

LCG: Informatica Applicata - C2

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

Scuola del Design
Laurea Triennale in Communication Design

Aula B6.3.1, Politecnico di Milano

September 17th, 2024

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



POLITECNICO
MILANO 1863



Who We Are (1/2)

Assistant Professor (RTDa) @



POLITECNICO
MILANO 1863



Who We Are (1/2)

Assistant Professor (RTDa) @



POLITECNICO
MILANO 1863

Research in System Architecture Area



Focus: (Co-)Design **Domain-Specific Computer Architectures** and **Systems**

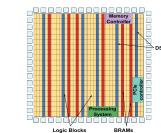
Performance



Energy Efficiency



Reconfigurable Systems
(e.g., FPGAs)





Who We Are (1/2)

Assistant Professor (RTDa) @



POLITECNICO
MILANO 1863

Research in System Architecture Area

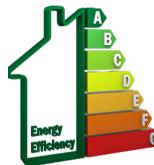


Focus: (Co-)Design **Domain-Specific** Computer Architectures and Systems

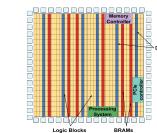
Performance



Energy Efficiency



Reconfigurable Systems
(e.g., FPGAs)

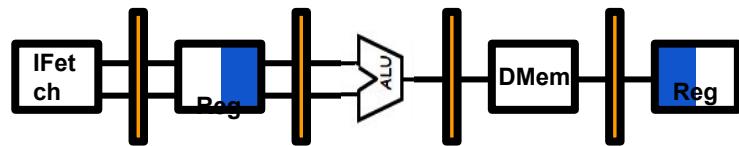


Teaching Activities@ Polimi

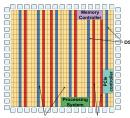
CS101 (Bachelor)



Advanced Computer Architecture (Master)



FPGA101 (Passion in Action)





Who We Are (1/2)

Assistant Professor (RTDa) @



POLITECNICO
MILANO 1863

Research in System Architecture Area



Focus: (Co-)Design **Domain-Specific** Computer Architectures and Systems

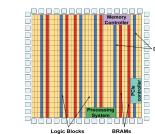
Performance



Energy Efficiency



Reconfigurable Systems
(e.g., FPGAs)

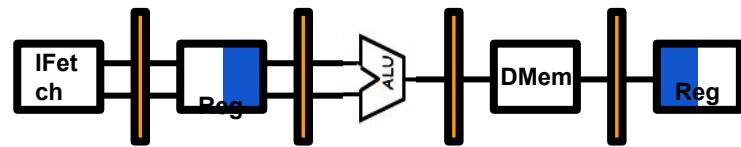


Teaching Activities@ Polimi

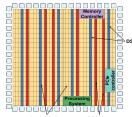
CS101 (Bachelor)



Advanced Computer Architecture (Master)



FPGA101 (Passion in Action)



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

IBM Research | Zurich

XILINX

ORACLE®



Who We Are (2/2)

PhD Student @



POLITECNICO
MILANO 1863

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



POLITECNICO
MILANO 1863



Who We Are (2/2)

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



POLITECNICO
MILANO 1863

Who We Are (2/2)



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 (1/2)

- 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.



Opportunità di Supporto dall'Ateneo (2/2)

- **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

- Seconda edizione del corso: abbiamo fatto aggiustamenti, il **vostro contributo sarà 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



Obiettivo di Informatica Applicata

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



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



Obiettivo di Informatica Applicata

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



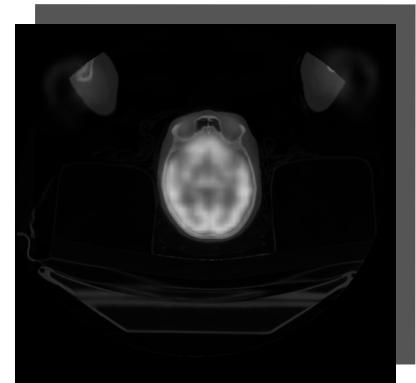
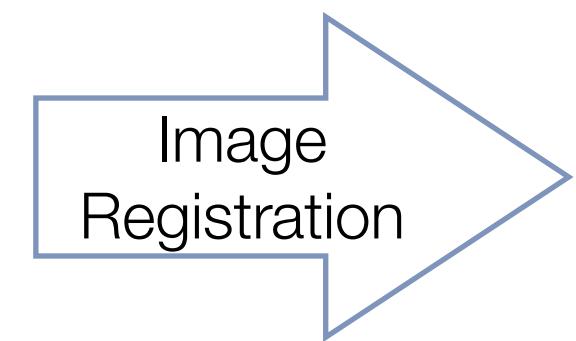
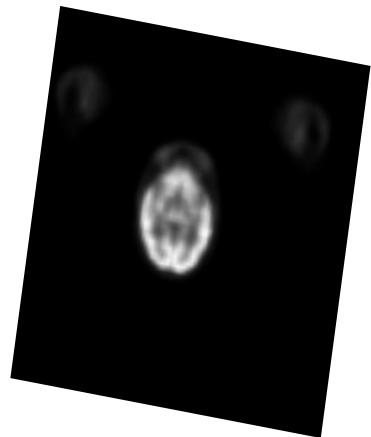
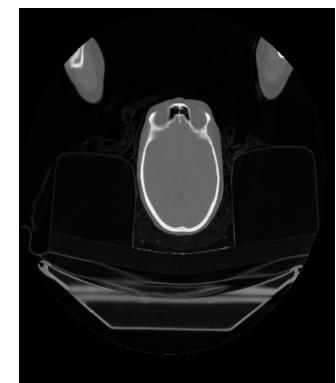
Sfruttare codice per creare artefatti digitali con una
focalizzazione a partire dai dati



Sfruttare piattaforme web per comunicare e creare tali
artefatti digitali



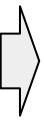
La Visualizzazione nella mia Ricerca



Visualizzare Per Favorire la Fruizione



Visualizzare Per Favorire la Fruizione



Floating
image

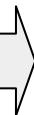


Reference
image



Visualizzare Per Favorire la Fruizione

$$MI(X; Y) = H(X) + H(Y) - H(X; Y)$$



Floating
image



Reference
image



Visualizzare Per Favorire la Fruizione

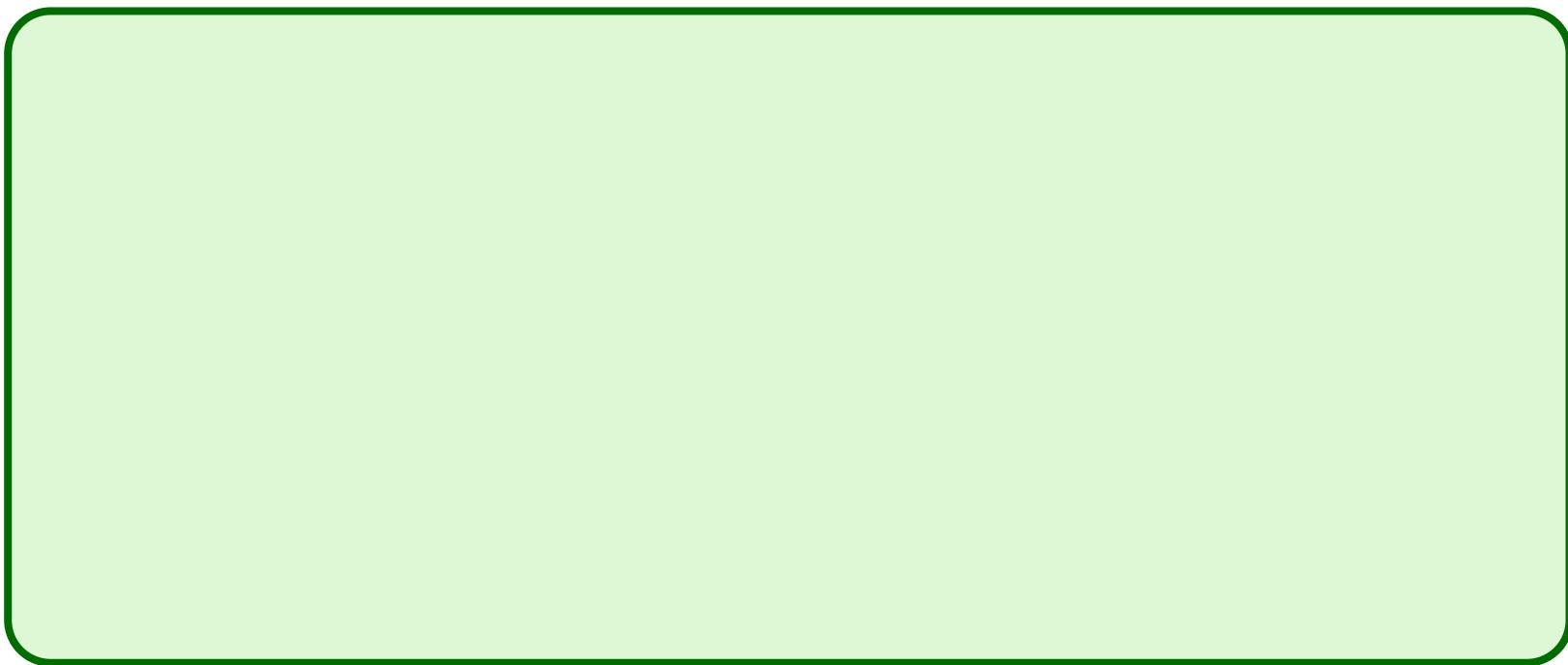
$$MI(X; Y) = H(X) + H(Y) - H(X; Y)$$



Floating
image

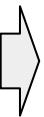


Reference
image



Visualizzare Per Favorire la Fruizione

$$MI(X; Y) = H(X) + H(Y) - H(X; Y)$$



Floating
image



Reference
image

A)

B)



Visualizzare Per Favorire la Fruizione

$$MI(X; Y) = H(X) + H(Y) - H(X; Y)$$

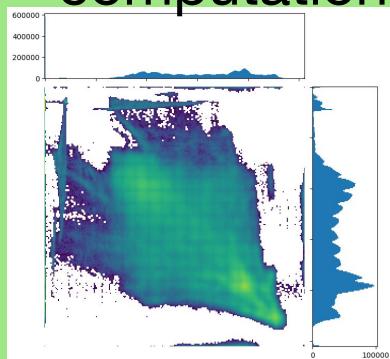


Floating
image



Reference
image

A) Histograms
computation



B)



Visualizzare Per Favorire la Fruizione

$$MI(X;Y) = H(X) + H(Y) - H(X;Y)$$

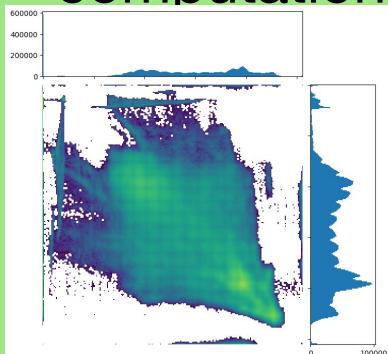


Floating
image



Reference
image

A) Histograms
computation



B) Entropies computation

$$H(X) = - \sum_{x \in X} p(x) \log_2 p(x)$$



Visualizzare Per Favorire la Fruizione

$$MI(X; Y) = H(X) + H(Y) - H(X; Y)$$

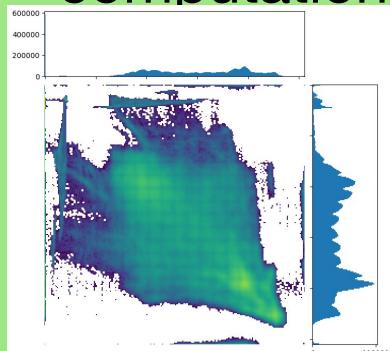


Floating image



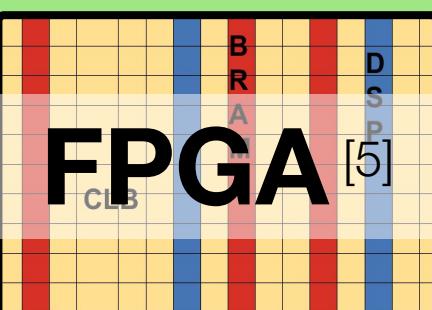
Reference image

A) Histograms computation



B) Entropies computation

$$H(X) = - \sum_{x \in X} p(x) \log_2 p(x)$$



Visualizzare Per Favorire la Fruizione

$$MI(X; Y) = H(X) + H(Y) - H(X; Y)$$

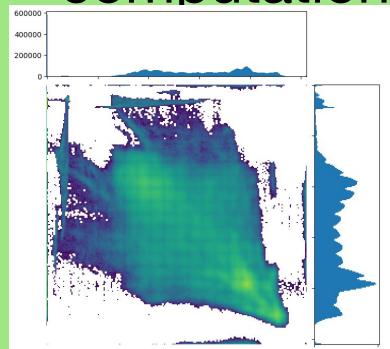


Floating image



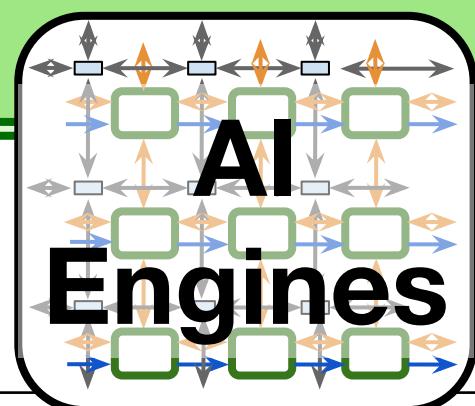
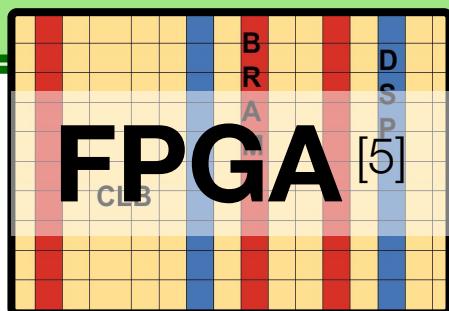
Reference image

A) Histograms computation



B) Entropies computation

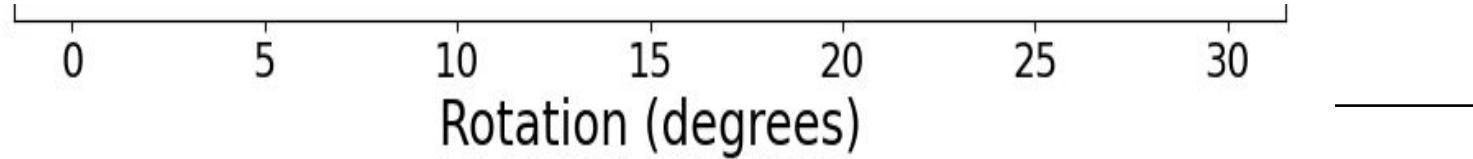
$$H(X) = - \sum_{x \in X} p(x) \log_2 p(x)$$



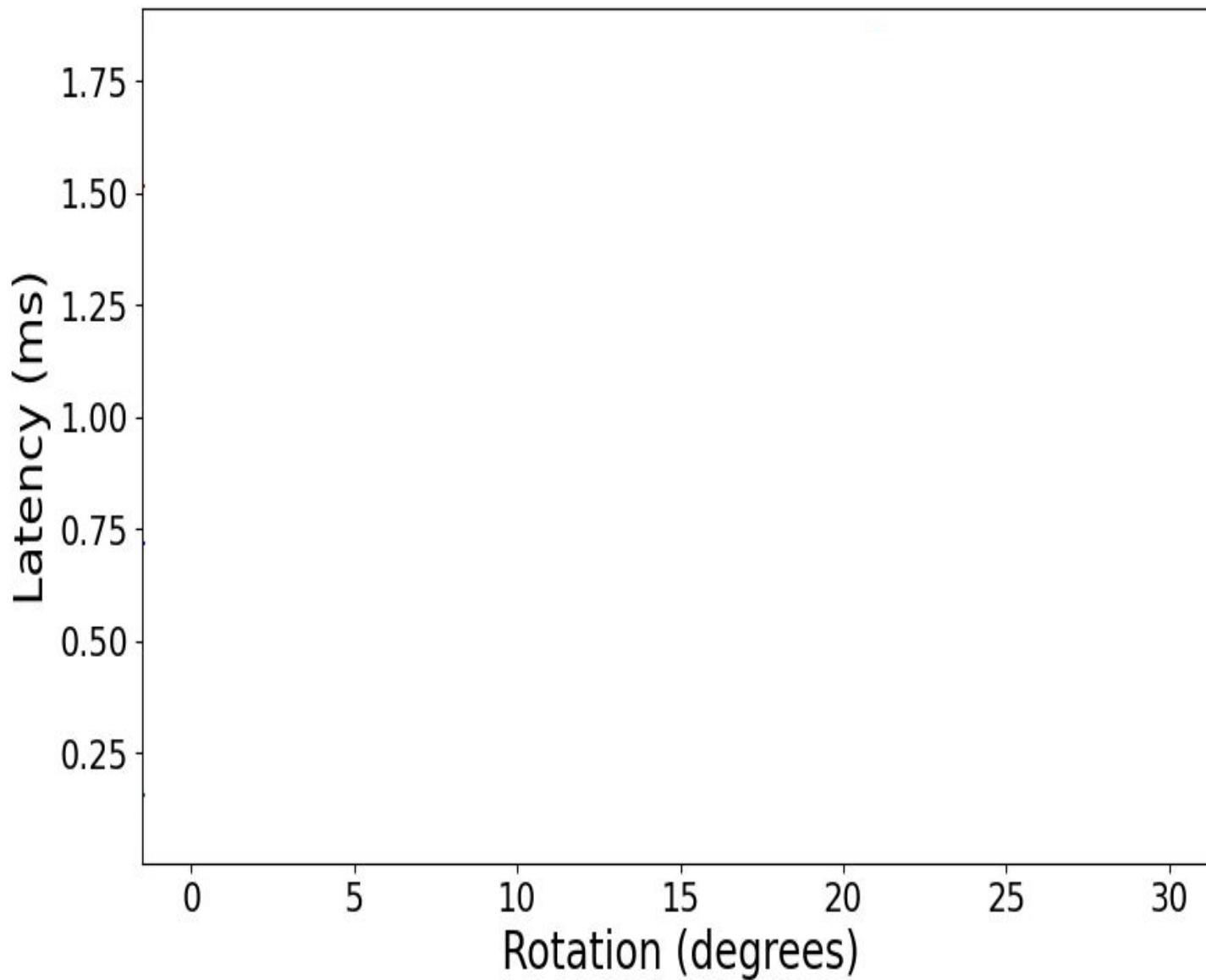
Approccio Computazionale per i Risultati



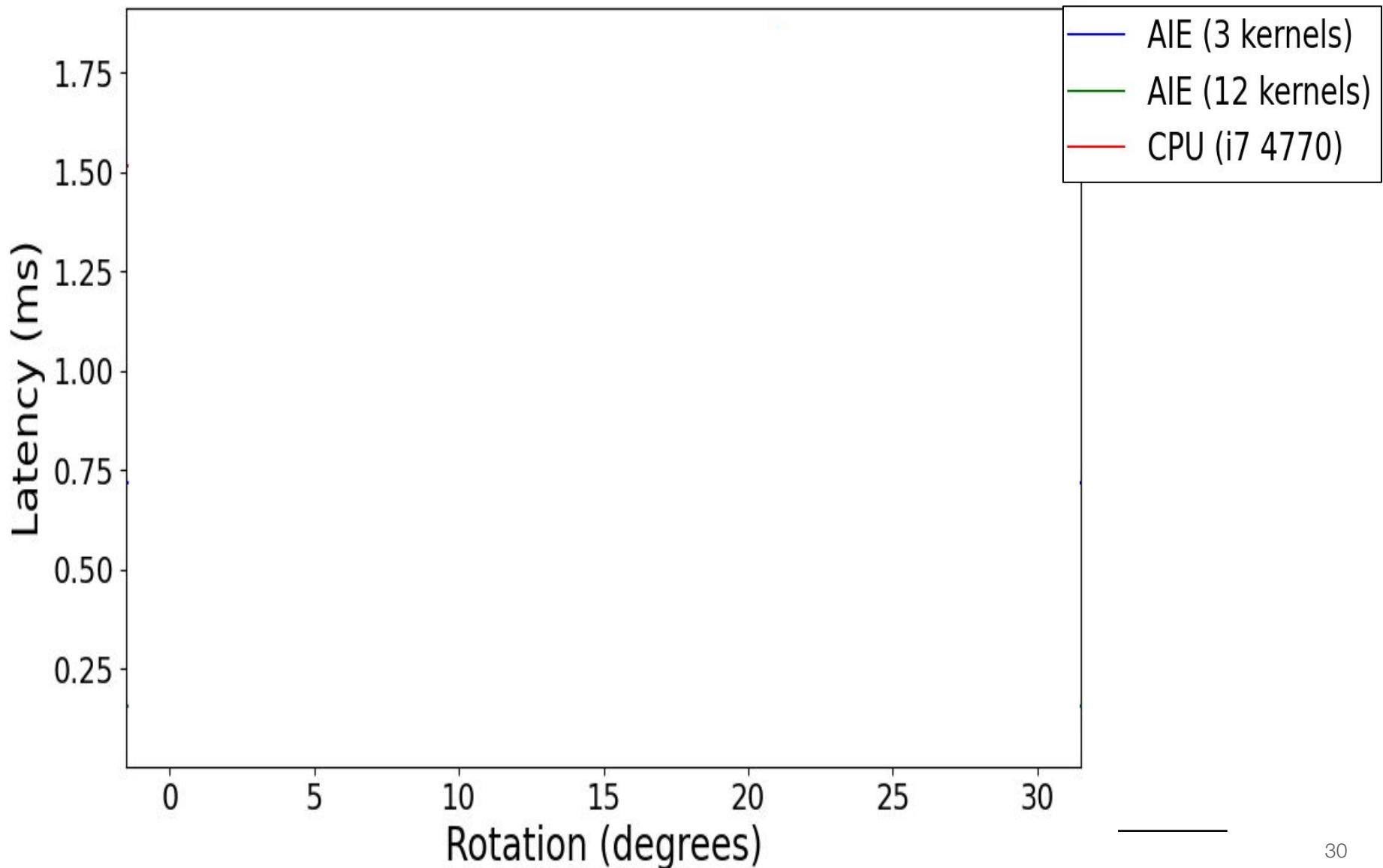
Approccio Computazionale per i Risultati



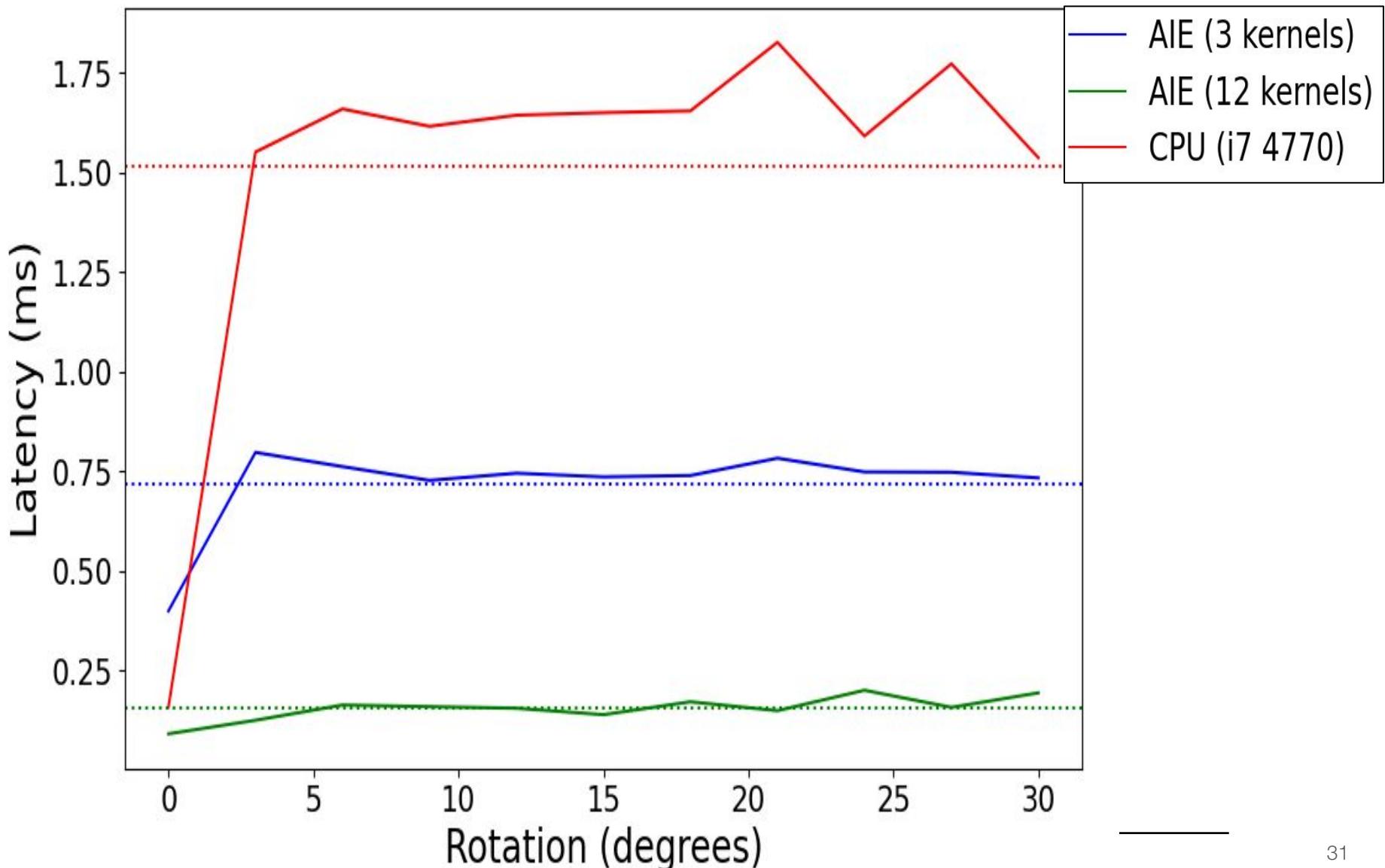
Approccio Computazionale per i Risultati



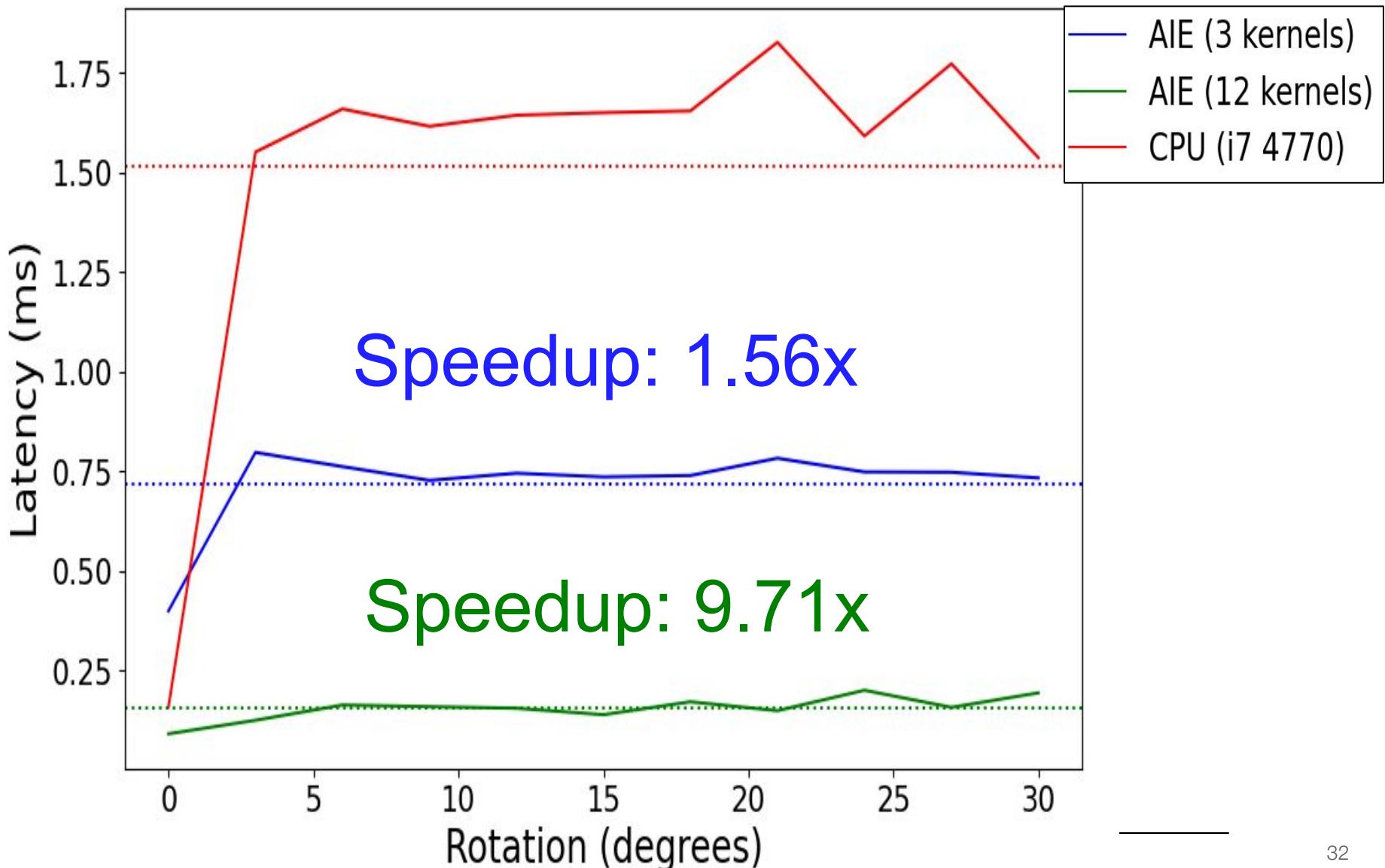
Approccio Computazionale per i Risultati



Approccio Computazionale per i Risultati



Approccio Computazionale per i Risultati



Contenuti di Informatica Applicata

Computazione tramite



Contenuti di Informatica Applicata

Computazione tramite



**057306 - FONDAMENTI DI
INFORMATICA PER IL WEB
DESIGN (PILATO CHRISTIAN)
[2023-24]**



POLITECNICO
MILANO 1863

Loghi proprietari, Images from flaticon.com

Contenuti di Informatica Applicata

Computazione tramite



**057306 - FONDAMENTI DI
INFORMATICA PER IL WEB
DESIGN (PILATO CHRISTIAN)
[2023-24]**

Strumenti Sviluppo



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



Remoto
Interattivo

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



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

Contenuti di Informatica Applicata

Computazione tramite



**057306 - FONDAMENTI DI
INFORMATICA PER IL WEB
DESIGN (PILATO CHRISTIAN)
[2023-24]**

Strumenti Sviluppo



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



Remoto
Interattivo

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

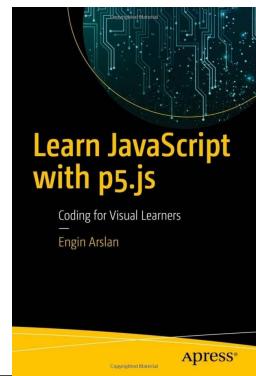
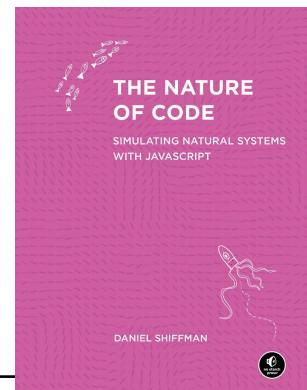
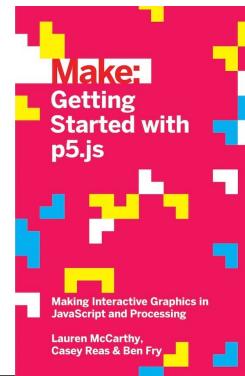


<https://desktop.github.com/>

Per visualizzare e manipolare ci appoggeremo a



p5*.js



POLITECNICO
MILANO 1863

Loghi proprietari, Images from flaticon.com

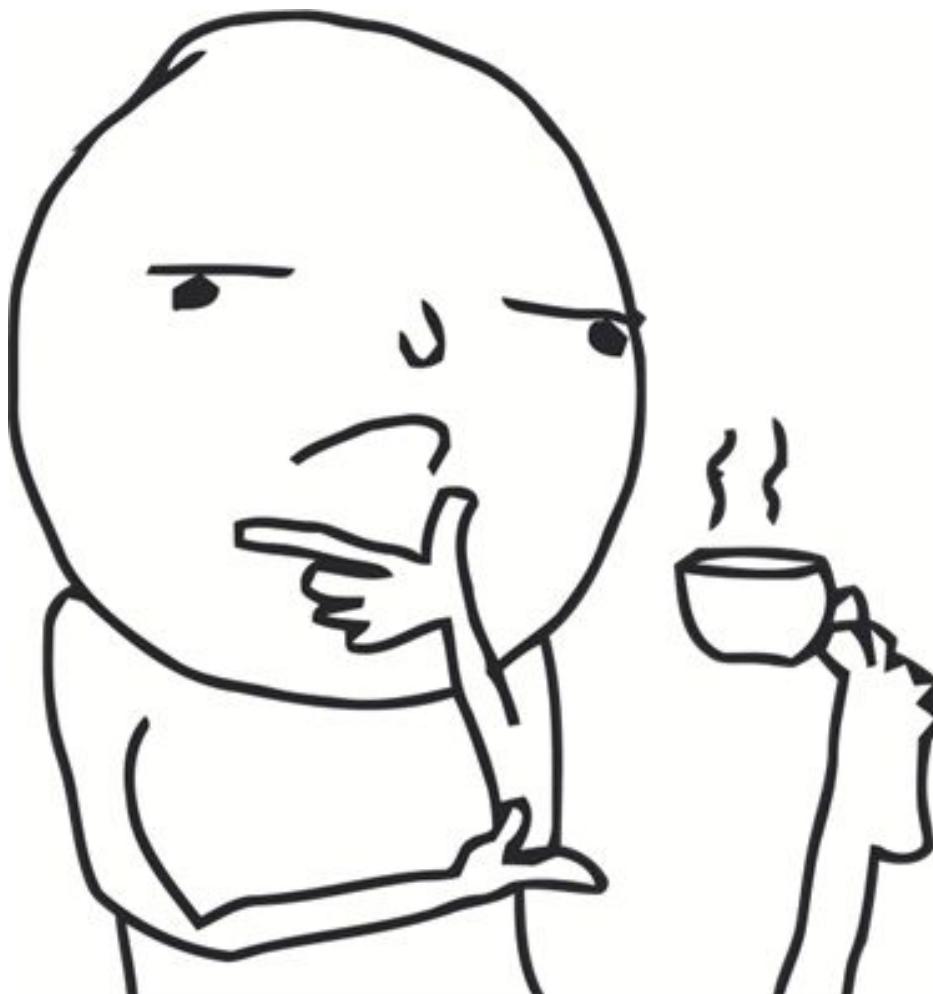
Ok



POLITECNICO
MILANO 1863

Image by [Robin Higgins](#) from [Pixabay](#)

Ok...ma Come Partiamo?



Informatica Self-Assessment



Informatica Self-Assessment

<https://forms.office.com/e/40z2YF0ZnA>



Today Objectives

1. ~~Informatica Self Assessment~~

2. Local dev with VSCode, Indice html for IA2425  

3. Create our Webpage with Githubpages  

4. Draw with P5, P5 Live Editor

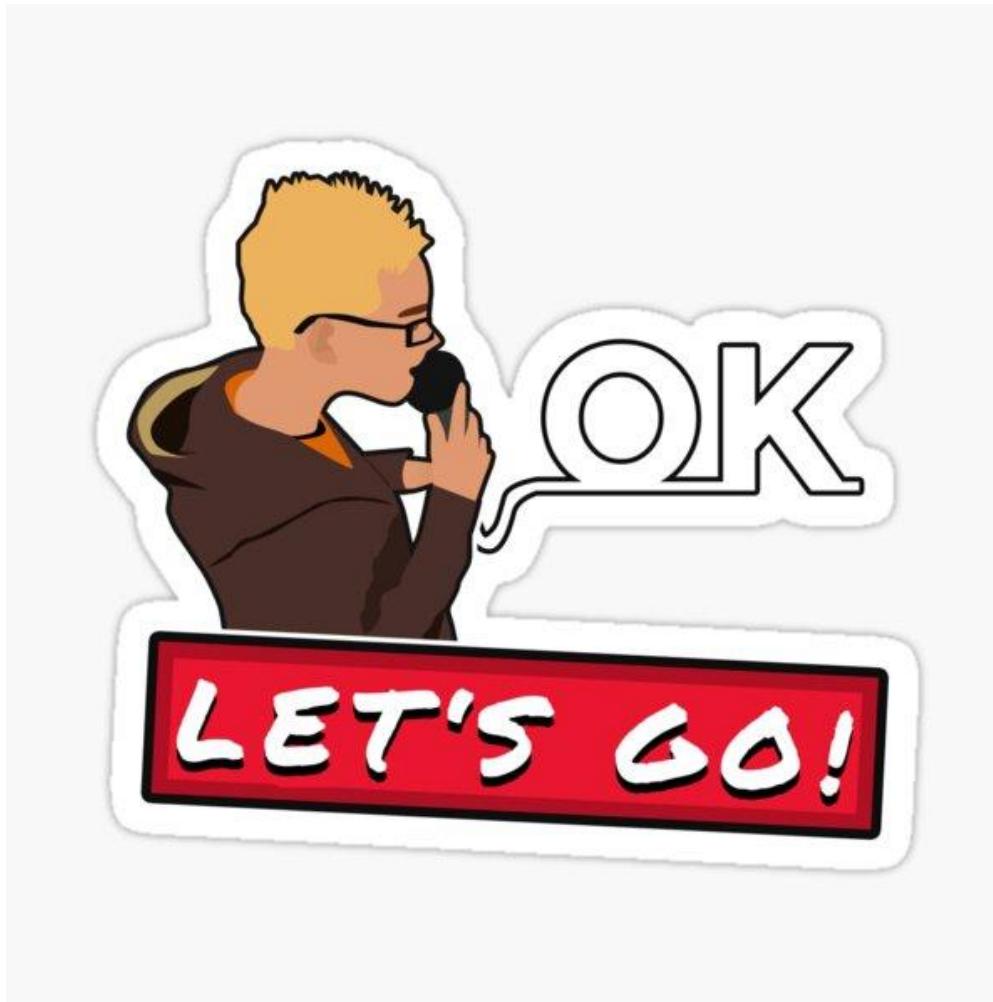
p5*js

p5*

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



Pillole di Informatica



Definizione di Informatica

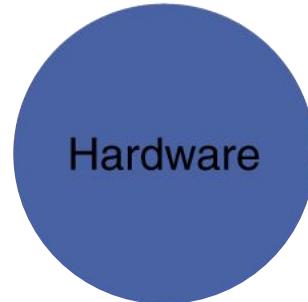


Definizione di Informatica

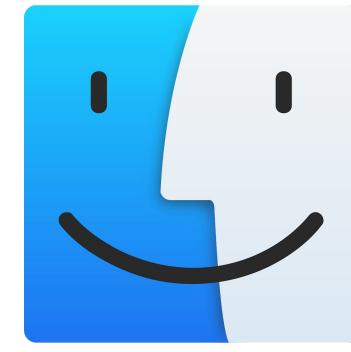
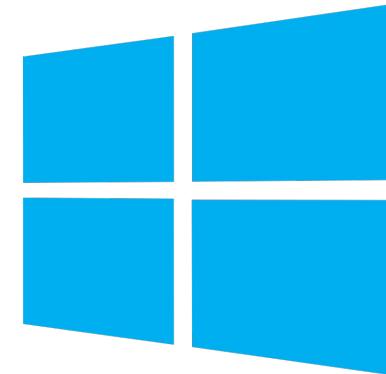
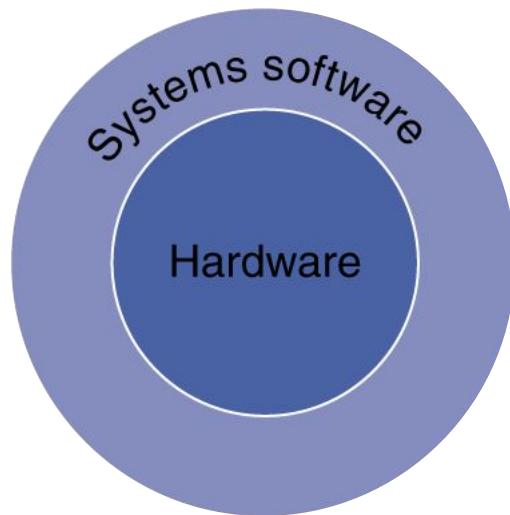
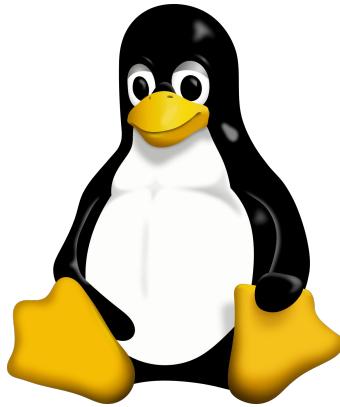
Scienza della
rappresentazione
e dell'
elaborazione
dell'informazione



Lo Stack di Astrazione di riferimento



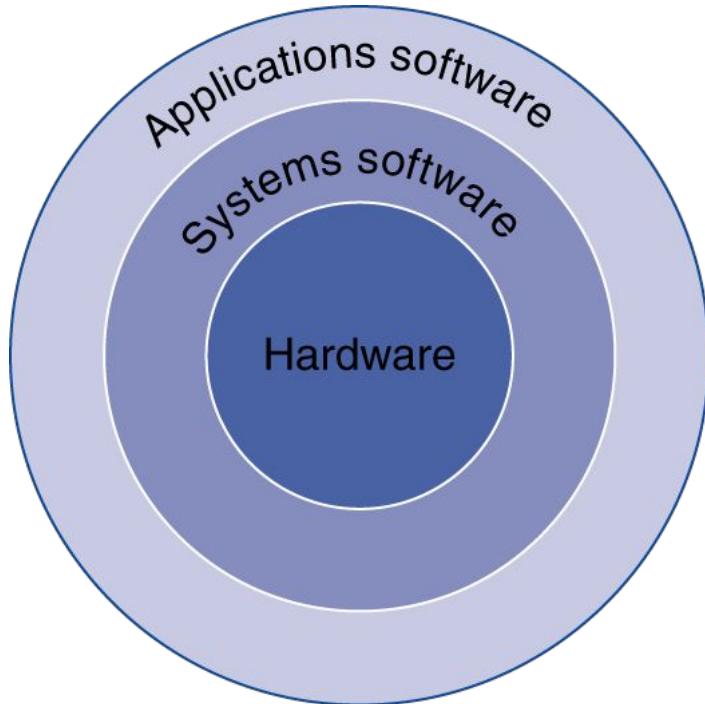
Lo Stack di Astrazione di riferimento



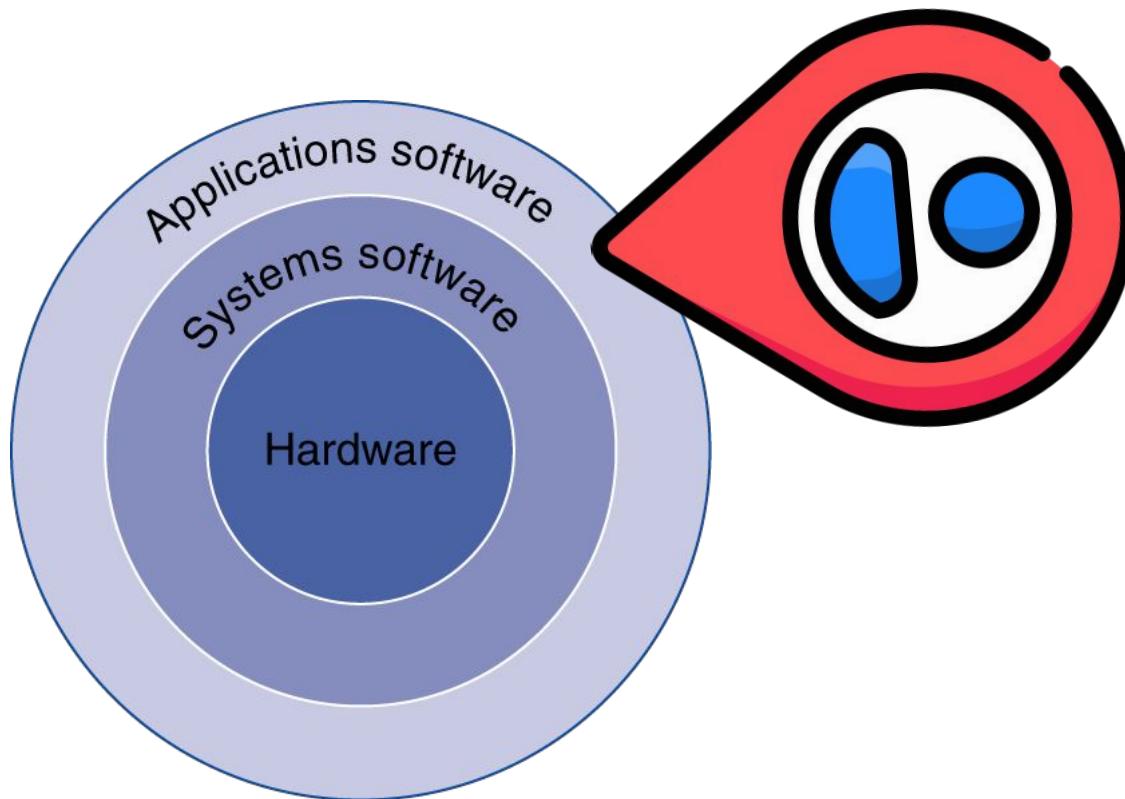
Mac OS



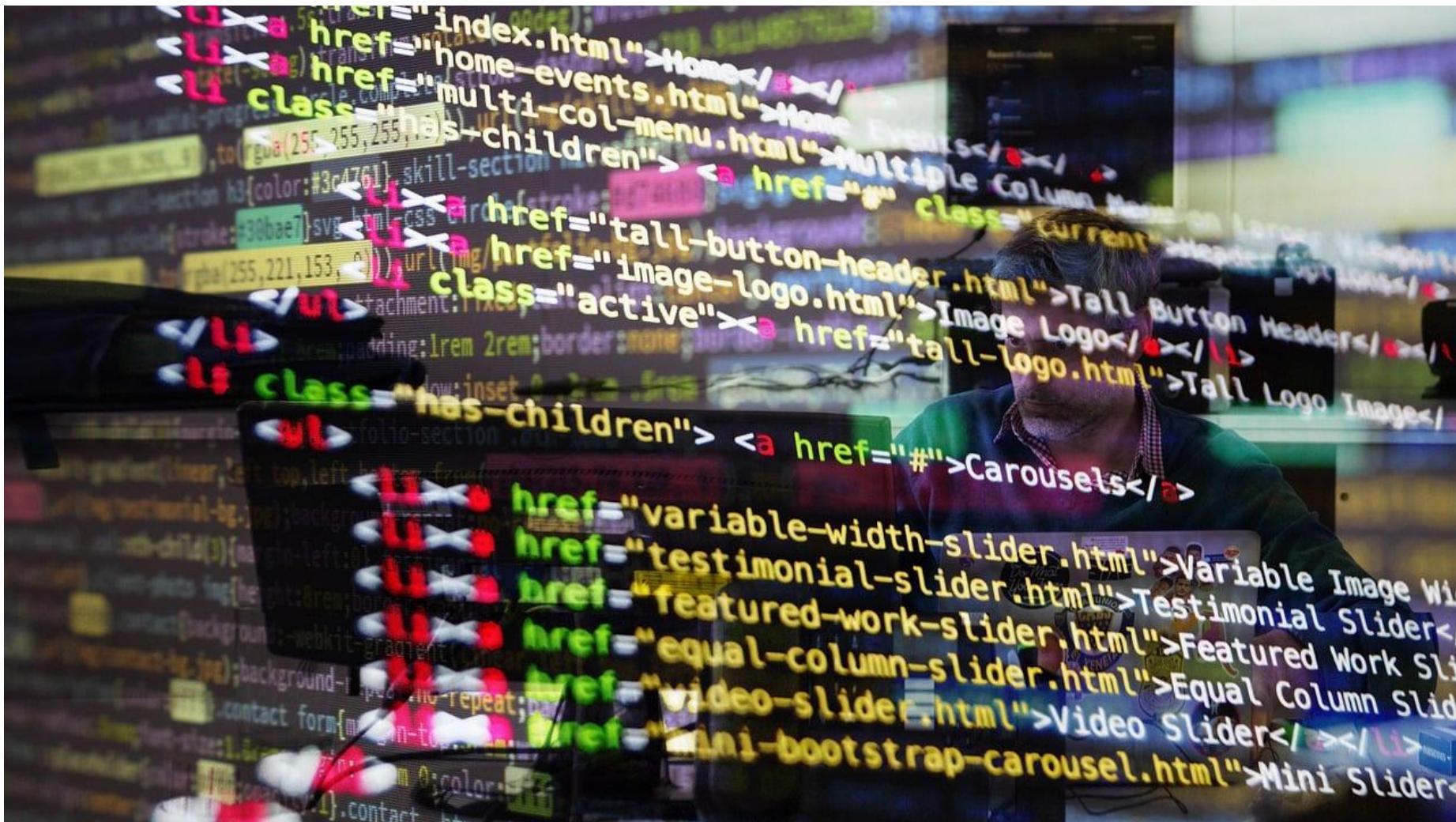
Lo Stack di Astrazione di riferimento



Lo Stack di Astrazione di riferimento



Configurare l'Ambiente di Sviluppo Locale



Visual Studio Code



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

Si tratta di un software liberamente installabile da:

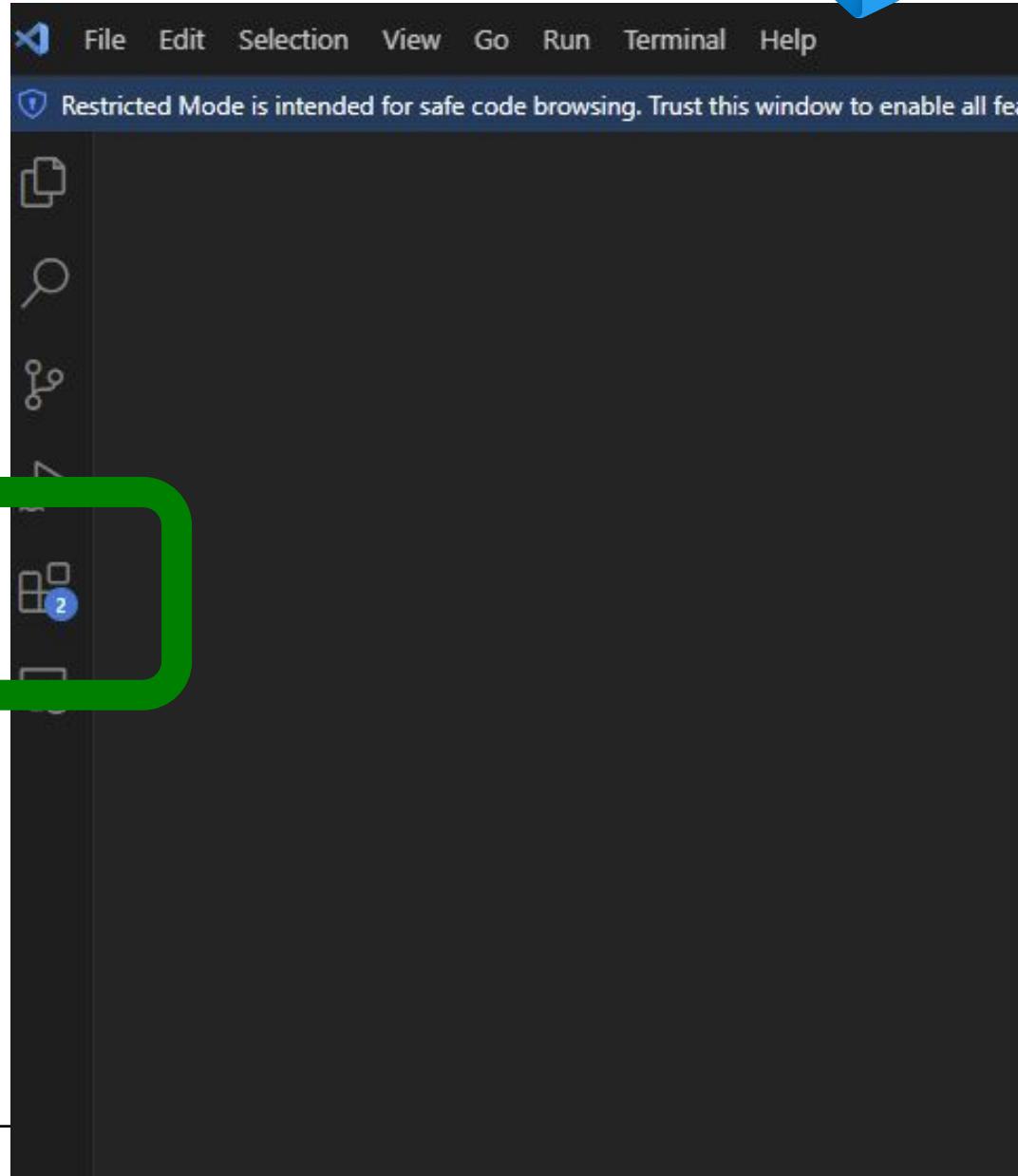
<https://code.visualstudio.com/download>

I repository/market del proprio sistema operativo

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



Visual Studio Code

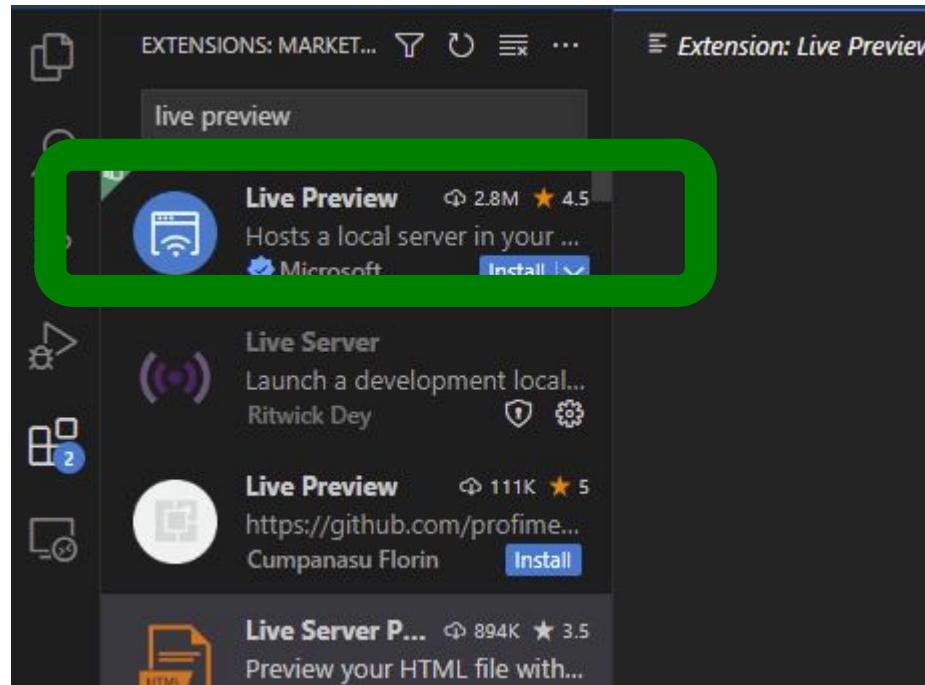


Aprire il pannello di
installazione
delle estensioni



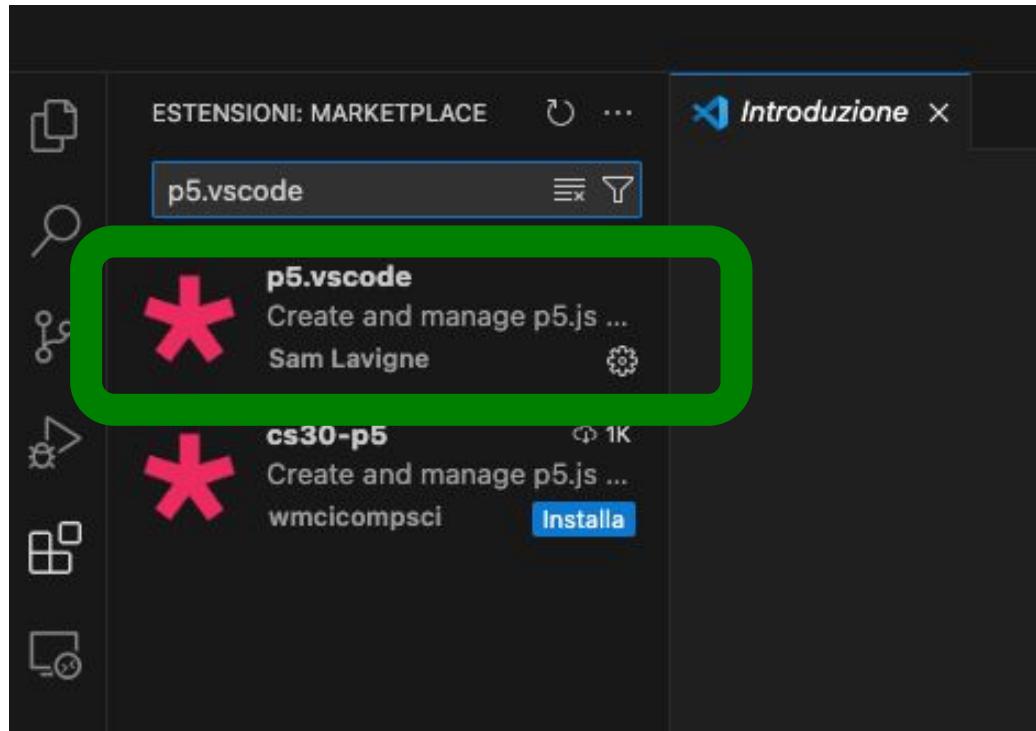
Visual Studio Code

Cercare l'estensione digitando
“live preview” nella barra di ricerca e installatela

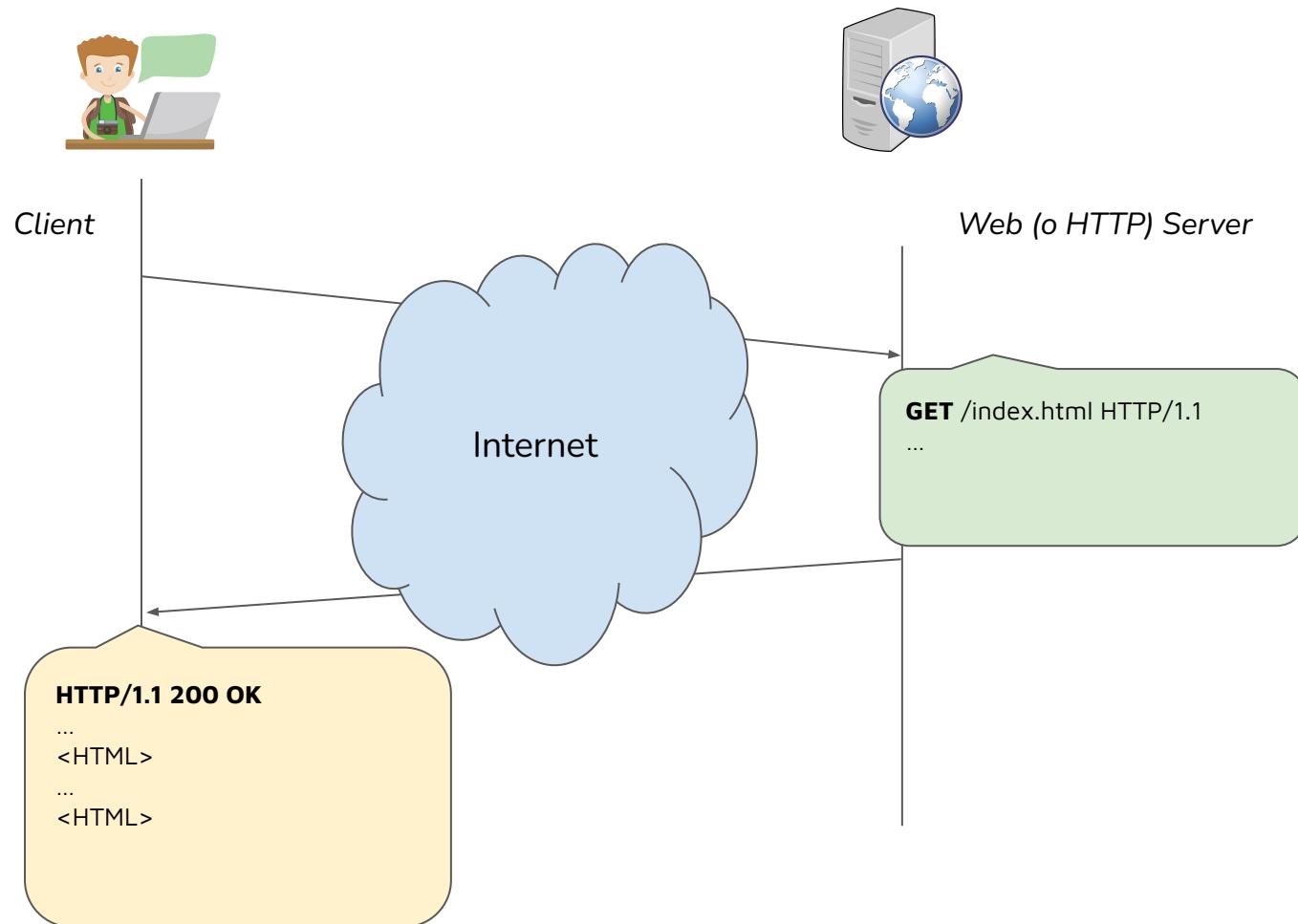


Visual Studio Code

Cercare l'estensione digitando
“p5.vscode” nella barra di ricerca e installatela



Internet in a Nutshell



Let's try it with VSCode!



POLITECNICO
MILANO 1863

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



Internet

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



Internet: indirizzamento



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

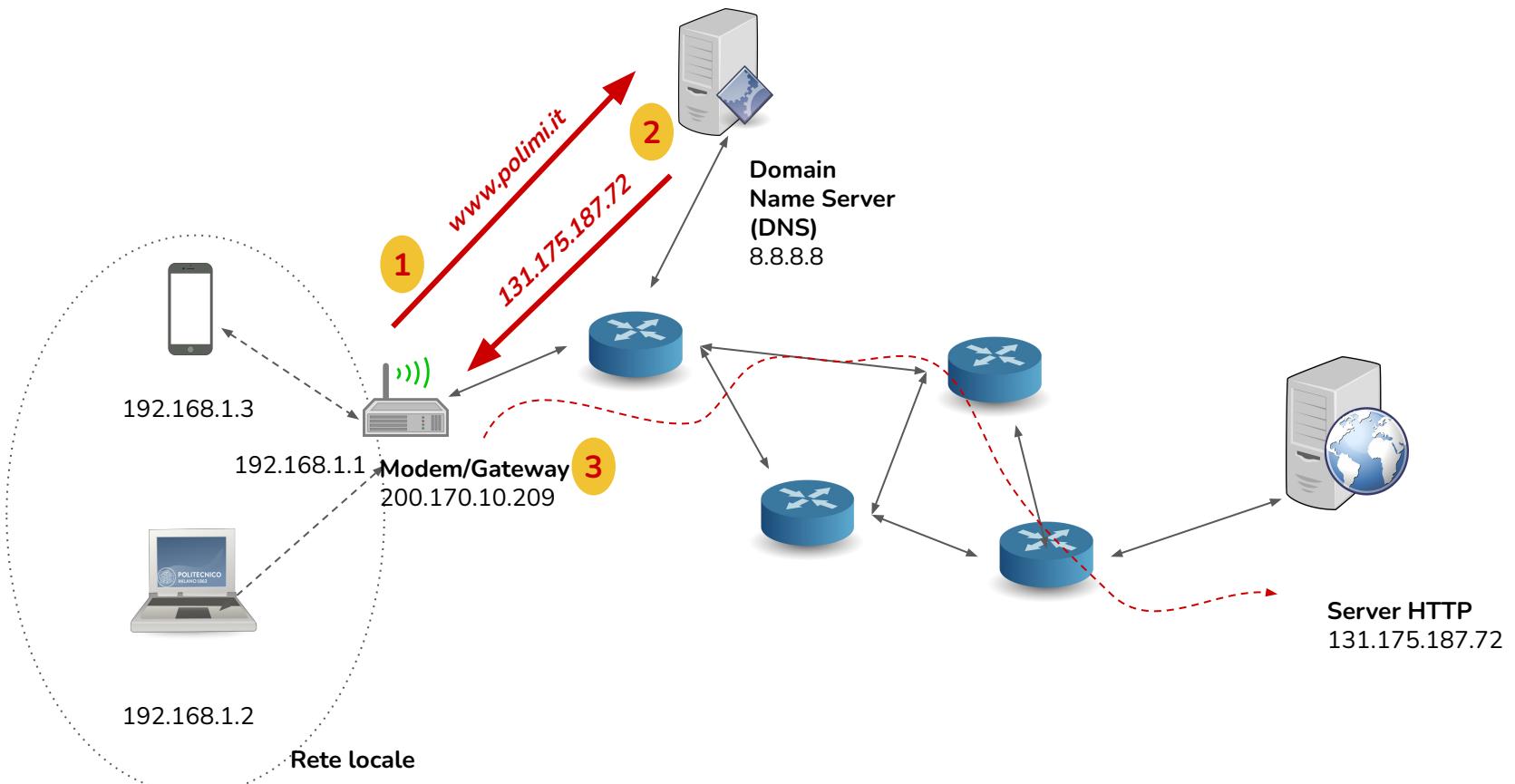
Una combinazione di 4 valori da 0 a 255, separati da un punto: esempio **131.175.187.72**

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

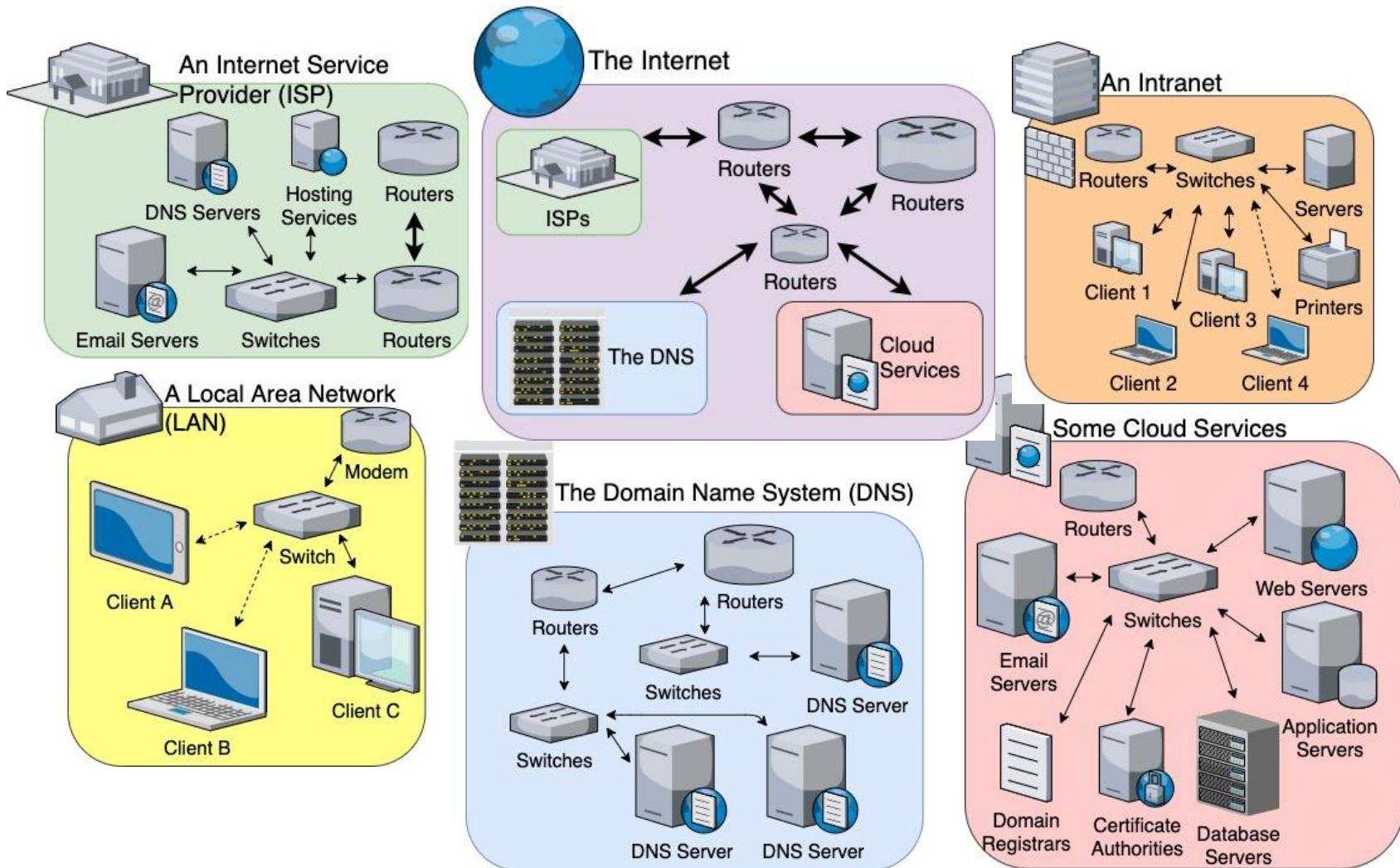
Oppunti sistemi di rete, detti **Domain Name Servers (DNS)** fungono da “rubriche” per la traduzione in indirizzi IP



Internet: indirizzamento



Internet: infrastruttura



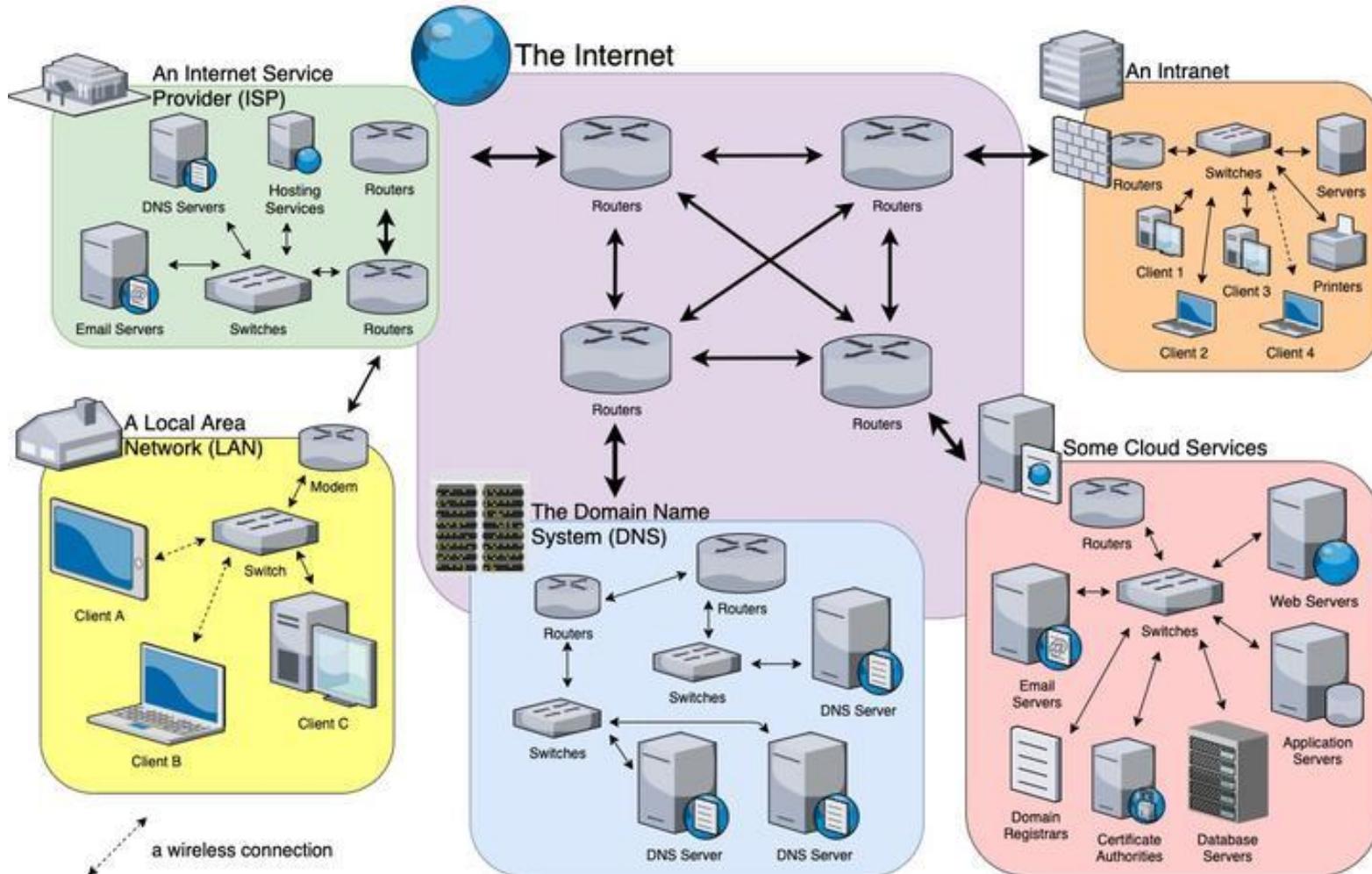
<https://vahid.blog/post/2020-12-15-how-the-internet-works-part-i-infrastructure/>





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...

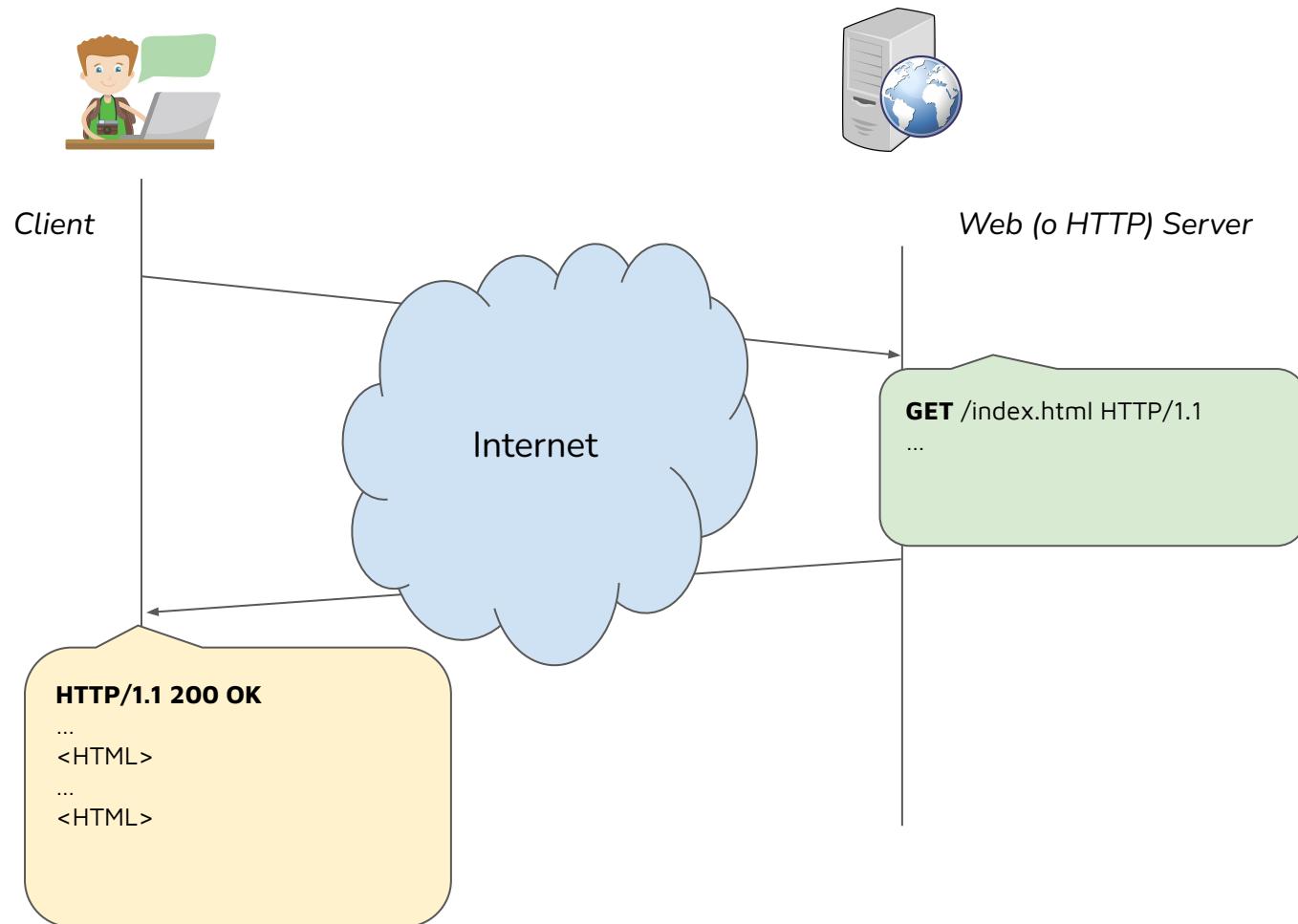


Il browser assolve due compiti principali

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)



HTML: HyperText Markup Language

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

HTML: HyperText Markup Language

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>



HTML e JavaScript

Le pagine HTML mostreranno 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>



HTML e 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>
```



HTML e JavaScript

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

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

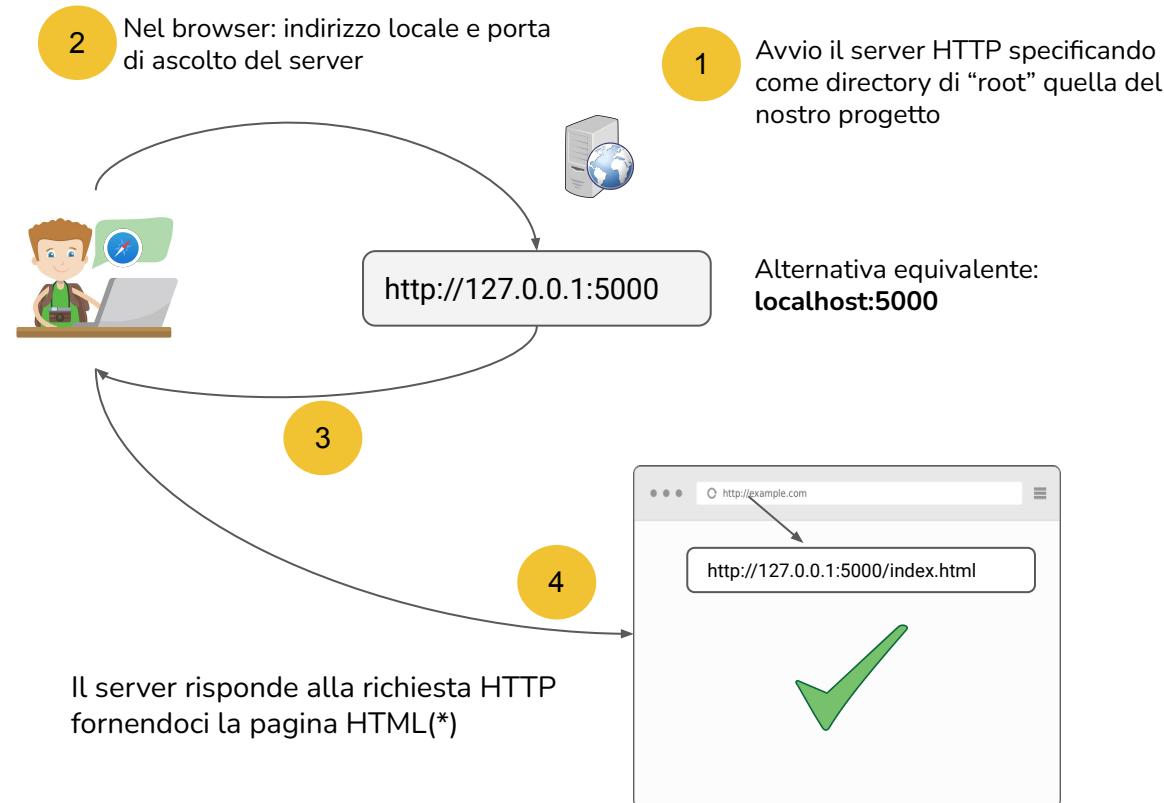
Una volta in esecuzione, il server HTTP rimarrà in ascolto sulla **porta** specificata in attesa di connessioni

...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



(*) La pagina HTML deve essere localizzata nella directory di lavoro del server



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

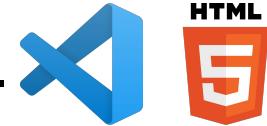
VSCode include infatti una estensione (**Live Preview**) attraverso la quale possiamo avviare un server HTTP locale e avviare con un click l'applicazione che stiamo sviluppando



Today Objectives

1. ~~Informatica Self Assessment~~

2. ~~Local dev with VSCode, Indice html for IA2425~~



3. Create our Webpage with Githubpages



GitHub
Desktop

4. Draw with P5, P5 Live Editor

p5*js



p5*

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



POLITECNICO
MILANO 1863

Github Desktop Installation Party

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



Today's topics



Git

- GitHub
- GitHub Desktop



GitHub Pages

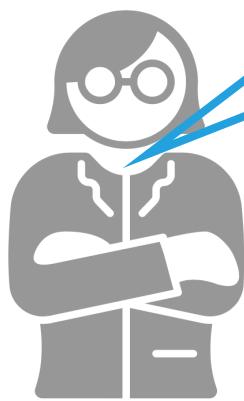


(Collaborative and Asynchronous) Development

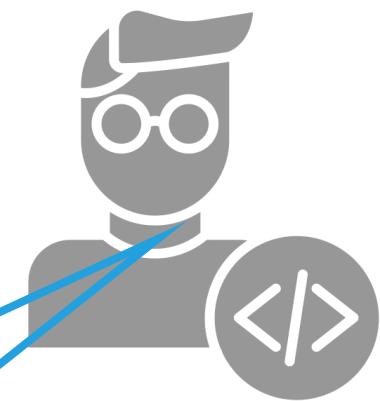
(vaguely) Inspired by «The Expert»

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

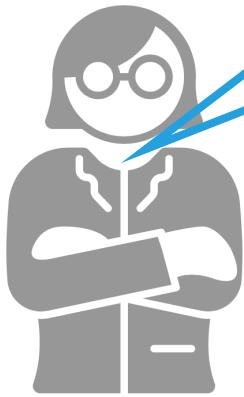




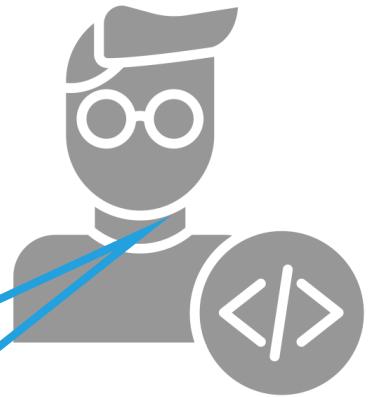
New feat req!
We want 7 red
lines!!!



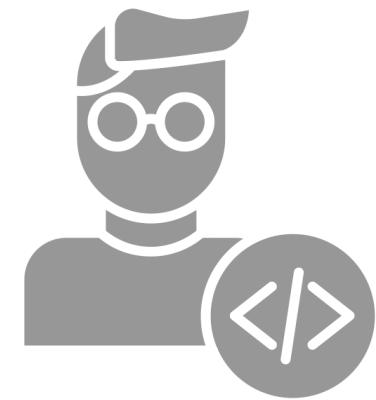
Sure! More precisely?

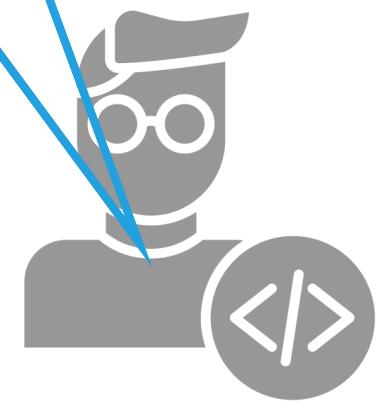


Weeeeeell, we need
lines ... we need
seven of those ...
they must be red!



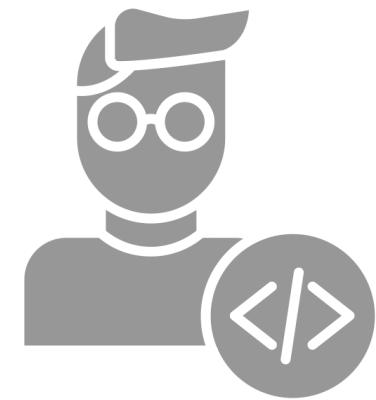
..... ok

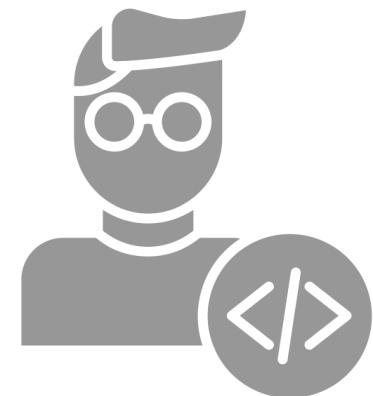
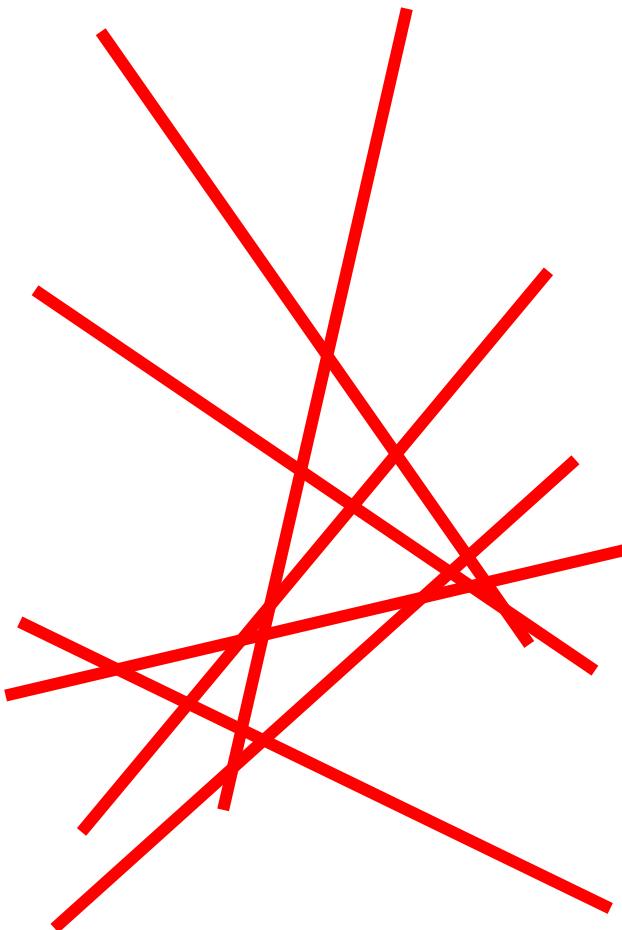


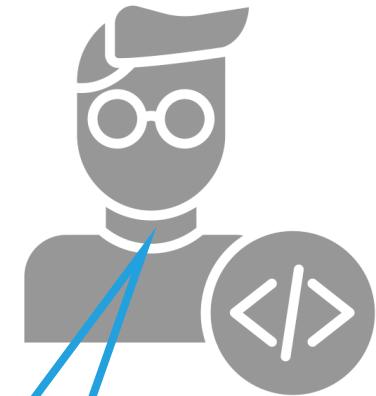
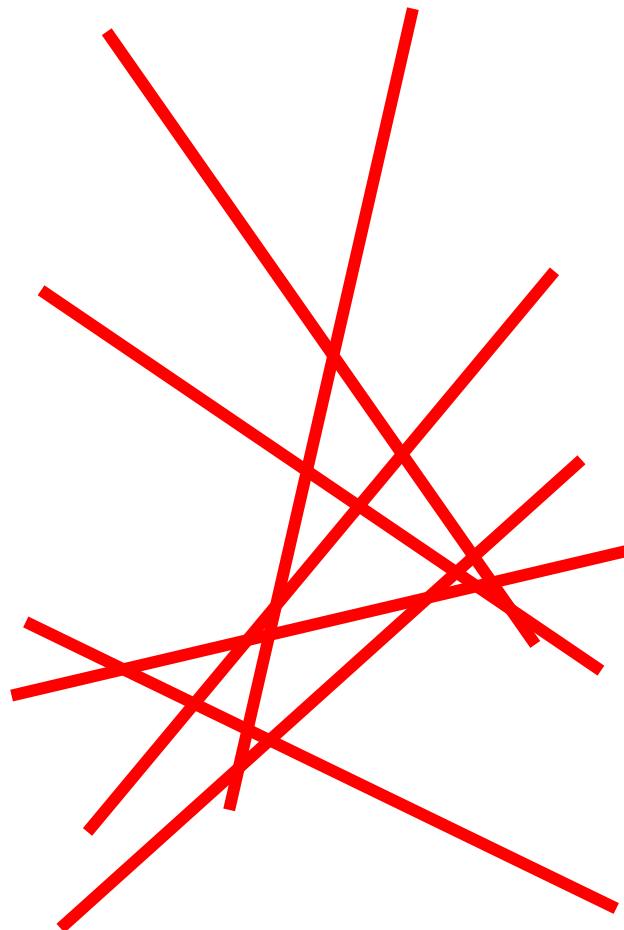
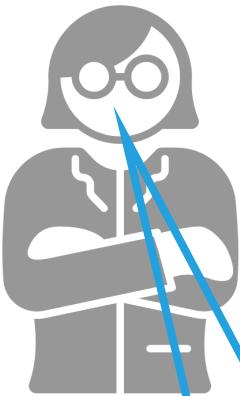


Booo!

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

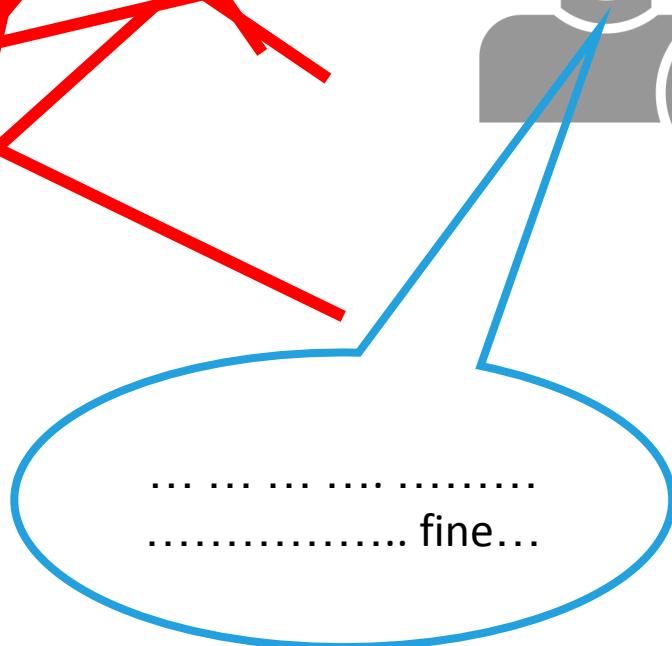
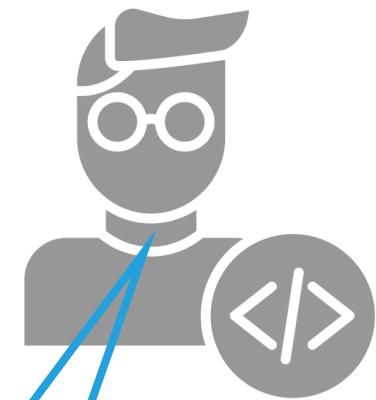
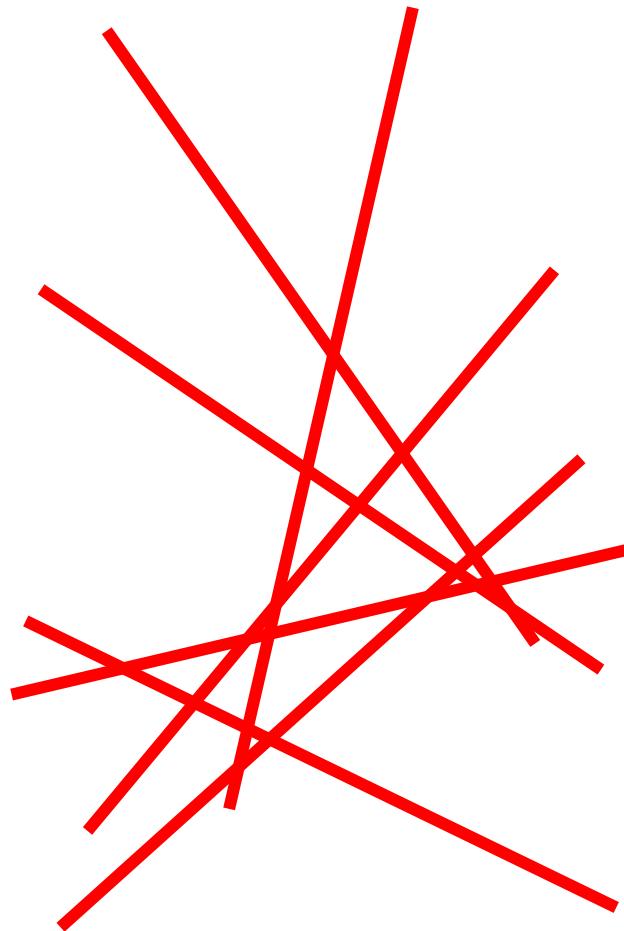


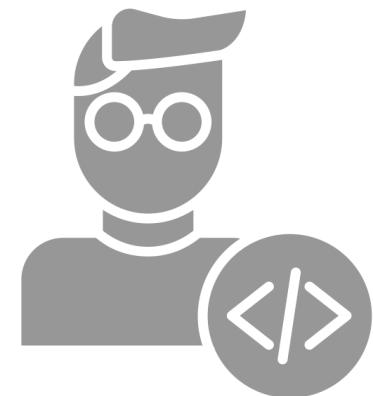
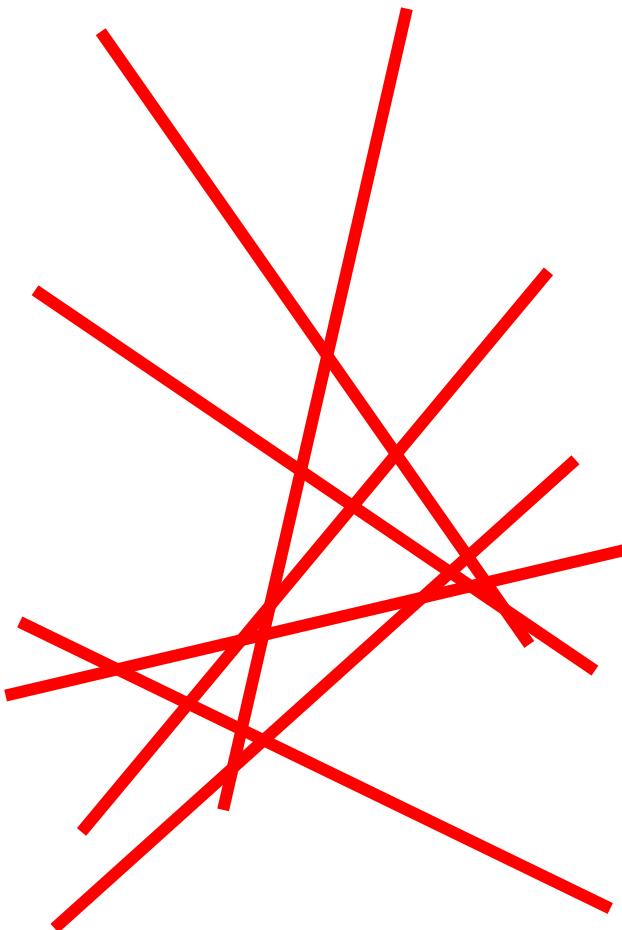


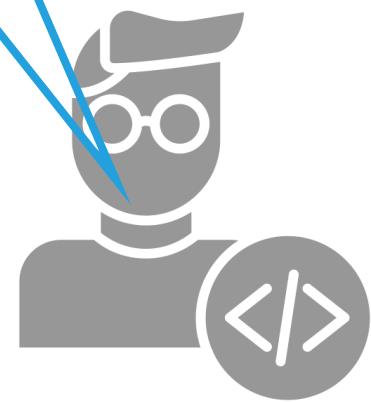
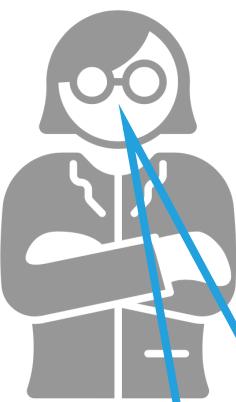


Ah! Too chaotic!
What about ... a green
line and all the other 6
lines perpendicular to it?

How about this?

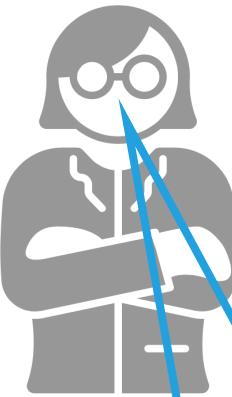




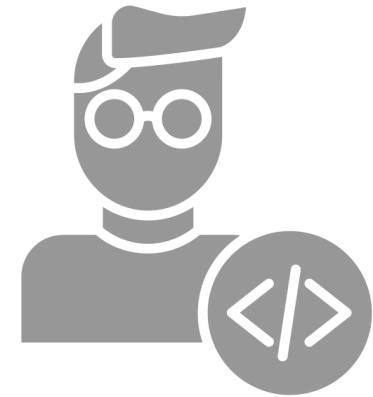
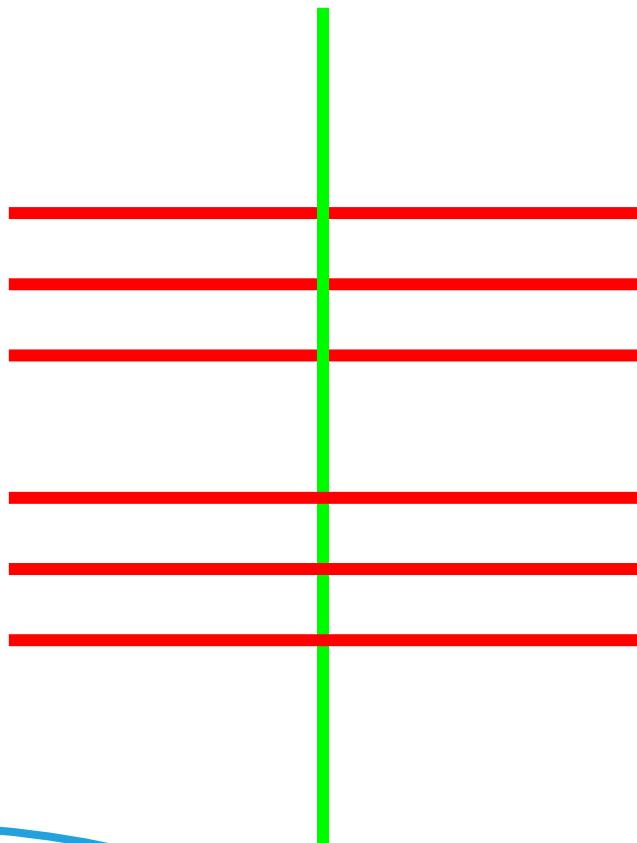


What about now?

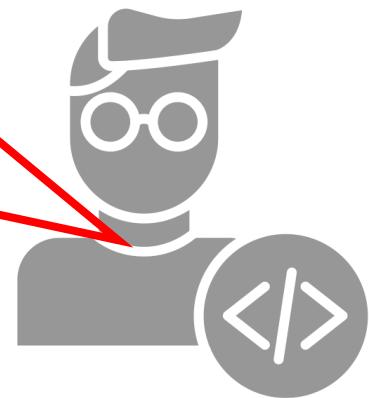
Sorry, uninspiring at all
yeah . . . let's go back to the
scrambled one you did
before!

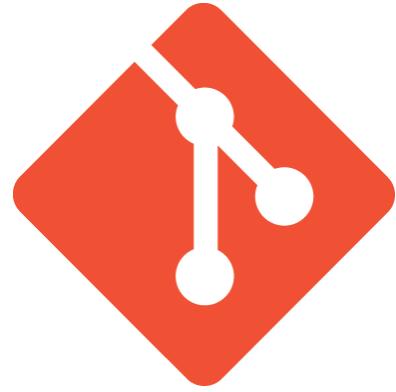


That
was
PERFECT



HOW? !?

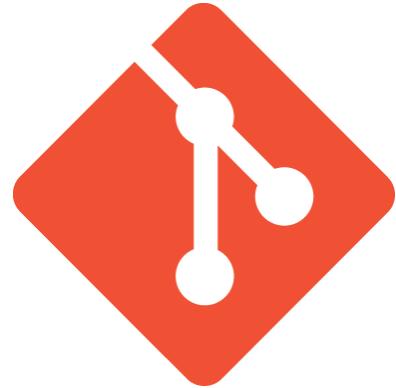




git



POLITECNICO
MILANO 1863



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

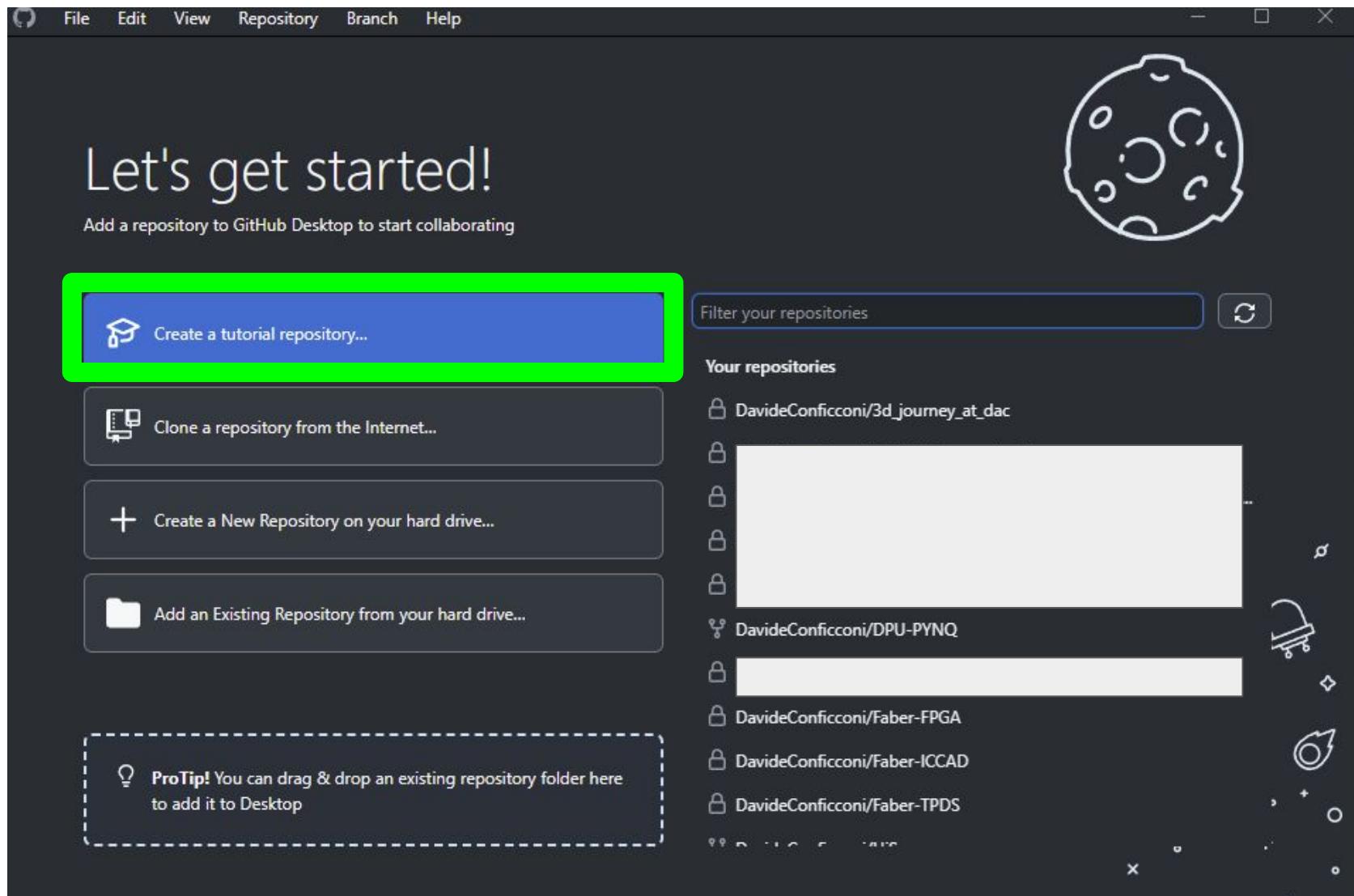
git



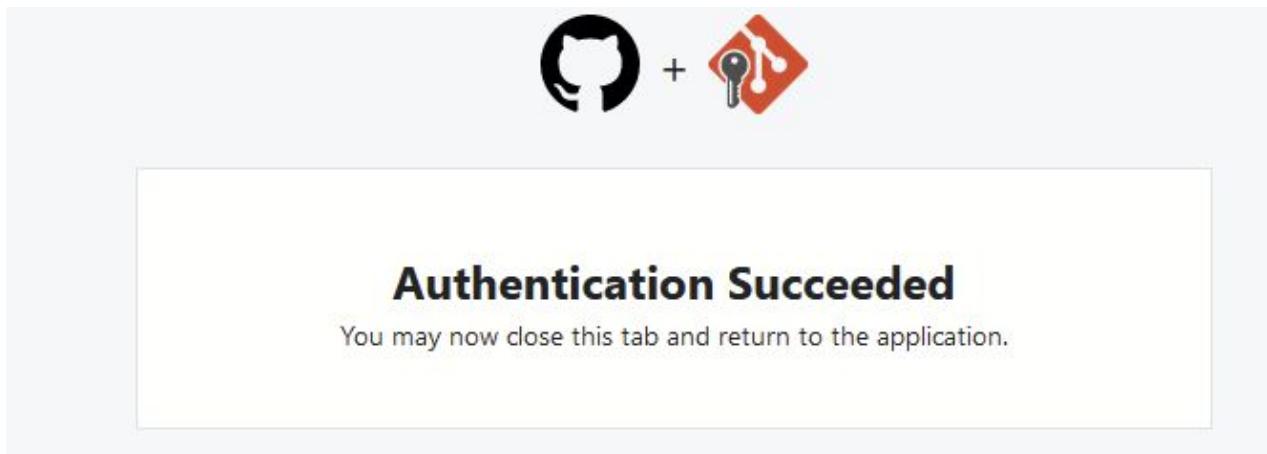
Demo Time :)



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

The screenshot shows the GitHub Desktop application interface. At the top, the menu bar includes File, Edit, View, Repository, Branch, and Help. The repository dropdown shows "Current repository desktop-tutorial" and the branch dropdown shows "Current branch main". A status bar at the top right indicates "Fetch origin Last fetched 2 minutes ago".

The main area features a large blue upward arrow icon above the text "Welcome to GitHub Desktop". Below this, a sub-header reads "Use this tutorial to get comfortable with Git, GitHub, and GitHub Desktop." To the left, there's a sidebar with tabs for "Changes" (selected) and "History", and a section showing "0 changed files".

The central content area contains three main sections:

- Git** (represented by a terminal icon): "Git is the version control system."
- GitHub** (represented by a people icon): "GitHub is where you store your code and collaborate with others."
- GitHub Desktop** (represented by a cloud icon): "GitHub Desktop helps you work with GitHub locally."

To the right, a "Get started" section provides a numbered tutorial:

- 1 **Create a branch**: "A branch allows you to work on different versions of a repository at one time. Create a branch by going into the branch menu in the top bar and clicking "New branch"." (Hotkey: `Ctrl + Shift + N`)
- 2 **Edit a file**
- 3 **Make a commit**
- 4 **Publish to GitHub**
- 5 **Open a pull request**

At the bottom left, there's a "Summary (required)" section with a "Description" input field containing placeholder text "What's new? Add a description of your changes here." and a "Commit to main" button.



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 and 0 tags. The main commit is from 'DavideConfigconi' titled 'Initial commit' at 3edd951, 17 minutes ago. The README.md file contains the text 'Welcome to GitHub Desktop!'. The repository has 1 commit, 0 stars, 1 watching, 0 forks, and no releases published. There are also sections for Packages and Activity.

DavideConfigconi / 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

DavideConfigconi 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

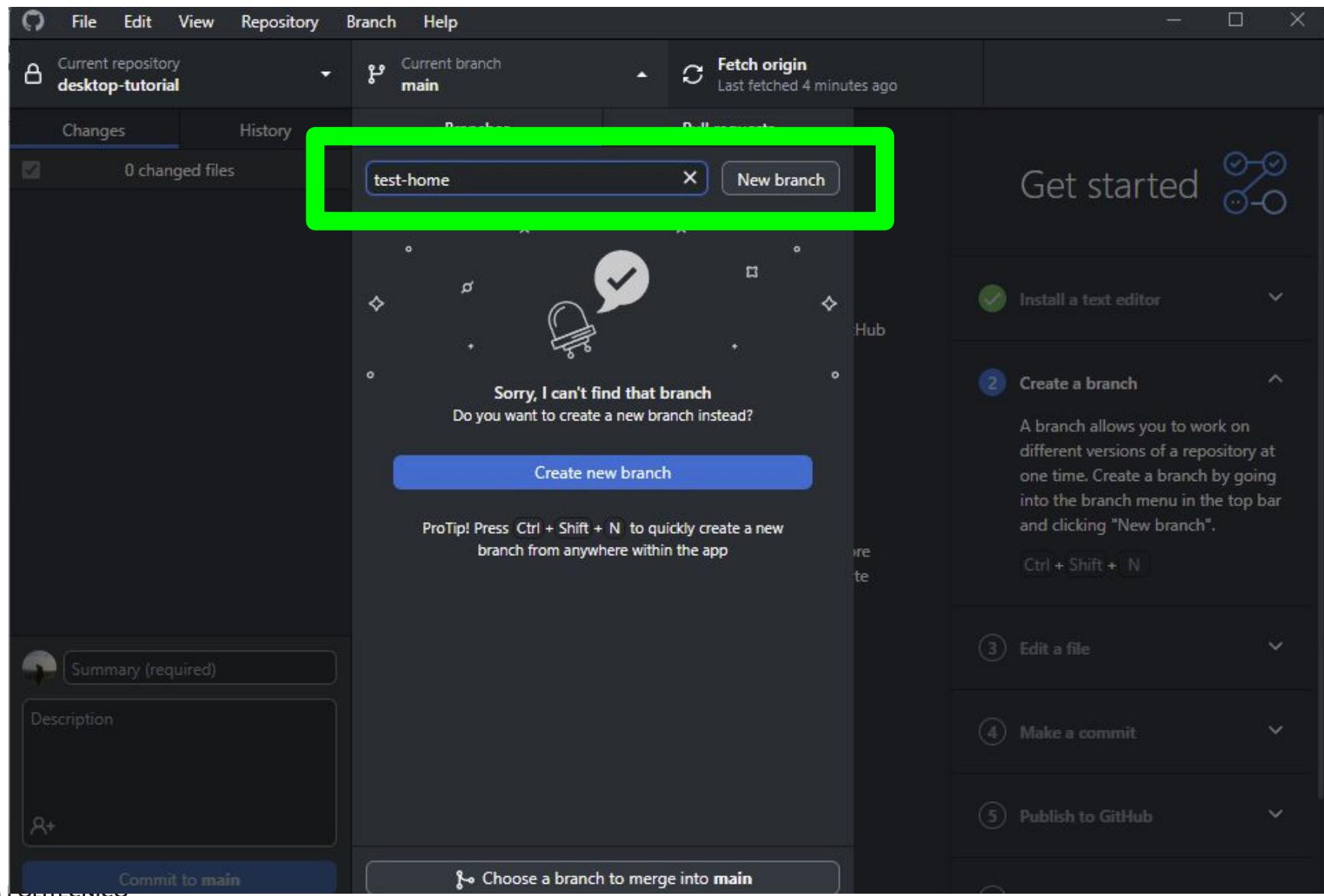
No releases published Create a new release

Packages

No packages published Publish your first package



GitHub Desktop New Branch



GitHub Desktop Local Modify

The screenshot shows the GitHub Desktop application interface. The top right corner features the GitHub logo. The main area displays a code editor with the file `README.md`. The content of the file is as follows:

```
desktop-tutorial > 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 bottom half of the screen shows the "Changes" tab in the "Changes" section of the interface. It indicates "1 changed file" and shows the change for `README.md`:

Line	Change
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 status bar at the bottom left shows "MILANO 1863".

GitHub Desktop Commit

The screenshot shows the GitHub Desktop application interface. At the top, the menu bar includes File, Edit, View, Repository, Branch, and Help. The current repository is "desktop-tutorial" and the current branch is "test-home". A "Changes" tab is selected, showing one changed file, "README.md". The main pane displays the content of "README.md":

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

A green box highlights the commit message area. Inside this box, the text reads:

Usually commit message is structured as “[TOPIC] details on the update”

Below this, a blue box highlights the commit dialog window. It contains fields for "Title" (labeled "[DOC] Add my name") and "Description", and a "Commit to test-home" button.

The right side of the interface shows a "Get started" sidebar with sections for "Create a branch", "Edit a file", "Make a commit", "Publish to GitHub", and "Open a pull request".

GitHub Desktop Push

The screenshot shows the GitHub Desktop application interface. At the top, the menu bar includes File, Edit, View, Repository, Branch, and Help. The current repository is "desktop-tutorial" and the branch is "test-home". In the center, the "Changes" tab is selected, showing one changed file, "README.md". The code editor displays the README content with a new line added at line 6: "+Davide Conficconi". A green callout box highlights the "Publish branch" button in the top right corner of the window. To the right, a "Get started" sidebar provides steps for using GitHub Desktop, including "Install a text editor", "Create a branch", "Edit a file", "Make a commit" (which is currently active), "Publish to GitHub", and "Open a pull request".

Publish branch
Publish this branch to GitHub

Get started

- Install a text editor
- Create a branch
- Edit a file
- 4 Make a commit
- 5 Publish to GitHub
- 6 Open a pull request



github.com Sync

The screenshot shows a GitHub repository page for the 'desktop-tutorial' repository. At the top, there's a banner indicating 'test-home' had recent pushes less than a minute ago. Below this, the repository details show 2 branches and 0 tags. A green 'Code' button is highlighted. The main content area displays a commit by 'DavideConfigconi' adding a name to the README. The README itself contains a welcome message and instructions for contributing, with the latter part highlighted by a large green box.

Recent activity: test-home had recent pushes less than a minute ago

Compare & pull request

Go to file Add file Code

This branch is 1 commit ahead of main.

DavideConfigconi [DOC] Add my name

9e82258 1 minute ago 2 commits

README.md [DOC] Add my name 1 minute 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. Davide Configconi

About

GitHub Desktop tutorial repository

Readme Activity 0 stars 1 watching 0 forks

Releases

No releases published Create a new release

Packages

No packages published Publish your first package





Git Summary





Git Summary

Create, edit, delete file(s)





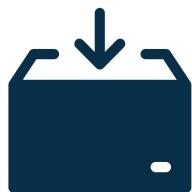
Git Summary

**Create, edit, delete
file(s)**

Compare the changes

git status
git diff





Git Summary

**Create, edit, delete
file(s)**

Compare the changes

git status
git diff

**Stage the changes for
commit**

git add *file-name*





Git Summary



**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"`





Git Summary

**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"`

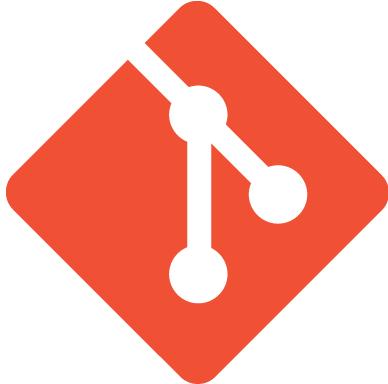
Keep the history in check



`git log`

Ok, but why do we need this?



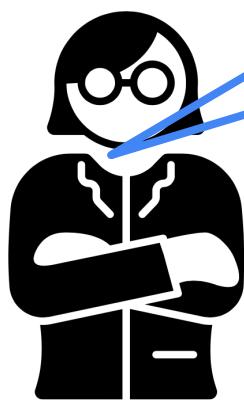


Single developer

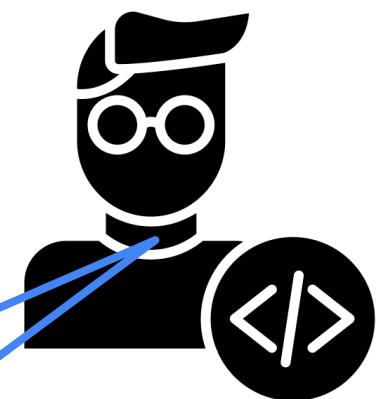
- Tracks evolution
- Builds history
- Navigate the history

git



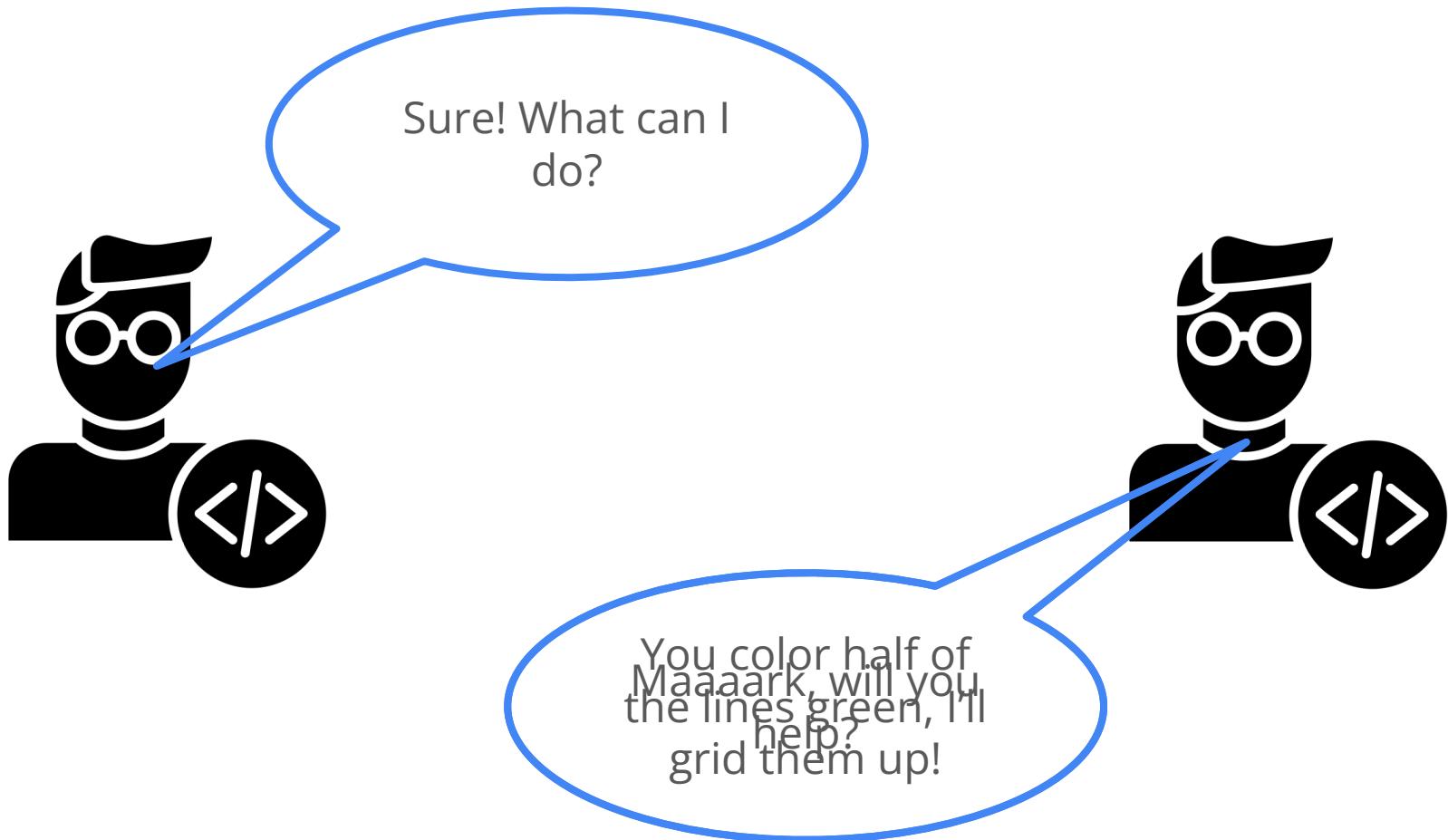


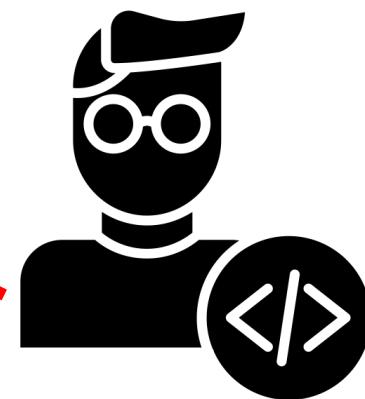
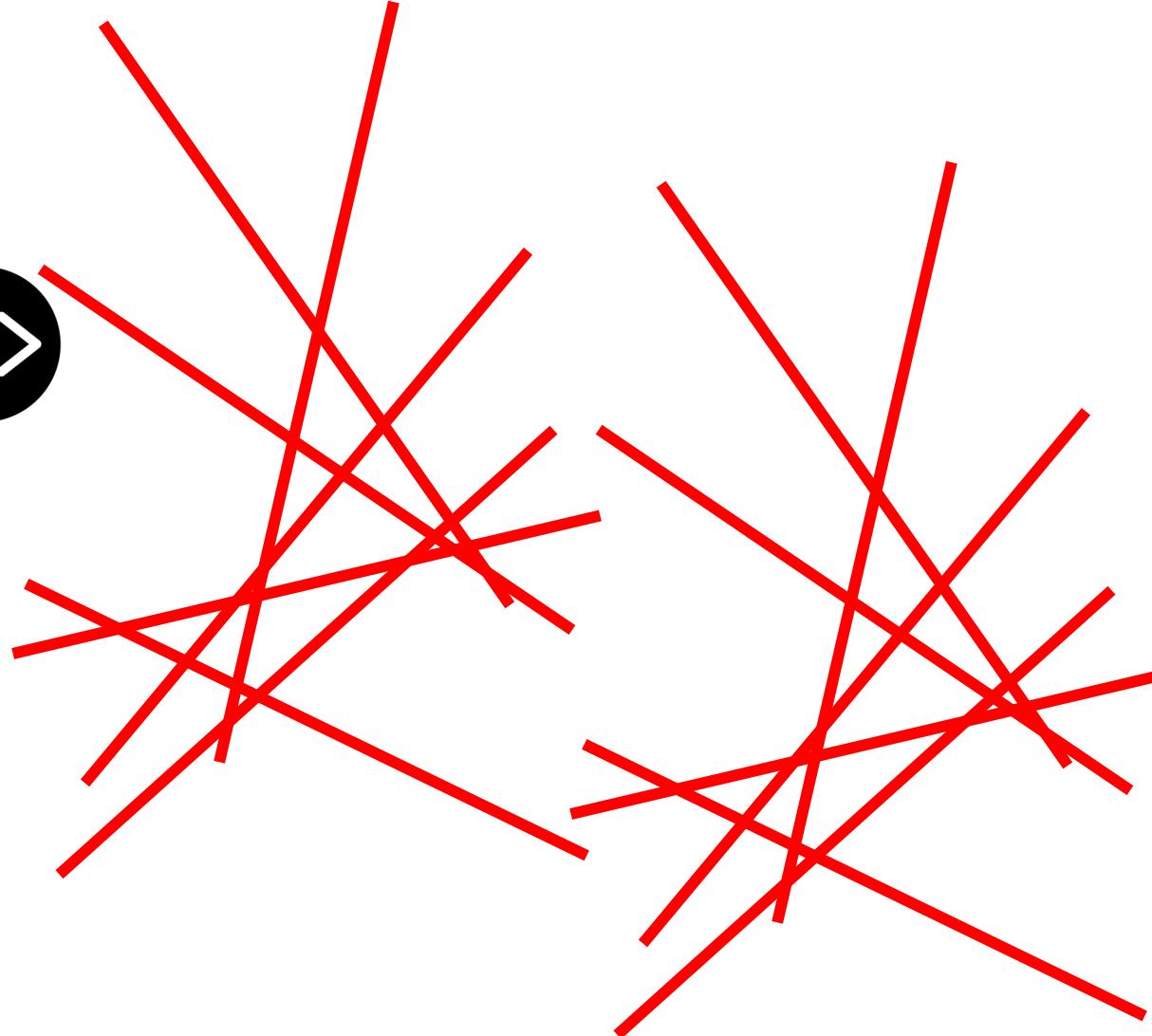
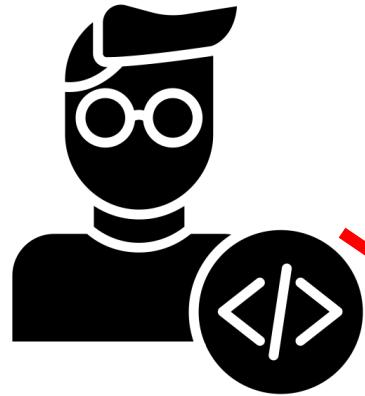
New feat req!
We need all the lines to
be perpendicular and
half of them should be
green!

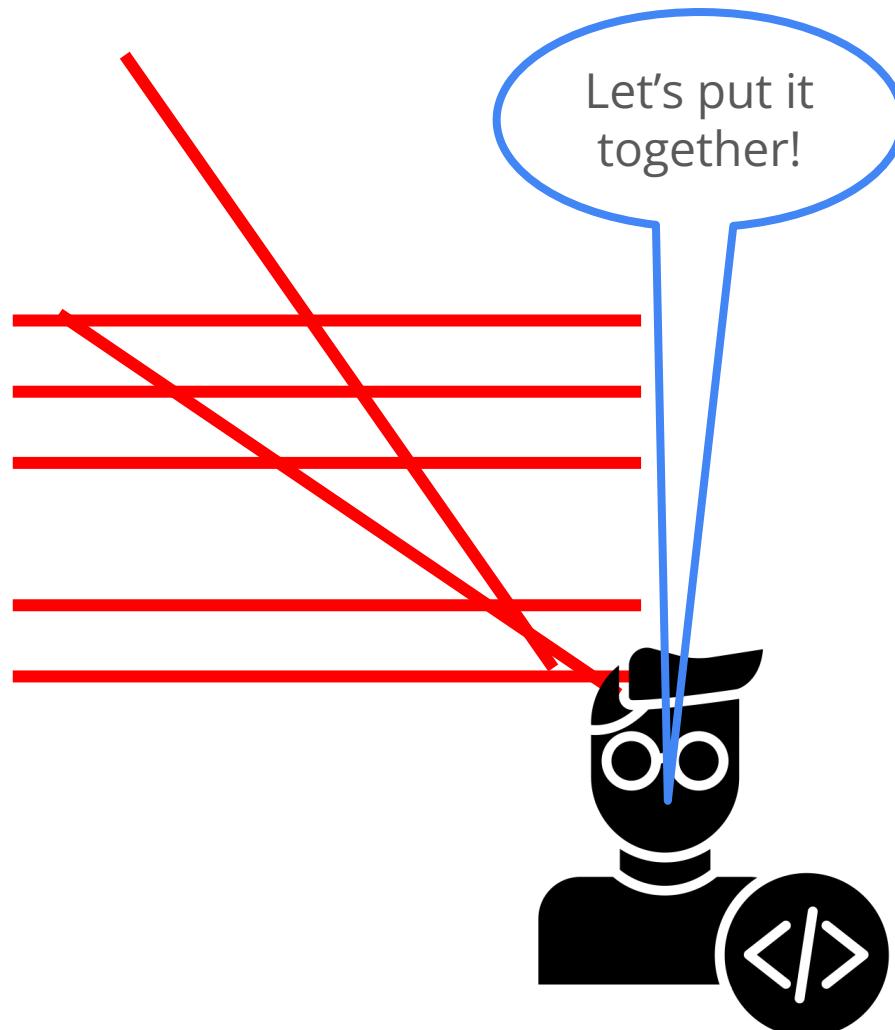
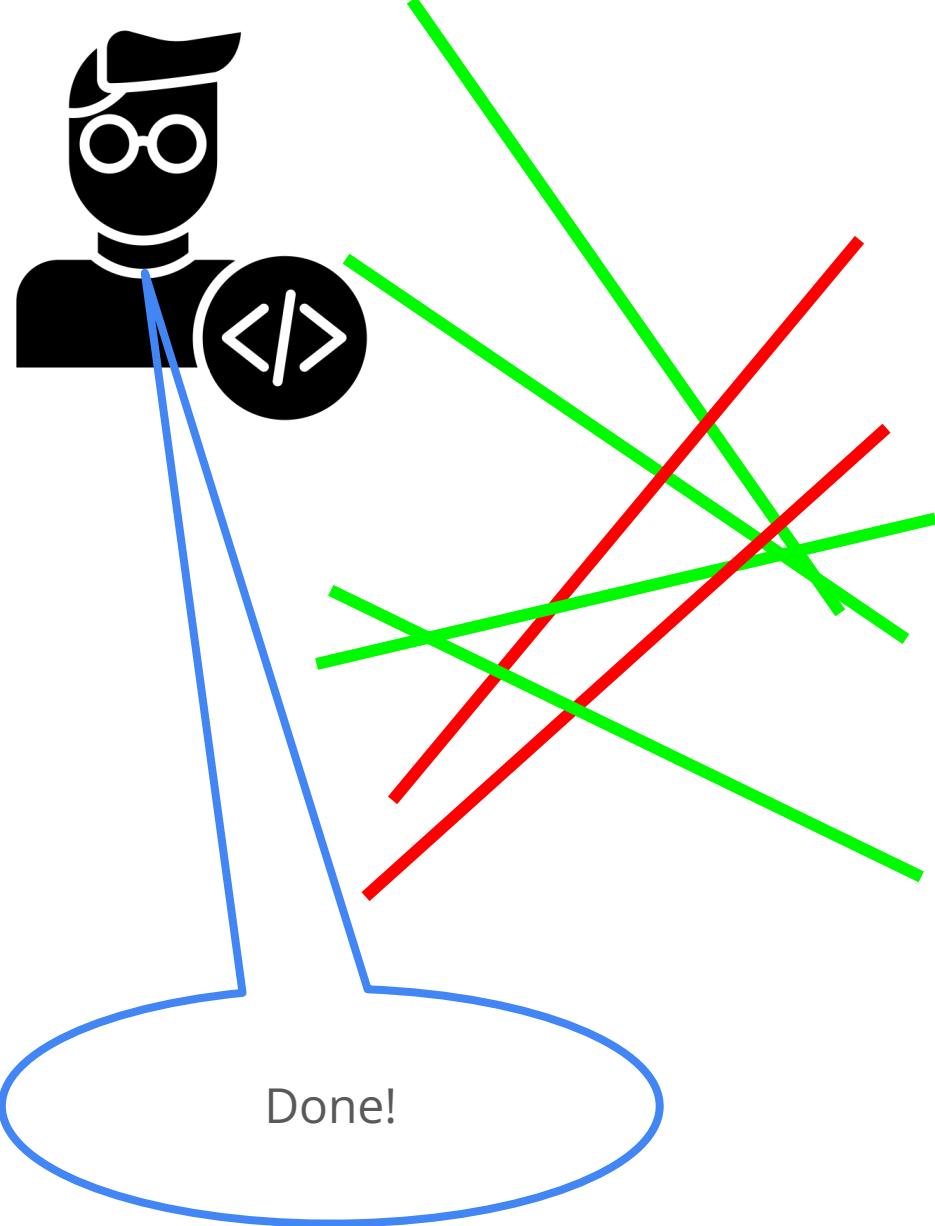


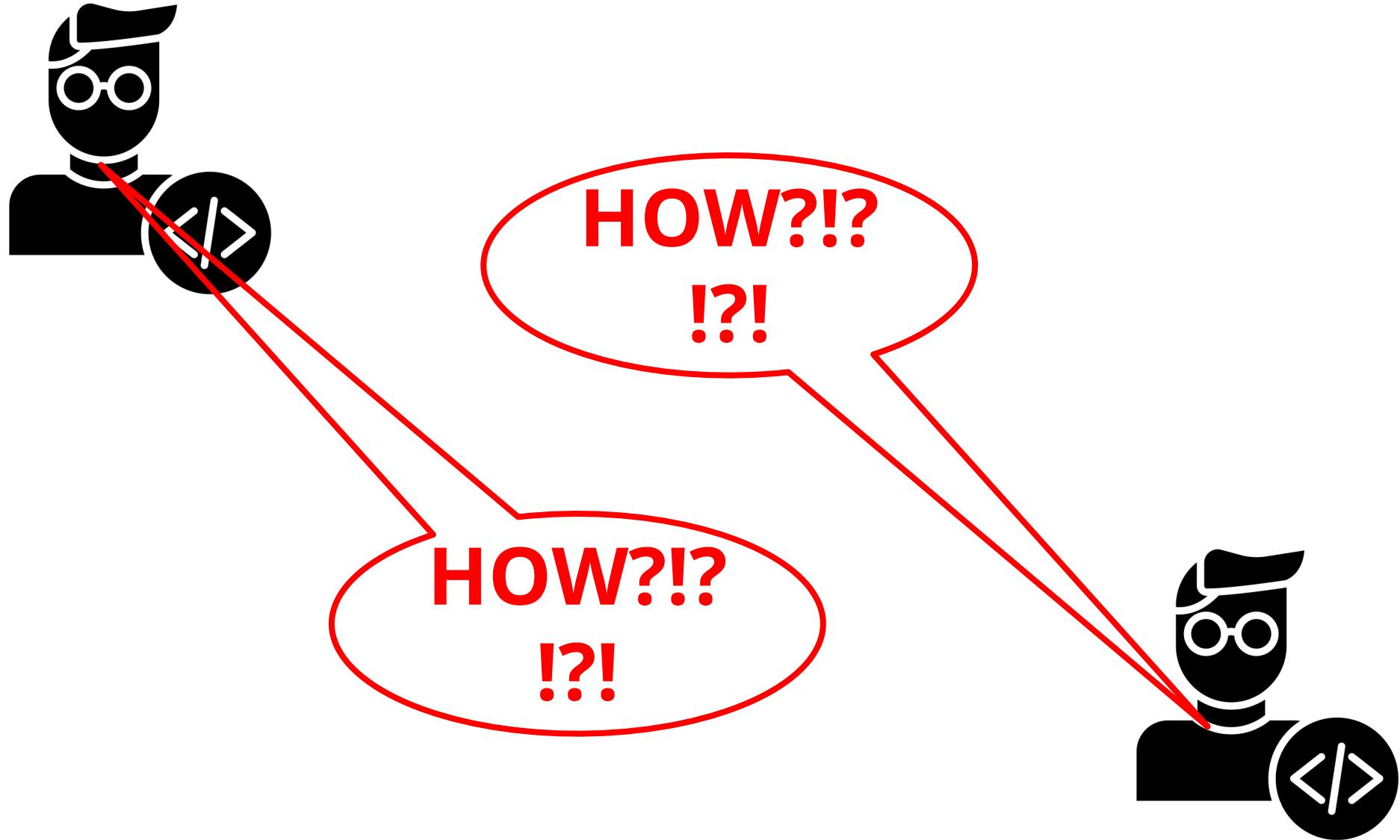
I can't make it on my
own!

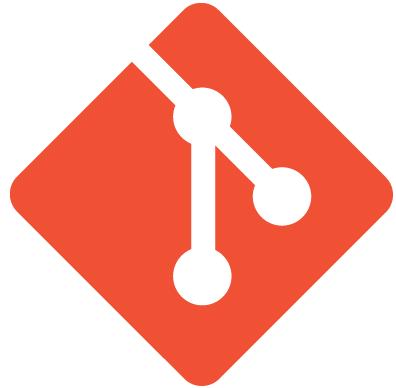








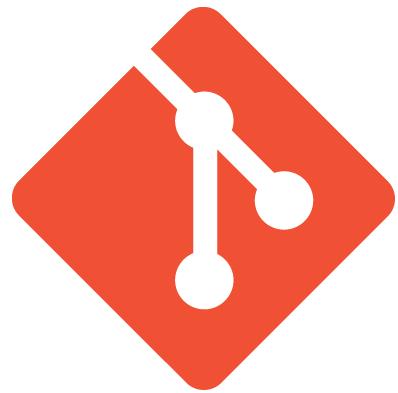




git



POLITECNICO
MILANO 1863



remote



local
devA

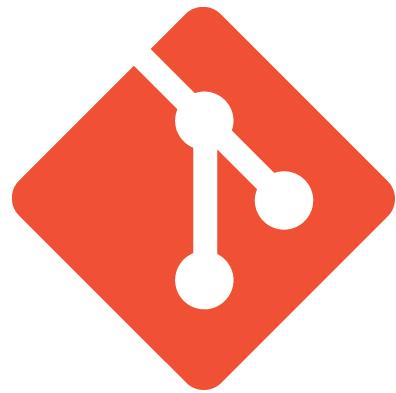


local
devB



git





remote



local
devA

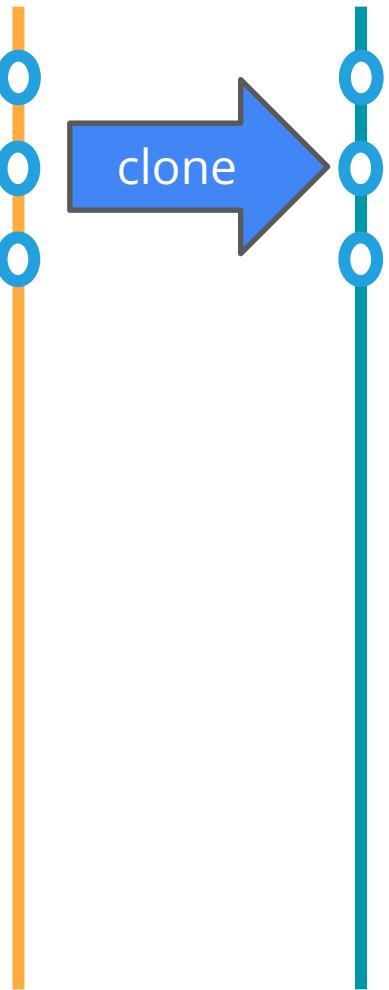


local
devB

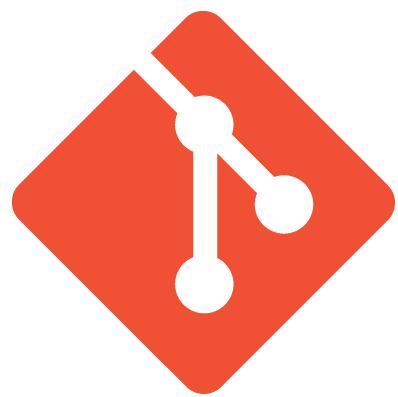


clone

git



POLITECNICO
MILANO 1863



remote



local
devA



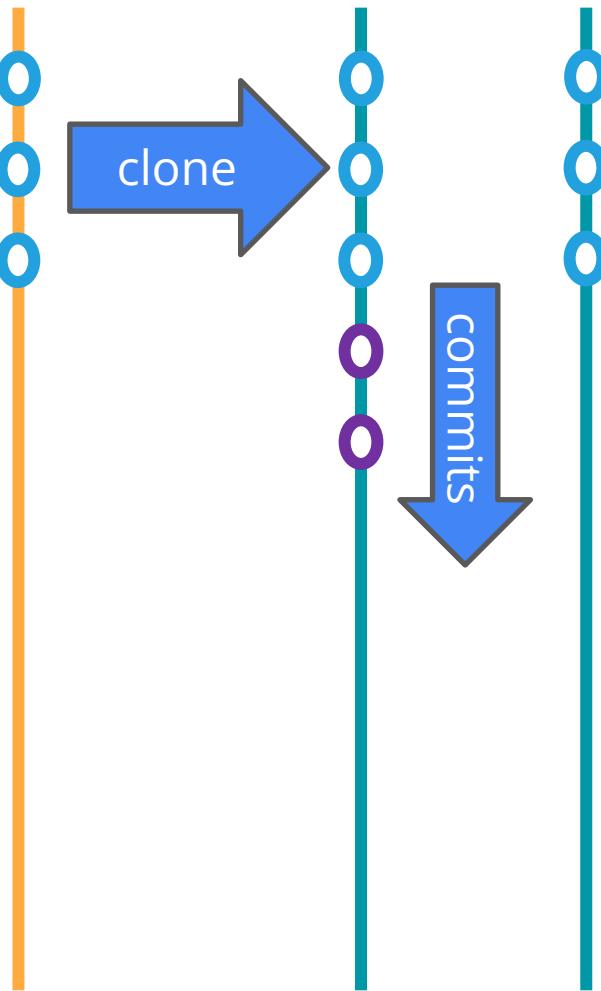
local
devB

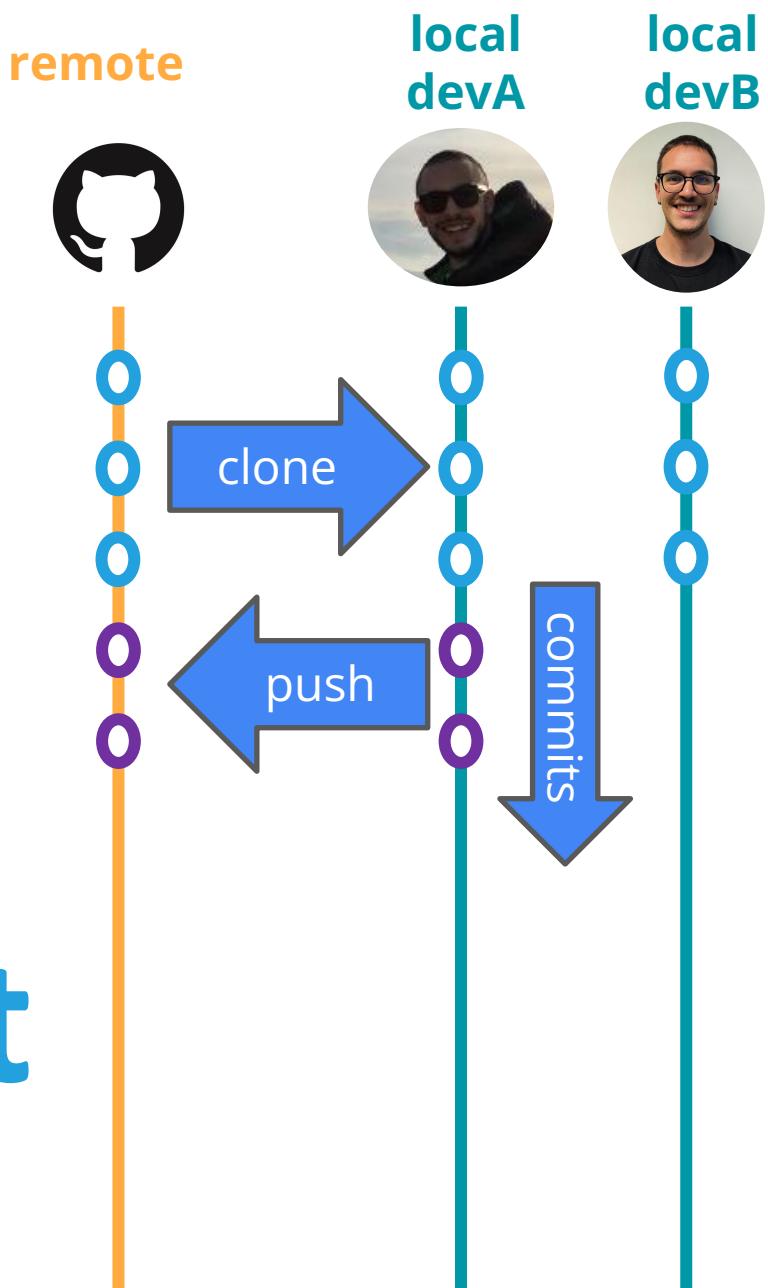
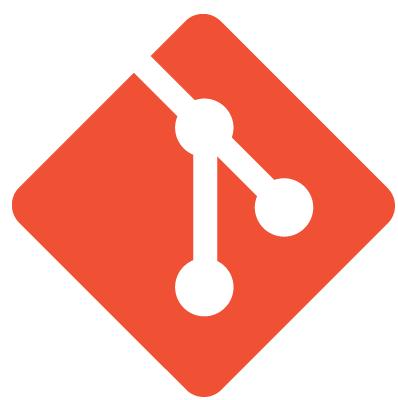


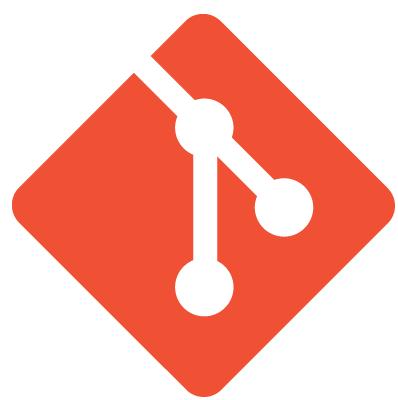
clone

commits

git







remote



local
devA

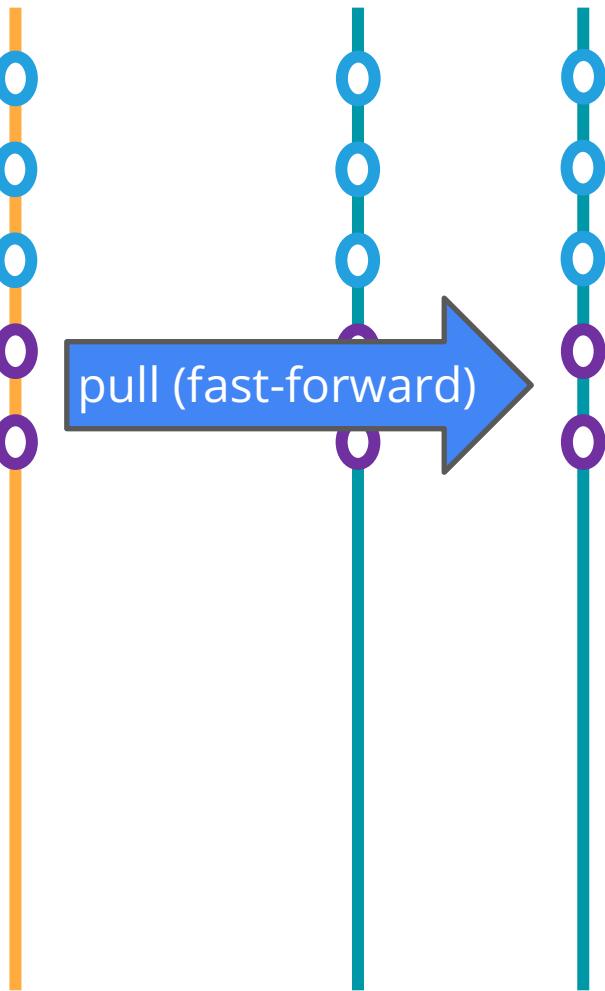


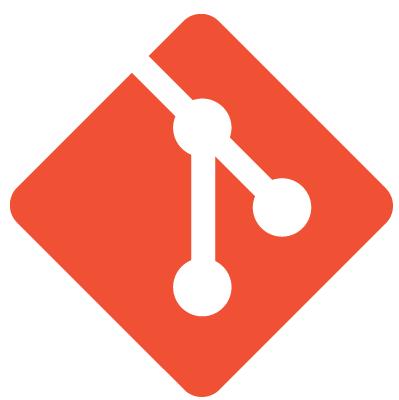
local
devB



git

pull (fast-forward)





remote



git

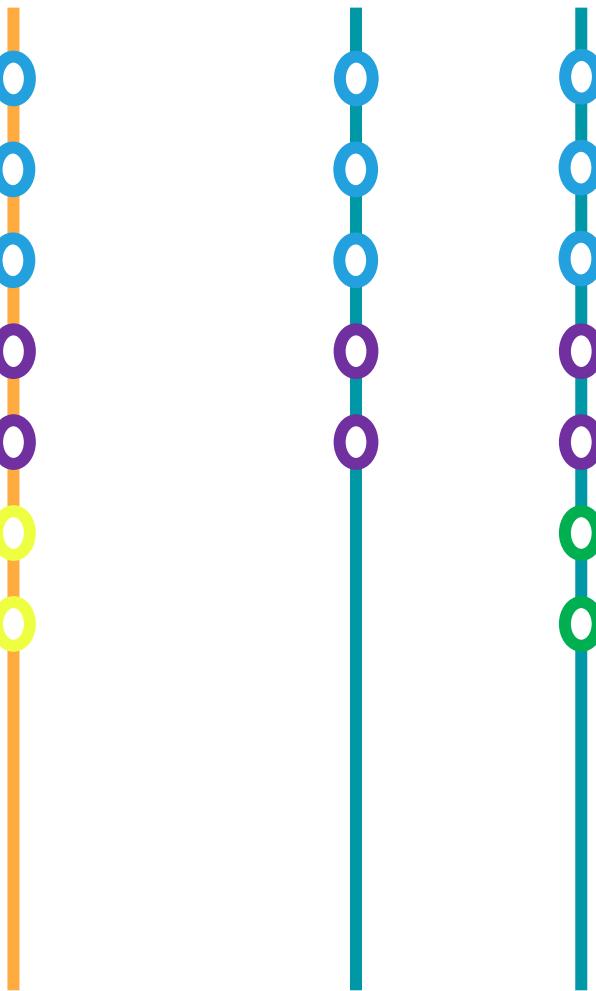
local
devA

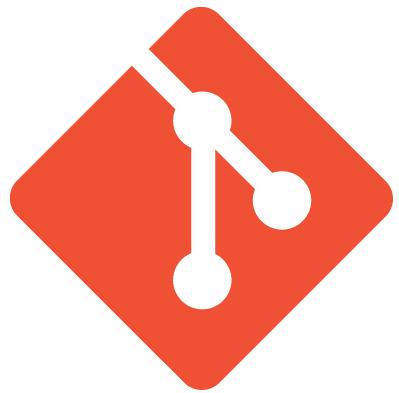


local
devB



local
devC





remote



git

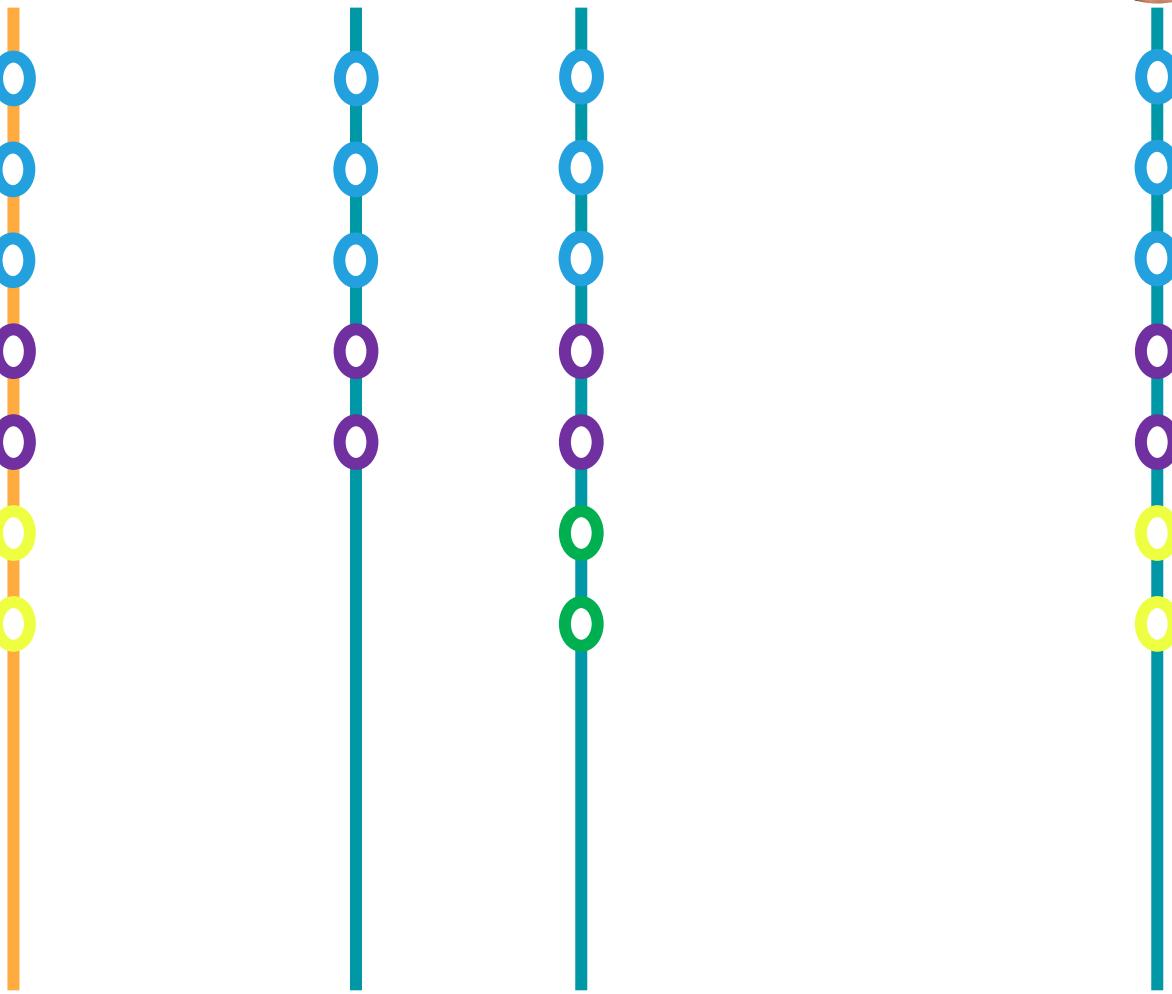
local
devA

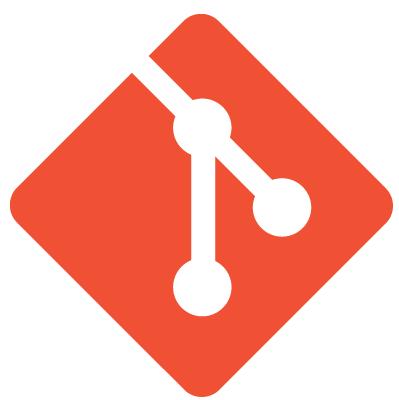


local
devB



local
devC





git

remote



local
devA



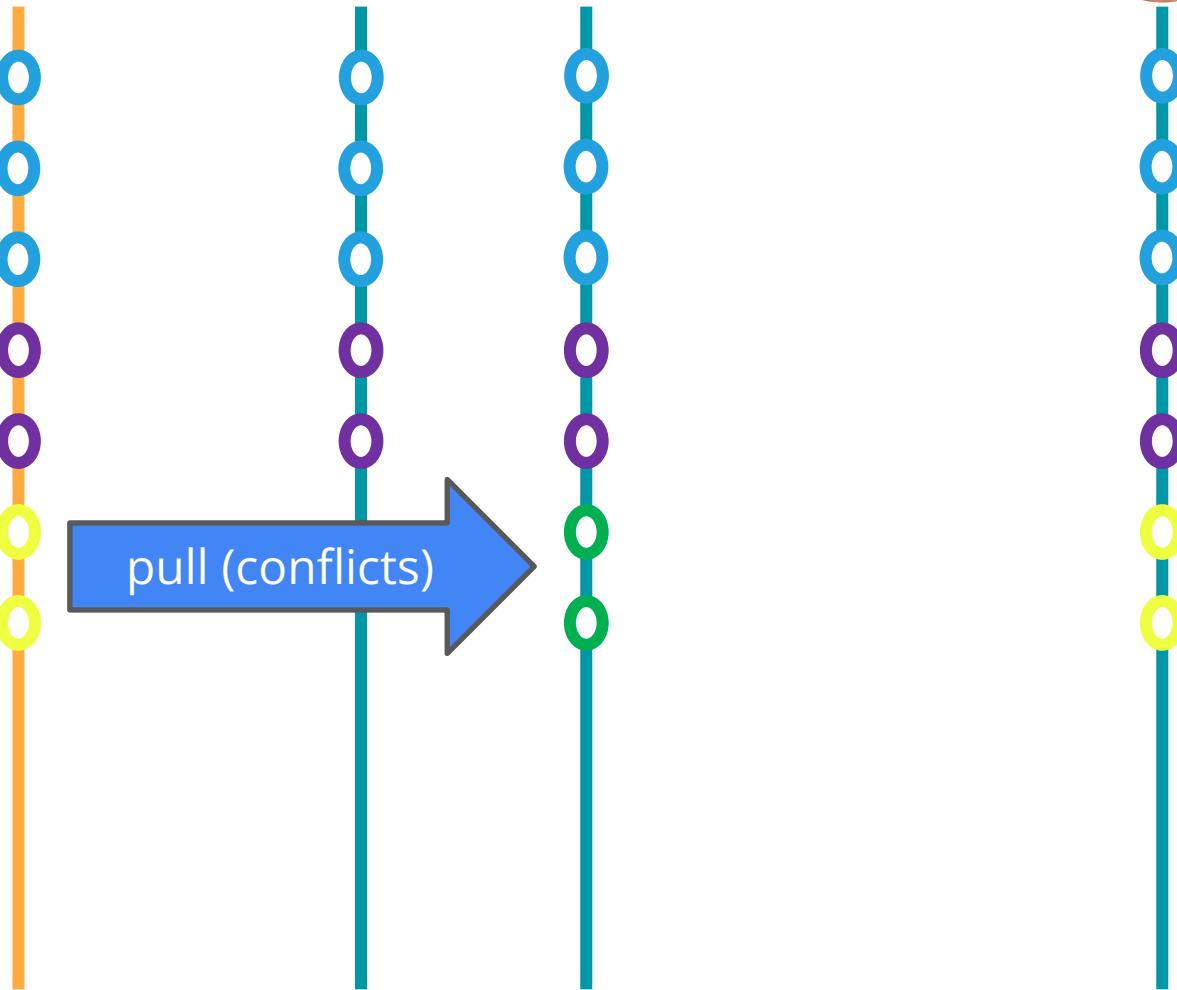
local
devB

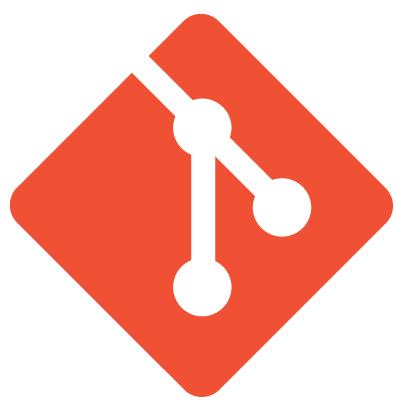


local
devC



pull (conflicts)





git

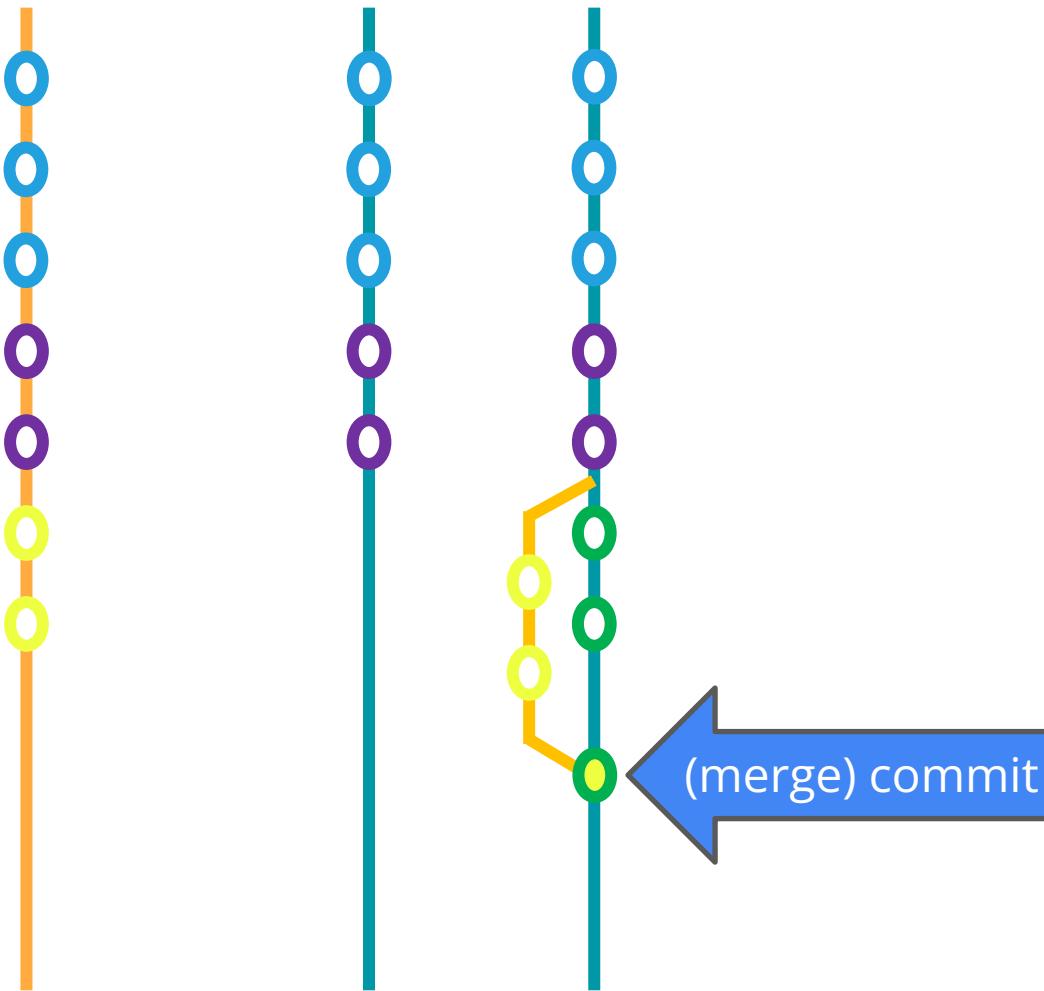
remote



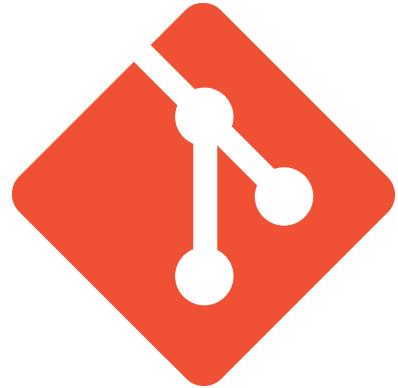
local
devA



local
devB



POLITECNICO
MILANO 1863



remote



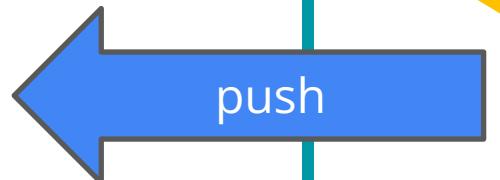
local
devA



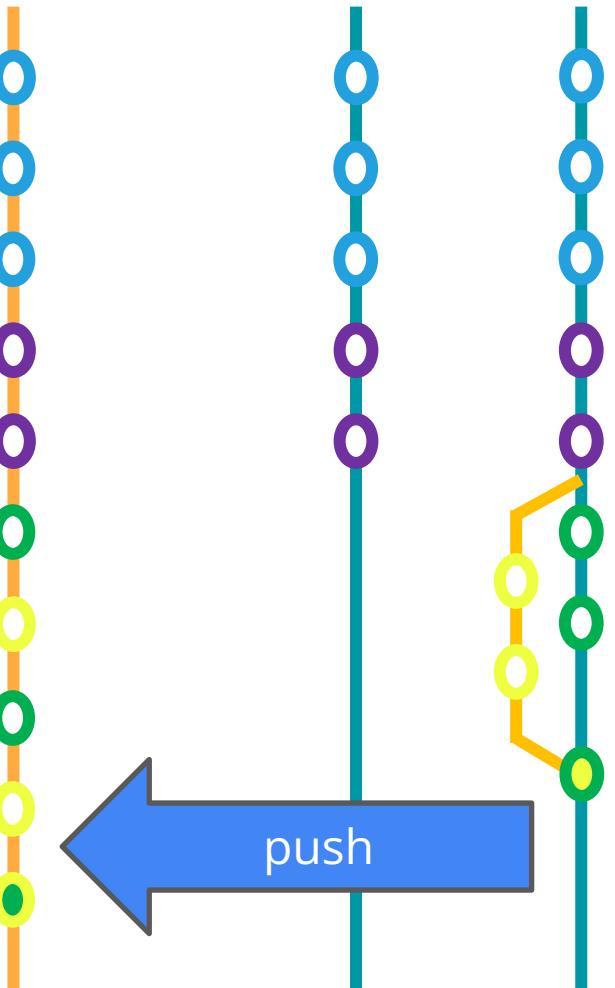
local
devB

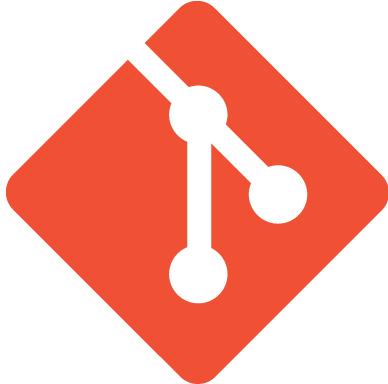


git



push





git

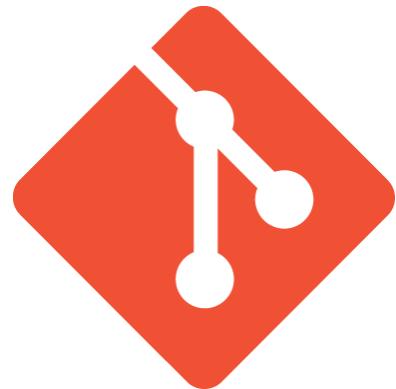
Single developer

- Tracks evolution
- Builds history
- Navigate the history

Team of developers

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

Tool?



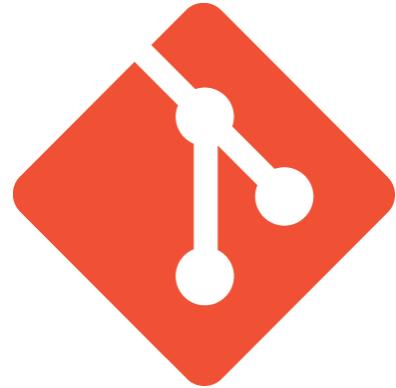
Method?

git

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



POLITECNICO
MILANO 1863



Method

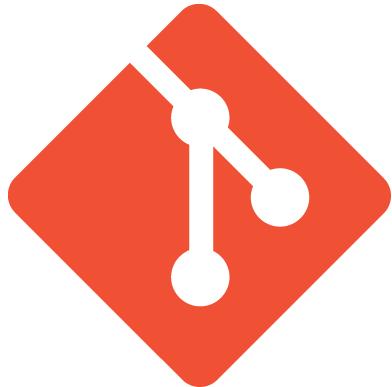
git Flow

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

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



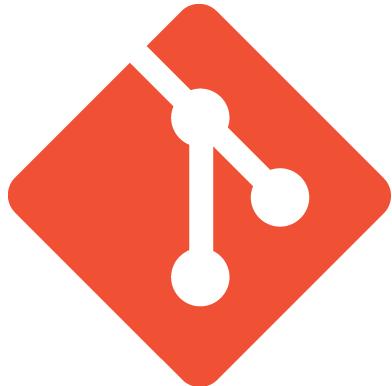
Advanced Topics



git

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

Other GUIs



Free
Win+Linux+mac
Os

Integrates with
all the major VCS
cloud providers

Free
Win+macOs

Direct cloning
from BitBucket

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

Integration with
advanced
versioning
features

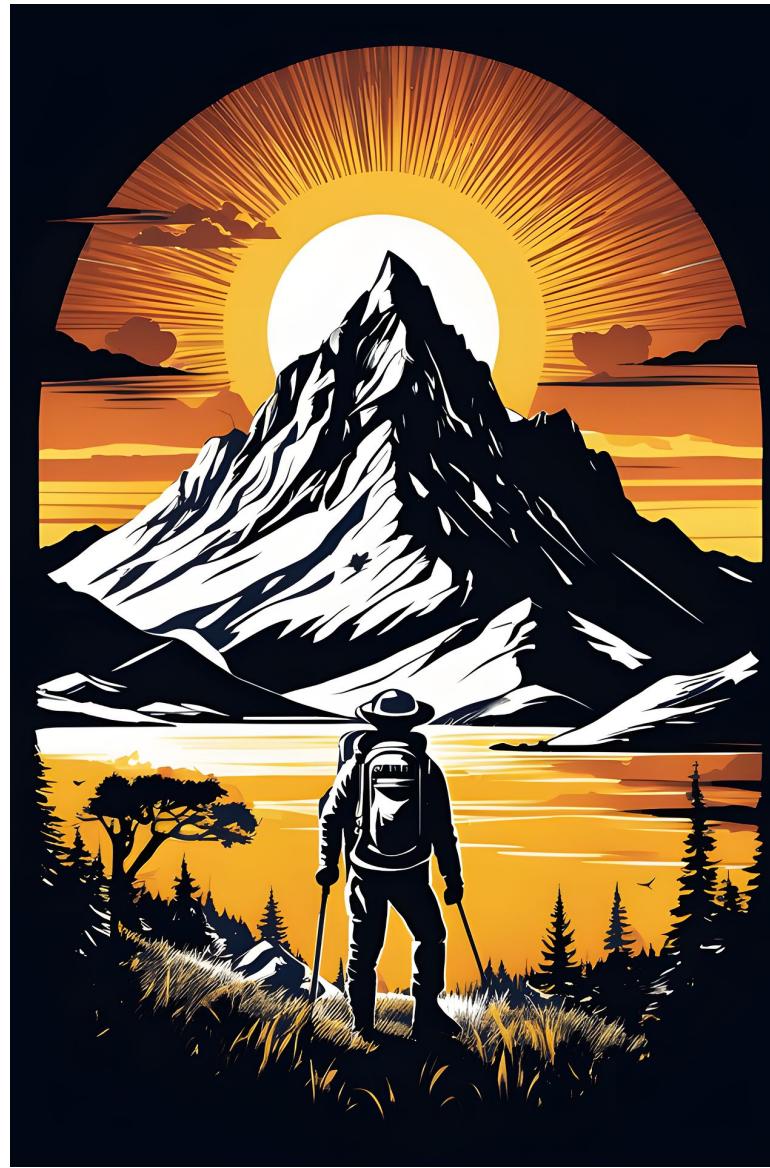
git



Cool...but why?



To Find the Saint Graal



POLITECNICO
MILANO 1863

Image by [Chil Vera](#) from [Pixabay](#)

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](#))

jekyll



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**

jekyll



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!!!

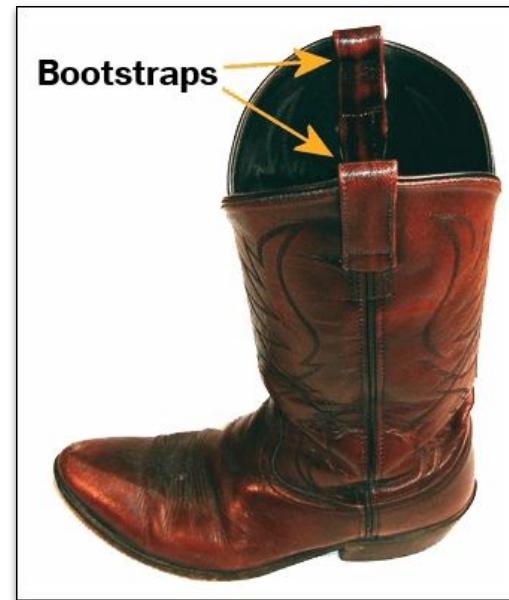


Let's publish our first website!

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>

The screenshot displays a website template named "Personal" from Start Bootstrap. The main header "Start Bootstrap" is at the top left, with a navigation bar including "Home", "Resume", "Projects", and "Contact". On the left, there's a sidebar with the text "DESIGN · DEVELOPMENT · MARKETING" and "I can help your business to **Get online and grow fast**". Below this are two buttons: "Resume" (blue) and "Projects" (white). The central feature is a large, rounded square placeholder for a profile picture, featuring a smiling man with glasses and a beard. At the bottom of this section are "About Me" and "My name is Start Bootstrap and I help brands grow." Below the main content are "Live Preview" and "View Source Code" buttons.

Personal
Personal is a modern, clean website theme perfect for professionals, creators, or businesses.

The IT Monitoring Platform
Ultra-scalable, automated, and easily extensible IT monitoring. Start free Checkmk trial.

Checkmk

Download

Free Download

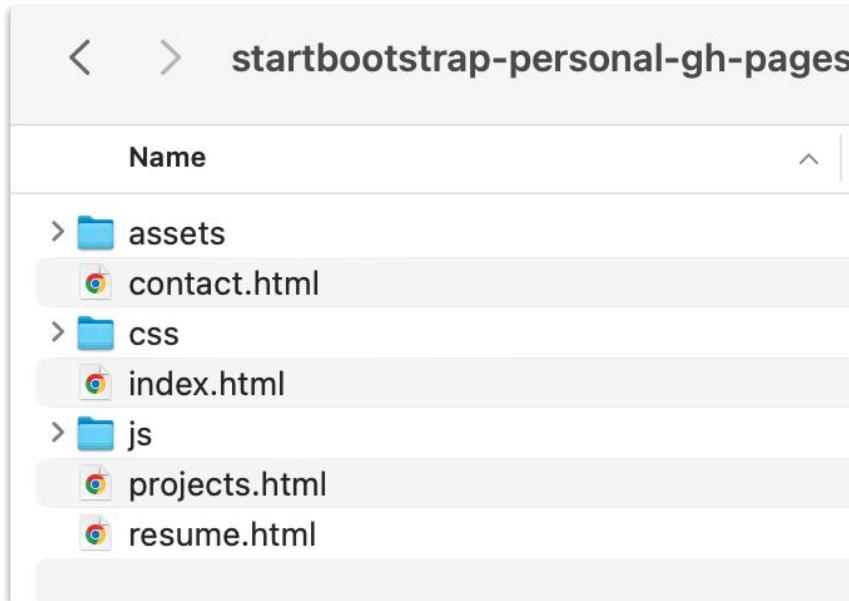
License MIT License

Downloaded 2,407



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

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

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

- 1)
- 2)
- 3)
- 4)
- 5)
- 6)
- 7)



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!

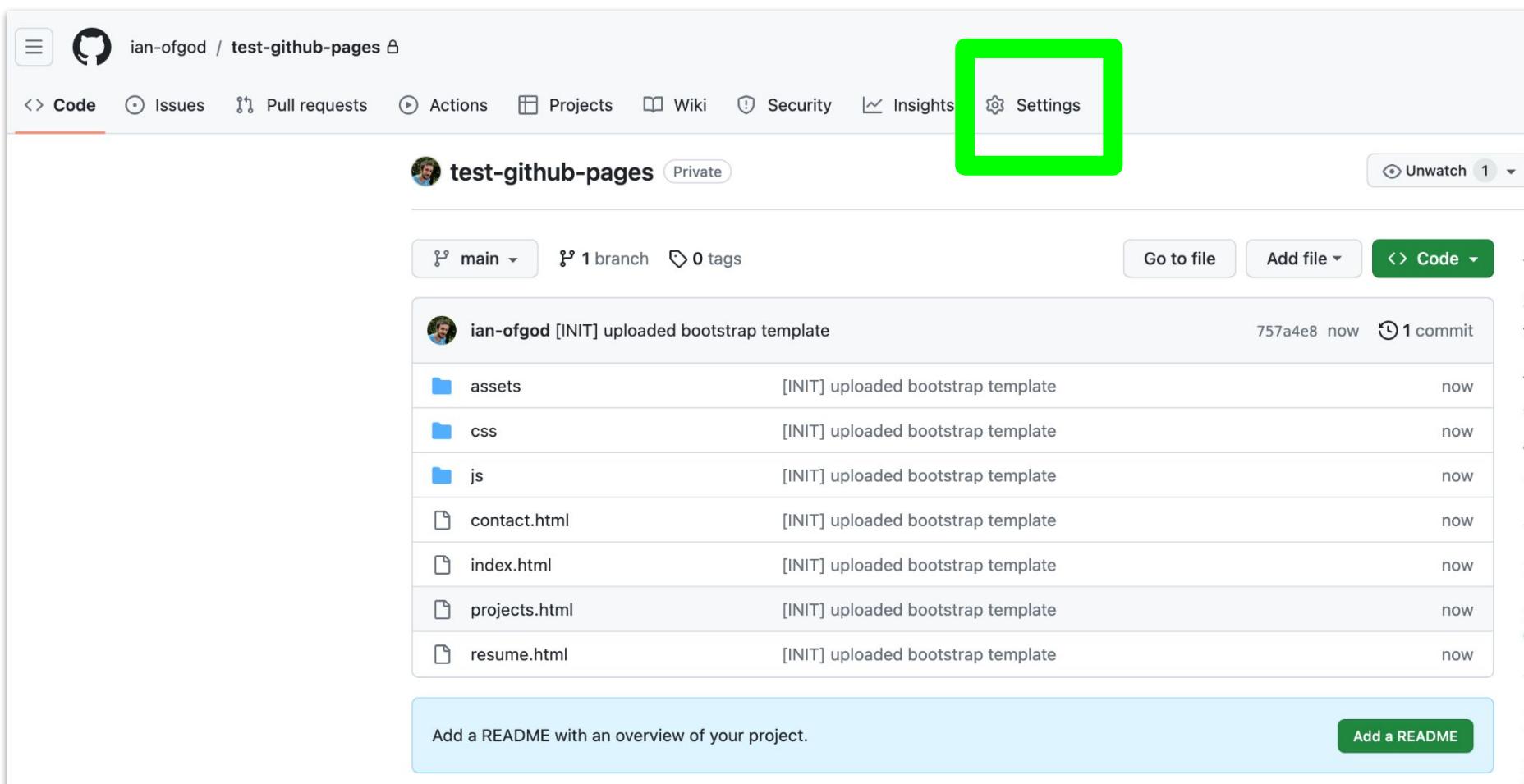
HOW?



Demo Time :)



Enabling GitHub pages



The screenshot shows a GitHub repository page for 'ian-ofgod / test-github-pages'. The 'Settings' tab is highlighted with a green box. The repository has 1 branch and 0 tags. A commit history is shown, with the first commit being '[INIT] uploaded bootstrap template' by 'ian-ofgod' at 757a4e8. The commit details show files like assets, css, js, contact.html, index.html, projects.html, and resume.html were uploaded. At the bottom, there's a call to action to 'Add a README'.

ian-ofgod / test-github-pages 

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

 test-github-pages Private 

 main  1 branch  0 tags Go to file Add file Code

 ian-ofgod [INIT] uploaded bootstrap template 757a4e8 now 1 commit

File	Description	Time
assets	[INIT] uploaded bootstrap template	now
css	[INIT] uploaded bootstrap template	now
js	[INIT] uploaded bootstrap template	now
contact.html	[INIT] uploaded bootstrap template	now
index.html	[INIT] uploaded bootstrap template	now
projects.html	[INIT] uploaded bootstrap template	now
resume.html	[INIT] uploaded bootstrap template	now

Add a README with an overview of your project. 



Enabling GitHub pages

The screenshot shows the GitHub repository settings for '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, etc. The 'Pages' section is highlighted with a green box. In the main area, there's a 'General' section with fields for Repository name (set to 'test-github-pages') and a 'Rename' button. There are also two checkboxes: 'Template repository' (unchecked) and 'Require contributors to sign off on web-based commits' (unchecked). Below that is a 'Default branch' section where the default branch is set to 'main'. The right side has a 'Features' section with checkboxes for Wikis (checked), Restrict editing to collaborators only (checked), and Issues (checked).

ian-ofgod / test-github-pages

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

General

Access

Collaborators

Code and automation

Branches

Tags

Rules

Actions

Webhooks

Environments

Pages

Security

Code security and analysis

Deploy keys

Secrets and variables

Integrations

GitHub Apps

Email notifications

Autolink references

Repository name

test-github-pages

Rename

Template repository

Template repositories let users generate new repositories with the same directory structure and files. [Learn more about template repositories](#).

Require contributors to sign off on web-based commits

Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Contributors to affirm that their commit complies with the repository's terms, commonly the [Developer Certificate of Origin](#). [Learn more about signing off on commits](#).

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests are automatically made, unless you specify a different branch.

main

Features

Wikis

Wikis host documentation for your repository.

Restrict editing to collaborators only

Issues

Issues integrate lightweight task tracking into your repository. Keep projects on track with issue labels and reference them in commit messages.



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 (Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces, Pages), Security (Code security and analysis, Deploy keys, Secrets and variables), and Integrations (GitHub Apps, Email notifications, Autolink references). The 'Pages' category is selected and highlighted with a blue bar.

The main content area is titled 'GitHub Pages'. It explains that GitHub Pages is designed to host personal, organization, or project pages from a GitHub repository. A section titled 'Build and deployment' contains a 'Source' dropdown set to 'Deploy from a branch'. Below it is a 'Branch' dropdown with 'None' selected. A 'Select branch' modal is open, showing a list with 'main' selected. A green rectangular highlight surrounds the 'main' branch entry in the modal. To the right of the modal, there is descriptive text about publishing privately and a link to learn more about visibility. At the bottom of the modal, there are 'Save' and 'Cancel' buttons.



Enabling GitHub pages

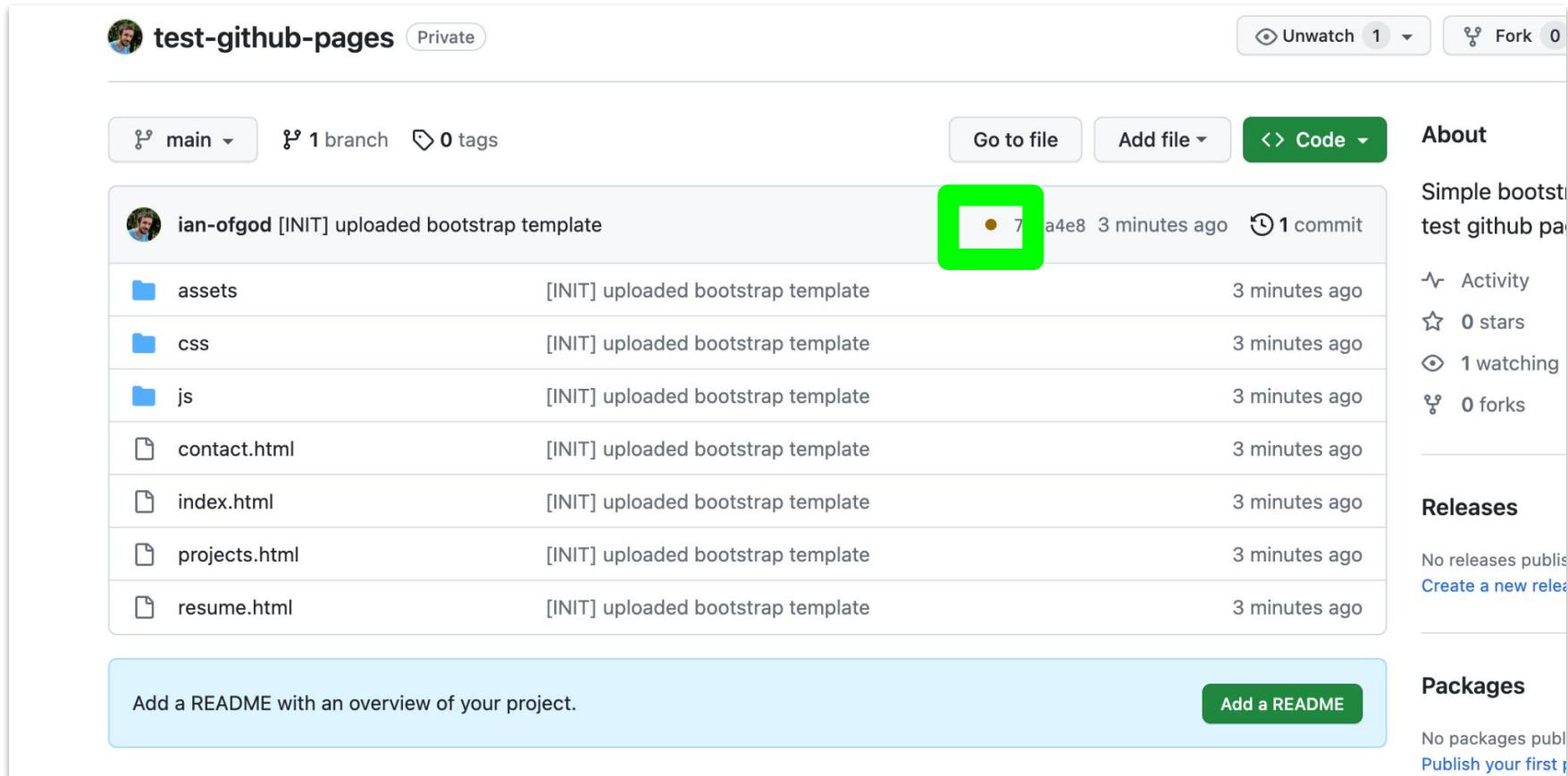
The screenshot shows the GitHub repository settings interface for enabling GitHub Pages. On the left, a sidebar lists various settings categories: General, Access, Collaborators, Code and automation (Branches, Tags, Rules, Actions, Webhooks, Environments, Codespaces), Pages (selected), Security (Code security and analysis, Deploy keys, Secrets and variables), and Integrations (GitHub Apps, Email notifications, Autolink references). The main content area is titled "GitHub Pages" and contains the following sections:

- GitHub Pages**: A brief description stating "GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository."
- Build and deployment**: A section for configuring the publishing source.
- Source**: A dropdown menu set to "Deploy from a branch".
- Branch**: A dropdown menu currently set to "main".
- Save**: A button to save the configuration changes, highlighted with a green box.
- Visibility**: A section for restricting access to the GitHub Pages site.
 - A "GITHUB ENTERPRISE" badge is present.
 - A note: "With a GitHub Enterprise account, you can restrict access to your GitHub Pages site by publishing it privately. Privately published site can only be accessed by people with read access to the repository the site is published. You can use privately published sites to share your internal documentation or knowledge base with members enterprise."
 - A "Try GitHub Enterprise risk-free for 30 days" button.
 - A link: "Learn more about the visibility of your GitHub Pages site"



Check build status

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



The screenshot shows a GitHub repository page for "test-github-pages". The repository is private. At the top, there are buttons for "Unwatch" (1), "Fork" (0), and "Code". Below the header, it shows "main" branch, 1 branch, 0 tags, and buttons for "Go to file", "Add file", and "Code". On the right, there's an "About" section with a brief description: "Simple bootstrap test github pages". It also shows activity stats: 1 commit, 0 stars, 1 watching, and 0 forks. The main content area lists files and their upload status: "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), and "resume.html" [INIT] uploaded bootstrap template (3 minutes ago). A green box highlights the commit status indicator for the first file. At the bottom, there's a callout to add a README with a "Add a README" button.

File	Status	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



Check build status

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

The screenshot shows a GitHub repository interface. At the top, it displays 'main' branch, 1 branch, 0 tags, and various navigation buttons like 'Go to file', 'Add file', and 'Code'. A green box highlights the 'Code' button.

The main content area lists files and their upload status:

File	Status	Last Updated
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	3 minutes ago
projects.html	[INIT] uploaded bootstrap template	3 minutes ago
resume.html	[INIT] uploaded bootstrap template	3 minutes ago

Below the file list, there's a note to 'Add a README with an overview of your project.' and a 'Add a README' button.

A central sidebar displays build status checks:

- Some checks haven't completed yet (1 successful and 2 queued checks)
- ✓ pages build and deployment / build (dynamic) Success (Details)
- pages build and deployment / report-build-status (Details)
- pages build and deployment / deploy (dynamic) Queued (Details)

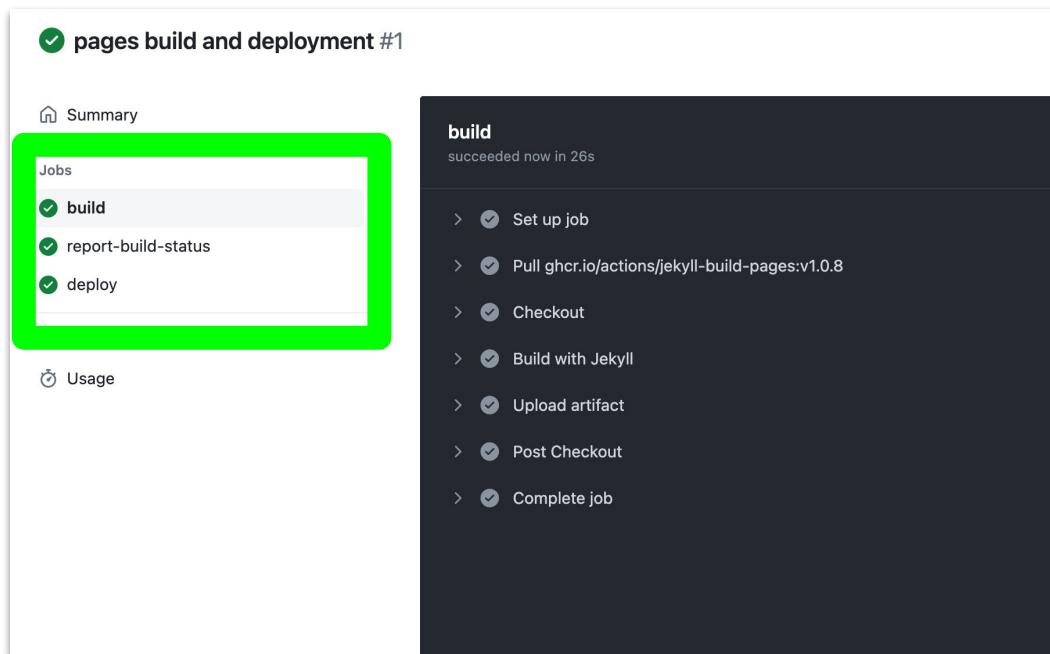
Other sections include 'Releases' (No releases published, Create a new release) and 'Packages' (No packages published, Publish your first package).



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)



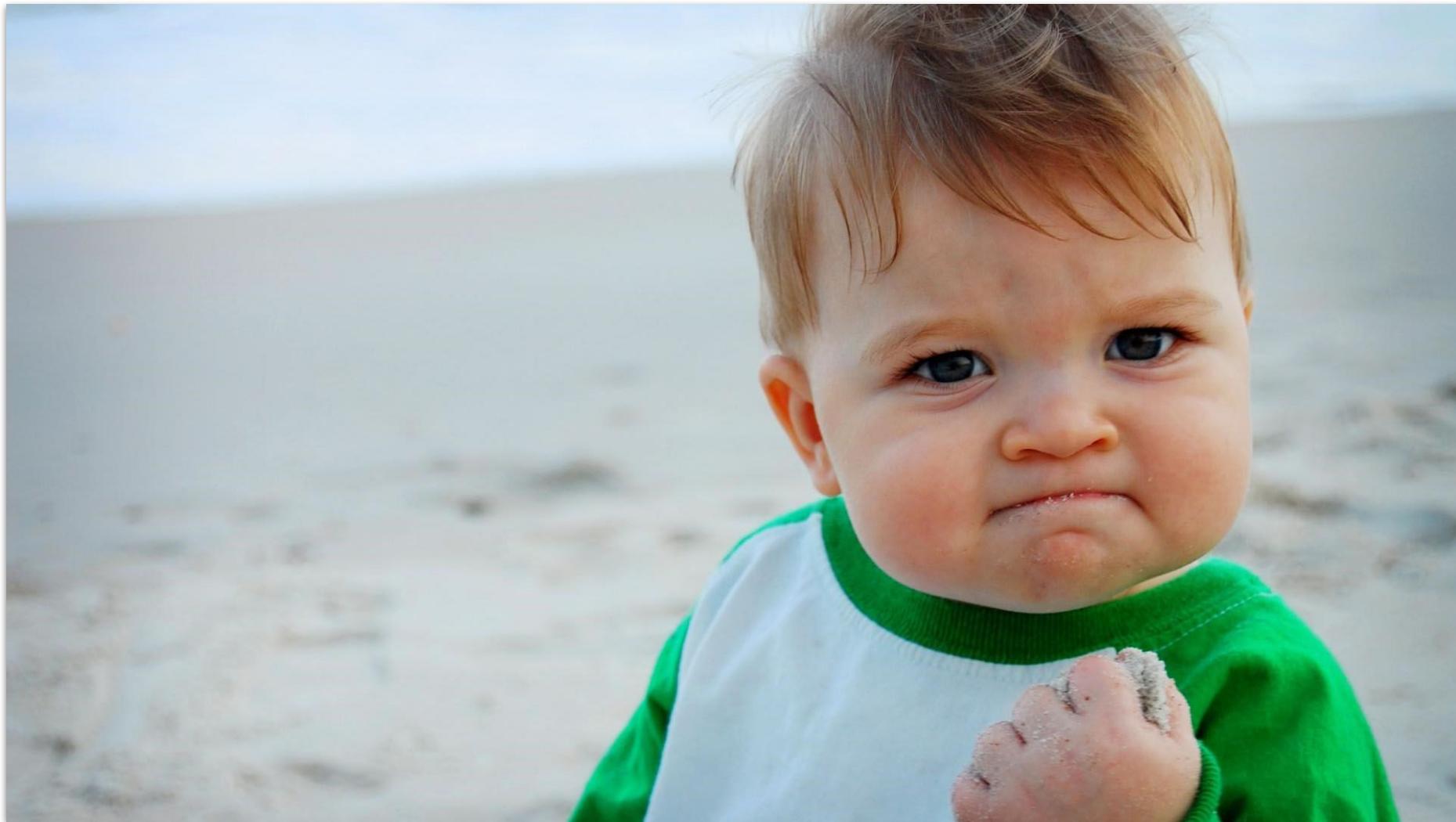
Enabling GitHub pages

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

The screenshot shows a professional website template. At the top left is a logo for "Start Bootstrap". At the top right are navigation links: Home, Resume, Projects, and Contact. Below the header, there's a large, rounded rectangular image of a smiling Black man with glasses and a beard, set against a blue-to-purple gradient background with white dots. To the left of the image, the text "I can help your business to" is followed by a large, bold, purple text block that reads "Get online and grow fast". Above this main text, there's a small purple button with the words "DESIGN · DEVELOPMENT · MARKETING". The overall design is clean and modern.



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

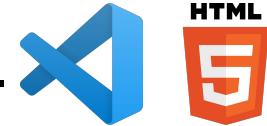
- 1) [Github pages Docs](#)
- 2) <https://docs.github.com/en/pages/getting-started-with-github-pages/creating-a-github-pages-site#next-steps>
- 3) [Bootstrap reference](#)
- 4) [From markdown to github pages](#)



Today Objectives

1. ~~Informatica Self Assessment~~

2. Local dev with VSCode, Indice html for IA2425



3. Create our Webpage with Githubpages



4. Draw with P5, P5 Live Editor

p5*js



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



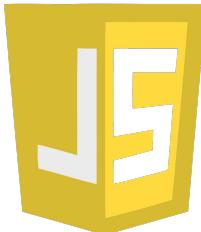
What is P5? <https://hello.p5js.org/>

p5*.js



What is P5? <https://hello.p5js.org/>

JavaScript



p5*.js



POLITECNICO
MILANO 1863

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?

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



Why p5.js?

Make programming interactive graphics easy

```
const width = 928; const height = 500; const marginTop = 30; const marginRight = 30; const marginBottom = 30; const marginLeft = 30;
// Set the scale (horizontal position) scale.
const x = d3.scaleBand()
    .domain(d3.groupSort(newData, ((d) => -d.frequency, (d) => d.letter))) // descending frequency
    .range([marginLeft, width - marginRight])
    .padding(0.1);
// Decide the (vertical position) scale.
const y = d3.scaleLinear()
    .domain([0, d3.max(newData, (d) => d.frequency)])
    .range([height - marginBottom, marginTop]);
// Create the SVG container.
const svg = d3.create("svg").attr("width", width).attr("height", height)
    .attr("viewBox", [0, 0, width, height])
    .attr("style", "max-width: 100%; height: auto;");
// Add a rect for each bar.
svg.append("g").attr("fill", "#steelblue")
    .selectAll().data(newData).join("rect")
        .attr("x", (d) => x(d.letter))
        .attr("y", (d) => y(d))
        .attr("width", x.bandwidth())
        .attr("height", (d) => y(0) - y(d));
// Add the x-axis and label.
svg.append("g").attr("transform", `translate(0, ${height - marginBottom})`)
    .call(d3.axisBottom(x).ticks(x.bandwidth()));
// Add the y-axis and label, and remove the domain line.
svg.append("g").attr("transform", `translate(${marginLeft}, 0)`)
    .call(d3.axisLeft(y).ticks(10).tickFormat((y) => (y * 100).toFixed()))
    .call((g) => g.selectAll("text").remove());
    .attr("x", -marginLeft).attr("y", 20)
    .attr("fill", "currentColor").attr("text-anchor", "start")
    .attr("font-size", "20px").text(`Frequency (h)`));
    .attr("font-size", "20px");
svg.selectAll("text")
    .attr("font-size", "20px");
```



Why p5.js?

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

Immediate feedback with few lines of code

```
const width = 928; const height = 500; const marginTop = 20; const marginBottom = 30; const marginLeft = 10; const marginRight = 10; // Set the horizontal position scale. d3.scaleBand().domain([0, 100]).range([marginLeft, marginRight]); // Sort the data. d3.groupSort(newData, ((i) => -i.frequency, (i) => i.letter)).sort((a, b) => a.letter < b.letter); // Set the vertical position scale. const y = d3.scaleLinear().domain([0, 100]).range([height - marginBottom, marginTop]); // Create the SVG container. const svg = d3.create("svg").attr("width", width).attr("height", height).attr("viewBox", [0, 0, width, height]).attr("style", "max-width: 100%; height: auto!"); // Add a rect for each bar. svg.append("g").attr("fill", "#f9c232").selectAll().data(newData).enter().append("rect").attr("y", (i) => y(i.letter) - y(i.frequency)).attr("height", (i) => y(i.letter) - y(i.frequency)).attr("width", x.bandwidth()); // Add the x-axis label. svg.append("g").attr("transform", `translate(0, ${height - marginBottom})`).call(d3.axisBottom(x).ticks(x.ticksCount(0))); // Add ticks and label, and remove the domain line. svg.append("g").attr("transform", `translate(${marginLeft}, 0)`).call(d3.axisLeft(y).tickFormat((i) => (y * 100).toFixed(1))).attr("x", -marginLeft).attr("y", 20).attr("fill", "currentColor").attr("text-anchor", "start").attr("font-size", "20").text(`Frequency (h)`)); svg.selectAll(".tick text").attr("font-size", "0.8");
```



Why 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 repository for p5.js. At the top, there are links for 'README', 'Code of conduct', and 'LGPL-2.1 license'. Below that, there are badges for 'npm package 1.10.0', 'all contributors 714', and 'downloads 1.3M'. The main content area features the title 'p5.js' and a 'Welcome!' message with hand icons. A paragraph describes p5.js as a free and open-source JavaScript library for accessible creative coding, emphasizing its inclusive nature. To the right, there is a 'Languages' section and a link to 'https://github.com/processing/p5.js'. Above the languages section, there is a row of small profile pictures of community members.

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

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



Why p5.js?

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



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/>

The Most Basic **p5.js** sketch

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



The Most Basic **p5.js** sketch

`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);
}
```



The Most Basic **p5.js** sketch

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

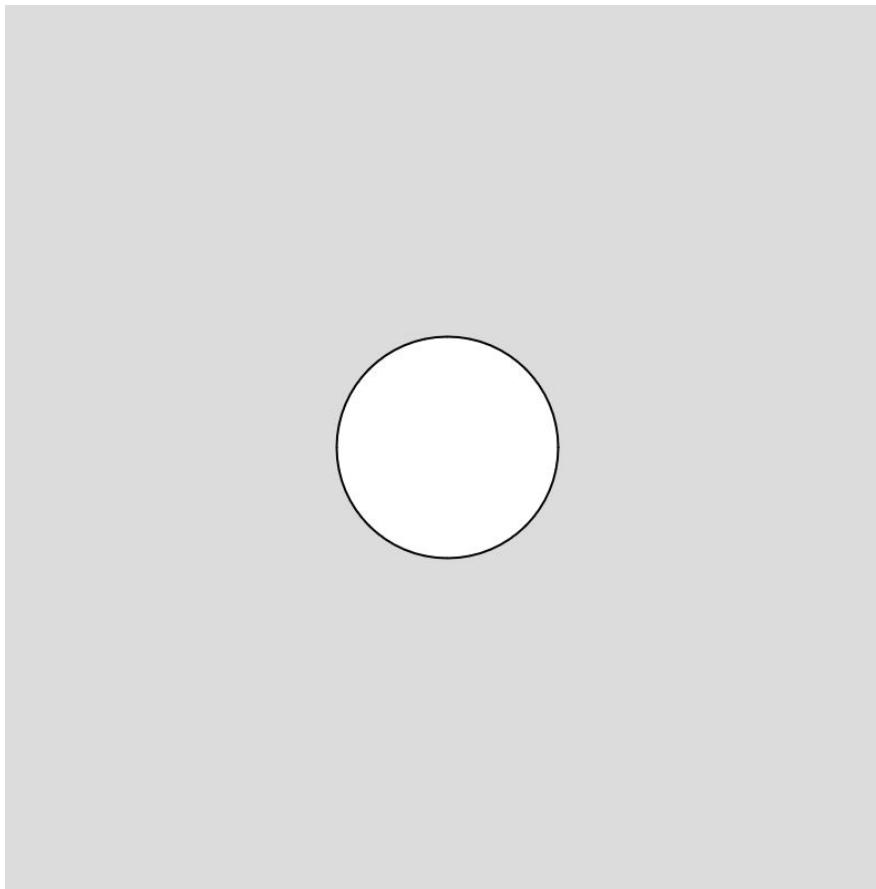
```
function setup(){
    createCanvas(400, 400);
}

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?

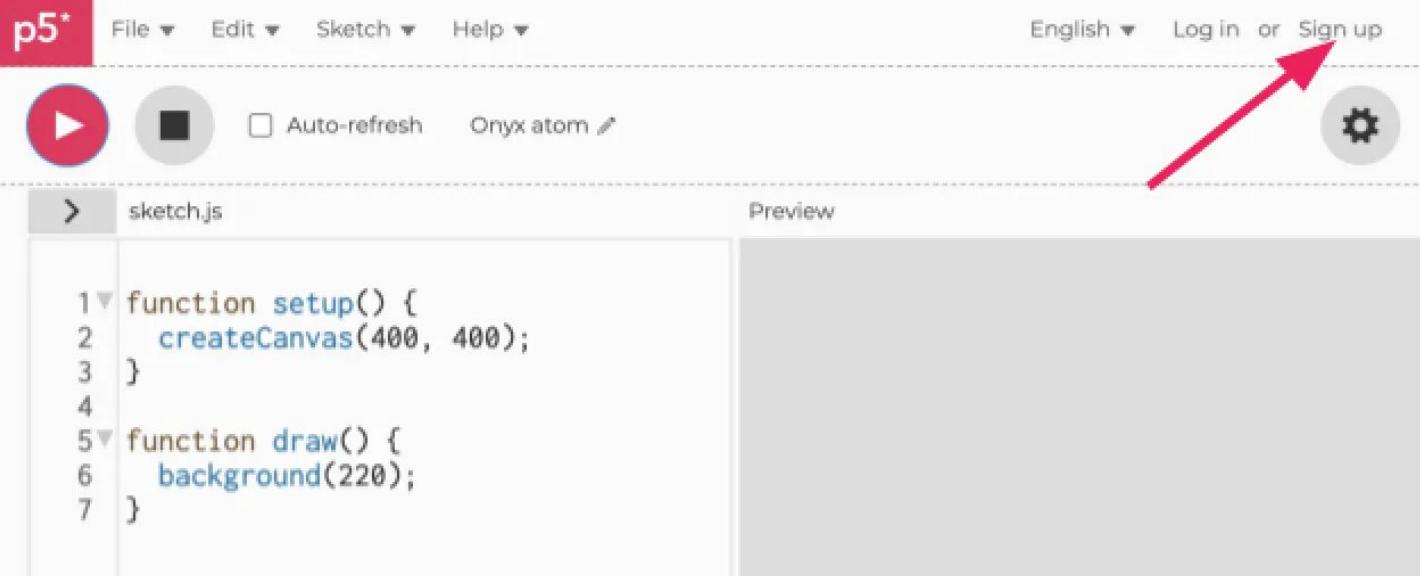


P5 Live Editor



<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.



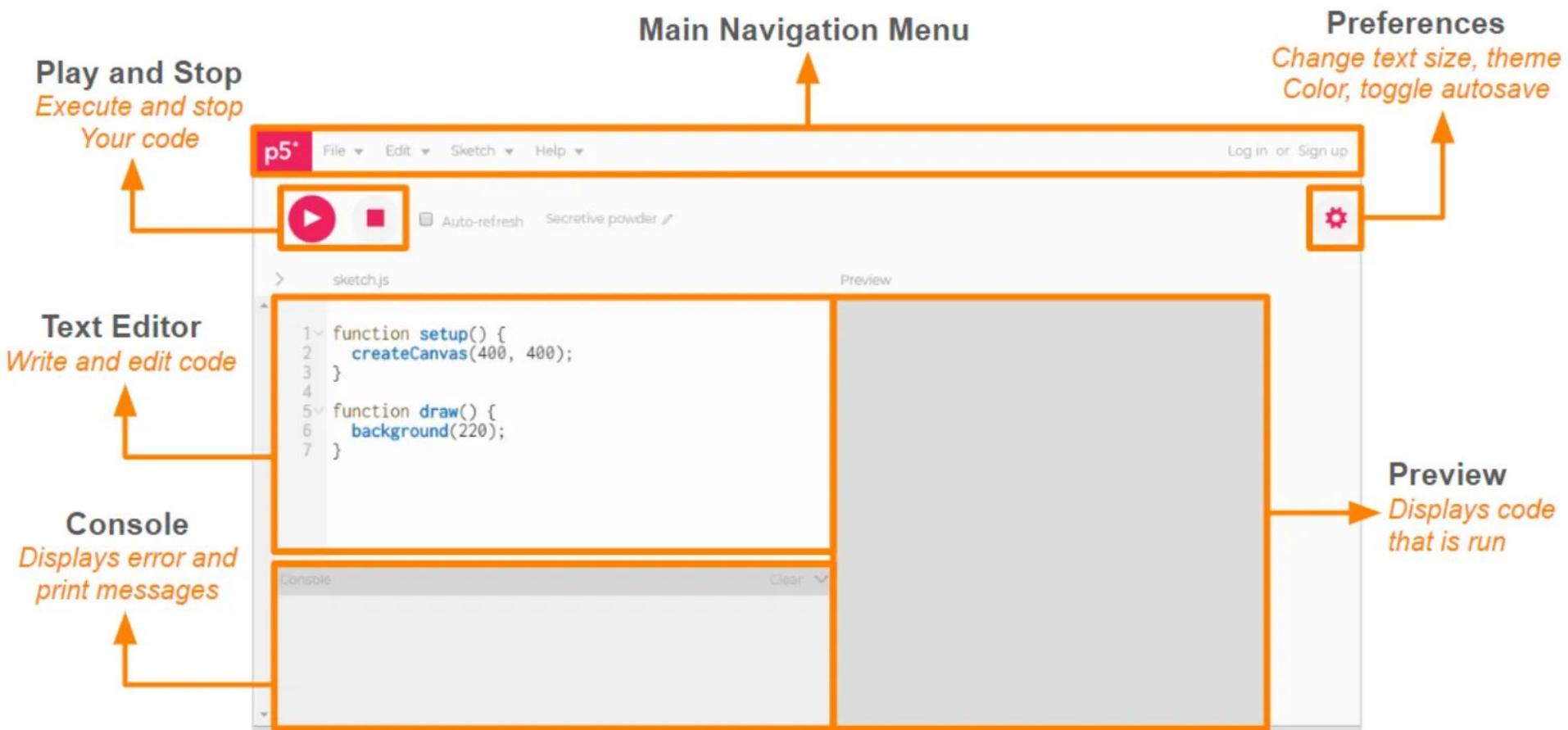
A screenshot of the p5.js Web Editor interface. At the top, there's a red header bar with the "p5*" logo. Below it is a navigation bar with "File", "Edit", "Sketch", and "Help" menus, followed by language selection ("English"), and "Log in" and "Sign up" buttons. A red arrow points from the "Sign up" button towards the gear icon in the top right corner. The main area has two panes: a code editor on the left containing the following p5.js code, and a preview window on the right.

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

Let's try to draw a circle with it!



p5.js Web Editor Interface



p5*

p5.js Drawing a Circle

p5*

File ▾ Edit ▾ Sketch ▾ Help ▾



Auto-refresh

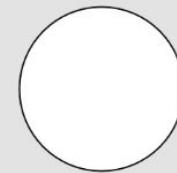
Seed crawdad 🖌 by DavideConficconi

> 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 ▾

Today Objectives

1. ~~Informatica Self Assessment~~

2. ~~Local dev with VSCode, Indice html for IA2425~~



3. ~~Create our Webpage with Githubpages~~



4. ~~Draw with P5, P5 Live Editor~~

p5*.js

p5*

<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**

189

Thank you 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, FdI 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

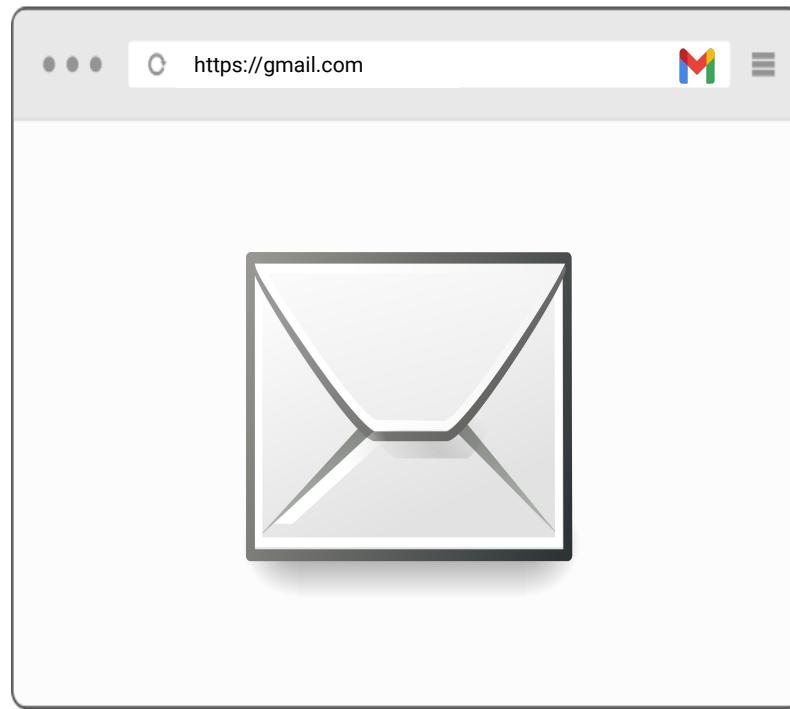


JavaScript: aspetti di sicurezza



Immaginiamo di aprire due schede nel nostro browser

Scheda 1: Il nostro account di posta elettronica

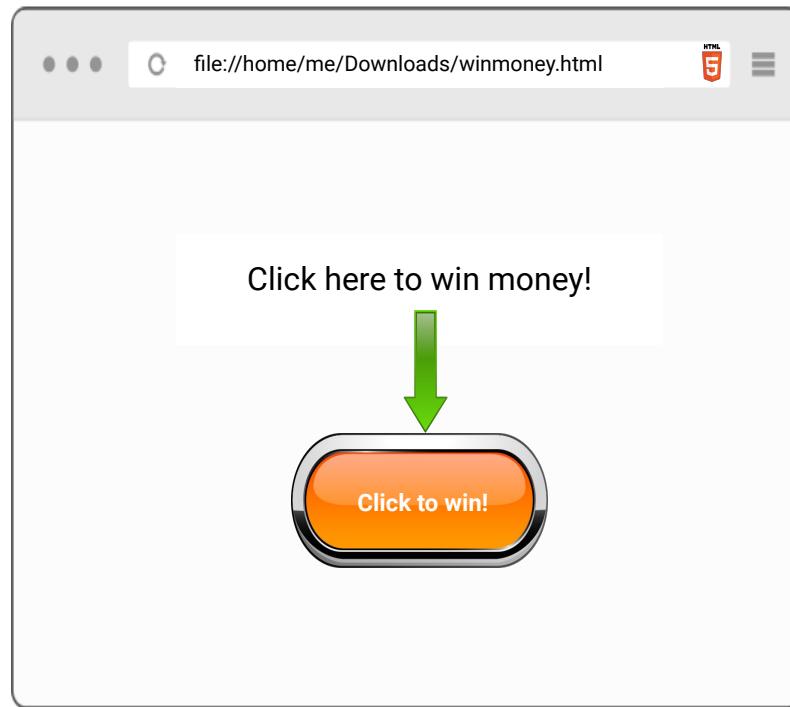


JavaScript: aspetti di sicurezza



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

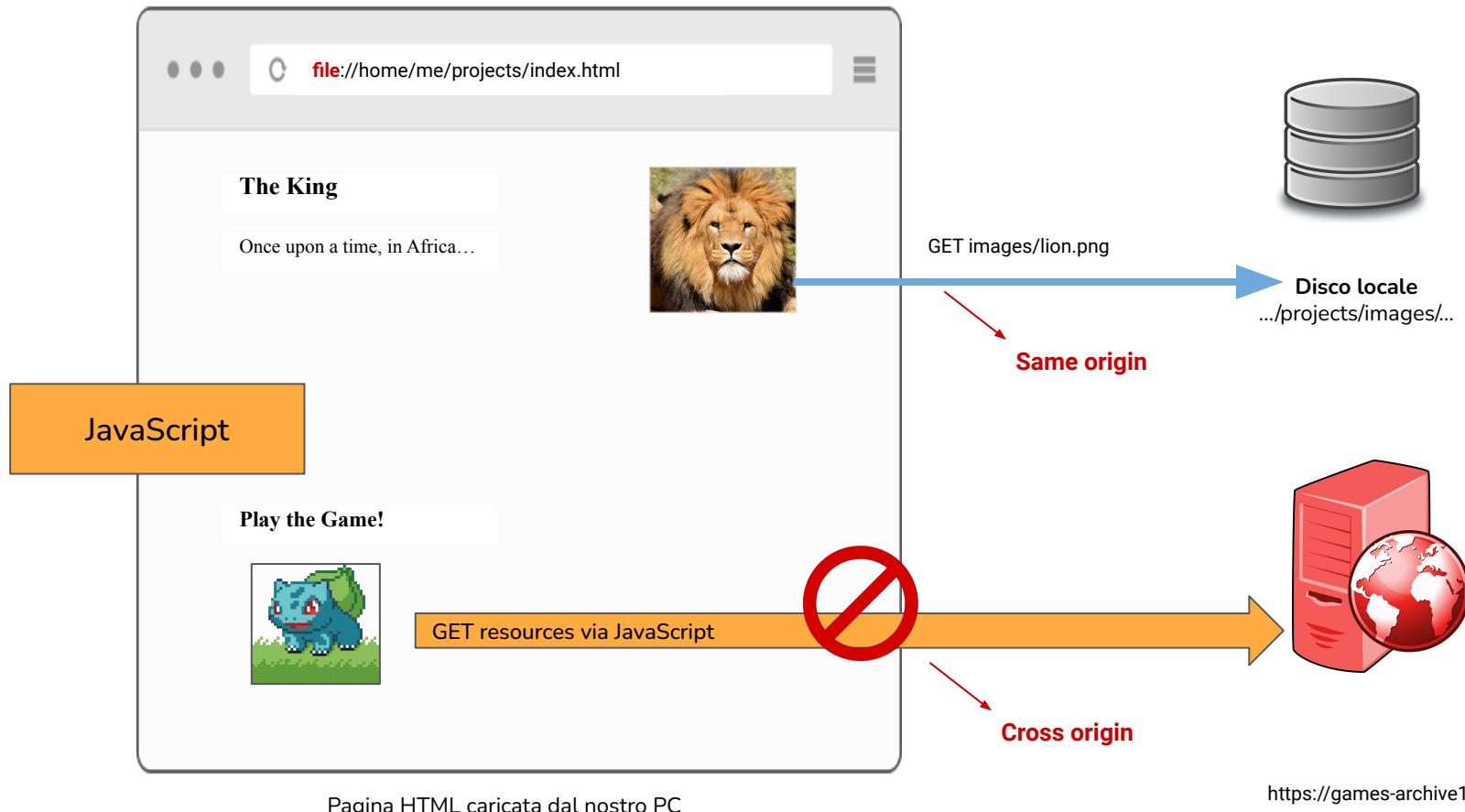
L'esecuzione di codice JavaScript di questo tipo consentirebbe il furto di dati e informazioni sensibili da parte di malintenzionati

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



JavaScript: aspetti di sicurezza



La same-origin policy introduce però delle restrizioni piuttosto forti

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

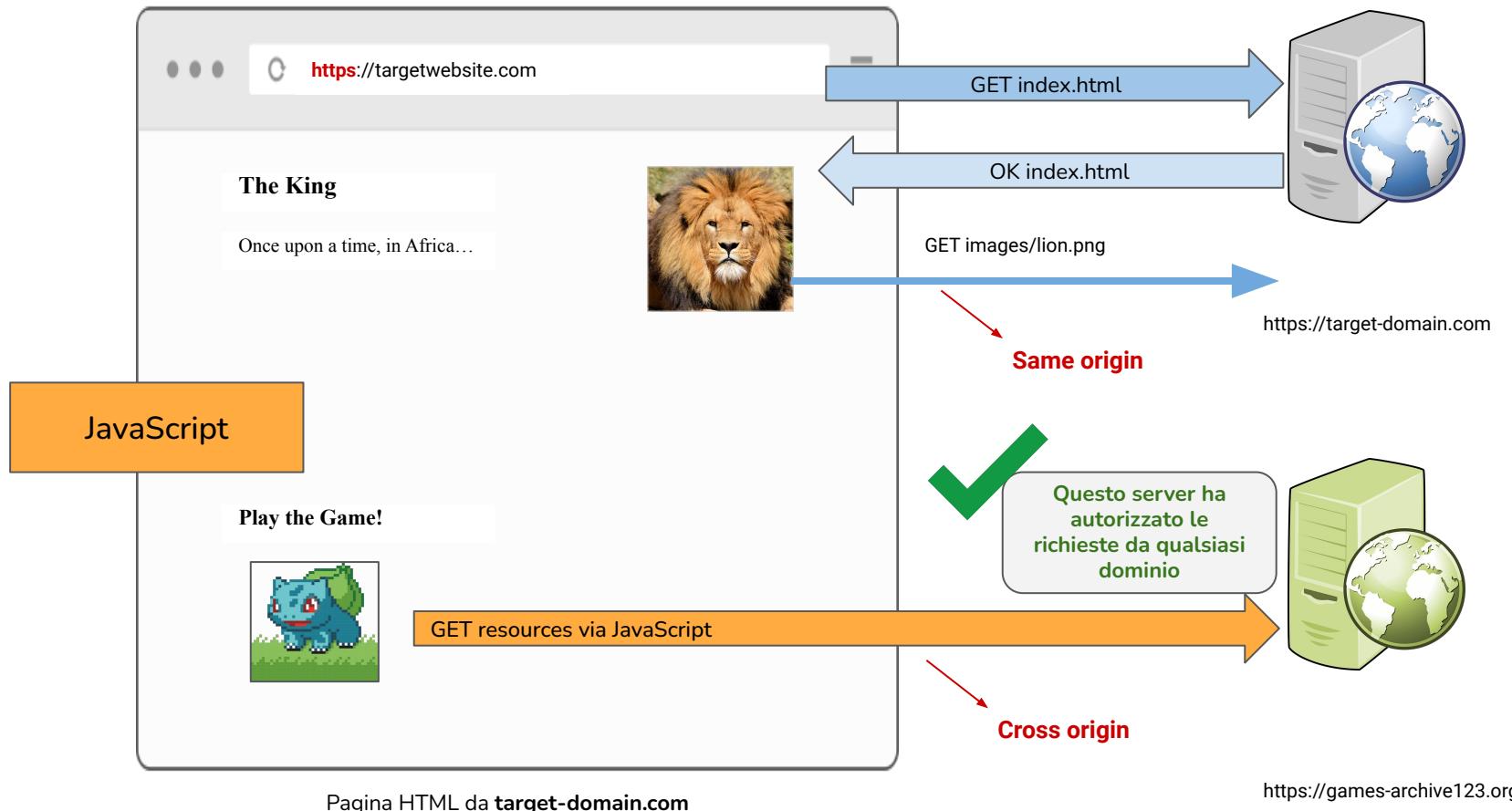
Per permettere l'accesso a risorse su altri domini, i server HTTP implementano la cosiddetta **Cross-Origin Reference Sharing (CORS)**

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

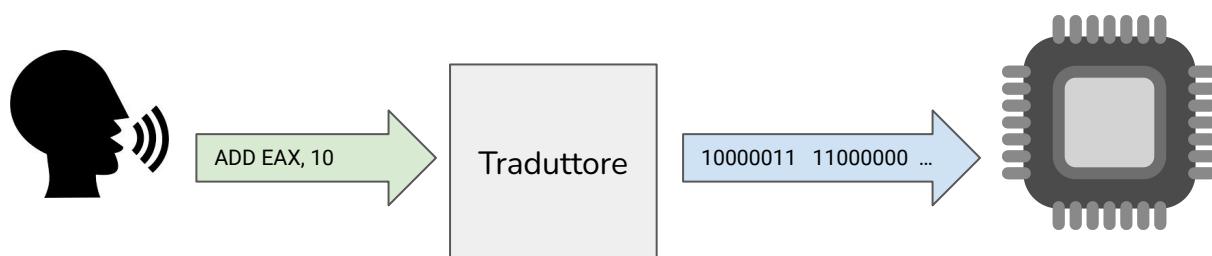
La nostra applicazione deve essere gestita tramite server HTTP, come se fosse un sito Web, anziché essere aperta come un file HTML locale

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