

CODING

Sviluppare il pensiero computazionale e la creatività: coding

Perché a Scuola si dovrebbe imparare a programmare, anche senza aver l'obiettivo di formare informatici e programmatori? Perché il pensiero logico che sottende l'attività di programmazione utilizza processi utili in ogni disciplina e in ogni momento della vita e perché le conoscenze della disciplina "Informatica" devono ormai – nella società dell'informazione e della comunicazione – essere un patrimonio di tutti. Quali competenze per i docenti?

I docenti della Scuola digitale, di ogni disciplina sanno riconoscere il valore formativo delle attività di programmazione e sanno progettare (o collaborare a progettazioni per la propria parte disciplinare) lezioni innovative finalizzate a condurre gli studenti a produrre "artefatti" esito di una attività di programmazione: storytelling, giochi, animazioni, app interattive, programmi che guideranno le azioni di robot, stampanti 3D,..

A questo fine è necessario che i docenti siano consapevoli dei concetti e pratiche computazionali sottese all'uso dei linguaggi di programmazione: affinché l'uso dei programmi più diffusi per la realizzazione di attività di coding, non sia solo una attività di allenamento al pensiero logico, ma diventi occasione per una formazione esplicita al linguaggio e al metodo di lavoro della disciplina informatica.

Il docente competente nell'uso degli strumenti e nelle metodiche della programmazione e del coding è in grado di:

TEMI	Il docente è in grado di:	ELEMENTI DI EVIDENZA
1. Aspetti generali di contesto per attività di coding a Scuola	<p>1.1. Indicare gli elementi di valore educativo e formativo delle attività didattiche basate sul coding per lo sviluppo di pensiero logico.</p> <p>1.2. Progettare attività didattiche disciplinari utilizzando il processo e gli strumenti del coding come tecnologia didattica per attività formazione e di apprendimento: storytelling, giochi, animazioni, app interattive.</p> <p>1.3 Scegliere le strategie didattiche per avviare gli studenti alla programmazione non specialistica: remix, riordino, descrizioni testuali, debugging,</p>	<p>1.1. 1Contenuti per comunicare in modo esplicito ai colleghi gli elementi che li spingono a scegliere attività di programmazione come strumento formativo per gli studenti (allenamento alle competenza pensiero logico, conoscenza del linguaggio e degli elementi essenziali di un settore imprescindibile dell'oggi).</p> <p>1.2.1/ 1.3.1 /1.4.1 Lesson plan con chiara descrizione delle strategie didattiche e prodotti richiesti agli studenti</p> <p>1.2.1/ 1.3.1/ 1.4.1. Lesson plan che esplicitano le attività da svolgersi spiccatamente disciplinari concorrenti alla realizzazione di un prodotto di codice.</p>

CODING

TEMI	Il docente è in grado di:	ELEMENTI DI EVIDENZA
	1.4 Progettare attività didattiche basate sulla programmazione collaborando con colleghi di diverse discipline. 1.5 Introdurre agli studenti la disciplina "Informatica" mettendo in evidenza come all'interno di essa si collochi la scrittura di codice. 1.6 Introdurre gli studenti al concetto di calcolatore: la struttura e il linguaggio	1.5.1 Contenuti didattici per fornire agli studenti – in base al grado di scuola – una descrizione della disciplina "Informatica" 1.6.1 Contenuti didattici per fornire agli studenti – in base al grado di scuola – una descrizione della struttura di un calcolatore e del concetto di linguaggi di programmazione.
2. Progettare e condurre attività di programmazione da far svolgere agli studenti	2.1 Introdurre agli studenti il concetto di algoritmo con esempi tratti dall'esperienza quotidiana e scolastica (pensiero logico e categorizzazione). 2.2 Guidare gli studenti a realizzare rappresentazioni grafiche di algoritmi 2.3 Progettare attività didattiche in modo incrementale rispetto all'ordine di scuola in relazione ai concetti e alle pratiche computazionali, avendo come riferimento i sillabi sull'informatica a scuola. 2.4 Condurre gli studenti a porre attenzione al concetto di errore quale pratica formativa e per la scrittura di algoritmi sempre più efficaci ed efficienti	2.1.1 / 2.2.1 Contenuti didattici (formativi e per attività di apprendimento) per guidare gli studenti – in base al grado di scuola – a comprendere e descrivere attività, situazioni, compiti in termini di algoritmo. 2.3.1 Piani di lezioni e materiali didattici esplicitamente calibrati in base agli obiettivi formativi per ordine di scuola facendo riferimento ai concetti e alle pratiche computazionali. 2.4.1 Piani di lezione che contemplino in modo sistematico una attenzione specifica al processo di debugging e alla riflessione e condivisione dei criteri che rendono un algoritmo / programma efficace ed efficiente.
3. Progettare e applicare la Valutazione	3.1 Definire rubriche di valutazione per una valutazione della qualità del processo di programmazione realizzato dagli studenti 3.2 Riconoscere e fare una evidenza delle competenze esercitate dagli studenti sulla base dei sillabi di riferimento (es.: linee guida CINI per l'informatica a Scuola).	3.1.1. Scrivere in base alle consegne fornite agli studenti rubriche di valutazione che guidano gli studenti a comprendere e cercare di raggiungere i livelli di prestazione migliori 3.2.1 Utilizzare o creare sistemi di elaborazione dei dati delle prestazioni degli studenti collegandole alle competenze descritte nei sillabi di riferimento per le competenze di informatica degli studenti.



CERTIFICAZIONE PEDAGOGICA EUROPEA
SULL'USO DELLE TECNOLOGIE DIGITALI

CODING

