

# Web <3 Data Science

Com els programadors web i els de dades treballen junts

---

# Aina Palacios

- Enginyera de Telecomunicacions especialitzada en Audiovisuals
- Màster en Tecnologies Avançades especialitzada en deep learning en Multimèdia!
- Experiència en programació web i machine learning.
- Mentora a IT Academy de DS



<https://www.linkedin.com/in/ainapc/>



ainaPali#2617

# Que fa un web?



## Frontend

Focuses on layout, animations, content organization, navigation, graphics.

**Programming languages:**  
JavaScript, HTML, CSS



## Backend

Focuses on building code, debugging, database management.

**Programming languages:**  
Node.js, Python, Java

# Que fa un fullstack?



# Que fa un Data Scientist?

## Data Scientist



uses statistics and machine learning to make predictions and answer key business questions

**Skills** - Math, Programming, Statistics



**Tech** - SQL, Python, R, Cloud

## Data Engineer



build and optimize the systems that allow data scientists and analysts to perform their work

**Skills** - Programming, BigData & Cloud



**Tech** - SQL, Python, Cloud, Distributed Computing

## Data Analyst



deliver value by taking data, communicating the results to help make business decisions

**Skills** - Communication, Business Knowledge



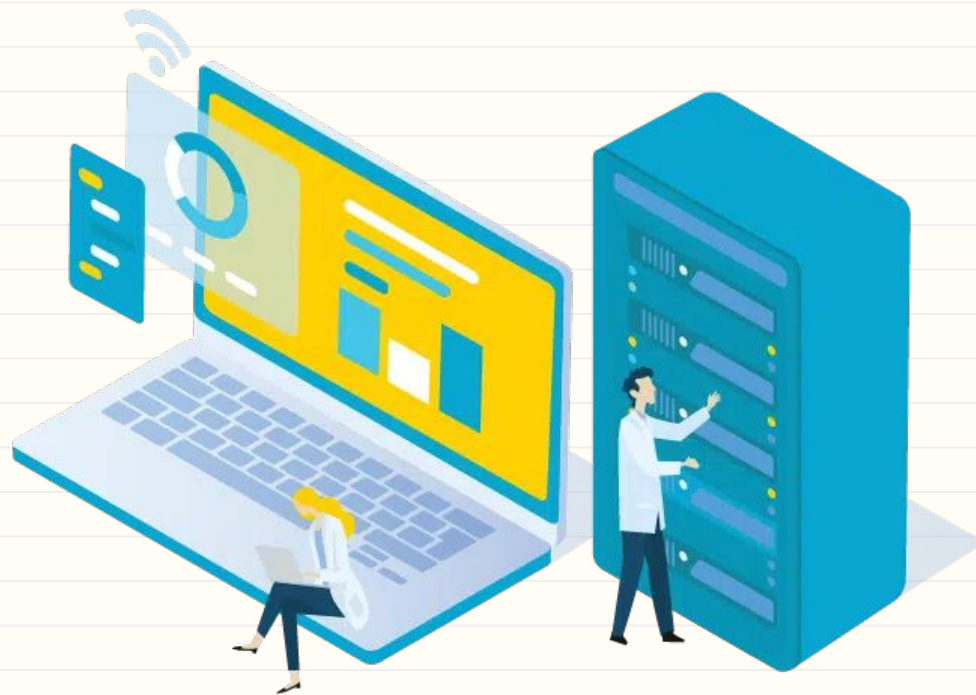
**Tech** - SQL, Excel, Tableau

# Com podem treballar junts?

1. **Desenvolupament d'aplicacions web basades en dades:** L'equip de programació web pot col·laborar amb l'equip de Data Science per desenvolupar aplicacions web que permetin visualitzar i analitzar les dades. Per exemple, poden crear una aplicació web que mostri les vendes d'un producte en un període de temps determinat.
2. **Integració de models de Machine Learning:** L'equip de Data Science pot desenvolupar models de Machine Learning que s'integrin en una aplicació web. Per exemple, poden desenvolupar un model que predigui si un client abandonarà o no un lloc web de comerç electrònic. L'equip de programació web pot integrar aquest model a l'aplicació web i mostrar els resultats a l'usuari.
3. **Desenvolupament de dashboards de dades:** L'equip de programació web pot col·laborar amb l'equip de Data Science per desenvolupar dashboards que mostren dades en temps real. Per exemple, poden desenvolupar un dashboard que mostri el tràfic d'un lloc web en temps real.
4. [...]

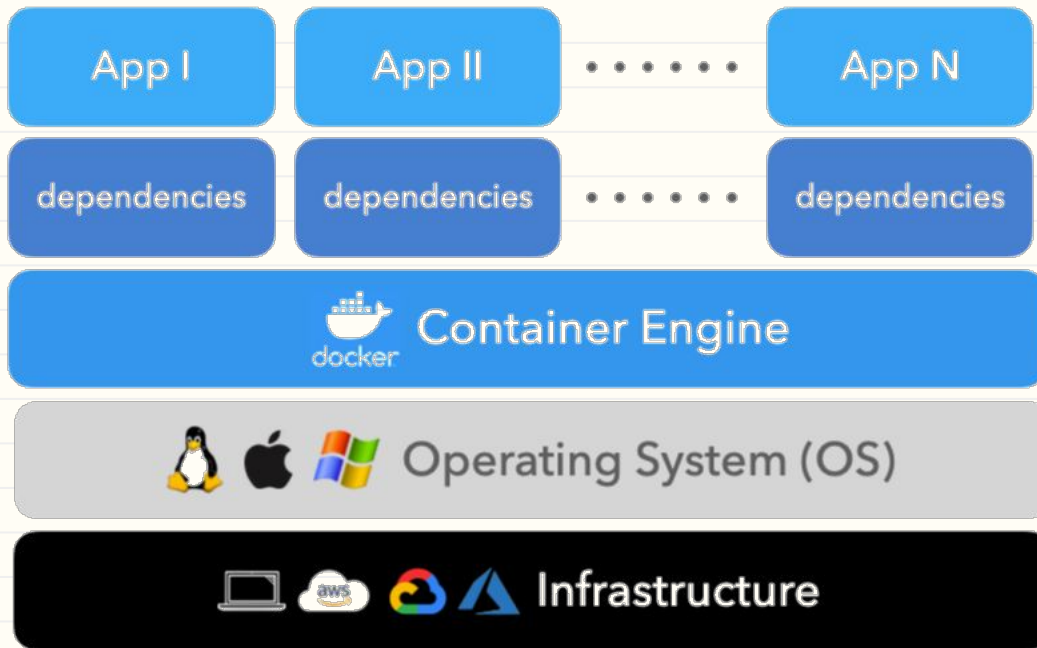


# Servidores!

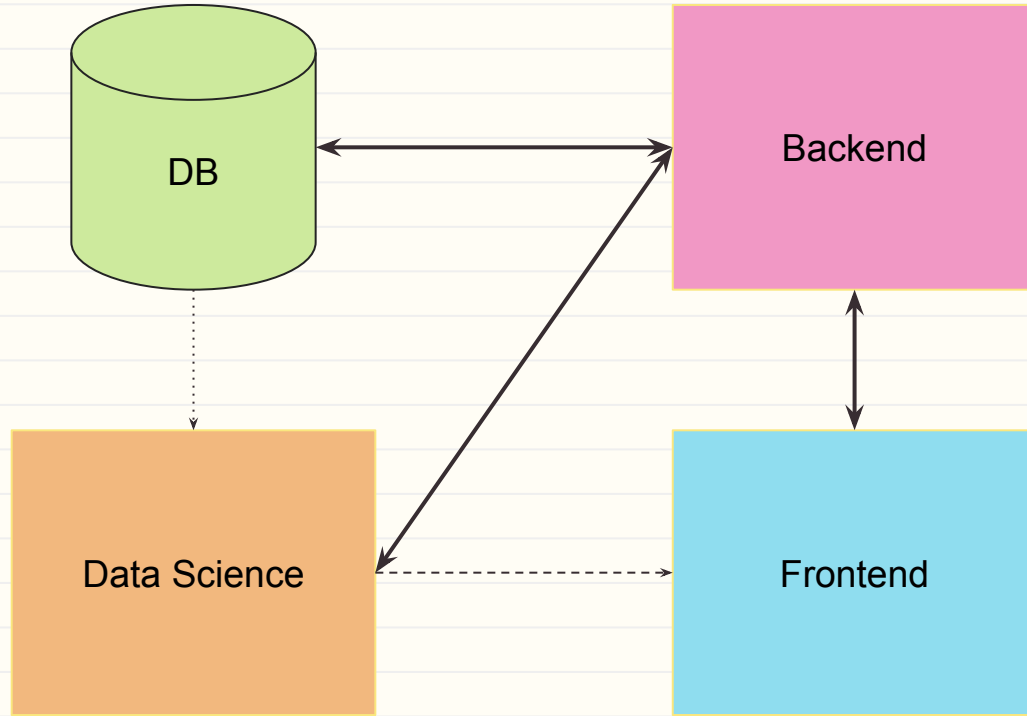




# Docker



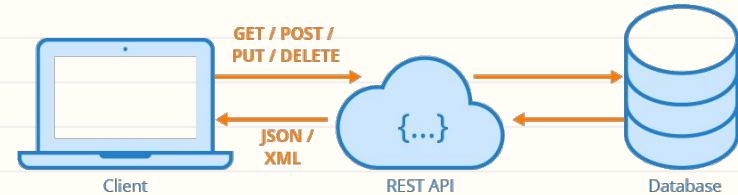




# API

**Application Programming Interface** -> interfície que permet la intercomunicació entre dos sistemes. Els mètodes d'aquestes ja estan predefinides!

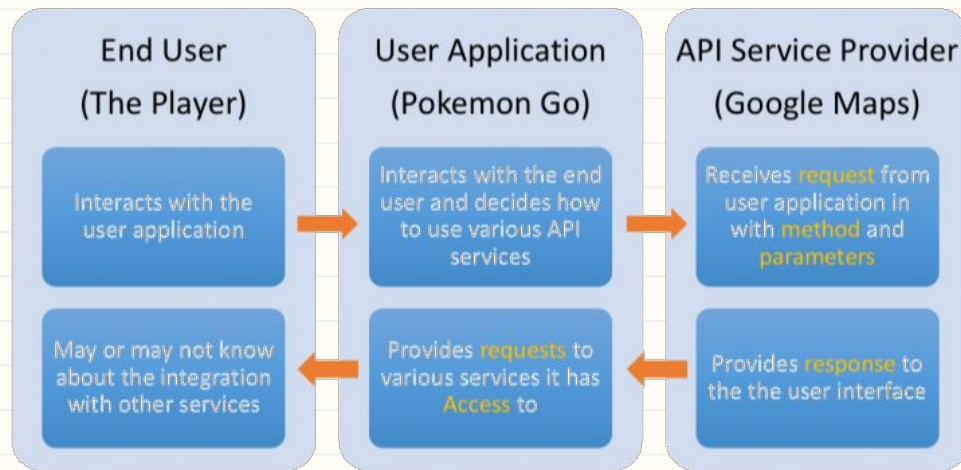
- És independent del llenguatge utilitzat
- Permeten l'entrada a un software que no ha de ser necessàriament de la teva propietat, i fer ús d'aquest sense haver-lo de tenir.
- No necessites saber com funciona l'altre software!



# Perque serveix la API?



- Intercanvi de dades o funcionalitats entre dos sistemes diferents! (Ex. El temps)
- Pots integrar la funcionalitat al teu programa.
- Pots accedir a una base de dades
- Pots modificar aquesta base de dades!
- Pots esborrar dades!



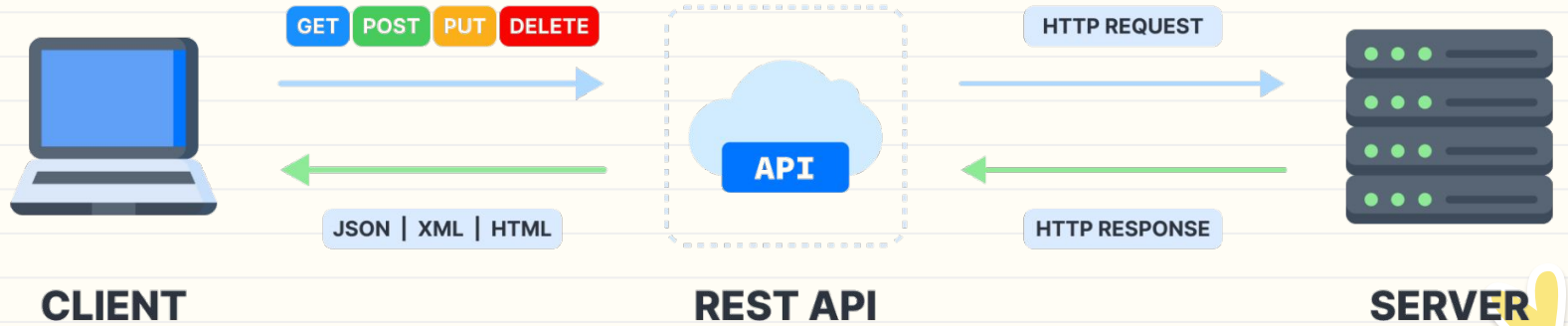
# Elements bàsics d'una API



- **Accés:** Autorització a comunicar-se amb l'API -> KEY
- **Request:** Dada o servei que es demana a l'API.  
Conté dues parts:
  - **Mètode/Method:** Pregunta que li fas a l'API
  - **Paràmetres:** Informació extra
- **Response:** Resposta o servei que et proporciona l'API depenent de la teva Request



# REST API Model

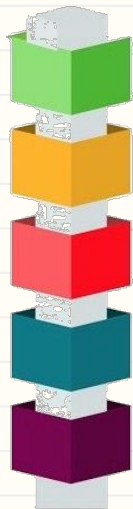


# Status Code



## HTTP Status Codes

Quan l'API ens retorna el **response** sol està acompanyat d'un codi que ens aporta info extra:



1XX  
INFORMATIONAL

2XX  
SUCCESS

3XX  
REDIRECTION

4XX  
CLIENT ERROR

5XX  
SERVER ERROR



# API Endpoints



Quan nosaltres estem cridant l'API, aquesta pot tenir diferents rutes que ens permeten accedir a diferent informació. Aquestes rutes s'anomenen endpoints. Mirem alguns exemples:

- Per accedir per exemple a la informació del client, podem utilitzar diferents endpoints:
  - Mètode HTTP serà **GET**
  - Per accedir a tots els clients, l'endpoint serà **/customers**
  - Per accedir a la informació d'un client en específic, podem tenir un altre endpoint com ara **/customers/<customer\_id>**
- Podem tenir un mateix endpoint per diferents mètodes:
  - Amb endpoint **/customers** i el mètode **GET** ens retornen la llista de clients.
  - Amb endpoint **/customers** i el mètode **PUT** podem crear un nou client





# Python API: Flask!



```
from flask import Flask

app = Flask(__name__)

@app.route('/')
def index():
    return 'Web App with Python Flask!'

app.run(host='0.0.0.0', port=81)
```







# Com començar?



El primer pas sempre serà preparar el nostre protocol. Això vol dir:

1. Primerament, hem de tenir una **base de dades**. Podem accedir directament o utilitzant una API al Backend
2. Hem de definir els nostres Endpoints i Query Params
3. Hem de baixar-nos les llibreries necessàries per crear aquesta API



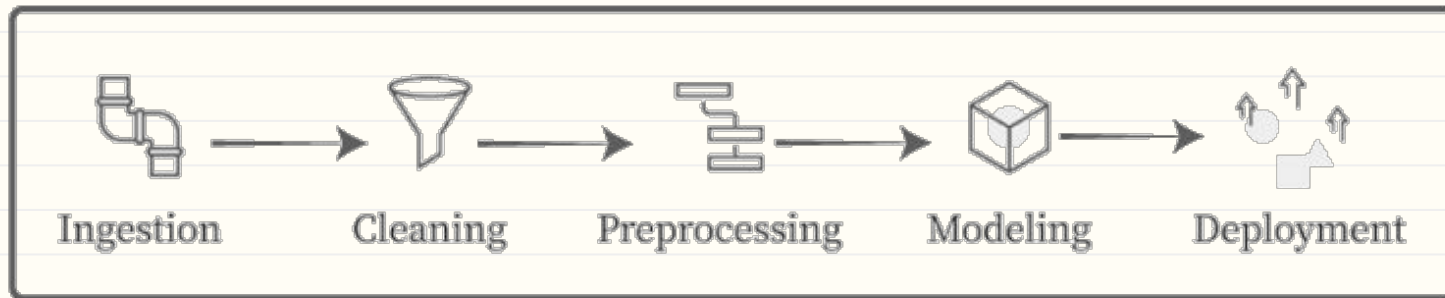


# Pipeline



Una manera de codificar i automatitzar el **workflow** per produir un model d'aprenentatge automàtic!

## Machine Learning Workflow





# Pipeline

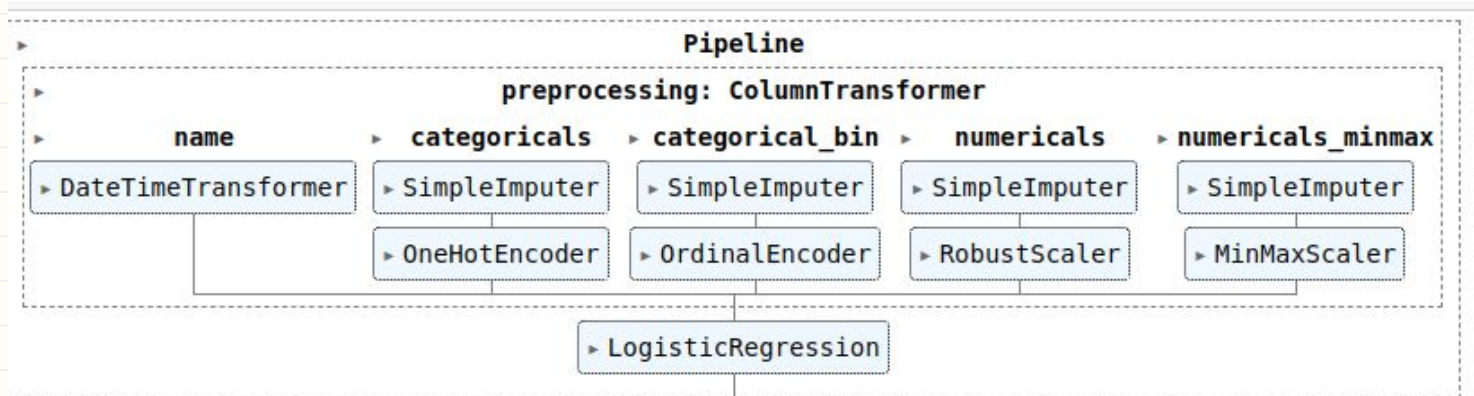
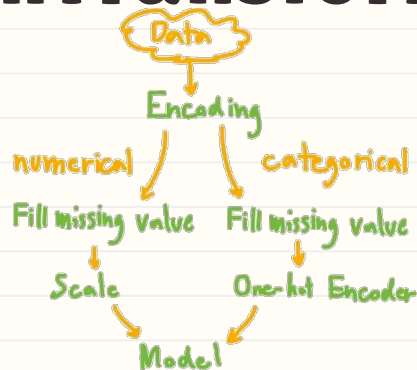
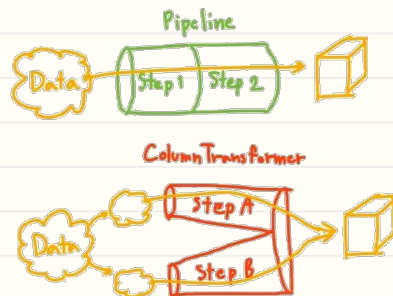


```
>>> from sklearn.svm import SVC
>>> from sklearn.preprocessing import StandardScaler
>>> from sklearn.datasets import make_classification
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.pipeline import Pipeline
>>> X, y = make_classification(random_state=0)
>>> X_train, X_test, y_train, y_test = train_test_split(X, y,
...                                                    random_state=0)
>>> pipe = Pipeline([('scaler', StandardScaler()), ('svc', SVC())])
>>> # The pipeline can be used as any other estimator
>>> # and avoids leaking the test set into the train set
>>> pipe.fit(X_train, y_train)
Pipeline(steps=[('scaler', StandardScaler()), ('svc', SVC())])
>>> pipe.score(X_test, y_test)
0.88
```





# Pipeline ColumnTransform





# Guardar pipeline



```
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
import joblib

# Cargar datos
iris = load_iris()
X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target, random_state=42)

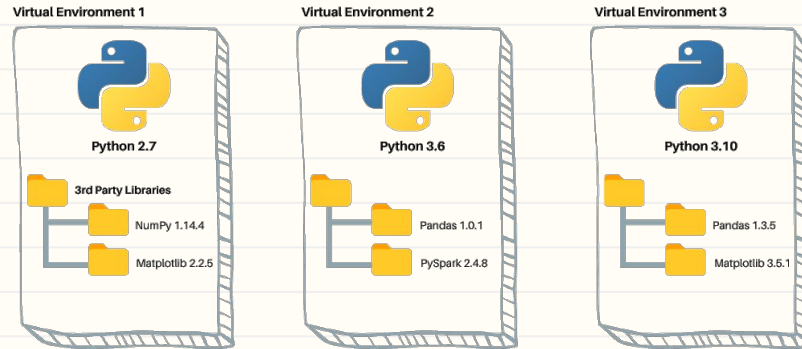
# Crear pipeline
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('classifier', LogisticRegression())
])

# Entrenar modelo
pipeline.fit(X_train, y_train)

# Guardar pipeline en archivo
joblib.dump(pipeline, 'pipeline_iris.joblib')
```

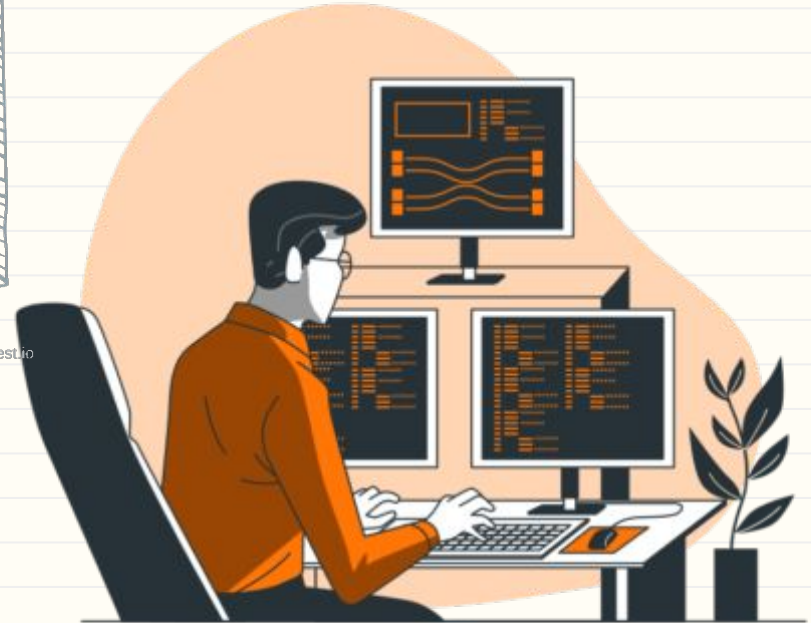


# Deployment



Podem crear diferents **virtualEnviroments** amb les diferents lliberies i **Flask** com a base de l'API

dataquest.io



# RESUM

Podem crear diferents **virtualEnviroments** amb les diferents llibreries i **Flask** com a base de l'API per donar serveis.

Aquests serveis poden ser filtrats per el Backend o la mateixa API Flask pot proporcionar-ho, dependrà del nostre objectiu i estroctura.

Per proporcionar serveis Machine Learning, hem de preparar la màquina per treballar ràpidament a les damandes!

# Instructions for use

If you have a free account, in order to use this template, you must credit Slidesgo by keeping the Thanks slide. Please refer to the next slide to read the instructions for premium users.

**As a Free user, you are allowed to:**

- Modify this template.
- Use it for both personal and commercial projects.

**You are not allowed to:**

- Sublicense, sell or rent any of Slidesgo Content (or a modified version of Slidesgo Content).
- Distribute Slidesgo Content unless it has been expressly authorized by Slidesgo.
- Include Slidesgo Content in an online or offline database or file.
- Offer Slidesgo templates (or modified versions of Slidesgo templates) for download.
- Acquire the copyright of Slidesgo Content.

For more information about editing slides, please read our FAQs or visit Slidesgo School:

<https://slidesgo.com/faqs> and <https://slidesgo.com/slidesgo-school>



# Instructions for use (premium users)

As a Premium user, you can use this template without attributing [Slidesgo](#) or keeping the "[Thanks](#)" slide.

## You are allowed to:

- Modify this template.
- Use it for both personal and commercial purposes.
- Hide or delete the "Thanks" slide and the mention to Slidesgo in the credits.
- Share this template in an editable format with people who are not part of your team.

## You are not allowed to:

- Sublicense, sell or rent this Slidesgo Template (or a modified version of this Slidesgo Template).
- Distribute this Slidesgo Template (or a modified version of this Slidesgo Template) or include it in a database or in any other product or service that offers downloadable images, icons or presentations that may be subject to distribution or resale.
- Use any of the elements that are part of this Slidesgo Template in an isolated and separated way from this Template.
- Register any of the elements that are part of this template as a trademark or logo, or register it as a work in an intellectual property registry or similar.

For more information about editing slides, please read our FAQs or visit Slidesgo School:

<https://slidesgo.com/faqs> and <https://slidesgo.com/slidesgo-school>

# Fonts & colors used

This presentation has been made using the following fonts:

## Lilita One

(<https://fonts.google.com/specimen/Lilita+One>)

## Poppins

(<https://fonts.google.com/specimen/Poppins>)

#372d31

#ffdf5

#fae77f

#cdea9d

#8fddef

#ddb2d5

#f198c5

#f2b87e

#efefef

#ffffff

# Storyset

Create your Story with our illustrated concepts. Choose the style you like the most, edit its colors, pick the background and layers you want to show and bring them to life with the animator panel! It will boost your presentation. Check out [how it works](#).



Pana



Amico



Bro



Rafiki

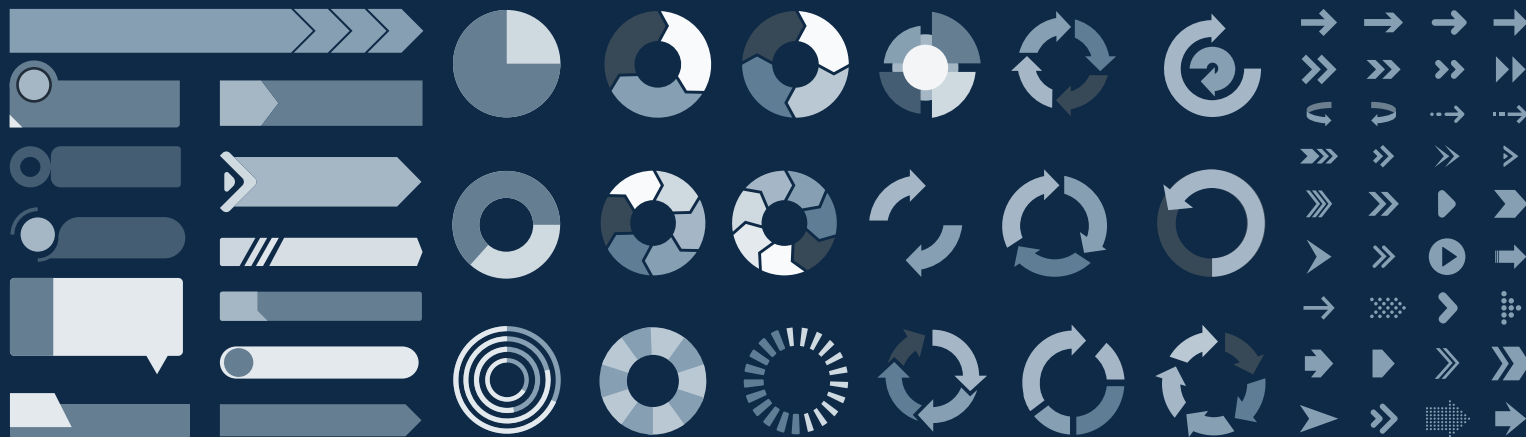


Cuate

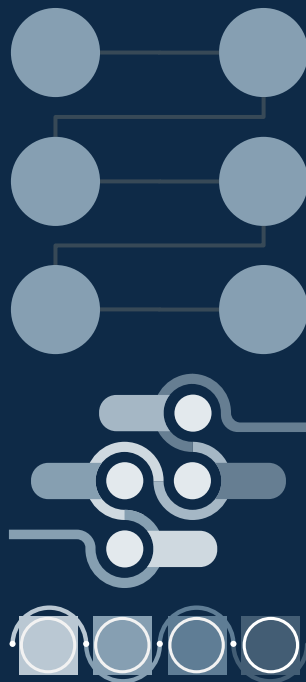
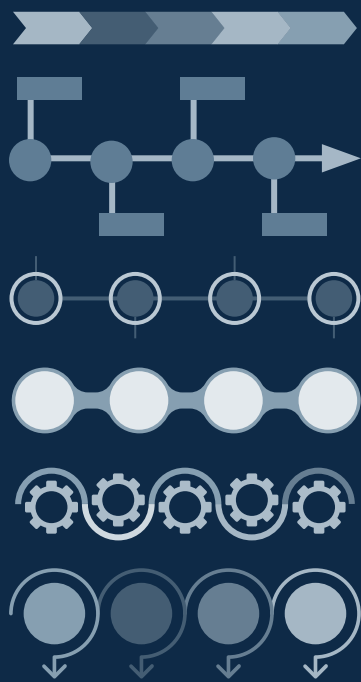
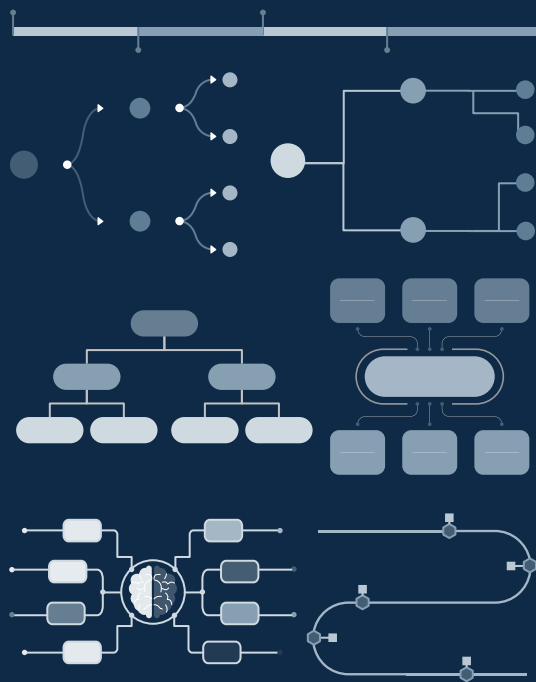
# Use our editable graphic resources...

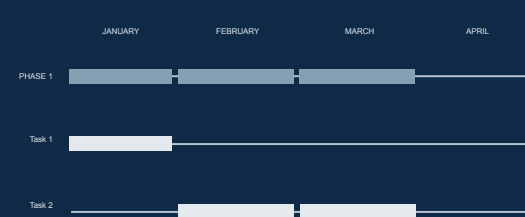
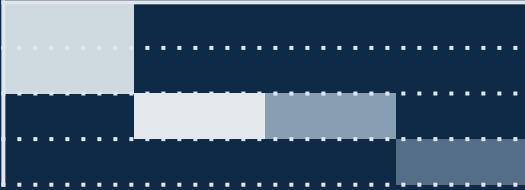
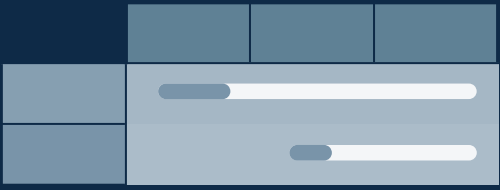
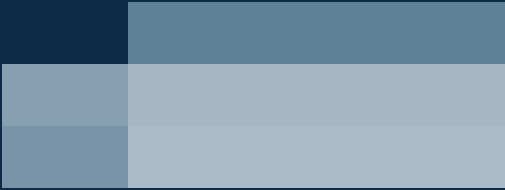
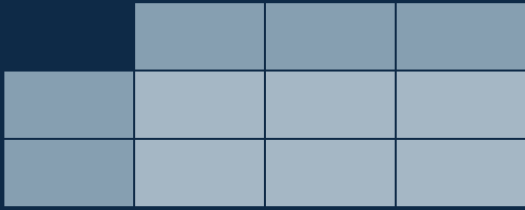
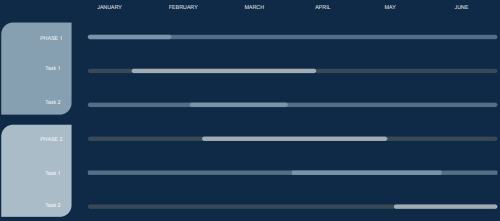
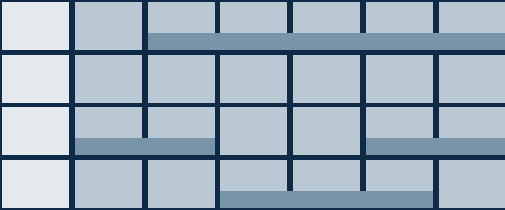
You can easily **resize** these resources without losing quality. To **change the color**, just ungroup the resource and click on the object you want to change. Then, click on the paint bucket and select the color you want.

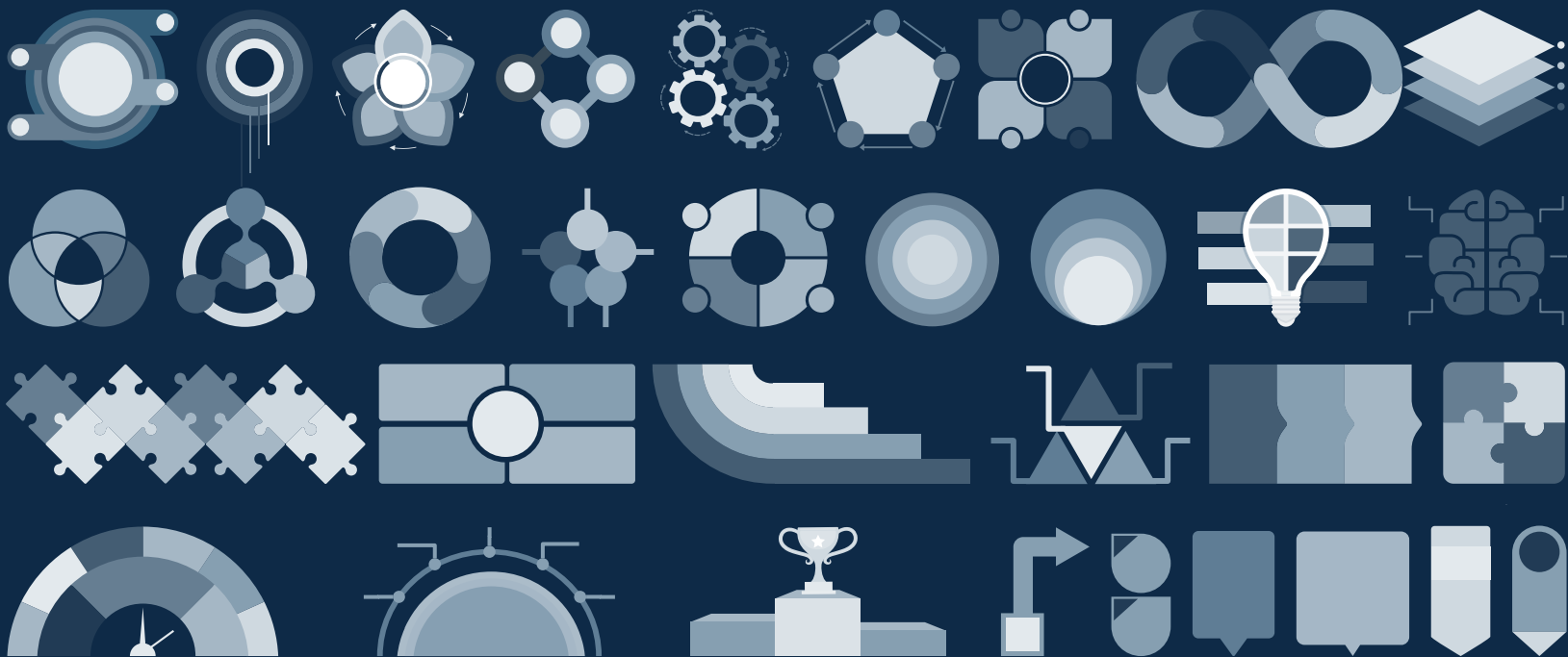
Group the resource again when you're done. You can also look for more **infographics** on [Slidesgo](#).















# ...and our sets of editable icons

You can **resize** these icons without losing quality.

You can **change the stroke and fill color**; **just select the icon** and click on the **paint bucket/pen**.

In Google Slides, you can also use **Flaticon's extension**, allowing you to **customize** and **add even more icons**.



## Educational Icons



## Medical Icons



## Business Icons



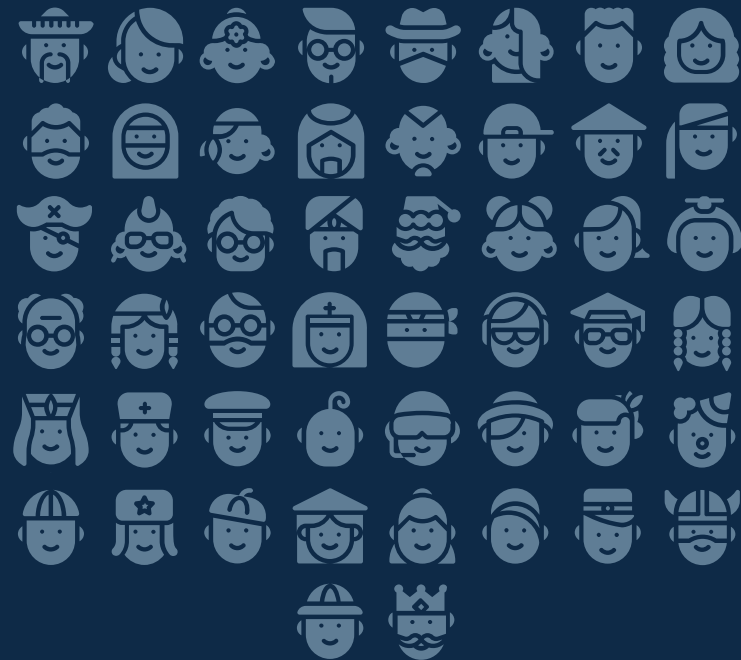
## Teamwork Icons



# Help & Support Icons



# Avatar Icons



[illegible]

# Nature Icons



# SEO & Marketing Icons





