CZĘŚĆ 4: Przygotowanie materiałów i zasobów

Struktura modułu

1 Cel i zadania
2 Efekty uczenia się
3 Zastosowane zasady myślenia komputacyjnego
4 Powiązanie z modułami WP2
5 Wyzwania
6 Uzasadnienie
7 Cwiczenia
8 Dodatkowe zasoby
9 Jaka jest wartość tego modułu?

1. Cel i zadania

Celem tego etapu jest pomoc w zrozumieniu, dlaczego staranne przygotowanie i dobór zasobów oraz materiałów ma tak duże znaczenie podczas wprowadzania myślenia komputacyjnego do edukacji dorosłych. To samo ćwiczenie może prowadzić do bardzo różnych doświadczeń edukacyjnych w zależności od sposobu jego przedstawienia.

CELE

Po ukończeniu tego etapu edukatorzy dorosłych będą potrafili:

- Analizować, w jaki sposób różne rodzaje materiałów dydaktycznych (cyfrowe i niecyfrowe) wspierają określone umiejętności myślenia komputacyjnego w kontekście edukacji dorosłych;
- Identyfikować jasne i praktyczne kryteria wyboru dostępnych oraz wartościowych materiałów dla dorosłych słuchaczy;
- Przygotowywać zadania edukacyjne, które rozwijają myślenie komputacyjne z wykorzystaniem niecyfrowych i codziennych zasobów bliskich dorosłym uczestnikom;
- Przygotowywać i dostosowywać zasoby, które promują aktywne uczestnictwo, refleksję oraz transfer umiejętności myślenia komputacyjnego do życia codziennego.

Dofinansowane przez Unię Europejską

Projekt został współfinansowany przy wsparciu programu ERASMUS+ Komisji Europejskiej. Niniejsza publikacja odzwierciedla wyłącznie stanowisko jej autorów i Komisja Europejska oraz SEPIE nie ponoszą odpowiedzialności za umieszczoną w niej zawartość merytoryczną ani za jakiekolwiek wykorzystanie tych informacji.



2. Efekty uczenia się

Po ukończeniu tego rozdziału trenerzy dorosłych będą potrafili:

- Rozumieć, dlaczego przygotowanie i dobór zasobów oraz materiałów bezpośrednio wpływają na rozwój umiejętności myślenia komputacyjnego u dorosłych słuchaczy;
- Rozumieć, w jaki sposób dobrze zaprojektowane materiały mogą uczynić myślenie komputacyjne dostępnym, wartościowym i angażującym dla dorosłych uczestników;
- Projektować zadania bez użycia komputera oraz ćwiczenia oparte na technologii, które czynią myślenie komputacyjne przystępnym dla dorosłych słuchaczy i seniorów;
- Prowadzić procesy uczenia się, które zachęcają do uczestnictwa, refleksji oraz stosowania myślenia komputacyjnego w rozwiązywaniu codziennych problemów;
- Reflektować nad swoją rolą jako trenerów w zakresie organizowania, dostosowywania i oceniania materiałów dydaktycznych zgodnie z potrzebami i kontekstem uczestników.



3. Zastosowane główne zasady myślenia komputacyjnego

W tej części zastosowano wszystkie cztery zasady myślenia komputacyjnego

- Dekompozycję
- Rozpoznanie wzorców
- Abstrahowanie
- Myślenie algorytmiczne



4. Powiązanie z modułami WP2

Ten rozdział jest ściśle powiązany z Modułem 3 (M3) „Projektowanie planów lekcji i zadań", w którym przedstawiono strukturę oraz kluczowe elementy planowania lekcji opartych na myśleniu komputacyjnym. Niniejszy rozdział rozwija te treści, prowadząc trenerów przez proces projektowania i kompletowania zasobów edukacyjnych. Łączy się on również z Modułem 4 (M4) „Dostosowywanie strategii myślenia komputacyjnego do różnych poziomów nauczania", wspierając ich adaptację do potrzeb i możliwości słuchaczy.



5. Wyzwania

Gdy trenerzy dorosłych przygotowują materiały i zasoby do kursu z zakresu myślenia komputacyjnego, mogą szybko zdać sobie sprawę, że zadanie to wymaga znacznie więcej niż tylko wyboru odpowiednich narzędzi czy ćwiczeń.

Kolejne wyzwanie wiąże się ze zrozumieniem, że zasoby przeznaczone do nauki myślenia komputacyjnego mają rzeczywiście wspierać myślenie, a nie tylko samo działanie. Dobrze znane, praktyczne ćwiczenia można stopniowo dostosowywać tak, aby rozwijały pożądane umiejętności intelektualne.
Ograniczony czas na przygotowania może być przytłaczający. Tworzenie jasnych, dostępnych i wartościowych materiałów często wymaga starannego planowania. Kluczem do sukcesu jest projektowanie niewielkich zadań wielokrotnego użytku oraz współpraca z innymi nauczycielami – nawet proste szablony potrafią zaoszczędzić czas, zachowując jednocześnie wysoką efektywność nauki.



6. Uzasadnienie

ażdy proces nauczania i uczenia się opiera się na materiałach, które aktywnie wspierają interakcję między trenerami a słuchaczami. W przypadku myślenia komputacyjnego materiały odgrywają szczególnie ważną rolę, ponieważ pomagają uczynić abstrakcyjne procesy myślowe widocznymi i namacalnymi.

Dobrze znane, codzienne materiały zmniejszają obciążenie poznawcze i pozwalają słuchaczom skupić się na rozwijaniu umiejętności myślenia komputacyjnego, zamiast na opanowywaniu skomplikowanych technologii.

Badania pokazują, że słuchacze zapamiętują znacznie więcej informacji, gdy pojęcia są poparte dobrze zaprojektowanymi materiałami, a nie tylko samym wyjaśnieniem słownym.

7.1 Ćwiczenia

LPrzyjrzyjmy się rodzajom działań, które można tworzyć w ramach stosowania myślenia komputacyjnego w klasie.

Ćwiczenia unplugged

Zadania unplugged nie wymagają urządzeń cyfrowych i mogą pomóc w przezwyciężeniu barier, takich jak brak znajomości języków programowania czy ograniczony dostęp do zasobów cyfrowych. Istnieją dowody na to, że tego typu ćwiczenia wspierają naukę programowania oraz ułatwiają wejście w świat robotyki.

Majsterkowanie edukacyjne

Majsterkowanie edukacyjne polega na analizowaniu elementów danego obiektu – takiego jak klocki, puzzle, symulatory czy kod programu – a następnie na ich zmienianiu lub modyfikowaniu. Pokazuje to, jak niewielka zmiana może wpłynąć na algorytm lub rozwiązanie danego problemu.

Tworzenie

Zadania te koncentrują się na umożliwieniu słuchaczom rozwiązywania problemów, planowania własnej pracy, wyboru odpowiednich narzędzi, komunikowania pomysłów oraz łączenia różnych pojęć.

Zadania remiksujące

W zadaniach polegających na remiksowaniu udostępnia się i modyfikuje istniejący kod lub algorytmy, aby dostosować je lub połączyć z innymi elementami w celu wykonania zadania, rozwiązania problemu bądź sprostania wyzwaniu.

7.1 Ćwiczenia Unplugged

Kto powiedział, że do nauki myślenia komputacyjnego potrzebne są komputery? Czasami najbardziej efektywna nauka odbywa się z dala od ekranu. Zadania bez użycia komputera pomagają słuchaczom skupić się na samych procesach myślowych, a nie na technologii, dzięki czemu nauka staje się bardziej dostępna i angażująca.

to kilka praktycznych przykładów:

- Rozłożenie sekwencji czynności na wszystkie pośrednie etapy
- Opisanie rysunku partnerowi, aby mógł go odtworzyć
- Napisanie zestawu instrukcji (kodu), aby pomóc innej osobie w wykonaniu konstrukcji, ruchu lub podobnego zadania
- Projektowanie drzew decyzyjnych
- Praca z algorytmami
- Granie w gry polegające na kodowaniu lub dekodowaniu kodu binarnego

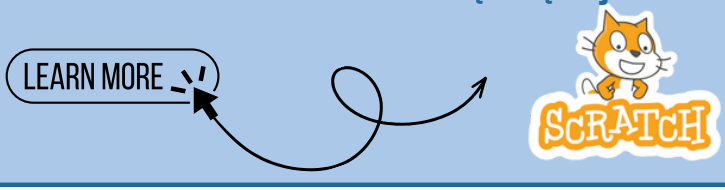


https://www.csunplugged.org

7.2 Działania edukacyjne oparte na technologii

Jednym z najbardziej znanych narzędzi do programowania wizualnego jest Scratch. Nie daj się zwieść jego zabawnemu wyglądowi. Scratch to potężne środowisko edukacyjne. Ułatwia przejście do bardziej zaawansowanych języków programowania, jednocześnie umożliwiając uczniom eksperymentowanie, tworzenie i naukę poprzez próbę i błąd. Dzięki Scratchowi można projektować quizy do pracy nad treścią, tworzyć proste aplikacje lub gry oraz uwidaczniać abstrakcyjne pojęcia, czyniąc je wizualnymi i intuicyjnymi.
Warto również zapoznać się z platformami związanymi z robotyką, takimi jak Arduino i Micro:bit, a także innymi związanymi z robotami i urządzeniami, takimi jak mBlock lub LEGO.
Narzędzia te oferują praktyczne możliwości połączenia myślenia komputerowego z działaniami w świecie rzeczywistym, dzięki czemu nauka staje się konkretna i bardzo motywująca.

Chcesz dowiedzieć się więcej o Scratch?



Scratch jest używany na całym świecie nie tylko w szkołach, ale także na uniwersytetach, w ośrodkach kultury i programach kształcenia dorosłych. Jego siła polega na przekształcaniu złożonych idei w coś, co uczniowie mogą zobaczyć, manipulować i zrozumieć.

7.3.1.Ćwiczenie: Wskazówki dotyczące projektowania treści

Tworząc zadania edukacyjne, trenerzy dorosłych powinni zadać sobie pytania: Kto będzie korzystał z tych treści? Jak łatwo można je dostosować? I jakie doświadczenie edukacyjne one stworzą? Pamiętanie o tych kwestiach pomaga upewnić się, że Twoje materiały rzeczywiście wspierają wartościową naukę.
Aby to osiągnąć, projektowane przez Ciebie treści powinny być:

- Dostępne: Powinny być zgodne z zalecanymi wytycznymi dotyczącymi dostępności, aby mogli z nich korzystać słuchacze o różnych możliwościach. W przypadku dorosłych odbiorców oznacza to jasne instrukcje, dobrze znane przedmioty oraz przejrzystość wizualną.
- Łatwe do edycji: Zadania należy tworzyć za pomocą bezpłatnych, prostych i wieloplatformowych narzędzi. Gdy treść jest łatwa do modyfikacji, staje się prostsza do dostosowania do różnych grup uczestników, kontekstów i potrzeb.
- Innowacyjne: Materiały dydaktyczne powinny wykraczać poza samo przekazywanie informacji i działać jako narzędzia stymulujące zmiany, kreatywność oraz innowacyjność.

Z tego powodu dobrze zaprojektowane zadanie edukacyjne to takie, które:

- Czyni sytuację edukacyjną namacalną, realistyczną i wykonalną;
- Jasno określa rolę zarówno słuchaczy, jak i trenerów, dzięki czemu każdy wie, czego się od niego oczekuje;
- Jasno wyjaśnia, w jaki sposób technologia cyfrowa jest wykorzystywana w procesie nauczania i uczenia się;
- Promuje aktywne uczenie się, w którym słuchacze rozwijają kompetencje poprzez uczestnictwo i refleksję;
- Może być powiązane z konkretnym obszarem tematycznym lub łączyć kilka obszarów jednocześnie, wspierając edukację międzyprzedmiotową.



7.3.2 Ćwiczenie. Wskazówki dotyczące projektowania treści

Jedną z najbardziej uznanych struktur jest model Brennana–Resnicka. Czy zastanawiałeś się kiedyś, jak edukatorzy porządkują wszystkie elementy myślenia komputacyjnego (CT)? Model ten jest potężnym narzędziem, które Cię poprowadzi. Jest on zorganizowany wokół trzech wymiarów: pojęć, praktyk i perspektyw. Pomaga to spojrzeć na zagadnienie z szerszej perspektywy, a jednocześnie podzielić je na mniejsze, łatwiejsze do opanowania części.

Mając do dyspozycji ten model, możesz opracować jasne kryteria podejmowania kluczowych decyzji:

- Czego musisz się nauczyć jako trener?
- Czego powinieneś nauczyć swoich uczestników?
- Jak możesz zintegrować myślenie komputacyjne (CT) z działaniami w swojej dziedzinie?
- Jak możesz skutecznie oceniać efekty uczenia się?
- Jakie treści powinieneś wybrać, potraktować priorytetowo lub pominąć?
- Jak możesz odróżnić wysokiej jakości zasoby od tych mniej skutecznych?

Aby zapoznać się z tym tematem bardziej szczegółowo i pogłębić swoją wiedzę, możesz zajrzeć do:
Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. Proceedings of the 2012 Annual Meeting of the American Educational Research Association. Vancouver: American Educational Research Association.



Click to download → Brennan, K., & Resnick, M. (2012)

7.4 Ćwiczenie. Gra w sortowanie z wykorzystaniem przedmiotów codziennego użytku

Zapewnij uczestnikom zestaw przedmiotów codziennego użytku (klucze, guziki, monety, długopisy itp.).

Poproś ich o pracę w małych grupach i posortowanie przedmiotów według różnych kryteriów, takich jak rozmiar, kolor, materiał lub funkcja



1. Niech każda grupa samodzielnie ustali własną zasadę sortowania i wyjaśni, dlaczego ją wybrała.
2. Poproś uczestników, aby zmienili tę zasadę i ponownie posortowali te same przedmioty, używając innego kryterium.
3. Zaproś osoby uczące się do opisania swojego procesu sortowania w postaci prostego zestawu instrukcji (czyli algorytmu).

To ćwiczenie rozwija umiejętności abstrakcji, klasyfikacji oraz myślenia algorytmicznego, pomagając uczestnikom zrozumieć, w jaki sposób komputery porządkują i przetwarzają informacje. Kiedy osoby uczące się sortują i ponownie klasyfikują przedmioty według różnych zasad, ćwiczą te same procesy logiczne, z których na co dzień korzystają analitycy danych i programiści!

7.5 Ćwiczenie. Tworzenie wzorów z wykorzystaniem figur geometrycznych

Zapewnij uczestnikom różnorodne figury geometryczne (koła, kwadraty, trójkąty itp.) i poproś ich o utworzenie wzorów z tych figur.

Powiedz uczestnikom, że muszą stworzyć wzór poprzez naprzemienne układanie kół i kwadratów lub poprzez stopniowe zwiększanie rozmiaru trójkątów. Postaraj się połączyć figury z różnymi kolorami, aby mogli oni pracować również nad kojarzeniem barw, co zawsze przynosi świetne rezultaty.

Następnie ich zadaniem jest przedstawienie tych wzorów w postaci prostych algorytmów, takich jak „koło, kwadrat, koło, kwadrat" lub „za każdym razem dodaj jeden trójkąt". To ćwiczenie rozwija umiejętności rozpoznawania wzorców oraz myślenia algorytmicznego, pomagając uczestnikom nauczyć się identyfikować na przykład błędy w kodzie programistycznym.

Czy wiesz, że...?

Tworzenie wzorów za pomocą kształtów i kolorów to nie tylko świetna zabawa – to także trening tych samych umiejętności poznawczych, których programiści używają do wykrywania błędów w kodzie! Poprzez naprzemienne układanie kształtów czy zwiększanie rozmiarów, uczestnicy ćwiczą rozpoznawanie wzorców i myślenie algorytmiczne. Są to fundamentalne umiejętności w programowaniu, rozwiązywaniu problemów, a nawet w podejmowaniu codziennych decyzji.

7.6 Ćwiczenie. Gra w odgrywanie ról z problemami logicznymi

Zorganizuj gry fabularne (odgrywanie ról), w których uczestnicy wcielają się w programistów oraz roboty i muszą komunikować się ze sobą, aby rozwiązywać problemy logiczne.

Zaprojektuj scenariusz, w którym robot musi wykonać serię precyzyjnych instrukcji, aby odnaleźć ukryty skarb, używając jasnych i zwięzłych poleceń.

This activity promotes collaboration, effective communication, and algorithmic thinking as learners work together to overcome challenges.

Zachęcaj uczestników do wspólnej pracy, aby oswoili się ze wspólnym tworzeniem projektów (collaborative development) – podejściem powszechnie stosowanym w profesjonalnych projektach programistycznych.

Kiedy uczestnicy wcielają się w role programistów i robotów, nie tylko świetnie się bawią – rozwijają również kluczowe umiejętności rozwiązywania problemów w prawdziwym świecie!

7.7 Ćwiczenie. Refleksja nad codziennymi problemami

W tym ćwiczeniu uczestnicy mają za zadanie omówić w parach problemy, z którymi mierzą się w codziennym życiu. Rozpoczniesz dialog, pytając: „Z jakimi problemami spotykacie się na co dzień w domu lub w szkole?"

Będziesz zapisywać pomysły na tablicy. Twoim celem jest oswojenie uczestników z koncepcją codziennych problemów, które można rozwiązać. Po 15 minutach poprowadzisz burzę mózgów z całą klasą, podczas której uczestnicy podzielą się omówionymi przez siebie problemami.

Następnie wyjaśnij im pojęcie algorytmu w prosty sposób, posługując się przykładami z życia codziennego (takimi jak robienie kanapki lub przygotowywanie koktajlu). Powiesz im, że algorytm to po prostu seria kroków, które wykonuje się w celu ukończenia danego zadania.

Następnie podziel klasę na grupy liczące od czterech do pięciu osób i poproś, aby każdy zespół wybrał z tablicy jeden problem do wspólnego rozwiązania. Ich zadaniem jest wyraźne zdefiniowanie tego problemu, a następnie naszkicowanie na papierze prostego algorytmu. W ten sposób uczestnicy zaczną dostrzegać, jak myślenie komputacyjne (CT) może być wykorzystywane do rozwiązywania codziennych problemów.

Przekształcając codzienne problemy w proste algorytmy, uczestnicy ćwiczą te same umiejętności myślenia, których informatycy używają do rozwiązywania skomplikowanych problemów!



7.8 Ćwiczenie. Wyzwania

Wyobraź sobie sytuację, w której zauważasz, że wielu dorosłych słuchaczy spóźnia się na zajęcia, czuje ciągły pośpiech lub ma trudności z organizacją codziennych zadań. Zamiast po raz kolejny po prostu rozmawiać o zarządzaniu czasem, postanawiasz przekształcić ten realny, codzienny problem w krótkie i angażujące ćwiczenie oparte na myśleniu komputacyjnym.

Twoim wyzwaniem jest zaprojektowanie 20-minutowego ćwiczenia, które pomoże dorosłym słuchaczom zaplanować realistyczną poranną rutynę przy użyciu wyłącznie prostych materiałów typu unplugged (papier, długopisy lub karty). Masz ograniczony czas na przygotowanie, brak narzędzi cyfrowych oraz grupę o zróżnicowanych umiejętnościach.

Wyznacz jasny cel:

- Co dokładnie chcę, aby moi słuchacze przećwiczyli podczas tego ćwiczenia?
- Czy główny nacisk kładę na dekompozycję problemu, sekwencyjność działań, ustalanie priorytetów czy myślenie warunkowe?
- Która z tych umiejętności miałaby największe znaczenie (przyniosłaby największą zmianę) w ich codziennym życiu?

Możesz poprosić uczestników o wyobrażenie sobie typowego poranka i wypisanie wszystkich swoich zadań na kartkach lub karteczkach samoprzylepnych. Niech ułożą te zadania w logicznej kolejności, a następnie postaw przed nimi wyzwanie w postaci realistycznych sytuacji typu „co jeśli” (zaspanie, pominięcie śniadania, wcześniejsze spotkanie). Zaproś ich do przetestowania swojej rutyny krok po kroku, zidentyfikowania tego, co nie działa, oraz ulepszenia planu poprzez refleksję i eliminowanie błędów (debugging).

8.1 Dodatkowe zasoby

Możesz skorzystać z internetowych narzędzi do tworzenia map myśli, takich jak MindMeister lub Coggle, aby pomóc grupom zwizualizować proces dekompozycji problemów oraz sekwencjonowania ich algorytmów. Dzięki temu cały proces stanie się dla nich jaśniejszy i bardziej ustrukturyzowany.

Używaj przykładów i materiałów dydaktycznych, które uwzględniają zróżnicowaną reprezentację – zarówno w odniesieniu do algorytmów, jak i sytuacji z życia codziennego. Weź pod uwagę problemy, które odzwierciedlają doświadczenia różnych grup etnicznych i społeczno-ekonomicznych.

Twórz grupy heterogeniczne (zróżnicowane), skupiające uczestników o różnych umiejętnościach i pochodzeniu. Sprzyja to uczeniu się od rówieśników (peer learning), dzięki czemu jedna osoba może wspierać drugą w osiąganiu jej celów edukacyjnych.

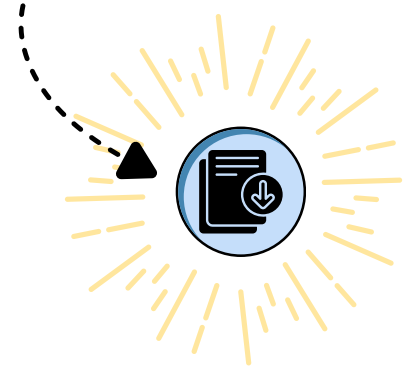


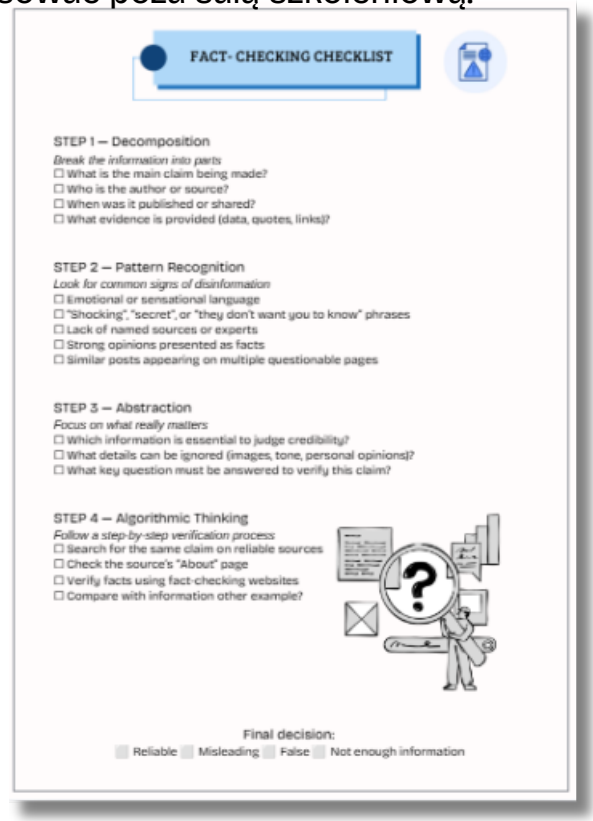
Kiedy uczestnicy dostrzegają problemy z różnych perspektyw i współpracują ze sobą, nie tylko tworzą algorytmy – budują zrozumienie, empatię i umiejętności, które pozostają na całe życie

8.2 Dodatkowe zasoby

Jako trener osób dorosłych masz okazję przekształcić codzienne informacje w inspirujący moment nauki. Ta lista kontrolna do weryfikacji faktów oferuje prosty i praktyczny sposób na przeprowadzenie uczestników przez zawiłości dezinformacji przy użyciu zasad myślenia komputacyjnego. Postępując zgodnie z jasnymi krokami, uczestnicy przechodzą od instynktownych reakcji do ustrukturyzowanego rozumowania, rozwijając umiejętności, które mogą pewnie stosować poza salą szkoleniową.

Pobierz szablon





FACT- CHECKING CHECKLIST

STEP 1 — Decomposition

Break the information into parts

- ☐ What is the main claim being made?
- ☐ Who is the author or source?
- ☐ When was it published or shared?
- ☐ What evidence is provided (data, quotes, links)?

STEP 2 — Pattern Recognition

Look for common signs of disinformation

- ☐ Emotional or sensational language
- ☐ "Shocking", "secret", or "they don't want you to know" phrases
- ☐ Lack of named sources or experts
- ☐ Strong opinions presented as facts
- ☐ Similar posts appearing on multiple questionable pages

STEP 3 — Abstraction

Focus on what really matters

- ☐ Which information is essential to judge credibility?
- ☐ What details can be ignored (images, tone, personal opinions)?
- ☐ What key question must be answered to verify this claim?

STEP 4 — Algorithmic Thinking

Follow a step-by-step verification process

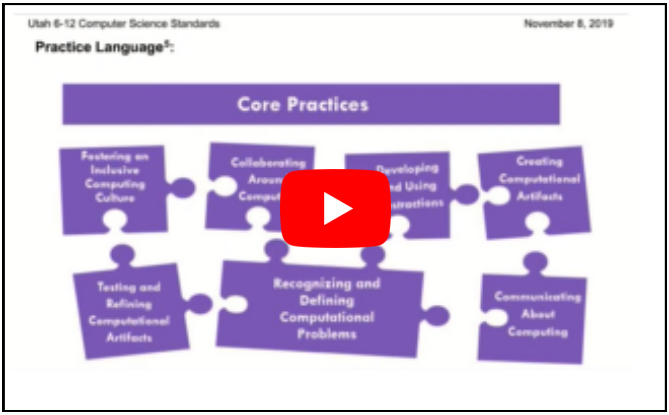
- ☐ Search for the same claim on reliable sources
- ☐ Check the source's "About" page
- ☐ Verify facts using fact-checking websites
- ☐ Compare with information other example?

Final decision:

Reliable Misleading False Not enough information

8.3 Dodatkowe zasoby







9. Jaka jest wartość tego modułu?

Wartość tej jednostki lekcyjnej polega na wykazaniu, że materiały i zasoby nie są jedynie uzupełniającymi pomocami dydaktycznymi, lecz stanowią niezbędne elementy skutecznego projektowania procesu uczenia się. Ich dobór, przygotowanie i prezentacja bezpośrednio wpływają na to, jak dorośli słuchacze rozumieją myślenie komputacyjne (CT), jak się w nie angażują i jak je stosują.

Ta jednostka lekcyjna umożliwia trenerom podejmowanie świadomych decyzji o tym, które materiały najlepiej wspierają określone cele edukacyjne oraz potrzeby uczestników. Pokazuje, w jaki sposób ćwiczenia typu unplugged, znane przedmioty codziennego użytku, narzędzia cyfrowe lub połączenie tych zasobów mogą uczynić abstrakcyjne pojęcia myślenia komputacyjnego (CT) bardziej konkretnymi, przystępnymi i odniesionymi do sytuacji z prawdziwego życia.

Ta jednostka lekcyjna podkreśla również, że skuteczna edukacja w zakresie myślenia komputacyjnego (CT) nie wymaga zaawansowanej wiedzy technologicznej. Jej wartość polega na wzmacnianiu zdolności trenerów do rozpoczynania od prostych, łatwych do przeprowadzenia ćwiczeń, dostosowywania zasobów do różnych uczestników, usuwania barier w uczestnictwie oraz kładzenia nacisku na procesy myślowe, a nie na złożoność technologiczną.

W ostatecznym rozrachunku ta jednostka lekcyjna wzmacnia zdolność trenerów do tworzenia włączających, praktycznych i angażujących doświadczeń edukacyjnych. Poprzez przygotowanie wartościowych i elastycznych zasobów trenerzy mogą pomóc dorosłym słuchaczom w budowaniu pewności siebie, praktykowaniu rozwiązywania problemów oraz przenoszeniu umiejętności myślenia komputacyjnego na ich codzienne życie.