

Każdy Klub Kodowania musi być zarejestrowany. Zarejestrowane kluby można zobaczyć na mapie na stronie [codeclubworld.org](https://codeclubworld.org) - jeżeli nie ma tam twojego klubu sprawdź na stronie [jumpto.cc/18CpLPy](https://jumpto.cc/18CpLPy) (ang.) co trzeba zrobić, by to zmienić.

## Wstęp:

W tym projekcie stworzysz quiz, którym możesz rzucić wyzwanie swoim przyjaciołom.



**Zadania do wykonania**

Wykonaj te **POLECENIA** krok po kroku



**Przetestuj swój projekt**

Kliknij na zieloną flagę, aby  
**PRZETESTOWAĆ** swój kod



**Zapisz swój projekt**

Teraz **ZAPISZ** swój projekt

# Krok 1: Zadawanie pytania



## Lista zadań

- Zaczniemy od napisania prostego quizu, który zadaje graczowi pytanie, a potem wyświetla uśmiechnięte buźki jeśli wprowadzi on prawidłową odpowiedź.



```
print("Jak w Pythonie nazywa się 'pudełko', w którym  
trzymamy dane?")  
odpowiedz = input() # input po Polsku znaczy dane  
wejściowe  
  
if odpowiedz == "zmienna":  
    print(" :) " * 100)  
  
print("Dzięki za grę!")
```

Pamiętaj o dodaniu dwukropka ( :) na końcu linii `if odpowiedz == "zmienna":` i wcięciu linii poniżej (przesunięciu w prawo) spacjami.

- Po napisaniu powyższego programu wypróbuj go! Co się stanie jeśli odpowiesz prawidłowo? Co się stanie jeśli odpowiedź będzie błędna?



The top screenshot shows the code being executed with the input "nie wiem". The output is "Dzięki za grę!".

The bottom screenshot shows the code being executed with the input "zmienna". The output is 100 smiley faces " :) " followed by "Dzięki za grę!".

Kod po wcięciu (który wyświetla uśmiechnięte buźki) wykonuje

się tylko jeśli (`if`) odpowiedź jest poprawna. Z kolei “Dzięki za grę!” pojawia się zawsze, bez względu na to, czy odpowiedź jest poprawna czy błędna. Czy potrafisz opowiedzieć dlaczego?

Python używa podwójnego znaku równa się `==` do sprawdzenia, czy dwie rzeczy są takie same. Musi używać podwójnego znaku, ponieważ pojedynczy `=` jest używany do zapisania czegoś w zmiennej (na przykład `odpowiedz = input()`).

- Program powyżej wyświetla uśmiechnięte buźki jeśli gracz odpowie prawidłowo, ale nie wyświetla nic kiedy odpowiedział źle. Aby to poprawić możesz użyć komendy `else` (po polsku: w przeciwnym wypadku) żeby wyświetlić smutne buźki jeśli użytkownik wprowadzi cokolwiek co jest inne od prawidłowej odpowiedzi.

```
print("Jak w pythonie nazywa się 'pudełko', w którym  
trzymamy dane?")  
odpowiedz = input()  
  
if odpowiedz == "zmienna":  
    print(" :) " * 100)  
else:  
    print(" :( " * 100)  
  
print("Dzięki za grę!")
```

Wypróbuj nowy program. Co się stanie jeśli wprowadzisz poprawną odpowiedź? Co się stanie jeśli wprowadzisz odpowiedź błędną?

[illegible]

## Wyzwanie: Czas na pytania

Korzystając z tego, czego już się nauczyłeś stwórz swój quiz. Możesz wybrać jakikolwiek temat, a twój quiz powinien używać komend `if` i `else` tak, żeby informować gracza o tym jak mu idzie.



## Krok 2: Testowanie

Zawsze dobrze jest sprawdzić swoje programy, aby upewnić się, że działają jak należy.



- Jeśli sprawdzałeś swój program, napewno zauważyłeś, że możliwe jest otrzymanie smutnych buziek nawet jeśli została wprowadzona prawidłowa odpowiedź! Tak jak w tym przykładzie, w którym gracz przypadkowo wcisnął CAPS LOCK!



The top screenshot shows a Trinket Python IDE interface. On the left, a code editor displays a Python script in a file named `main.py`. The script asks the user for a variable name and checks if it is `"zmienna"`. If yes, it prints `" * 100"`; otherwise, it prints `" * 100"` (likely a typo for `" * 100"`). The right panel shows the output in a terminal window, displaying the prompt and the result `" * 100"`.

The bottom screenshot shows the same Trinket IDE interface. The code editor now has a variable `zmienna` defined, and the terminal window shows the output `" * 100"`.

## Zapisz Swój Projekt

## Krok 3: Wielokrotny wybór

### ✓ Lista Zadań

- Do tej pory używaliśmy `if` i `else` żeby dać graczowi znać, czy jego odpowiedź była prawidłowa czy nie. Co jeśli chcielibyśmy zadać pytanie z wieloma odpowiedziami, w którym gracz mógłby wybrać jedną z 4 odpowiedzi? Służy do tego komenda `elif`.

```
print('''
P1 - "Jak w Pythonie nazywa się 'pudełko', w którym
trzymamy dane?
a - tekst
b - zmienna
c - pudełko na buty
''')
odpowiedz = input().lower()

if odpowiedz == "a":
```

```
print(" Niestety - tekst to typ danych :( ")
elif odpowiedz == "b":
    print(" Zgadza sie!! :) ")
elif odpowiedz == "c":
    print(" Chyba się wygłupiasz... :( ")
else:
    print(" Nie wybrałeś/-aś ani a, ani b, ani c :( ")
```

`elif` to skrót od angielskiego “else if”, które po polsku oznacza “w przeciwnym wypadku jeżeli”. Dlatego w programie powyżej gracz zobaczy jedną z 4 wiadomości w zależności od tego jaką dał odpowiedź.

- Dodaj powyższy kod do swojego quizu tak, aby zawierał pytanie wielokrotnego wyboru. ☐
- Sprawdź to nowe pytanie na 4 sposoby tak, żebyś otrzymał każdą z 4 odpowiedzi. ☐

main.py

```

1 print(''''
2 P1 - "Jak w Pythonie nazywa się 'pudełko', w którym trzymamy dane?
3 a - tekst
4 b - zmienna
5 c - pudełko na buty
6 ''')
7 odpowiedz = input().lower()
8
9 if odpowiedz == "a":
10     print("Niestety - tekst to typ danych :( ")
11 elif odpowiedz == "b":
12     print("Zgadza się!! :) ")
13 elif odpowiedz == "c":
14     print("Chyba się wygłupiasz... :( ")
15 else:
16     print("Nie wybrałeś/-aś ani a, ani b, ani c :( ")
17

```

Powered by trinket

```

P1 - "Jak w Pythonie nazywa się 'pudełko',
w którym trzymamy dane?
a - tekst
b - zmienna
c - pudełko na buty

a
Niestety - tekst to typ danych :(

```

main.py

```

1 print(''''
2 P1 - "Jak w Pythonie nazywa się 'pudełko', w którym trzymamy dane?
3 a - tekst
4 b - zmienna
5 c - pudełko na buty
6 ''')
7 odpowiedz = input().lower()
8
9 if odpowiedz == "a":
10     print("Niestety - tekst to typ danych :( ")
11 elif odpowiedz == "b":
12     print("Zgadza się!! :) ")
13 elif odpowiedz == "c":
14     print("Chyba się wygłupiasz... :( ")
15 else:
16     print("Nie wybrałeś/-aś ani a, ani b, ani c :( ")
17

```

Powered by trinket

```

P1 - "Jak w Pythonie nazywa się 'pudełko',
w którym trzymamy dane?
a - tekst
b - zmienna
c - pudełko na buty

b
Zgadza się!! :)

```

main.py

```

1 print(''''
2 P1 - "Jak w Pythonie nazywa się 'pudełko', w którym trzymamy dane?
3 a - tekst
4 b - zmienna
5 c - pudełko na buty
6 ''')
7 odpowiedz = input().lower()
8
9 if odpowiedz == "a":
10     print("Niestety - tekst to typ danych :( ")
11 elif odpowiedz == "b":
12     print("Zgadza się!! :) ")
13 elif odpowiedz == "c":
14     print("Chyba się wygłupiasz... :( ")
15 else:
16     print("Nie wybrałeś/-aś ani a, ani b, ani c :( ")
17

```

Powered by trinket

```

P1 - "Jak w Pythonie nazywa się 'pudełko',
w którym trzymamy dane?
a - tekst
b - zmienna
c - pudełko na buty

c
Chyba się wygłupiasz... :(

```

main.py

```

1 print(''''
2 P1 - "Jak w Pythonie nazywa się 'pudełko', w którym trzymamy dane?
3 a - tekst
4 b - zmienna
5 c - pudełko na buty
6 ''')
7 odpowiedz = input().lower()
8
9 if odpowiedz == "a":
10     print("Niestety - tekst to typ danych :( ")
11 elif odpowiedz == "b":
12     print("Zgadza się!! :) ")
13 elif odpowiedz == "c":
14     print("Chyba się wygłupiasz... :( ")
15 else:
16     print("Nie wybrałeś/-aś ani a, ani b, ani c :( ")
17

```

Powered by trinket

```

P1 - "Jak w Pythonie nazywa się 'pudełko',
w którym trzymamy dane?
a - tekst
b - zmienna
c - pudełko na buty

nie wiem
Nie wybrałeś/-aś ani a, ani b, ani c :(

```



Zapisz Swój Projekt



## Wyzwanie: Quiz z pytaniami wielokrotnego wyboru

Dodaj kilka pytań wielokrotnego wyboru do swojego programu. Kiedy skończysz, poproś kogoś o udział w grze! Jak im poszło? Czy dobrze się bawili? Czy Twój quiz był za łatwy albo za trudny?



Zapisz Swój Projekt

## Wyzwanie: Liczenie punktów

Czy potrafisz wykorzystać zmienną `punkty` w swoim programie do liczenia punktów gracza? Oto jak taka zmienna może być użyta:

- ☐ Na początku programu, ustaw wartość zmiennej na 0.
- ☐ Za każdym razem kiedy zostaje udzielona prawidłowa odpowiedź dodaj 1 do punktów gracza ( `punkty = punkty + 1` )
- ☐ Na końcu programu wyświetl liczbę punktów gracza.



Zapisz Swój Projekt

## Wyzwanie: Jak mi poszło?

Czy potrafisz wyświetlić osobistą wiadomość dla gracza na końcu każdej gry?

- ☐ Powiedz “bardzo dobrze” jeśli ( `if` ) gracz odpowiedział prawidłowo na wszystkie pytania.
- ☐ W przeciwnym wypadku ( `else` ) powiedz “jeszcze raz” jeśli którakolwiek odpowiedź była błędna.

(Bedziesz potrzebował zmiennej `punkty` , żeby zdecydować, którą wiadomość wyświetlić!)



Zapisz Swój Projekt