LEKCJA 1 • 45 MIN

Logika boolowska w Pythonie

Imię i nazwisko: ..... Klasa: ..... Data: .....

Cel zajęć: poznasz wartości **True** i **False** oraz użyjesz operatorów **and**, **or** i **not** do tworzenia warunków w programach.

Plan lekcji 1

Czas	Co robimy
0–5 min	Rozgrzewka: kiedy warunek jest prawdziwy?
5–15 min	Teoria i przykłady: True / False, and, or, not
15–25 min	Ćwiczenia 1–2: przewidywanie wyników i tabele prawdy
25–37 min	Ćwiczenie 3: warunek z dwiema przesłankami
37–43 min	Ćwiczenie 4: zakres liczb i poprawa warunku
43–45 min	Bilet wyjścia: jedno pytanie podsumowujące

1. Co oznacza wartość logiczna?

W Pythonie wynik warunku może być **True** (prawda) albo **False** (fałsz). Zapisuj je wielką literą. Porównania >, <, >=, <=, == i != mogą zwracać takie wartości.

```
print(7 > 3)       # True
print(4 == 9)      # False
print(8 != 5)      # True
print(type(True))  # <class 'bool'>
```

2. Trzy najważniejsze operatory

Operator	Działanie	Przykład	Wynik
and	Prawda, gdy **oba** warunki są prawdziwe	`True and False`	False
or	Prawda, gdy **co najmniej jeden** jest prawdziwy	`True or False`	True
not	Odwraca wartość logiczną	`not True`	False

Ważne: w wyrażeniach logicznych obowiązuje kolejność `not` → and → `or`. Nawiasy pomagają uniknąć pomyłek. Operatory and i `or` mogą w Pythonie zwracać także inne wartości niż `bool`, ale w tych zadaniach łączymy warunki logiczne.

LEKCJA 1 • TEORIA I PRAKTYKA

Tabele prawdy i pierwsze ćwiczenia

Tabela prawdy

Przetestuj wszystkie cztery możliwe zestawy wartości dwóch warunków A i B.

A	B	A and B	A or B	not A
False	False	False	False	True
False	True	False	True	True
True	False	False	True	False
True	True	True	True	False

Ćwiczenie 1 • Przewidź wynik (5 min)

Bez uruchamiania programu wpisz **True** lub **False**. Następnie sprawdź odpowiedzi w interpreterze Pythona.

Wyrażenie	Mój wynik
`False or True and False`	.....
`not (5 > 3)`	.....
`7 >= 7 and 2 != 3`	.....
`not (False or False)`	.....
`(4 % 2 == 0) and not (9 < 5)`	.....

Ćwiczenie 2 • Uzupełnij tabelę (5 min)

Wpisz wyniki dla dwóch wyrażeń: `not A or B` oraz `not (A and B)`.

A	B	not A or B	not (A and B)
False	False	.....	.....
False	True	.....	.....
True	False	.....	.....
True	True	.....	.....

Sprawdź w Pythonie:

```
a = True
b = False
print(not a or b)
print(not (a and b))
```

Uwaga na typowy błąd: = służy do przypisania wartości, natomiast == do sprawdzenia równości.

LEKCJA 1 • ĆWICZENIA

Instrukcje warunkowe: and / or / not

Ćwiczenie 3 • Zgoda na udział (12 min)

Uczeń może wziąć udział w wydarzeniu, gdy ma **co najmniej 16 lat i posiada zgodę**. Uruchom program, uzupełnij warunek i przetestuj trzy zestawy danych.

```
wiek = int(input("Podaj wiek: "))
zgoda = input("Czy jest zgoda? (tak/nie): ").strip().lower() == "tak"

if _____________________________:
    print("Możesz wziąć udział")
else:
    print("Brak możliwości udziału")
```

Próba	Wiek	Zgoda	Oczekiwany komunikat
1	16	tak	Możesz wziąć udział
2	15	tak	Brak możliwości udziału
3	18	nie	Brak możliwości udziału

Pytanie: Co by się zmieniło, gdybyś użył(a) operatora `or` zamiast `and`?

Ćwiczenie 4 • Sprawdzanie zakresu (6 min)

Napisz program, który wczyta liczbę całkowitą i wyświetli **W zakresie**, gdy liczba należy do przedziału **od 1 do 10 włącznie**; w przeciwnym razie wyświetli **Poza zakresem**.

Użyj warunku z operatorem `and`; następnie sprawdź, czy zapis `1 <= liczba <= 10` daje te same wyniki. Przetestuj: **0, 1, 10 i 11**.

Szybkie sprawdzenie • Bilet wyjścia (2 min)

W programie poniżej warunek jest błędny: wymagana jest **pełnoletność i zgoda**. Zmień dokładnie jeden operator.

```
if wiek >= 18 or zgoda:
    print("Można rozpocząć")
```

Poprawiony operator: .....

LEKCJA 2 • 45 MIN

Prawa de Morgana w Pythonie

Dziś upraszczasz zaprzeczone warunki i sprawdzasz ich równoważność programem.

Czas	Co robimy
0–5 min	Powtórzenie operatorów logicznych
5–17 min	Dwa prawa de Morgana: omówienie i przykład
17–27 min	Ćwiczenie 5: tabela prawdy obu praw
27–37 min	Ćwiczenie 6: przekształcanie warunków i weryfikacja
37–43 min	Ćwiczenie 7: miniprojekt – symulator kontroli dostępu
43–45 min	Podsumowanie i samoocena

Co mówią prawa de Morgana?

Prawo 1: zaprzeczenie „A **i** B” oznacza „nie A **lub** nie B”.
`not (A and B) == (not A) or (not B)`

Prawo 2: zaprzeczenie „A **lub** B” oznacza „nie A **i** nie B”.
`not (A or B) == (not A) and (not B)`

Zapamiętaj trzy kroki

1. Zaneguj każdy warunek. **2.** Zmień `and` na `or` (lub odwrotnie). **3.** Zachowaj sens porównań i nawiasów.

Przykład praktyczny

Uczeń **nie** spełnia wszystkich wymagań, gdy nie ma legitymacji **lub** nie ma zgody. Obydwie wersje poniżej dają dokładnie ten sam wynik.

```
legitymacja = True
zgoda = False

wersja_1 = not (legitymacja and zgoda)
wersja_2 = (not legitymacja) or (not zgoda)
print(wersja_1, wersja_2)  # True True
```

Nieprawidłowo: `not (A and B)` to **nie** jest `(not A) and (not B)`. Zaprzeczając nawias, **zmieniasz też operator**.

LEKCJA 2 • SPRAWDZAMY PRAWA

Od tabeli prawdy do kodu

Ćwiczenie 5 • Dwie równoważności (10 min)

Uzupełnij tabelę. Porównaj kolumny 3 z 4 oraz 5 z 6. Powinny być identyczne w każdym wierszu.

A	B	not (A and B)	not A or not B	not (A or B)	not A and not B
F	F	.....	.....	.....	.....
F	T	.....	.....	.....	.....
T	F	.....	.....	.....	.....
T	T	.....	.....	.....	.....

`F = False, T = True`. Przetestuj obydwa prawa za pomocą poniższego kodu.

```
for a in (False, True):
    for b in (False, True):
        p1 = (not (a and b)) == ((not a) or (not b))
        p2 = (not (a or b)) == ((not a) and (not b))
        print(a, b, p1, p2)
```

Oczekiwanie: obie ostatnie wartości w każdym wierszu powinny wynosić `True`. Nawiasy oddzielają sprawdzane wyrażenia od operatora porównania `==`.

Ćwiczenie 6 • Przekształć warunki (10 min)

Usuń zewnętrzne zaprzeczenie `not (...)`, stosując prawa de Morgana i zaprzeczenia porównań. Zapisz równoważny warunek.

Warunek początkowy	Postać równoważna – uzupełnij
`not (x >= 0 and x <= 10)`	.....
`not (admin or nauczyciel)`	.....
`not (plik_ok and zgoda)`	.....
`not (temp < 0 or wilgotnosc > 90)`	.....

Wybierz **dwa** wiersze. Nadaj zmiennym przykładowe wartości i sprawdź w Pythonie, czy obie postacie zwracają ten sam wynik.

> **Podpowiedź:** zaprzeczenie `x >= 0` to `x < 0`, a zaprzeczenie `temp < 0` to `temp >= 0`.

LEKCJA 2 • ZADANIE PODSUMOWUJĄCE

Miniprojekt i samoocena

Ćwiczenie 7 • Symulator kontroli dostępu (6 min)

Zbuduj krótki symulator. Osoba otrzymuje dostęp tylko wtedy, gdy **oba warunki** są spełnione: poprawne dane użytkownika i poprawny kod jednorazowy. **Nie wpisuj prawdziwych haseł ani kodów.**

Wymagania: pobierz dwie odpowiedzi `tak/nie`, zamień je na wartości logiczne; utwórz zmienną `odmowa` z użyciem `not (... and ...)`; utwórz `odmowa_dm` z prawem de Morgana; wyświetl komunikat **Dostęp przyznany** albo **Odmowa dostępu** i sprawdź zgodność obu warunków.

```
dane_ok = input("Dane poprawne? (tak/nie): ").strip().lower() == "tak"
kod_ok = input("Kod poprawny? (tak/nie): ").strip().lower() == "tak"

odmowa = ________________________________________
odmowa_dm = ________________________________________

print("Warunki zgodne:", odmowa == odmowa_dm)
if odmowa:
    print("Odmowa dostępu")
else:
    print("Dostęp przyznany")
```

Przetestuj wszystkie cztery kombinacje:

Dane poprawne?	Kod poprawny?	Oczekiwany wynik
nie	nie	Odmowa dostępu
nie	tak	Odmowa dostępu
tak	nie	Odmowa dostępu
tak	tak	Dostęp przyznany

Dla chętnych • Rozszerzenie

Dodaj trzeci warunek `konto_aktywne`. Zapisz w dwóch postaciach warunek odmowy, gdy **nie wszystkie trzy** przesłanki są spełnione.

Podsumowanie (2 min)

Zaznacz, co już potrafisz: ☐ obliczyć `and`/`or`/`not` ☐ uzupełnić tabelę prawdy ☐ zastosować prawa de Morgana ☐ sprawdzić warunek w Pythonie.

Jedno zdanie, które warto zapamiętać: gdy zaprzeczam całemu wyrażeniu, zmieniam operator logiczny i zaprzeczam częściom.