



POLITECNICO
MILANO 1863

DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

POLITECNICO MILANO 1863

NECST
laboratory

LCG: Informatica Applicata - C2

Intro al corso, Assessment Info, Sviluppo locale, Git/Githubpages, P5 first

Scuola del Design Laurea Triennale in Communication Design

Aula B B2.0.1 - D.I., Politecnico di Milano, September 23rd, 2025

Davide Conficconi <davide.conficconi@polimi.it>
Alessandro Nazzari <alessandro.nazzari@polimi.it>

Who we are



Assistant Professor (RTDa) @



DIPARTIMENTO DI ELETTRONICA
INFORMAZIONE E BIOINGEGNERIA

Computer Systems Research

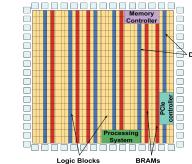
Codesign Domain-Specific Computer Architectures and Systems



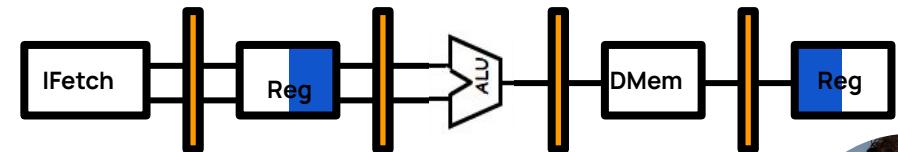
Performance



Energy Efficiency



Reconfigurable and
Heterogeneous Systems
(e.g., FPGAs)

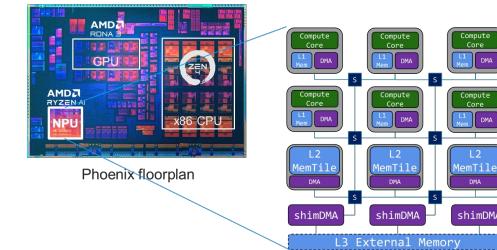


Lecturer in basic informatics and Advanced Computer Architectures

Lecturer for *FPGA101: from reconfigurable to domain-specific systems*

Powered by **AMD University Program**

Lecturer for PhD-level: *Advanced Topics in Codesign of Domain-Specific Accelerated Computing Architectures and Systems*



Intern at research teams of IBM (21/22), Xilinx (18/19), and Oracle (18)

IBM Research | Zurich

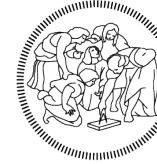
XILINX

ORACLE

Who we are



PhD Student @



POLITECNICO
MILANO 1863

Research in ML and control for Unmanned Aerial Vehicles at the FlyArt lab - Department of Aerospace Science and Technology



Who we are



PhD Student @



POLITECNICO
MILANO 1863

Research in ML and control for Unmanned Aerial Vehicles at the FlyArt lab - Department of Aerospace Science and Technology



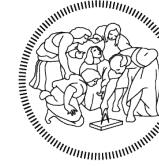
Focus on swarm coordination, safe trajectory planning and edge devices



Who we are



PhD Student @



POLITECNICO
MILANO 1863

Research in ML and control for Unmanned Aerial Vehicles at the FlyArt lab - Department of Aerospace Science and Technology



Focus on swarm coordination, safe trajectory planning and edge devices



But deep inside I'm a computer science guy, Bachelor and Master in Computer Science and Engineering at DEIB, research focus on dependable systems and robust Machine Learning

Opportunità di Supporto dall'Ateneo

- **Attività di didattica innovativa - Passion in Action <https://www.polimi.it/corsi/didattica-innovativa>**

"Passion in action" è il catalogo di attività didattiche a partecipazione libera che il Politecnico propone ai propri studenti, per favorire in loro lo sviluppo di competenze trasversali, di soft e social skills, e per incoraggiare/facilitare un arricchimento personalizzato del loro bagaglio personale, culturale e professionale. Chi lo desidera può cogliere questa opportunità e scegliere quali attività frequentare, spaziando tra le diverse materie in base ai propri interessi e alle attitudini personali. Gli studenti che partecipano a "Passion in action" possono iscriversi a tutte le attività in catalogo, senza vincoli di vicinanza tematica rispetto al percorso di studio cui sono iscritti (fermo restando gli eventuali prerequisiti di accesso alle singole iniziative).

Il riconoscimento delle abilità acquisite avviene mediante riconoscimenti di cfu extra Curriculari menzionati nel Diploma Supplement e trasmissione di un badge elettronico.

- **Vi è la possibilità di chiedere un tutorato «PEER TO PEER» a sostegno degli studenti dei corsi di laurea triennale.**

Il servizio di supporto erogato da parte di studenti-tutor già selezionati (del secondo e terzo anno di corso) avrà lo scopo di fornire un sostegno, one-to-one nello studio di insegnamenti in corso di frequenza, in particolare in quelli individuati con maggiore criticità.

Gli studenti che effettueranno richieste di tutorato Peer to Peer, dovranno inoltrare domanda nell'intervallo di apertura bando che la Presidenza attiva sia per il primo semestre che per il secondo semestre. Per info scrivere a tutorato-design@polimi.it

- **Vi è la possibilità di rivolgersi a Multichance e Polipsi**

<https://www.polimi.it/campus-e-servizi/pari-opportunita-e-inclusione/disabilita-e-dsa> (per disabilità/disturbi dell'apprendimento/problemi di salute)

<https://www2.polimi.it/servizi-e-opportunita/altri-servizi-e-opportunita/servizi-di-supporto-e-ascolti/polipsi.html> (sostegno psi)

- **Vi è la possibilità per studenti in transizione di genere possono chiedere l'attivazione di un alias,**

scrivendo a: sgs@polimi.it <https://www.polimi.it/campus-e-servizi/pari-opportunita-e-inclusione/lgbti>

Who are you?



Tempo dell'Appello



Disclaimers

Terza edizione del corso: abbiamo fatto ANCORA aggiustamenti, il vostro contributo resta fondamentale.

Cercheremo di fornire informazioni sempre in anticipo su temi e lezioni, ma ci potranno essere assestamenti in base a come la classe reagisce

Useremo molto la scrittura di codice: ripartendo dalle basi, cercheremo di trasmettere perchè è bello progettare con il codice, ma un po' deve già piacervi

Restiamo ingegneri informatici :)

Obiettivo di Informatica Applicata

Comprendere meglio le potenzialità del mondo **informatico** nel **campo della comunicazione** considerando possibili sbocchi professionali



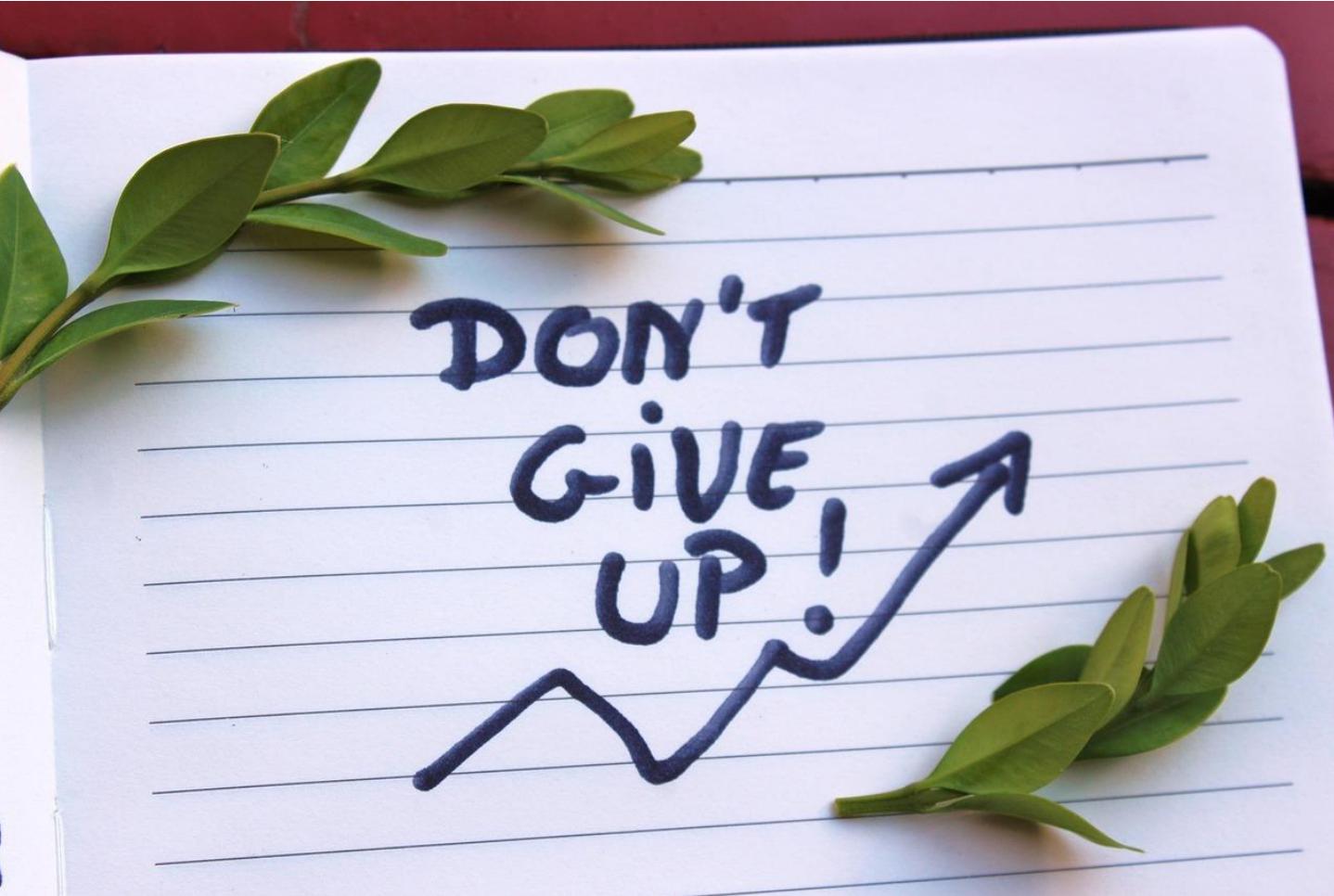
Sfruttare **codice** per **creare artefatti digitali** con una focalizzazione a **partire dai dati**



Sfruttare **piattaforme web** per comunicare e creare tali artefatti digitali



Obiettivo di Informatica Applicata



Fornire le basi per una **visione di dettaglio** di dati in maniera **computazionale**

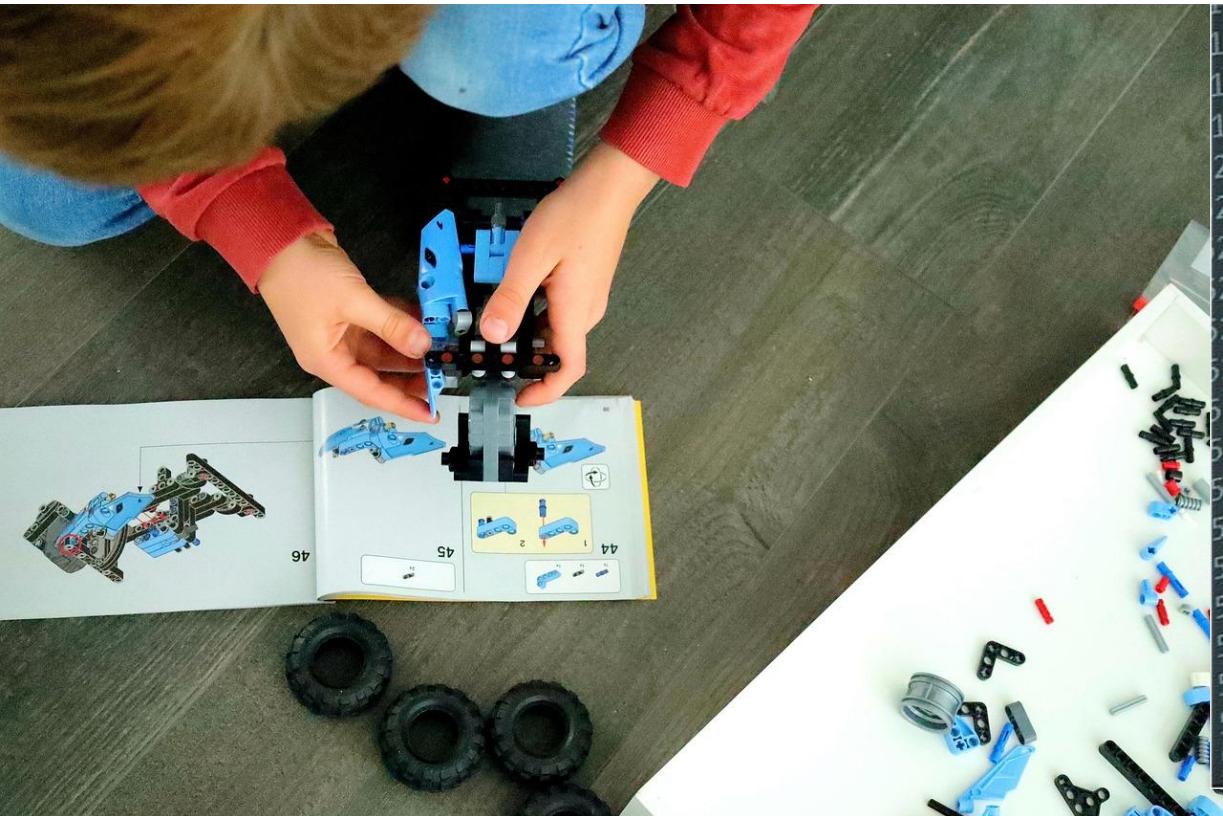
The Path Ahead

- 1) Warm up, basics, tools
- 2) Basics of computational approach to make a draw computationally
- 3) Correlating Data and Drawing, adding functions, data/media, arrays
- 4) User interaction, objects and transformations
- 5) More dataviz!, big projects managements, modules, debugging
- 6) End to end example of quantitative data visualization

Each lecture has additional challenges to grant at most 3 bonus points (globally)

Information Visualization

handcrafted



13

Image by [Thomas G.](#) from [Pixabay](#)

algorithmically/ computationally

```
string sInput;
int iLength, iN;
double dblTemp;
bool again = true;

while (again) {
    iN = -1;
    again = false;
    getline(cin, sInput);
    system("cls");
    stringstream(sInput) >> dblTemp;
    iLength = sInput.length();
    if (iLength < 4) {
        again = true;
        continue;
    } else if (sInput[iLength - 3] != '.') {
        again = true;
        continue;
    } while (++iN < iLength) {
        if (isdigit(sInput[iN])) {
            continue;
        } else if (iN == (iLength - 3)) {
```

Image by [Christopher Kuszajewski](#) from [Pixabay](#)

Visualizing My Research: Application-Driven HW/SW Co-Design of Computing Systems

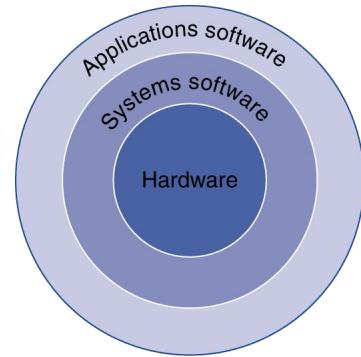


Visualizing My Research: Application-Driven HW/SW Co-Design of Computing Systems



Visualizing My Research

Computer Systems Research (a.k.a. Hardware and Software)



Fast Data Analysis

Biomedical Image Processing

Regex Matching



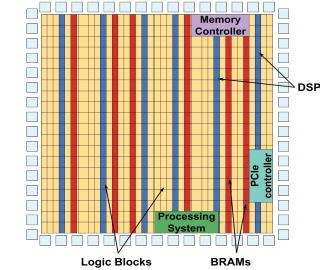
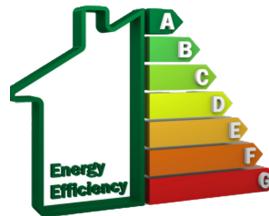
$$\text{ACGT}(\text{A|C})^*$$

Efficient Image Registration



AI acceleration

Codesign Domain-Specific Computer Architectures and Systems



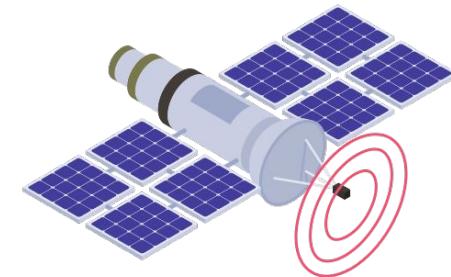
Performance

Energy Efficiency

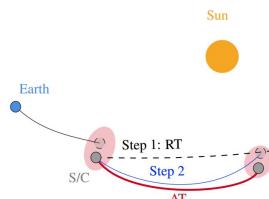
Reconfigurable and Heterogeneous Systems

Autonomous Deep Space Satellites

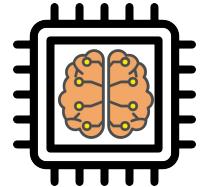
Remote Sensing



Guidance Navigation and Control

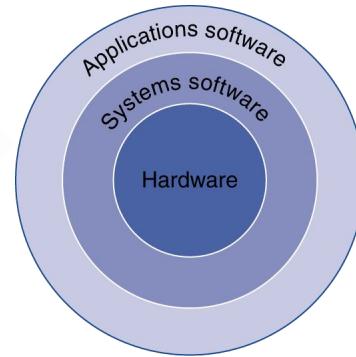


Neuromorphic Computing



Visualizing My Research

Computer Systems Research (a.k.a. Hardware and Software)

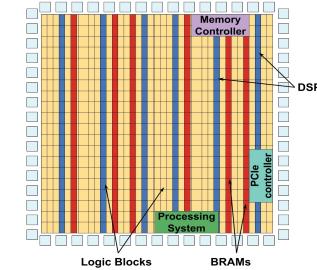


Fast Data Analysis

Biomedical Image Proce

Interdisciplinary team with Polimi, TUM, Imperial

Codesign Domain-Specific Computer Architectures and Systems



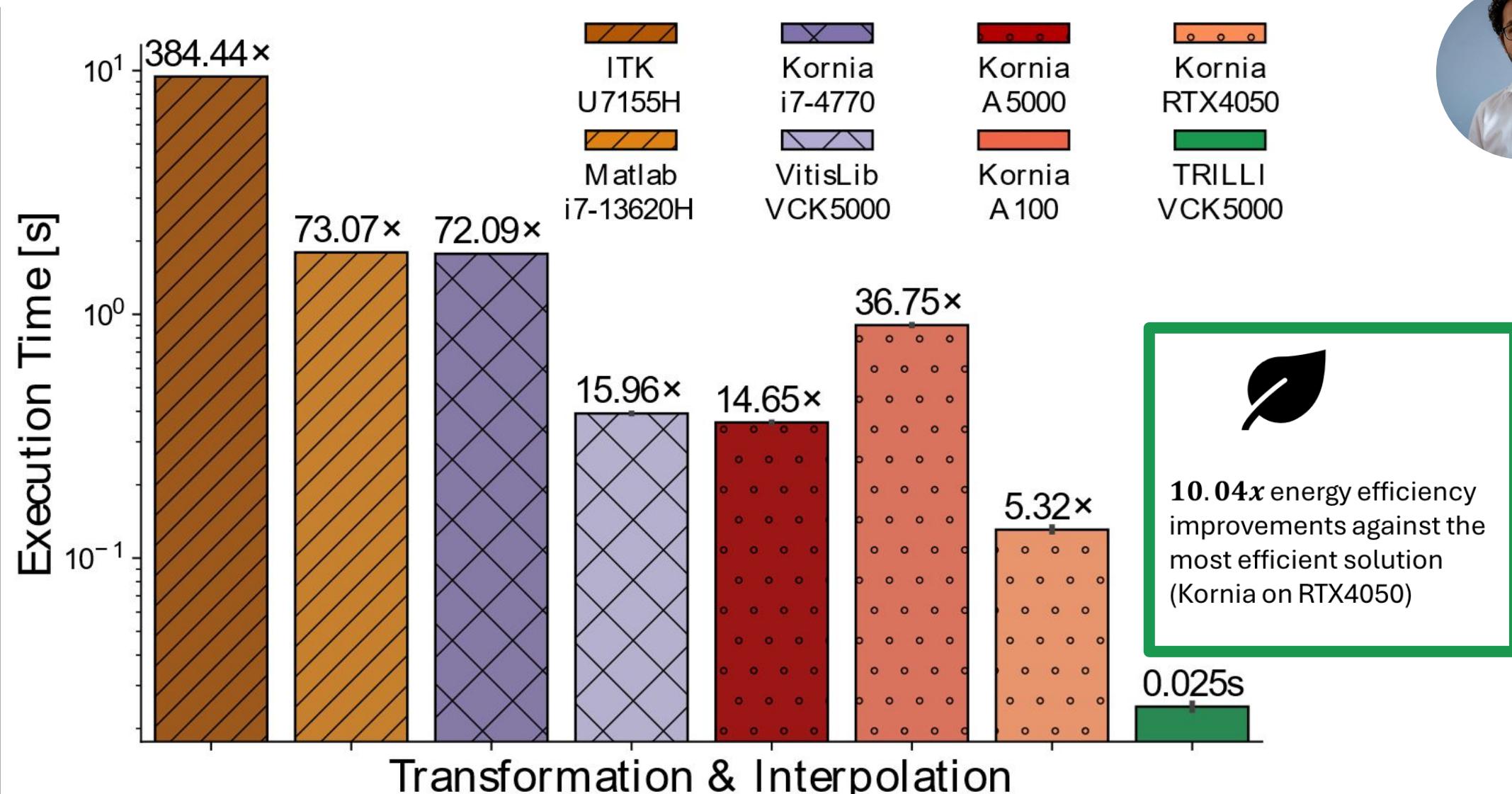
Performance

Energy Efficiency

Reconfigurable and Heterogeneous Systems



Computational Approach to Data Visualization



Informatica Applicata Technical Contents

Computazione tramite

JavaScript



057306 - FONDAMENTI DI INFORMATICA PER IL WEB DESIGN (PILATO CHRISTIAN)

Strumenti Sviluppo



Locale

<https://code.visualstudio.com/download>



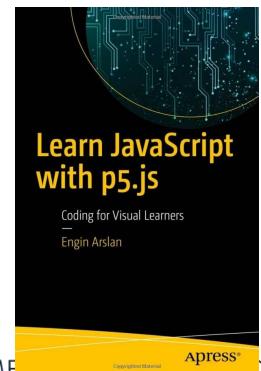
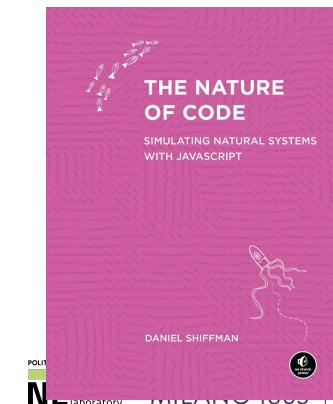
Remoto
Interattivo

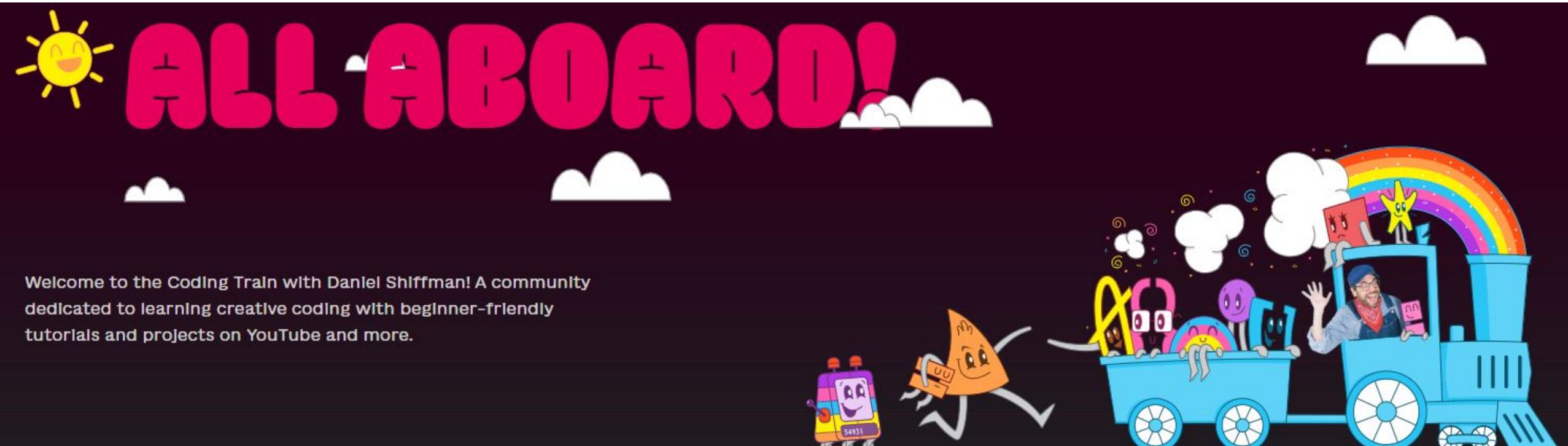
<https://editor.p5js.org/>



<https://desktop.github.com/>

Per visualizzare e manipolare ci appoggeremo a



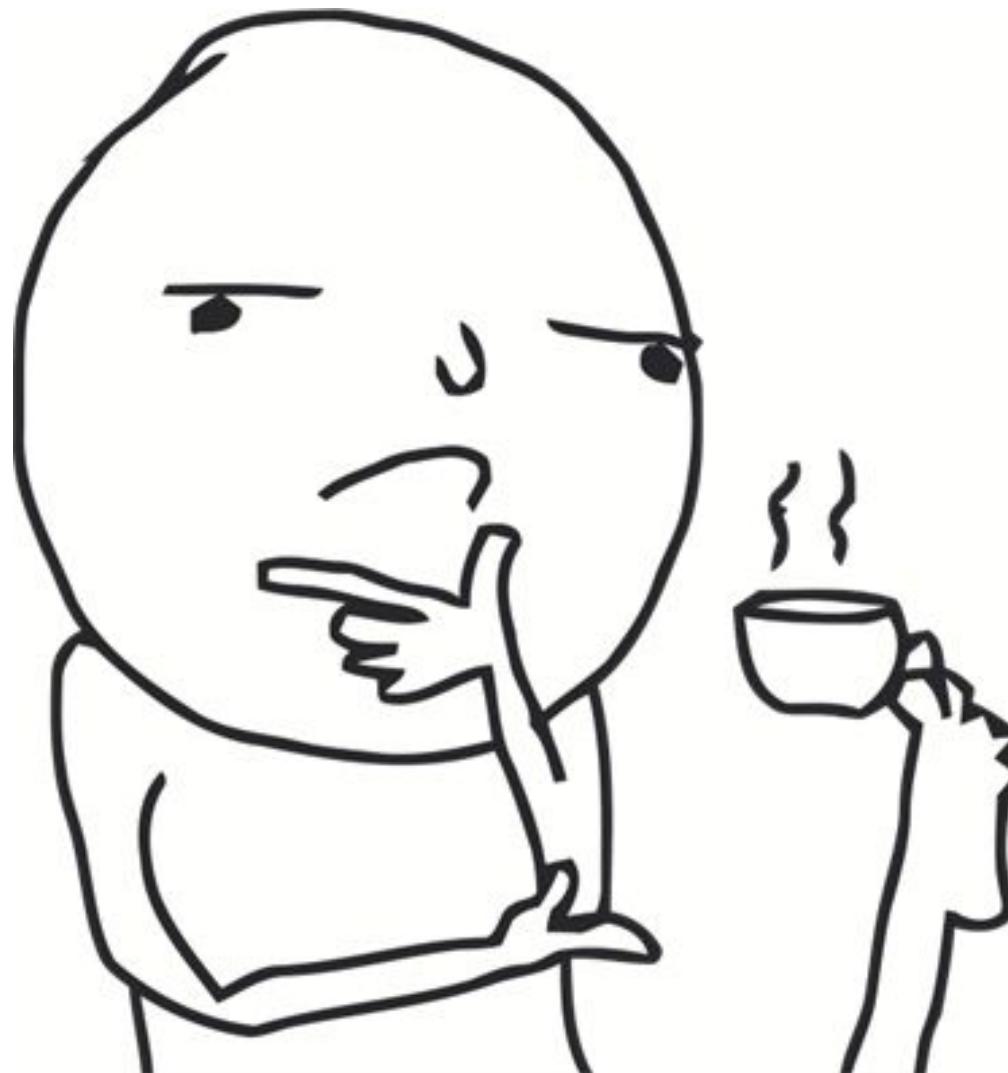


<https://thecodingtrain.com/>

Cool!



But how shall we start?



Informatica Self-Assesment



Informatica Self-Assesment

<https://forms.office.com/e/04ggS9PFn3>



Informatica Self-Assesment

1. ~~Informatica Self Assessment~~



2. Local dev with VSCode, Indice html for IA2526

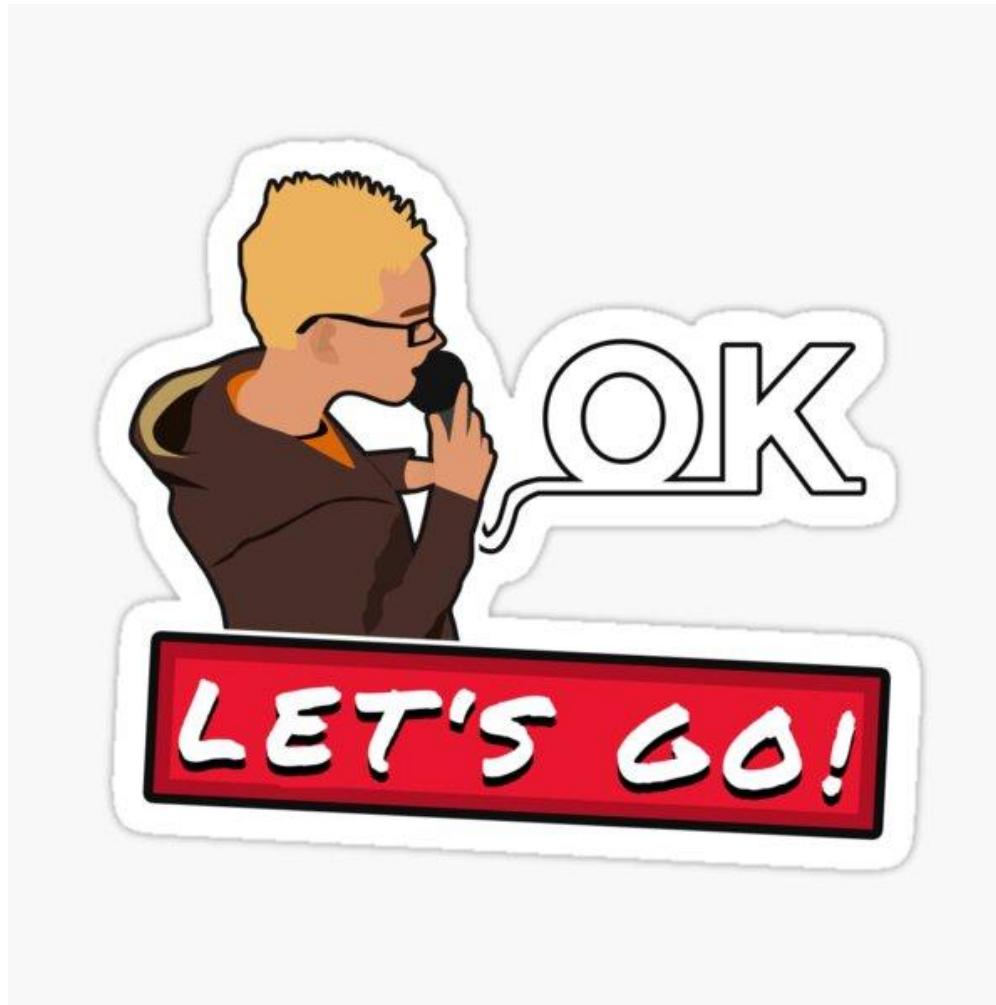
3. Create our Webpage with Githubpages



4. Draw with P5, P5 Live Editor



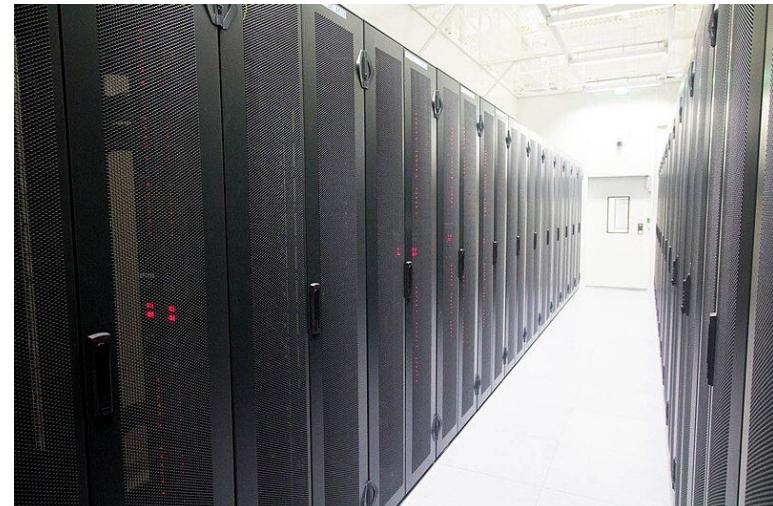
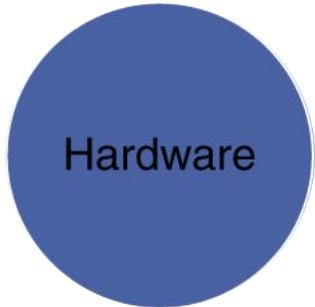
<https://editor.p5js.org/>



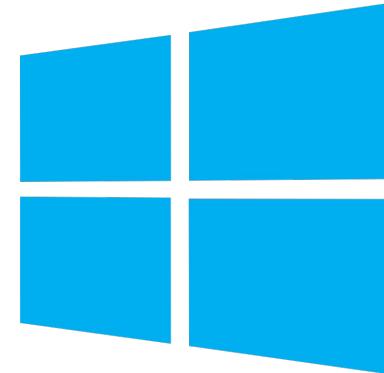
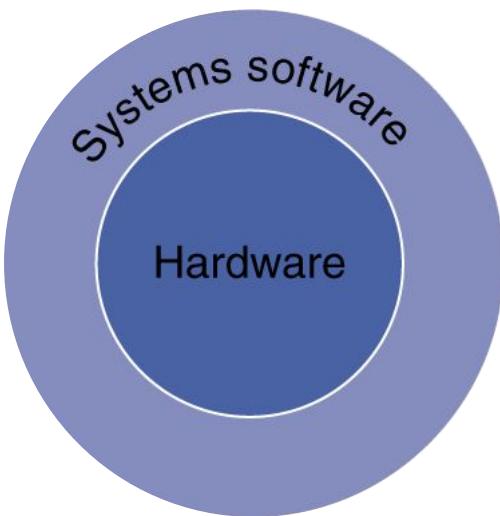
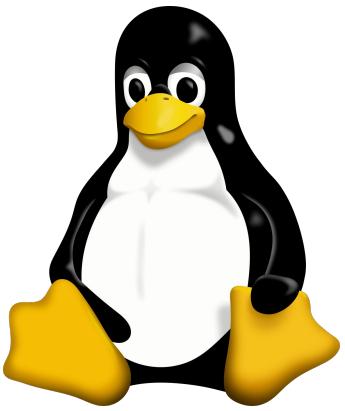
CS101: What is Computer Science (a.k.a Informatica)?

Scienza della
rappresentazione
e dell'
elaborazione
dell'informazione

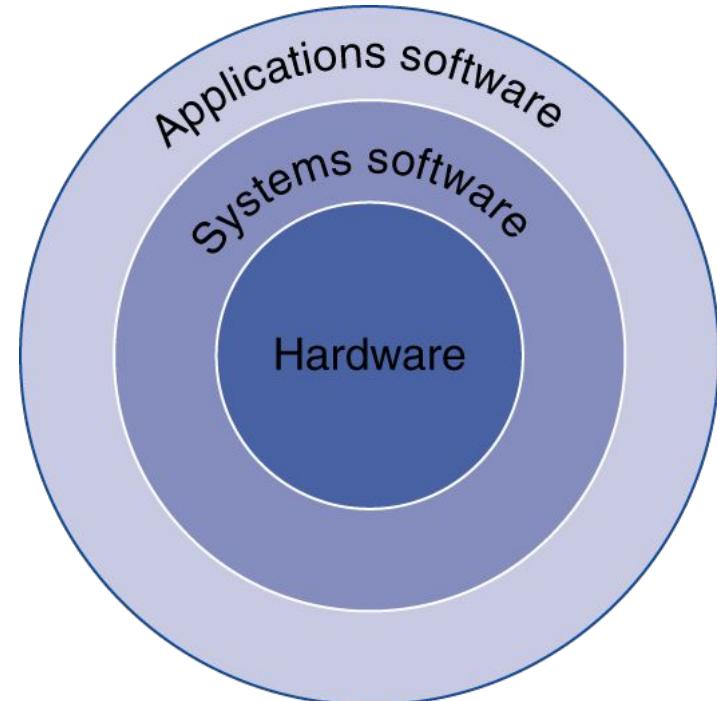
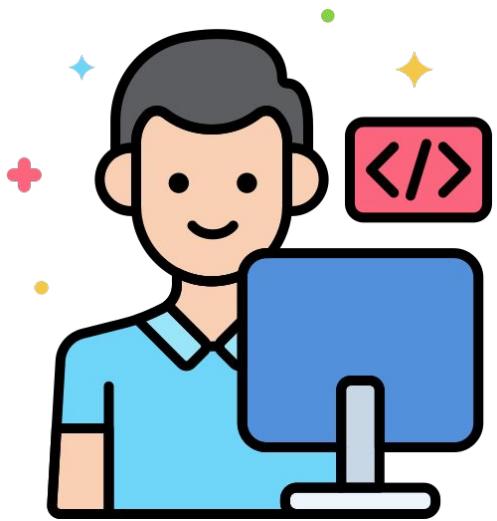
CS101: Computing Stack



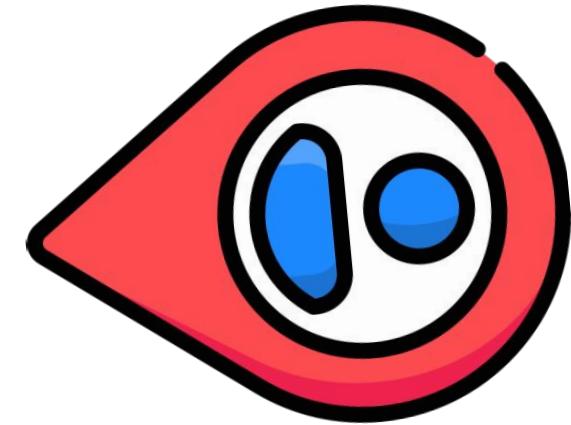
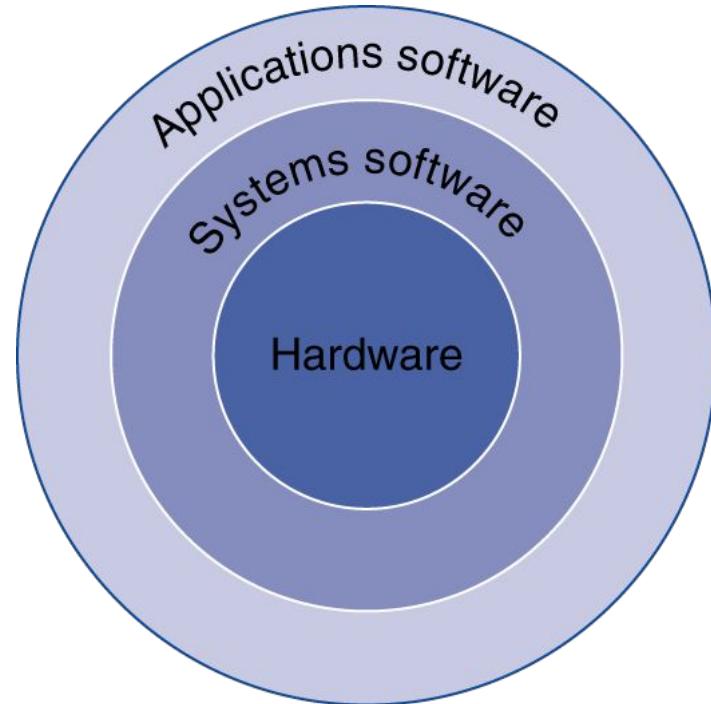
CS101: Computing Stack



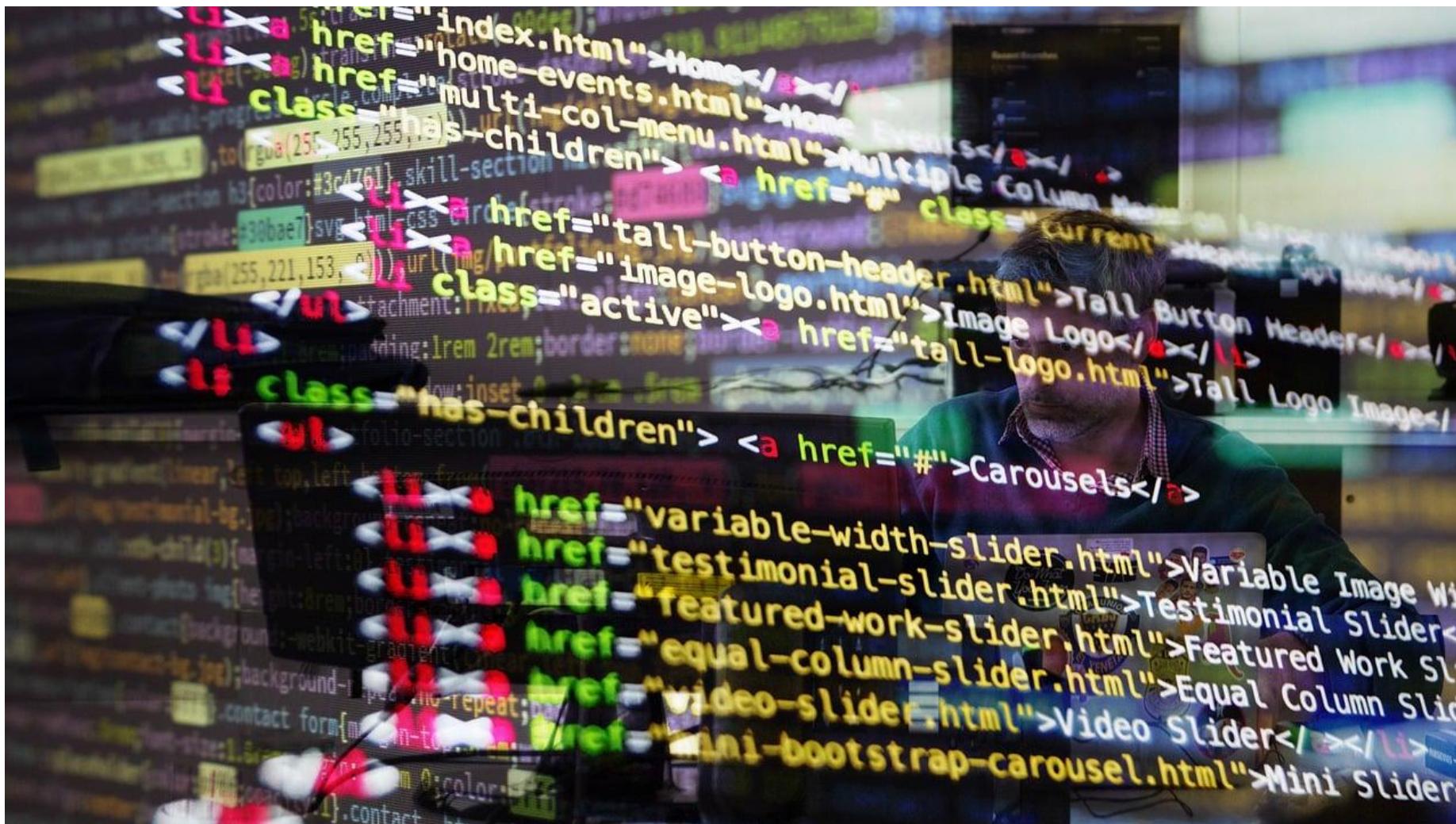
CS101: Computing Stack



CS101: Computing Stack



CS101: Local Development





Visual Studio Code (vSCode) è l'ambiente di sviluppo di riferimento per il nostro corso

Si tratta di un software liberamente installabile da:

<https://code.visualstudio.com/download>

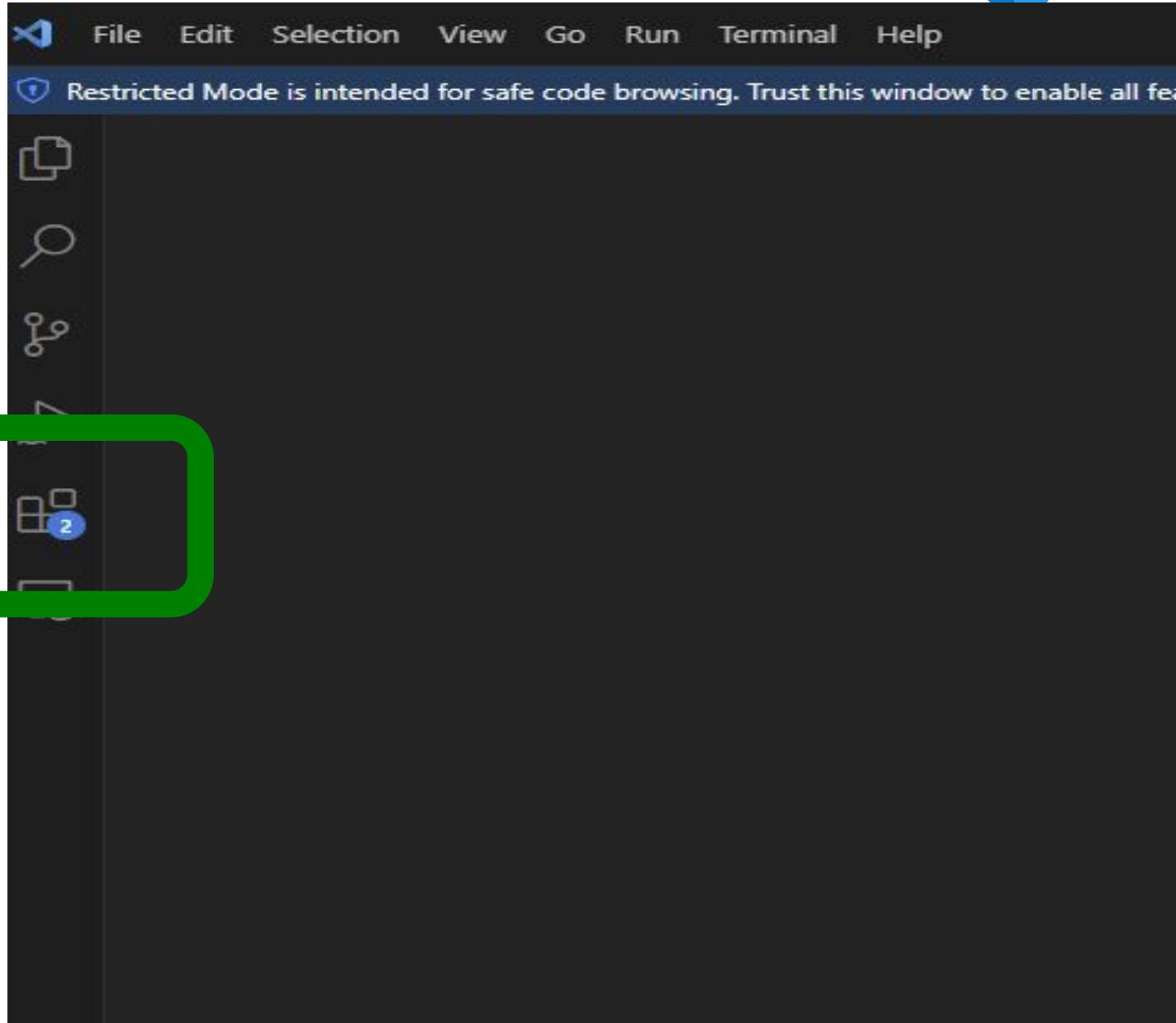
I repository/market del proprio sistema operativo

Dopo aver installato VSCode, aviamolo e procediamo con l'installazione dell'estensione per lo sviluppo locale

Visual Studio Code: Extensions



Aprire il pannello di
installazione
delle estensioni

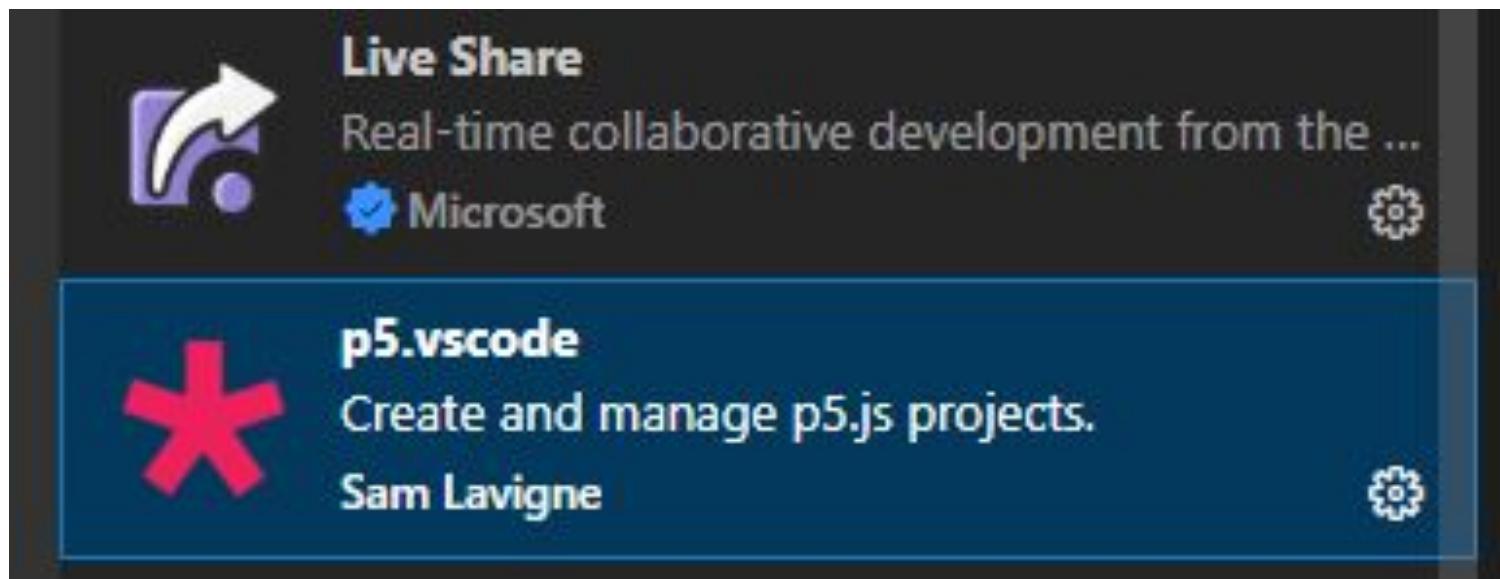
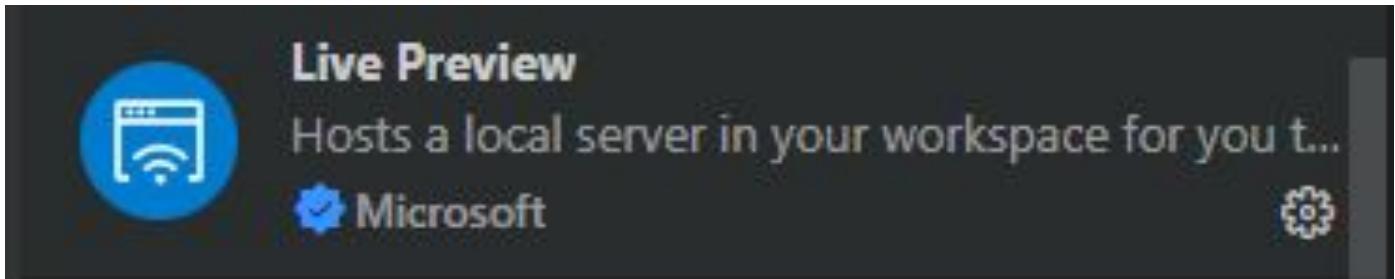


Visual Studio Code: Extensions

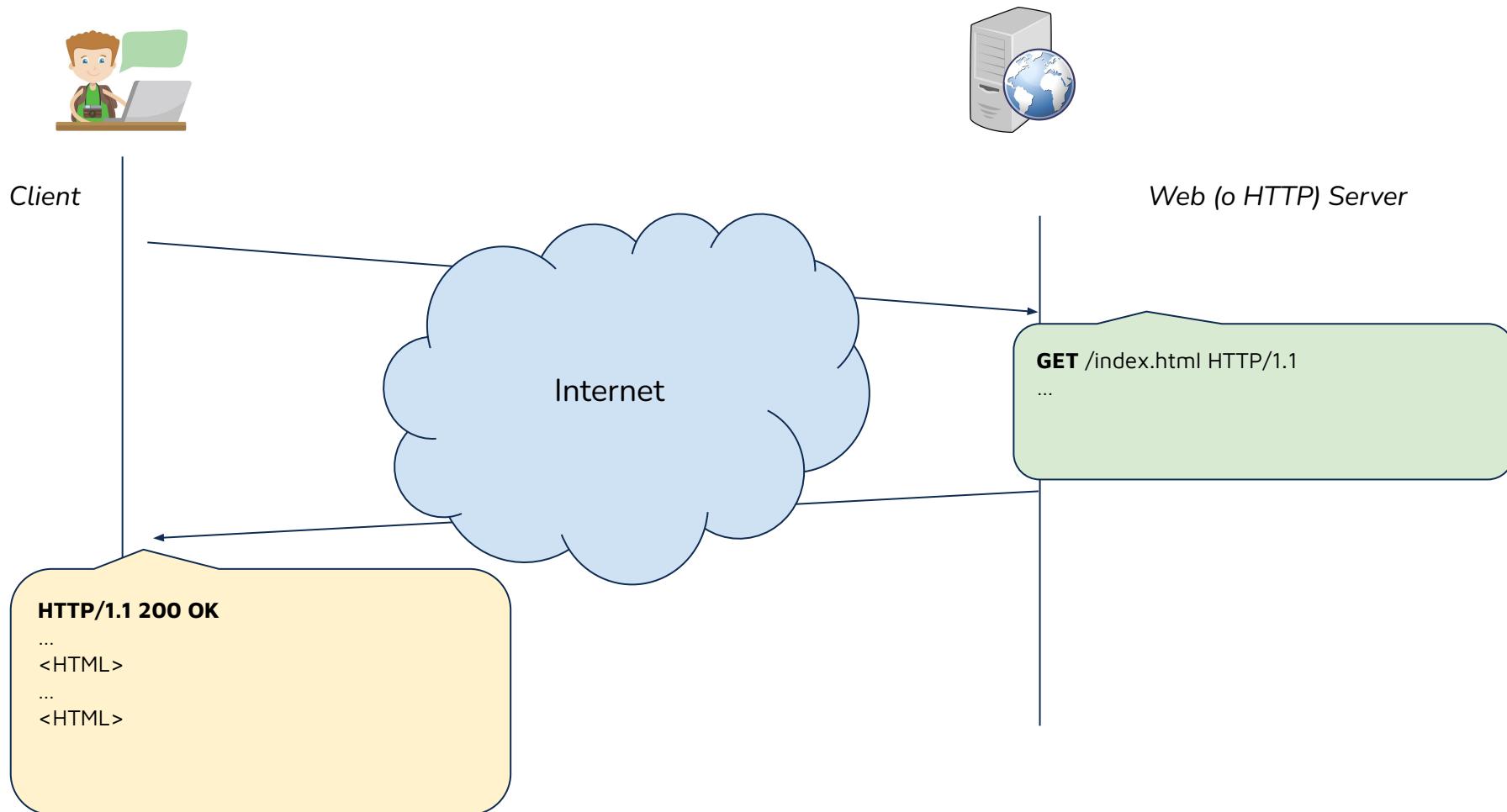


Cercare l'estensione
digitando
nella barra di ricerca e
installatela

1. “live preview”
2. “p5.vscode”
3. “live share”



Internet in a Nutshell



Let's try it with VSCode!



Image by [Nghĩa Trinh](#) from [Pixabay](#)



Un contributo importante allo sviluppo delle tecnologie informatiche è arrivato anche grazie alla diffusione di Internet

L'idea di creare una **rete** di sistemi di elaborazione, per lo scambio di dati si è concretizzata attraverso il progetto militare U.S. ARPANET (1969)

Col tempo si sono sviluppate altre reti di calcolatori, soprattutto in ambito accademico

Fine anni 80' / primi anni 90', opportuni **protocolli di comunicazione** rendono possibile lo scambio di informazioni tra reti diverse



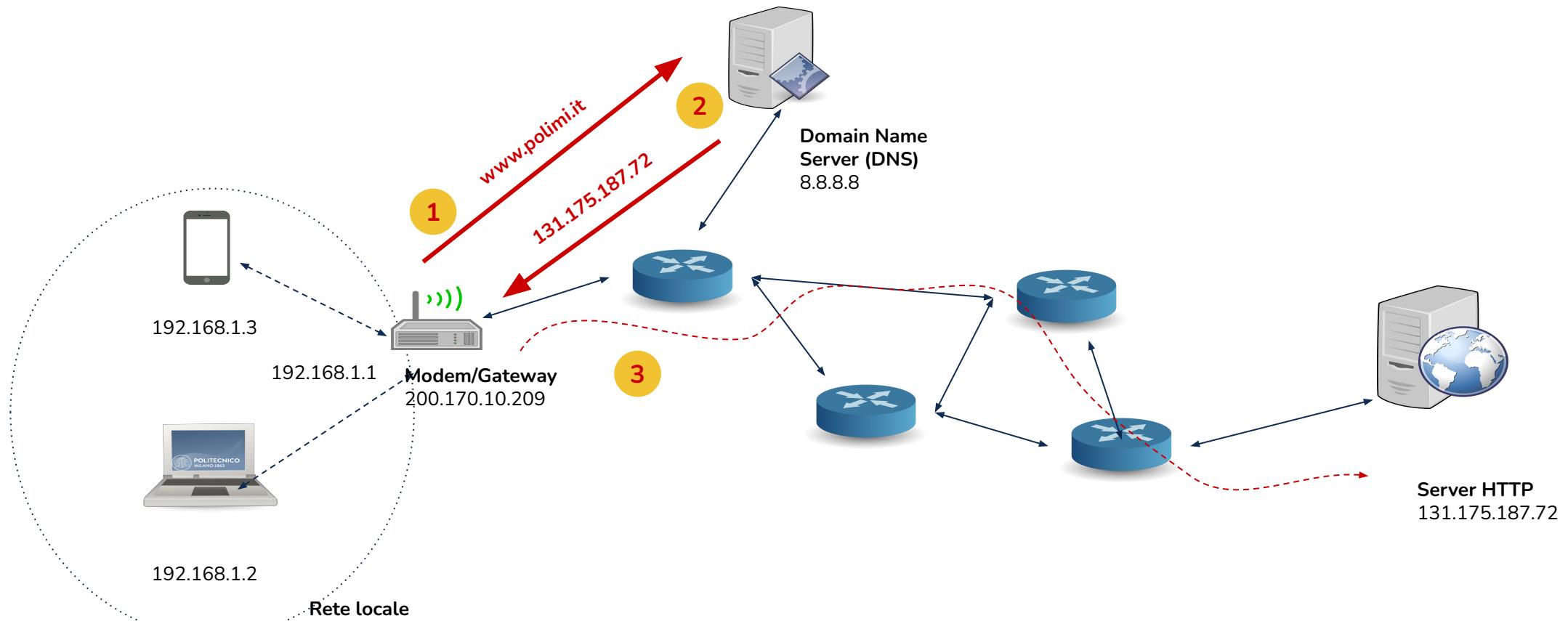
Ogni dispositivo connesso a Internet è raggiungibile attraverso il **Internet Protocol (IP) address**

Una combinazione di 4 valori da 0 a 255, separati da un punto: esempio
131.175.187.72

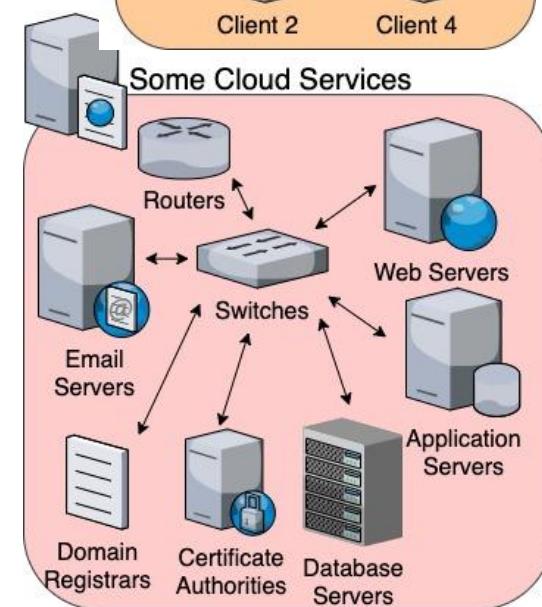
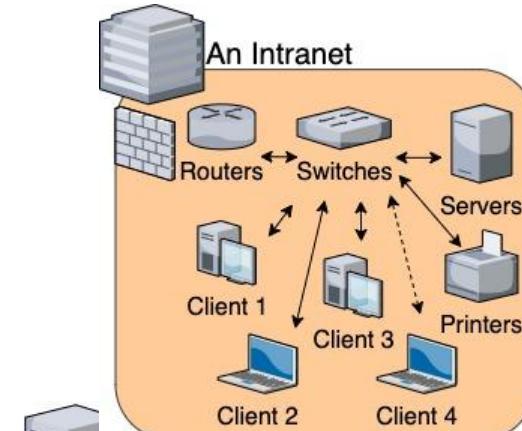
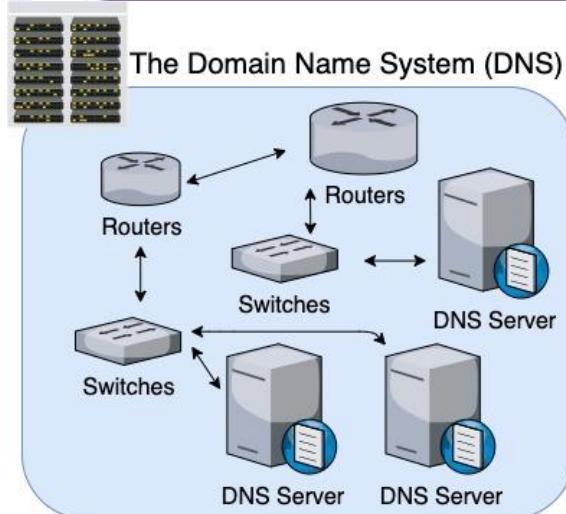
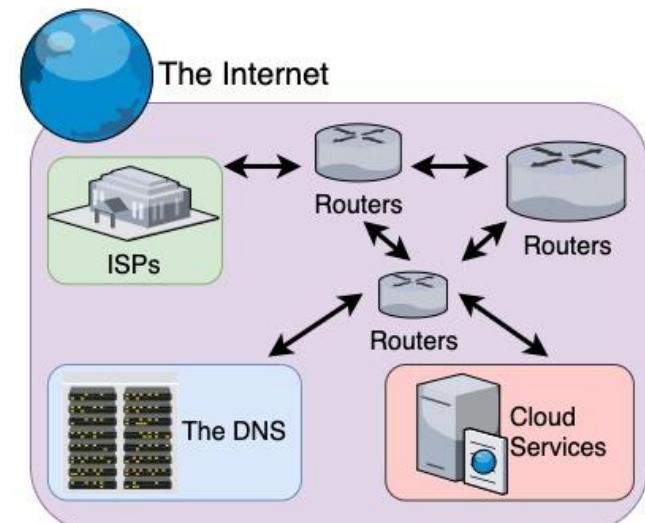
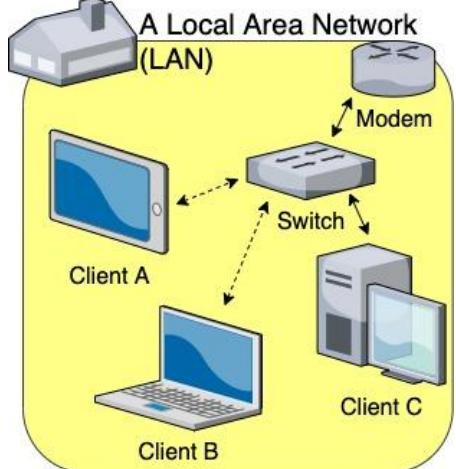
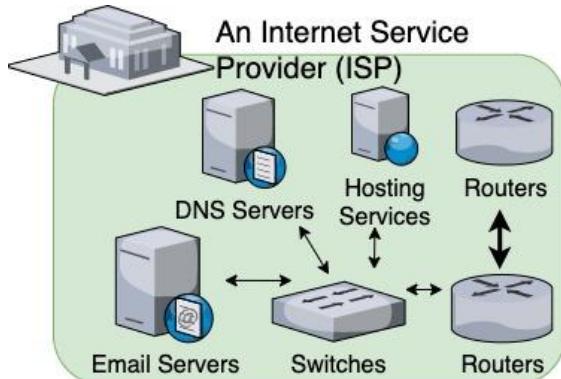
Gli indirizzi IP non sono sempre immediati da memorizzare di conseguenza, si è introdotto un approccio basato su **nomi di dominio**: esempio
www.polimi.it

Oppunti sistemi di rete, detti **Domain Name Servers (DNS)** fungono da “rubriche” per la traduzione in indirizzi IP

Internet: indirizzamento



Internet: infrastruttura

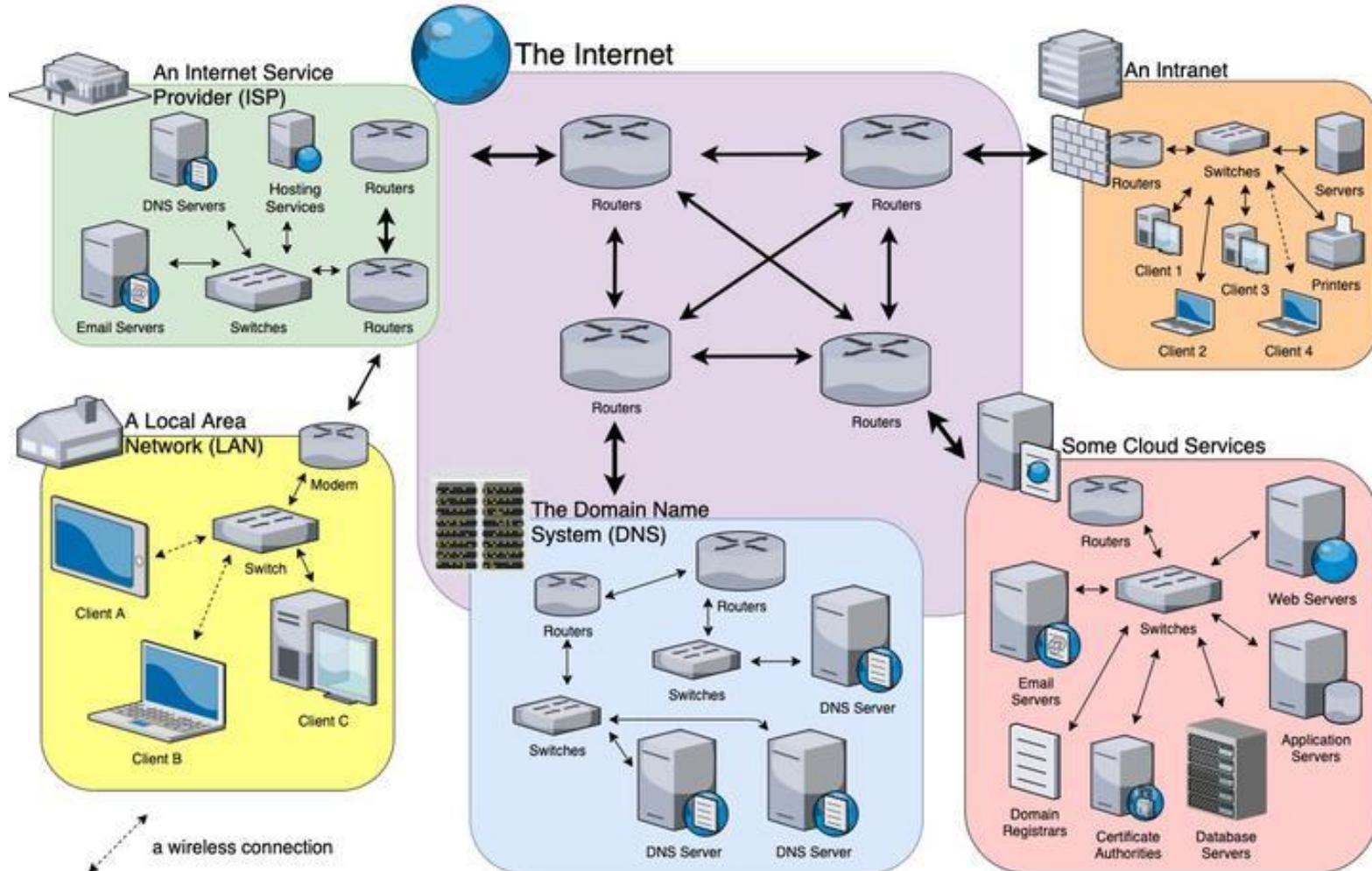


<https://vahid.blog/post/2020-12-15-how-the-internet-works-part-i-infrastructure/>

Internet: infrastruttura



The Internet Infrastructure: A bird's eye view



Source: Vahid Dejwakh, 2020

<https://vahid.blog/post/2020-12-15-how-the-internet-works-part-i-infrastructure/>

Internet: i server

In una rete di sistemi di elaborazione, tipicamente alcuni “nodi” svolgono il ruolo di **server**, ovvero forniscono un servizio specifico

Un servizio è reso disponibile dall'esecuzione di un programma dedicato, che si pone in attesa di connessioni



Internet: i server

In una rete di sistemi di elaborazione, tipicamente alcuni “nodi” svolgono il ruolo di **server**, ovvero forniscono un servizio specifico

Un servizio è reso disponibile dall'esecuzione di un programma dedicato, che si pone in attesa di connessioni

Poiché un sistema di elaborazione può fornire molteplici servizi, ognuno di essi è caratterizzato da una **porta** di ascolto

Ogni sistema hardware può ospitare l'esecuzione
di più servizi software



Internet: i server

Per richiedere accesso ad un dato servizio, l'indirizzo IP quindi non basta. Dobbiamo anche specificare il numero di porta:

«**indirizzo IP**»:«**porta**» (es. 131.175.187.72:**80**)

Esistono una serie di servizi Internet noti, ai quali è già stata assegnato per convenzione un numero di porta predefinito...

Port 20/21: File Transfer Protocol (FTP)

Port 22: Secure Shell (SSH) . . .

Port 25: Simple Mail Transfer Protocol (SMTP)

Port 53: *Domain Name System (DNS)*

Port 80: Hypertext Transfer Protocol (HTTP)

...



Internet: il World Wide Web (WWW)

Nel 1989 Tim Berners-Lee, fisico del CERN di Ginevra, popone il progetto di un'interfaccia grafica ipertestuale denominata World Wide Web (WWW)

Il **World Wide Web** è un sistema basato su pagine e ipertesti per la gestione di documenti su Internet

Il World Wide Web è oggi un enorme deposito di informazioni accessibili attraverso Internet:

- Per consultarle gli utenti usano **programmi di navigazione (browser)**, che utilizzano un'interfaccia di tipo "point-and-click"
- Le informazioni visualizzate dai browser sono **ipertesti multimediali**

ATTENZIONE: è un deposito NON
controllato di informazioni

Internet: il WWW e il Browser

Per la navigazione di questi ipertesti (siti Web) sono stati sviluppati software denominati **browser**

Es. Safari, Google Chrome, Mozilla Firefox, etc...

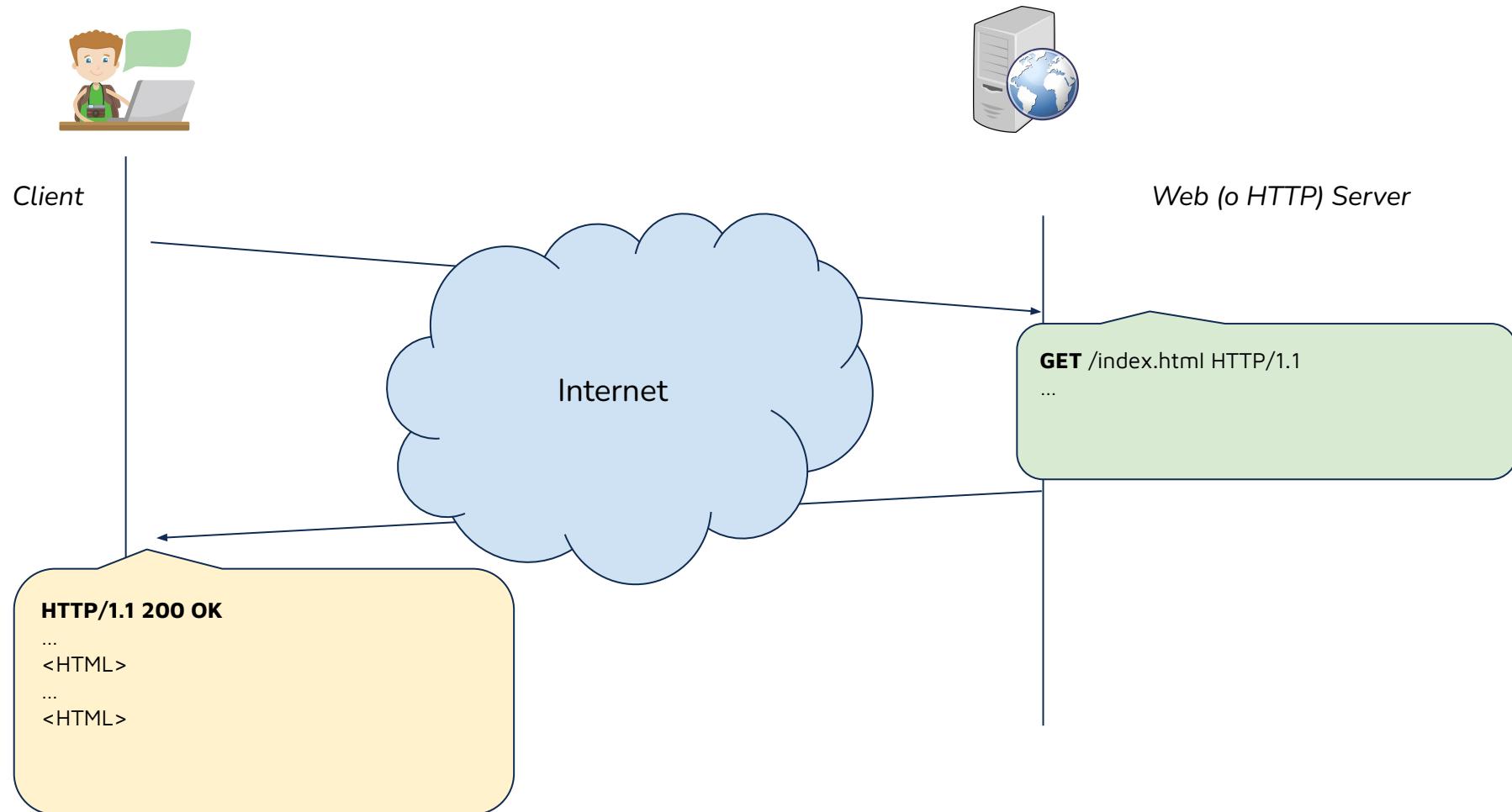


Il browser assolve due compiti principali

Inviare le richieste di accesso al server Web secondo il **protocollo HTTP(S)**

Gestire le risposte del server e visualizzare la pagina ottenuta, interpretando il **codice HTML**

HTTP: HyperText Transfer Protocol



HTTP: HyperText Transfer Protocol

HTTP (Hypertext Transfer Protocol) è il principale protocollo per la trasmissione di informazioni sul Web

E' basato su una *architettura client-server*

- Il client effettua delle richieste attraverso l'invio di messaggi

- Il server risponde alle richieste

- Richieste e risponde sono strutturate in un opportuno formato testuale (es. HTML)



Quando la richiesta di accesso ad un sito Web ha successo, il server include nella risposta il testo necessario alla sua visualizzazione

Tale testo è codificato in **HTML**, linguaggio di markup (di ‘contrassegno’ o ‘di marcatura’), che permette di descrivere gli elementi e l’aspetto di un sito web

Quando la richiesta di accesso ad un sito Web ha successo, il server include nella risposta il testo necessario alla sua visualizzazione

Tale testo è codificato in **HTML**, linguaggio di markup (di ‘contrassegno’ o ‘di marcatura’), che permette di descrivere gli elementi e l’aspetto di un sito web

Il browser interpreta e processa i cosiddetti **tag** (e.g., `<tag>`) HTML, per poi visualizzare la pagina risultante

Inserimento di un’immagine in un dato punto, formattazione di un paragrafo o intestazione, ecc...

Il linguaggio HTML non è un linguaggio di programmazione in quanto non risolve un algoritmo ma si occupa della rappresentazione dei dati, è un linguaggio dichiarativo

HTML: HyperText Markup Language

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Informatica Applicata</title>
</head>
<body>
    <h1>Esempio di pagina HTML</h1>
    <p align="center">Questa è una banale pagina HTML </p>
</body>
</html>
```



HTML: HyperText Markup Language

Esempio di pagina HTML

Questa è una banale pagina HTML

<https://www.w3schools.com/html/default.asp>



Le pagine HTML mostrarono ben presto i loro limiti

- Contenuti statici

- Interazione con l'utente limitata (si può solo navigare tra le pagine)

Nei primi anni 90' nacque il linguaggio Java, che venne impiegato anche nelle pagine HTML, encapsulato in oggetti detti **applet**

Nel 1995, Netscape creò un proprio browser che integrava l'interprete di un linguaggio di **scripting**, per interagire con le componenti della pagina e le applet Java

- Altri (Microsoft) imitarono la scelta introducendo a loro volta un proprio linguaggio

<https://developer.mozilla.org/en-US/docs/Web/JavaScript>

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Informatica Applicata</title>
    <script type="text/javascript">
        alert("Modulo di Informatica Applicata");
    </script>
</head>
<body>
    <h1>Esempio di pagina HTML</h1>
    <p align="center">Questa è una banale pagina HTML </p>
</body>
```

Possiamo quindi eseguire del codice JavaScript in fase di apertura della pagine, inserendo delle righe in un tag **<script>**

Oppure usare sempre un tag **<script>** per referenziare del codice JavaScript scritto in un file esterno

```
<script type="text/javascript" src="scripts/run.js"></script>
```

- Il secondo approccio è solitamente da preferire
- Il codice JavaScript separato dal HTML produce una organizzazione di file più pulita

JavaScript è quindi come linguaggio *interpretato* (come tutti i linguaggi di scripting)

L'interprete JavaScript è parte integrante delle funzionalità del browser

Da diversi anni, JavaScript ha superato i confini della programmazione Web/HTML e viene impiegato in altri contesti

Mai sentito parlare di *nodeJS*?

JavaScript: usare un HTTP server locale

Quindi per sviluppare applicazioni JavaScript senza restrizioni abbiamo bisogno di un server?

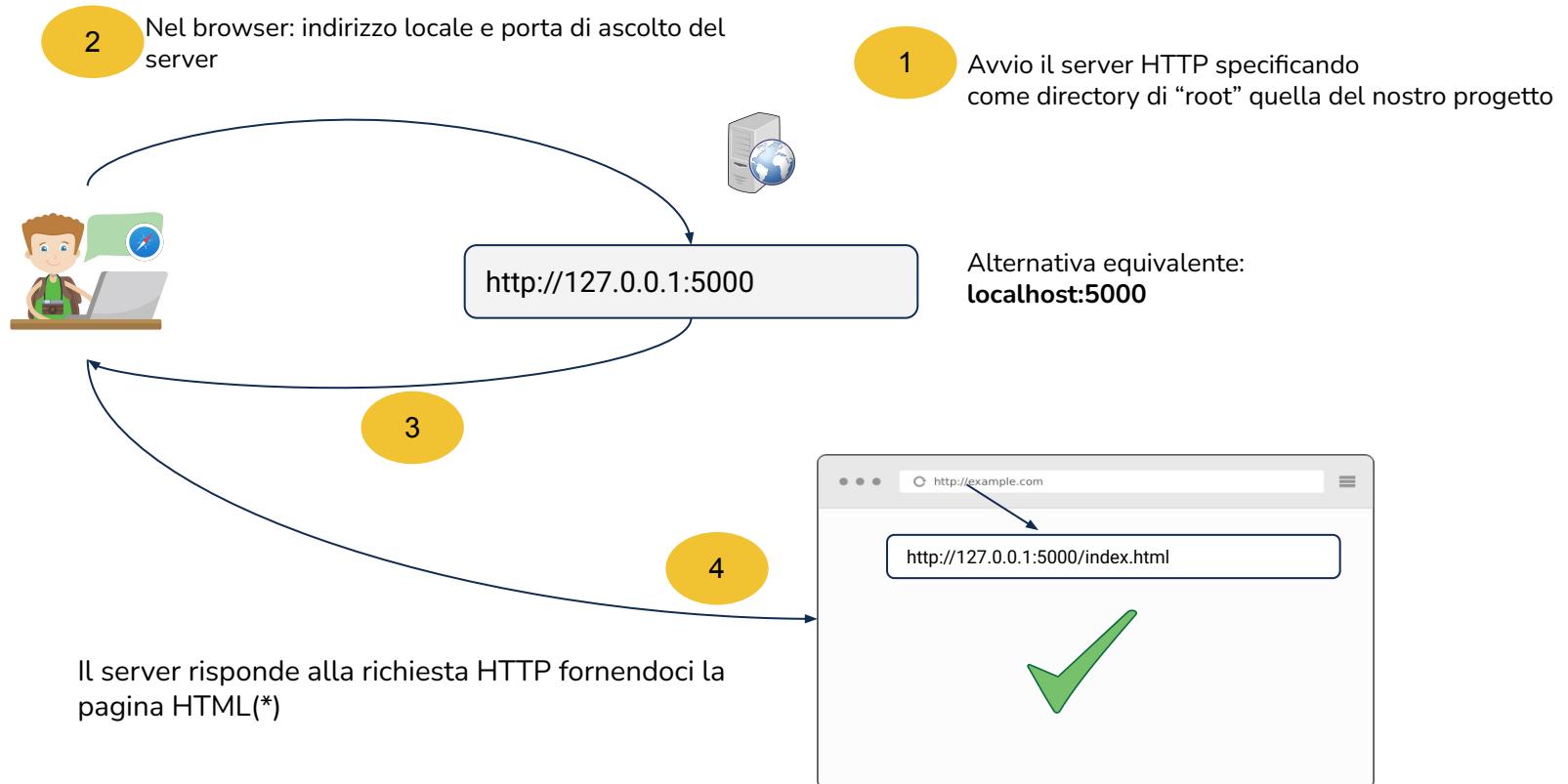
Sì, ma possiamo installarlo “localmente” sul nostro sistema ed avviarlo come un programma

Una volta in esecuzione, il server HTTP rimarrà in ascolto sulla **porta** specificata in attesa di connessioni

...usiamo una porta diversa dalla 80

Per l'apertura della nostra pagina con codice JavaScript dovremmo quindi connetterci al nostro server HTTP locale

JavaScript: usare un HTTP server locale



JavaScript: strumenti per il corso

Per questo corso, avremo bisogno di installare un server HTTP

Fortunatamente, l'ambiente di sviluppo **Visual Studio Code (VSCode)** ci fornisce delle funzionalità ad-hoc per lo sviluppo di applicazioni HTML/JavaScript

VSCode include infatti una estensione (**Live Preview**) attraverso la quale possiamo avviare un server HTTP locale e avviare con un click l'applicazione che stiamo sviluppando

Informatica Self-Assesment

1. ~~Informatica Self Assesment~~

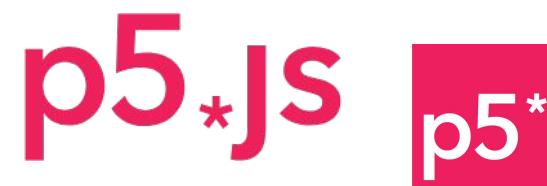


2. Local dev with VSCode, Indice html for IA2526

3. Create our Webpage with Githubpages



4. Draw with P5, P5 Live Editor



<https://editor.p5js.org/>

Github Desktop Installation Party

<https://desktop.github.com/download/>



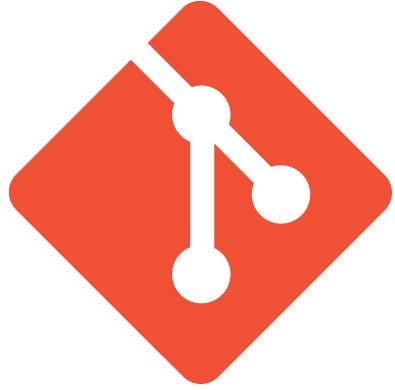


Git

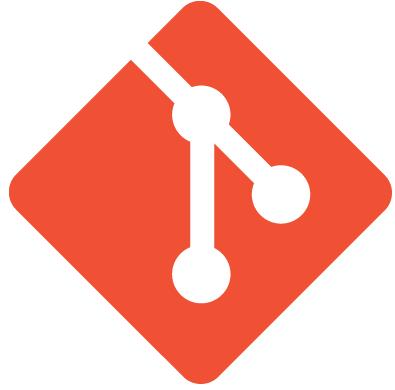
- GitHub
- GitHub Desktop



GitHub Pages



git



remote

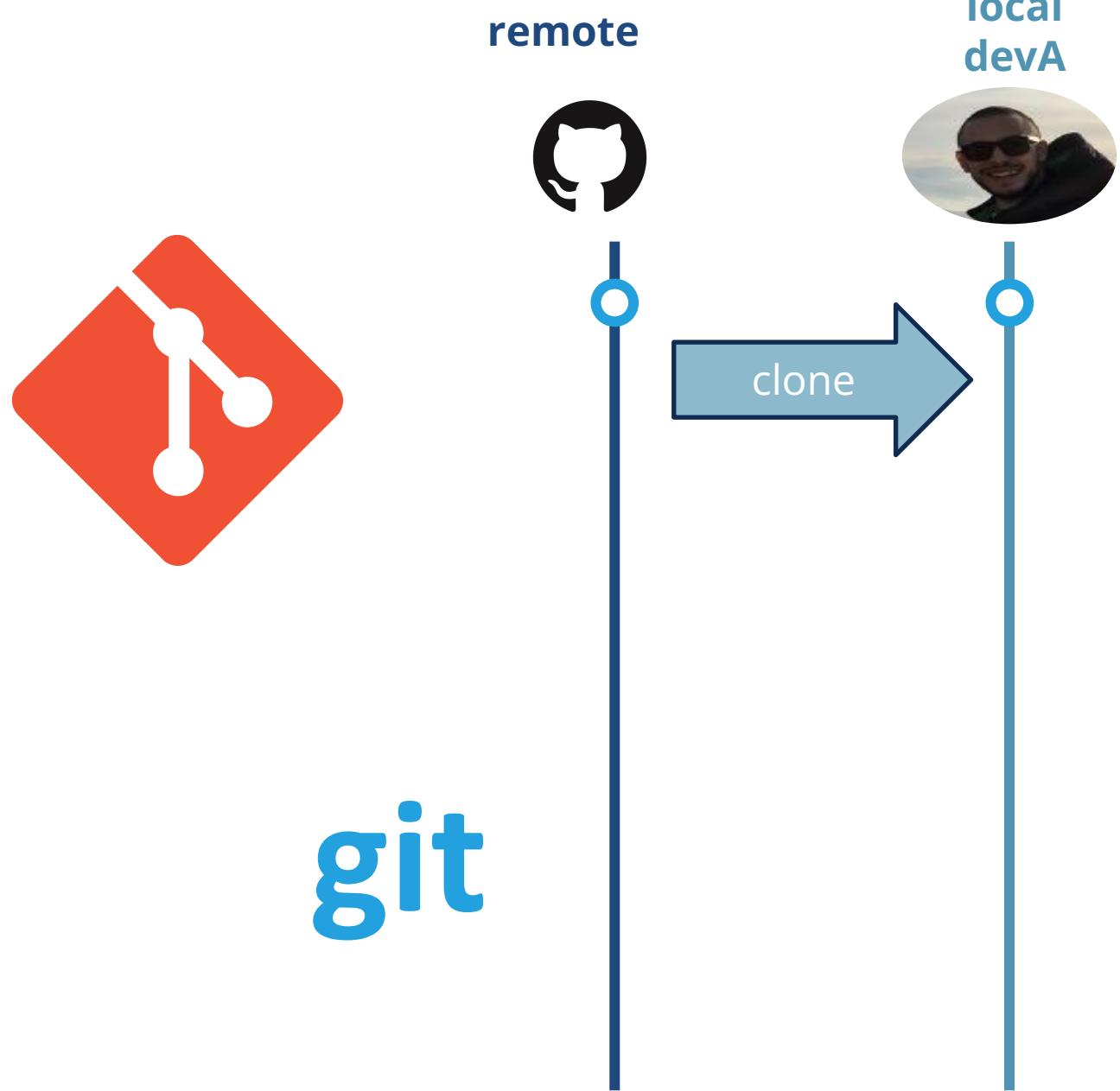


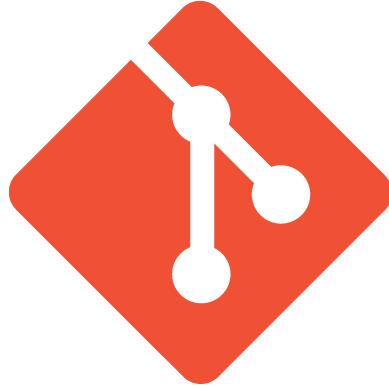
git



local
devA



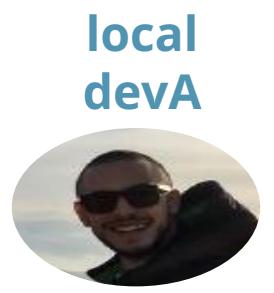


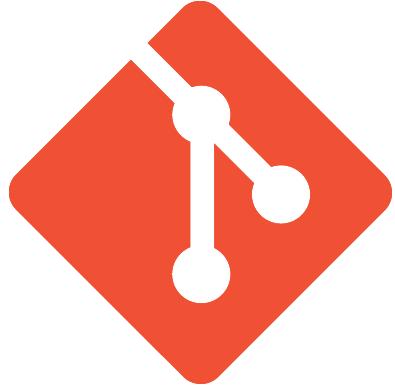


remote



git





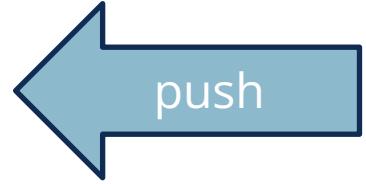
remote

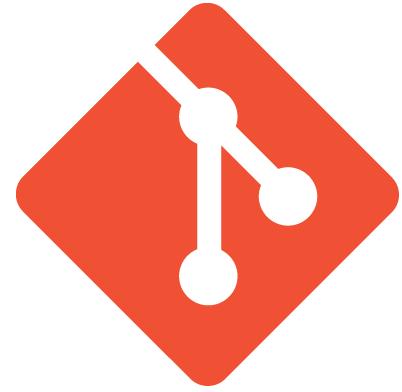


local
devA



git





remote



git

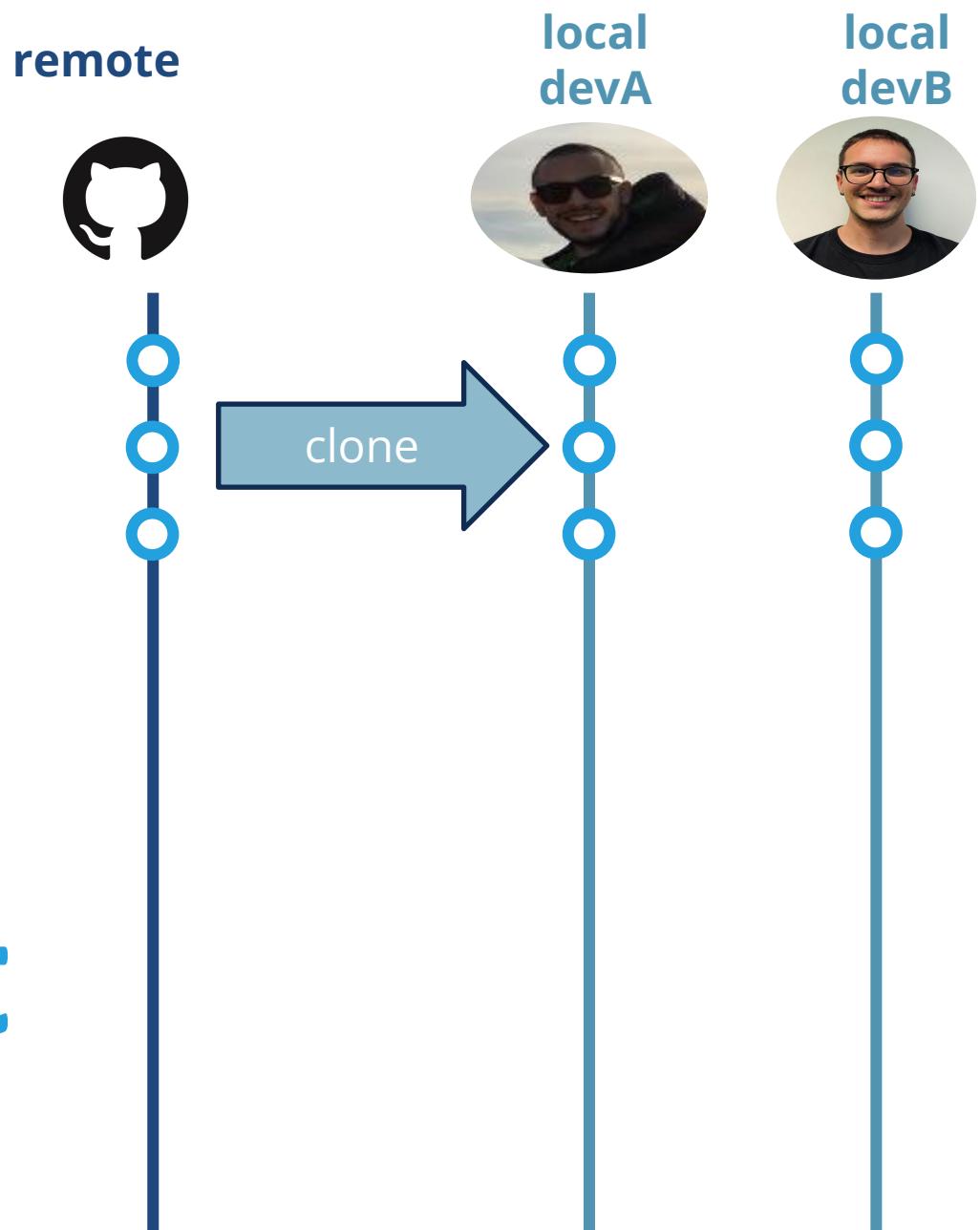
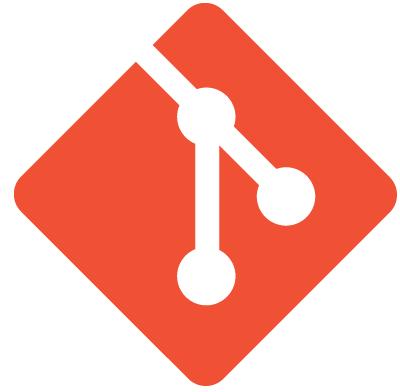


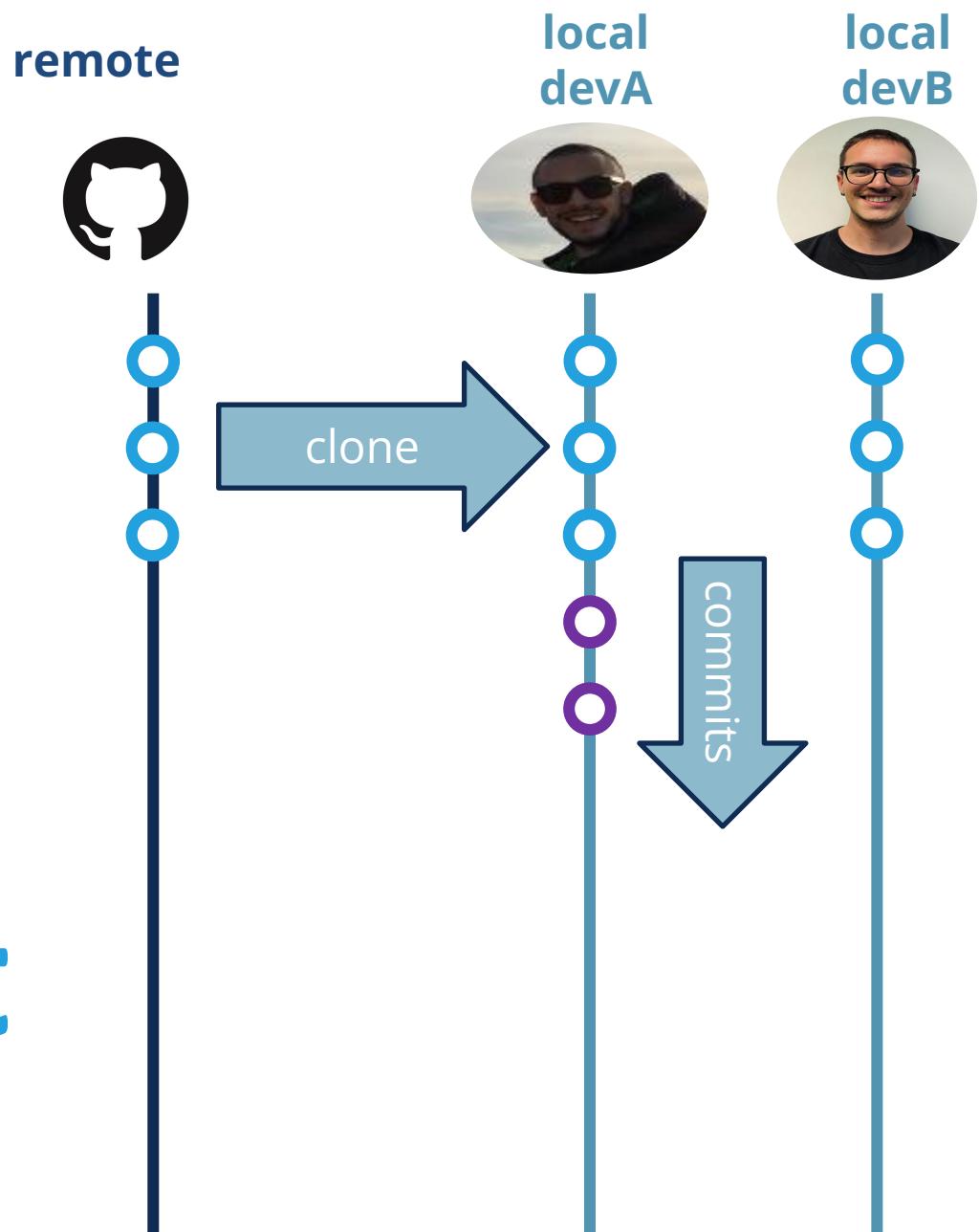
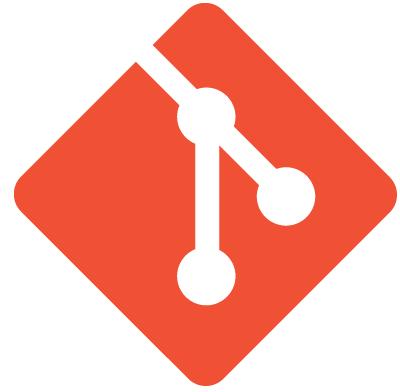
local
devA

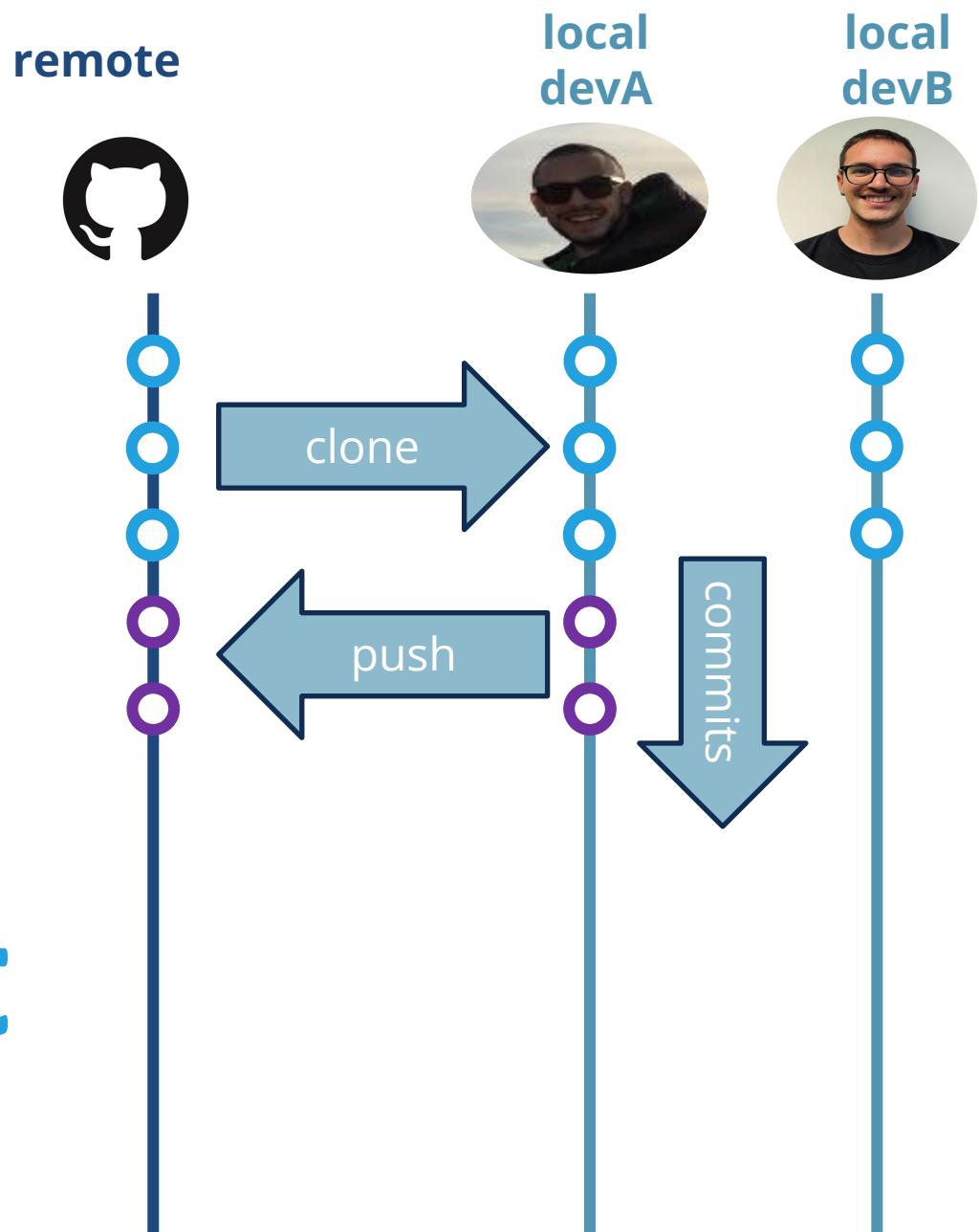
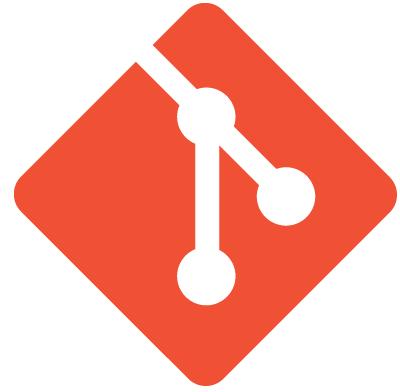


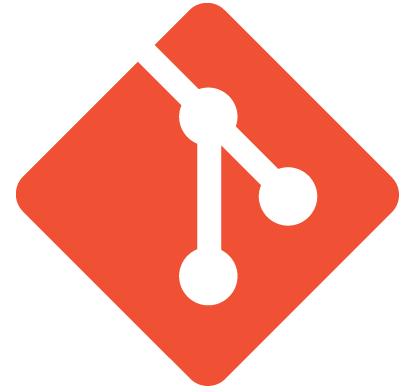
local
devB











git

remote

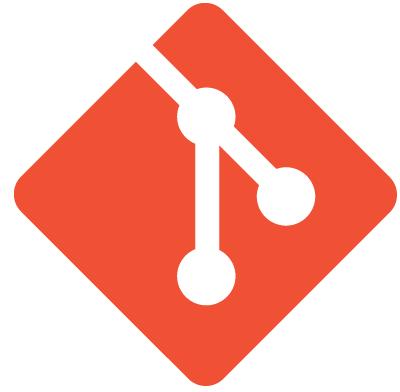


local
devA



local devB





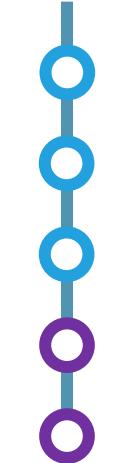
remote



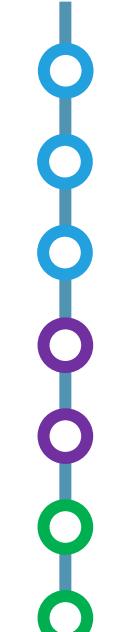
git



local
devA



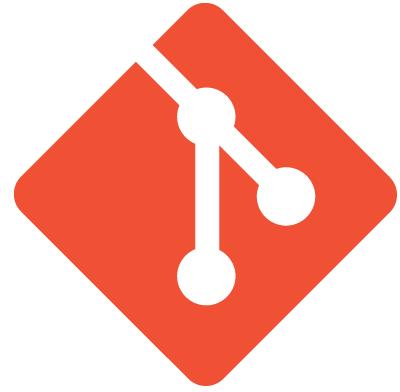
local
devB



commits



local
devC



git

remote



local
devA

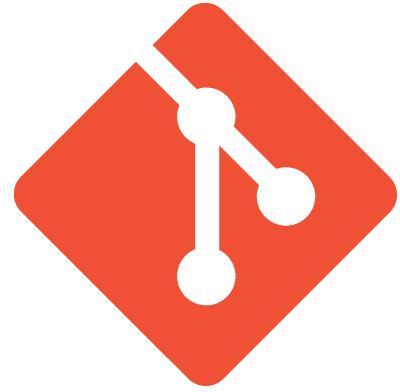


local
devB

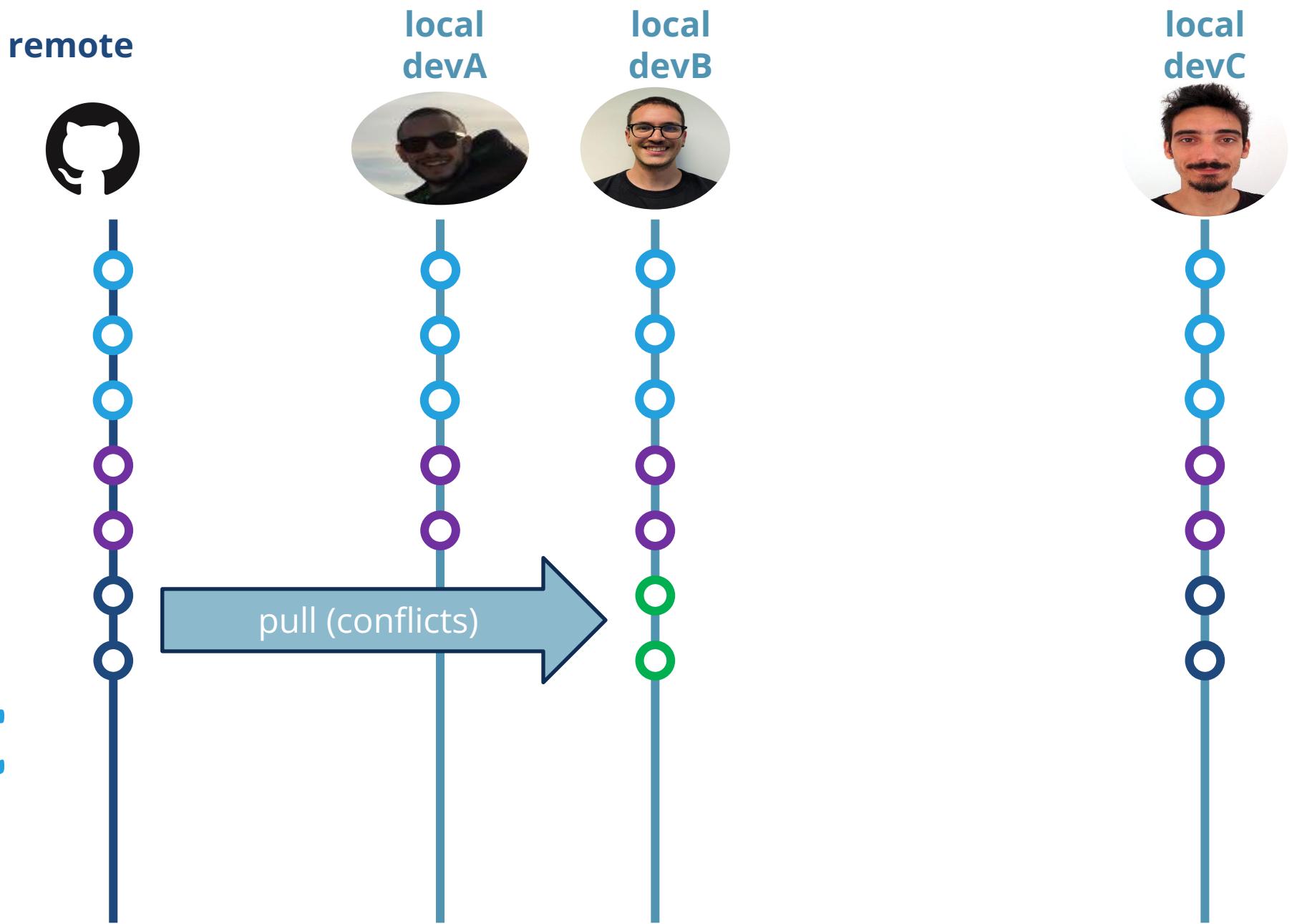


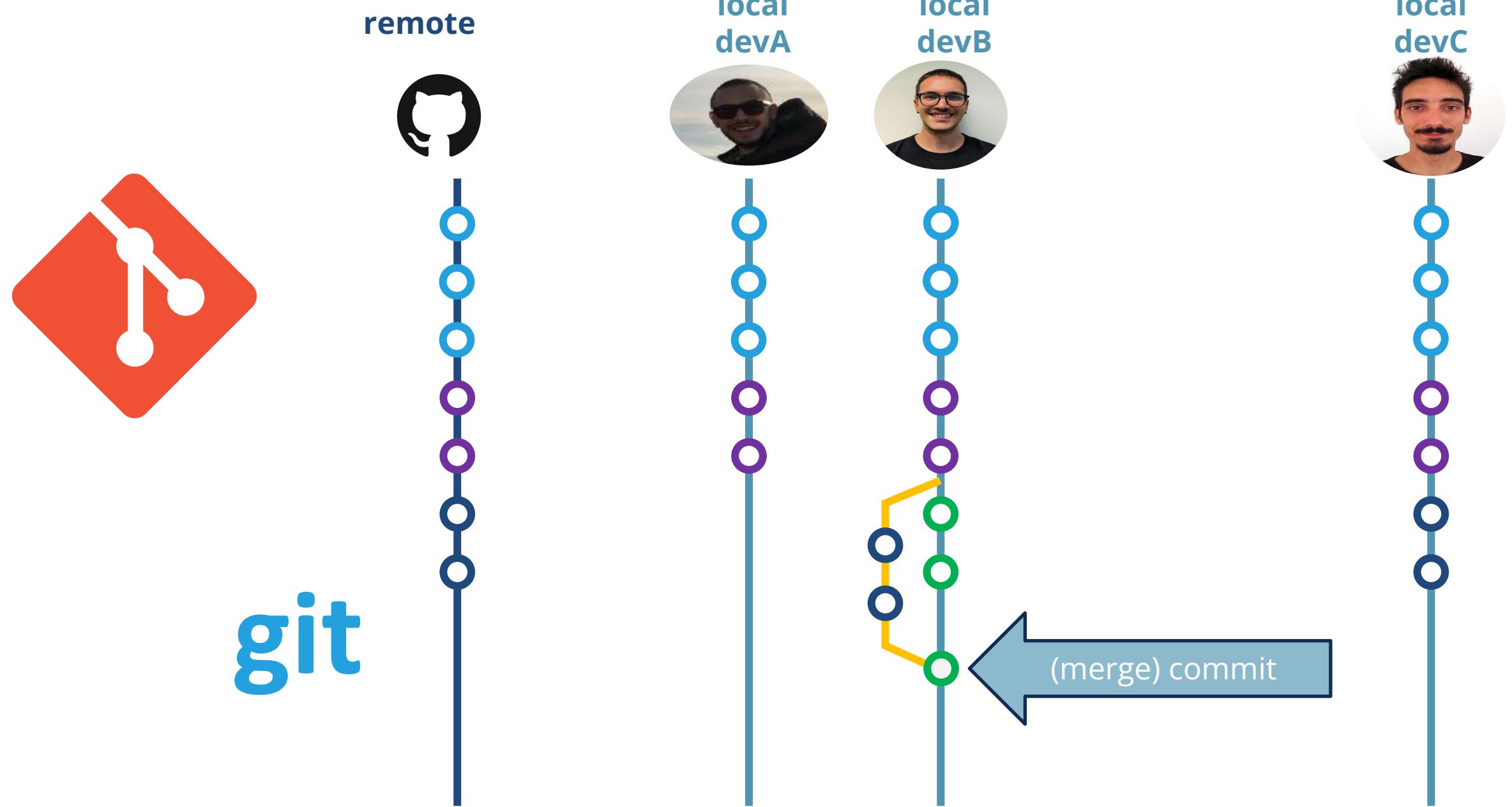
local
devC

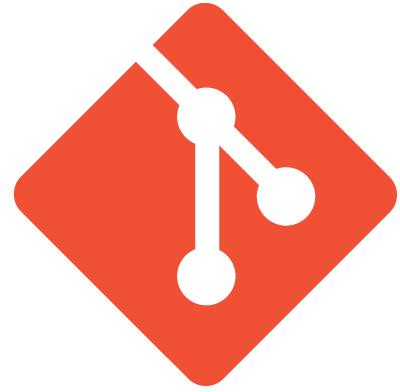




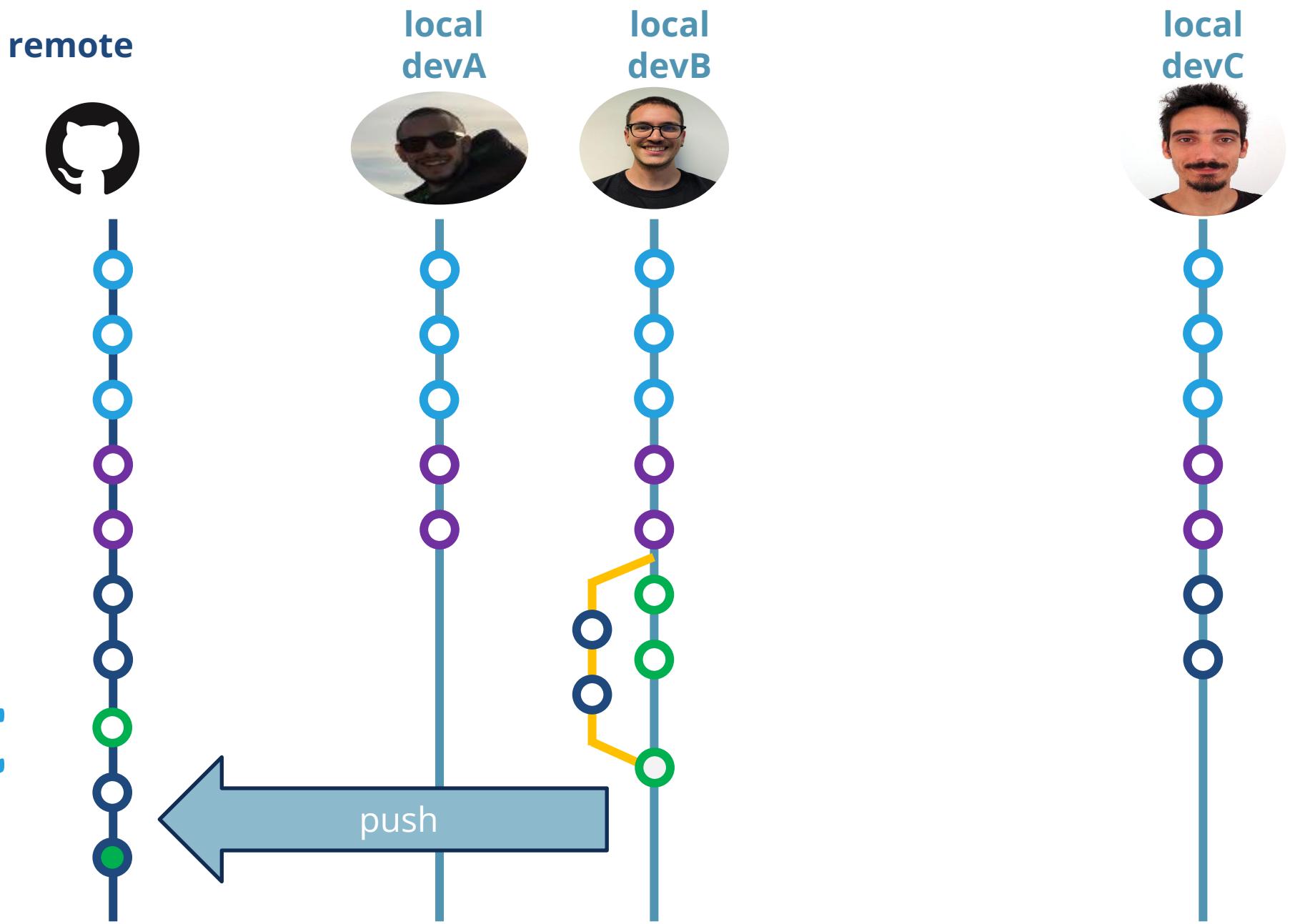
git







git



Demo Time :)



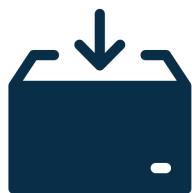


Create, edit, delete file(s)



Compare the changes

`git status`
`git diff`



Stage the changes for commit

`git add file-name`



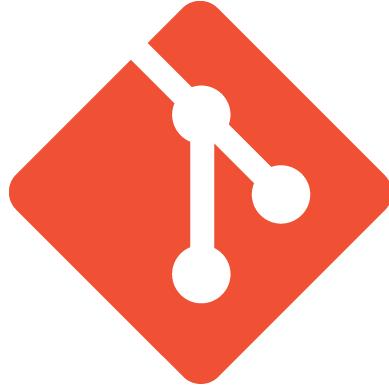
Commit the staged files

`git commit -m "The commit message"`



Keep the history in check

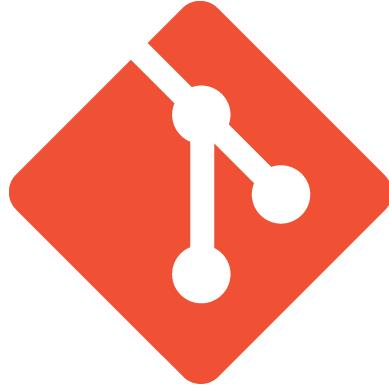
`git log`



git

Single developer

- Tracks evolution
- Builds history
- Navigate the history



git

Single developer

- Tracks evolution
- Builds history
- Navigate the history

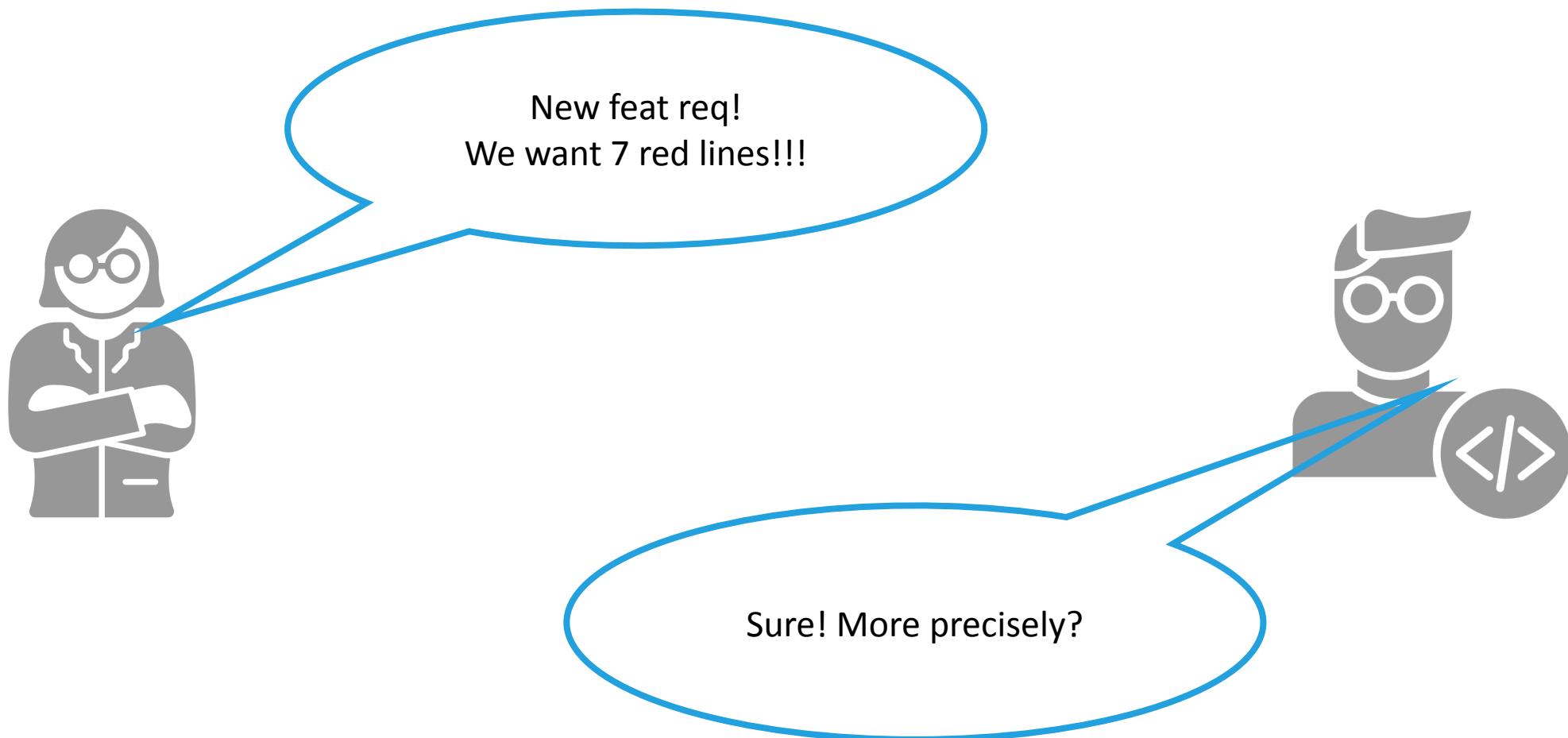
Team of developers

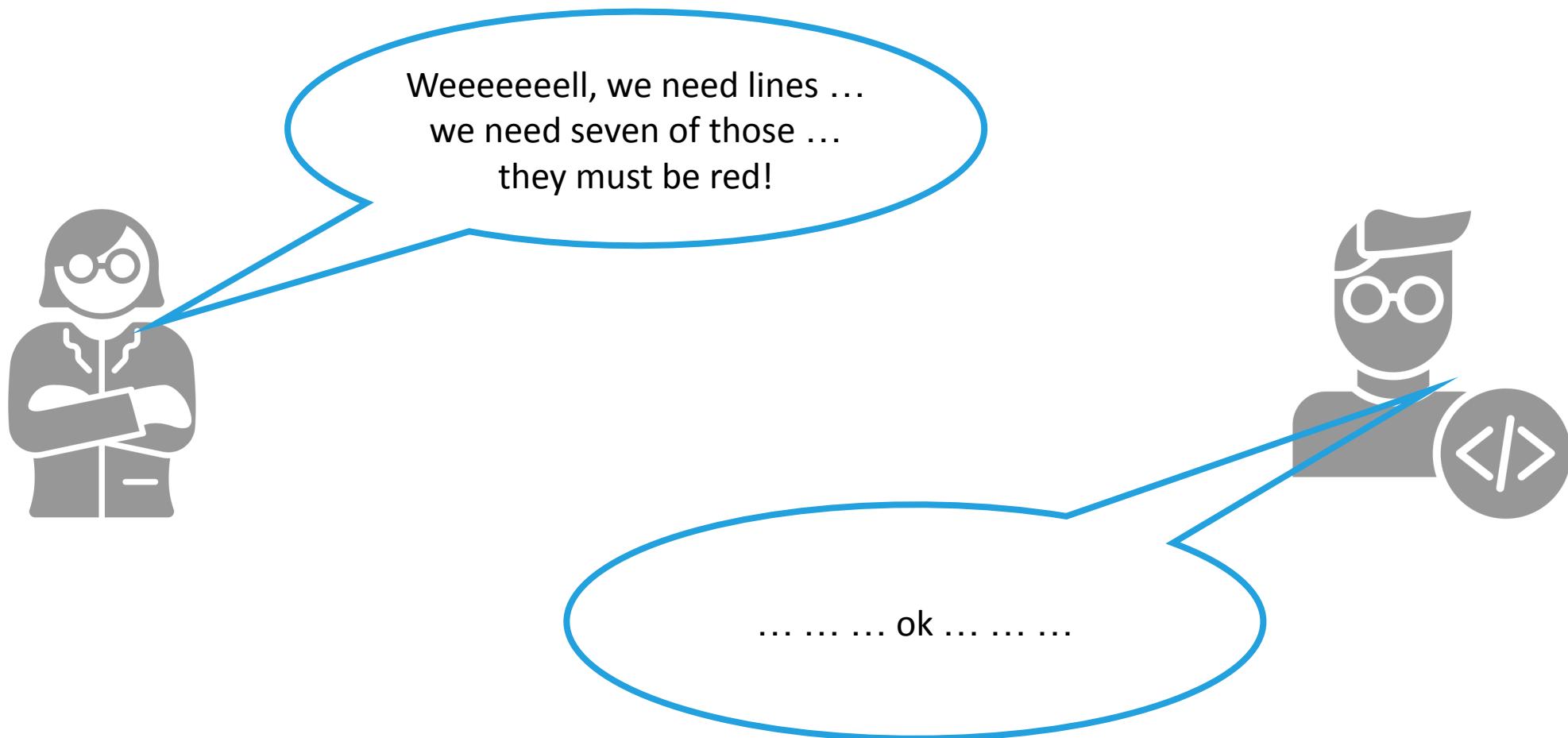
- Allow concurrent development
- Track the responsible
- Support in merging changes

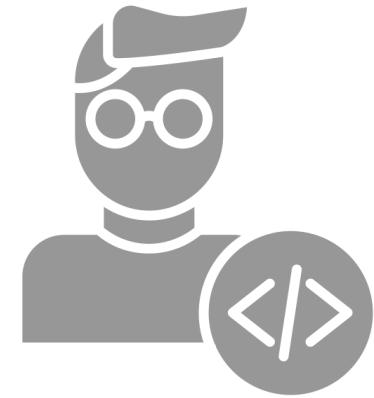
(Collaborative and Asynchronous) Development

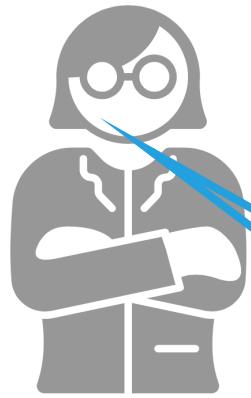
(vaguely) Inspired by «The Expert»

<https://www.youtube.com/watch?v=BKorP55Aqva>

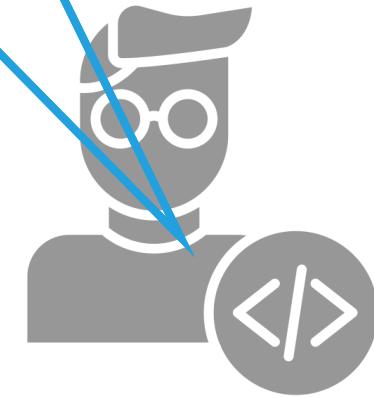




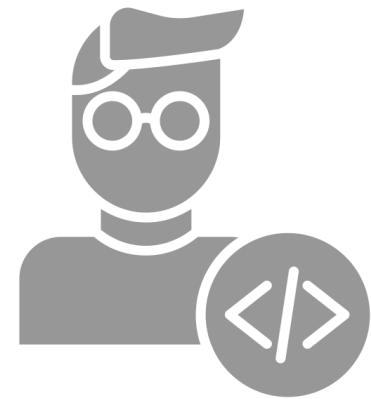


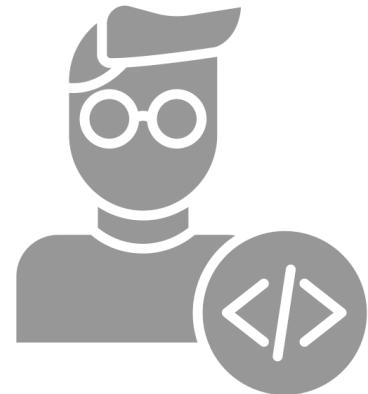
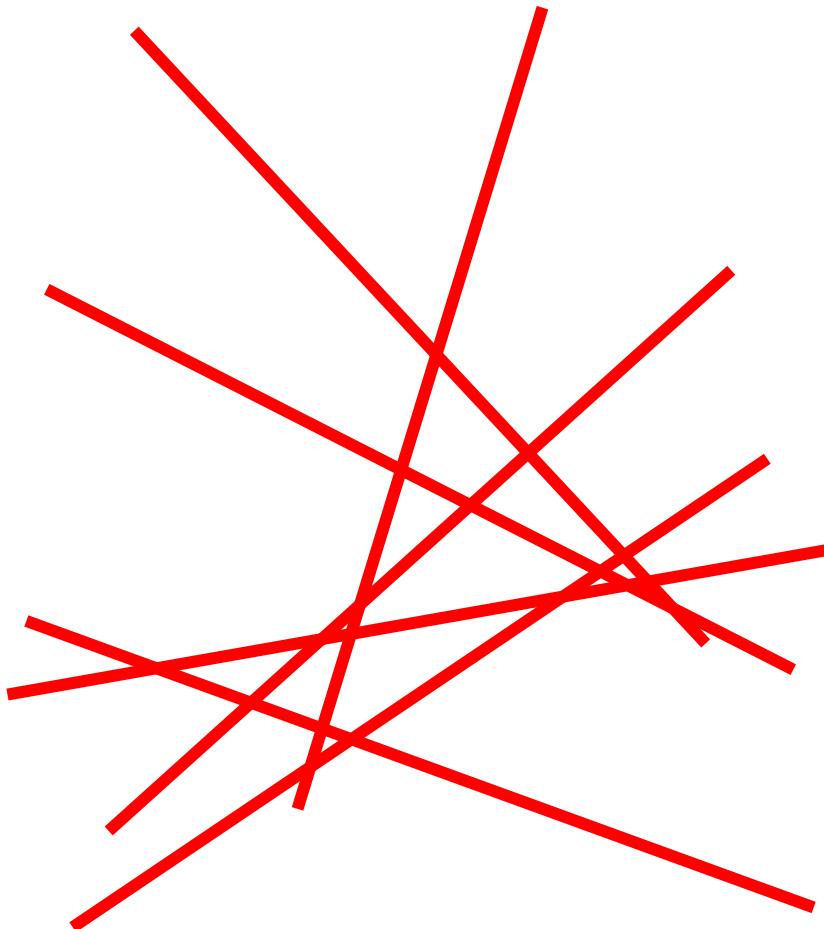


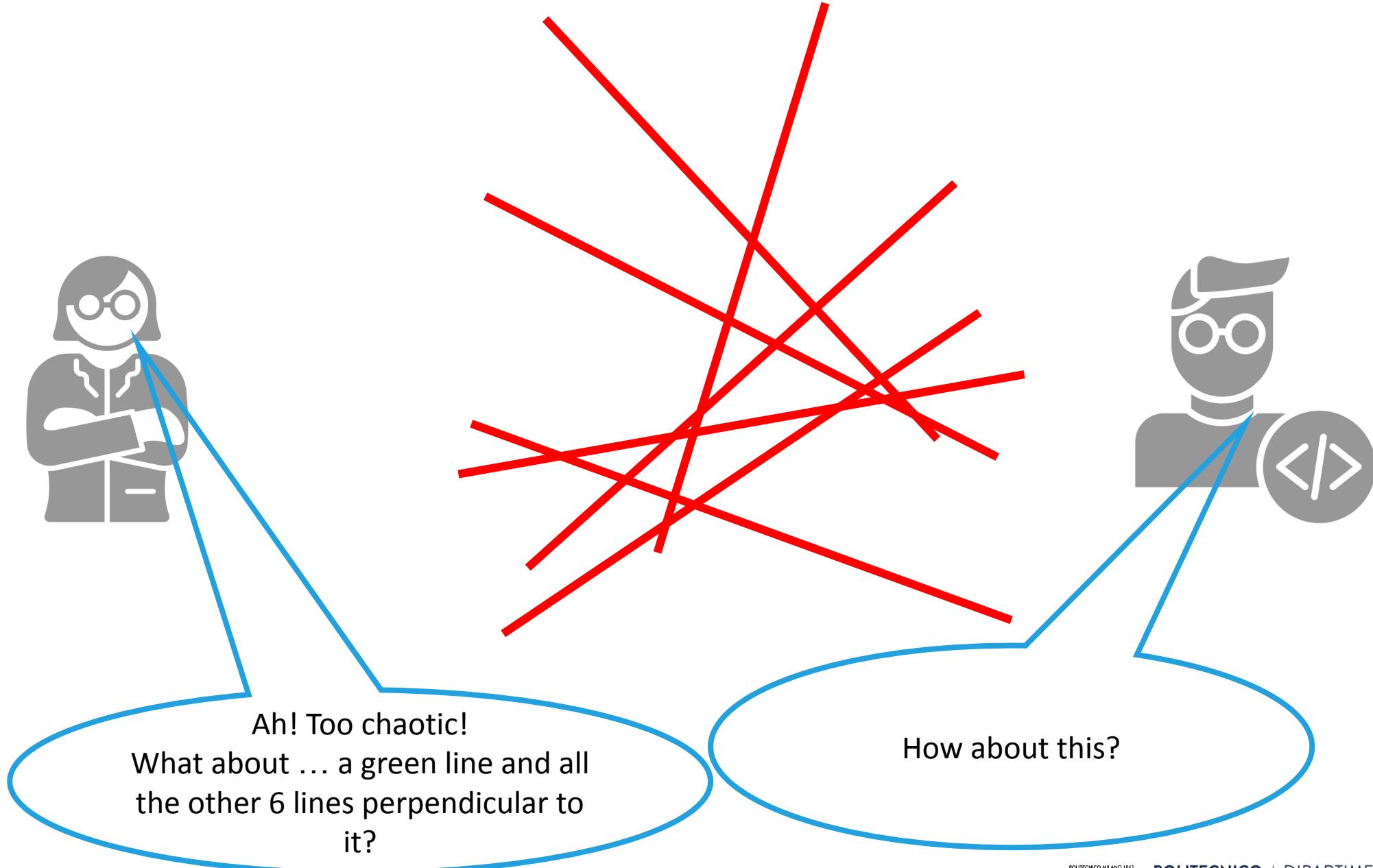
Booo!

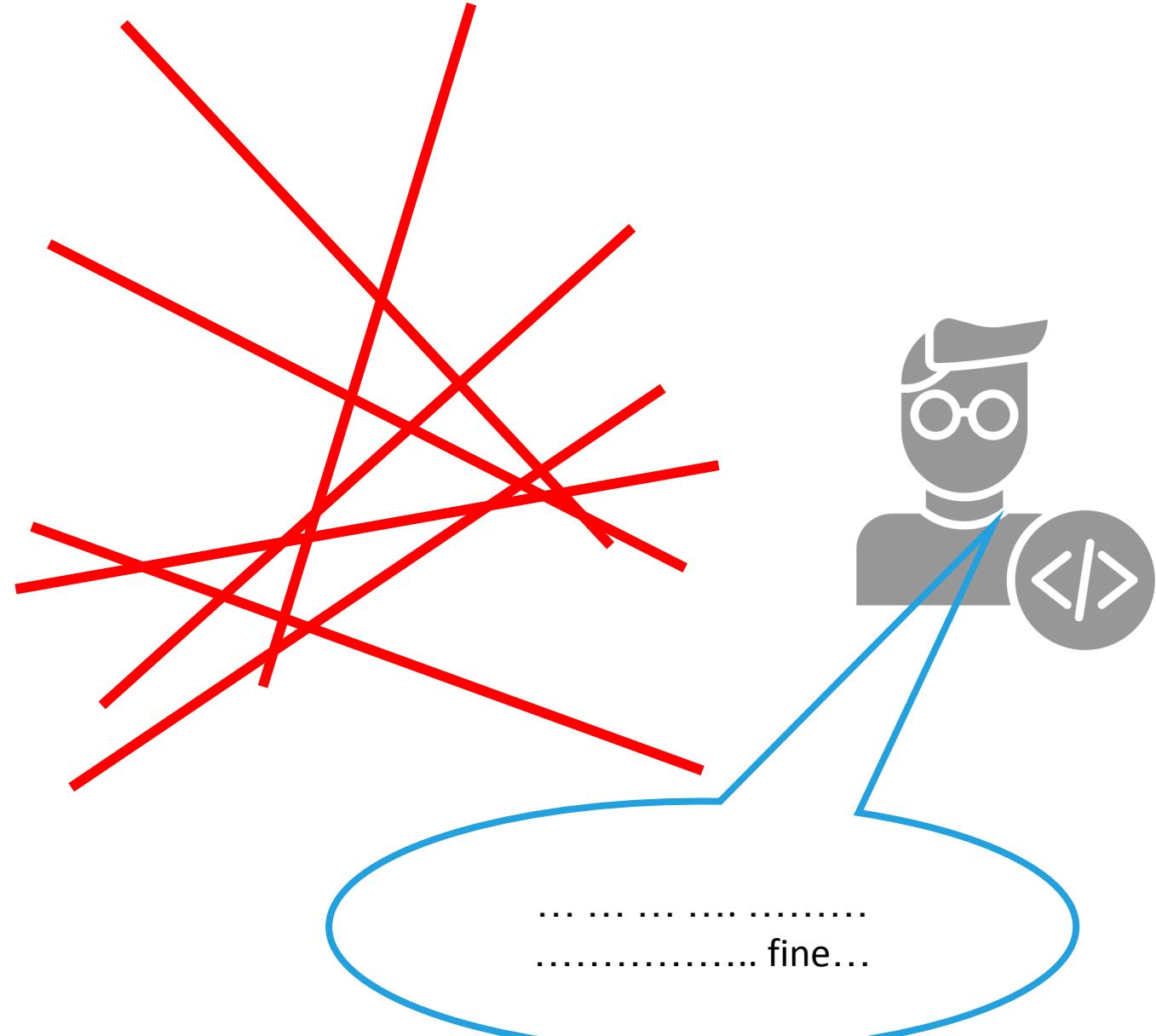


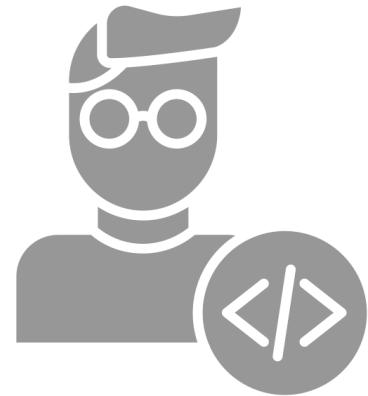
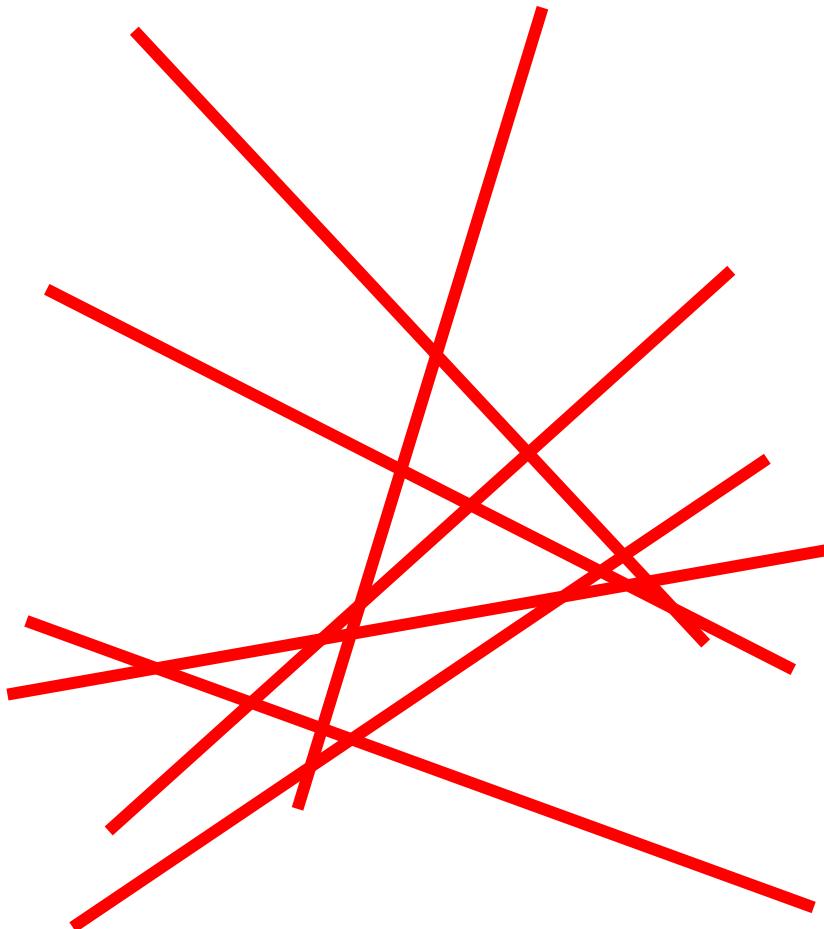
Yeeeeeeeah... not exactly
... they are not crossing each
other ...

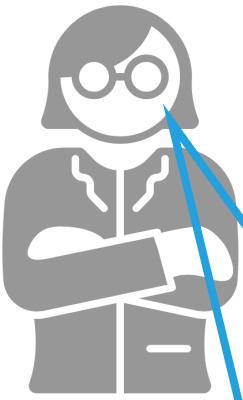




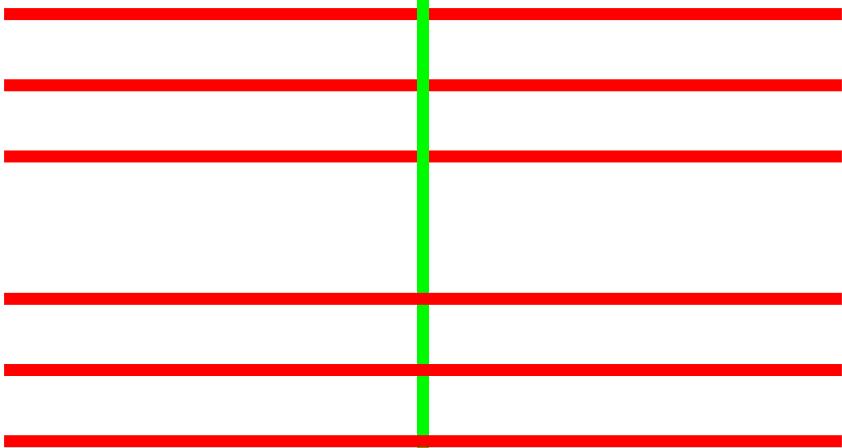




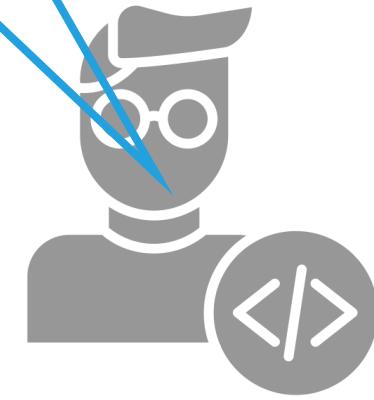


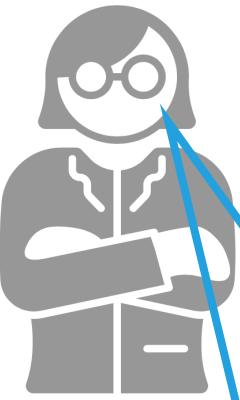


Sorry, uninspiring at all
yeah . . . let's go back to the scrambled
one you did before!

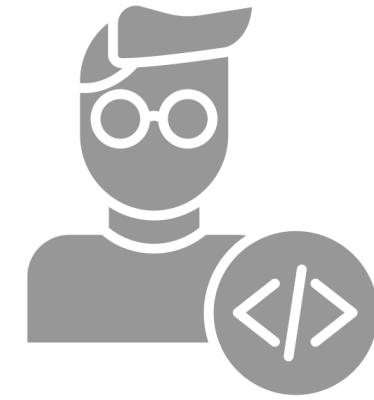
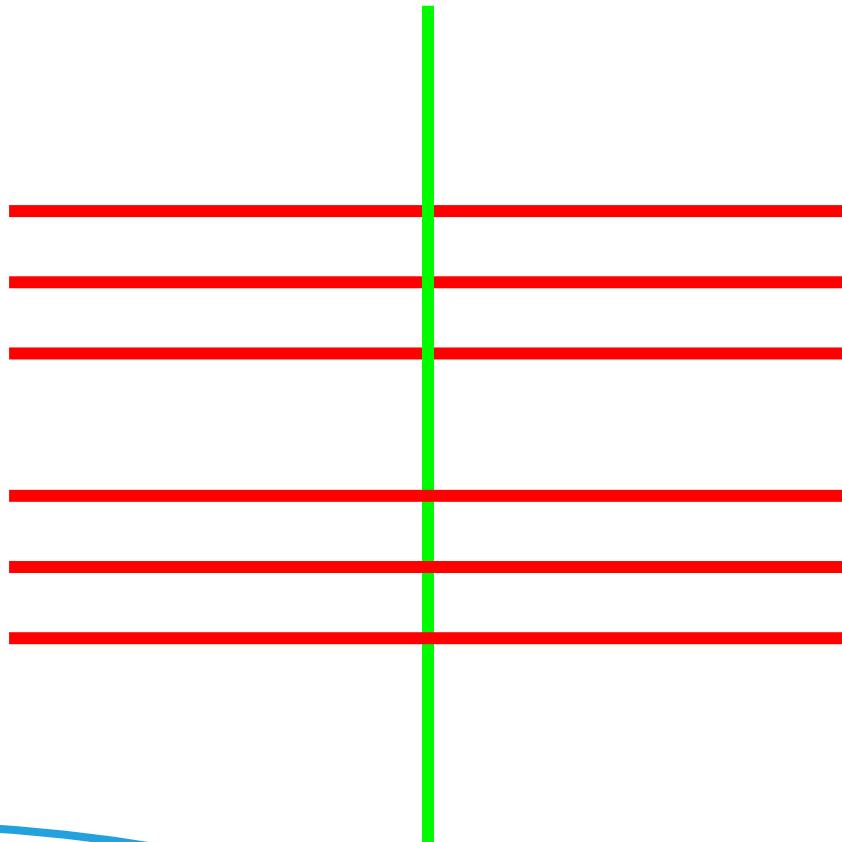


What about now?

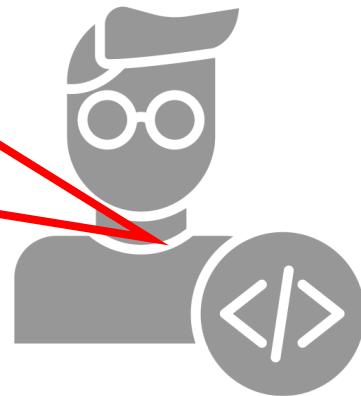


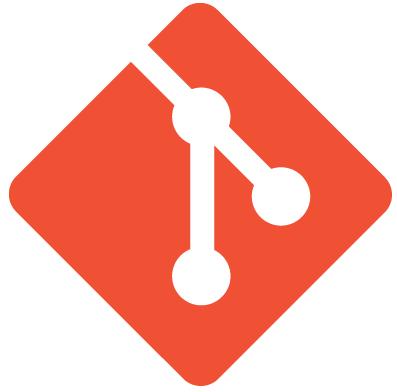


That
was
PERFECT

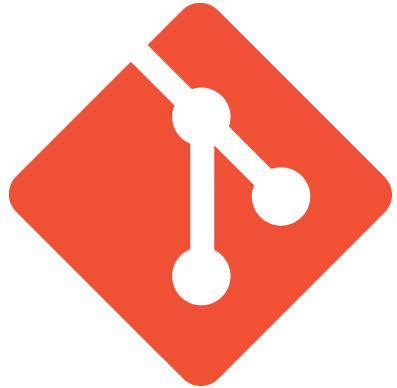


HOW?!?!!?





git



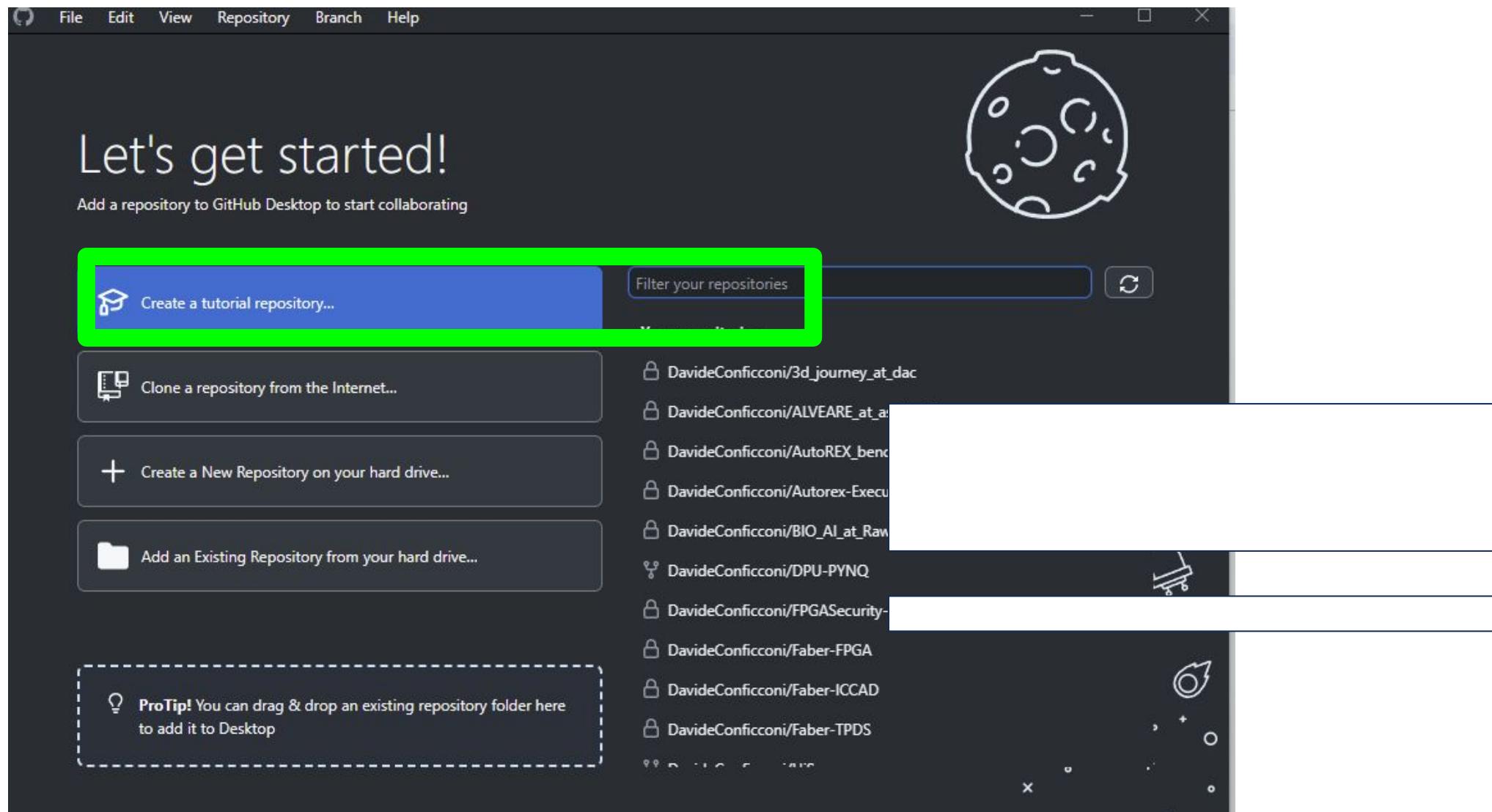
<https://desktop.github.com/>

git

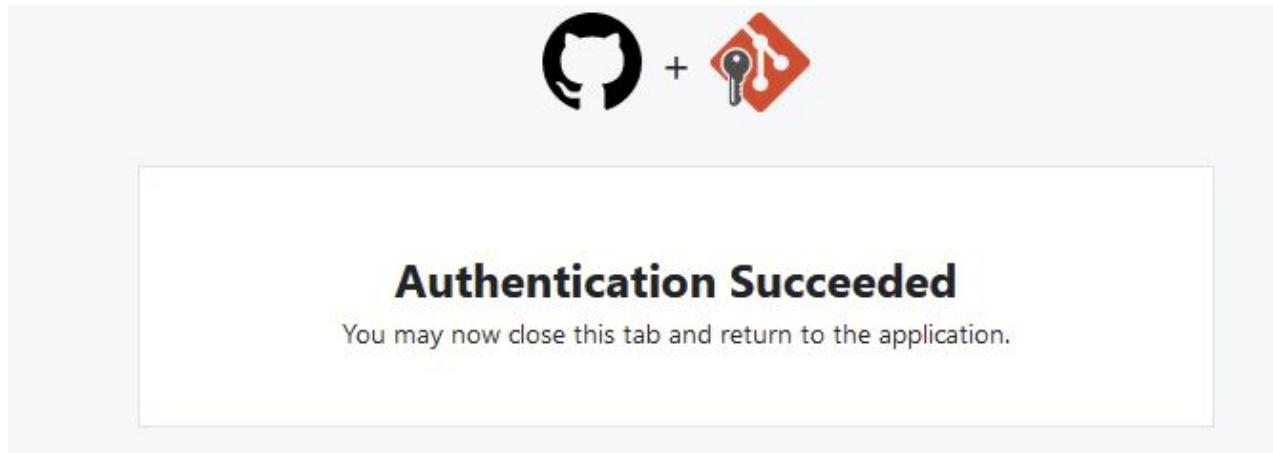
Demo Time :)



GitHub Desktop Getting Started

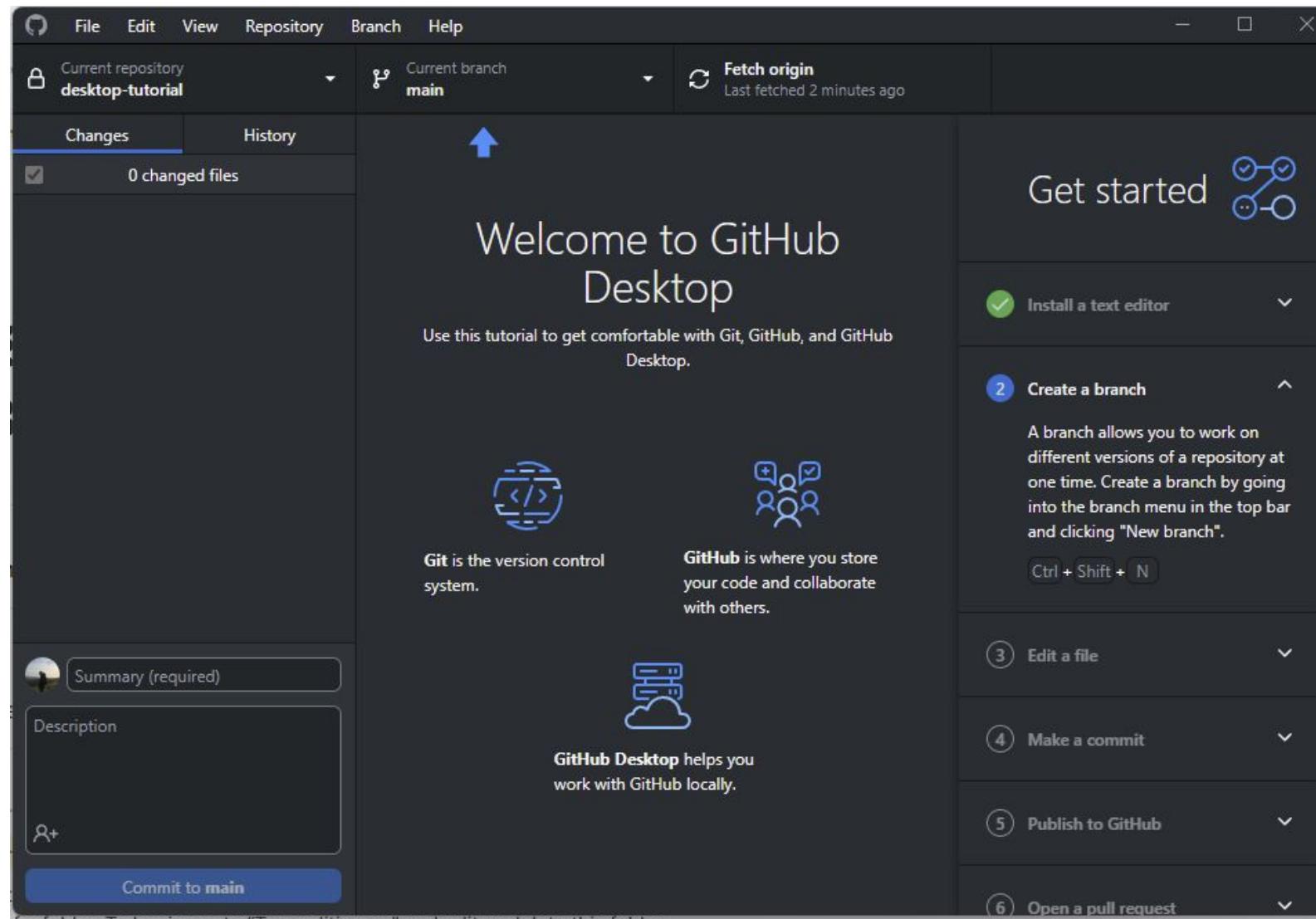


GitHub Desktop Getting Started



<https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent>

GitHub Desktop Sample View



github.com View

The screenshot shows a GitHub repository page for 'desktop-tutorial'. The repository is private, as indicated by the 'Private' badge. It has 1 branch ('main') and 0 tags. The 'Code' tab is selected. The repository contains one commit from 'DavideConficconi' titled 'Initial commit' (commit hash: 3edd951, 17 minutes ago). The README.md file shows the content 'Welcome to GitHub Desktop!'. The 'About' section describes it as a 'GitHub Desktop tutorial repository' with 0 stars, 1 watching, and 0 forks. The 'Releases' section indicates no releases have been published. The 'Packages' section also indicates no packages have been published.

DavideConficconi / desktop-tutorial

Type to search | > | +

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

desktop-tutorial Private

Unwatch 1 Fork 0 Star 0

main 1 branch 0 tags Go to file Add file Code

DavideConficconi Initial commit 3edd951 17 minutes ago 1 commit

README.md Initial commit 17 minutes ago

README.md

Welcome to GitHub Desktop!

This is your README. READMEs are where you can communicate what your project is and how to use it.

Write your name on line 6, save it, and then head back to GitHub Desktop.

About

GitHub Desktop tutorial repository

- Readme
- Activity
- 0 stars
- 1 watching
- 0 forks

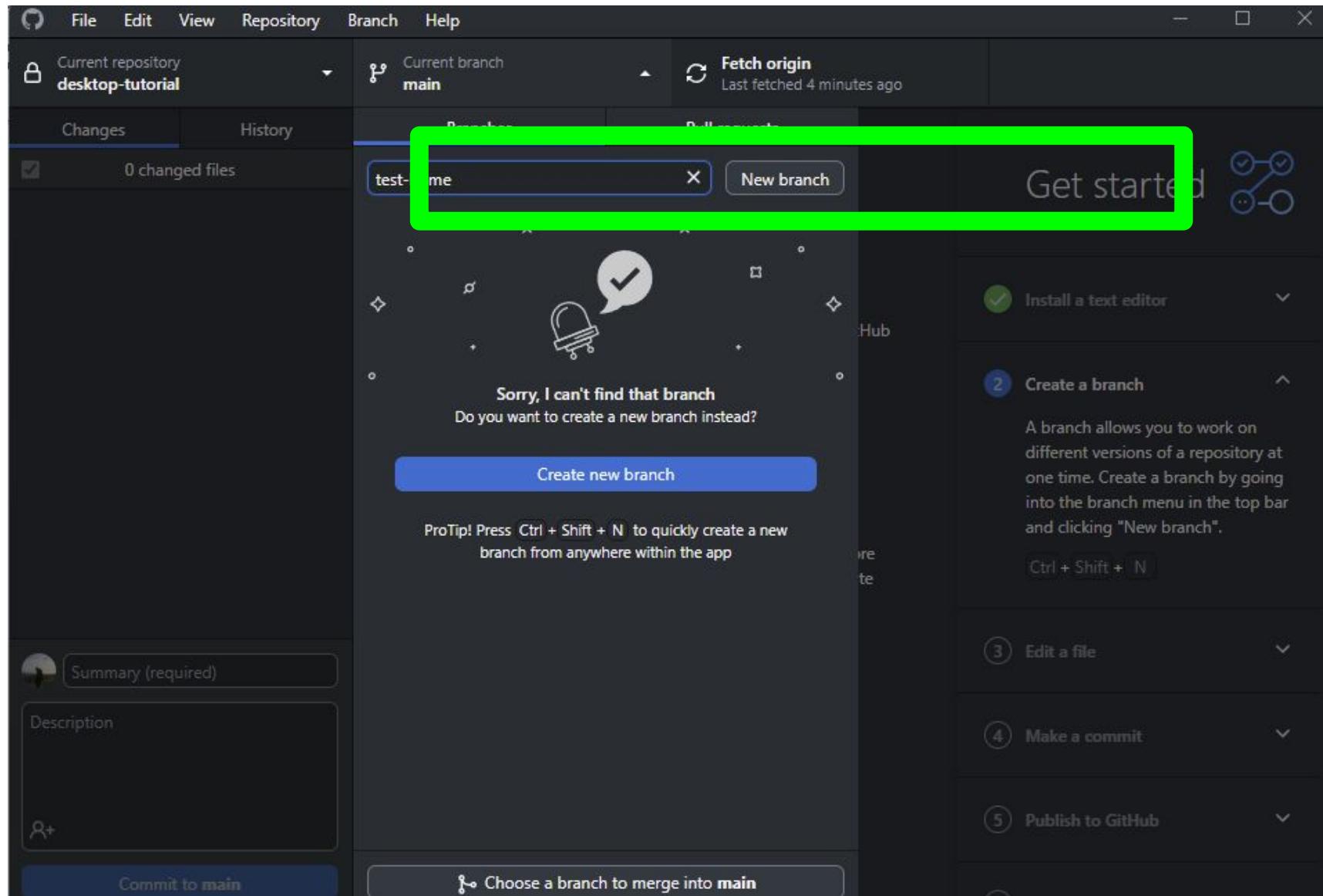
Releases

No releases published

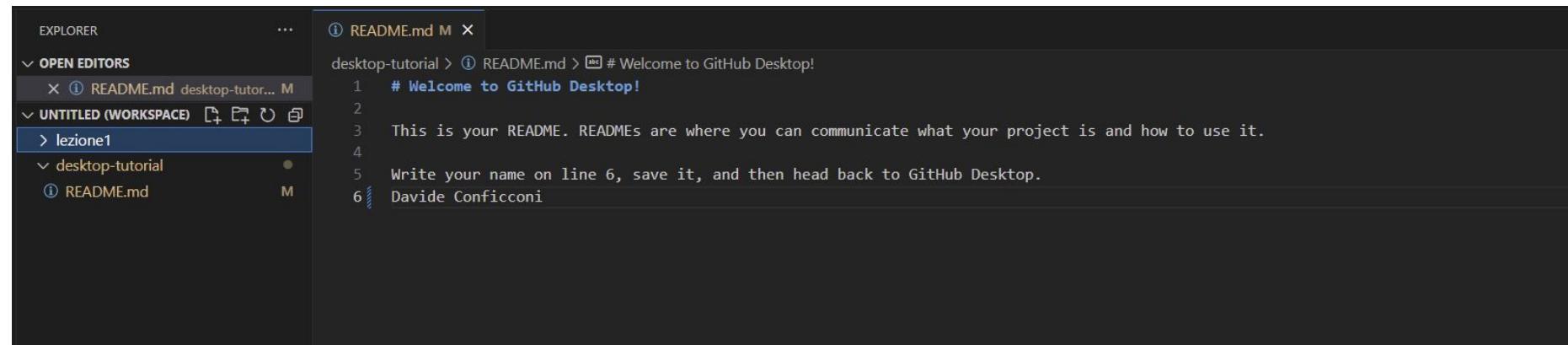
Packages

No packages published

GitHub Desktop New Branch

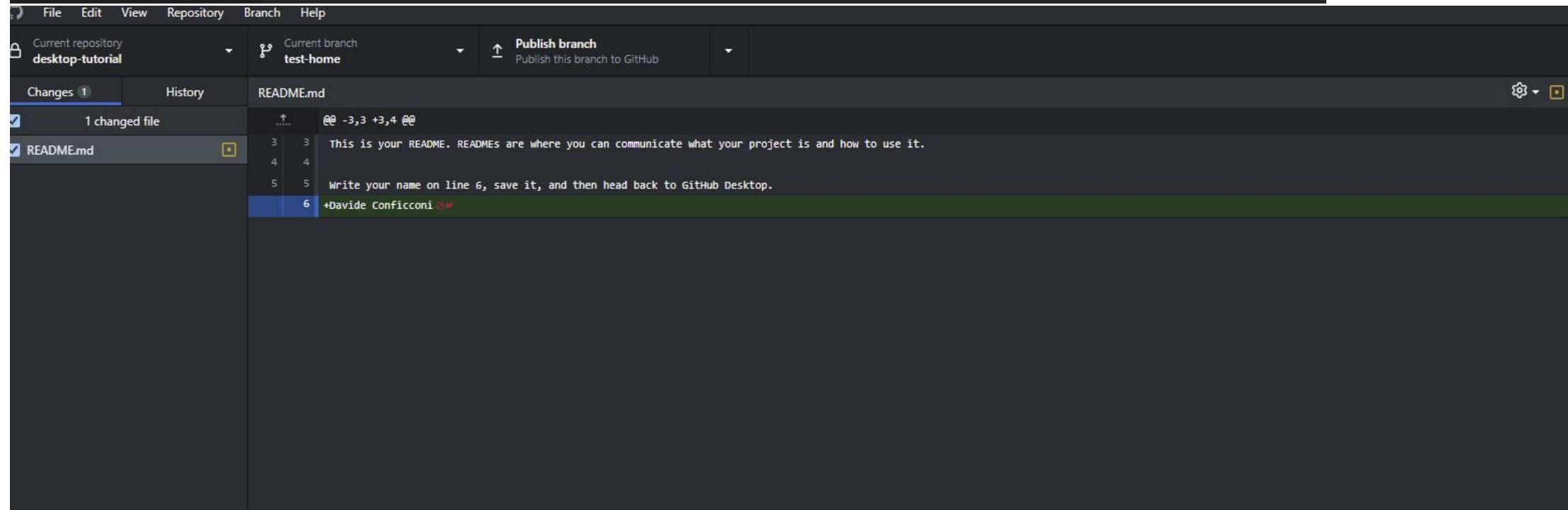


GitHub Desktop Local Modify



The screenshot shows the GitHub Desktop application interface. In the top left, the 'EXPLORER' sidebar lists 'OPEN EDITORS' containing 'README.md' and 'UNTITLED (WORKSPACE)' containing 'lezione1'. The main area displays the 'README.md' file with the following content:

```
① README.md M X
desktop-tutorial > ① README.md > # Welcome to GitHub Desktop!
1  # Welcome to GitHub Desktop!
2
3 This is your README. READMEs are where you can communicate what your project is and how to use it.
4
5 Write your name on line 6, save it, and then head back to GitHub Desktop.
6 Davide Conficconi
```



The screenshot shows the GitHub Desktop application interface. The top navigation bar includes 'File', 'Edit', 'View', 'Repository', 'Branch', and 'Help'. The repository dropdown shows 'Current repository desktop-tutorial' and the branch dropdown shows 'Current branch test-home'. The 'Changes' tab is selected, showing '1 changed file' named 'README.md'. The commit history for 'README.md' shows the following changes:

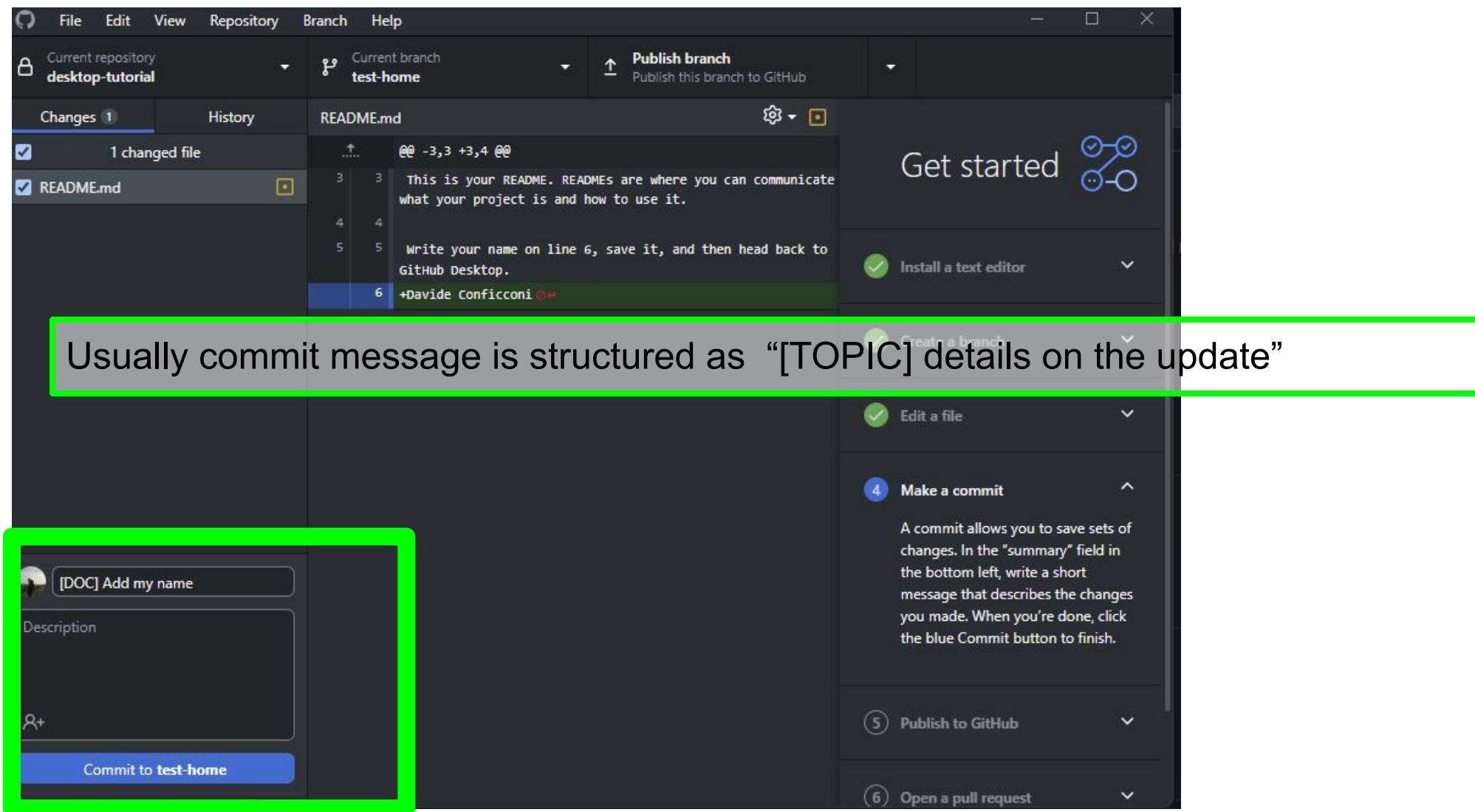
```
Changes 1 History README.md
1 changed file
README.md
@@ -3,3 +3,4 @@
3 This is your README. READMEs are where you can communicate what your project is and how to use it.
4
5 Write your name on line 6, save it, and then head back to GitHub Desktop.
6 +Davide Conficconi
```

To the right, a 'Get started' sidebar provides steps for new users:

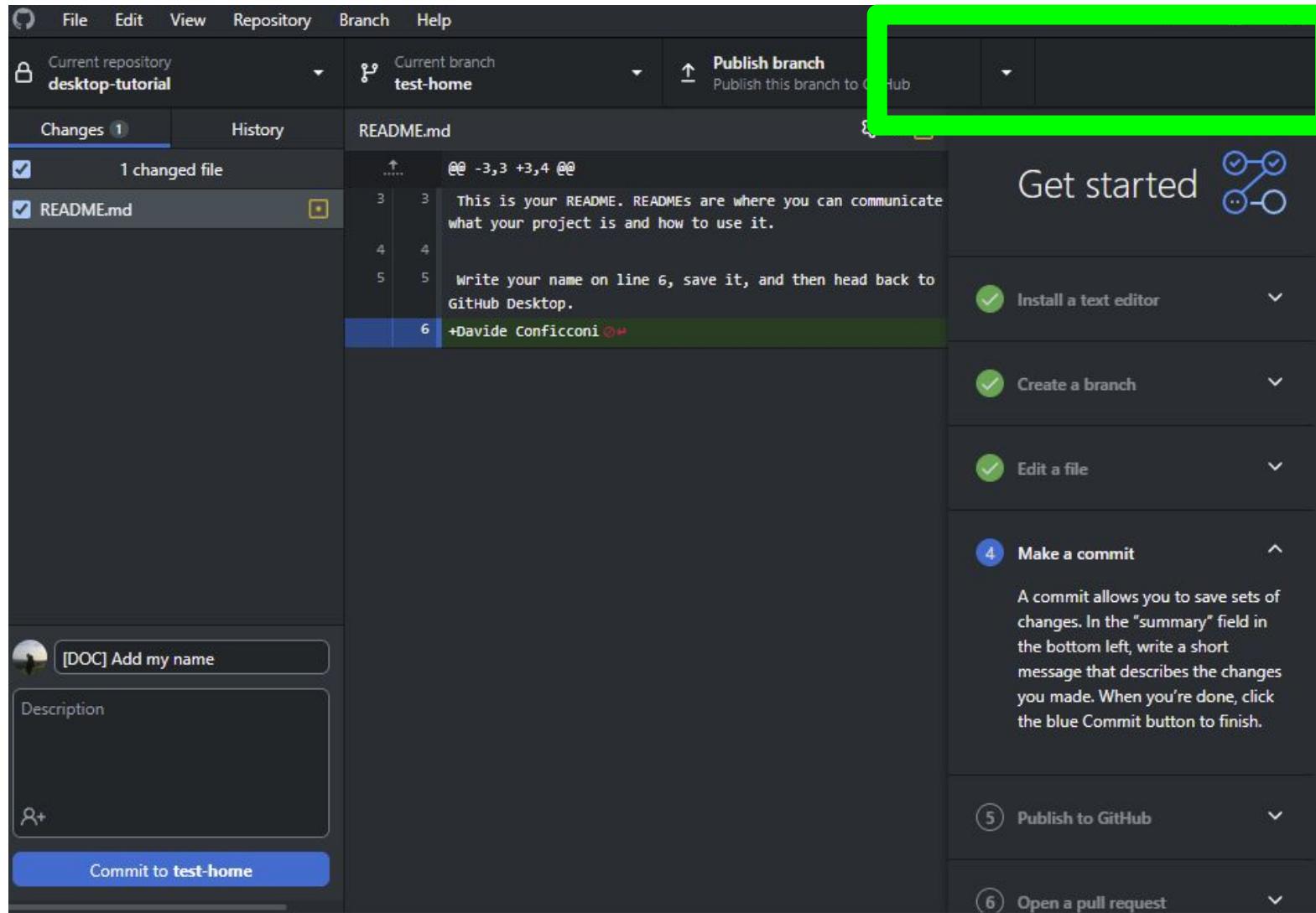
- Install a text editor
- Create a branch
- Edit a file
- Make a commit

A detailed description for 'Make a commit' states: 'A commit allows you to save set the "summary" field in the bottom short message that describes the made. When you're done, click the button to finish.'

GitHub Desktop Commit



GitHub Desktop Push



github.com Sync

The screenshot shows a GitHub repository page for 'desktop-tutorial'. At the top, there's a search bar with placeholder text 'Type / to search' and a '+' button. Below the search bar are navigation links: Actions, Projects, Wiki, Security, Insights, and Settings.

The main header includes the repository name 'desktop-tutorial' (Private), Unwatch (1), Fork (0), and Star (0) buttons. A green 'Compare & pull request' button is also present.

A message通知显示'test-home had recent pushes less than a minute ago'. On the right, there's an 'About' section with the repository description 'GitHub Desktop tutorial repository' and a settings gear icon.

The repository stats show 2 branches and 0 tags. A note indicates that the 'test-home' branch is 1 commit ahead of 'main'. There are two commits from 'DavideConfigconi' with timestamps of 9e82258 1 minute ago and 2 commits 1 minute ago. A 'Contribute' button is also visible.

The 'Code' section has buttons for 'Go to file', 'Add file', and 'Code'. Below this, a 'README.md' file is shown with its content: 'Welcome to GitHub Desktop!'. The README text instructs the user to save their name and head back to GitHub Desktop. A large green rectangular box highlights the 'Publish your first pack' button in the 'Packages' section.

The 'About' section lists Readme, Activity, 0 stars, 1 watching, and 0 forks. The 'Releases' section shows 'No releases published' and a 'Create a new release' link. The 'Packages' section shows 'No packages published' and a 'Publish your first pack' link.





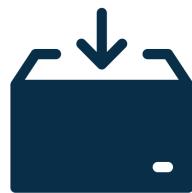
Create, edit, delete file(s)



Create, edit, delete file(s)

Compare the changes

git status
git diff



Create, edit, delete file(s)

Compare the changes

`git status`
`git diff`

Stage the changes for commit

`git add file-name`



Create, edit, delete file(s)



Compare the changes

`git status`
`git diff`



Stage the changes for commit

`git add file-name`



Commit the staged files

`git commit -m "The commit message"`

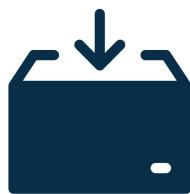


Create, edit, delete file(s)



Compare the changes

`git status`
`git diff`



Stage the changes for commit

`git add file-name`



Commit the staged files

`git commit -m "The commit message"`

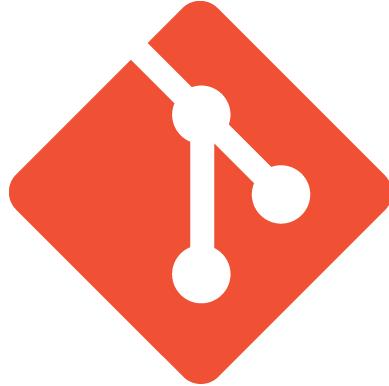


Keep the history in check

`git log`

Ok, but why do we need this?

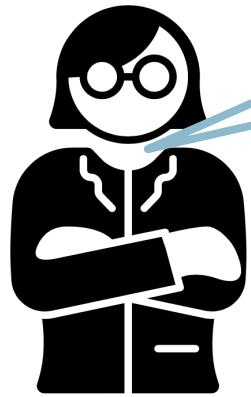




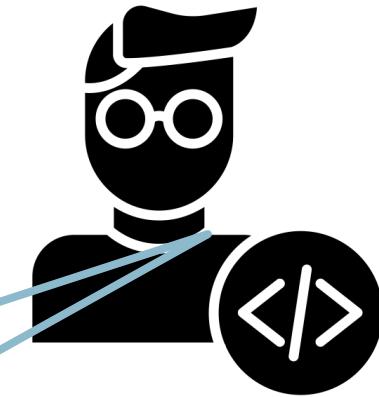
Single developer

- Tracks evolution
- Builds history
- Navigate the history

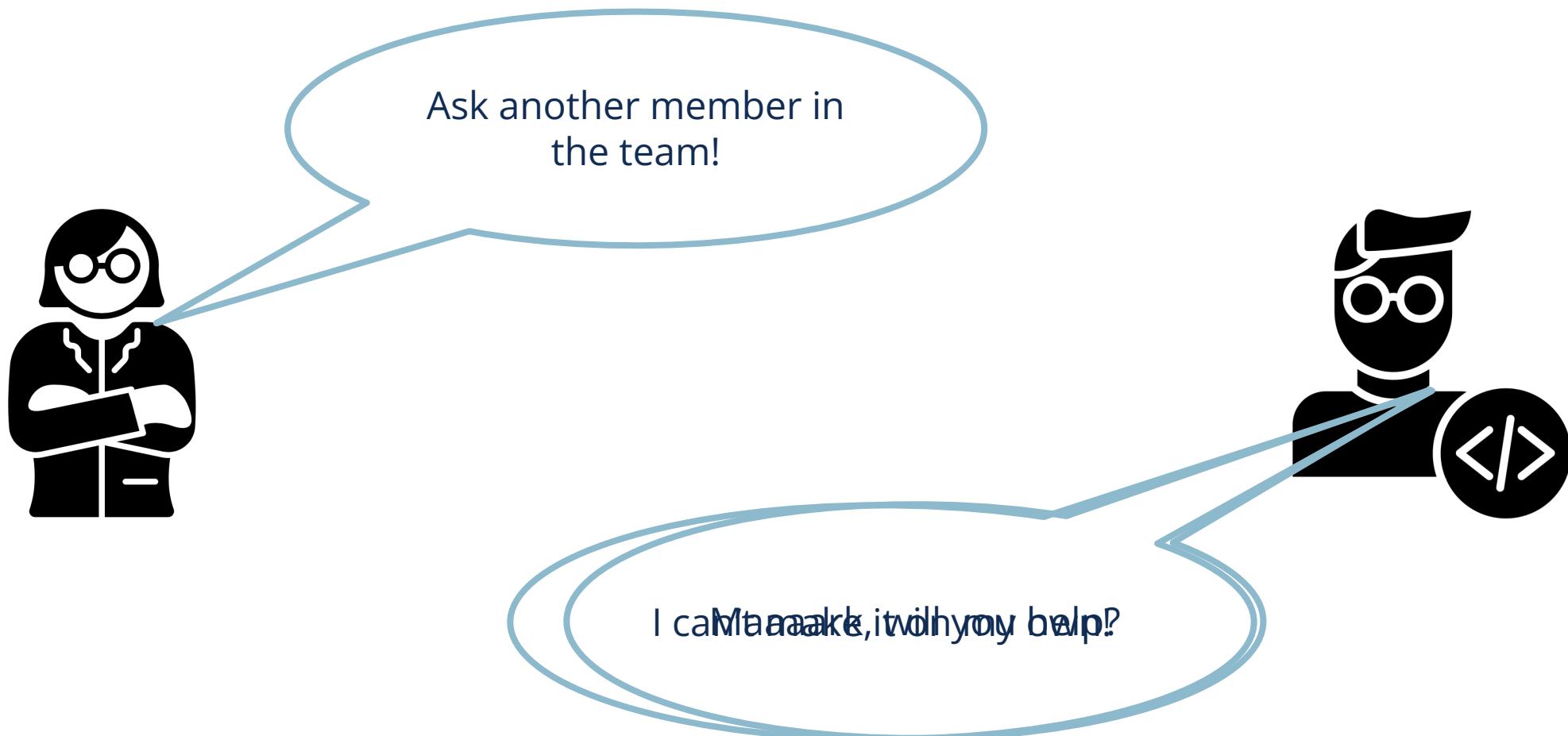
git

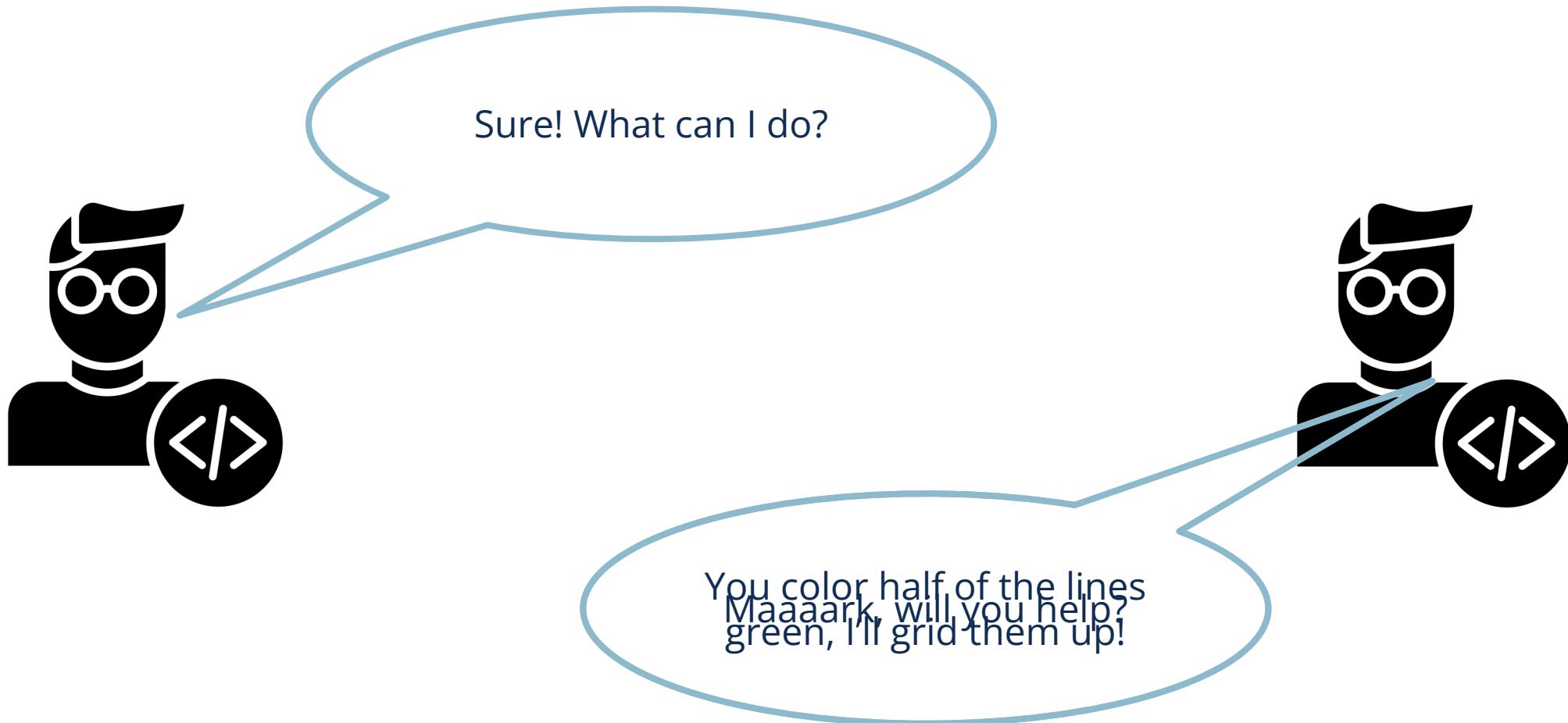


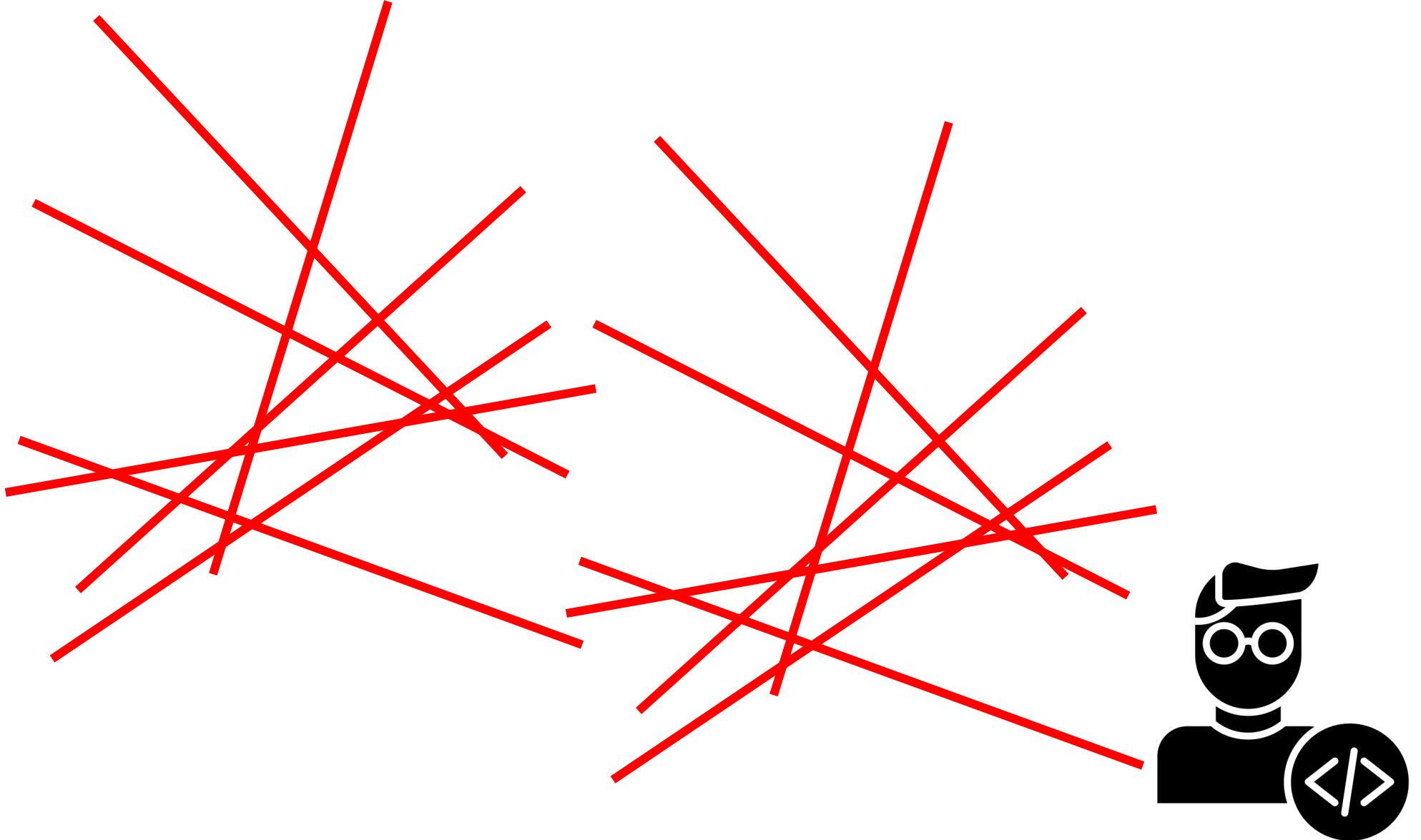
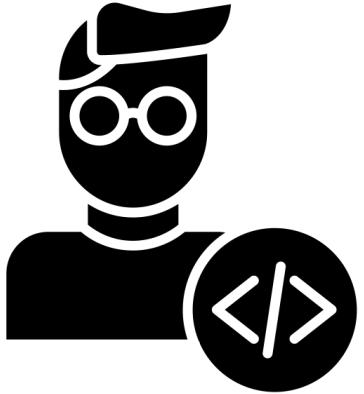
New feat req!
We need all the lines to be perpendicular and half of them
should be green!

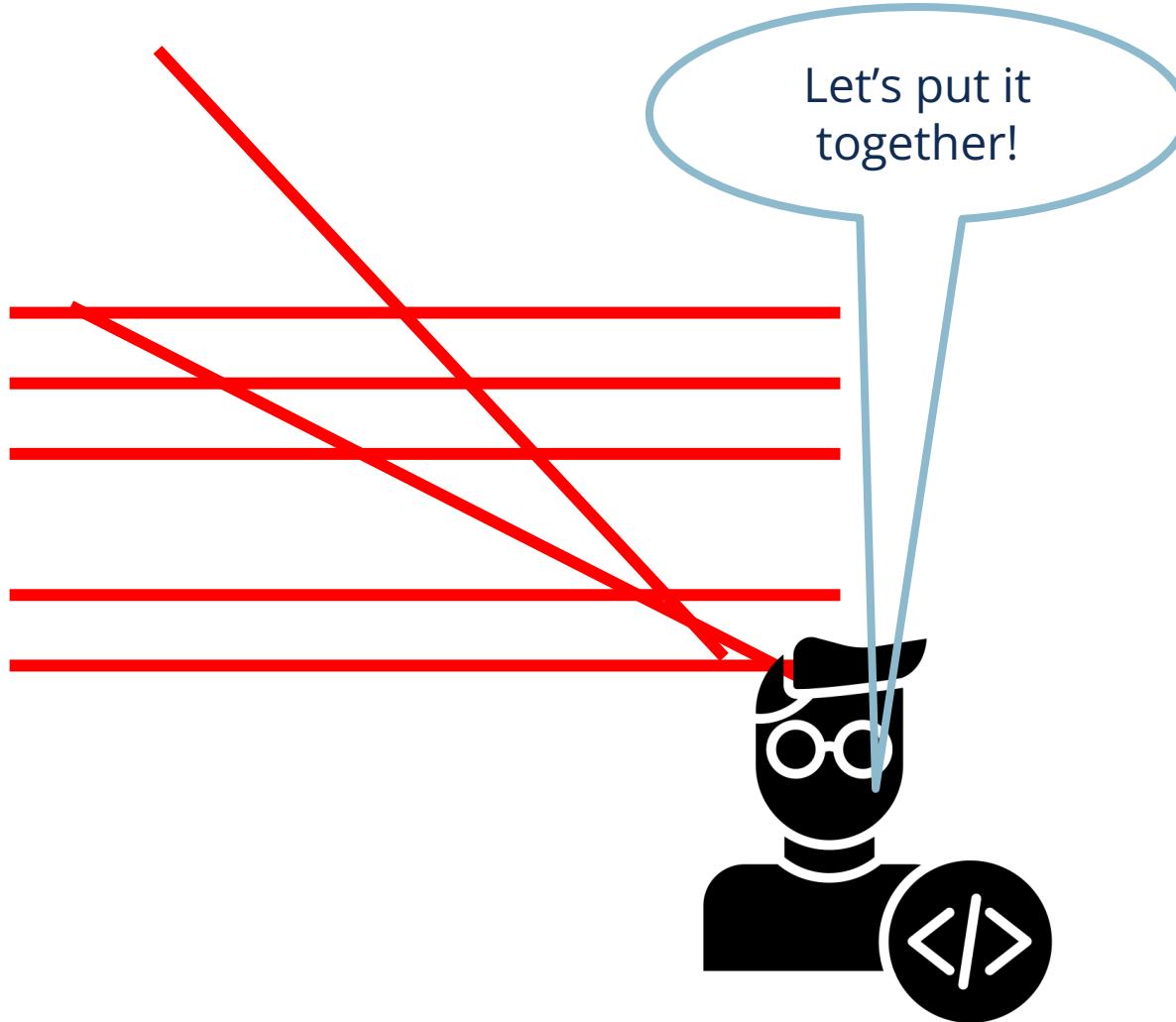
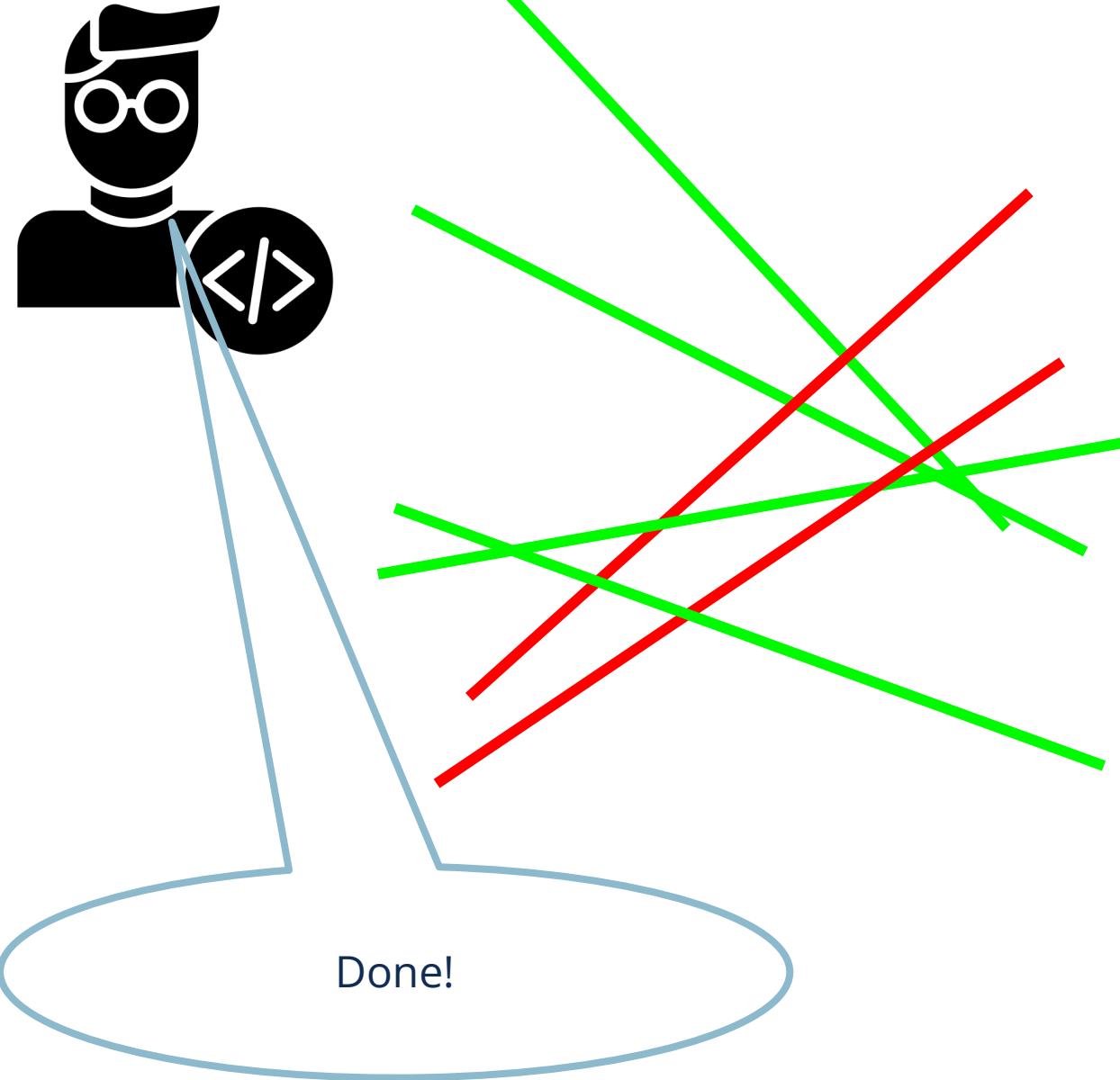


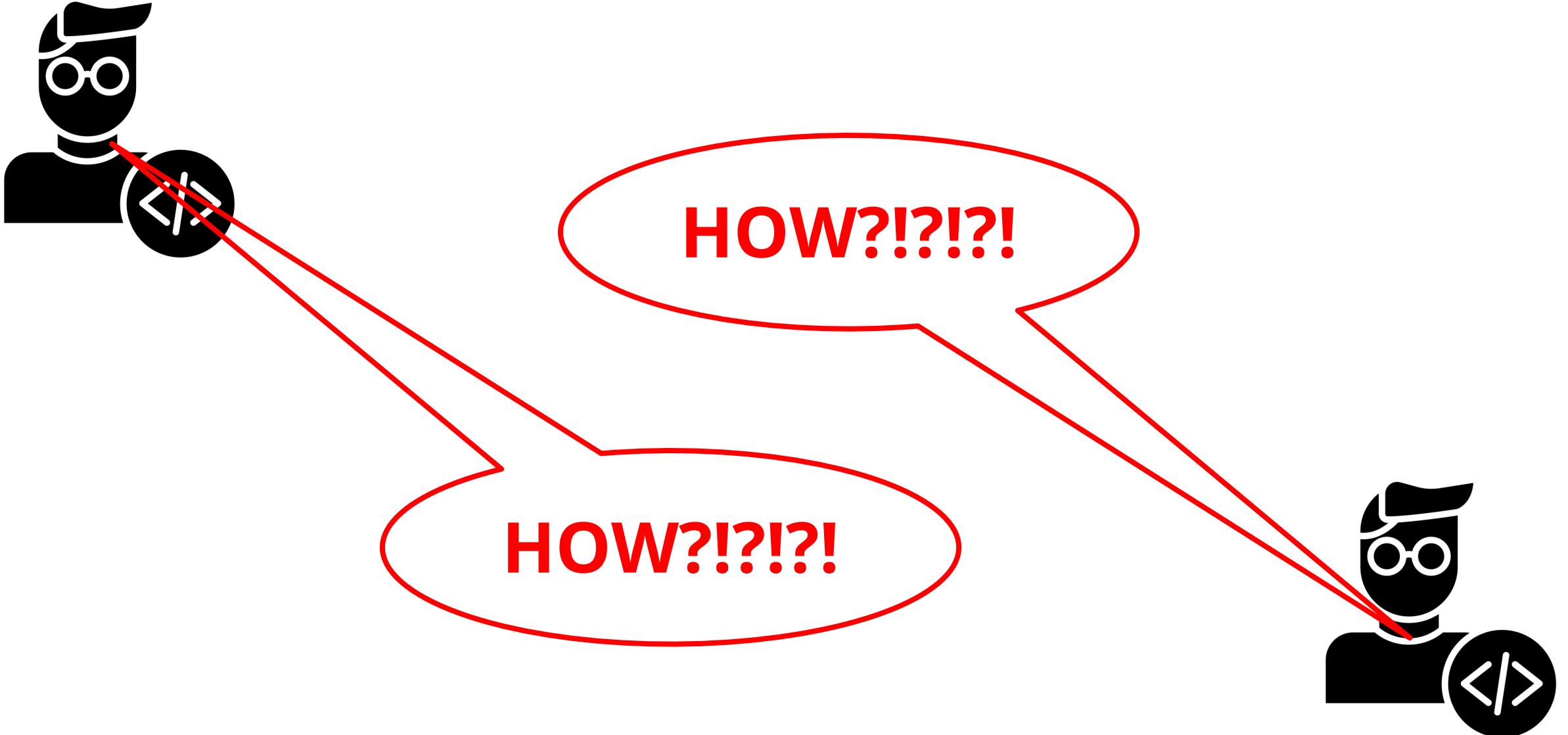
I can't make it on my own!

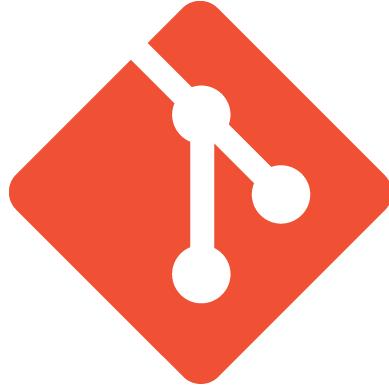












git

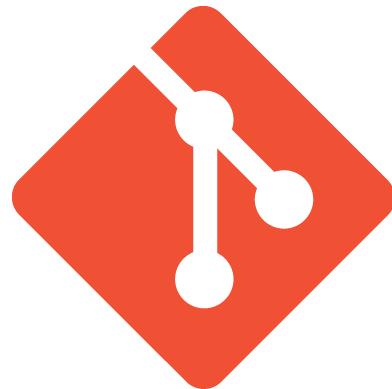
Single developer

- Tracks evolution
- Builds history
- Navigate the history

Team of developers

- Allow concurrent development
- Track the responsible
- Support in merging changes

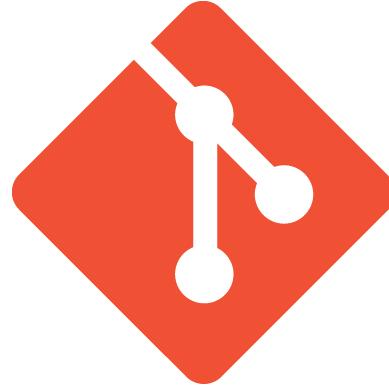
Tool?



Method?

git

<https://git-scm.com/book/en/v2>

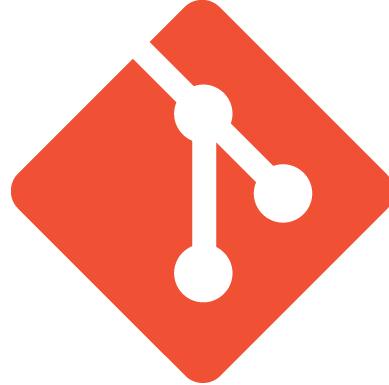


Method

git Flow

<https://nvie.com/posts/a-successful-git-branching-model/>

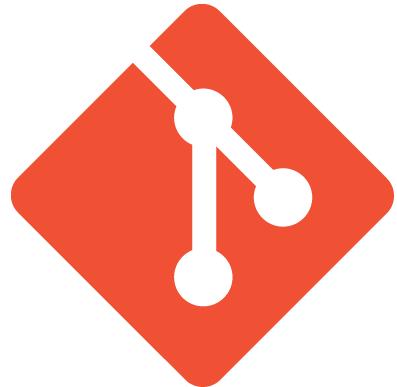
<https://git-scm.com/book/en/v2>



git

- Pull Requests
- Issue tracking
- Documentation
- .gitignore
- submodules
- history rewriting

Other GUIs



git

Free
Win+Linux+macOs

Integrates with all the
major VCS cloud
providers

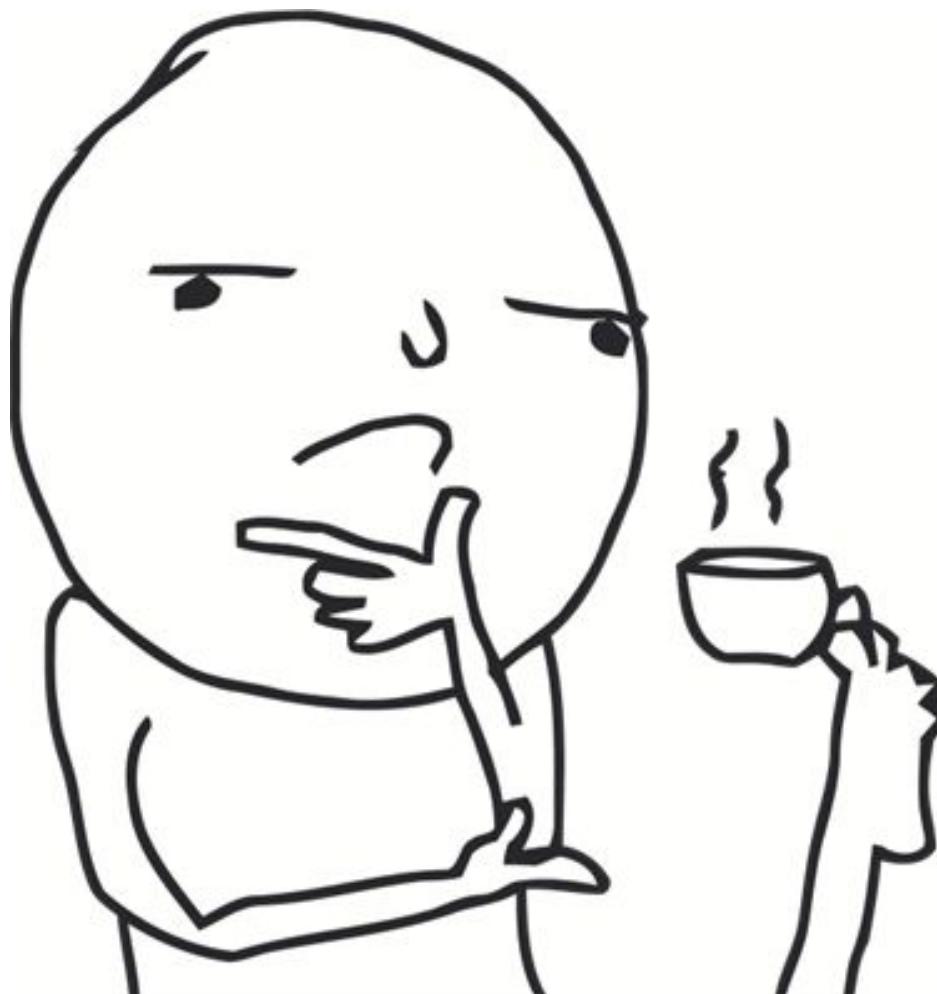
Free
Win+macOs

Direct cloning from
BitBucket

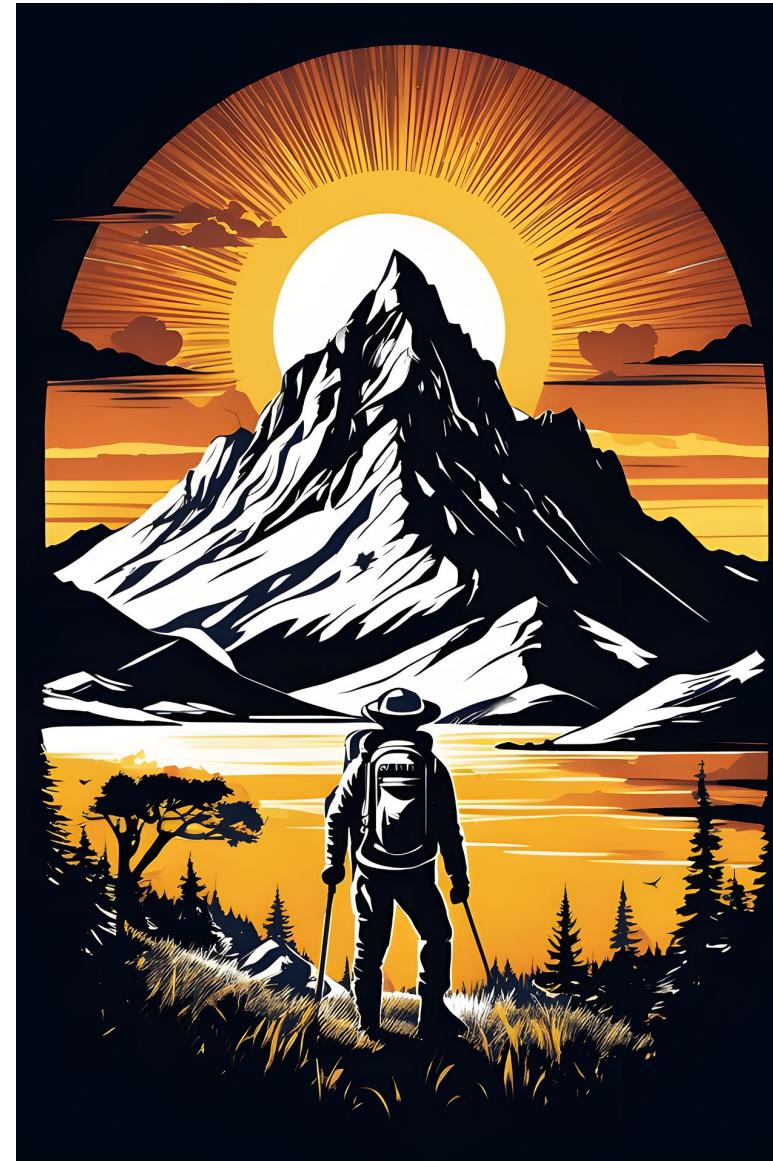
Freemium
(free for GitHub
Student Program
members)
Win+Linux+macOs

Integration with
advanced versioning
features

Cool...but why?



To Find the Saint Graal



To Build our “web portfolio” of LCG

Indice di Informatica Applicata C2

Terremo traccia degli sviluppi tramite questo indice.

- [Lezione 1](#)
- Lezione 2
- Lezione 3
- Lezione 4
- Lezione 5
- Lezione 6

To Build our “web portfolio” of LCG

Indice di Informatica Applicata C2

Terremo traccia degli sviluppi tramite questo indice.

- [Lezione 1](#)
- Lezione 2
- Lezione 3
- Lezione 4
- Lezione 5
- Lezione 6

HOW?

To Build our “web portfolio” of LCG

Indice di Informatica Applicata C2

Terremo traccia degli sviluppi tramite questo indice.

- [Lezione 1](#)
- Lezione 2
- Lezione 3
- Lezione 4
- Lezione 5
- Lezione 6



<https://pages.github.com/>

<https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site#next-steps>

GitHub Pages: Basic workflow



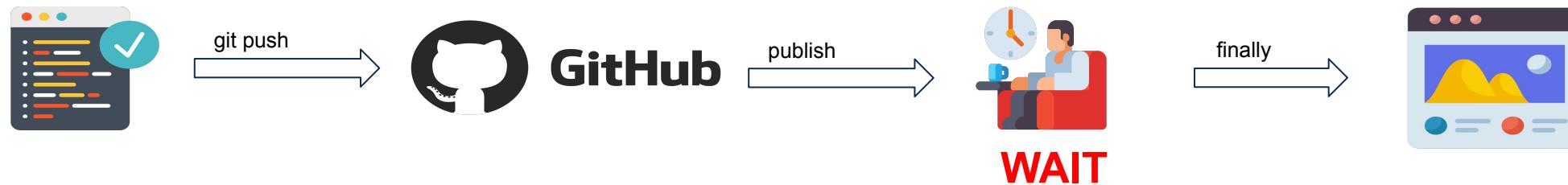
GitHub Pages: Basic workflow



That easy!?

GitHub Pages: Limitations

- You have to **wait** (2/3 minutes)



GitHub Pages: Limitations

- You have to **wait** (2/3 minutes)



- Supports only plain **HTML, CSS, JavaScript**

GitHub Pages: Limitations

- You have to **wait** (2/3 minutes)



- Supports only plain **HTML, CSS, JavaScript**

and Jekyll (which is a static site generator with built-in support for GitHub Pages.)



GitHub Pages: Limitations

- You have to **wait** (2/3 minutes)



- Supports only plain **HTML, CSS, JavaScript**

and Jekyll (which is a static site generator with built-in support for GitHub Pages.)



- No free personal domain → all sites hosted at:
`user-name.github.io/git-repo-name` (please check [the usage limitations](#))

GitHub Pages: Limitations

- You have to **wait** (2/3 minutes)



- Supports only plain **HTML, CSS, JavaScript**

and Jekyll (which is a static site generator with built-in support for GitHub Pages.)



- No free personal domain → all sites hosted at:
`user-name.github.io/git-repo-name` (please check [the usage limitations](#))
- Size limitation: **1GB**

GitHub Pages: Limitations

- You have to **wait** (2/3 minutes)



- Supports only plain **HTML, CSS, JavaScript**

and Jekyll (which is a static site generator with built-in support for GitHub Pages.)



- No free personal domain → all sites hosted at:
`user-name.github.io/git-repo-name` (please check [the usage limitations](#))
- Size limitation: **1GB**
- Publish limitations: **10 “publish”/hour**

Let's Try this!!!

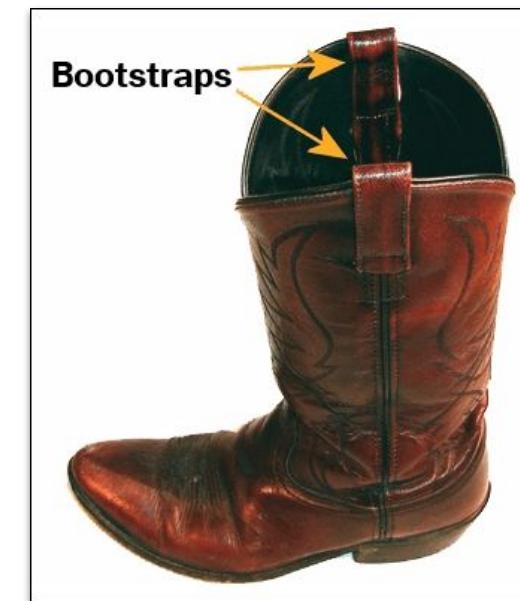


Let's publish our first website!

For this first example we will use a Bootstrap template

Bootstrap is a framework for fast and dynamic websites

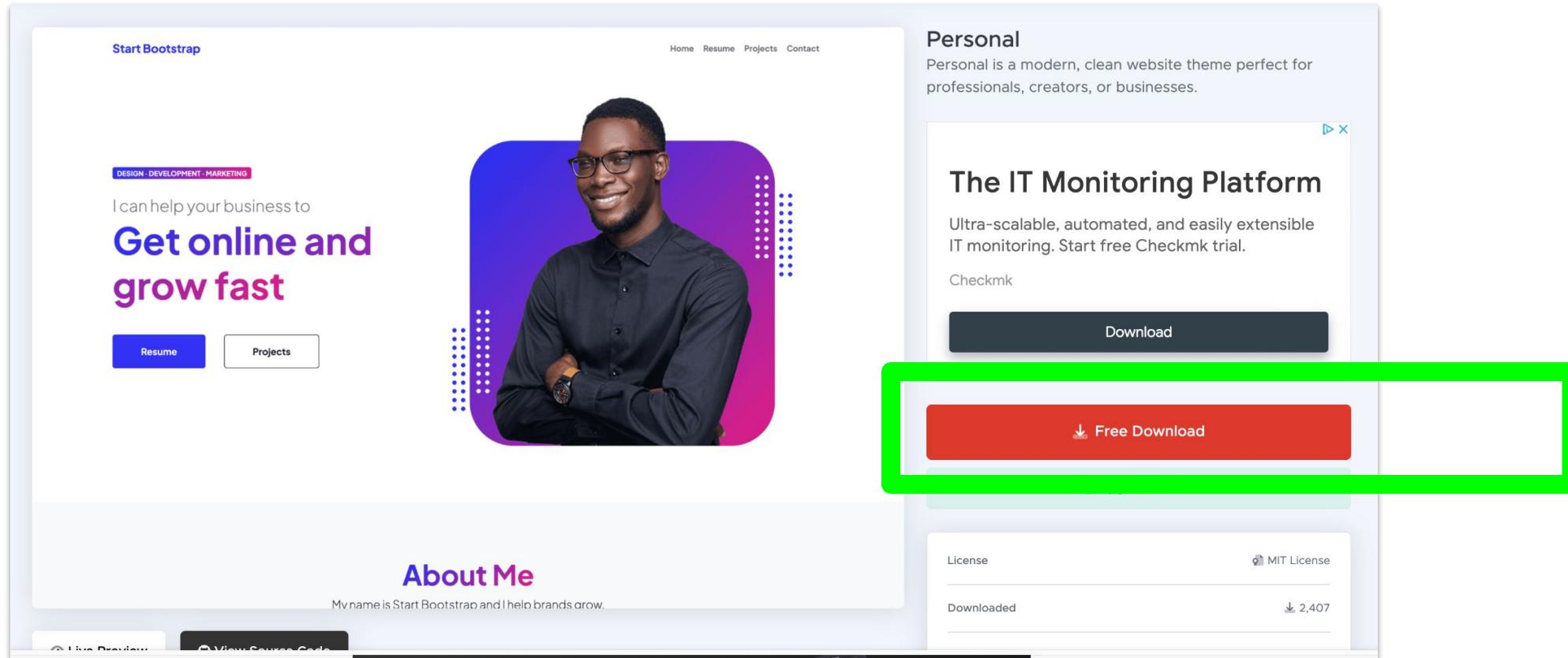
- faster development cycle
- LOT of templates to start from
- easy to use



PICK YOUR FAVOURITE: <https://startbootstrap.com/themes>

Download your template

Example template: <https://startbootstrap.com/theme/personal>



Folder structure

Example template: <https://startbootstrap.com/theme/personal>



assets: images/videos etc

css: well, CSS

js: well, JavaScript aka JS

*.html: HTML :D

DEMO time: Publishing steps

PICK YOUR FAVOURITE: <https://startbootstrap.com/themes>

- 1) Create a GitHub repository (or use the already created one)
- 2) Open the repository in VSCode
- 3) Unzip the downloaded template
- 4) Copy the content of the folder in VSCode
- 5) Create a commit with “[init] basic template upload”
- 6) Push the commit to the repository
- 7) Go to YourUserName.github.io/YourRepositoryName
(ex. ian-ofgod.github.io/test-github-pages)

DEMO time: Publishing steps



Enabling GitHub pages

GitHub does not publish automatically all GitHub repositories as websites

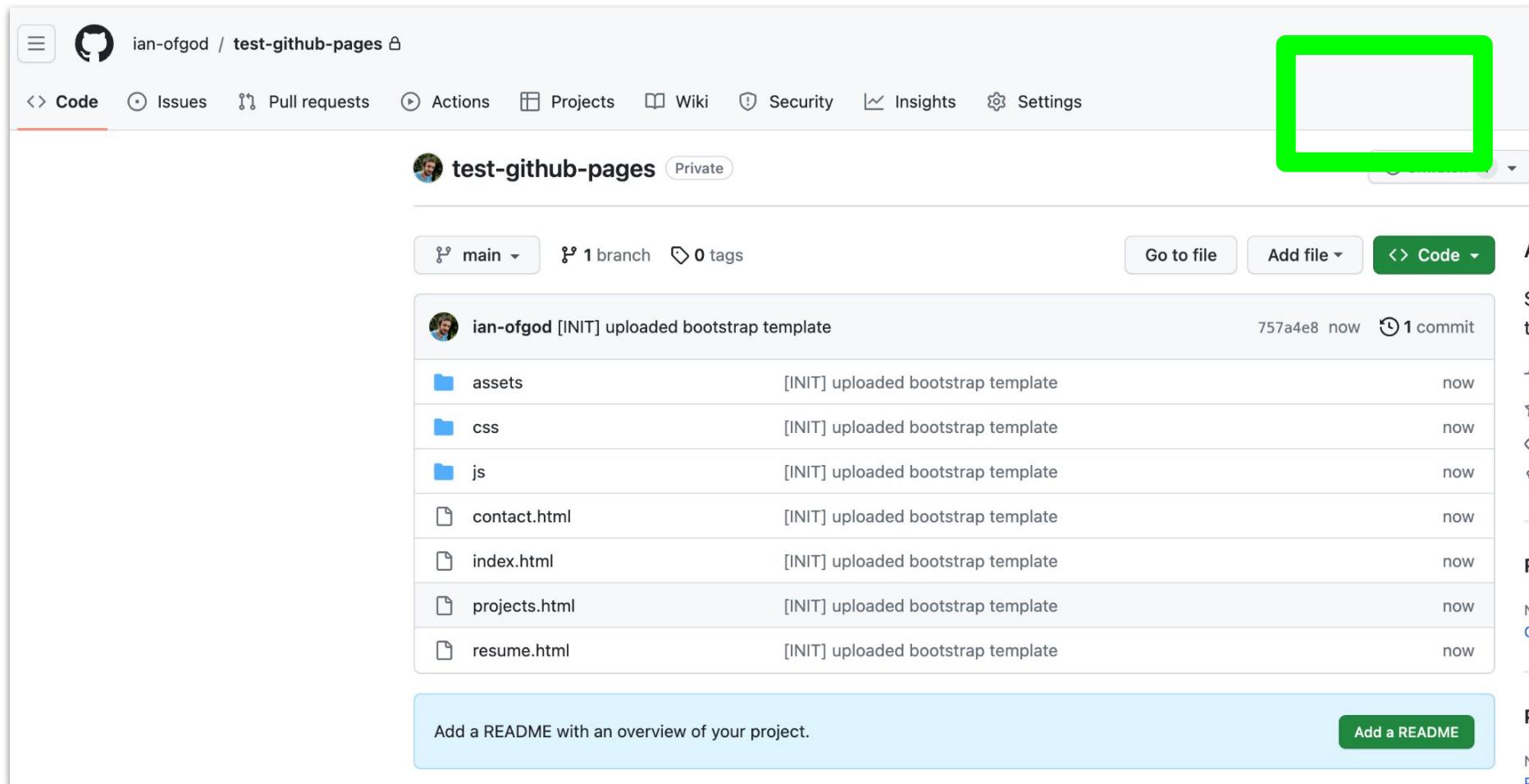
- Not all repositories are websites (software, books, plain images)
- YOU have to ask!

HOW?

Demo Time :)



Enabling GitHub pages



The screenshot shows a GitHub repository named "test-github-pages". The repository is private, as indicated by the "Private" badge. It has one branch, "main", and no tags. The commit history shows a single commit from "ian-ofgod" titled "[INIT] uploaded bootstrap template" at commit hash 757a4e8, which occurred "now". The commit includes files: assets, css, js, contact.html, index.html, projects.html, and resume.html. All files were uploaded using the "[INIT] uploaded bootstrap template" message. Below the commit history, there is a call-to-action to "Add a README" with a button labeled "Add a README". A large green rectangular box highlights the "Code" tab in the top navigation bar.

Enabling GitHub pages

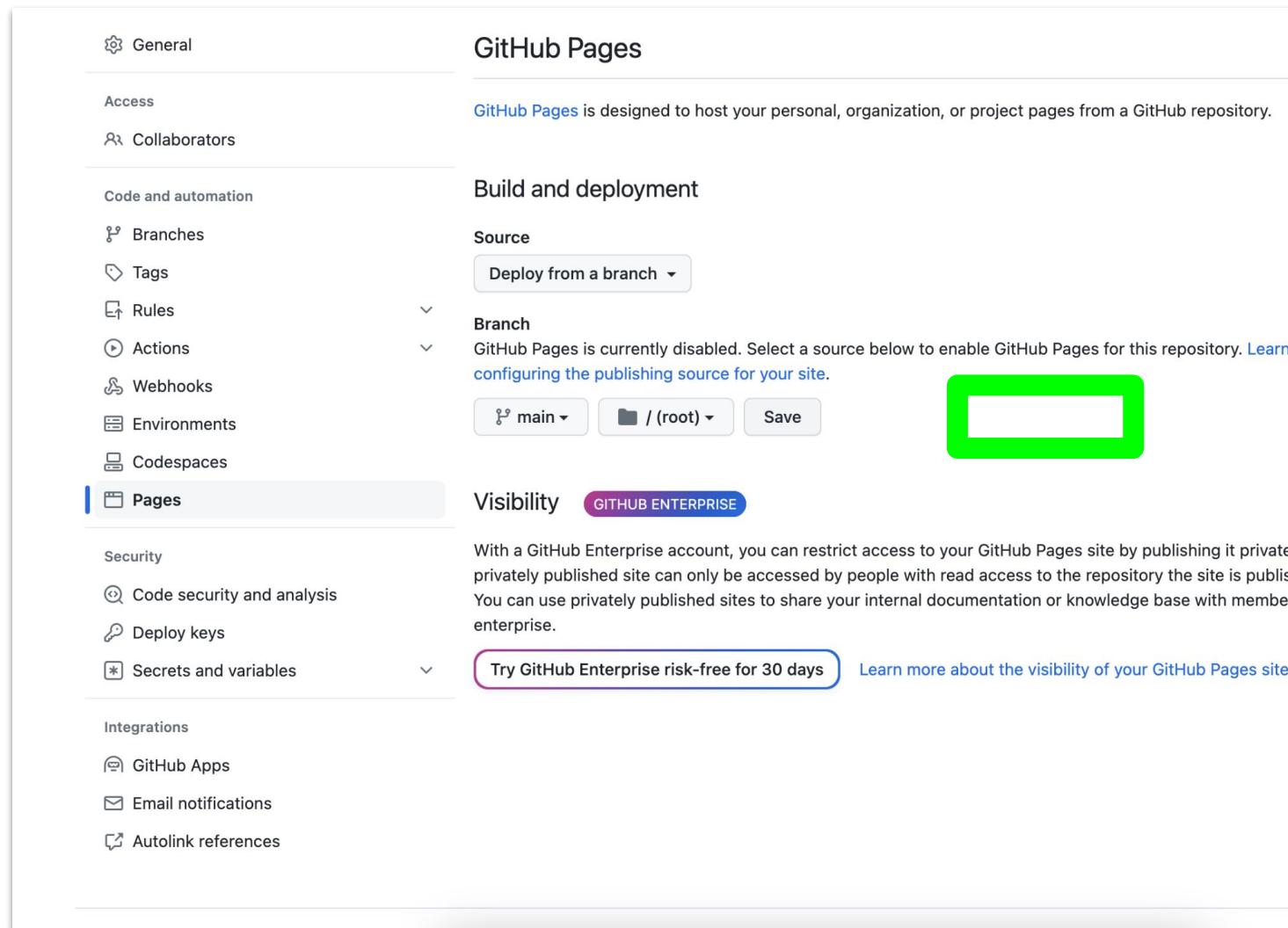
The screenshot shows the GitHub repository settings for 'ian-ofgod / test-github-pages'. The 'General' tab is selected. On the left, there's a sidebar with various settings sections like 'Access', 'Collaborators', 'Code and automation' (which contains 'Branches', 'Tags', 'Rules', 'Actions', 'Webhooks', 'Environments', and 'Pages'), 'Security', 'Code security and analysis', 'Deploy keys', 'Secrets and variables', 'Integrations', 'GitHub Apps', 'Email notifications', and 'Autolink references'. The 'Pages' section is highlighted with a green box. In the main area, under 'General', there's a 'Repository name' field set to 'test-github-pages' with a 'Rename' button. Below it are two checkboxes: 'Template repository' (unchecked) and 'Require contributors to sign off on web-based commits' (unchecked). Under 'Default branch', it says 'The default branch is considered the "base" branch in your repository, against which all pull requests are automatically merged, unless you specify a different branch.' with 'main' selected. In the 'Features' section, 'Wikis' and 'Issues' are checked. There's also a 'Restrict editing to collaborators only' option.

Enabling GitHub pages

The screenshot shows the GitHub repository settings page for enabling GitHub Pages. The left sidebar lists various settings categories: General, Access, Collaborators, Code and automation (with sub-options like Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, and Pages), Security, Code security and analysis, Deploy keys, Secrets and variables, Integrations (GitHub Apps, Email notifications, Autolink references), and Pages. The Pages option is currently selected, indicated by a blue bar at the top of the sidebar.

The main content area is titled "GitHub Pages" and contains the following text: "GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository." Below this, under "Build and deployment", there is a "Source" section with a dropdown menu set to "Deploy from a branch". A "Branch" section indicates that GitHub Pages is currently disabled and provides a link to learn more about configuring the publishing source. A "Select branch" dropdown menu is open, showing options: "main" (selected) and "None". A green rectangular highlight surrounds the "main" option in the dropdown menu. At the bottom of the page, there are two buttons: "Try GitHub Enterprise risk-free for 30 days" and "Learn more about the visibility of your GitHub Pages site".

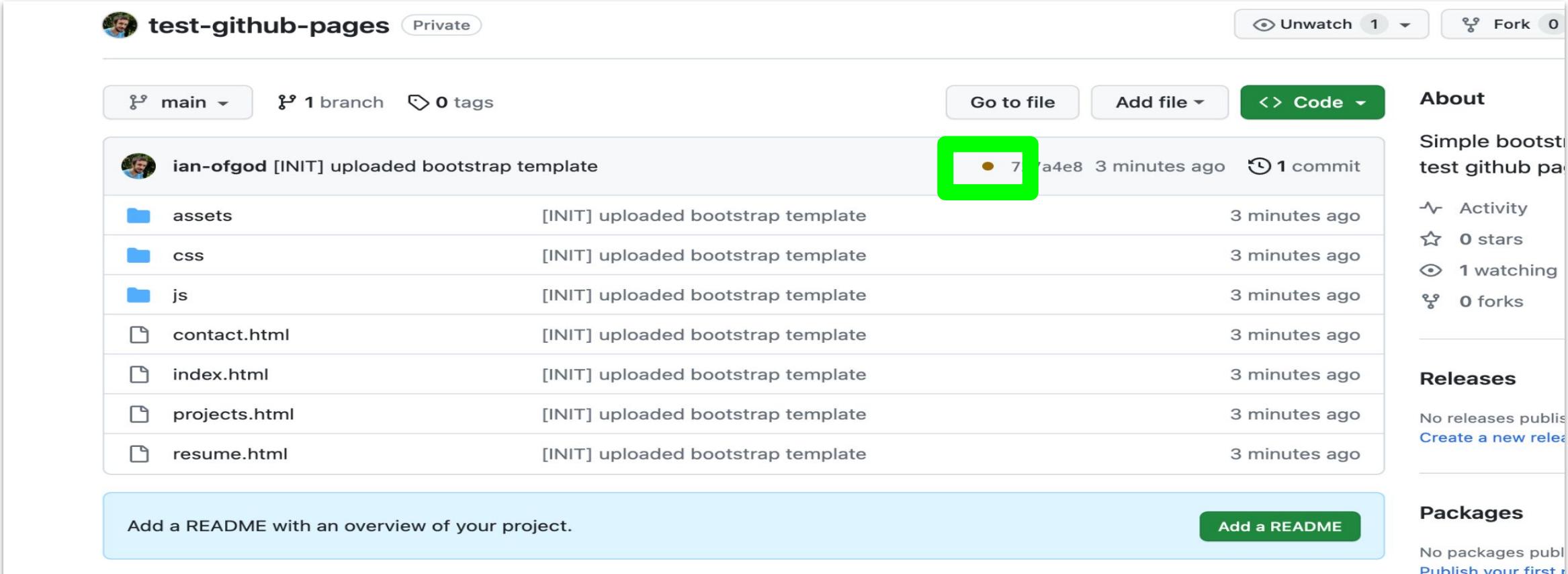
Enabling GitHub pages



The screenshot shows the GitHub Pages settings interface. On the left, there's a sidebar with various repository management options like General, Access, Collaborators, and Code and automation. Under Code and automation, the 'Pages' option is selected and highlighted with a blue bar. The main content area is titled 'GitHub Pages' and contains a brief description: 'GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository.' Below this, there's a section titled 'Build and deployment' with a 'Source' dropdown set to 'Deploy from a branch'. A 'Branch' dropdown is shown with 'main' and '/(root)' options, both of which are highlighted with a green box. There's also a 'Save' button. At the bottom, there's a 'Visibility' section with a 'GITHUB ENTERPRISE' badge. It explains that with a GitHub Enterprise account, you can restrict access to your GitHub Pages site by publishing it privately. It also mentions that a privately published site can only be accessed by people with read access to the repository the site is published in. It encourages users to try GitHub Enterprise risk-free for 30 days.

Check build status

You can check the status of the “build” aka the publishing process



The screenshot shows a GitHub repository named "test-github-pages". The repository is private, has 1 branch, and 0 tags. The "Code" button is highlighted in green. The commit history shows a single commit from "ian-ofgod" uploaded 3 minutes ago, which includes files like assets, css, js, contact.html, index.html, projects.html, and resume.html. A green box highlights the first commit. On the right side, there are sections for "About", "Activity", "Releases", and "Packages".

About
Simple bootstrap template
test github pages

Activity
0 stars
1 watching
0 forks

Releases
No releases published
Create a new release

Packages
No packages published
Publish your first package

Commit History

File	Message	Time
assets	[INIT] uploaded bootstrap template	3 minutes ago
css	[INIT] uploaded bootstrap template	3 minutes ago
js	[INIT] uploaded bootstrap template	3 minutes ago
contact.html	[INIT] uploaded bootstrap template	3 minutes ago
index.html	[INIT] uploaded bootstrap template	3 minutes ago
projects.html	[INIT] uploaded bootstrap template	3 minutes ago
resume.html	[INIT] uploaded bootstrap template	3 minutes ago

Add a README with an overview of your project. [Add a README](#)

Check build status

You can check the status of the “build” aka the publishing process

The screenshot shows a GitHub repository interface. On the left, there's a list of files and folders: main (branch), 1 branch, 0 tags. The files listed are assets, css, js, contact.html, index.html, projects.html, and resume.html, all showing '[INIT] uploaded bootstrap template' status. Below this is a button to 'Add a README'. At the top right are buttons for 'Go to file', 'Add file', 'Code', and 'About'. A modal window titled 'Simple bootstrap's template upload' is open, showing 'Some checks haven't completed yet' with 1 successful and 2 queued checks. It lists three items: 'pages build and deployment / build (dynamic)' (successful), 'pages build and deployment / report-build-status' (queued), and 'pages build and deployment / deploy (dynamic)' (queued). To the right of the modal are sections for 'Releases' (no releases) and 'Packages' (no packages). A green box highlights the 'About' button at the top right.

ian-ofgod [INIT] uploaded bootstrap template

assets [INIT] uploaded bootstrap template

css [INIT] uploaded bootstrap template

js [INIT] uploaded bootstrap template

contact.html [INIT] uploaded bootstrap template

index.html [INIT] uploaded bootstrap template

projects.html [INIT] uploaded bootstrap template

resume.html [INIT] uploaded bootstrap template

Add a README

Some checks haven't completed yet

1 successful and 2 queued checks

- ✓ pages build and deployment / build (dynamic) Suc... Details
- pages build and deployment / report-build-status (...) Details
- pages build and deployment / deploy (dynamic) Q... Details

Releases

No releases published

Create a new release

Packages

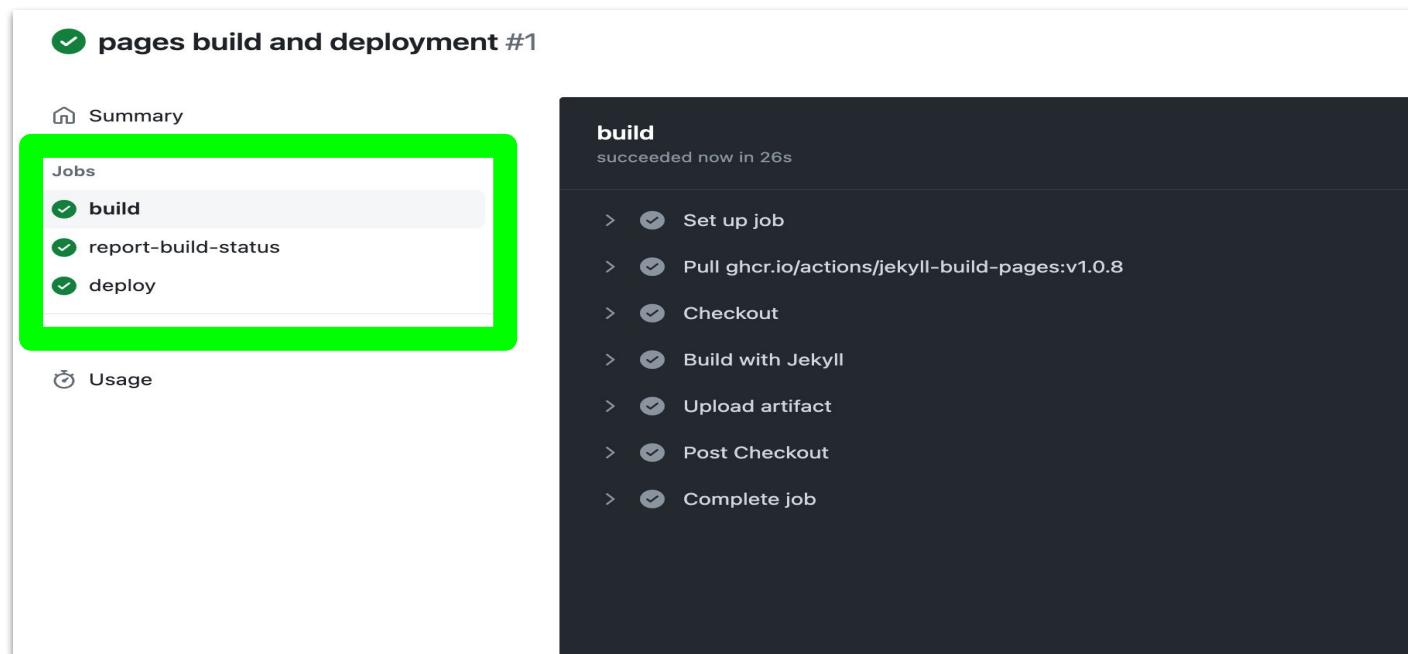
No packages published

Publish your first package

Check build status

Once all the steps are done, the site is up and running.

It might happen that your browser is still displaying the old version, in that case I suggest you to open it in an incognito window (typically Ctrl+Alt+N or Cmd+alt+N)



Enabling GitHub pages

Go to <https://user-name.github.io/repo-name>



The screenshot shows a website template for GitHub Pages. At the top left is a "Start Bootstrap" logo. At the top right are navigation links: Home, Resume, Projects, and Contact. Below the header, there's a large image of a smiling man with glasses and a beard. To the left of the image, there's a section with a "DESIGN · DEVELOPMENT · MARKETING" badge, followed by the text "I can help your business to" and a large, bold, blue and purple text block that reads "Get online and grow fast".

Enabling GitHub pages



Key takeaways

- 1) GitHub pages: Use only to check the final results
- 2) During development use the preview plugin of VSCode
- 3) Bootstrap is an easy way to obtain nice-looking websites

Further readings

- 1) [Github pages Docs](#)
- 2) <https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site#next-steps>
- 3) [Bootstrap reference](#)
- 4) [From markdown to github pages](#)

Informatica Self-Assesment

1. ~~Informatica Self Assesment~~



2. Local dev with VSCode, Indice html for IA2526

3. Create our Webpage with Githubpages



4. Draw with P5, P5 Live Editor



<https://editor.p5js.org/>

What is P5?

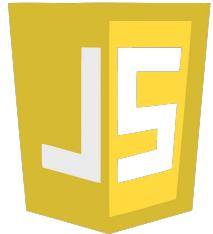
<https://hello.p5js.org/>

p5*js

What is P5?

<https://hello.p5js.org/>

JavaScript

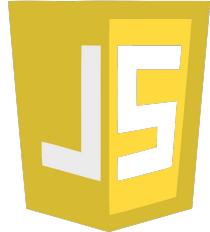


p5*js

What is P5?

<https://hello.p5js.org/>

JavaScript



p5.js

p5.js is a friendly tool for
learning to code and make art.

Why p5.js?

p5*JS

Make **programming** interactive graphics **easy**

Why p5.js?

p5.js



Make **programming** interactive graphics **easy**

Why p5.js?

p5.js



Make **programming** interactive graphics **easy**

Immediate feedback with few lines of code

Why p5.js?

p5.js



Make **programming** interactive graphics **easy**

Immediate feedback with few lines of code

It is a **free** and **open-source** JavaScript library built by an inclusive, nurturing **community**. p5.js welcomes artists, designers, beginners, educators, and anyone else!

The screenshot shows the GitHub homepage for the p5.js repository. At the top, there are navigation links for 'README', 'Code of conduct', and 'LGPL-2.1 license'. Below the header, there's a row of badges: 'npm package 1.10.0', 'all contributors 714', and 'downloads 1.3M'. The main title 'p5.js' is displayed in a large, blue, sans-serif font. Below it, a 'Welcome!' message is followed by a series of hand icon emojis. A descriptive paragraph at the bottom explains what p5.js is: 'p5.js is a free and open-source JavaScript library for accessible creative coding. It is a nurturing community, an approachable language, an exploratory tool, an accessible environment, an inclusive platform, welcoming and playful for artists, designers, educators, beginners, and anyone else!'

<https://p5js.org/community/>



<https://github.com/processing/p5.js>

Languages

Make **programming** interactive graphics **easy**

The screenshot shows the top navigation bar of the p5.js Reference page. It includes language selection (English), accessibility options, and a search bar. Below the header, there's a section titled "Reference" with a sub-section "Find easy explanations for every piece of p5.js code." A "Filter by keyword" input field is also present. To the right, three blue hyperlinks are displayed: <https://p5js.org/reference/>, <https://p5js.org/examples/>, and <https://p5js.org/tutorials/>.

Reference

Find easy explanations for every piece of p5.js code.

Filter by keyword

<https://p5js.org/reference/>

<https://p5js.org/examples/>

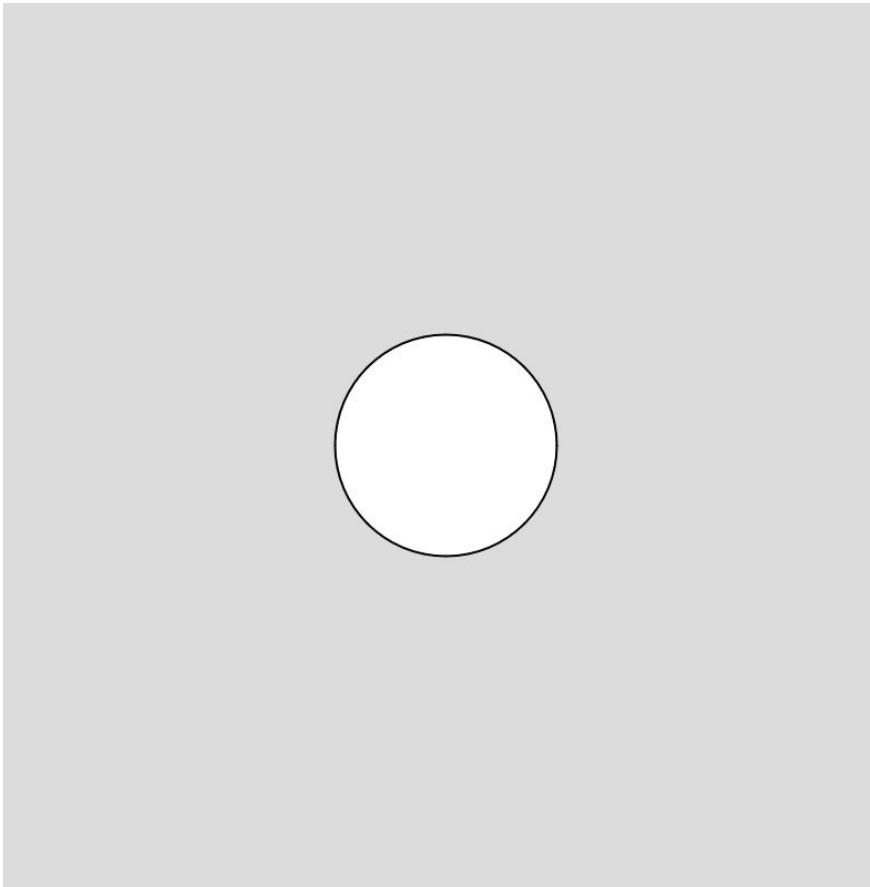
<https://p5js.org/tutorials/>

setup() is called and runs one time. It can be used to set default values for your project.

```
function setup() {  
    createCanvas(400, 400);  
}  
  
function draw() {  
    background(220);  
}
```

draw() is called directly after setup() and executes the lines of code inside its curly brackets 60 times per second until the program is stopped or the noLoop() function is called.

Can We Draw a Circle with P5.js?





<https://editor.p5js.org/>

The [p5.js Web Editor](#) is a website where programmers can write, test, share, or remix p5.js programs without needing to download or configure a *code editor* on a computer.

A screenshot of the p5.js Web Editor. At the top, there's a navigation bar with "File", "Edit", "Sketch", and "Help" menus, followed by language selection ("English"), and "Log in" and "Sign up" buttons. A red arrow points to the "Sign up" button. Below the menu is a toolbar with a play button, a square button, and checkboxes for "Auto-refresh" and "Onyx atom". The main area shows a code editor with a file named "sketch.js" containing the following code:

```
1 function setup() {
2   createCanvas(400, 400);
3 }
4
5 function draw() {
6   background(220);
7 }
```

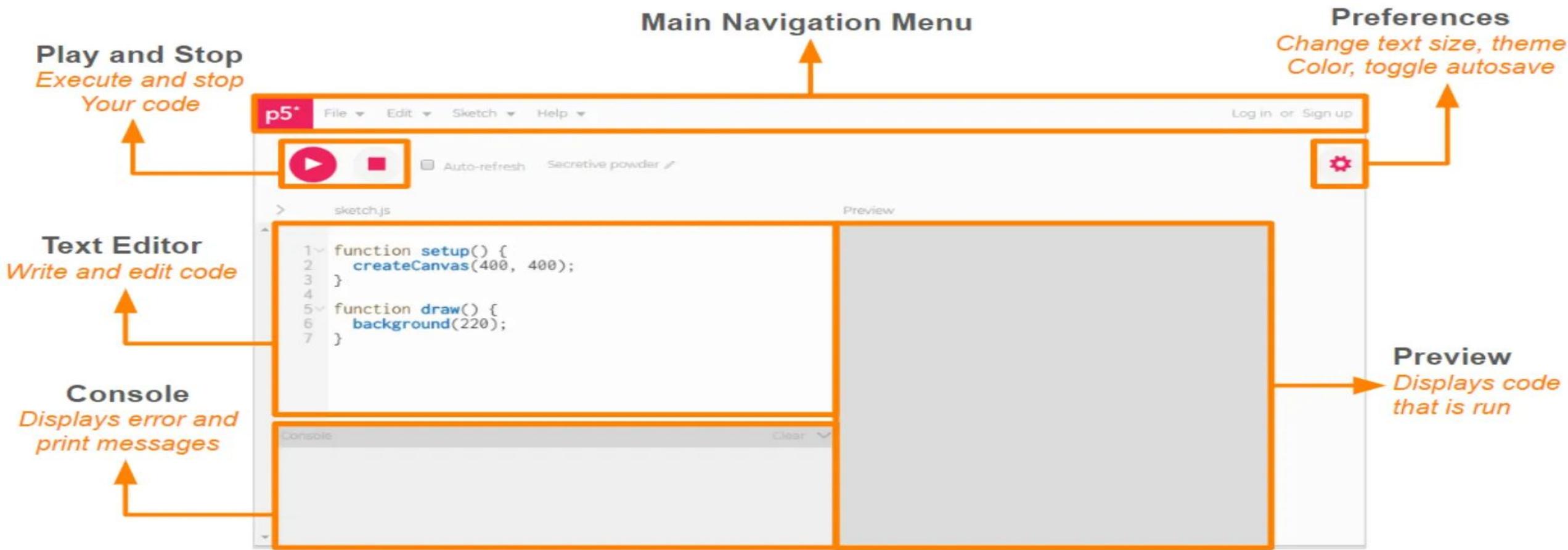
Next to the code editor is a "Preview" window which is currently blank. On the far right, there's a gear icon.

Let's try to draw a circle with it!

p5*



p5.js Web Editor Interface



p5.js Drawing a Circle

p5*

p5* File ▾ Edit ▾ Sketch ▾ Help ▾

Auto-refresh Seed crawdad by DavideConficconi

> sketch.js Saved: 17 minutes ago Preview

```
1 function setup() {
2   createCanvas(400, 400);
3 }
4
5 function draw() {
6   background(220);
7   //A circle is a round shape defined by the x, y, and d parameter
8   circle(200,200,100);
9 }
```

Console Clear ▾



>

Informatica Self-Assesment

1. ~~Informatica Self Assesment~~



2. Local dev with VSCode, Indice html for IA2526

3. Create our Webpage with Githubpages



4. Draw with P5, P5 Live Editor



<https://editor.p5js.org/>

Additional Challenges of IA Module :)

Additional challenges that might grant you extra points to the IA part will be proposed

One of the best will be presented to colleagues the following IA lecture and we will briefly discuss them

THESE CHALLENGES ARE NOT MANDATORY BUT WARMLY RECOMMENDED

to keep the pace and be prepared for the official assignments and group project

Lecture 1 Challenge

Prepare a better index visualization than the one with plain html I presented.

You might use your CSS knowledge or bootstrap templates

Deadline: Monday 30th September 12:00

Start Bootstrap

Home Resume Projects Contact

DESIGN · DEVELOPMENT · MARKETING

I can help your business to

**Get online and
grow fast**

... INEL laboratory MILANO 1863 - INFORMAZIONE E DIFESA CREDITIZIO



Thanks for your attention

Davide Conficconi <davide.conficconi@polimi.it>
Alessandro Nazzari <alessandro.nazzari@polimi.it>

Acknowledgements

Thanks to all the authors of LCG- IA 2022-2023 edition

Part of this material comes from:

- LCG- IA 2022-2023 edition; 23/25 edition, especially I. Di Dio Lavoro
- Fondamenti di Informatica per il Web Design 22, C. Pilato
- IEIM'23 → IEIM'22 from M. D. Santambrogio, Fdl C. Bolchini (and their previous credits)
- Logos from respective proprietaries
- references cited throughout the lecture!
- JS → LCG- IA 2022-2023 edition, L. Mottola, NETWORKED SOFTWARE FOR DISTRIBUTED SYSTEMS
- Git → A. Damiani, Git course
- Git & Github websites

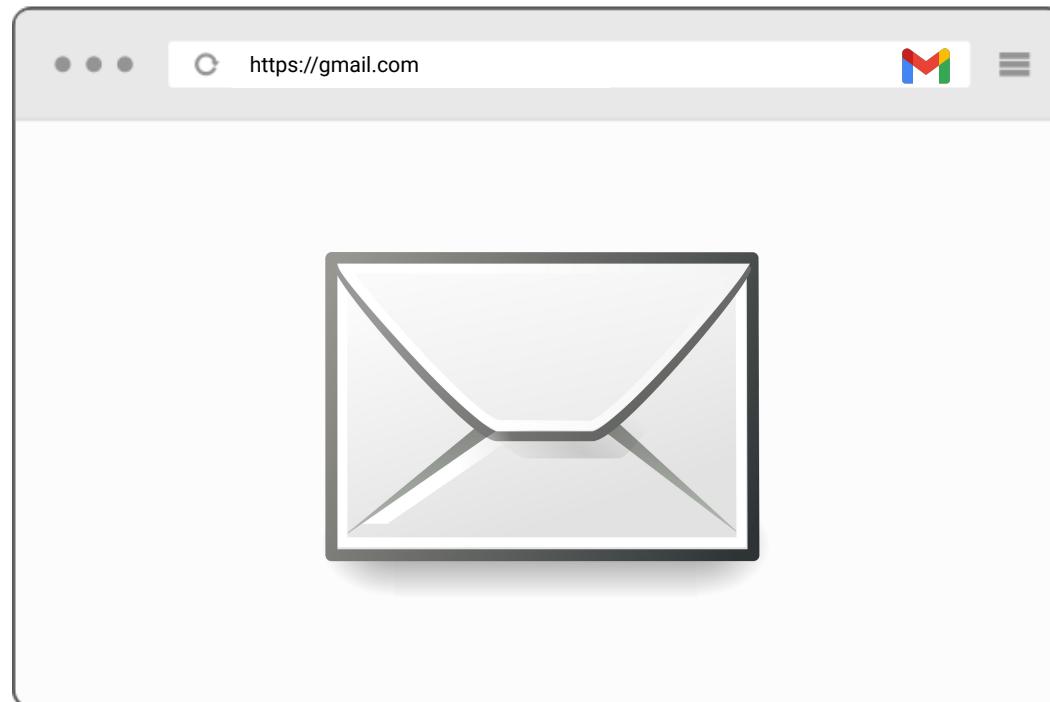
and are *properties of their respective owners*

JavaScript: aspetti di sicurezza



Immaginiamo di aprire due schede nel nostro browser

Scheda 1: Il nostro account di posta elettronica

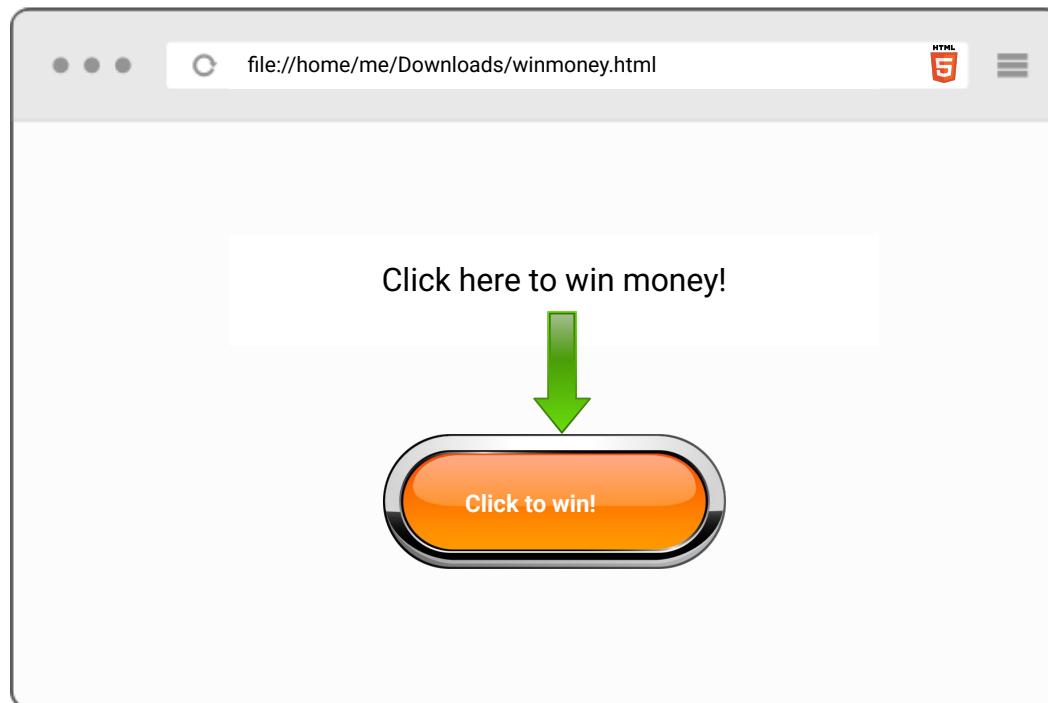


JavaScript: aspetti di sicurezza



Immaginiamo di aprire due schede nel nostro browser

Scheda 2: Un file HTML di dubbia provenienza



JavaScript: aspetti di sicurezza



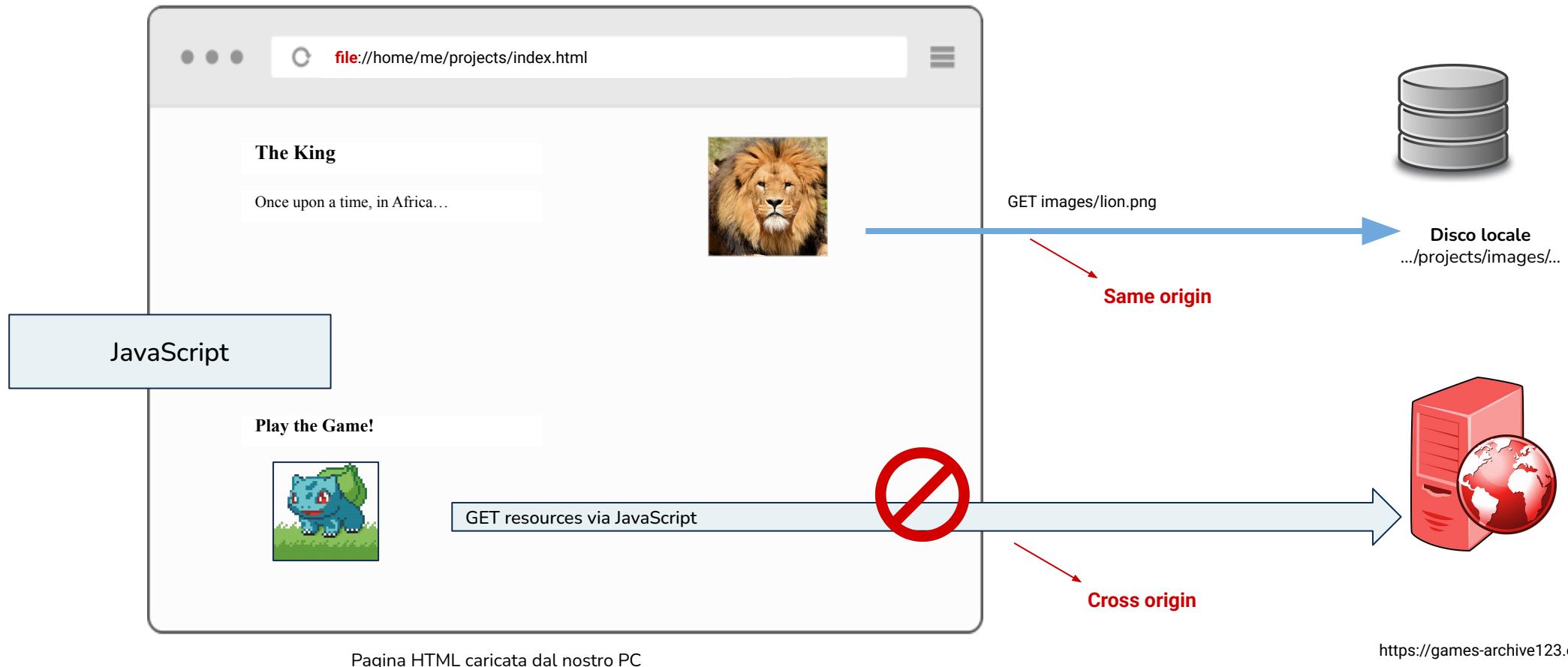
Il file HTML aperto nella Scheda 2 potrebbe, attraverso del codice JavaScript, accedere al contenuto della Scheda 1

L'esecuzione di codice JavaScript di questo tipo consentirebbe il furto di dati e informazioni sensibili da parte di malintenzionati

Per ostacolare questo tipo di attacchi, i browser implementano la cosiddetta **same-origin policy**

Restrizione di accesso a risorse localizzate su domini diversi da quello di provenienza dalla pagina HTML

JavaScript: aspetti di sicurezza





La same-origin policy introduce però delle restrizioni piuttosto forti

...i siti web spesso includono risorse provenienti da domini differenti!

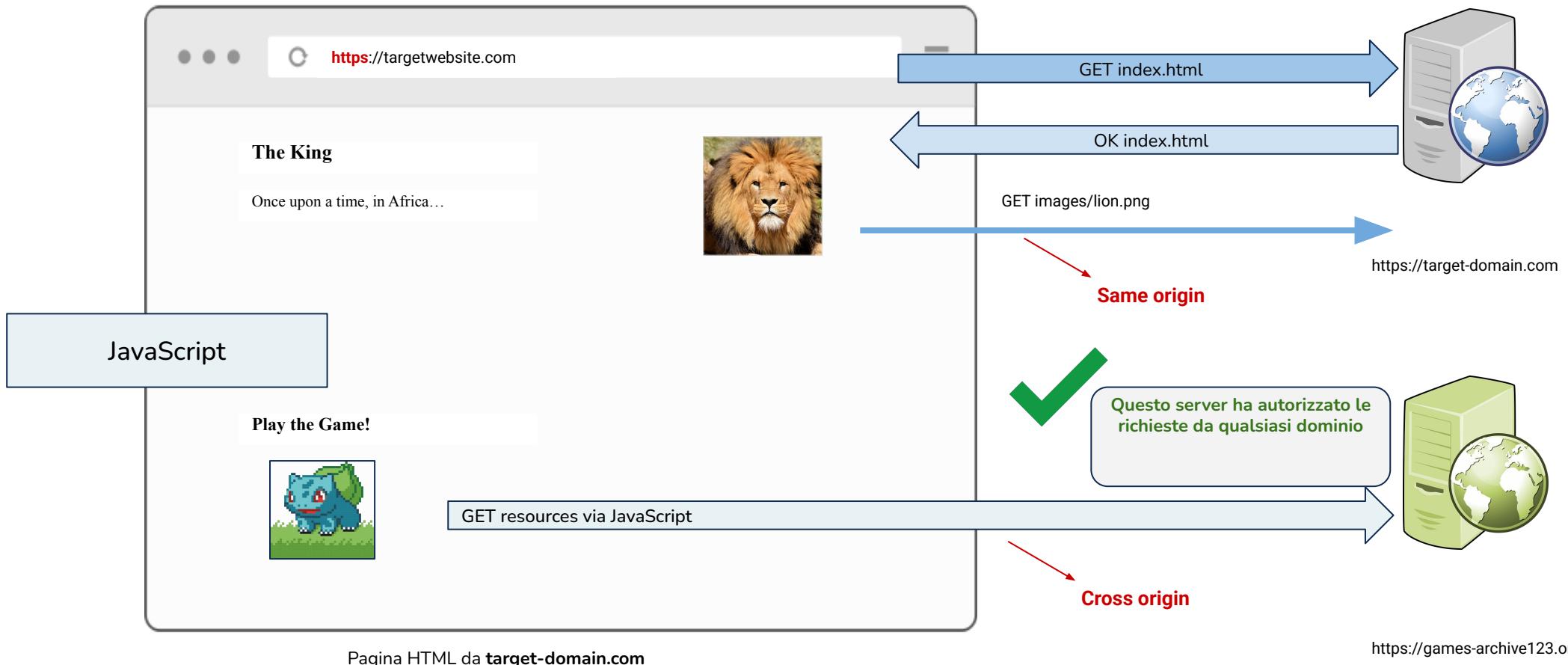
Per permettere l'accesso a risorse su altri domini, i server HTTP implementano la cosiddetta **Cross-Origin Reference Sharing (CORS)**

Le autorizzazioni per le richiesta di accesso vengono gestite dal server HTTP in base al dominio di provenienza

Alcune funzionalità JavaScript possono richiedere l'abilitazione della CORS

La nostra applicazione deve essere gestita tramite server HTTP, come se fosse un sito Web, anziché essere aperta come un file HTML locale

JavaScript: aspetti di sicurezza



Codice binario e sistemi di elaborazione

Se il programmatore (o sviluppatore software) è un essere umano, scrivere sequenze di codice binario non è decisamente una operazione comoda...

Abbiamo bisogno di “parlare” al processore con una lingua diversa (**linguaggio di programmazione**) e lasciare che uno strumento si occupi di tradurla in istruzioni binarie



Linguaggi compilati e interpretati

Il processo di **traduzione** da linguaggio di programmazione ad istruzioni macchina in formato binario può essere di due tipi...

Compilazione: un programma speciale (*compilatore*) genera un file *eseguibile* contenente il programma in formato binario

Interpretazione: un programma speciale (*interprete*) si occupa di tradurre al volo ed eseguire le istruzioni scritte in linguaggio di programmazione

Processo e Programma

Processo ≠ programma !

Processo = programma in esecuzione, composto da:
codice eseguibile (il programma stesso)
dati

Lo stesso programma può essere associato a più processi:

Un programma può essere scomposto in varie parti e ognuna di esse può essere associata a un diverso processo

Lo stesso programma può essere associato a diversi processi quando esso viene eseguito più volte, anche simultaneamente