

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

Wstęp

Python umożliwia przekształcenie listy poleceń w użyteczne programy i ciekawe gry! Wykonując ten projekt nauczysz się, jak uruchomić program napisany w języku Python a także jak za jego pomocą wyświetlić tekst na ekranie.



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: Powiedz "cześć"

✓ Lista zadań

- Zaczniemy od napisania bardzo prostego programu w języku Python, abyś wiedział(a) jak go uruchomić. Otwórz edytor: <https://trinket.io/python/7c0a7396c0>. Jeśli pracujesz online, możesz również posłużyć się wersją wyświetloną poniżej.

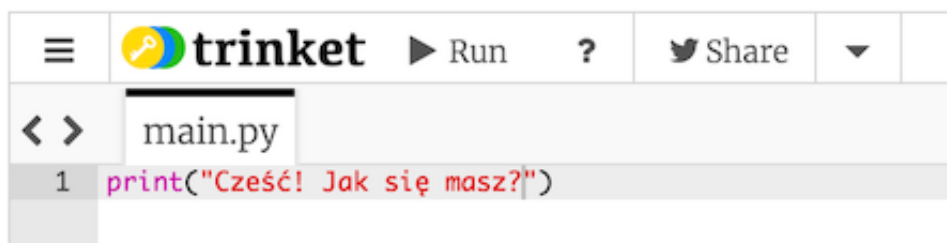


- W pliku `main.py` wpisz:

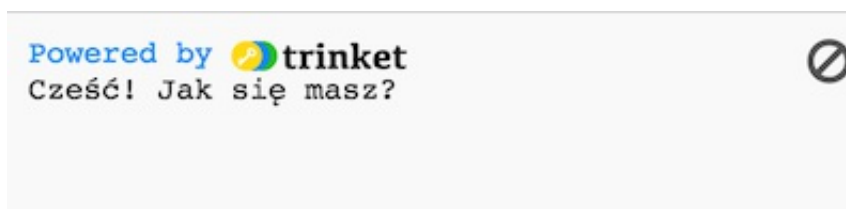


```
print("Cześć! Jak się masz?")
```

Ten program wyświetli na ekranie tekst. Zauważ, że tekst, który chcesz wyświetlić jest wpisany w cudzysłowie ("). Twój kod powinien wyglądać mniej więcej tak:

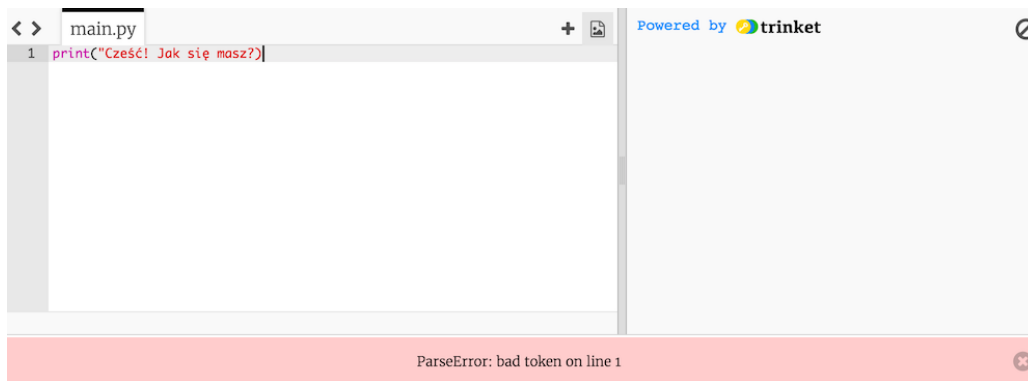


- Uruchom program klikając `Run`. Po prawej stronie zobaczysz, tekst wpisany w polecenie `print`.



- Jeśli zrobisz błąd, na przykład zapomnisz wpisać drugiego cudzysłowu ("), wtedy pod edytorem wyświetlony zostanie komunikat, który w języku angielskim mówi na czym polega błąd. Spróbuj!





```
main.py
1 print('Cześć! Jak się masz?')
```

Powered by trinket

ParseError: bad token on line 1

- Gratulacje, od teraz oficjalnie jesteś programistą języka Python! Możesz z dumą się tym pochwalić.



Zapisz swój projekt

Wyzwanie: Co masz na myśli?

Zmień powyższy program, aby wyświetlał na ekranie coś ciekawszego!



```
Powered by trinket
Jestem głodny!
```



Zapisz swój projekt

Krok 2: Coś o Tobie

✓ Lista zadań

- Wyświetlmy coś zabawniejszego niż zwykły tekst... ASCII art! ☐
ASCII art polega na tworzeniu obrazków z tekstu. Poniżej przykład - to miał być pies!

```
Obrazek przedstawia psa...  
0_____  
| | | |
```

Aby zrobić takie arcydzieło, możesz wpisać w edytorze poniższe polecenia i uruchomić program:

```
print("Obrazek przedstawia psa...")  
print(" 0____ ")  
print("  | | | | ")
```

- Jeśli wolisz, możesz użyć trzech pojedynczych apostrofów (`'''`) zamiast cudzysłowów. Pozwalają one wyświetlić za pomocą jednego polecenia tekst, który ma wiele linii. Tak jak tutaj: ☐

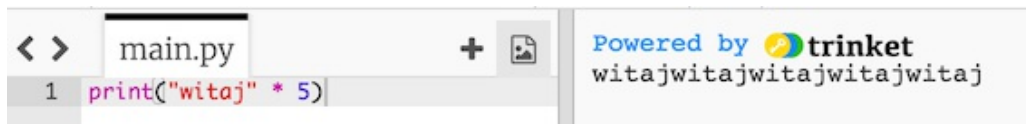
```
print('''  
Obrazek przedstawia psa...  
0_____  
| | | |  
''')
```

Kiedy uruchomisz ten program zobaczysz, że wyświetla takiego samego psa, jak poprzednio.



Zapisz swój projekt

Uruchom ten program, a zobaczysz odpowiedź:



```
main.py
1 print("witaj" * 5)
```

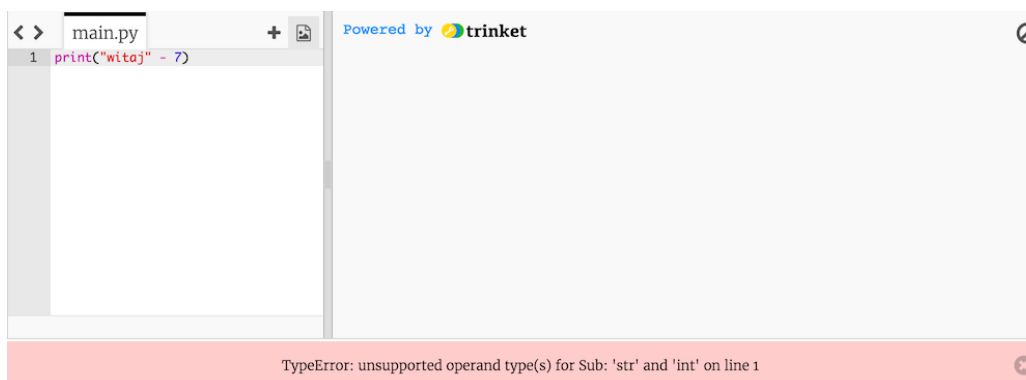
Powered by trinket
witajwitajwitajwitajwitaj

- Możesz zmienić wyświetlony tekst, aby był łatwiejszy do przeczytania. Wystarczy w programie dodać spację po słowie "witaj":

```
print("witaj " * 5)
```

Uruchom ten program, a zobaczysz, że odpowiedź jest nieco czytelniejsza niż poprzednio.

- Jeżeli "witaj " pomnożone przez 5 daje "witaj witaj witaj witaj witaj", to co się stanie z "witaj" - 7? Czy takie działanie ma jakiś sens?



```
main.py
1 print("witaj" - 7)
```

Powered by trinket

TypeError: unsupported operand type(s) for Sub: 'str' and 'int' on line 1

Oj, nie działa! Zamiast wyniku dostaliśmy komunikat błędu. Wygląda na to, że takie działanie nie ma sensu w Pythonie!

- A co z dodawaniem? Jak myślisz, jaki będzie wynik "witaj " + "świecie"? Spróbuj uruchamiając taki program:

```
print("witaj " + "świecie")
```



```
main.py
1 print('witaj ' + 'świecie')
```

Powered by trinket
witaj świecie

Czy takiego wyniku oczekiwałeś/oczekiwałaś?



Wyzwanie: Wyrazy i liczby

Co wyświetli na ekranie poniższy program? Zastanów się, a potem uruchom program i sprawdź czy zgadłeś/zgadłaś.

```
print("ha "*4)
print("ta"*2 + "rak")
print("He" + "l"*2 + "o" + "!"*10)
```

Potrawisz wymyślić swoje wyrazy?



Krok 4: Wzory ASCII

✓ Lista zadań

- Wiesz już, jak działają obliczenia na tekście, to co teraz? ☐
Dlaczego warto tego używać? Cóż, powiedzmy, że chcesz narysować prostokąt w ASCII art, który ma 30 znaków szerokości i 3 znaki wysokości. Możesz albo narysować go trudniejszym sposobem, tak jak tu:

```
print("#####")
print("#####")
print("#####")
```

Albo oszczędzić trochę czasu i narysować go prościej - jak tu:

```
print("#" * 30)
print("#" * 30)
print("#" * 30)
```

Oba wyświetlą na ekranie dokładnie ten sam prostokąt:

```
1 print("Trudniejszy sposób:")
2 print("#####")
3 print("#####")
4 print("#####")
5
6 print("Łatwiejszy sposób:")
7 print("#" * 30)
8 print("#" * 30)
9 print("#" * 30)
10
```

Powered by trinket

Trudniejszy sposób:

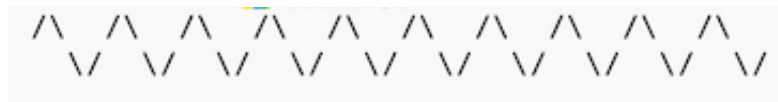
Łatwiejszy sposób:

#####

- Możesz nawet użyć działań do stworzenia ciekawych wzorów, takich jak ta fala:



```
print("/\ " * 10)
print(" \/" * 10)
```



Zapisz swój projekt

Wyzwanie: Nakoduj szalik

Twój najlepszy przyjaciel urządza przyjęcie z okazji 11. urodzin. Postanowiłeś/aś, że w prezencie nakodujesz szalik! Użyj działań, gdzie tylko możesz, aby stworzyć swój niepowtarzalny wzór szalika.

Jeśli masz gest, możesz nawet nakodować dla niego tort (razem z 11 świeczkami)!

```
Wszystkiego najlepszego!!!!!!!!!!!!!!  
To szalik dla Ciebie!  
  
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|-|  
  
A tutaj tort!  
  
      .       .       .       .       .       .       .       .       .       .  
    i   i   i   i   i   i   i   i   i   i   i  
#####  
=====
```



Zapisz swój projekt