



INTRODUZIONE AL PENSIERO COMPUTAZIONALE

DOCENTE CASTIELLO RAFFAELLA

Cos'è il pensiero computazionale?

Il concetto di PC è stato introdotto per la prima volta da Seymour Papert nel 1996 parlando di LOGO , il linguaggio di programmazione sviluppato al MIT di Boston per insegnare la programmazione ai bambini. Papert padre del costruzionismo sostiene che la mente umana per poter imparare bene abbia bisogno di costruire artefatti (learning by doing). La definizione che è stata maggiormente condivisa dagli esperti è quella coniata dalla Dott.ssa Jeanette Wing (direttrice di dipartimento di informatica USA) che ha definito il PC come “ processo mentale che sta alla base della formulazione dei problemi e delle soluzioni così che possa essere implementata in maniera efficace da un elaboratore di informazioni sia umano che artificiale”.

Negli ultimi anni, le teorie innovative di Seymour Papert sono state riprese, ampliate e approfondite dal MIT Media Lab di Boston USA che ha sviluppato il programma Scratch. Il pensiero computazionale come strumento metodologico perche'

- ha ricadute nell'ambito logico-matematico e nella risoluzione dei problemi contribuendo allo sviluppo del pensiero logico
- il saper scrivere e leggere sono gia' sono operazioni algoritmiche
- Il saper tradurre idee in codice è espressione di creatività ed incentiva la responsabilizzazione ed il ruolo attivo dell'individuo.

Per un bambino, imparare a programmare concorre al raggiungimento delle competenze digitali nella sua componente scientifica e tecnologica .I ragazzi sanno fare un uso esclusivamente strumentale della tecnologia ma non ne conoscono in modo approfondito le potenzialità e sono soggetti passivi nella fruizione dei dispositivi.



DETTO IN PAROLE SEMPLICI

QUANDO AFFRONTIAMO UN PROBLEMA O ABBIAMO UN' IDEA, SPESSO INTUIAMO LA SOLUZIONE MA NON SIAMO IN GRADO DI FORMULARLA IN MODO OPERATIVO PER METTERLA IN PRATICA. IL PENSIERO COMPUTAZIONALE E' QUESTO: LA CAPACITÀ DI DESCRIVERE UN PROCEDIMENTO COSTRUTTIVO CHE PORTI AD UNA SOLUZIONE CREATIVA , EFFICACE E NON AMBIGUA

Coding e pensiero computazionale

CODING: indica l'uso di strumenti e metodi di programmazione visuale a blocchi per favorire lo sviluppo del pensiero computazionale

PENSIERO COMPUTAZIONALE : e' la capacita' di individuare un procedimento costruttivo, fatto di passi semplici che ci porta alla soluzione di un problema complesso. La programmazione in blocchi e' un metodo di rappresentazione che ci permette di concatenare una sequenza di azioni utilizzando blocchi colorati.

Esistono vari modi per esprimere un procedimento ma questa modalita' e' intuitiva, divertente e rigorosa perche' richiama al funzionamento per istruzioni dei computer.

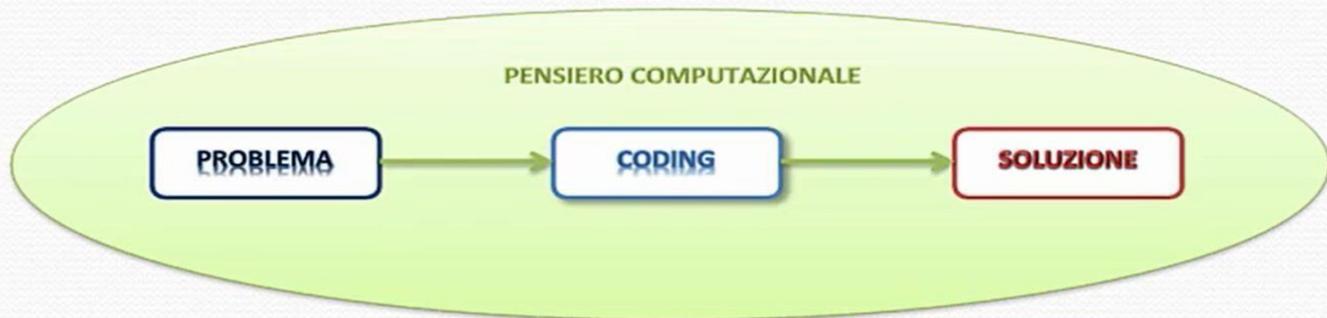
Non dobbiamo confondere il coding con la programmazione. L'obiettivo non è diffondere la programmazione propria della professione degli informatici ma avviare le generazioni future al pensiero computazionale come processo logico di realizzazione di un 'idea in modo creativo .Il pensiero computazionale viene prima di qualsiasi competenza digitale in quanto offre gli strumenti per acquisire ulteriori competenze in modo consapevole.

Saper programmare è importante quanto conoscere una lingua straniera o come imparare a leggere e scrivere (Mitch Resnick), perché la lingua dei computer è parlata in tutto il mondo .

Pensiero computazionale

E' un **processo mentale** che consente di risolvere problemi di varia natura seguendo metodi e strumenti specifici.

Quando parliamo di pensiero computazionale nella didattica, ci riferiamo a quella metodologia basata sulla risoluzione dei problemi (**problem solving**) mediante un approccio tipicamente **informatico**.

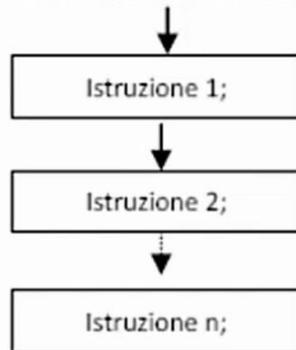


ALGORITMI

- La **soluzione di un problema** deve passare attraverso il pensiero algoritmico.
- Un **algoritmo** è una sequenza di passi che devono essere eseguiti secondo un ordine prefissato per raggiungere il risultato voluto.
- Un algoritmo può essere rappresentato mediante schemi a blocchi ed è stato dimostrato (Teorema di Jacopini-Bohm) che qualunque algoritmo si basa su tre strutture fondamentali: **sequenziale, condizionale, iterativa.**

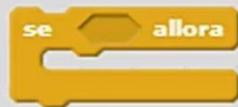
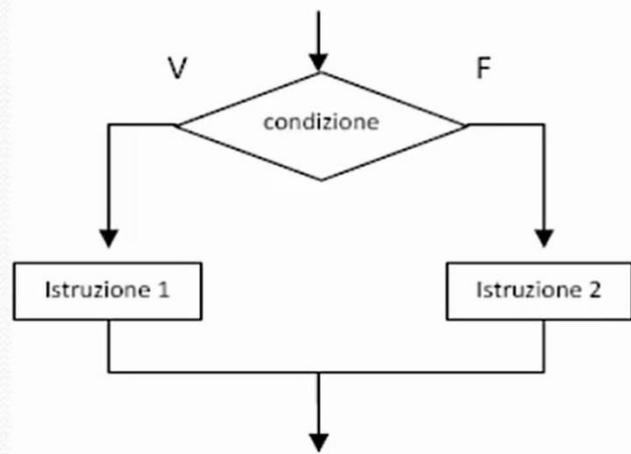
STRUTTURA SEQUENZIALE

SCHEMA A BLOCCHI



STRUTTURA DI SELEZIONE

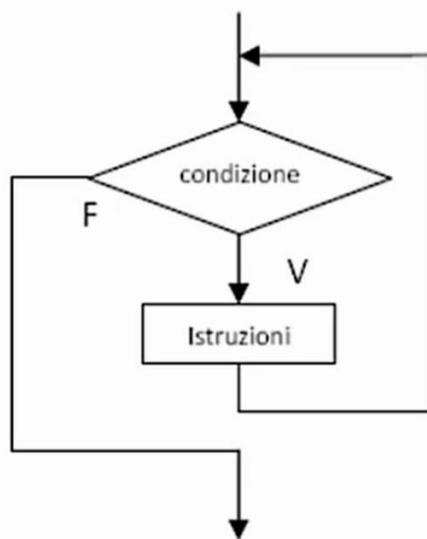
SCHEMA A BLOCCHI



STRUTTURA ITERATIVA

Ripetizione delle istruzioni. Ciclo.

SCHEMA A BLOCCHI



ripeti fino a quando

ripeti 10 volte

per sempre

Allenati per riuscire
nell'impresa.



LE PERSONE USANO IMPLICITAMENTE IL PENSIERO COMPUTAZIONALE
NELLA VITA DI TUTTI I GIORNI.

SONO CHIAMATE AD ESPLICITARLO QUANDO DEVONO ISTRUIRE UN
SOGGETTO TERZO (L'ESECUTORE) A RISOLVERE UN PROBLEMA

alcuni esempi :

Il ragionamento algoritmico è presente:

1. quando organizziamo la nostra giornata
2. quando condividiamo i passi di una ricetta che abbiamo sperimentato
3. quando facciamo le operazioni aritmetiche
4. quando spieghiamo un gioco agli altri
5. quando dobbiamo fornire delle istruzioni per raggiungere un luogo
6. quando vogliamo costruire una mappa concettuale...

PENSIERO COMPUTAZIONALE INFORMALE





PROBLEMA

DEVI USCIRE IMPROVVISAMENTE E ...
DEVI LASCIARE LE ISTRUZIONI A TUO
FIGLIO AFFINCHÉ 'RIESCA A CENARE
DA SOLO IN TUA ASSENZA !!!!!!!

CHE FARE ?

OPPURE

CARICARE LA LAVATRICE
FARE IL CAFFÈ
PREPARARSI UN PANINO
FARE LA CARTELLA



<http://link-and-think.blogspot.it/2016/04/cinque-parole-chiave-pensiero-computazionale.html>

Enrico Nardelli – Creative Commons BY-NC-SA 4.0

1) POTENZA COMPUTAZIONALE

Quanti anni ha tuo figlio? Cosa e' in grado di eseguire? Cosa comprende ?

Si deve adeguare il linguaggio e le istruzioni alle capacita'dell'esecutore

2) LIVELLO DI ASTRAZIONE

Come gli spieghi le cose? il linguaggio deve essere dettagliato?

es" riempi la pentola di acqua"

"riempi la pentola a meta'" Metti la pentola sotto il rubinetto, apri l'acqua fredda, chiudi quando l'acqua raggiunge il livello "

3) DECOMPOSIZIONE

Scomporre il problema in tanti piccoli problemi. In questa fase la decomposizione del problema dipende dalla potenza e dal livello di astrazione dell'esecutore :
mettere la tovaglia, sistemare posate e piatti

4) ALGORITMO

Le istruzioni elementare devono essere sequenziate in un ordine logico e preciso per risolvere in maniera efficace il problema

5) VERIFICA

Verificare la correttezza della procedure e delle istruzioni mettendosi nei panni dell'esecutore , importanza del debugging cioe' dell'errore e riformulazione delle procedure

Concetti chiave del pensiero computazionale

1) RAGIONAMENTO LOGICO: PERMETTE ALL'INDIVIDUO DI INTERROGARSI SUL SENSO DELLE COSE, COGLIERE IL PERCHÉ, ANALIZZANDO, VERIFICANDO, E INTEGRANDO NUOVE CONOSCENZE SU QUELLE GIÀ POSSEDUTE. IMPORTANZA DEL DEBUGGING ; CIOÈ NELLA PROGRAMMAZIONE RIMUOVERE L'ERRORE E TROVARE LE CORRETTE PROCEDURE

2) SCOMPOSIZIONE : SUDDIVIDERE PROBLEMI GRANDI E DIFFICILI IN QUALCOSA DI MOLTO PIÙ SEMPLICE. SPESSO GRANDI PROBLEMI SONO TANTI PICCOLI PROBLEMI INSIEME (STEP BY STEP) E QUESTO APPROCCIO PERMETTE DI COGLIERE MEGLIO LE SITUAZIONI NUOVE ED AFFRONTARLE CON MENO PAURE.

3) SCHEMI RICORRENTI O GENERALIZZAZIONE: INDIVIDUARE LE REGOLARITA' E LA RICORSIVITA' CON ALTRI PROBLEMI ANALOGHI E CERCARE SOMIGLIANZE O ANALOGIE, CONNESSIONI CON ALTRE ESPERIENZE PRECEDENTI. " E' SIMILE AD UN PROBLEMA CHE HO GIA' RISOLTO?" IN COSA E' DIVERSO? " CIO' AIUTA LA METACOGNIZIONE E L'AUTOCONSAPEVOLEZZA. ALGORITMI CHE SONO STATI UTILIZZATI PER RISOLVERE IL PROBLEMA POSSONO ESSERE ADATTATI ALLA NUOVA SITUAZIONE.

4)ASTRAZIONE : ISOLARE I DETTAGLI E FOCALIZZARE L'ATTENZIONE SULLO SCHEMA GENERALE PER FACILITARNE LA COMPrensIONE E LA RISOLUZIONE

5)PENSIERO ALGORITMICO E' IL MODO DI ARRIVARE AD UNA SOLUZIONE ATTRAVERSO UNA CHIARA DEFINIZIONE DEI PASSAGGI. ALCUNI PROBLEMI NON DEVONO ESSERE ANALIZZATI DI NUOVO QUANDO SI PRESENTANO PERCHE' SE SI E' UTILIZZATO UN ALGORITMO EFFICACE SI ARRIVA FACILMENTE ALLA SOLUZIONE (REGOLE DI MATEMATICA O DI ITALIANO)

6)VERIFICA : VERIFICARE CHE LA PROCEDURA UTILIZZATA SIA EFFICACE. COME?

-CORRETTEZZA

- E' FACILMENTE UTILIZZABILI DA PIU' ESECUTORI

- NON IMPLICA DISPENDIO DI RISORSE ED ENERGIE

-VARIABILE TEMPO (E' ABBASTANZA VELOCE O CI SONO TROPPI PASSAGGI DA EFFETTUARE?)

RIASSUMIAMO

*IL PENSIERO COMPUTAZIONALE RICORRE A SCHEMI RICORRENTI PER FACILITARE LA
RISOLUZIONI DI PROBLEMI DA PARTE DELL'ESECUTORE*

*NON PROBLEM SOLVING IN PRIMA PERSONA MA PER QUALCUNO CHE DEVE AGIRE AL TUO
POSTO PER RISOLVERE IL PROBLEMA AUTONOMAMENTE (ESECUTORE)*

*VA AL DI LA' DELLE TECNOLOGIE O DEL LINGUAGGIO INFORMATICO SPECIFICO DI
PROFESSIONISTI :*

*E' UNA ABILITA' TRASVERSALE CHE PERMETTE DI TROVARE SOLUZIONI INNOVATIVE E
CREATIVE AI PROBLEMI DI OGNI GIORNO , NON ESERCITANDOLO IN SOLUZIONI PREDEFINITE ,
MECCANICHE E RIPETITIVE MA ORIGINALI E PERSONALIZZATE CON UN PROCESSO DI
PENSIERO LOGICO E ALGORITMICO !!!*

*E' UNA ABILITA' CHE VA SVILUPPATA IL PRIMA POSSIBILE E IL MODO MIGLIORE E'
PROGRAMMARE*

- *ESSENDO DEFINITA UNA QUARTA ABILITÀ DI BASE DOVREBBE ESSERE INSEGNATA A SCUOLA PER OFFRIRE A TUTTI L'OPPORTUNITÀ' DI SVILUPPARLA.*
- *RIVALUTAZIONE DELL'ERRORE (DEBUGGING) : STIMOLARE GLI ALUNNI ALLA CORREZIONE DEGLI ERRORI LOGICO-PROCEDURALI E RIFORMULAZIONE DELLA PROCEDURA CORRETTA*
- *SI PUÒ FARE CODING CON IL COMPUTER O CON ATTIVITÀ UNPLUGGED*
- *IL PENSIERO COMPUTAZIONALE FAVORISCE IL LAVORO COOPERATIVO E LA MOTIVAZIONE AL COMPITO*



In un' ottica di longlife learning (educazione per tutta la vita) il pensiero computazionale diventa importante per offrire un' opportunità a tutti , partendo dai piu' giovani al fine di acquisire queste abilita' indipendentemente dalla professione che gli alunni andranno ad intraprendere.

Dal 2013 e' oggetto di campagna di alfabetizzazione e ogni anno viene scelto un tema diverso :

2013 : il tema era incentrato sulle competenze e lo sviluppo

2014 : parita' di genere , sensibilizzare le ragazze ad intraprendere studi e carriere nell'ambito delle discipline STEM

2015: creativita'

2016 : coding come realizzazione

Il Presidente degli Stati Uniti B.Obama nel 2013 e' diventato testimonial della Computer Science Education Week e si e' rivolto ai giovani dicendo : “ Non limitatevi a giocare con gli smarthphone ma programmateli””



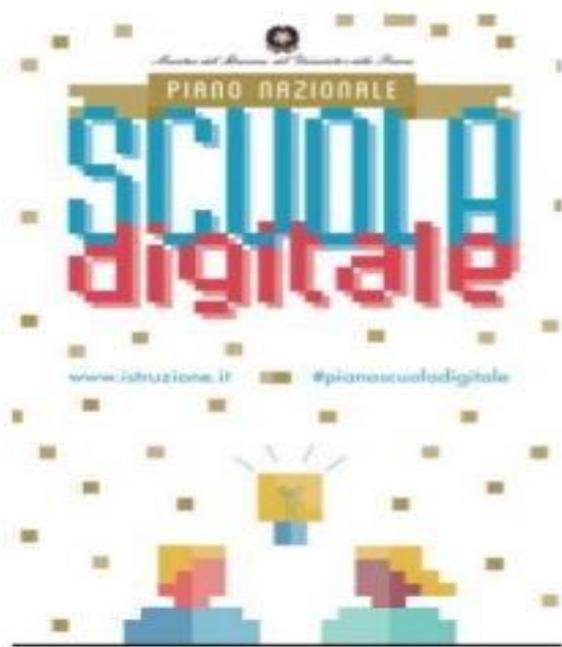
Nel Piano Nazionale Scuola Digitale, presentato il 27 ottobre 2015 dal ministro Giannini, riprende ed esplicita quanto già annunciato nella legge 107 Buona Scuola:

L'azione #17 del PNSD introduce di fatto il pensiero computazionale nella scuola primaria come esperienza di duplice valenza.

- avviare alla comprensione e alla conoscenza delle potenzialità della rete e della tecnologia fin da giovani
- vengono delineati gli indirizzi strategici per l'innovazione digitale facendo riferimento alle 8 competenze chiave tra cui la competenza digitale

Il futuro ... di Programma il Futuro

- Nel Piano Nazionale Scuola Digitale l'insegnamento del pensiero computazionale diventa parte dei programmi della Scuola Primaria
- L'azione 17 del PNSD cita espressamente Programma il Futuro come programma di riferimento per questa attività didattica
- Ogni studente della scuola elementare dovrà svolgere un corpus annuale di 10 ore



Programma il Futuro



STRUMENTI

ACCESSO

- Fibra e banda ultra-larga alla porta di ogni scuola
- Cablaggio interno di tutti gli spazi delle scuole (LAN/Wi-Lan)
- Canone di connettività: il diritto a Internet parte a scuola

SPAZI E AMBIENTI PER L'APPRENDIMENTO

- Ambienti digitali per la didattica
- Challenge Prize per la scuola digitale
- Linee guida per politiche attive di BYOD (Bring Your Own Device)
- Piano per l'apprendimento pratico
- Edilizia Scolastica Innovativa

IDENTITÀ DIGITALE

- Sistema di Autenticazione unica (Single-Sign-On)
- Un profilo digitale per ogni studente
- Un profilo digitale per ogni docente

AMMINISTRAZIONE DIGITALE

- Digitalizzazione amministrativa della scuola
- Registro elettronico
- Strategia "Dati della scuola"



COMPETENZE E CONTENUTI

COMPETENZE DEGLI STUDENTI

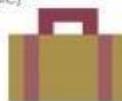
- Un framework comune per le competenze digitali degli studenti
- Scenari innovativi per lo sviluppo di competenze digitali applicate
- Una research unit per le Competenze del 21mo secolo
- Portare il pensiero computazionale a tutta la scuola primaria
- Aggiornare il curriculum di "Tecnologia" alla scuola secondaria di primo grado

DIGITALE, IMPRENDITORIALITÀ E LAVORO

- Un curriculum per l'imprenditorialità (digitale)
- Girls in Tech & Science
- Piano Carriere Digitali
- Alternanza Scuola-Lavoro per l'impresa digitale

CONTENUTI DIGITALI

- Standard minimi e interoperabilità degli ambienti on line per la didattica
- Promozione delle Risorse Educative Aperte (OER) e linee guida su autoproduzione dei contenuti didattici
- Biblioteche Scolastiche come ambienti di alfabetizzazione all'uso delle risorse informative digitali



FORMAZIONE E ACCOMPAGNAMENTO

FORMAZIONE DEL PERSONALE

- Formazione in servizio per l'innovazione didattica e organizzativa
- Rafforzare la formazione iniziale sull'innovazione didattica
- Assistenza tecnica per le scuole del primo ciclo
- La nuova formazione per i neoassunti

ACCOMPAGNAMENTO

- Un animatore digitale in ogni scuola
- Accordi territoriali
- Stakeholders' Club per la scuola digitale
- Un galleria per la raccolta di pratiche
- Dare alle reti innovative un ascolto permanente
- Osservatorio per la Scuola Digitale
- Un comitato Scientifico che allinei il Piano alle pratiche internazionali
- Il monitoraggio dell'intero Piano
- Un legame paese con il Piano Triennale per l'Offerta Formativa





Durata consigliata: 1 ora

Stimolare la fantasia e la creatività'

Obiettivo: dare consapevolezza delle potenzialita' del coding , riconoscere gli oggetti programmabili e stimolare la fantasia e la creativita'

Visione del video introduttivo (8 minuti)

Lavoro di gruppo (gioco , sfida, gara a squadre) per fare una lista di oggetti programmabili)

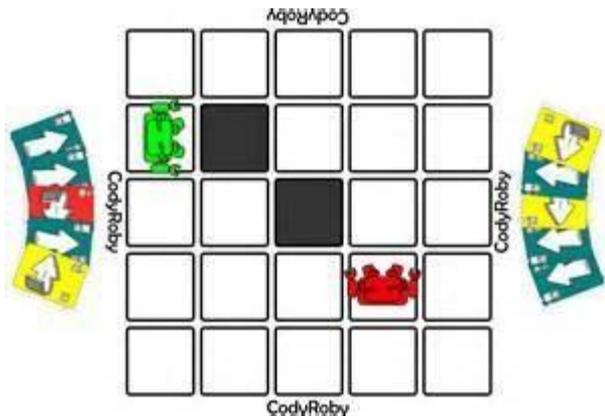
Riflettere su cosa si potrebbe fare con gli oggetti programmabili di diverso da cio' che gia' fanno

Riflettere su quali oggetti non programmabili potrebbero diventare programmabili

<https://prezi.com/q4mvtlre8rf/cose-che-parlano/>

ATTIVITA' UNPLUGGED

CODYROBY



CodyRoby è uno strumento di programmazione unplugged fai da te concepito per abbattere le barriere d'accesso al coding educativo. Due edizioni speciali di CodyRoby sono state dedicate ad [Africa Code Week](#) e a [Europe Code Week](#). Il materiale che compone ogni kit può essere scaricato liberamente dalla rete e stampato su 6 fogli A4. [CODY ROBY](#)

<http://bricks.maieutiche.economia.unitn.it/2014/06/14/coderkids-un-laboratorio-di-programmazione-unplugged-per-i-bambini-di-5-6-anni/>



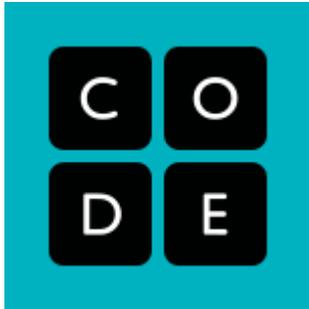
Strumento di lavoro unplugged che utilizza un linguaggio di programmazione a blocchi.

[codyway](https://www.codyway.com)

ATTIVITA' TECNOLOGICHE



L'ora del codice e' uno dei principali strumenti di alfabetizzazione orientati al coding . E' stata proposta da Hadi Partovi fondatrice di Code.org durante la Settimana dell'educazione informatica nel 2013. E' un gioco didattico a 20 livelli di complessita' che permettono ai bambini di acquisire gli schemi ricorrenti della programmazione visuale a blocchi . ([CODE.ORG](https://code.org))



Perche' si promuovono queste iniziative?

Perche' tra le alfabetizzazioni funzionali (leggere, scrivere, contare..) l'uso dell'informatica e l'uso consapevole di internet e' attualmente la competenza piu' carente ma anche la piu' richiesta.

In Italia, a partire da settembre 2014 quest'iniziativa dell'Ora del Codice è stata proposta a tutte le scuole nell'ambito del progetto **Programma il Futuro**, qui descritto <https://programmailfuturo.it/progetto/descrizione-del-progetto>

L'evento si svolge nell'ambito della **Settimana di Educazione all'Informatica**, quest'anno dal 6 al 12 dicembre 2021, e l'Italia è sempre stata il primo Paese al mondo per partecipazione all'Ora del Codice.

Se vuoi condividere l'Ora del Codice, consulta e diffondi questo materiale in italiano

<https://programmailfuturo.it/media/docs/hoc-one-pager-ITALIANO-2021.pdf>

<https://programmailfuturo.it/media/docs/hoc-participation-guide-ITALIANO-2021.pdf>

In Italia il consorzio CINI e il MIUR hanno promosso la versione italiana di code.org



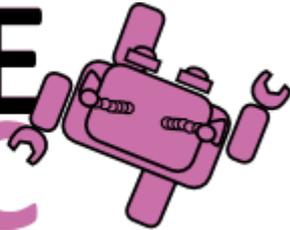
<http://www.programmailfuturo.it/>



SCRATCH E' UN LINGUAGGIO DI PROGRAMMAZIONE VISUALE CHE PERMETTE DI REALIZZARE STORIE INTERATTIVE, GIOCHI, ANIMAZIONE, MUSICA E TANTI ALTRI PROGRAMMI UTILI A SVILUPPARE LA LOGICA E LA CREATIVITA' DEGLI ALUNNI E AD IMPARARE IMPORTANTI CONCETTI DI CALCOLO MATEMATICO E TUTTO CIO' CHE PUO' ESSERE CONDIVISO CON ALTRI GRUPPI PARTECIPATIVI.

LA LOGICA CHE STA ALLA BASE DI QUESTO PROGRAMMA E' CHE GLI ALUNNI DA PASSIVI UTILIZZATORI DIVENTINO ATTIVI PROGRAMMATORI AL FINE DI SPERIMENTARE LE POTENZIALITA' E PADRONEGGIARE LE TECNOLOGIE E CONOSCERNE I PRINCIPI SCIENTIFICI E DI FUNZIONAMENTO.

#CODE MOOC



Formazione

Coding in your Classroom, Now! - Emma

Corso Mooc online erogato dall 'universita' di Urbino. Prof Alessandro Bogliolo

Informazioni corsi Mooc

<https://platform.europeanmoocs.eu/courses.php>

Sito di riferimento [Codemooc.org](https://codemooc.org)

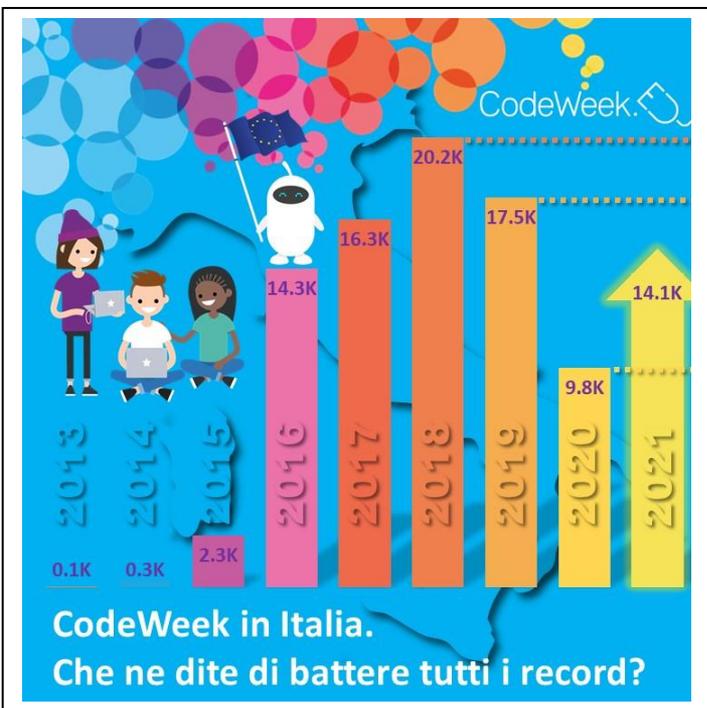
La settimana della programmazione

code week

Dal 9 al 24 Ottobre 2021 si svolge la settimana europea della programmazione, Europe Code Week, arrivata alla nona edizione. Nel corso della settimana si svolgeranno migliaia di eventi in ogni parte d'Europa per offrire a giovani e giovanissimi l'opportunità di iniziare a programmare! Non si tratterà di semplici seminari, ma di vere e proprie esperienze di programmazione, che consentiranno anche ai più piccoli di divertirsi in modo costruttivo vedendo le proprie idee prendere forma.

DIVERTIAMOCI A PROGRAMMARE !!!

<http://www.codeweek.it/eventi-online-aperti-2021/>



Materiale e siti utilizzati

- **Videotutorial** Elementi base del pensiero computazionale e SCRATCH di Virginia Ruggeri
- **Presentazione video** : Il pensiero computazionale Simonetta Anelli e Monica Boccoli
- **Articolo di Enrico Nardelli** <http://link-and-think.blogspot.it/2016/04/cinque-parole-chiave-pensiero-computazionale.html>

Cose che parlano!!Coding in your Classroom, Now! di Gabry Sergi

- **Video** Il linguaggio delle cose tratto dal corso MOOC Coding in your classroom Prof Bogliolo Alessandro Dipartimento di Informatica applicata Urbino
<https://www.youtube.com/watch?v=AEXF33EgH0w>
- **Articolo di Agnese Addone e Caterina Moschetti** *“CoderKids, un laboratorio di programmazione unplugged per i bambini di 5-6 anni*
- **Lezione** tratte dal corso : Coding in your classroom Prof Bogliolo Alessandro Università' di Urbino Facoltà' Informatica Applicata

Si ringraziano i colleghi che hanno condiviso e postato i loro lavori e le esperienze di coding applicate a scuola. Grazie

