

Learning

After Le Wagon

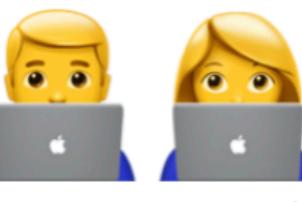


le wagon

We saw this

in “Landing your First job”

Structure your day

- 1h active « job search »
- 1h answering / writing mails 
- **4-6h coding** on your projects / learning 
- 1-2 meetings / lunch / coffee per week 



Keep Coding.

Every working day!

- Knowledge leaves quickly
- Adopt a learning mindset
- Write down what you learn
- Work in a group or with a buddy



Be Organised.

Use your time wisely

- Make a list to learn each month
- Think before you start a course
- Learn in-demand skills
- Don't get sidetracked by fads
- Follow Le Wagon's post-bootcamp track!



Become more **employable.**

LW Challenges

Consolidate your learnings from
the bootcamp

New Skills

Learn new skills after the bootcamp to
upgrade your CV

Portfolio

Build small projects to showcase what
you have learned.



Bootcamp Challenges

Review key challenges on
the bootcamp



Data Challenges.

Data Analyst

- Olist
- Data Visualisation
- NY Taxi Fare

Data Engineer

- Olist
- NY Taxi Fare
- Deployment
on GCP

ML Engineer

- Olist
- NY Taxi Fare
- ML Week

Data Scientist

- Olist
- Data Visualisation
- NY Taxi Fare
- ML & DL Weeks

Web Dev Challenges.

Fullstack

- Louchebem
- Orange Tree
- Cookbook (from scratch)
- Movie Watchlist (build a final version)

Backend

- Louchebem
- Orange Tree
- Cookbook (from scratch)
- Food Delivery (from scratch)

Frontend

- Cookbook (from scratch)
- Personal Page (from scratch)
- Movie Watchlist (build a final version)

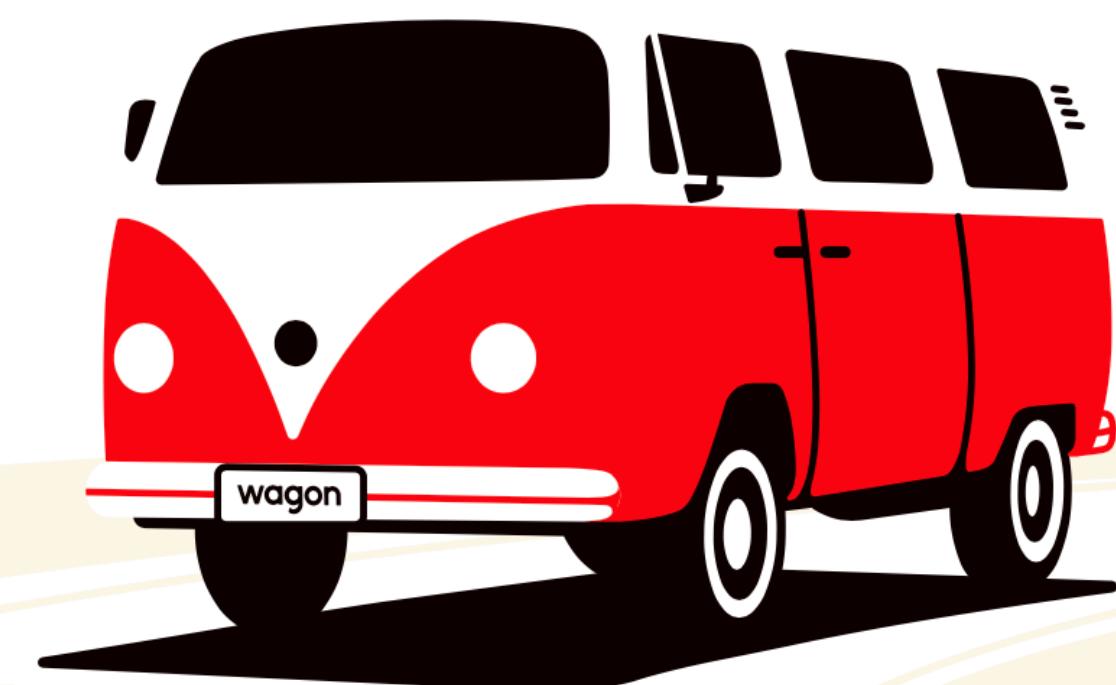
Product Challenges.

Product Manager

- Product Design Sprint
- User Journey with Whimsical
- Final project Figma

New Skills

Areas to keep learning after
the bootcamp



Data Skills.

Data Analyst

Advanced SQL

Data visualisation

Product Analytics

Business Intelligence

A/B Testing

Data Engineer

Level-up Python

Advanced GCP

Data Warehouses

Containerisation

CI & CD

Data Pipelines

ML Engineer

Level-up Python

Advanced ML

Data Warehouses

Advanced GCP

Data Scientist

Level-up Python

ML/DL specialisation

Data Warehouses

Data Pipelines

Big Data

Web Dev Skills.

Fullstack

- Testing in Ruby/Rails
- JS Framework (React)
- Building APIs
- CI & CD
- 2nd backend language

Backend

- Eloquent Ruby
- Testing in Ruby/Rails
- Building APIs
- CI & CD
- 2nd backend language
- Docker, Kubernetes

Frontend

- Modern frontend
- JS Framework (React)
- NPM & Webpack
- Progressive Web Apps
- React Native

Product **Skills.**

Product Manager

- APIs
- Agile, Scrum, Kanban
- Roadmapping
- User Testing
- AARRR Framework
- Prototyping

Where to go next

Specializing



(A) Frontend

Notion Gameplan Template ↗



Frontend

The priorities are to understand the bootcamp fundamentals from the bootcamp and start your portfolio

- Subscribe to [Javascript weekly](#)
- Subscribe to [Frontend weekly](#)
- Subscribe to [GoRails Le Wagon Perk: Discount for the Pro subscription](#) ↗ [here](#)
- Redo key exercises from Le Wagon: Cookbook, Rails Mr Cocktail
- Ensure you understand [REST principles](#)
- Learn how to debug like a pro in [Javascript](#)
- Ensure you understand how the [internet works](#)
- Learn more Javascript and manipulation of the DOM by completing half of [Javascript 30](#)



Frontend

The priority is to learn React (or another Javascript framework):

- Subscribe to the [React newsletter](#)
- Understand the fundamentals of React with [React Tutorial App](#), [Educative](#) or [Wes Bos's React For Beginners \(Starter Package\)](#)
- Understand more about package managers ([npm](#)), task runners ([npm scripts](#)) and module bundlers ([webpack](#))
- Build your first project in React and add to your portfolio
- Learn more about modern CSS with [styled components](#) and [CSS modules](#)



Upskill (2 - 3 months)

Frontend

Explore (1 - 2 months)

The priority is to learn about some of the key areas in frontend today:

- Understand what Typescript is as a language on top of Javascript with an [Educative course](#)
- Understand more about Progressive Web Apps (PWAs) with an [Educative course](#)
- Understand [GraphQL](#) and the difference with REST principles
- Learn more about mobile with [React Native](#)



(A) Frontend

★ HTML ★

- ✓ What makes HTML the main building block of the web?
- ✓ What separates good markup from bad markup?
- ✓ Why some tags are “semantic” and some are not?
- ✓ What tools does HTML give us to ensure a page can accommodate all users?



(A) Frontend

⭐ CSS & Methodologies ⭐

- ✓ What does a “cascade” mean in CSS? What are its problems?
- ✓ What would you do differently about CSS?
- ✓ Why is it important to care about browser support for different properties?
- ✓ CSS or JS for animations / simple UI elements?
- ✓ What is BEM? Why is naming and handling CSS so important?
- ✓ How do you make pages accessible and internationalised?



(A) Frontend

★ Javascript ★

- ✓ Why did JS become so popular?
- ✓ What are the “good parts” and “bad parts” of JS?
- ✓ What is ECMAScript? What are the main features of ES6?
- ✓ How is OOP done in Javascript and why it so different?
- ✓ What separates frontend and backend Javascript?



(A) Frontend

★ JS Frameworks ★

- ✓ What are the backgrounds to Vue, React, and Angular?
- ✓ What are their similarities and differences?
- ✓ What is a Virtual DOM?
- ✓ Which problems do modern front end framework solve and which problems do they introduce?



(A) Frontend

★ Bundling assets ★

- ✓ How do you deliver assets to a client?
- ✓ What are bundlers and which ones are most popular?
- ✓ How do you save bandwidth and increase page load speeds?
- ✓ Which problems arise with a JS-heavy application?
- ✓ How do Webpack and others attempt to solve them?



(B) Backend

[Notion Gameplan Template ↗](#)



Backend

The priorities are to understand the bootcamp fundamentals from the bootcamp and start your portfolio

- Subscribe to [Ruby Weekly](#)
- Subscribe to [Drifting Ruby](#)
- Subscribe to [GoRails](#) Le Wagon Perk: Discount for the Pro subscription  [here](#)
- Review and redo key exercises from Le Wagon: Louchebem, Orange Tree, Cookbook, Rails Mr Cocktail
- Ensure you understand [REST principles](#)
- Learn how to debug like a pro in [Ruby](#) and [Javascript](#)
- Learn how to write [Eloquent Ruby](#)
- Build a simple REST API with bare Sinatra to learn about endpoints, status codes, request/response types and HTTP message parsing - follow this demo from [Andrew Havens](#)



Review (1 month)

Backend

The priorities are to push your knowledge of Ruby and Rails:

- Learn more about [Test Driven Development](#) and then look at testing in [Rails](#)
- Learn more about [Continuous Integration/Delivery](#) to automate deployments and code checks
- Learn how to debug like a pro in [Ruby](#)



Backend

Explore (2 - 3 months)

The priority is to explore different technologies and learn a second programming language:

- Translate your knowledge into a second backend programming language: **Python** with [Educative](#) or [CS50 video series](#) (used in startups and corporations and very similar to Ruby), **Node.JS** with [Educative](#) or [Wes Bos](#) (used in startups and great if you're a Javascript fanatic!) or **Java** with [Educative](#) (useful for larger corporations and fintechs)
- Learn more about DevOps starting with our own Docker tutorial and then moving on to more technologies such as Kubernetes and Jenkins with a [full course](#).
- Understand [GraphQL](#) and the difference with REST principles



(B) Backend

★ Ruby ★

- ✓ What makes Ruby stand out?
- ✓ Why is it called dynamic?
- ✓ What is “duck typing” and what are its pros and cons?
- ✓ What are the main principles of OOP in general and how are they used in Ruby?
- ✓ What is metaprogramming?
- ✓ How do you create a gem?



>



(B) Backend

★ Rails ★

- ✓ What is a web framework?
- ✓ What are the main design principles behind Rails? Which problems do they solve?
- ✓ What are the main frameworks inside Rails?
- ✓ How does Rails handle HTTP request-response cycle?
- ✓ How do you run tests in Rails with RSpec and Minitest.
- ✓ What are the advantages and disadvantages of “monolithic” web apps?
- ✓ How did the framework evolve over time?



(B) Backend

★ Databases ★

- ✓ What kinds of databases do you know?
- ✓ What is a relational database?
- ✓ Which SQL databases have you heard about? How are they different?
- ✓ Why did PostgreSQL become so popular.
- ✓ What are advantages and disadvantages of Active Record?



(B) Backend

★ Operating Systems ★

- ✓ What is the philosophy behind UNIX-based operating systems?
- ✓ What are vim commands? How do you use them?
- ✓ What are the following: Files, directories, processes, threads, sockets, ports, kernel, system calls, memory management?

(B) Backend

★ Developer Operations (DevOps) ★

- ✓ What is a deployment?
- ✓ How can one host an application?
- ✓ What are the most popular clouds?
- ✓ What are containers? How are they different from virtual machines?
- ✓ How does Docker work? How does it close a gap between development and production environments?



(B) Backend

★ Networking ★

- ✓ How HTTP is different from TCP/IP? What are the different network layers?
- ✓ What is a client and what is a server? What is: an application server? Reverse proxy? Load balancing?
- ✓ What is a TCP packet? HTTP request?
- ✓ How do different types of requests work? What is idempotency?
- ✓ What are the REST principles?



(B) Backend

★ APIs ★

- ✓ What are HTTP response codes
- ✓ What are the best practices in web API design.
- ✓ How do you build a simple API that performs CRUD actions on a single resource, handles client and server errors, and sets corrects response codes?
- ✓ How does GraphQL work?.



(C) Product

Notion Gameplan Template ↑



Product Manager

The priorities are to review the bootcamp and get a strong understanding of the different areas of product management:

- Get in the know with the world of tech by subscribing to [Hacker News](#) and [Techcrunch](#)
- Start discovering new products with [Product Hunt](#)
- Start understanding how products are built with - [Masters of Scale](#) with Reid Hoffman and [How I Built This](#) with Guy Raz
- Review and redo the Product Design Sprint from Le Wagon: Product Pitch, User Journey with Whimsical, full prototype Figma
- Understand modern startups with [Lean Startup](#)
- Start understanding the different areas of Product Management with [The Product Book](#)
- Build a portfolio website showcasing your projects from Le Wagon (Final project with Figma, Whimsical and App, then add your API)
- Start understanding APIs more with this lecture from [Zapier](#)



Product Manager

The priorities are to understand management methodology and get fundamental skills in business and UX design.

- Start a Coursera certification on [Digital Product Management](#)
- Skill up your product design skills and subscribe to the [Intercom Blog](#) and [Invision design Blog](#)
- Learn more about the [Product Lifecycle](#)
- Understand product management methodology: [Agile](#) vs [Waterfall](#)
- Understand how to use [Kanban boards](#)
- Find out more about [SCRUM principles](#)
- Learn about product roadmaps prioritisation and ticketing with [Jira](#)
- Understand more about [user testing](#)
- Understand the user lifecycle with [AARRR framework](#) (known as “pirate metrics”)
- Improve your [prototyping skills](#)
- Get familiar with the best product tools such as [Google Analytics](#) (product analytics), [MixPanel](#) (product analytics), [Jira](#) (roadmapping), [Typeform](#) (customer surveys), [Zapier](#) (no-code integrations), [Hotjar](#) (heatmapping), [Intercom](#) (user feedback). Explore similar tools to them and use demos to understand what they do.



Product Manager

Explore (1 - 2 months)

The priorities are to build your own mini startup and upskill yourself in technical concepts

- Challenge yourself to build a small Product Hunt project in under a week - use the tools and principles above
- Improve your contextual knowledge with the following books: [Don't Make Me Think](#) by Steve Krug (for user obsession), [Radical Candor](#) by Kim Scott (for management) and - [Hooked](#) by Nir Eyal (for product engagement)
- Improve your technical knowledge by understanding the fundamentals of React, cloud, servers, containerisation, microservices, mobile vs web, scalability, data pipelines and vulnerabilities. Watch 30 minutes of youtube videos on each.



(C) Product

★ Function ★

- ✓ What are the main roles of a product manager?
- ✓ How do they differ to an engineer and designer?
- ✓ How are they at the intersection of business, design and technology?
- ✓ What is your USP as a product manager?



(C) Product

★ Methodologies ★

- ✓ What are the main methodologies used in management?
- ✓ What is Scrum and how does it work?
- ✓ What are the core principles of lean?



(C) Product

⭐ Users & Data ⭐

- ✓ How does a company track their users?
- ✓ How do you learn more about your user base?
- ✓ What are the benefits of qualitative and quantitative data?
- ✓ How can you use data to discover product opportunities?
- ✓ What are the key tools / apps a PM uses?

(C) Product

⭐ Planning ⭐

- ✓ How do you create a product roadmap?
- ✓ How do you prioritise on what to build next?
- ✓ What are the most popular road-mapping tools and how do you use them?



(D) Fullstack

[Notion Gameplan Template ↑](#)



Fullstack

The priorities are to understand the bootcamp fundamentals from the bootcamp and start your portfolio

- Subscribe to [Ruby Weekly](#)
- Subscribe to [Drifting Ruby](#)
- Subscribe to [GoRails](#) Le Wagon Perk: Discount for the Pro subscription  [here](#)
- Sign up to [Javascript weekly](#)
- Review and redo key exercises from Le Wagon: Louchebem, Orange Tree, Cookbook, Rails Mr Cocktail
- Learn more about Test Driven Development in [Ruby](#) and in [Rails](#)
- Learn how to debug like a pro in [Ruby](#) and [Javascript](#)
- Learn how to write [Eloquent Ruby](#)
- Ensure you understand [REST principles](#)
- Build a simple REST API with bare Sinatra to learn about endpoints, status codes, request/response types and HTTP message parsing - follow this demo from [Andrew Havens](#)



Review (1 month)

Fullstack

The priorities are to push your knowledge of Javascript and learn a frontend framework:

- Learn more about [Continuous Integration/Delivery](#) to automate deployments and code checks
- Learn more Javascript and manipulation of the DOM by completing half of [Javascript 30](#)
- Understand the fundamentals of a JS framework - we suggest React with [React Tutorial App](#), [Educative](#) or Wes Bos's [React For Beginners \(Starter Package\)](#)
- Build your first project in React and add to your portfolio



Fullstack

Explore (2-3 months)

The priority is to explore another backend language and upgrade your fullstack knowledge

- Translate your knowledge into a second backend programming language: **Python** with [Educative](#) or [CS50 video series](#) (used in startups and corporations and very similar to Ruby), **Node.JS** with [Educative](#) or [Wes Bos](#) (used in startups and great if you're a Javascript fanatic!) or **Java** with [Educative](#) (useful for larger corporations and fintechs)
- Understand more about package managers ([npm](#)), task runners ([npm scripts](#)) and module bundlers ([webpack](#))
- Understand [GraphQL](#) and the difference with REST principles
- Learn how frontend frameworks get data from the backend (REST API, GraphQL) and try to build a simple app that provides full frontend/backend separation: backend API in Ruby, frontend app in JS
- Learn more about DevOps starting with our own Docker tutorial and then moving on to more technologies such as Kubernetes and Jenkins with a [full course](#).



(E) Data Analyst



Data Analyst

Review (1 month)

The priorities are to understand the bootcamp fundamentals from the bootcamp and improve your SQL:

- Write a medium blog post describing your Final project!
- Sign up to Mode's [newsletter](#)
- Start listening to some Data podcasts [Data Science Weekly](#) and [Data Skeptic](#)
- Redo key exercises from Le Wagon: Olist and NY Taxi Fare.
- Start pushing your SQL skills via [SQLHabit](#)
- Add to your Le Wagon (Final project, O-List)



Data Analyst

The priorities are to understand more about data visualisation, product analytics and databases

- Become familiar with Product Analytics tools such as [Google Analytics](#)
- Learn the fundamentals of one BI tool, such as [Looker](#), [Tableau](#) or [Power BI](#)
- Learn about [A/B testing in Python](#)
- Understand what DB Management systems are and what they do. Learn a bit about [SQLite](#) or [MySQL](#)
- [Participate in a Kaggle competition](#) and add to portfolio



Upskill (2-3 months)

Data Analyst

Explore (1-2 months)

The priorities are to level up your programming and explore some other technologies:

- Level-up programming with Python on [HackerRank](#)
- Understand what a Data Warehouse does. Learn a bit about [Google Big Query]((<https://cloudacademy.com/learning-paths/google-bigquery-44/>) or [Snowflake](#)
- Improve your knowledge of [containerisation](#) and Docker



(F) Data Engineer



Data Engineer

Review (1 month)

The priorities are to understand the bootcamp fundamentals from the bootcamp and start your portfolio:

- Write a medium blog post describing your Final project!
- Sign up to newsletters: [ML In Production](#), [NLP News fro Ruder](#) and [TWIML](#)
- Start listening to some Data podcasts [Data Science Weekly](#), [Data Skeptic](#) and [TWIML Podcast](#)
- Redo key exercises from Le Wagon: Olist, NY Taxi Fare
- Build a portfolio website showcasing your projects from Le Wagon (Final project, O-List)
- Level-up programming with Python on [HackerRank](#)



Data Engineer

The priorities are to go deeper into the fundamentals of data engineering and understand cloud architecture and deployment:

- Upskill your GCP knowledge with a course on [GCP Architecture](#) the [Cloud Engineer exam](#). *Sign up for a discount [here](#)*
- Level-up programming with Python on [HackerRank](#)
- Understand what a Data Warehouse does. Learn a bit about [Google Big Query]((<https://cloudacademy.com/learning-paths/google-bigquery-44/>) or [Snowflake](#))
- Ensure you understand [REST principles](#)
- [Participate in a Kaggle competition](#) and add to portfolio



Data Engineer

Go deeper into production and deployment:

- Work through DataCamp's [Introduction to Airflow in Python](#) course
- Improve your knowledge of [containerisation and Docker](#)
- Improve your knowledge of [Applied ML in Production](#)
- Learn more about [Continuous Integration/Delivery](#) to automate deployments and code checks



Explore (2-3 months)

(G) Machine Learning Engineer



Machine Learning Engineer

Review (1 month)

The priorities are to understand the bootcamp fundamentals from the bootcamp and start your portfolio:

- Write a medium blog post describing your Final project!
- Sign up to newsletters: [ML In Production](#), [NLP News fro Ruder](#) and [TWIML](#)
- Start listening to some Data podcasts [Data Science Weekly](#), [Data Skeptic](#) and [TWIML Podcast](#)
- Redo key exercises from Le Wagon: Olist, NY Taxi Fare, ML Week
- Build a portfolio website showcasing your projects from Le Wagon (Final project, O-List)
- Level-up programming with Python on [HackerRank](#)



Machine Learning Engineer Upskill (2-3 months)

The priorities are to go deeper into machine learning or deep learning and understand more about cloud architecture:

- Push your Machine Learning knowledge further with O'Reilly's [Hands On Machine Learning](#)
- Take a deeper dive into Deep Learning with [deeplearning.ai](#)
- Improve your knowledge of [Applied ML in Production](#)
- Understand what a Data Warehouse does. Learn a bit about [Google Big Query](#) or [Snowflake](#)
- [Participate in a Kaggle competition](#) and add to portfolio



Machine Learning Engineer Explore (2-3 months)

The priorities are to go deeper into data engineering:

- Upskill your GCP knowledge with a course on [GCP Architecture](#) the [Cloud Engineer exam](#). *Sign up for a discount [here](#).
- Ensure you understand [REST principles](#)
- Learn more about [Continuous Integration/Delivery](#) to automate deployments and code checks
- Improve your knowledge of [containerisation](#) and Docker



(H) Data Scientists



Data Scientists

Review (1 month)

The priorities are to understand the bootcamp fundamentals from the bootcamp and start your portfolio:

- Write a medium blog post describing your Final project!
- Sign up to newsletters: [ML In Production](#), [NLP News fro Ruder](#) and [TWIML](#)
- Start listening to some Data podcasts [Data Science Weekly](#), [Data Skeptic](#) and [TWIML Podcast](#)
- Redo key exercises from Le Wagon: Olist, NY Taxi Fare
- Build a portfolio website showcasing your projects from Le Wagon (Final project, O-List)
- Level-up programming with Python on [HackerRank](#)



Data Scientists

The priorities are to go deeper into data architecture and modeling:

- Understand what a Data Warehouse does. Learn a bit about [Google Big Query](#) or [Snowflake](#)
- Work through DataCamp's [Introduction to Airflow in Python course](#)
- Learn the fundamentals of one BI tool, such as [Looker](#), [Tableau](#) or [Power BI](#)
- Learn about [A/B testing in Python](#)
- Understand what DB Management systems are and what they do. Learn a bit about [SQLite](#) or [MySQL](#)
- Understand more about [Kafka](#) and data pipelines
- Become familiar with data processing with [Apache Spark](#)
- [Participate in a Kaggle competition](#) and add to portfolio



Upskill (2-3 months)

Data Scientists

Explore (2-3 months)

The priorities are to go deeper into machine learning and deep learning:

- Take a deeper dive into Deep Learning with [deeplearning.ai](#)
- Push your Machine Learning knowledge further with O'Reilly's [Hands On Machine Learning](#)
- Improve your knowledge of [containerisation and Docker](#)



A day in the life...

2 minute videos of each position:

<https://career-paths.netlify.app/>

