

## Premessa

Il coding a scuola non deve essere considerato una disciplina di insegnamento o una materia a sé stante, ma è un metodo didattico adatto a discipline sia scientifiche che umanistiche.

È basato principalmente sul problem solving e ha l'obiettivo di stimolare lo sviluppo del cosiddetto pensiero computazionale.

Definire le competenze di cui i nostri studenti hanno bisogno è una sfida ben più ampia e strutturata di quella che il sentire comune sintetizza nell'uso critico della Rete, o nell'informatica. Dobbiamo affrontarla partendo da un'idea di competenze allineata al ventunesimo secolo: fatta di nuove alfabetizzazioni, ma anche e soprattutto di competenze trasversali e di attitudini da sviluppare. In particolare, occorre rafforzare le competenze relative alla comprensione e alla produzione di contenuti complessi e articolati anche all'interno dell'universo comunicativo digitale, nel quale a volte prevalgono granularità e frammentazione. Proprio per questo è essenziale lavorare sull'alfabetizzazione informativa e digitale (information literacy e digital literacy), che mettono al centro il ruolo dell'informazione e dei dati nello sviluppo di una società interconnessa basata sulle conoscenze e l'informazione. È in questo contesto che occorre guardare alle sfide rappresentate dal rapporto fra pubblico e privato, dal rapporto tra creatività digitale e artigianato, e tra imprenditorialità digitale, manifattura e lavoro. Ed è ancora in questo contesto che va collocata l'introduzione al pensiero logico e computazionale e la familiarizzazione con gli aspetti operativi delle tecnologie informatiche. In questo paradigma, gli studenti devono essere utenti consapevoli di ambienti e strumenti digitali, ma anche produttori, creatori, progettisti.

Tratto dal PNSD, Competenze degli studenti, p. 29

Questo percorso di certificazione è rivolto alla Scuola Secondaria di II grado e certifica le competenze dell'utente nell'utilizzare linguaggi di programmazione a scopo didattico, quali Twine, Kojo, HTML, CSS e Javascript.

La certificazione si struttura in 4 moduli, dei quali i primi 3 sono relativi all'acquisizione dei principi di programmazione informatica, il quarto contiene esempi di applicazioni di coding in diverse discipline. Data la natura di questo ultimo modulo, abbiamo previsto che l'esame di certificazione si svolga solo sui primi 3 moduli, ma non sull'ultimo.

*Centro Studi CERTIPASS*

# Programma analitico d'esame

## EIPASS CODING

### Metodo

Superando il vecchio schema “argomento”, “ambito di intervento” e “testing di competenza”, proponiamo un nuovo modo di elencare e descrivere i contenuti dei moduli previsti, basato su quello utilizzato nell'*e-Competence Framework for ICT Users – Part 2: User Guidelines*.

È un sistema intellegibile e immediato per chi deve affrontare il percorso di certificazione e, soprattutto, per chi deve valutare la congruenza delle competenze possedute dall'Utente certificato. Per ognuno degli argomenti previsti, quindi, troverete un quadro di riferimento che indica:

- la definizione sintetica della competenza di cui si tratta;
- tutto ciò che l'Utente certificato conosce di quell'argomento (*conoscenza teorica/knowledge*);
- tutto ciò che l'Utente certificato sa fare concretamente, in relazione alle conoscenze teoriche possedute (*conoscenze pratiche/Skills*);

### Procedure e strumenti

Per prepararsi alla prova d'esame, il candidato usufruisce dei servizi e del supporto formativo online disponibile nell'[Aula Didattica 3.0](#).

Per superare la prova d'esame, è necessario rispondere correttamente ad almeno il 75% delle 30 domande previste per ogni modulo. Si precisa, infine, che ciascun modulo rappresenta uno specifico ambito di competenze e che, quindi, al di là delle interconnessioni esistenti tra i vari settori, il candidato può stabilire autonomamente l'ordine con cui affrontarli.

### Moduli d'esame

**Modulo 1** | Il coding come metodo didattico

**Modulo 2** | La programmazione informatica

**Modulo 3** | Linguaggi di programmazione

**Modulo 4** | Progetti di esempio



## *Modulo 1*

# IL CODING COME METODO DIDATTICO

## **Cosa sa fare il Candidato che si certifica con EIPASS Coding**

Il candidato certificato sa descrivere come il coding diventi un metodo didattico all'interno delle discipline curriculari. Conosce la definizione di pensiero computazionale.

Sa definire il modello pedagogico che sottende l'applicazione del coding nella didattica, ossia il costruzionismo.

Sa spiegare come il coding sia un metodo didattico ideale nella realizzazione di didattiche per competenze.

## **Contenuti del modulo**

### **Definizione**

- Il metodo didattico
- Pensare come un informatico
- Linguaggi di programmazione



## 1 | DEFINIZIONE

Riconoscere e descrivere il coding come metodo didattico. Definire come il coding sia utile per sviluppare il pensiero computazionale e ai fini della didattica per competenze.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
1.1	Il metodo didattico	1.1.1	Definire il concetto di pensiero computazione, l'origine dell'espressione e il suo significato
1.2	Pensare come un informatico	1.2.1	Definire il costruzionismo e descrivere come Seymour Papert ha contribuito alla sua definizione
		1.2.2	Riconoscere il linguaggio di programmazione LOGO e descriverlo brevemente
1.3	Linguaggi di programmazione	1.3.1	Comprendere come il problem solving attraverso il Coding sia utile nella didattica per competenze



## Modulo 2

# LA PROGRAMMAZIONE INFORMATICA

## Cosa sa fare il Candidato che si certifica con EIPASS Coding

Il Candidato certificato possiede le competenze di base e le tecniche della programmazione informatica classica.

Sa definire cosa rappresenta un programma e cosa sono le istruzioni.

Sa distinguere le diverse strutture di controllo.

Conosce l'utilizzo delle variabili e delle strutture di dati.

È in grado di definire procedure e funzioni. Conosce gli algoritmi e la loro rappresentazione grafica come diagramma di flusso.

## Contenuti del modulo

### Definizione

- Programma
- Strutture di controllo

### Variabili e strutture di dati

- Istruzioni e dati

### Procedure e funzioni

- Procedure
- Funzioni

### Algoritmi

- Definizione
- Formalismo grafico



## 1 | DEFINIZIONE

Comprendere i principi tecnici e sociali di Internet. Riconoscere e utilizzare gli elementi principali di una pagina web. Sapere cosa sia possibile fare in rete. In relazione alla sicurezza, conoscere il significato dei protocolli e definire il concetto di crittografia, applicato all'informatica.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
1.1	Programma	1.1.1	Definire il concetto di istruzioni e la modalità in cui vanno fornite
1.2	Strutture di controllo	1.2.1	Cos'è e a cosa serve la sequenza
		1.2.2	Riconoscere una selezione e da cosa è caratterizzata
		1.2.3	Definire l'iterazione e il suo significato

## 2 | VARIABILI E STRUTTURE DI DATI

Comprendere e definire come lavorano le istruzioni sui dati e come avviene la loro elaborazione.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
2.1	Istruzioni e dati	2.1.1	Definire le variabili e il loro utilizzo
		2.1.2	Definire le liste o matrici e la loro funzione
2.3	Configurazione	2.3.1	Impostare la pagina iniziale del browser
		2.3.2	Riconoscere, definire e gestire i pop-up
		2.3.3	Riconoscere, definire e gestire i cookie

## 3 | PROCEDURE E FUNZIONI

Definire procedure e funzioni, sapendone distinguere le differenze. Conoscere il valore dei parametri e delle parole chiave per realizzare procedure e funzioni.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
3.1	Procedure	3.1.1	Definire il parametro in relazione alla creazione di procedure
3.2	Funzioni	3.2.1	Definire la parola chiave in relazione alla creazione di funzioni



## 4 | ALGORITMI

Descrivere il funzionamento degli algoritmi e il formalismo grafico per rappresentarli. Rappresentare procedure con il flow-chart.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
4.1	Definizione	4.1.1	Descrivere l'etimologia del termine
4.2	Le applicazioni tramite cui gestire le email	4.2.1	Definire un flow-chart e i diversi blocchi che lo compongono
		4.2.2	Definire il flow-chart delle strutture di controllo base
		4.2.3	Rappresentare procedure complesse con il formalismo del flow-chart



## Modulo 3

# LINGUAGGI DI PROGRAMMAZIONE

## Cosa sa fare il Candidato che si certifica con EIPASS Coding

Il Candidato certificato conosce i linguaggi di programmazione Twine, Kojo, HTML, CSS e Javascript.

Sa realizzare una storia interattiva a bivi con Twine, utilizzando anche le variabili.

Conosce il linguaggio Kojo e lo utilizza per applicazioni geometrico-matematiche.

Sa definire la logica rappresentativa e il funzionamento dei linguaggi del web.

Sa realizzare più pagine web in linguaggio HTML, collegate tra loro.

Applica le regole per definire le proprietà stilistiche con CSS alle pagine web create

Utilizza Javascript a livello basico per rendere interattiva e dinamica una pagina web.

## Contenuti del modulo

### Twine

- Utilizzo di Twine

### Kojo

- Definizione e utilizzo
- Disegnare un quadrato
- L'algoritmo
- Disegnare un poligono
- Salvare e aprire i progetti

### Linguaggi del Web: HTML, CSS, Javascript

- Linguaggi
- HTML
- CSS
- Twine, HTML e CSS

### Javascript

- Definizione
- Scrivere in Javascript



## 1 | TWINE

Realizzare una storia interattiva non lineare con Twine. Realizzare testo e nodi della storia, inserire le variabili, esportare, importare e visualizzare la storia.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
1.1	Utilizzi di Twine	1.1.1	Creare una storia
		1.1.2	Definire il testo e i nodi
		1.1.3	Visualizzare l'anteprima
		1.1.4	Esportare la storia
		1.1.5	Importare una storia
		1.1.6	Utilizzare i comandi per condizionare il contenuto con Harlowe

## 2 | KOJO

Utilizzare il linguaggio di programmazione Kojo. Disegnare un quadrato e un poligono realizzando un programma che fornisce le istruzioni adeguate al compito. Utilizzare i parametri per le procedure.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
2.1	Definizione e utilizzo	2.1.1	Installare Kojo
		2.1.2	Descrivere e riconoscere l'interfaccia del programma Kojo
2.2	Disegnare un quadrato	2.2.1	Rappresentare le istruzioni con l'algoritmo
		2.2.2	Utilizzare le istruzioni in linguaggio Kojo
		2.2.3	Avviare l'esecuzione del programma
		2.2.4	Creare la definizione del programma "quadrato"
		2.2.5	Inserire parametri nella procedura
2.3	Disegnare un poligono	2.3.1	Definire il codice per disegnare un poligono
		2.3.2	Impostare la velocità della tartaruga che traccia il poligono
2.4	Salvare e aprire progetti	2.4.1	Come si salvano e si aprono i progetti



### 3 | LINGUAGGI DEL WEB: HTML, CSS, JAVASCRIPT

Conoscere i principali linguaggi del web. Realizzare pagine web sia nei contenuti che nello stile. Rendere le pagine web interattive.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
3.1	Linguaggi	3.1.1	Definire i principali linguaggi del web
3.2	HTML	3.2.1	Definire e utilizzare i marcatori (tag)
		3.2.2	Creare un contenuto di esempio
		3.2.3	Come scrivere in codice HTML
		3.2.4	Utilizzare tag HTML più avanzati
3.3	CSS	3.3.1	Le regole per definire le proprietà stilistiche delle pagine web
		3.3.2	Definire e utilizzare le regole per il parametro "class"
		3.3.3	Definire e utilizzare le regole per gli attributi identificativi
		3.3.4	Utilizzare il tagging HTML
3.4	CSS	3.4.1	Applicare i linguaggi HTML e CSS per una storia realizzata con Twine

### 4 | JAVASCRIPT

Utilizzare Javascript a livello basico per rendere interattive le pagine web realizzate. Creare un esempio di pagina interattiva come progetto didattico grammaticale.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
3.1	Definizione	3.1.1	Riconoscere la sintassi e il cosiddetto HTML DOM
3.2	Scrivere in Javascript	3.2.1	Inserire il codice dentro la pagina HTML
		3.2.2	Realizzare un esempio di tag Javascript
		3.2.3	Visualizzare il pannello di controllo
		3.2.4	Scrivere il codice dell'algoritmo nell'esempio



## Modulo 4

# PROGETTI DI ESEMPIO

## Cosa sa fare il Candidato che si certifica con EIPASS Coding

In questo modulo sono presentati diversi esempi di applicazioni didattiche di coding. Il modulo, per tale ragione, non è oggetto di esame di certificazione.

In questa parte il candidato apprende come realizzare una storia fantasy con Twine. Sa realizzare il grafico di una funzione con Kojo. Sempre con Kojo sa realizzare un progetto musicale.

Realizza, con i linguaggi del web, un generatore di Haiku.

## Contenuti del modulo

### Twine

- Trama della storia interattiva non lineare

### Kojo

- Grafico di una funzione
- Musica elettronica

### Linguaggi del web

- Un generatore di Haiku



## 1 | TWINE

Realizzare un gioco di avventura complesso, con diversi nodi e variabili.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
1.1	Trama della storia interattiva non lineare	1.1.1	Utilizzare i tag HTML nei nodi della storia
		1.1.2	Applicare lo stile di visualizzazione con CSS
		1.1.3	Gestire i comportamenti interattivi

## 2 | KOJO

Conoscere l'utilizzo del linguaggio Kojo per realizzare il grafico di una funzione. Realizzare un progetto di musica elettronica.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
2.1	Grafico di una funzione	2.1.1	Realizzare un esempio di grafico
		2.1.2	Realizzare un'applicazione per tracciare un grafico
2.2	Musica elettronica	2.2.1	Realizzare un esempio di coding per l'esecuzione di un brano musicale

## 3 | LINGUAGGI DEL WEB

Realizzare un'attività didattica creando un generatore automatico di Haiku, delle particolari poesie giapponesi.

Knowledge/Conoscenze		Skills/Capacità pratiche	
L'utente certificato conosce...		L'utente certificato sa...	
3.1	Un generatore di Haiku	3.1.1	Realizzare un esempio di generatore di Haiku

