

KOMPLET LEKCJI

Python od zera, zmienne i dane od użytkownika

Programowanie

Klasy VII i VIII

45 minut

Lekcja otwarta

Uczeń pisze w Pythonie program, który pobiera dane od użytkownika, zapisuje je w zmiennych i wypisuje policzony wynik.

W tym pliku są wszystkie materiały do lekcji: scenariusz z kluczem do karty pracy i kryteriami oceny, prezentacja, karta pracy, materiały dla ucznia, kartka dla rodzica oraz quiz w wersji dla ucznia i z odpowiedziami.

Co znajdziecie w środku

1. Scenariusz lekcji
2. Prezentacja na tablicę
3. Karta pracy do druku
4. Materiały dla ucznia
5. Kartka dla rodzica
6. Quiz dla ucznia
7. Quiz z odpowiedziami

SCENARIUSZ LEKCJI

Python od zera, zmienne i dane od użytkownika

Cel lekcji

Uczeń pisze w Pythonie program, który pobiera dane od użytkownika, zapisuje je w zmiennych i wypisuje policzony wynik.

Czego uczy ta lekcja

Zmienne i typy danych

Zapisujesz dane w zmiennych i rozróżniasz tekst od liczby.

Dane wejściowe i wynik

Pobierasz dane funkcją input, zamieniasz je na liczbę i wypisujesz wynik funkcją print.

Potrzebny sprzęt

Laptopy szkolne z przeglądarką, jeden laptop na ucznia lub na parę, projektor albo tablica do pokazu. Internet potrzebny tylko wtedy, gdy pracujecie w edytorze w przeglądarce. Jeśli na laptopach jest zainstalowane Thonny (darmowy edytor Pythona), lekcja działa bez internetu. W przeglądarce użyjcie edytora, który uruchamia Python 3 bez zakładania konta przez ucznia, na przykład Online Python Compiler na stronie programiz.com albo trinket.io w trybie bez logowania. Sprawdźcie dzień wcześniej, że wybrany edytor przyjmuje dane z klawiatury przez input, bo nie każdy edytor online to obsługuje.

Podstawa programowa

Informatyka, Szkoła podstawowa, klasy VII i VIII

Cele kształcenia, wymagania ogólne

- I. Rozumienie, analizowanie i rozwiązywanie problemów na bazie logicznego i abstrakcyjnego myślenia, myślenia algorytmicznego i sposobów reprezentowania informacji.
- II. Programowanie i rozwiązywanie problemów z wykorzystaniem komputera oraz innych urządzeń cyfrowych: układanie i programowanie algorytmów, organizowanie, wyszukiwanie i udostępnianie informacji, posługiwanie się aplikacjami komputerowymi.

Treści nauczania, wymagania szczegółowe

- II.1. projektuje, tworzy i testuje programy w procesie rozwiązywania problemów. W programach stosuje: instrukcje wejścia / wyjścia, wyrażenia arytmetyczne i logiczne, instrukcje warunkowe, instrukcje iteracyjne, funkcje oraz zmienne i tablice. W szczególności programuje algorytmy z działu I pkt 2; (Pełne pokrycie celu lekcji. Punkt wymienia wprost instrukcje wejścia / wyjścia oraz zmienne, czyli dokładnie to, co uczeń robi na tej lekcji w Pythonie.)
- I.1. formułuje problem w postaci specyfikacji (czyli opisuje dane i wyniki) oraz wyróżnia kroki w algorytmicznym rozwiązywaniu problemów. Stosuje różne sposoby przedstawiania algorytmów, w tym w języku naturalnym, w postaci listy kroków; (Podstawa dla wątku specyfikacji: uczeń opisuje dane (to, co pobiera od użytkownika) i wyniki (to, co program wypisuje).)

Podstawa prawna: Rozporządzenie Ministra Edukacji Narodowej z dnia 14 lutego 2017 r. w sprawie podstawy programowej wychowania przedszkolnego oraz podstawy programowej kształcenia ogólnego dla szkoły podstawowej, w tym dla uczniów z niepełnosprawnością intelektualną w stopniu umiarkowanym lub znacznym, kształcenia ogólnego dla branżowej szkoły I stopnia, kształcenia ogólnego dla szkoły specjalnej przysposabiającej do pracy oraz kształcenia ogólnego dla szkoły policealnej, Dz.U. 2017 poz. 356.

Brzmienie: Załącznik nr 2 do Dz.U. 2017 poz. 356 w brzmieniu nadanym przez Dz.U. 2024 poz. 996, część KLASY VII i VIII

Lista kontrolna przed lekcją

Do odhaczenia dzień wcześniej i 10 minut przed dzwonkiem.

- ☐ Dzień wcześniej: uruchomcie na jednym szkolnym laptopie program z jedną linią `print("test")` w edytorze, którego użyje klasa (Thonny albo edytor w przeglądarce bez konta).
- ☐ Dzień wcześniej: sprawdźcie w tym samym edytorze program z `input`, bo część edytorów online nie przyjmuje danych z klawiatury. Wpiszcie liczbę i upewnijcie się, że program ją odczytał.
- ☐ Dzień wcześniej: wydrukujcie kartę pracy, po jednej na ucznia, a dla wariantu łatwiejszego przygotujcie wydruk programu z trzema lukami (kod jest w kluczu karty pracy, zadanie 3).
- ☐ Dzień wcześniej: zapiszcie w jednym pliku działające wzorce zadań 1 do 3 z klucza karty pracy, żeby przy obchodzie porównywać kod ucznia ze wzorcem.
- ☐ 10 minut przed lekcją: włączcie projektor i otwórzcie edytor w powiększeniu czcionki, tak żeby kod był czytelny z ostatniej ławki (w Thonny: menu Widok, Powiększ).
- ☐ 10 minut przed lekcją: napiszcie na tablicy adres edytora w przeglądarce albo nazwę programu do otwarcia oraz nazwę pliku do zapisania na koniec, na przykład `lekcja1_imie.py`.
- ☐ 10 minut przed lekcją: przygotujcie na kartce trzy linie z rozgrzewki, żeby nie pisać ich na żywo i nie tracić czasu na start.
- ☐ Na wypadek awarii: przygotujcie po trzy karteczki na ucznia z napisami imię, cena, `liczba_osob` do wariantu bez sprzętu.

Rozgrzewka na start

BEZ SPRZĘTU, 3 MIN

Pokażcie na projektorze albo napiszcie na tablicy trzy linie i zapytajcie: co pojawi się na ekranie po uruchomieniu?

```
imie = "Ola"
```

```
wiek = 13
```

```
print(imie, "ma", wiek, "lat")
```

Uczniowie odpowiadają z miejsca, bez komputerów. Potem zapytajcie, co się zmieni, jeśli w drugiej linii wpisze 14.

Czego się spodziewać: Klasa powinna dojść do wypisu: Ola ma 13 lat. Po zmianie drugiej linii: Ola ma 14 lat. Jeśli ktoś odpowie, że wypisze się słowo imie albo wiek, to dobry moment na zdanie: nazwa bez cudzysłowu oznacza zawartość pudełka, a z cudzysłowem sam napis.

Przebieg lekcji

Pięć etapów, razem 45 minut.

1

Wejście, 5 min

Nauczyciel pokazuje na projektorze trzy linijki kodu w Pythonie (kod jest w polu rozgrzewka) i pyta klasę, co się wypisze. Uczniowie odpowiadają z miejsca, bez komputerów. Nauczyciel podaje cel lekcji: każdy wychodzi z działającym programem, który pyta o dane i liczy wynik. Uczniowie logują się do laptopów i otwierają edytor wskazany przez nauczyciela: Thonny albo edytor w przeglądarce, którego adres jest zapisany na tablicy.

2

Pokaz, 10 min

Nauczyciel pisze na żywo program liczący koszt wycieczki. Krok po kroku: przypisanie do zmiennej, wypisanie zmiennej przez print, pobranie danych przez input, zamiana tekstu na liczbę przez int albo float, wypisanie wyniku. Po każdym kroku uruchamia program, żeby klasa widziała efekt. Nauczyciel celowo popełnia jeden błąd: dodaje wartość prosto z input do liczby, na przykład liczba_uczniow + 1, pokazuje komunikat TypeError i naprawia go, owijając input funkcją int. Zapisuje na tablicy cztery słowa do zapamiętania: zmienna, print, input, int.

3

Praca uczniów, 20 min

Uczniowie pracują z kartą pracy. Zadanie 1: program wypisuje powitanie z imieniem pobranym od użytkownika. Zadanie 2: program pyta o dwie liczby i wypisuje sumę. Zadanie 3: kalkulator kosztu wyjścia klasowego, cena biletu razy liczba uczniów plus koszt przejazdu. Zadania 4 i 5 na karcie pracy to tabela typów i krótkie doświadczenie z usunięciem int, do zrobienia po zadaniu 3. Dla chętnych, którzy skończą wszystko: program liczy, ile minut ma podana liczba godzin i minut. Nauczyciel chodzi po sali, nie poprawia kodu za ucznia, tylko pyta, którą linię uruchomił jako ostatnią i co pokazał komunikat błędu.

4

Sprawdzenie, 6 min

Dwoje uczniów pokazuje swój program na projektorze i wpisuje dane podane przez klasę. Nauczyciel prosi o wskazanie w kodzie miejsca, w którym powstaje zmienna, i miejsca, w którym tekst staje się liczbą. Klasa sprawdza, czy program działa też dla liczby uczniów równej 0. Program z zadania 3 ma wtedy wypisać sam koszt przejazdu.

5

Podsumowanie, 4 min

Nauczyciel wraca do czterech słów z tablicy i prosi, żeby każdy uczeń zapisał w zeszycie jedno zdanie: co robi input i dlaczego potrzebna jest funkcja int. Zapowiada następną lekcję, na której te same zmienne posłużą do zbudowania gry tekstowej. Uczniowie zapisują swój plik pod nazwą wskazaną przez nauczyciela.

Do powiedzenia na głos

„Zmienna to pudełko z nazwą. Wkładacie do niego wartość, a potem wołacie po nazwie i dostajecie to, co w środku.”

„Funkcja input zawsze oddaje tekst, nawet gdy wpisaliście liczbę. Dlatego zanim policzycie, zamieniacie tekst na liczbę funkcją int albo float.”

„Komunikat błędu, ten wyświetlany zwykle na czerwono pod programem, nie znaczy, że zepsuliście komputer. To najkrótsza informacja, w której linii jest problem, więc przeczytajcie ostatnią linijkę komunikatu.”

„Na koniec lekcji każdy ma działający program, który pyta o dane i podaje wynik. Nie o piękny kod chodzi, tylko o to, żeby ruszył.”

Typowe błędy uczniów i co z nimi zrobić

- Uczeń liczy na tekście z funkcji input i dostaje `TypeError`. Poproś o pokazanie linii z input i wspólnie dopiszcie `int` wokół niej.
- Uczeń myli przypisanie z porównaniem i pisze dwa znaki równości przy tworzeniu zmiennej. Przypomnij, że jeden znak równości zapisuje wartość do zmiennej.
- Uczeń zapomina nawiasów albo cudzysłowów i dostaje `SyntaxError`. Pokaż, że Python wskazuje numer linii, i naucz sprawdzać parę otwierającą i zamykającą.
- Uczeń nazywa zmienną od cyfry albo wstawia spację w nazwie i dostaje `SyntaxError`. Ustalcie zasadę: nazwa zaczyna się od litery, nie ma spacji, słowa łączy podkreślenie. Polskie znaki Python 3 przyjmie, ale odradźcie je, bo łatwo pomylić `a` z `ą` i szukać literówki przez pół lekcji.
- Uczeń pisze `Print` zamiast `print`. Przypomnij, że Python rozróżnia wielkie i małe litery.

Warianty lekcji

Bez sprzętu i internetu

Pracujecie na kartkach i na tablicy. Każdy uczeń dostaje trzy karteczki podpisane jako zmienne: `imie`, `cena`, `liczba_osob`. Nauczyciel dyktuje instrukcje, a uczniowie wpisują wartości na karteczki i wykreślają stare przy ponownym przypisaniu, żeby zobaczyć, że zmienna trzyma jedną wartość naraz. Potem w parach zapisują w zeszytach program liczący koszt wyjścia klasowego i wykonują go ręcznie dla trzech zestawów danych podanych przez nauczyciela. Na koniec każda para czyta kod na głos, a klasa podaje wynik.

Dla szybszych uczniów

Rozbuduj kalkulator tak, żeby pytał o cenę biletu, liczbę uczniów, liczbę opiekunów i koszt przejazdu, a potem wypisał koszt całkowity oraz koszt na jednego ucznia zaokrąglony do dwóch miejsc funkcją `round`. Dodaj wypisanie krótkiego podsumowania w jednej linii, na przykład: Wyjście dla 24 uczniów kosztuje 720 złotych, czyli 30 złotych na osobę. Uwaga: przy zerowej liczbie uczniów dzielenie zatrzyma program komunikatem `ZeroDivisionError`. Zapisz w komentarzu, dlaczego tak się dzieje. Zabezpieczenie warunkiem `if` poznasz na kolejnych lekcjach.

Przy trudnościach

Dostajesz gotowy program z trzema lukami oznaczonymi komentarzem. Twoje zadanie to wpisać tylko nazwę zmiennej, funkcję `int` i nazwę zmiennej w `print`. Zamiast pobierania danych funkcją `input` możesz na starcie wpisać wartości wprost do zmiennych i uruchomić program, a `input` dodać dopiero na końcu.

Kryteria oceny

Trzy poziomy dla każdej umiejętności. Opis mówi, co uczeń pokazuje, a nie ile punktów dostaje.

UMIĘJĘTNOŚĆ	PODSTAWOWY	DOBRY	BARDZO DOBRY
Zmienne i typy danych	Uczeń tworzy zmienną z jednym znakiem równości, nadaje jej poprawną nazwę (bez spacji, nie od cyfry) i wypisuje jej wartość funkcją <code>print</code> . Rozróżnia tekst w cudzysłowie od liczby bez cudzysłowu.	Uczeń dobiera typ do wartości: <code>int</code> dla liczby uczniów, <code>float</code> dla ceny z groszami. Wyjaśnia, że nowe przypisanie kasuje starą wartość, i przewiduje wynik programu, w którym zmienna zmienia wartość dwa razy.	Uczeń przewiduje bez uruchamiania wynik działań na różnych typach, na przykład sklejenie dwóch tekstów albo powtórzenie tekstu pomnożonego przez liczbę całkowitą, i uzasadnia, dlaczego to nie jest błąd programu. Nazywa zmienne tak, że kod czyta się bez komentarzy.
Dane wejściowe i wynik	Uczeń pobiera dane funkcją <code>input</code> z tekstem zachęty, zapisuje je do zmiennej i wypisuje wynik funkcją <code>print</code> . Powtarza zasadę: <code>input</code> zawsze oddaje tekst.	Uczeń samodzielnie owija <code>input</code> funkcją <code>int</code> albo <code>float</code> i pisze działający program liczący sumę oraz kalkulator kosztu wyjścia klasowego, w którym wynik jest liczony w osobnej zmiennej.	Uczeń czyta komunikat błędu, wskazuje linię i naprawia <code>TypeError</code> bez pomocy nauczyciela. Sprawdza program dla danych brzegowych, na przykład zera, i wyjaśnia, dlaczego wynik 78 zamiast 15 nie kończy się komunikatem błędu.

Pytania do dyskusji

Na podsumowanie lekcji. Nie mają jednej poprawnej odpowiedzi, chodzi o uzasadnienie.

1. Program wypisał 78 zamiast 15 i nie pokazał żadnego błędu. Który błąd jest groźniejszy: taki, który zatrzymuje program, czy taki, który po cichu daje zły wynik? Gdzie w prawdziwym życiu taki cichy błąd mógłby kosztować pieniądze?

2. Komputer nie zgaduje, czy 5 to liczba, czy znak. Czy to wada, czy zaleta? Jak wyglądałoby programowanie, gdyby komputer próbował domyślać się, o co nam chodzi?
3. Nasz kalkulator liczy koszt wyjścia klasowego. Jakie dane wpisane przez użytkownika mogłyby go zmylić (ujemna liczba uczniów, litera zamiast ceny, przecinek zamiast kropki) i kto powinien to sprawdzać: program czy człowiek?

Częste pytania uczniów

Po co zamieniać tekst na liczbę, skoro wpisałem cyfrę?

Bo komputer nie zgaduje, co miałeś na myśli. Input oddaje ciąg znaków, a znak 5 to nie to samo co liczba 5. Funkcja `int` mówi wprost: potraktuj to jako liczbę.

Czym różni się `int` od `float`?

`int` to liczba całkowita, `float` to liczba z częścią dziesiętną. Liczbę uczniów zapisz jako `int`, cenę z groszami jako `float`.

Czy mogę nazwać zmienną dowolnie?

Prawie. Nazwa nie może mieć spacji ani zaczynać się od cyfry, a wielkość liter ma znaczenie. Wybieraj nazwy mówiące, co jest w środku, na przykład `cena_biletu`, a nie `x`.

Mini projekt na kolejne lekcje

Na 2 do 3 kolejne lekcje albo na kółko. Buduje na tym, co uczniowie zrobili na tej lekcji.

Kalkulator wyjścia klasowego z podsumowaniem

Na dwóch do trzech kolejnych lekcjach albo na kółku uczniowie rozbudowują kalkulator z tej lekcji w narzędzie, którym samorząd klasowy policzy prawdziwe wyjście: do kina, na basen albo na wycieczkę. Projekt buduje na zmiennych, input, int i float, a w kolejnych krokach dokłada zaokrąglanie funkcją round i pierwsze warunki if.

1. Krok 1: kalkulator pyta o cenę biletu, liczbę uczniów, liczbę opiekunów i koszt przejazdu, a potem wypisuje koszt całkowity i koszt na jednego ucznia zaokrąglony funkcją round do dwóch miejsc.
2. Krok 2: uczniowie dodają zmienną z kwotą zebraną do tej pory i program wypisuje, ile jeszcze brakuje albo ile zostanie. Sprawdzają program dla trzech prawdziwych zestawów danych z życia klasy.
3. Krok 3: po poznaniu instrukcji if program sprawdza, czy liczba uczniów jest większa od zera, i zamiast błędu ZeroDivisionError wypisuje czytelny komunikat. Opiekunowie wchodzić bezpłatnie, jeśli klasa tak ustali.
4. Krok 4: pary uczniów wymieniają się programami i testują cudzy kod, wpisując dane, które mogą go zepsuć. Każda para zapisuje, co znalazła, i oddaje autorowi do naprawy.

Co powstaje na końcu: Działający program w jednym pliku, który liczy koszt prawdziwego wyjścia klasowego i podaje kwotę na osobę, oraz krótka lista danych wejściowych, które program obsłużył poprawnie, przetestowana przez inną parę.

PREZENTACJA NA TABLICĘ

Python od zera, zmienne i dane od użytkownika

Slajdów: 8. Notatka pod slajdem jest dla nauczyciela, uczniowie nie widzą jej na tablicy.

SLAJD 1

Po co nam programowanie

- Komputer robi dokładnie to, co mu napiszesz
- Program to lista poleceń w ustalonej kolejności
- Dziś pisziesz program, który liczy za Ciebie

NOTATKA DLA NAUCZYCIELA

Zacznijcie od pytania, co uczniowie liczą ręcznie na telefonie, i pokażcie, że program robi to raz na zawsze.

SLAJD 2

Zmienna to pudełko z nazwą

- Nazwa mówi, co jest w środku
- Jeden znak równości zapisuje wartość
- Nowe przypisanie kasuje starą wartość

NOTATKA DLA NAUCZYCIELA

Pokażcie na żywo dwa przypisania do tej samej zmiennej i wypiszcie ją po każdym, żeby zobaczyli zmianę.

SLAJD 3

Tekst czy liczba

- Tekst zapisujemy w cudzysłowie
- Liczby całkowite to int, z przecinkiem to float
- Tekst i liczby nie dodają się tak samo

NOTATKA DLA NAUCZYCIELA

Pokażcie, że dodanie dwóch tekstów skleja je, a dodanie dwóch liczb sumuje.

SLAJD 4

print, czyli wypisz

- print pokazuje wynik użytkownikowi
- Można wypisać tekst i zmienną naraz
- Bez print program liczy po cichu

NOTATKA DLA NAUCZYCIELA

Uruchomcie program bez print, żeby zobaczyli puste okno wyniku.

SLAJD 5

input, czyli zapytaj

- input zatrzymuje program i czeka na dane
- W nawiasie wpisujesz pytanie dla użytkownika
- input zawsze oddaje tekst

NOTATKA DLA NAUCZYCIELA

To najważniejszy slajd lekcji. Powtórzcie na głos: input oddaje tekst.

SLAJD 6

Zamiana tekstu na liczbę

- int zamienia tekst na liczbę całkowitą
- float zamienia tekst na liczbę z przecinkiem
- Bez zamiany dostaniesz błąd albo sklejony tekst

NOTATKA DLA NAUCZYCIELA

Pokażcie oba efekty: komunikat TypeError przy mnożeniu tekstu przez tekst i sklejanie przy dodawaniu.

SLAJD 7

Czytamy komunikat błędu

- Numer linii mówi, gdzie szukać
- Ostatnia linijka mówi, co jest nie tak

- Błąd to wskazówka, nie kara

NOTATKA DLA NAUCZYCIELA

Poproście uczniów, żeby przed podniesieniem ręki przeczytali ostatnią linijkę komunikatu na głos.

SLAJD 8**Twoje zadanie**

- Powitanie z imieniem
- Suma dwóch liczb
- Kalkulator kosztu wyjścia klasowego
- Uruchamiaj po każdej linii

NOTATKA DLA NAUCZYCIELA

Podajcie czas: 20 minut. Powiedzcie, że zadanie trzecie jest obowiązkowe, czwarte dla chętnych.

KARTA PRACY DO DRUKU

Python od zera, zmienne i dane od użytkownika

Imię ucznia: Klasa: Data:

Dziś napiszesz swoje pierwsze programy w Pythonie. Każde zadanie uruchamiaj od razu po napisaniu, bo krótki program łatwiej naprawić niż długi.

1

Napisz program, który pyta o imię i wypisuje powitanie, na przykład: Cześć, Kasia, dobrze Cię widzieć.

Wskazówka: Użyj input z tekstem zachęty w nawiasie, zapisz wynik do zmiennej imie, a potem wypisz ją funkcją print razem z tekstem powitania, oddzielając elementy przecinkiem.

2

Napisz program, który pyta o dwie liczby i wypisuje ich sumę.

Wskazówka: Obie wartości z input zamień na liczby funkcją int, dopiero potem dodawaj. Sprawdź program dla liczb 7 i 8.

3

Napisz kalkulator kosztu wyjścia klasowego. Program pyta o cenę biletu, liczbę uczniów i koszt przejazdu, a potem wypisuje koszt całkowity.

Wskazówka: Cenę biletu i koszt przejazdu zapisz jako float, liczbę uczniów jako int. Koszt całkowity to cena biletu razy liczba uczniów plus koszt przejazdu. Policz go w osobnej zmiennej, zanim go wypiszesz.

4

Uzupełnij tabelę: dla każdej z trzech zmiennych pobranych od użytkownika w zadaniu 3 wpisz jej nazwę, przykładową wartość i typ, czyli tekst, liczba całkowita albo liczba z częścią dziesiętną.

Wskazówka: Typ poznasz po tym, jak zamierzasz użyć wartości. Jeśli będziesz liczyć, to liczba. Jeśli liczba może mieć grosze albo połówki, to liczba z częścią dziesiętną.

5

Zapisz w dwóch zdaniach, co się stanie, jeśli usuniesz obie funkcje int z programu z zadania 2. Sprawdź to w edytorze.

Wskazówka: Uruchom program i wpisz 7 oraz 8. Przeczytaj, co wypisał, i porównaj z wynikiem dodawania.

MATERIAŁY DLA UCZNIA

Python od zera, zmienne i dane od użytkownika

Czego się nauczysz

Zmienne i typy danych

Zapisujesz dane w zmiennych i rozróżniasz tekst od liczby.

Dane wejściowe i wynik

Pobierasz dane funkcją input, zamieniasz je na liczbę i wypisujesz wynik funkcją print.

Program w Pythonie to lista poleceń wykonywanych po kolei, od góry do dołu. Żeby napisać coś użytecznego, wystarczą cztery rzeczy: zmienna, print, input i zamiana tekstu na liczbę. Z tego materiału nauczysz się każdej z nich osobno i dowiesz się, dlaczego program, który pyta o dwie liczby i wypisuje 78 zamiast 15, wcale się nie zepsuł.

Zmienna, czyli pudełko z nazwą

Zmienna to miejsce w pamięci, do którego zapisujesz wartość i któremu nadajesz nazwę, żeby móc do niej wrócić. Zapis wygląda tak: `cena = 12`. Po lewej stronie stoi nazwa, po prawej wartość, a jeden znak równości oznacza polecenie „zapisz to po prawej do tego po lewej”. To nie jest równanie z matematyki i nie znaczy, że cena równa się 12. Znaczy: włóż liczbę 12 do pudełka o nazwie cena.

Zmienna trzyma jedną wartość naraz. Gdy napiszesz `cena = 12`, a liniijkę niżej `cena = 20`, stara wartość znika bezpowrotnie i w pudełku zostaje 20. Ta kolejność jest ważna, bo Python wykonuje linie po kolei i zawsze używa tego, co w zmiennej jest w danym momencie.

Dwa znaki równości to co innego. Zapis `cena == 12` nie zapisuje niczego, tylko pyta, czy w zmiennej cena siedzi 12, i daje odpowiedź prawda albo fałsz. To częste źródło pomyłek na początku.

Nazwa zmiennej podlega kilku regułom. Nie może zaczynać się od cyfry, nie może zawierać spacji, lepiej nie używać w niej polskich znaków. Poza tym Python rozróżnia wielkie i małe litery, więc `Cena` i `cena` to dwie różne zmienne. Wybieraj nazwy, które mówią, co jest w środku: `cena_biletu` jest lepsze niż `x`, bo za tydzień nadal będziesz wiedzieć, o co chodziło.

SPRÓBUJ SAM

Wpisz w edytorze trzy linie i uruchom: `cena = 12`, potem `cena = 20`, potem `print(cena)`. Zastanów się przed uruchomieniem, co się wypisze, a dopiero potem sprawdź. Następnie zamień kolejność dwóch pierwszych linii i uruchom ponownie.

Tekst czy liczba, czyli typy danych

Python rozróżnia rodzaje wartości i nazywa je typami. Na tej lekcji potrzebujesz trzech. Tekst, czyli napis, zapisujesz w cudzysłowie: "Kasia". Liczbę całkowitą, czyli int, zapisujesz bez cudzysłowu: 24. Liczbę z częścią dziesiętną, czyli float, zapisujesz z kropką, nie z przecinkiem: 12.50.

Typ decyduje o tym, co da się z wartością zrobić. Liczby się dodają, mnożą i dzielą. Teksty też można dodać, ale dodawanie tekstów oznacza sklejanie: "7" + "8" daje "78", bo Python nie liczy, tylko dostawia jeden napis za drugim. To nie jest błąd programu, tylko poprawne zachowanie zgodne z typem.

Ta sama cyfra może być jednym albo drugim, w zależności od zapisu. 5 to liczba, a "5" to tekst zawierający znak pięć. Wyglądają na ekranie tak samo, a zachowują się zupełnie inaczej. Dlatego przy każdej wartości warto sobie odpowiedzieć na jedno pytanie: czy będę tym liczyć. Jeśli tak, potrzebujesz liczby.

Int czy float wybierasz według tego, czy wartość ma część dziesiętną. Liczba uczniów to int, bo pół ucznia nie istnieje. Cena z groszami to float, bo int obciąłby grosze.

SPRÓBUJ SAM

Uruchom kolejno dwie linie: `print(7 + 8)`, a potem `print("7" + "8")`. Zapisz oba wyniki obok siebie i napisz jednym zdaniem, czym różnią się te dwa działania, choć znak dodawania jest w obu ten sam.

print i input, czyli rozmowa z użytkownikiem

Funkcja `print` wypisuje na ekranie to, co dostanie w nawiasie. Bez niej program nadal liczy, tylko po cichu, i nie zobaczysz żadnego wyniku. To częste rozczarowanie: program działa poprawnie, a ekran jest pusty, bo nikt nie kazał niczego wypisać.

W nawiasie `print` możesz podać tekst w cudzysłowie, zmienną bez cudzysłowu albo kilka rzeczy po przecinku. Zapis `print("Koszt to", koszt, "złotych")` wypisze wszystkie trzy elementy w jednej linii, oddzielone spacjami. Uważaj na cudzysłowy: `print("koszt")` wypisze słowo koszt, a `print(koszt)` wypisze zawartość zmiennej. To dwa różne polecenia.

Funkcja `input` działa odwrotnie: zatrzymuje program i czeka, aż użytkownik coś wpisze i naciśnie Enter. Tekst podany w nawiasie `input` pojawia się jako zachęta, na przykład `input("Podaj swoje imię: ")`. To, co użytkownik wpisze, trzeba od razu zapisać do zmiennej, inaczej wartość przepadnie: `imię = input("Podaj swoje imię: ")`.

Jest jedna rzecz o input, którą trzeba zapamiętać na zawsze i którą opisuje następna sekcja. Input zawsze oddaje tekst. Zawsze, nawet gdy użytkownik wpisał same cyfry.

SPRÓBUJ SAM

Napisz program z dwóch linii: `imie = input("Podaj swoje imie: ")` oraz `print("Czesc", imie, "dobrze Cie widziec")`. Uruchom go i wpisz swoje imię. Potem usuń cudzysłowy wokół słowa Czesc i uruchom ponownie. Przeczytaj komunikat, który się pojawi.

Zamiana tekstu na liczbę, czyli int i float

Skoro input zawsze oddaje tekst, to program, który pyta o dwie liczby i je dodaje, bez dodatkowego kroku nie policzy sumy. Użytkownik wpisuje 7 i 8, ale do zmiennych trafiają napisy "7" i "8", a dodawanie napisów skleja je w "78". To najczęstsza pułapka pierwszych programów i wcale nie kończy się komunikatem błędu, więc łatwo jej nie zauważyć.

Rozwiązaniem jest jawna zamiana typu. Funkcja `int` bierze tekst i oddaje liczbę całkowitą: `liczba = int(input("Podaj liczbę: "))`. Funkcja `float` robi to samo dla liczb z częścią dziesiętną. Zapis czyta się od środka na zewnątrz: najpierw wykonuje się input, potem jego wynik trafia do `int`.

Jeśli spróbujesz liczyć na tekście bez zamiany, na przykład pomnożyć napis przez liczbę zmiennoprzecinkową, Python zatrzyma program i pokaże `TypeError`. Ten komunikat oznacza dokładnie jedno: użyłeś typu, który do tego działania nie pasuje. To nie jest awaria, tylko informacja, gdzie brakuje `int` albo `float`.

Działa to też w drugą stronę. Jeśli użytkownik wpisze słowo zamiast cyfry, `int` nie ma czego zamienić i program zatrzyma się z komunikatem `ValueError`. Na razie wystarczy wiedzieć, skąd on się bierze.

SPRÓBUJ SAM

Napisz program: `a = int(input("Pierwsza liczba: "))`, `b = int(input("Druga liczba: "))`, `print(a + b)`. Uruchom i wpisz 7 oraz 8. Potem usuń obie funkcje `int`, uruchom ponownie z tymi samymi danymi i zapisz oba wyniki obok siebie razem z wyjaśnieniem różnicy.

Czytanie komunikatu błędu

Komunikat błędu, wyświetlany zwykle na czerwono pod programem, nie jest karą ani znakiem, że nie nadajesz się do programowania. To najbardziej użyteczna informacja, jaką Python może Ci dać, bo zwykle wskazuje palcem miejsce i rodzaj problemu. Warto czytać go w dwóch miejscach: numer linii mówi, gdzie szukać, a ostatnia linijka komunikatu mówi, co jest nie tak.

Trzy rodzaje spotkasz najczęściej. `SyntaxError` oznacza, że Python nie rozumie zapisu: zwykle brakuje nawiasu zamykającego, cudzysłowu albo dwukropka. Sprawdź wtedy, czy każdy nawias otwierający ma swoją parę. `TypeError` oznacza działanie na niepasujących typach, czyli najczęściej liczenie na tekście z `input`. `NameError` oznacza, że użyłeś nazwy, której Python nie zna: literówka w nazwie zmiennej albo `Print` zamiast `print`, bo wielkość liter ma znaczenie.

Warto też wiedzieć, że numer linii bywa przybliżony. Przy brakującym nawiasie Python zauważa problem dopiero w następnej linii, więc sprawdzaj również linię powyżej wskazanej.

Najlepszy nawyk na początek to uruchamianie programu po każdej dopisanej linii, a nie po napisaniu całości. Kiedy dopisujesz jedną linię i program przestaje działać, wiesz dokładnie, która linia zawiniła. Kiedy dopisujesz dwadzieścia, zaczyna się zgadywanie.

SPRÓBUJ SAM

Zepsuj celowo działający program na trzy sposoby, za każdym razem na chwilę: usuń jeden nawias zamykający, napisz `Print` wielką literą, wykonaj dodawanie na wartości z `input` bez `int`. Zapisz nazwę błędu i numer linii dla każdego przypadku. Zbuduj z tego własną ściągę, którą sprawdzisz przy następnym programie.

Słowniczek

zmienna

Nazwane miejsce w pamięci, do którego zapisujesz jedną wartość i możesz ją później odczytać albo nadpisać.

przypisanie

Zapis z jednym znakiem równości, który wkłada wartość z prawej strony do zmiennej z lewej strony.

typ danych

Rodzaj wartości decydujący o tym, co można z nią zrobić: tekst, liczba całkowita albo liczba z częścią dziesiętną.

int

Liczba całkowita, a także funkcja zamieniająca tekst na taką liczbę.

float

Liczba z częścią dziesiętną zapisywaną po kropce, a także funkcja zamieniająca na nią tekst.

print

Funkcja wypisująca na ekranie to, co dostanie w nawiasie.

input

Funkcja, która zatrzymuje program, czeka na dane od użytkownika i zawsze oddaje je jako tekst.

TypeError

Komunikat oznaczający próbę wykonania działania na wartościach niepasującego typu, najczęściej liczenia na tekście.

Częste pomyłki

- Liczenie na wartości prosto z input bez zamiany na liczbę. Dodawanie skleja wtedy napisy i z 7 oraz 8 wychodzi 78 zamiast 15. Owiń input funkcją int albo float, zanim zaczniesz liczyć.
- Użycie dwóch znaków równości przy tworzeniu zmiennej. Jeden znak zapisuje wartość, dwa znaki tylko pytają, czy wartości są równe. Przy tworzeniu zmiennej zawsze jeden.
- Nazwanie zmiennej od cyfry albo ze spacją. Python odmówi wykonania takiej linii i pokaże SyntaxError. Nazwa zaczyna się od litery, nie ma spacji, a słowa łączysz podkreśleniem, na przykład cena_biletu. Polskich znaków unikaj, bo działają, ale łatwo pomylić a z ą i długo szukać literówki.
- Pisanie Print albo INPUT wielkimi literami. Python rozróżnia wielkość liter i nie zna takiej nazwy, więc pokaże NameError. Nazwy funkcji piszemy małymi literami.

Sprawdź, czy rozumiesz

1. Program pyta o dwie liczby i wypisuje 1520 zamiast 35. Czego w nim brakuje i w której dokładnie linii to naprawisz?
2. Kiedy przy pobieraniu ceny użyjesz int, a kiedy float, i co konkretnie stanie się z ceną 12 złotych 50 groszy, jeśli wybierzesz źle?
3. Program wypisuje NameError w linii z print. Wymień dwie różne przyczyny, które mogą dawać ten sam komunikat, i powiedz, jak je od siebie odróżnisz.

Wyzwanie dla chętnych

PONAD PROGRAM

Napisz program, który pyta o liczbę sekund (na przykład 3725) i wypisuje, ile to godzin, minut i sekund. Podpowiedź: operator `//` dzieli bez reszty, a operator `%` daje resztę z dzielenia. Sekundy to liczba całkowita, więc użyj `int`.

Jak sprawdzisz, że Ci wyszło: Dla 3725 program ma wypisać 1 godzinę, 2 minuty i 5 sekund. Dla 59 ma wypisać 0 godzin, 0 minut i 59 sekund. Dla 3600 ma wypisać 1 godzinę, 0 minut i 0 sekund. Jeśli wszystkie trzy się zgadzają, wyszło.

Zadanie do domu na trzy poziomy

Wybierz jeden poziom. Możesz zacząć od łatwego i wrócić po trudniejszy.

POZIOM: ŁATWE

Napisz program, który pyta o Twoje imię i ulubioną liczbę, a potem wypisuje jedno zdanie z obiema wartościami, na przykład: Ola lubi liczbę 7. Liczbę pobierz funkcją `int` i sprawdź, że program działa także dla liczby 0.

POZIOM: ŚREDNIE

Napisz kalkulator kosztu wyjścia klasowego: cena biletu (`float`), liczba uczniów (`int`), koszt przejazdu (`float`), a na końcu koszt całkowity w jednej czytelnej linii. Sprawdź go dla trzech zestawów danych, w tym dla liczby uczniów równej 0, i zapisz w zeszycie, czy wyniki mają sens.

POZIOM: TRUDNE

Rozbuduj kalkulator o liczbę opiekunów i koszt na jednego ucznia zaokrąglony funkcją `round` do dwóch miejsc. Następnie celowo wpisz zero uczniów, przeczytaj komunikat `ZeroDivisionError` i zapisz w komentarzu w kodzie, w której linii i dlaczego program się zatrzymał.

Samoocena

Zaznacz, co już potrafisz. To tylko dla Ciebie, nikt tego nie ocenia.

- ☐ Potrafię utworzyć zmienną z jednym znakiem równości i nadać jej nazwę, która mówi, co jest w środku.

- ☐ Potrafię odróżnić tekst w cudzysłowie od liczby i wybrać int albo float w zależności od tego, czy wartość ma część dziesiętną.
- ☐ Potrafię pobrać dane funkcją input, zamienić je na liczbę funkcją int albo float i wypisać wynik funkcją print.
- ☐ Potrafię przeczytać komunikat błędu, znaleźć numer linii i wyjaśnić, dlaczego program wypisał 78 zamiast 15.

KARTKA DLA RODZICA

Python od zera, zmienne i dane od użytkownika

Szanowni Rodzice, dziecko miało dziś lekcję z Cyfrowej Pracowni. Poniżej w dwóch zdaniach, czego się nauczyło, i jedno pytanie, które możecie zadać przy kolacji.

Czego dziecko nauczyło się na tej lekcji

Wasze dziecko napisało dziś pierwszy program w języku Python: pyta użytkownika o kilka liczb, zapisuje je w zmiennych i wypisuje policzony wynik, na przykład koszt wyjścia klasowego. Potrafi też przeczytać komunikat błędu i wskazać, w której linii programu coś nie działa.

Pytanie na kolację

Zapytajcie: dlaczego komputer, który dostał liczby 7 i 8, może wypisać 78 zamiast 15? Jeśli dziecko odpowie, że komputer potraktował je jak tekst i skleił, lekcja zadziałała.

Uwaga o bezpieczeństwie

Jeśli dziecko chce ćwiczyć w domu, wystarczy darmowy edytor Thonny albo edytor Pythona w przeglądarce bez zakładania konta; nie potrzeba żadnych płatnych aplikacji ani podawania danych osobowych.

Zadanie domowe do wyboru

Dziecko wybiera jeden poziom. Każdy jest do zrobienia bez pomocy nauczyciela.

POZIOM: ŁATWE

Napisz program, który pyta o Twoje imię i ulubioną liczbę, a potem wypisuje jedno zdanie z obiema wartościami, na przykład: Ola lubi liczbę 7. Liczbę pobierz funkcją `int` i sprawdź, że program działa także dla liczby 0.

POZIOM: ŚREDNIE

Napisz kalkulator kosztu wyjścia klasowego: cena biletu (float), liczba uczniów (int), koszt przejazdu (float), a na końcu koszt całkowity w jednej czytelnej linii. Sprawdź go dla trzech zestawów danych, w tym dla liczby uczniów równej 0, i zapisz w zeszycie, czy wyniki mają sens.

POZIOM: TRUDNE

Rozbuduj kalkulator o liczbę opiekunów i koszt na jednego ucznia zaokrąglony funkcją round do dwóch miejsc. Następnie celowo wpisz zero uczniów, przeczytaj komunikat ZeroDivisionError i zapisz w komentarzu w kodzie, w której linii i dlaczego program się zatrzymał.

QUIZ DLA UCZNIA

Python od zera, zmienne i dane od użytkownika

Zaznacz jedną odpowiedź w każdym pytaniu. Wynik nigdzie się nie zapisuje.

1. Co robi w Pythonie zapis `cena = 12`?

1 Z 10, POZIOM PODSTAWOWY

- ☐ A. Zapisuje liczbę 12 do zmiennej o nazwie `cena`
- ☐ B. Sprawdza, czy `cena` jest równa 12
- ☐ C. Wypisuje na ekranie liczbę 12

2. Która nazwa zmiennej jest poprawna w Pythonie?

2 Z 10, POZIOM PODSTAWOWY

- ☐ A. 2liczba
- ☐ B. `cena biletu`
- ☐ C. `cena_biletu`

3. Jakiego typu wartość oddaje funkcja `input`?

3 Z 10, POZIOM PODSTAWOWY

- ☐ A. Zawsze tekst
- ☐ B. Zawsze liczbę całkowitą
- ☐ C. Liczbę, jeśli użytkownik wpisał cyfry

4. Użytkownik wpisał 7 i 8 w programie, w którym nie użyto funkcji `int`. Co wypisze dodawanie tych wartości?

4 Z 10, POZIOM PODSTAWOWY

- ☐ A. 15
- ☐ B. 78
- ☐ C. Komunikat o błędzie składni

5. Cena biletu wynosi 12 złotych i 50 groszy. Której funkcji użyjesz przy pobieraniu jej od użytkownika?

5 Z 10, POZIOM PODSTAWOWY

- ☐ A. `int`
- ☐ B. `print`
- ☐ C. `float`

6. Co wypisze ten program?

```
punkty = 10
punkty = punkty + 5
print(punkty)
```

6 Z 10, POZIOM ROZSZERZONY

- ☐ A. 10
- ☐ B. 15
- ☐ C. punkty + 5
- ☐ D. Błąd, bo zmienna nie może stać po obu stronach znaku równości

7. Program zapisuje wzrost ucznia w metrach, na przykład 1.62, oraz liczbę książek w jego plecaku. Jakie typy danych pasują do tych dwóch zmiennych?**7 Z 10, POZIOM ROZSZERZONY**

- ☐ A. int i int
- ☐ B. tekst i int
- ☐ C. float i float
- ☐ D. float i int

8. Użytkownik wpisuje 13. Co się stanie po uruchomieniu tego programu?

```
wiek = input("Ile masz lat? ")
print(wiek + 1)
```

8 Z 10, POZIOM ROZSZERZONY

- ☐ A. Wypisze 14
- ☐ B. Wypisze 131
- ☐ C. Zatrzyma się z błędem TypeError w linii 2
- ☐ D. Zatrzyma się z błędem SyntaxError w linii 1

9. Użytkownik wpisuje 3, a potem 4. Co wypisze ten program?

```
a = input("Pierwsza: ")
b = int(input("Druga: "))
print(a * b)
```

9 Z 10, POZIOM ROZSZERZONY

- ☐ A. 12
- ☐ B. 3333
- ☐ C. 34
- ☐ D. Błąd TypeError

10. Program ma pobrać cenę biletu, na przykład 4.50, i policzyć koszt dla 3 osób. Która linia poprawnie pobiera cenę?

10 Z 10, POZIOM ROZSZERZONY

- ☐ **A.** `cena = float(input("Cena: "))`
- ☐ **B.** `cena = int(input("Cena: "))`
- ☐ **C.** `cena = input("Cena: ")`
- ☐ **D.** `cena = print(input("Cena: "))`

QUIZ Z ODPOWIEDZIAMI

Python od zera, zmienne i dane od użytkownika

Wersja dla nauczyciela. Poprawna odpowiedź jest zaznaczona, pod każdym pytaniem jest wyjaśnienie.

1. Co robi w Pythonie zapis `cena = 12`?

1 Z 10, POZIOM PODSTAWOWY

- ☒ **A. Zapisuje liczbę 12 do zmiennej o nazwie `cena`**
- ☐ B. Sprawdza, czy `cena` jest równa 12
- ☐ C. Wypisuje na ekranie liczbę 12

Poprawna odpowiedź: A.

Jeden znak równości to przypisanie, czyli zapisanie wartości 12 do zmiennej `cena`. Porównanie robi się dwoma znakami równości, a wypisanie na ekran wymaga funkcji `print`, której tu nie ma.

Umiejętność: Zmienne i typy danych

2. Która nazwa zmiennej jest poprawna w Pythonie?

2 Z 10, POZIOM PODSTAWOWY

- ☐ A. 2liczba
- ☐ B. cena biletu
- ☒ **C. `cena_biletu`**

Poprawna odpowiedź: C.

Nazwa nie może zaczynać się od cyfry (2liczba) ani zawierać spacji (cena biletu), więc poprawne jest `cena_biletu`, gdzie słowa łączy podkreślenie.

Umiejętność: Zmienne i typy danych

3. Jakiego typu wartość oddaje funkcja `input`?

3 Z 10, POZIOM PODSTAWOWY

- ☒ **A. Zawsze tekst**
- ☐ B. Zawsze liczbę całkowitą
- ☐ C. Liczbę, jeśli użytkownik wpisał cyfry

Poprawna odpowiedź: A.

Input zawsze oddaje tekst, nawet gdy użytkownik wpisał same cyfry. Nie oddaje liczby całkowitej i nie zgaduje typu po wpisanych znakach, dlatego przed liczeniem trzeba użyć `int` albo `float`.

Umiejętność: Dane wejściowe i wynik

4. Użytkownik wpisał 7 i 8 w programie, w którym nie użyto funkcji `int`. Co wypisze dodawanie tych wartości?

4 Z 10, POZIOM PODSTAWOWY

- ☐ A. 15
- ☒ B. 78
- ☐ C. Komunikat o błędzie składni

Poprawna odpowiedź: B.

Bez `int` obie wartości są tekstami, a dodawanie tekstów skleja je w 78. Wynik 15 wymagałby liczb, a błąd składni nie pojawi się, bo zapis jest poprawny, tylko typ nie pasuje do zamiaru.

Umiejętność: Dane wejściowe i wynik

5. Cena biletu wynosi 12 złotych i 50 groszy. Której funkcji użyjesz przy pobieraniu jej od użytkownika?

5 Z 10, POZIOM PODSTAWOWY

- ☐ A. `int`
- ☐ B. `print`
- ☒ C. `float`

Poprawna odpowiedź: C.

Liczba z częścią dziesiętną to `float`. `int` przyjmuje tylko liczby całkowite i przy wpisaniu 12.50 zatrzyma program błędem `ValueError`, a `print` niczego nie zamienia, tylko wypisuje.

Umiejętność: Zmienne i typy danych

6. Co wypisze ten program?

```
punkty = 10
punkty = punkty + 5
print(punkty)
```

6 Z 10, POZIOM ROZSZERZONY

- ☐ A. 10
- ☒ B. 15

- ☐ C. punkty + 5
- ☐ D. Błąd, bo zmienna nie może stać po obu stronach znaku równości

Poprawna odpowiedź: B.

Python najpierw liczy prawą stronę, czyli 10 plus 5, a potem zapisuje 15 do zmiennej punkty, więc stara wartość 10 znika. Zmienna może stać po obu stronach, bo to przypisanie, nie równanie. Napis punkty + 5 wypisałby się tylko wtedy, gdyby stał w cudzysłowie.

Umiejętność: Zmienne i typy danych

7. Program zapisuje wzrost ucznia w metrach, na przykład 1.62, oraz liczbę książek w jego plecaku. Jakie typy danych pasują do tych dwóch zmiennych?

7 Z 10, POZIOM ROZSZERZONY

- ☐ A. int i int
- ☐ B. tekst i int
- ☐ C. float i float
- ☒ D. float i int

Poprawna odpowiedź: D.

Wzrost ma część dziesiętną, więc potrzebuje float, a liczba książek jest zawsze całkowita, więc int. Dwa int obcięłyby wzrost do 1. Tekst uniemożliwiłby liczenie średniego wzrostu klasy. Dwa float zadziałają, ale pół książki nie istnieje, więc int lepiej opisuje tę wartość.

Umiejętność: Zmienne i typy danych

8. Użytkownik wpisuje 13. Co się stanie po uruchomieniu tego programu?
wiek = input("Ile masz lat? ")
print(wiek + 1)

8 Z 10, POZIOM ROZSZERZONY

- ☐ A. Wypisze 14
- ☐ B. Wypisze 131
- ☒ C. Zatrzyma się z błędem `TypeError` w linii 2
- ☐ D. Zatrzyma się z błędem `SyntaxError` w linii 1

Poprawna odpowiedź: C.

Zmienna `wiek` trzyma tekst `13`, a tekstu nie da się dodać do liczby `1`, więc Python zgłasza `TypeError` w linii z `print`. Wynik `14` wymagałby `int` wokół `input`. Sklejenie do `131` nastąpiłoby tylko przy dodaniu tekstu do tekstu, czyli `wiek + "1"`. Zapis obu linii jest poprawny, więc `SyntaxError` nie wystąpi.

Umiejętność: Dane wejściowe i wynik

9. Użytkownik wpisuje 3, a potem 4. Co wypisze ten program?

```
a = input("Pierwsza: ")  
b = int(input("Druga: "))  
print(a * b)
```

9 Z 10, POZIOM ROZSZERZONY

- ☐ A. 12
- ☒ B. 3333
- ☐ C. 34
- ☐ D. Błąd `TypeError`

Poprawna odpowiedź: B.

Zmienna `a` to tekst `3` (brak `int`), a `b` to liczba `4`. Tekst pomnożony przez liczbę całkowitą Python powtarza, więc powstaje `3333`. Wynik `12` wymagałby `int` także przy `a`. Sklejenie w `34` dałoby dodawanie dwóch tekstów, nie mnożenie. `TypeError` pojawia się dopiero przy mnożeniu tekstu przez tekst albo przez `float`.

Umiejętność: Dane wejściowe i wynik

10. Program ma pobrać cenę biletu, na przykład 4.50, i policzyć koszt dla 3 osób. Która linia poprawnie pobiera cenę?

10 Z 10, POZIOM ROZSZERZONY

- ☒ A. `cena = float(input("Cena: "))`
- ☐ B. `cena = int(input("Cena: "))`
- ☐ C. `cena = input("Cena: ")`
- ☐ D. `cena = print(input("Cena: "))`

Poprawna odpowiedź: A.

Cena z groszami wymaga float, więc poprawna jest linia z float wokół input. Int przy wpisanym 4.50 zatrzyma program błędem ValueError. Sam input zostawi tekst i mnożenie przez 3 da 4.504.504.50 zamiast 13.5. Print tylko wypisuje i nie oddaje wartości, więc w zmiennej cena nie byłoby niczego użytecznego.

Umiejętność: Dane wejściowe i wynik

SKRYPT WIDEO DLA NAUCZYCIELA

Python od zera, zmienne i dane od użytkownika

Nagranie powstaje na podstawie tego skryptu. Do czasu publikacji wideo możecie przeczytać skrypt przed lekcją.

0:00 DO 0:40

Witajcie. To lekcja pierwsza z Pythona dla klas siódmej i ósmej. W 45 minut każdy uczeń napisze program, który pyta o dane i liczy wynik. Nie potrzebujecie wcześniejszego doświadczenia w programowaniu, wystarczy laptop z przeglądarką albo zainstalowane Thonny.

0:40 DO 1:40

Zanim wejdziecie do sali, sprawdźcie jedną rzecz: otwórzcie edytor na swoim laptopie i uruchomcie program z jedną linią print. Jeśli szkoła ma zainstalowane Thonny, użyjcie go, bo działa bez internetu i bez logowania. Jeśli pracujecie w przeglądarce, wybierzcie edytor, który nie wymaga konta od ucznia, i zapiszcie adres na tablicy.

1:40 DO 2:50

Pierwsze pięć minut to wejście. Pokazujecie trzy linijki kodu i pytacie klasę, co się wypisze. To ważne, bo uczniowie od razu wchodzą w rolę czytającego kod, a nie tylko przepisującego. Nie tłumaczcie jeszcze niczego, zbierzcie odpowiedzi i przejdźcie dalej.

2:50 DO 4:00

Pokaz trwa dziesięć minut i piszecie kod na żywo, linia po linii, uruchamiając po każdej zmianie. Kluczowy moment to input. Powiedzcie uczniom wprost i głośno: input zawsze oddaje tekst. Potem popełnijcie błąd celowo, pokażcie komunikat i naprawcie go funkcją int. Uczniowie zapamiętają naprawę lepiej niż gotowy kod.

4:00 DO 5:10

Dwadzieścia minut pracy własnej. Chodźcie po sali i nie dotykajcie klawiatury ucznia. Zadajecie dwa pytania: co pokazał komunikat błędu i którą linię uczeń uruchomił jako ostatnią. Uczeń, który sam znajdzie literówkę, zapamiętuje ją na stałe. Kto skończy wcześniej, dostaje wariant trudniejszy z zaokrągleniem kosztu na osobę.

5:10 DO 6:10

Ostatnie dziesięć minut to sprawdzenie i podsumowanie. Dwoje uczniów pokazuje program na projektorze, a klasa podaje dane wejściowe, także zero, bo to najczęstszy przypadek brzegowy. Na koniec każdy zapisuje w zeszycie jedno zdanie o tym, co robi input i po co jest int.

6:10 DO 6:50

Jeśli laptopy nie działają, macie wariant bez sprzętu z karteczkami jako zmiennymi. Jeśli w klasie jest uczeń z trudnościami, dajcie mu program z trzema lukami zamiast pisanie od zera. Na następnej lekcji te same zmienne posłużą do zbudowania gry tekstowej. Do zobaczenia.