



## La nuova certificazione “Computing” e un corso per i docenti sul pensiero computazionale

**Pierfranco Ravotto**

[pierfranco.ravotto@gmail.com](mailto:pierfranco.ravotto@gmail.com)

La Fondazione ECDL promuove e gestisce, da più di un ventennio, la certificazione *European Computer Driving Licence* o Patente del computer. Non una certificazione informatica ma una patente per gli utenti del computer. Se è vero che usando un computer - a differenza di un'auto - non si possono far danni fisici alle persone, una patente d'uso del computer può garantire che non si perdano dati, che non si perda tempo, che si usi lo strumento in modo efficiente ed efficace, che si protegga la propria privacy e che non si violi quella altrui. Questa la filosofia ECDL iniziale che è stata mantenuta pur modificando, nel corso degli anni, i syllabus dei diversi moduli ed i moduli stessi in coerenza con l'evoluzione della tecnologia digitale.

Ma un modulo sulla programmazione non esce da questo filone per andare verso le competenze propriamente informatiche? I programmatori sono professionisti informatici, non semplici utenti del computer.



Figura 1 - Il frontespizio del documento della Fondazione ECDL con cui viene presentato "Computing in education".

Nell'introduzione alla brochure di presentazione del nuovo modulo "Computing", la Fondazione scrive: *"Coding, computing e pensiero computazionale sono termini che attualmente dominano il dibattito sullo sviluppo delle competenze digitali in Europa e nel mondo. Ciò è influenzato dalla crescente domanda di specialisti ICT nel mercato del lavoro e nei paesi in cerca di modi per fornire le competenze richieste alle giovani generazioni. Inoltre l'informatica è considerata un argomento che contribuisce allo sviluppo intellettuale e personale dei giovani: stimola la creatività, il pensiero critico e logico, la risoluzione dei problemi e facilita la comprensione di come funzionano le tecnologie digitali. In altre parole, il computing trasforma gli studenti da consumatori a creatori del mondo digitale"*.

Dunque tre le motivazioni per questo nuovo modulo:

- c'è una crescente domanda di specialisti IT,
- l'informatica contribuisce allo sviluppo intellettuale e personale,
- il computing permette di passare da consumatori a creatori.

Si tratta di tre motivazioni assolutamente condivise da un ampio movimento che propone l'introduzione del *computational thinking* nella scuola. Negli USA è stato lo stesso presidente Obama, nel 2015, a farsi promotore di questo movimento con l'appello "Non comprare un nuovo videogioco, costruisce uno!"



Figura 2 - [Il video di Barack Obama a favore dell'insegnamento dell'informatica nella scuola.](#)

Dire, con Obama, "Costruisci il tuo videogioco", oppure "non scaricare una nuova APP, realizzala", significa stimolare a non essere utilizzatori passivi della tecnologia. Scrivere un'APP o un videogioco, per quanto semplici siano, permette di capire come funzionano i computer, sviluppa capacità progettuali, contribuisce alla formazione STEM (Science, Technology, Engineering, Mathematics) e, contemporaneamente, stimola la creatività.

Introdurre nella scuola, per tutti gli studenti, la programmazione - il *coding* - è anche il modo per favorire l'orientamento verso le professioni ICT e quindi verso percorsi di studio informatici. Un obiettivo importante, se è vero che da un lato esistono forti pericoli di disoccupazione e dall'altro c'è una carenza di persone specializzate nel campo dell'ICT, una carenza che è un freno per lo sviluppo del sistema Paese.

Ma come non si insegna a tutti gli studenti a leggere e scrivere perché tutti divengano critici letterari o scrittori, come non si insegna a tutti la biologia perché diventino tutti ricercatori nel campo del DNA, così l'informatica per tutti non serve a far diventare tutti degli informatici.

Si deve farlo - così ha proposto la Fondazione ECDL nel documento "Computing and Digital literacy - Call for a Holistic Approach" ([qui la versione in italiano](#) a cura di AICA) - per il valore "trasversale" dell'informatica quale disciplina che educa all'analisi e alla soluzione dei problemi (*problem solving*) e alla formalizzazione della soluzione in modo che possa essere eseguita.

Che la programmazione, ad un livello base, faccia parte della Competenza digitale lo afferma anche il framework DigComp, arrivato alla versione 2.1 ([qui in italiano](#)), che colloca nell'area di competenza 3, "Creazione di contenuti digitali", la competenza **3.4 Programmazione** così definita: "Pianificare e sviluppare una sequenza di istruzioni

comprensibili per un sistema di calcolo per risolvere un determinato problema o eseguire un'attività specifica”.

## **ECDL Computing, il Syllabus, l'esame, la certificazione**

Il Syllabus del modulo ECDL Computing - [disponibile qui](#) - indica quali obiettivi del modulo i sottostanti.

*Il candidato che ha superato il test è in grado di:*

- Comprendere i concetti fondamentali relativi all'informatica e alle attività tipiche necessarie alla creazione di un programma.
- Comprendere e usare tecniche di pensiero computazionale quali decomposizione del problema, riconoscimento di modelli, astrazione e algoritmi per analizzare un problema e sviluppare soluzioni.
- Scrivere, verificare e modificare algoritmi per un programma usando schemi di flusso e pseudocodice.
- Comprendere i principi e i termini fondamentali associati alla programmazione e l'importanza di codice ben strutturato e documentato.
- Comprendere e usare costrutti di programmazione all'interno di un programma, quali variabili, tipi di dati e logica.
- Migliorare l'efficienza e la funzionalità utilizzando iterazioni, istruzioni condizionali, procedure e funzioni all'interno di un programma, insieme ad eventi e comandi.
- Sottoporre un programma a test e debug, assicurandosi che risponda ai requisiti richiesti prima del rilascio.

Questa la struttura del syllabus:

Sezione	Tema
<b>1. Termini informatici</b>	<b>1.1 Concetti fondamentali</b>
<b>2. Metodi di pensiero computazionale</b>	<b>2.1 Analisi del problema 2.2 Algoritmi</b>
<b>3. Iniziare a programmare</b>	<b>3.1 Per iniziare 3.2 Variabili e tipi di dati</b>
<b>4. Costruire con il codice</b>	<b>4.1 Logica 4.2 Iterazione 4.3 Condizionalità 4.4 Procedure e funzioni 4.5 Eventi e comandi</b>
<b>5. Test, debug e rilascio</b>	<b>5.1 Esecuzione, test e debug 5.2 Rilascio</b>

Le conoscenze e le abilità sono valutate attraverso il superamento di una prova d'esame di tipo automatico attraverso il sistema ATLAS, presso i Test Center accreditati, e prevede domande a risposta multipla che implicano, tra l'altro, la comprensione di esempi di codice scritto in linguaggio Python.

Il superamento dell'esame dà diritto di ottenere il Certificato ECDL Computing.

## Un corso di Fare didattica nel web sul computational thinking

"Fare didattica nel web" è un insieme di corsi che AICA propone - sulla piattaforma [www.faredidattica.it](http://www.faredidattica.it) - alle scuole e ai singoli docenti sulla Didattica digitale o, per essere più precisi, su come l'uso di dispositivi digitali e ambienti web può aiutare ad insegnare e apprendere.

La proposta si è ora arricchita di un nuovo corso: "Fare didattica con il coding" rivolto a tre target di persone:

- insegnanti che vogliono iniziare a proporre attività di *computational thinking* ai propri studenti,
- insegnanti che già propongono attività in tal senso ma sentono la necessità di acquisire i principi base della programmazione o che vogliono passare da un linguaggio grafico come Scratch ad un linguaggio testuale quale Python,
- insegnanti e personale di Test Center intenzionati ad organizzare corsi finalizzati a far acquisire la certificazione ECDL Computing.

Il corso, come gli altri, è offerto ai singoli in modalità completamente online e potrà, a breve, essere acquistato su [AicaShop](http://AicaShop) oppure con il bonus docente tramite la piattaforma SOFIA (Codice identificativo 20804). Chiunque può iniziarlo quando preferisce, seguirlo e completarlo negli orari e con i ritmi che preferisce. E' garantita una intensa e costante attività di tutoraggio. I corsisti hanno a disposizione un percorso formativo strutturato costituito da:

- risorse didattiche - prevalentemente video-lezioni o video-tutorial - in parte appositamente realizzate, in parte selezionate nel web,
- proposte di lavoro,
- stimoli per la discussione e il confronto con i tutor e con gli altri corsisti.

Il corso è organizzato nelle seguenti aree:

### 1. Perché il coding

Perché introdurre attività di programmazione nella scuola. Video, testi, documenti, articoli, il framework europeo DigComp, esempi e casi d'uso.

### 2. La metodologia

Analisi del problema e progettazione della soluzione, descrizione dell'algoritmo risolutivo come pseudocodice o come diagramma di flusso, implementazione in un linguaggio di programmazione, test e miglioramento. Tools per disegnare

diagrammi di flusso. La programmazione strutturata: sequenza, alternativa/condizione, ciclo/iterazione.

### **3. Gli strumenti**

I linguaggi di programmazione. Scratch e Python. Con cenni alla programmazione unplugged, a Code.org e a Robotica e Arduino.

### **4. ECDL Computing**

Il modulo, il syllabus, test di simulazione dell'esame.

Il corso è finalizzato a formare i docenti all'uso del coding/programmazione/computational thinking nella scuola. Ed è adatto per chi voglia farlo a tutti i livelli: dalla scuola dell'infanzia alla superiore; cambieranno da un contesto all'altro le attività da proporre agli studenti, in particolare cambierà la loro complessità.

Come ambienti/linguaggi di programmazione vengono proposti sia Scratch, che è un linguaggio grafico, sia Python che è un linguaggio testuale; molte soluzioni sono proposte con entrambi i linguaggi sia per permetterne un confronto, sia, soprattutto, per attirare l'attenzione sulla progettazione dell'algoritmo risolutivo - che può essere implementato con linguaggi diversi - piuttosto che sul linguaggio.

Il percorso contiene più di quello che serve per l'esame ECDL Computing ma sicuramente fornisce tutte le conoscenze e le abilità necessarie a superarlo.