Project Management

🏁 **Get Started:**

- Official Atlassian Jira 101 - Get Started

📎 **Documentation:**

- JIRA Tutorial: What is, How to use Jira Testing Software Tool

📖 **Books:**

- Top 10 Books For Learning JIRA

• Atlassian CONFLUENCE

Organisational and team knowledge is created, recorded, shared, and used through knowledge management. Undocumented knowledge of individuals is potentially lost knowledge, which we definitely don't want to happen in the real world! Thus, irrespective of what an organisation or team does, the people have knowledge that is valuable to record, transfer, and use.

One of the most popular software solutions for knowledge management out there is another product from **Atlassian called Confluence,** which is a wiki-based system used for team collaboration.
The idea behind this is to have all the necessary documentation based in one place, and only certain team members can view documentation relevant to their teams, work on them, edit and update; ensuring there is always the latest version of the documents available out there. Furthermore, you have the ability to see who edited what, when they wrote it, and so on. Putting these tools in place and implementing them effectively will facilitate software development and increase transparency, which is always a positive step in the right direction.

As you work with your team to continuously improve your documentation, it's very helpful to be able to add comments, share ideas, discuss what has already been written, suggest improvement, correct what has already been written, and thus contribute to your colleagues' projects by correcting a document or what has been written.

The following can also be stored:

- Notes from meetings or daily-standups,
- Specify "system requirements" for projects,
- Have folders with files and documents,
- Manage user rights (who can do what, what they can see, etc.)
- And much more

It may seem empty at first, but as it fills up, its advantages become evident. Neither of us are big fans of documentation, but we love Confluence.

ATLASSIAN

www.atlassian.com

🎥 **Videos:**

- Jira Tutorial for Beginners: Jira Project Management

🏁 **Get Started:**

- Official Atlassian Get Started with Confluence

🖇️ **Documentation:**

- Official Atlassian Confluence Docs

📖 **Books:**

- Official Atlassian Confluence book

Part IV

Resources

Knowledge is acquired in different ways by different people. There is no one-size-fits-all approach to learning. Depending on the person, reading may be more effective for some, while listening or oral explanations may be more effective for others. Some people prefer learning by doing.

Because of this, it's important to not limit yourself to one form of learning when you're just starting out. You should also avoid limiting yourself to one source of material. One source might explain certain concepts well, but they may not be explained in detail, and some more advanced levels may not be included in the material. Alternatively, you could find thorough descriptions of all advanced levels in another source, though not all may be clearly explained. The bottom line is that you would never know that other sources existed if you followed just one. If you want a better explanation of the material, look at the third, fourth etc sources. It is inevitable that you will go through the same material on most of these "platforms", and only then will you be ready to move on.

Occasionally, you revisit something you have already read (new or old) just to refresh the memory of some information that you previously stored. In fact, you may not fully grasp something merely by covering the theory, but when confronted with more challenging concepts! **Do not panic** - A mentor can also make a tremendous difference in your ability to learn new technologies, and we will provide that to you here at FelpApp to ensure you are on the right path.

Having a mentor is like having someone who has experience working as a programmer or engineer and who can offer you advice and answers. When you are learning, you will encounter many challenges, and it might be difficult to find all the answers on the Internet at first until your mind is trained to do so, which takes time. During such times, you can reach out to your mentors or our community group for assistance.

This training will provide you with only information and knowledge that will be relevant to your future career. You will have the opportunity to practice what has been taught to you in theory by putting it into practice until it makes complete sense to you, and you are ready to move on.

Image by Freepik

There are a variety of resources you can use during your training, including:

E-LEARNING PLATFORMS

Despite their theoretical foundation, these platforms also emphasise practice, and most of them do not lack either. If you want more theory, you can read our recommended books that you will be provided with throughout your training covering relevant tools and technologies.

If you would like to gain more practice, you can do so with the projects we will offer you within this training, which will sometimes be reviewed by the mentors themselves, as well as your colleagues and other students from the FelpApp community. You are also encouraged and highly desired to start your own projects, as this will demonstrate your initiative.

In terms of Computer Science, one of the best courses available at present is the one offered by Harvard University; below is an overview of the course and an actual enrollment link that you can read about and take advantage of during your free time.

- CS50 FULL REVIEW - Best Course to Learn Computer Science in 2022? (edX, Harvard)
- Harvard University - CS50's Introduction to Computer Science

There is also a GitHub repo with an endless amount of resources in Computer Engineering and Programming:

- Computer Science courses with video lectures

In addition to the above, some of the most familiar platforms you will come across on your learning journey are as follows:

- Freecodecamp
- Codecademy
- A Cloud Guru
- Udemy
- Pluralsight
- Udacity
- LinkedIn Learning
- Coursera

Image by Freepik

📖 BOOKS

In these areas of Technology, the FelpApp library includes the following topics and tools:

- **Front-End Development**
- **DevOps**

🎙 TECH PODCAST

Development

- **Web Development:**
  - Syntax - Tasty Web Development Treats
  - CodeNewbie
- **JavaScript:**
  - JS Party: JavaScript, CSS, Web Development

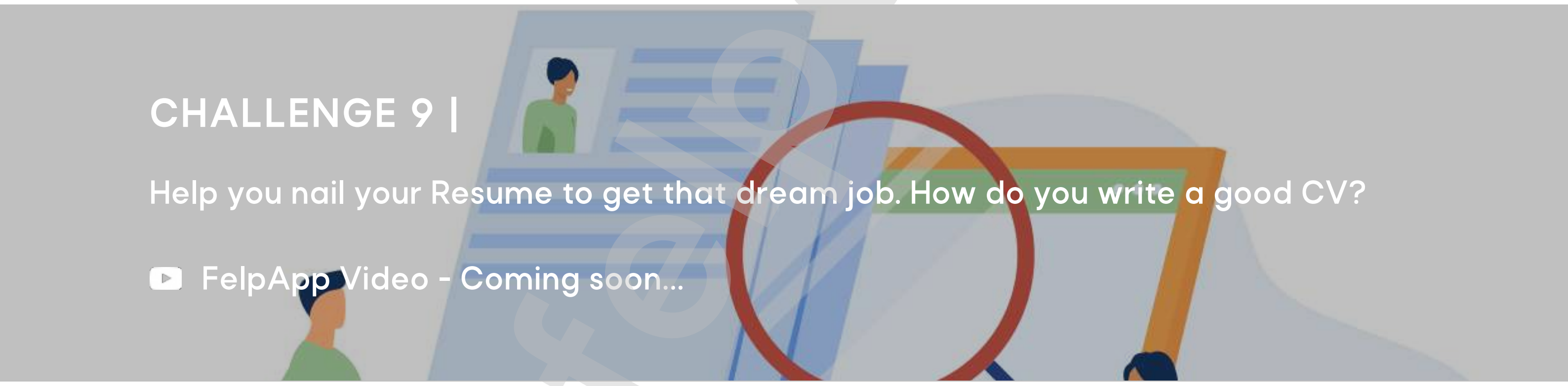
DevOps

- Ship it! DevOps, Infra, Cloud Native
- Adventures in DevOps
- **Linux**
  - The Linux Cast
- **Amazon Web Services (AWS)**
  - AWS Podcast
- **Microsoft Azure**
  - Azure Friday
- **Google Cloud Platform (GCP)**
  - Google Cloud Platform Podcast
- **Docker**
  - DevOps and Docker Talk

11:04
4G
THE LINUX CAST
14:58
-1:24:53
Are Containers the Future of Lin
The Linux Cast — 2 de julio de 20
15
30
x1

Résumé & Job Hunting

For a starting point, you may want to research what a CV is and what its importance is to your employer before thinking about how to write one.
In the simplest terms, a CV is a business resume. A CV (Latin Curriculum Vitae, English Resume) or resume for work is a mandatory part of every business application, and its purpose is to present you and your professional qualifications to a prospective employer.
An outline of everything that makes you an ideal candidate for a particular position. As a candidate for a new job, it is essential to make sure that your CV is well-written, relevant to the job, and meaningful so that you will be able to attract potential employers to your resume. It is your opportunity to present yourself in the most positive light and distinguish yourself from the competition. Invest the time into writing your CV, because a good CV can make or break your chances of success. Therefore, CV is not just your job application. Quite simply, your CV is a ticket to the world you dream of joining.



CHALLENGE 9 |

Help you nail your Resume to get that dream job. How do you write a good CV?

FelpApp Video - Coming soon...

A CV is like the saying: you only get one chance to make a first impression. As appearances often play an important role in first impressions, make sure that your CV conveys the right image of you. It is therefore essential to pay close attention to every detail when writing a CV. Here are some tips to help you make your job resume stand out:

- **Formatting** - Studies show that employers spend 20 to 30 seconds reading resumes. It will therefore be easier for an employer to find what they are looking for if it is formatted well.

- **Strengths** - Put emphasis on your strengths by using words such as “created”, “led”, and “analysed” when describing your achievements. Your initiative can be viewed as a positive attribute.
- **Honesty** - Ensure that the information you provide is accurate. Maintain an honest attitude. Your information should set you apart from the crowd and present you in the best light, but above all, it should be accurate.
- **Experience** - Organise your work experience in reverse chronological order so that a potential employer can see your latest positions and accomplishments first.
- **Image** - If possible, take a business-like picture with a professional photographer. You should also pay attention to your email address - it should also indicate professionalism.
- **Focus** - In your CV, list your personal qualities that are relevant to the job you are applying for. Keep your focus on the important things.
- **Length** - CVs should not be too long. One or two sides are the standard length.

Image by Freepik

If you’re looking for a tool to help you build your CV, here’s one that you’ll love: Zety

Here at FelpApp we will help you:

- Get The Skills You Need
- Make your portfolio stand out
- Professional resume + LinkedIn profile
- Start Applying
- Get the job of your dreams
- Continue developing your skill sets

🎥 Videos:

- StandOut CV: How to write a CV [Get noticed by employers]
- Programmers - Listen Up
- How I Learned to Code in 4 MONTHS & Got a Job Offer (no CS Degree)

🏁 Get Started:

- How to Prepare Your Beginner Resume: Tips and Examples
- FreeCodeCamp: Prepare for coding interview

🖇 Documentation:

- Medium: AnnFelix - What your resume is up against?

📖 Books:

- 20 Best CV and Resume Books of All Time

💡 Cheat Sheet:

- CV Cheat Sheet

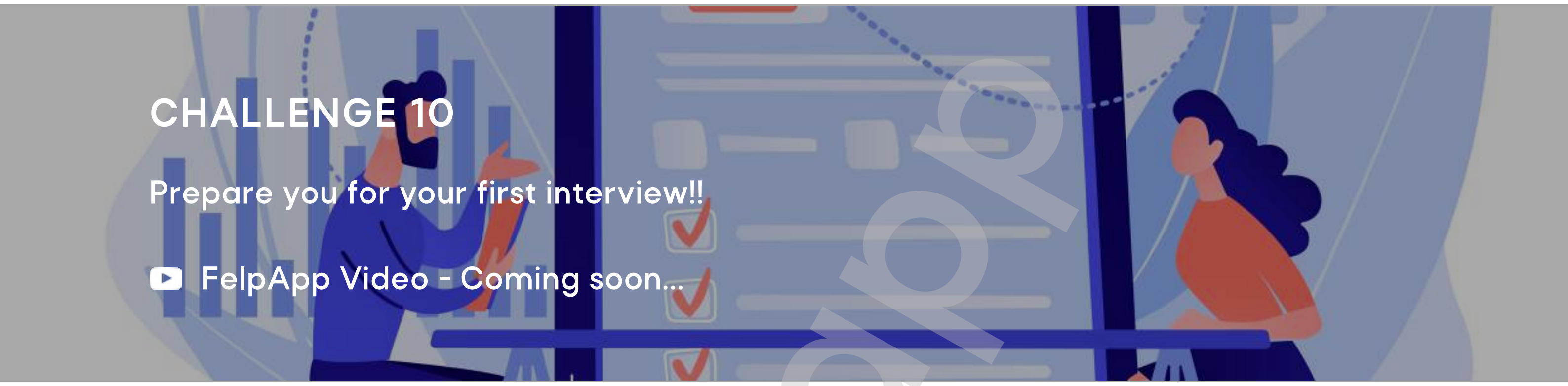
🔎 References:

- 9 things you should always include in a CV
- What to Include in a CV [in 2022]

1, 2, 3 - INTERVIEW!

Preparation for the interview allows you to successfully answer the interview questions, as well as reduce the amount of stress and uncertainty that you may feel before and during the interview. Job interviews can be intimidating, especially if it’s your first one.

Similarly to writing a CV, it is necessary to prepare for a job interview in detail, as it is your best chance to impress the employer. In the recruitment selection process, the interview is the second stage. An interview invitation means that your application documents were well written and successful.



CHALLENGE 10

Prepare you for your first interview!!

▶ FelpApp Video - Coming soon...

A job interview can be stressful, as you are put in the spotlight so that the employer can decide if you are a good fit. Whether your application is successful or not will depend on the answers you provide in your interview. A thorough and well-prepared approach is one of the most effective ways to reduce stress in such situations. As a result, you will be able to answer any questions the employer may have and, therefore, leave a huge impression on the employer.

At FelpApp, we will be focusing on the following topics when you get to this stage of your interview preparation:

- Our first step will be to learn as much as we can about the company you applied to before the actual interview.
- Leave an impression at the interview by preparing one or two meaningful questions.
- During an interview, express interest in the position and the company itself, such as what they do, where they are based, what’s the structure of the team etc.

- Answer questions in a flexible manner. Consider potential problems and general questions. This will allow you to adjust your answers much more easily during the interview itself, and will enable you to handle several specific questions more effectively.

👉 Here's a video you should check out on Tips From a Top Tech Recruiter | How To Land a Tech Job With No Experience? Bootcamps vs. Degrees?

Telephone Interview 📱

A telephone interview is another method of interviewing candidates. These interviews are typically used as a preliminary or first round of selection.
The purpose of a phone interview is to confirm, based on your answers, that all the items on your resume are true and to determine your seriousness as a job candidate. This also allows the recruiter to make an early assessment of your personality.

Online Interview 🖥️

Job interviews can often be conducted online using one of these apps: Zoom, Teams, Google Meet, etc. Whether you are preparing to go abroad or you have applied for a remote position in a foreign company, you will certainly experience an online job interview. What are the specifics of this type of interview, and how can you convince the employer that you are the best choice?

You can expect most of the same rules to apply to an online interview as to a standard interview, so just be yourself, relaxed, and prepared.
Furthermore, these interviews are generally conducted in English or another language, depending on the country you are applying to.

In order to better prepare for the online interview, ensure that you:

- You should verify that the camera, microphone, speakers, and headphones are working before you go to the interview.
- Ensure your laptop has a sufficient battery life.
- Make sure you follow the dress code.
- Think of questions you would like to ask the employer.
- Be prepared with answers to questions that may arise.
- Pay attention to non-verbal communication/body language.
- If you want to avoid disruptions, inform your housemates about the interview.

Make a difference
Be the difference

“They say that it is not the strongest members of a species that survive, nor the most intelligent ones, but rather those that adapt best to change. We are witnessing huge changes taking place on our beautiful planet Earth. It is entirely up to us whether or not we are able to adapt to that rhythm.

This is excellent news, since a decision of this importance should not be left to others. Everything we need, we already have within us. In the end, it comes down to how flexible we are in bending our ego and starting from scratch. This time, however, you are not starting from scratch, but from experience. It is the experiences of others that have been shared with you selflessly in this book. To serve you in your growth on the path of personal development.

All you need to do is choose which species you would like to belong to. One that keeps going or one that fades into oblivion. In order to find the answer to your question, you should listen to your heart. It is only through self-love that we can create the life of our dreams.”

The Munja
Inspirational speaker

Photo by Ana Quirosa

FelpApp Challenge

Part 1.

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88
89	90						

FelpApp Challenge

Part 2.

91	92	93	94	95	96	97	98
99	100	111	112	113	114	115	116
117	118	119	120	121	122	123	124
125	126	127	128	129	130	131	132
133	134	135	136	137	138	139	140
141	142	143	144	145	146	147	148
149	150	151	152	153	154	155	156
157	158	159	160	161	162	163	164
165	166	167	168	169	170	171	172
173	174	175	176	177	178	179	180

FelpApp Challenge

I __________ pledge to complete the 180 days of the FelpApp Challenge.

I will make this a priority in my life and hold myself accountable, despite my busy schedule.

I commit to learning new skills for 180 days (6 months) at least one hour a day.

I will remain dedicated to my goals, no matter how tired or frustrated I am on certain days.

I will succeed in my goal, despite obstacles.

I will become what I desire to become.

It is in my power to succeed.

Signature: __________

Date Signed: __________

FelpApp
www.felpapp.com