



Giulia Peserico
Maria Serafini
Francesca Voltolini



UNIVERSITÀ
DI TRENTO
Dipartimento di
Ingegneria e Scienza dell'Informazione

Federica Picasso
Daniele Agostini
Francesca Fiore
Anna Serbati
Alberto Montresor



Per l'insegnamento della programmazione

La sperimentazione del metodo

PRIMM

IL PERCORSO

Indice

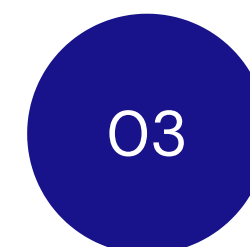
IL PROGETTO



IL CONTESTO



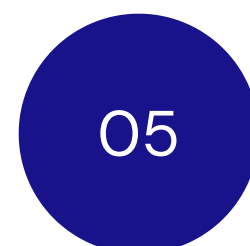
I RISULTATI



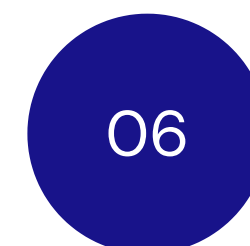
ESEMPIO ATTIVITÀ



CONCLUSIONI



I MATERIALI



Il progetto

01

PROBLEMA

Sindrome della finestra vuota

Gli studenti trovano **difficile la programmazione?**
Fissano con sguardo assente la finestra dell'editor vuota e non sanno da dove iniziare?
Alcune ricerche hanno dimostrato che molti studenti hanno difficoltà a scrivere un programma.

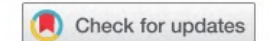


Metodologia PRIMM


Approccio pedagogico che parte dall'osservazione che l'insegnamento "tradizionale" della programmazione richiede fin da subito la scrittura di codice da parte dello studente. Questo differisce dalle metodologie utilizzate nell'apprendimento delle lingue (compresa la madrelingua) dove l'attività di produzione è preceduta dalla lettura e dalla comprensione del testo.

Sito PRIMM:

<https://primmportal.com/>



Teaching computer programming with PRIMM: a sociocultural perspective

Sue Sentance ^a, Jane Waite^b and Maria Kallia^a

^aDepartment of Education, Communication and Society, King's College London, London, UK;

^bSchool of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

ABSTRACT

Background and Context: Vygotsky's sociocultural theory emphasises the importance of language, mediation, and the transfer of skills and knowledge from the social into the cognitive plane. This perspective has influenced the development of PRIMM (Predict, Run, Investigate, Modify, Make), a structured approach to teaching programming.

Objective: The objective of the study was to find out if using PRIMM to teach programming had an impact on learner attainment in secondary school, and the extent to which it was valuable for teachers.

Method: We evaluated the use of PRIMM in 13 schools with 493 students aged 11-14 alongside a control group, using a mixed-methods approach. Teachers delivered programming lessons using the PRIMM approach for 8-12 weeks. Data were collected via a combination of a baseline test, a post-test to compare control and experimental groups, and teacher interviews.

Findings: Learners performed better in the post test than the control group. Teachers reported several benefits of the PRIMM approach, including that PRIMM helped them to teach effectively in mixed-ability classes, enabling all learners to make progress.

Implications: PRIMM makes a contribution to programming education research by building on previous work in effective pedagogy for teachers, and encouraging the use of language and dialogue to facilitate understanding.

ARTICLE HISTORY

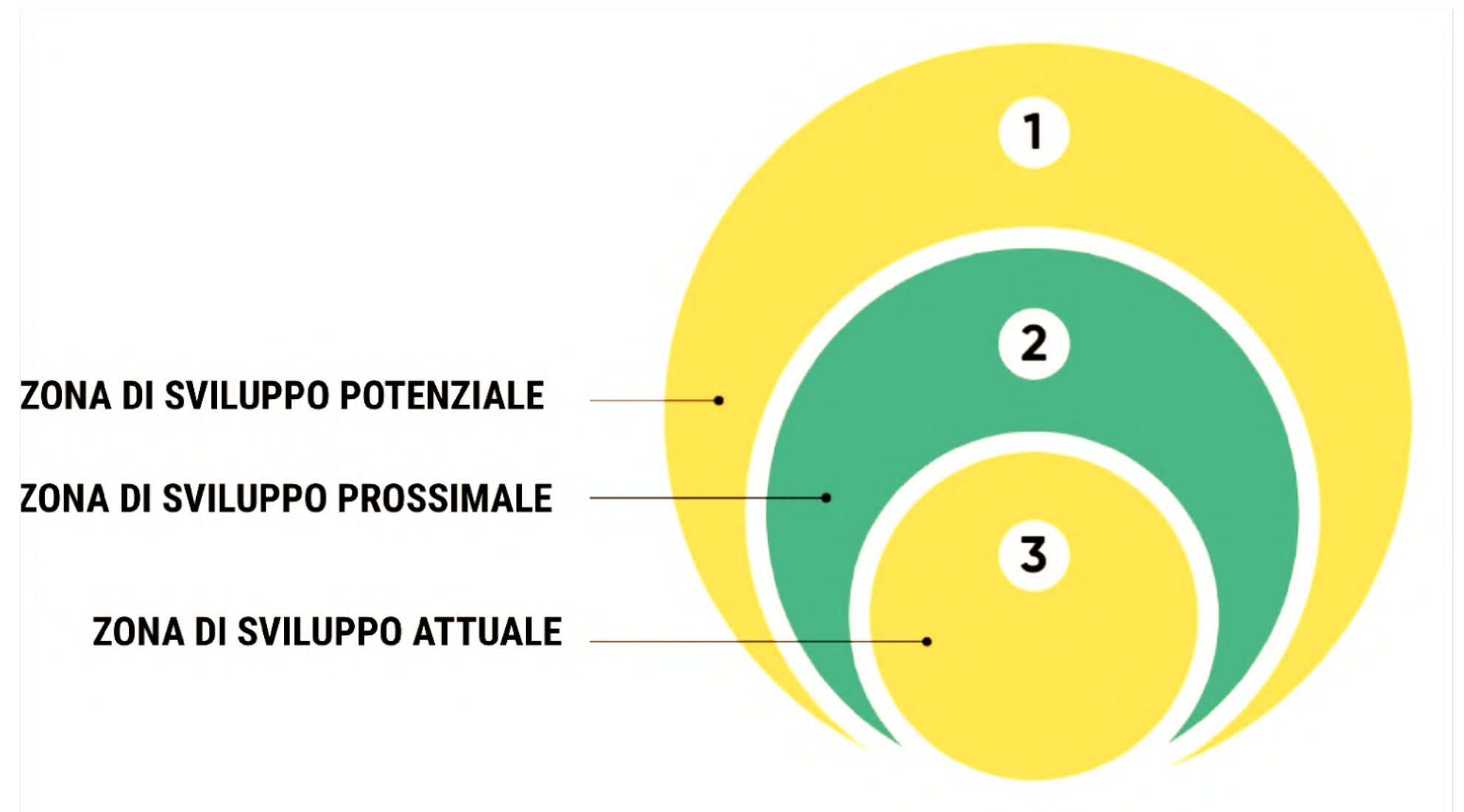
Received 6 August 2018

Accepted 15 April 2019

KEYWORDS

Computer programming; computer science education; K-12 education; pedagogy; sociocultural theory; Vygotsky

Zona di sviluppo prossimale



Teoria del pedagogista Vygotskij:

approccio strutturato all'apprendimento che vede l'insegnante come esperto che supporta lo studente nell'acquisire nuove competenze tramite il processo di scaffolding.

PRINCIPI FONDANTI

Leggere prima di scrivere

Partendo dall'osservazione che il **carico cognitivo necessario per arrivare alla scrittura del codice è molto elevato**, la metodologia PRIMM inverte la sequenza tradizionale, partendo prima dalla comprensione del codice e solo successivamente arrivando alla scrittura di codice vero e proprio



PRINCIPI FONDANTI

Mediazione attraverso il linguaggio

Una volta capito come funzionano i programmi, gli studenti sono incoraggiati a discutere tra loro, **verbalizzando** quanto appreso gli studenti riescono a **interiorizzare meglio i concetti**. Ciò può avvenire attraverso attività collaborative.



PRINCIPI FONDANTI

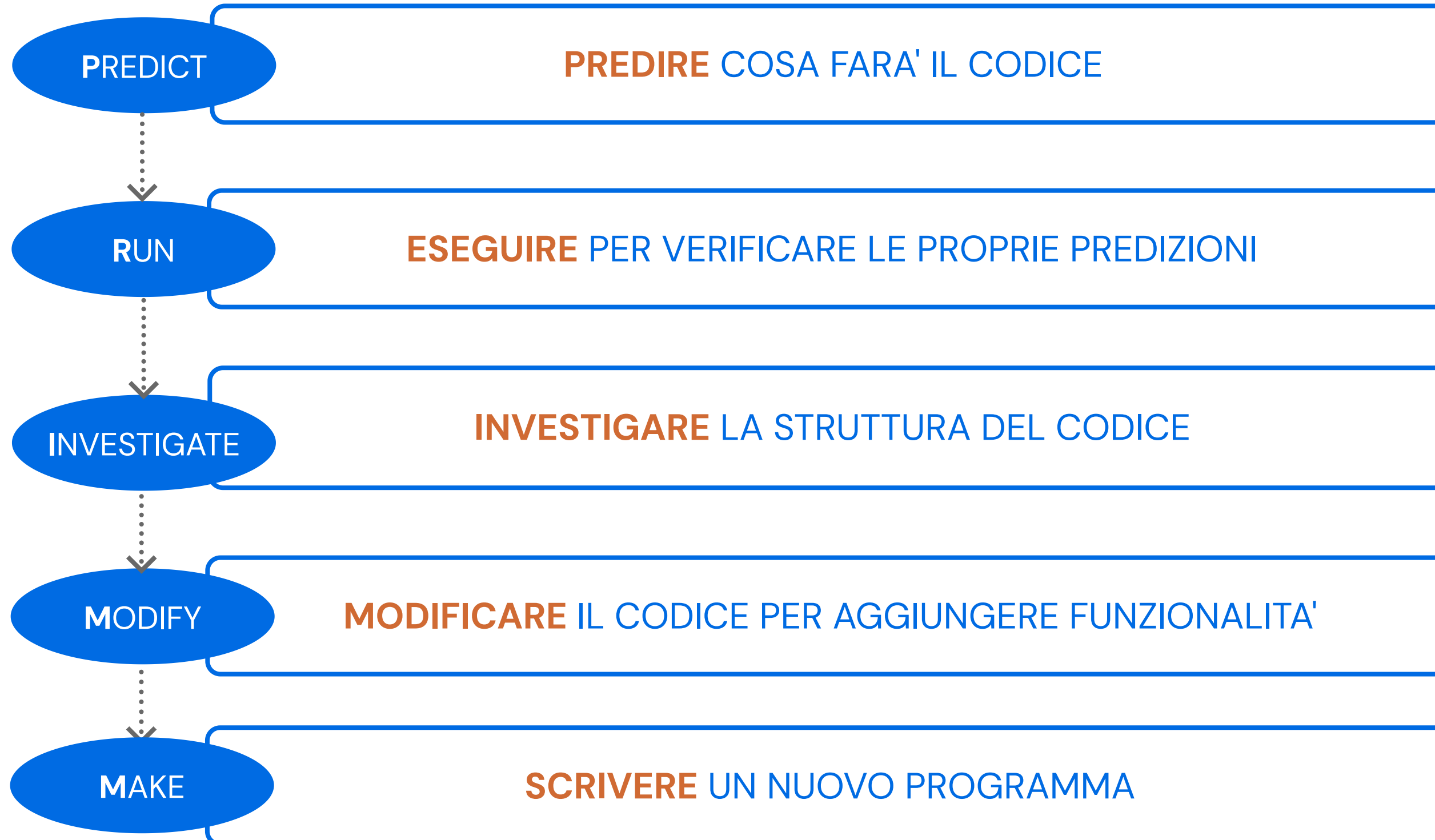
Imparare da chi è più informato

Gli studenti hanno bisogno che chi ha più conoscenza mostri loro il modello per risolvere un problema:

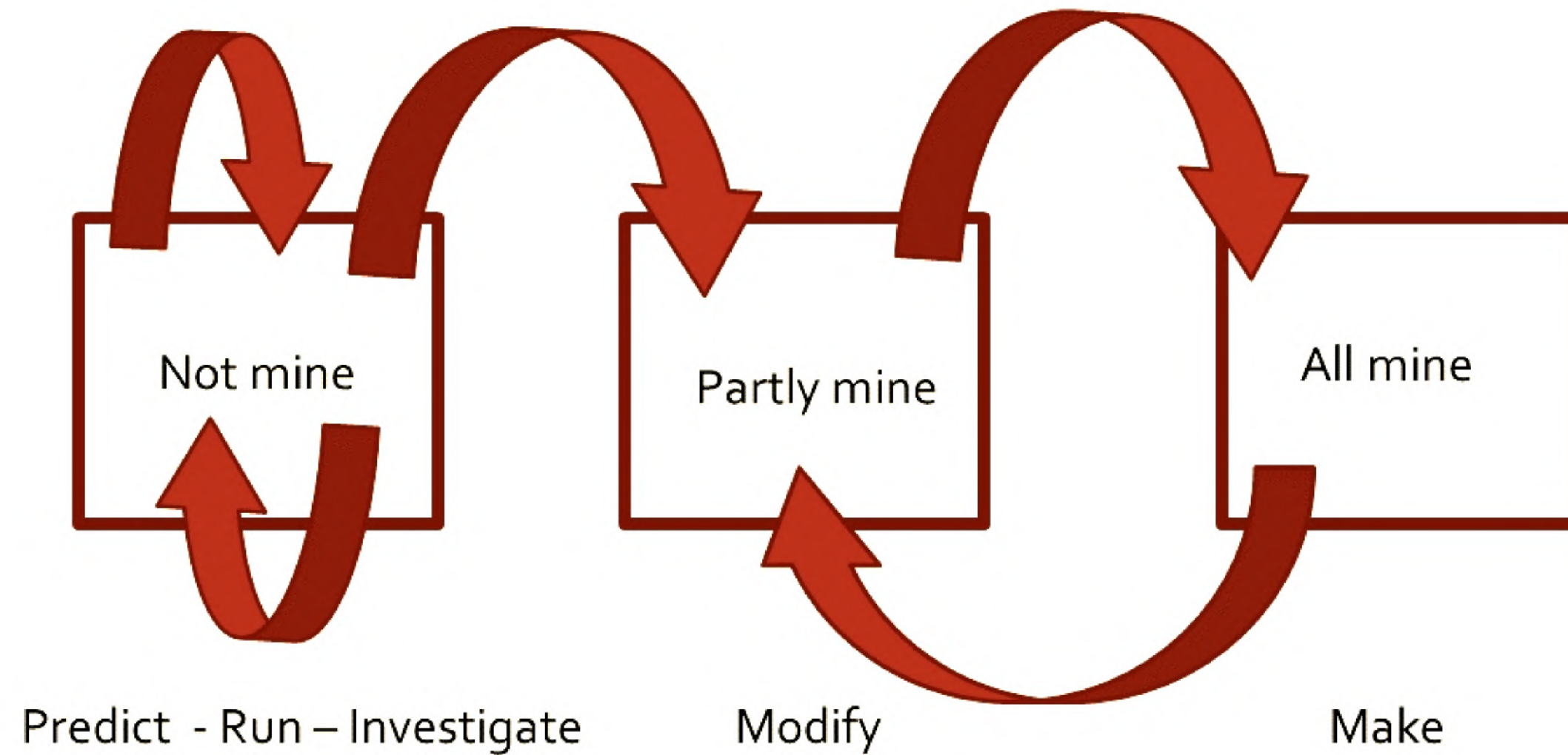
- insegnante
- peer education



Predict – Run – Investigate – Modify – Make



"Appropriazione" del codice



Il contesto

02

LA SCUOLA

Liceo Scientifico "Leonardo Da Vinci" Trento

- 1530 alunni
- 75 classi
- 33 ordinamentale + 42 scienze applicate
- Insegnamento Linguaggio **Python** dalla II° alla V°



CLASSI COINVOLTE

CLASSI SECONDE – Opzione Scienze Applicate

CLASSI SPERIMENTAZIONE

- 5 classi seconde
- 102 studenti
- 3 diversi insegnanti

CLASSI DI CONTROLLO

- 4 classi seconde
- 86 studenti
- 2 diversi insegnanti

I risultati

03

VERIFICA DELL'EFFICACIA DEL METODO

I tre tipi di dato

Questionario IMI

Il seguente questionario serve per stabilire le tue attitudini nei confronti della programmazione. In tutte le domande, laddove troverai l'indicazione 'questa attività' pensa all'attività di programmare (inteso come scrivere codice in python).

giulia.peserico@liceodavincitn.it [Cambia account](#)

* Indica una domanda obbligatoria

Email *

Il tuo indirizzo email

Indica per ciascuna delle seguenti affermazioni quanto sei d'accordo *

1 (per niente vero) 2 3 4 (parzialmente vero) 5 6 7 (completamente vero)

Quest'attività non ha attirato per niente la mia attenzione

Questionario IMI

(Intrinsic Motivation Inventory)
 Motivazione dello studente a svolgere questo tipo di attività.

1. Quale dei seguenti script fa dire allo sprite: "Sto bene!" per 3 volte?

A

B

C

D

2. Quale dei seguenti script fa correre lo sprite...

A

C

Quando esegui il percorso effett...

Test finale classe seconda

Completa il seguente questionario ricordando che nelle risposte con caselle rotonde c'è UNA sola risposta esatta, nelle risposte con caselle quadrate ci sono PIU' risposte esatte

Questo modulo raccoglie automaticamente gli indirizzi email di chi risponde [Modifica impostazioni](#)

Sezione 1 di 2

Quale dei seguenti frammenti di codice visualizza a video la scritta 'sto bene' per 3 volte? *

`print("sto bene","sto bene")`

`for i in range(30):
 print("sto bene")
 print("sto bene")
 print("sto bene")`

`for i in range(3):
 print("sto bene")`

`print("sto bene",3)`

Test strutturato

L'apprendimento di conoscenze e abilità riguardanti la programmazione

1	Nominativo	IONE CATTOLI	NATURA ITALIA	INFORMATICA	ANIERA INGLI	ANIERA TEDES	RIA E GEOGRA	MATEMATICA	FISICA	SCIENZE NATUR	TORIA DELL'AF	TORE E SPORT	TA RELAZIONI	ED. CIVICA	
2	1	4001	9	7	10	7	10	7	10	10	7	9	9	10	8
3	2	4002	10	8	10	7	9	7	8	9	7	9	10	10	9
4	3	4003	N,A	7	7	7	6	7	7	7	6	8	10	9	7
5	4	4004	N,A	7	5	8	7	8	5	5	6	7	8	8	7
6	5	4005	9	9	8	9	9	8	9	7	9	8	9	9	8
7	6	4006	10	7	8	6	8	7	9	9	8	9	9	10	7
8	7	4007	N,A	8	8	8	9	7	8	7	9	10	9	9	8
9	8	4008	9	7	7	8	9	6	7	6	7	8	9	9	7
10	9	4009	N,A	7	7	7	7	7	9	8	8	9	9	8	8
11	10	4010	N,A	9	8	8	9	8	9	7	9	9	10	9	8
12	11	4011	8	6	5	6	7	7	6	5	7	8	10	8	6
13	12	4012	9	7	7	9	6	9	7	7	9	7	9	10	8
14	13	4013	9	6	7	6	7	7	6	7	8	7	9	8	7
15	14	4014	9	6	6	6	6	7	7	6	8	9	10	8	7
16	15	4015	9	7	7	6	7	6	7	6	8	8	8	8	7
17	16	4016	N,A	6	7	7	6	7	8	7	7	7	9	8	7
18	17	4017	N,A	6	6	6	6	6	7	5	7	8	9	8	6
19	18	4018	N,A	6	6	6	6	6	7	4	6	7	8	9	6

Informazioni di contesto

Informazioni su classe di appartenenza, il genere e i voti finali nelle varie discipline scolastiche.

VERIFICA DELL'EFFICACIA DEL METODO

Analisi

Svolta dal dipartimento di Psicologia e Scienze Cognitive dell'Università di Trento (settore scientifico di pedagogia sperimentale)

Metodo

I primi due tipi di dato sono stati rilevati secondo un disegno **Pre-Post** sperimentazione

Risultati:

- **Dal test strutturato:** punteggi finali rispetto agli iniziali **lievemente più alti** con la sperimentazione PRIMM.
- **Dal test IMI:** la metodologia PRIMM potrebbe **promuovere l'impegno** degli studenti nelle attività, **potrebbe favorire la percezione della propria competenza** e gli studenti potrebbero essere in grado di **auto-valutarsi in modo più preciso**.
- **Dall'analisi sulla differenza di genere:** la metodologia PRIMM potrebbe essere **più inclusiva** rispetto a quella tradizionale.

Esempio di attività

04

IN CLASSE

Svolgimento lezioni



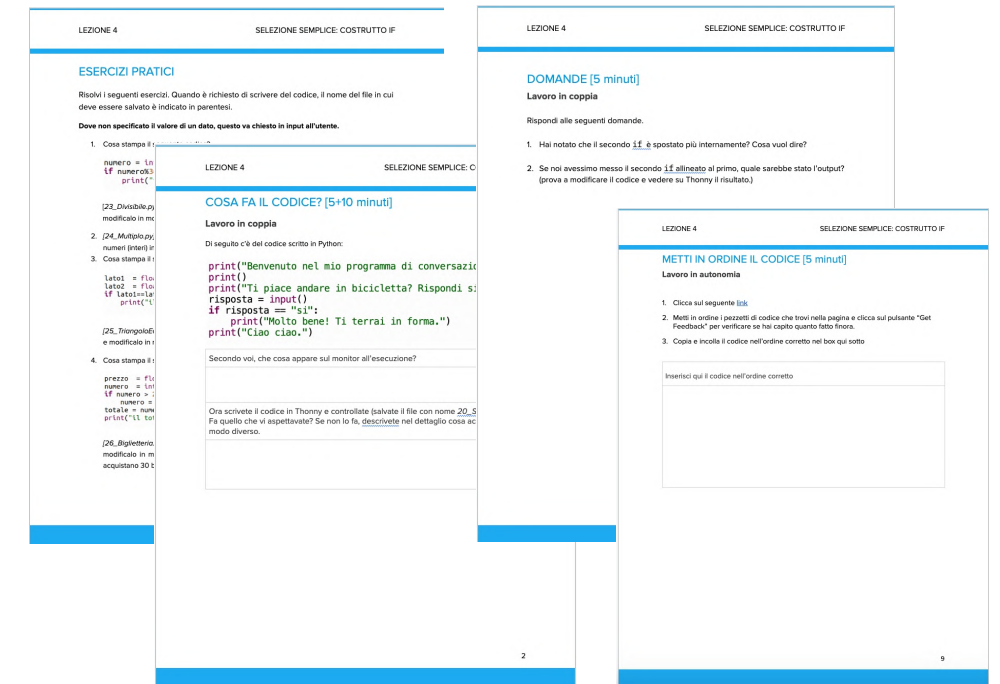
Laboratorio

La maggior parte delle lezioni è stata svolta in laboratorio



Disposizione a gruppi

Da 2 a 3 studenti per gruppo

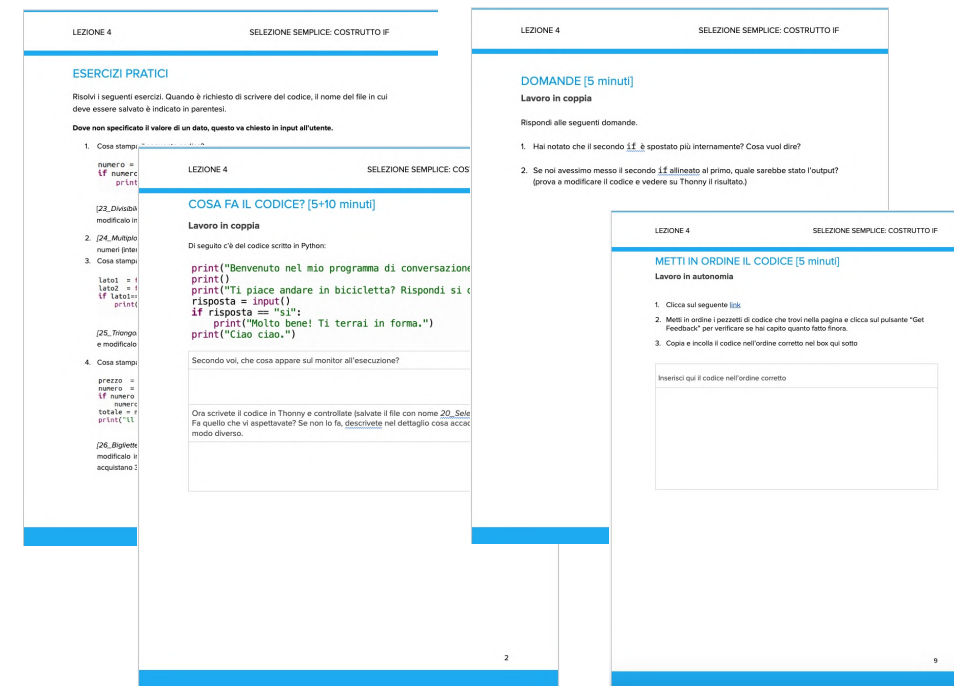


Schede didattiche

Sono state progettate delle schede di lavoro basate sul metodo PRIMM

IN CLASSE

Svolgimento lezioni



Schede didattiche

Sono state progettate delle schede di lavoro basate sul metodo PRIMM

STRUTTURA DELLE SCHEDE DIDATTICHE



STRUTTURA DELLE SCHEDE DIDATTICHE

PREDICT

Proposte di codice con nuovi concetti su cui **prevedere il funzionamento**

- Viene presentato un codice contenente i concetti della lezione prima di averli spiegati
- Viene chiesto di predire il risultato dell'esecuzione rispondendo a parole

LEZIONE 4

SELEZIONE SEMPLICE: COSTRUTTO IF

COSA FA IL CODICE? [5+10 minuti]

Lavoro in coppia

Di seguito c'è del codice scritto in Python:

```
print("Benvenuto nel mio programma di conversazione")
print()
print("Ti piace andare in bicicletta? Rispondi si o no")
risposta = input()
if risposta == "si":
    print("Molto bene! Ti terrai in forma.")
print("Ciao ciao.")
```

Secondo voi, che cosa appare sul monitor all'esecuzione?

STRUTTURA DELLE SCHEDE DIDATTICHE

RUN

A seguire, **l'esecuzione del codice** proposto per verificare la propria predizione

- Successivamente viene chiesto di ricopiare o di copiare il codice
- Eseguirlo
- Verificare il risultato con quanto predetto prima
- Nel caso in cui i risultati non coincidessero, è chiesto di scrivere cosa è cambiato e riflettere

LEZIONE 4

SELEZIONE SEMPLICE: COSTRUTTO IF

COSA FA IL CODICE? [5+10 minuti]

Lavoro in coppia

Di seguito c'è del codice scritto in Python:

```
print("Benvenuto nel mio programma di conversazione")
print()
print("Ti piace andare in bicicletta? Rispondi si o no")
risposta = input()
if risposta == "si":
    print("Molto bene! Ti terrai in forma.")
print("Ciao ciao.")
```

Secondo voi, che cosa appare sul monitor all'esecuzione?

Ora scrivete il codice in Thonny e controllate (salvate il file con nome 20_Selezione.py).
Fa quello che vi aspettavate? Se non lo fa, descrivete nel dettaglio cosa accade in modo diverso.

STRUTTURA DELLE SCHEDE DIDATTICHE

INVESTIGATE

Domande esplorative sui nuovi concetti spiegando a parole quanto capito
Esercizi **"trova gli errori"** di sintassi, esecuzione e semantica
Esercizi di **riordino del codice**

- Una volta che si è riflettuto sull'esecuzione del codice, si sono proposti degli esercizi di esplorazione del codice
 - Domande esplorative sui concetti, sul funzionamento, sulla sintassi
 - "Trova gli errori"
 - Riordino del codice

LEZIONE 4

SELEZIONE SEMPLICE: COSTRUTTO IF

LEZIONE 4

SELEZIONE SEMPLICE: COSTRUTTO IF

LEZIONE 4

SELEZIONE SEMPLICE: COSTRUTTO IF

TROVA GLI ERRORI [3 minuti]

Lavoro in coppia

In questo programma ci sono **2 errori**, riesci ad individuarli? Evidenziali con un cc
(Se non riesci a trovare alcuni errori puoi aiutarti copiando il codice in Thonny e verificandone il funzionamento)

```
print("Ciao! sono un po' curioso.")
risposta = input("Ti piace la pizza? ")
if risposta > "si"
    print("Bene. Allora possiamo fare una serata pizza e cinema.")
print("A presto.")
```

METTI IN ORDINE IL CODICE [5 minuti]

Lavoro in autonomia

1. Clicca sul seguente [link](#)
2. Metti in ordine i pezzetti di codice che trovi nella pagina e clicca sul pulsante "Get Feedback" per verificare se hai capito quanto fatto finora.
3. Copia e incolla il codice nell'ordine corretto nel box qui sotto

Inserisci qui il codice nell'ordine corretto

DOMANDE [5 minuti]

Lavoro in coppia

Rispondi alle seguenti domande.

1. Hai notato che il secondo `if` è spostato più internamente? Cosa vuol dire?
2. Se noi avessimo messo il secondo `if` allineato al primo, quale sarebbe stato l'output?
(prova a modificare il codice e vedere su Thonny il risultato.)

STRUTTURA DELLE SCHEDE DIDATTICHE

MODIFY

Alcuni esercizi di **modifica del codice**

- Dopo la fase di esplorazione, si passa alla modifica del codice:
 - Si presenta un codice ai ragazzi
 - Si chiede loro di capire cosa fa e che problema risolve
 - Successivamente si chiede di modificarlo in modo tale che venga soddisfatta una nuova richiesta.

ESERCIZI PRATICI

4. Cosa stampa il seguente codice?

```
prezzo = float(input("inserisci il prezzo del biglietto"))
numero = int(input("inserisci il numero di biglietti acquistati"))
if numero > 20:
    numero = numero-1
totale = numero * prezzo
print("il totale da pagare è", totale)
```

[26_Biglietteria.py] Ricopia il codice per accertarti di aver risposto correttamente e modificalo in modo tale che venga regalato un biglietto omaggio **ogni** 20 biglietti (se si acquistano 30 biglietti se ne pagano 29, acquistandone 50 se ne pagano 48, ecc) .

- Vengono proposti
 - esercizi pratici di scrittura di un codice dall'inizio che risolva un dato problema
 - oppure esercizi di scrittura di codice libero, che lasciano i ragazzi spaziare anche con la loro creatività

ESERCIZI PRATICI

Risolvi i seguenti esercizi. Quando è richiesto di scrivere del codice, il nome del file in cui deve essere salvato è indicato in parentesi.

Dove non specificato il valore di un dato, questo va chiesto in input all'utente.

1. *[45_NomeGenerazione.py]*

Chiedere all'utente l'anno di nascita e mostrare a video il nome della sua generazione: se è nato prima del 1946, mostrare "Prima dei Boomer"; se è nato tra il 1946 e il 1964, mostrare "Boomer"; se è nato tra il 1965 e il 1980, mostrare "Generazione X"; se è nato tra il 1981 e il 2000, mostrare "Generazione Y (Millennials)". In ultimo, se è nato dopo il 2000, indicare la "Generazione Z".

ESERCIZI PRATICI

Risolvi i seguenti esercizi. Quando è richiesto di scrivere del codice, il nome del file in cui deve essere salvato è indicato in parentesi.

4. *[104_SonoUnArtista.py]* Disegna qualcosa a tua scelta utilizzando tutto quanto visto fino ad ora relativamente alla libreria turtle.

STRUTTURA DELLE SCHEDE DIDATTICHE

REVIEW

Revisione dei **concetti**

- Per fissare ulteriormente i concetti si sono proposti i "termini chiave" a fine di ogni lezione e brevi accorgimenti "da ricordare" in Python.

LEZIONE 4

SELEZIONE SEMPLICE: COSTRUTTO IF

TERMINI CHIAVE DELLA LEZIONE

TERMINE	SIGNIFICATO	PYTHON
Operatore relazionale	Un simbolo che serve per fare un confronto tra due dati.	>, >=, <, <=, ==, != sono gli operatori relazionali di Python
Condizione	Il test che inseriamo nel costrutto <i>if</i> . Si costruisce inserendo un <i>operatore relazionale</i> tra due dati. Il risultato della condizione deve essere vero (True) o falso (False)	risposta == "sì" a > 5 b != 7 10 >= x b == a
Selezione	Quando nel codice c'è un	if a > 10:

LEZIONE 4

SELEZIONE SEMPLICE: COSTRUTTO IF

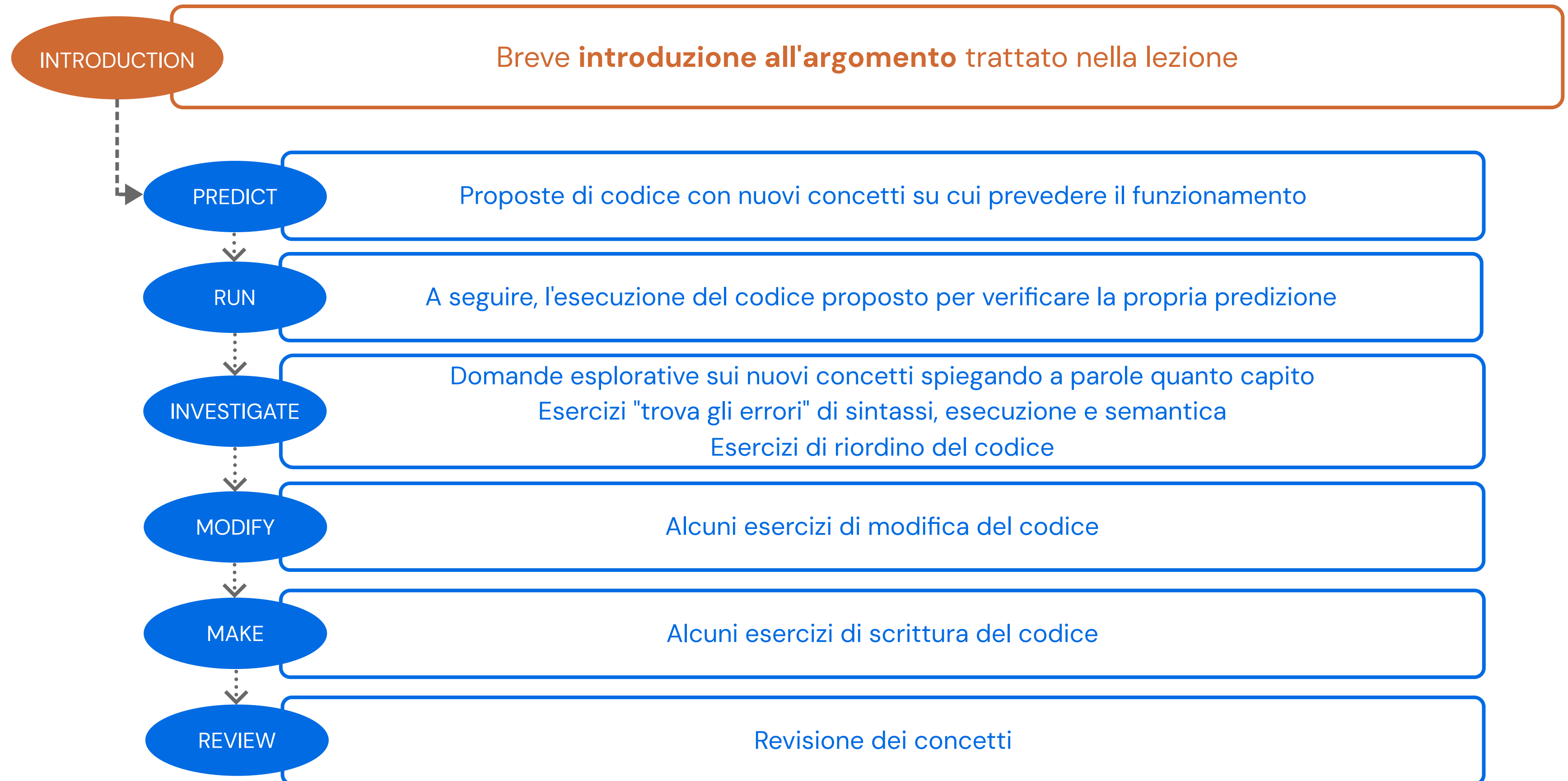
DA RICORDARE IN PYTHON

I due punti : sono usati per annunciare che sta arrivando un nuovo blocco di codice e tutte le righe del blocco **DEVONO** essere rientrate (indentate) della stessa quantità di spazi. Alla fine del blocco, puoi interrompere il rientro. Tutto ciò che accade se una condizione è vera dopo un'istruzione *if*, è anche un blocco di codice.

Esempio:

```
if risposta=="si":  
    print("ciao")  
    print("arrivederci")
```

STRUTTURA DELLE SCHEDE DIDATTICHE: AGGIUSTAMENTI NEL PERCORSO



STRUTTURA DELLE SCHEDE DIDATTICHE: AGGIUSTAMENTI NEL PERCORSO

INTRODUCTION

Breve **introduzione all'argomento** trattato nella lezione e ripasso

- Si è notato che i ragazzi avevano bisogno di una breve spiegazione iniziale che li introducesse all'argomento per poter affrontare meglio la parte di predizione.

LEZIONE 13

STRINGHE

IL TIPO DI DATO “STRING”

Finora abbiamo imparato che in programmazione esistono diversi tipi di dato (`int`, `float`, `bool`, `string`). In particolare, sappiamo che per rappresentare **parole, caratteri alfabetici e punteggiatura** dobbiamo utilizzare le stringhe.

Esempio: `messaggio = "Ciao come va? lo tutto bene."`

In questa lezione andremo ad approfondire le **stringhe** considerandole **come** una **sequenza di caratteri**, a ciascuno dei quali, si associa un indice che inizia da 0 (come le liste viste nella lezione 12).

Le conclusioni

05

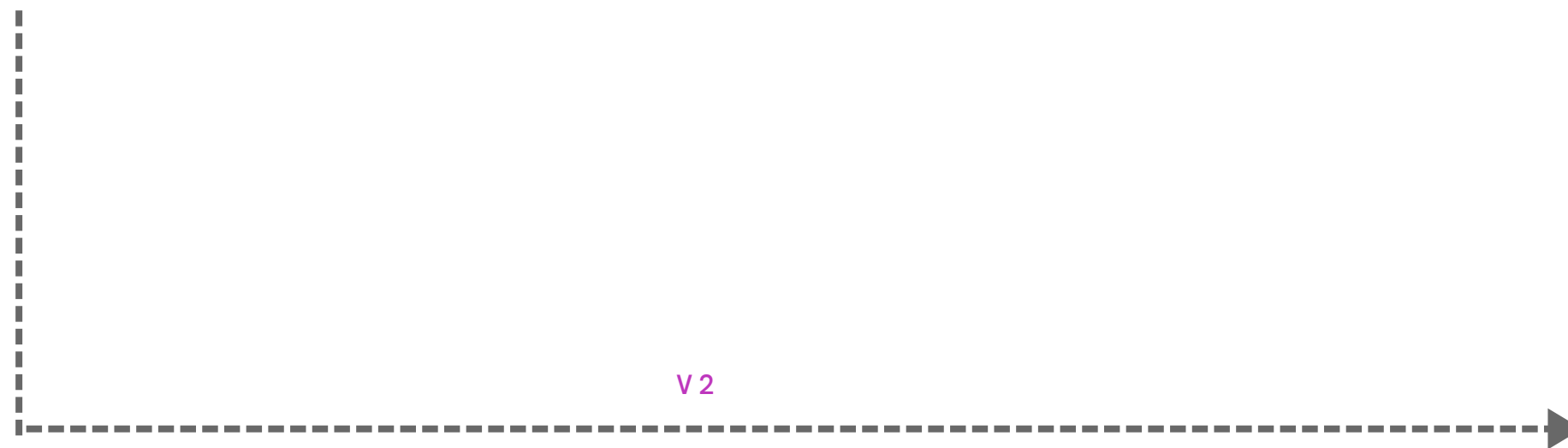
Le conclusioni e gli sviluppi futuri

La prima iterazione:

- **risultati positivi**
- **necessità di ampliare il campione**

Si sta lavorando alla seconda versione:

- miglioramenti dopo alcuni feedback



INTRODUZIONE

In questa lezione vedremo come gestire le variabili di diverso tipo e gli operatori disponibili in Python. Il **tipo** di una variabile indica che dato abbiamo intenzione di metterci dentro. I tipi principali che vedremo in questa lezione sono il numero **intero** (in inglese *integer*, abbreviato *int*), il numero **decimale/reale** (in inglese *floating point*, abbreviato *float*) e il tipo **testuale**, che in informatica viene detto *stringa* (in inglese *string*, abbreviato *str*).

COSA FA IL CODICE (1)? [10 minuti]

Lavoro in coppia

Di seguito c'è del codice scritto in Python:

```
n1 = input("scrivi un numero: ")
n2 = input("scrivi un altro numero: ")
risultato1 = n1+n2
print("il primo risultato è:", risultato1)
risultato2 = n1*3
print("il secondo risultato è:", risultato2)
```

Secondo voi, che cosa appare sul monitor all'esecuzione?

Ora copiate il codice in Thonny (salvate con nome [04_Concatenazione.py](#)) e controllate cosa fa. Fa quello che vi aspettate? Se non lo fa, cosa c'è di diverso rispetto alla previsione?

I materiali

06

DOWNLOAD MATERIALI



Sito FABLAB Unitrento

A screenshot of the FabLab UniTrento website. The header includes the site name 'FabLab UniTrento' and navigation links for 'Events', 'Tools & Machines', 'Projects', 'Activities', 'Join us', and 'Who we are'. There are also social media icons for Facebook, Instagram, and a share icon. The main content area discusses the PRIMM methodology for teaching programming, starting with a paragraph about its purpose and a list of five phases: Predict, Run, Investigate, Modify, and Make. Below this is a paragraph about the theoretical perspective and another about group work. A 'Materiali:' section lists two resources: an article from ITADINFO'23 and a set of didactic materials.

FabLab UniTrento

Events Tools & Machines Projects Activities Join us Who we are

PRIMM mira a ridurre la scrittura di codice da parte dello studente: questo avviene con metodologie utilizzate nell'apprendimento delle lingue (comprensione madrelingua), dove l'attività di produzione è preceduta dalla lettura e dalla comprensione del testo.

Partendo dall'osservazione che il carico cognitivo necessario per arrivare alla scrittura del codice è molto elevato, la metodologia PRIMM inverte la sequenza tradizionale, partendo prima dalla comprensione del testo e solo successivamente arrivando alla scrittura di codice vero e proprio.

- **Predict:** gli studenti leggono un segmento di codice, scritto in Python, e fanno previsioni su cosa farà il codice quando viene eseguito;
- **Run:** gli studenti eseguono il codice proposto nell'attività precedente, confrontando il comportamento effettivo rispetto alla previsione;
- **Investigate:** agli studenti è richiesto di analizzare più in profondità il codice o sue varianti, utilizzando varie tipologie di esercizi, come per esempio la correzioni di bug, l'annotazione del codice, l'uso di Parson Puzzle, domande esplorative, etc;
- **Modify:** agli studenti è chiesto di modificare il codice, partendo da variazioni molto semplici e poi con modifiche sempre più complesse;
- **Make:** infine, gli studenti creano un programma totalmente nuovo, ispirandosi al codice visto in precedenza, ma realizzando una nuova funzionalità o risolvendo un problema diverso.

Dal punto di vista teorico, PRIMM si colloca nell'intersezione fra un approccio più strutturato basato sulla scoperta guidata e sull'istruzione diretta, e un approccio più costruzionista basato sulla scoperta pura e su problemi aperti. Le cinque fasi partono da attività fortemente guidate ad attività completamente libere, riducendo mano a mano il livello di scaffolding.

Per quanto possibile, molte delle attività si svolgono favorendo il lavoro di gruppo e lo scambio fra pari, seguendo l'idea che l'articolazione verbale delle soluzioni favorisce in generale l'apprendimento, in particolar modo nell'ambito della programmazione dove la componente linguistica è fondamentale.

Materiali:

- Un articolo che descrive gli esiti della sperimentazione: Giulia Peserico, Maria Serafini, Francesca Voltolini, Federica Picasso, Daniele Agostini, Francesca Fiore, Anna Serbati, Alberto Montresor. [Sperimentazione del metodo PRIMM per l'insegnamento della programmazione](#). Atti del 1° Convegno Italiano sulla Didattica dell'Informatica (ITADINFO'23), Bari, Ottobre 2023.
- Un insieme di materiali didattici pronti all'uso. A titolo di esempio, [qui](#) trovate la scheda per quanto riguarda l'insegnamento dell'if-then-else a più vie. Se siete interessati a ricevere tutti i materiali, siete pregati di compilare questo [modulo di richiesta](#). Riceverete i materiali via email.

grazie per l'attenzione!