



GitHub, Classroom e Student Developer Pack

GitHub offre a studenti e professori Github Classroom e GitHub Student Developer Pack, pacchetti straordinari di strumenti e servizi gratuiti.

 III Liceo Scientifico Biella - Scienze Applicate

 Python Biella Group



8 Vantaggi "standard" + 2 extra!

★ 1. GitHub è gratuito (e potente!)

Gli studenti possono usare GitHub senza limiti per:

- creare repository pubblici o privati
- lavorare su progetti personali o scolastici
- imparare gli stessi strumenti usati dagli sviluppatori professionisti



2. Salvataggio sicuro del codice

GitHub è una “nuvola del codice”:

- non rischi di perdere tutto perché il PC si rompe
- il prof non può più accettare la scusa “me l’ha mangiato il cane”
- puoi accedere ai tuoi progetti da qualsiasi computer



3. Collaborazione facilitata

Perfetto per lavori di gruppo:

- più persone possono lavorare sullo stesso progetto
- nessuno sovrascrive il lavoro degli altri
- si può vedere *chi ha fatto cosa* (utile anche ai prof 😊)



4. Storico delle modifiche

Puoi:

- tornare indietro se rompi qualcosa
 - confrontare versioni
 - capire come un progetto è cresciuto nel tempo
- È come una “macchina del tempo” del tuo codice.



5. Portfolio professionale

Gli studenti possono costruire un vero portfolio:

- utile per l'università
- utile per colloqui futuri
- mostrare competenze senza dover dire "so programmare, fidati"

Gli studenti che arrivano alle superiori o all'università con un GitHub attivo sono avvantaggiati tantissimo.



6. Materiale educativo gratuito

GitHub è pieno di:

- esempi di codice
- strumenti open-source
- progetti da cui imparare
- tutorial e guide ufficiali

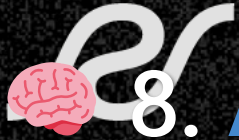
È come una grande biblioteca open source.



7. Ottimo per AI e la Data Science

Per un liceo con "Scienza dei dati e AI", GitHub permette:

- usare notebook Jupyter
- condividere dataset
- tenere versioni dei modelli
- sperimentare collaborando
- provare workflow moderni di sviluppo AI



8. Abituarsi agli strumenti reali del lavoro

Git + GitHub sono nel 99% dei lavori IT:

- programmazione
- ingegneria del software
- intelligenza artificiale
- cybersecurity
- elettronica
- sviluppo web
- ricerca scientifica

Impararli al liceo = vantaggio competitivo gigantesco.



9. GitHub Classroom (per la scuola)

Molte scuole lo usano per:

- assegnare esercizi
- consegnare compiti di informatica
- gestire versioni e verifiche di codice
- automatizzare controlli e correzioni

Gli studenti possono consegnare il compito con **un click**.



Cos'è GitHub Classroom?

GitHub Classroom è una piattaforma gratuita che permette ai professori di:

- Creare e distribuire compiti/progetti
- Gestire repository per ogni studente
- Correggere automaticamente il codice
- Monitorare i progressi della classe
- Dare feedback direttamente sul codice

In pratica: È come Google Classroom, ma per il codice! 💻



Come Funziona (Lato Professore)

1. Creazione Classroom

Il prof crea una "classe virtuale" su GitHub Classroom collegata a un'organizzazione GitHub.

2. Creazione Assignment (Compito)

```
Assignment: "Gioco Snake con PyGameZero"  
├── Repository template (codice base)  
├── Test automatici  
├── Deadline  
└── Istruzioni nel README
```

3. Distribuzione

Il prof condivide un **link di invito** con la classe.



Come Funziona (Lato Studente)

1. Click sul link condiviso dal prof
2. Autorizza GitHub Classroom ad accedere al tuo account
3. Accetta l'assignment
4. GitHub crea automaticamente un repository personale per te:

```
snake-game-mario-rossi/
```

5. Clona il repository e lavora al progetto
6. Commit e push le tue modifiche
7. GitHub testa automaticamente il codice
8. Il prof vede i risultati e dà feedback



Esempio pratico: "alieno PRO"

Il prof prepara:

```
# starter_code.py
import pgzrun

# TODO: Aggiungi il contatore di punteggio
# TODO: Implementa movimento casuale alieno
# TODO: Aggiungi timer di gioco

alieno = Actor('alien')
alieno.pos = 100, 56

def draw():
    screen.fill((0, 0, 255))
    alieno.draw()
    # TODO: Mostra punteggio

pgzrun.go()
```




Ogni studente riceve:

- Repository personale con questo codice
- README con istruzioni dettagliate
- Test automatici per verificare le funzionalità



Vantaggi per gli Studenti

1. Repository Personale

- Ogni studente ha il suo spazio privato
- Il prof vede il tuo lavoro, i compagni no
- Nessun rischio di copiare accidentalmente

2. Feedback Automatico

- ✓ Test punteggio: PASSED
- ✓ Test movimento: PASSED
- ✗ Test timer: FAILED

Sai subito se hai fatto giusto!



3. Storico Completo

- Tutti i commit sono tracciati
- Il prof vede quando hai lavorato
- Puoi tornare indietro se rompi qualcosa

4. Apprendimento Reale

- Usi gli stessi strumenti dei professionisti
- Portfolio GitHub si arricchisce
- Esperienza pratica con Git



1. Distribuzione Automatica

Un click e 25 studenti hanno il loro repository pronto

2. Correzione Semi-Automatica

```
# test_game.py
def test_score():
    assert game.score >= 0
    assert game.score == game.clicks

def test_timer():
    assert game.timer > 0
```

Test automatici verificano funzionalità base



3. Monitoraggio Progressi

Dashboard che mostra:

- Chi ha accettato l'assignment
- Chi ha fatto commit
- Risultati dei test
- Chi è in ritardo



4. Feedback Diretto

Commenti inline sul codice:

```
# ✗ Prof: Usa una funzione invece di ripetere codice  
score = score + 1  
score = score + 1  
score = score + 1
```




1. Individual Assignment Ogni studente lavora da solo

Esempio: "Crea il tuo gioco PyGameZero"

2. Group Assignment Team di studenti su un repository condiviso

Esempio: "Progetto AI in gruppo (3-4 persone)"

3. Autograded Assignment Con test automatici che danno voto

Test pass: 8/10 = 80/100



1. Template Repositories

Il prof prepara codice starter:

```
template-python-game/
├── README.md (istruzioni)
├── requirements.txt
├── game.py (codice base)
├── tests/ (test automatici)
└── .github/workflows/ (CI/CD)
```

2. Deadline e Late Submissions

- Scadenza automatica
- Possibilità di consegne in ritardo
- Penalità configurabili



3. GitHub Actions Integration

Test automatici che girano ad ogni push:

```
# .github/workflows/test.yml
name: Test Game
on: [push]
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v2
      - name: Run tests
        run: python -m pytest
```

4. Feedback Pull Requests

Il prof può creare PR con suggerimenti di miglioramento



Workflow Tipico di classe

Settimana 1: Setup

1. Prof crea GitHub Classroom
2. Studenti accettano invito alla "classe"
3. Tutti configurano Git in Thonny



Settimana 2-N: Assignments

Esempio: "Snake Game"

Lunedì: Prof assegna homework

Link: classroom.github.com/a/snake-game-hw

Lunedì-Venerdì: Studenti lavorano

```
git clone <repo-url>
# lavora in Thonny
git add .
git commit -m "Aggiunto movimento"
git push
```




Sabato: Deadline

- Test automatici verificano tutto
- Prof rivede codice e dà feedback
- Studenti vedono risultati



1. Commit Frequenti

- ✗ 1 commit alla fine: "fatto tutto"
- ✓ Tanti commit piccoli:
 - "Aggiunto setup iniziale"
 - "Implementato movimento"
 - "Fixato bug collisioni"

2. Testa Prima di Pushare

```
# Assicurati che funzioni in Thonny!  
python game.py  
# Poi puoi pushare
```




3. Leggi i Feedback

Il prof commenta il codice → leggi e migliora!

4. Non Aspettare l'Ultimo Minuto

GitHub Classroom traccia QUANDO lavori 😊



Privacy e Sicurezza

Cosa Vede il Prof:

- ✓ Il tuo codice
- ✓ I tuoi commit (con date/ore)
- ✓ Risultati test
- ✓ Statistiche contributi

Cosa NON Vede:

- ✗ Altri tuoi repository privati
- ✗ La tua email (se privata nelle impostazioni)
- ✗ Il tuo lavoro su altri progetti



Cosa Vedono i Compagni:

- ✗ NIENTE! (ogni repository è privato)
- ✓ Solo in group assignment vedono il lavoro del team



Come Iniziare (Per il Prof)

Setup in 10 Minuti:

1. Vai su: classroom.github.com
2. Crea Organization (se non esiste):

Nome: "Liceo-ScienzeApplicate-2024-25"



3. Crea Classroom:

Nome: "Classe 3A - IA e Data Science"

4. Crea primo Assignment:

Titolo: "Colpisci Alieno Avanzato"

Template: repository con codice base

Deadline: 1 settimana

5. Condividi link con studenti!



Integrazione con Thonny

Gli studenti possono usare GitHub direttamente da Thonny:

```
Thonny → Strumenti → Apri shell di sistema
```

```
# Clone del repository assignment
```

```
git clone https://github.com/classroom/repo-studente
```

```
# Apri il progetto in Thonny
```

```
# Lavora sul codice...
```

```
# Commit e push
```

```
git add .
```

```
git commit -m "Implementata nuova feature"
```

```
git push
```




10. GitHub Student Developer Pack (super bonus!)

Registrandosi come studenti (gratis), si ottiene accesso a decine di strumenti professionali:

- servizi cloud
- domini web
- IDE avanzati
- strumenti di modellazione 3D
- risorse per machine learning
- crediti per piattaforme professionali
- software di design e DevOps

Valore totale: centinaia di euro → gratis per studenti.

PBG Vantaggi Principali GitHub per studenti

1. GitHub Pro GRATIS (normalmente \$4/mese)

- Repository privati illimitati
- GitHub Copilot gratis (AI che scrive codice!)
- GitHub Actions con più minuti
- GitHub Pages con dominio personalizzato
- Badge "PRO" sul profilo

2. GitHub Copilot (normalmente \$10/mese)

- Assistente AI che suggerisce codice
- Autocomplete intelligente
- Spiega codice complesso
- Funziona in Thonny, VS Code, etc.



Software e Servizi Inclusi (oltre 100!)

Sviluppo e Hosting

- DigitalOcean: \$200 di crediti cloud
- Heroku: Hosting gratuito per 2 anni
- Microsoft Azure: \$100 crediti/anno
- Namecheap: Dominio .me gratuito per 1 anno
- Name.com: Dominio gratuito per 1 anno



- Canva Pro: 12 mesi gratis
- Adobe Creative Cloud: 60% sconto
- Figma: Team gratuito per studenti

Apprendimento e Sviluppo

- DataCamp: 3 mesi gratis (Python, Data Science)
- Educative: 6 mesi premium
- Frontend Masters: 6 mesi gratis
- Interview Cake: Accesso completo



Tool per Programmatori

- **JetBrains:** Tutti gli IDE gratis (PyCharm, IntelliJ, etc.)
- **Bootstrap Studio:** Licenza gratuita
- **GitKraken:** Client Git Pro gratis
- **Termius:** SSH client premium

Database e Backend

- **MongoDB:** \$200 in credits
- **Mailgun:** Email service gratuito
- **Twilio:** Crediti per SMS/API



Come Ottenere il Pack

1. Vai su: education.github.com/pack
2. Click su "Sign up for Student Developer Pack"
3. Verifica il tuo status studente con:
 - Email istituzionale della scuola (@scuola.edu.it)
 - Oppure carica documento studente/tessera
4. **Attendi** approvazione (di solito 1-7 giorni)
5. **Attiva** i benefici che ti interessano



Consigli per la Classe

Must Have per il vostro corso:

1. GitHub Copilot - per Python e progetti AI
2. JetBrains PyCharm Professional - IDE potente
3. DataCamp - per Data Science
4. DigitalOcean/Heroku - per deployare progetti

Per Portfolio Professionale:

- Dominio personalizzato (portfolio.nome.me)
- GitHub Pages Pro per sito personale
- Canva Pro per presentazioni fighe



Valore Totale

Il pack vale oltre \$200,000 in software e servizi!
È come avere un budget professionale mentre studiate.



Requisiti

- Essere studente attivo
- Avere almeno 13 anni
- Account GitHub
- Email scolastica O documento che provi status studente



Durata

- Il pack dura finché sei studente
- Devi rinnovare la verifica ogni anno
- Alcuni servizi hanno limiti temporali specifici



Bonus: GitHub Classroom + Student pack

- Copilot aiuta a scrivere codice
- Classroom verifica che sia corretto
- Feedback del prof per migliorare

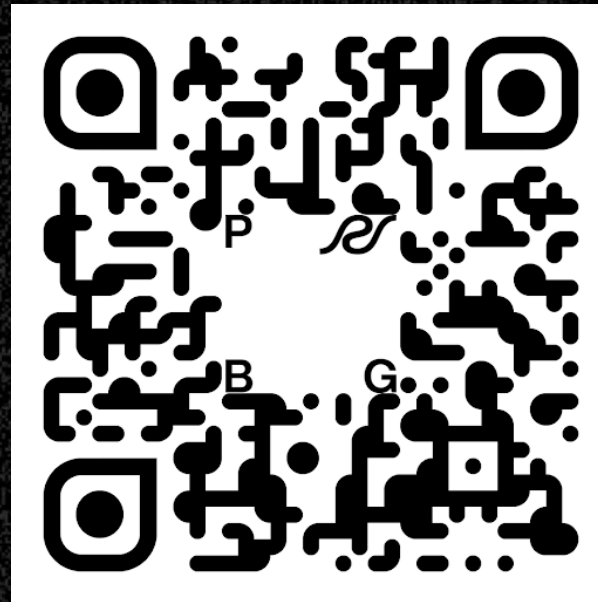
= Apprendimento potenziato dall'AI! 



Feature	Email/Teams	GitHub Classroom
Versionamento	✗	✓
Test automatici	✗	✓
Feedback inline	✗	✓
Portfolio	✗	✓
Collaborazione	Limitata	✓ Pro
Tracking lavoro	✗	✓ Dettagliato



Grazie per l'attenzione...



"C'è sempre qualcosa da imparare per migliorarci e crescere...insieme!"