

**Titolo corso:** *Insegnare coding (corso base o avanzato) – ore da concordare*

*Gli obiettivi previsti dal Piano Nazionale Scuola Digitale (azione #17) puntano a "permettere a ogni studente della scuola primaria di svolgere un corpus di 10 ore annuali di logica e pensiero computazionale" e (azione #18) di "potenziare le attività laboratoriali e aggiornare il curriculum di tecnologia della scuola secondaria di primo grado".*

*A partire da queste esigenze, i nostri corsi base e avanzato vogliono mettere docenti e studenti nelle condizioni di raggiungere tali obiettivi.*

**Competenze attese in uscita dal corso base (Introduzione al coding)**

- Comprendere i principi base del coding, attraverso un inquadramento teorico/didattico;
- Conoscere le basi di programmazione e saper lavorare con code.org.

**Competenze attese in uscita dal corso avanzato (Fare coding)**

- Saper attivare metodologie didattiche legate all'utilizzo del coding con gli studenti;
- Saper utilizzare i principali strumenti legati al coding disponibili online, con particolare riferimento a code.org, all'animazione (Scratch) e modellazione 3D.

**Obiettivi specifici e descrizione attività**

Il corso di formazione ha come obiettivo – nel corso base – di introdurre i docenti alle tematiche del coding (dall'inglese, significa scrittura di codice, ovvero in linguaggio informatico programmazione) proponendo un inquadramento teorico/didattico e fornendo esempi concreti di attività di avvicinamento alle tematiche, da fare anche con i più piccoli, per approdare all'ambiente guidato di code.org.

Nel corso avanzato l'obiettivo è fornire ai docenti un utilizzo consapevole degli strumenti e dei software per essere in grado di organizzare delle attività didattiche di coding con gli studenti, in particolare lavorando sull'animazione e modellazione 3D.

Con tali attività si vuole mettere in condizione i docenti di affrontare gli obiettivi previsti dal Piano Nazionale Scuola Digitale (azione #17) di "permettere a ogni studente della scuola primaria di svolgere un corpus di 10 ore annuali di logica e pensiero computazionale" e (azione #18) di "potenziare le attività laboratoriali e aggiornare il curriculum di tecnologia della scuola secondaria di primo grado".

Il Piano ritiene fondamentale partire dagli studenti più giovani per due ragioni:

- anticipare la comprensione della logica della Rete e delle tecnologie, in quanto l'avvicinamento alle tecnologie stesse avviene sempre più precocemente;
- preparare da subito gli studenti allo sviluppo delle competenze che sono al centro del nostro tempo, e saranno al centro delle loro vite e carriere.

Per tali motivi, e anche per ragioni di continuità, è stato posto un focus sulle classi della SP e della SSPG in ottica di sviluppo del pensiero computazionale dello studente e delle competenze derivanti da esso.

Sviluppare il pensiero computazionale infatti significa aumentare la capacità di analizzare le situazioni, valutarne i limiti, conoscere gli strumenti a disposizione, organizzare strategie efficaci di soluzione alle problematiche. Sviluppare tale pensiero è un bisogno formativo delle future generazioni, adatto a sviluppare una didattica delle competenze.

Il corso base non prevede conoscenze pregresse da parte dei docenti, può essere modulato in base alle esigenze della scuola, e conterrà esercitazioni pratiche per le quali in alcune delle sue ore sarà fondamentale la presenza di un'aula informatica o l'utilizzo dei pc/device degli insegnanti.

Nel corso avanzato, l'attività prevede conoscenze di base di coding da parte dei docenti, può essere modulato in base alle esigenze della scuola, e conterrà esercitazioni pratiche per le quali in alcune delle sue ore sarà fondamentale la presenza di un'aula informatica o l'utilizzo dei pc/device degli insegnanti.

## **Dettaglio contenuti del corso base (Introduzione al coding)**

### **1. Introduzione al coding**

- a. Computer Science nella didattica
- b. Il pensiero computazionale e le competenze
- c. Che cos'è il coding
- d. Un approccio al coding: l'applicazione TripGeo
- e. La creazione di percorsi animati (Blockly Games Labirinto)

### **2. Primi concetti di programmazione**

- a. Programmare a blocchi
- b. Un esempio didattico: Zimmer Twins
- c. Programmare senza computer: le attività unplugged

### **3. Utilizziamo code.org**

- a. Code.org: percorsi a tema per imparare il coding
- b. Fare coding con code.org
- c. Fare coding con Scratch Junior

## **Dettaglio contenuti del corso avanzato (Fare coding)**

### **1. Il coding con code.org**

- a. La computer science: acquisire competenze
- b. Gli elementi di base della programmazione tramite code.org
- c. Laboratorio: sperimentiamo con code.org

### **2. Utilizziamo Scratch (Prima parte)**

- a. Dal pensiero computazionale al pensiero divergente
- b. Il coding con Scratch
- c. Laboratorio: creiamo una storia in aula

### **3. Utilizziamo Scratch (Seconda Parte) e sperimentiamo la modellazione 3D**

- a. L'interattività tramite Scratch
  - b. Laboratorio: creiamo un gioco con Scratch
  - c. Impariamo a disegnare oggetti in tre dimensioni propedeutici alla stampa 3d
- Per ognuna delle attività descritte, sono previsti momenti di applicazione pratica per i partecipanti, a gruppi in un'aula informatica dotata di più postazioni o individualmente alla LIM.

## **Per informazioni e contatti**

Pre/Post Sales Coordinator  
Sales&Services School&ELT&HE  
Pearson Italia Spa  
T: 011.750.21.529/518  
F: 011.750.21.510  
e-mail: [info.corsi@pearson.it](mailto:info.corsi@pearson.it)  
sito: [www.pearson.it](http://www.pearson.it)