

Polecenie PRINT

M@я3k Pүđ€£kØ

Programowanie w Pythonie

Polecenie print

- Polecenie `print` wyświetla zawarty w nawiasach i apostrofach tekst.

```
print ('Hello World')
```

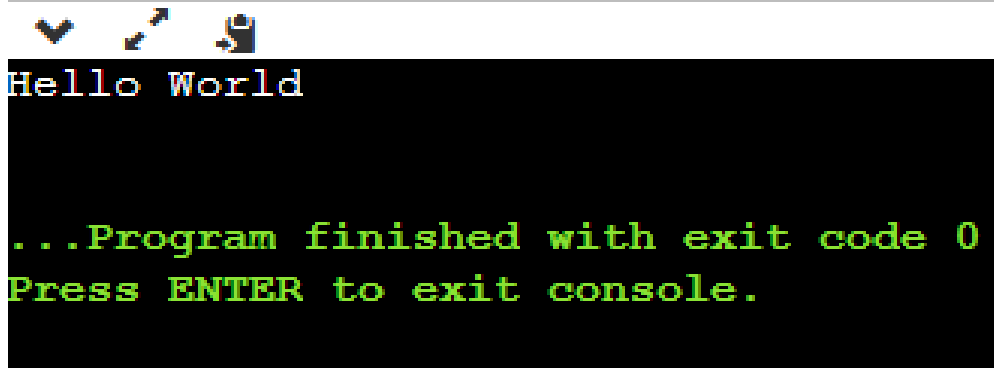
- Może też wyświetlać zmienne i inne obiekty, które są zapisane w nawiasach

```
print (x)
```

```
print (2)
```

Przykład wykonania polecenia

```
1  
2 print ('Hello World')  
3
```



Terminal window showing the execution of a Python program. The output is 'Hello World'. Below the output, it says '...Program finished with exit code 0' and 'Press ENTER to exit console.'

```
Hello World  
  
...Program finished with exit code 0  
Press ENTER to exit console.
```

- W górnym okienku mamy kod programu (na razie tylko jedna linijka).
- W dolnym są wyświetlone efekty działania programu

Sklejanie tekstu

- Polecenie `print` pozwala wyświetlić tekst rozbity na kilka części. Są one schowane w apostrofach i rozdzielone przecinkami

```
print ('Hello', ' World')
```

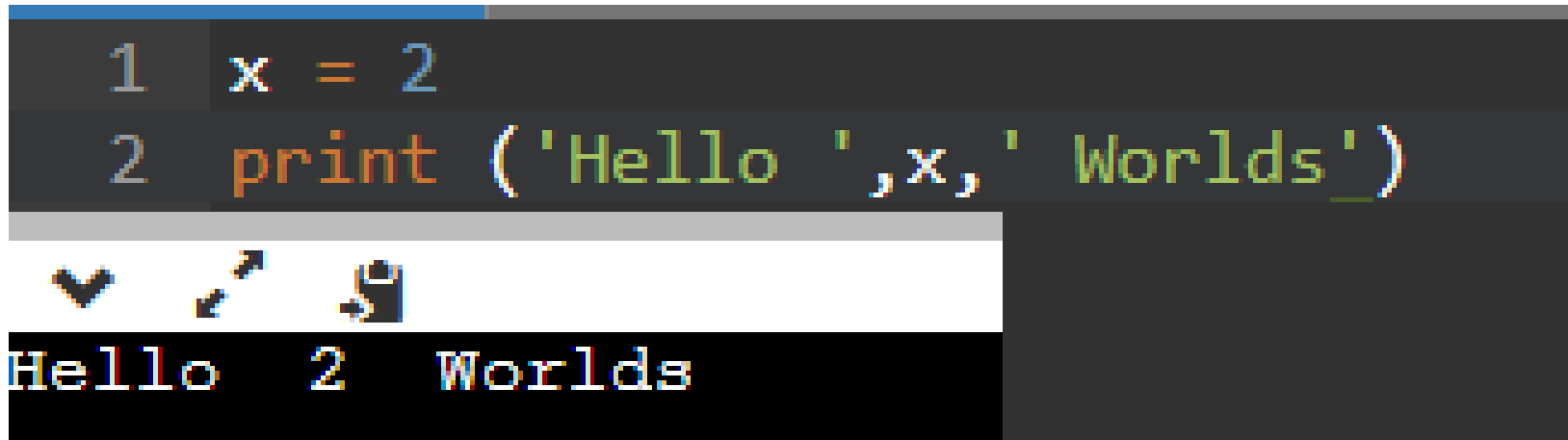
- Dzięki temu można łączyć tekst ze zmiennymi i innymi obiektami języka Python

```
x = 2
```

```
print ('Hello ', x, ' Worlds')
```

Przykład wykonania polecenia

```
1 x = 2
2 print ('Hello ',x,' Worlds')
```

The image shows a screenshot of a Python IDE. The top part is a dark-themed editor with two lines of code: '1 x = 2' and '2 print ('Hello ',x,' Worlds')'. Below the editor is a toolbar with icons for undo, redo, and a search icon. At the bottom, there is a black console window displaying the output 'Hello 2 Worlds' in a monospaced font.

Hello 2 Worlds

- W górnym okienku mamy tekst przedzielony zmienną x.
- W dolnym widzimy tekst rozdzielony liczbą 2

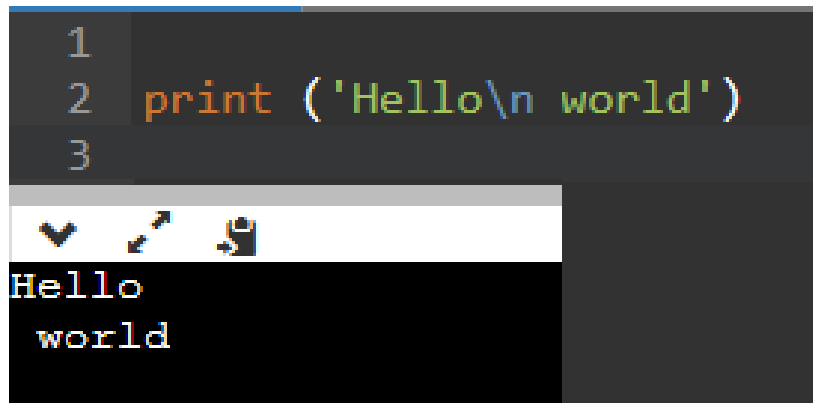
Koniec linii

- Koniec linii jest dodawany automatycznie przy poleceniu `print`.
- Chcąc wstawić czystą linię możemy użyć pustej instrukcji `print`.

```
print ()
```

- Innym rozwiązaniem jest użycie znaku końca linii `\n`.

```
print ('Hello\n world')
```

A screenshot of a code editor with a dark background. The editor shows three lines of code: line 1 is empty, line 2 contains `print ('Hello\n world')`, and line 3 is empty. Below the code editor, there is a toolbar with icons for undo, redo, and search. Below the toolbar, the output of the code is displayed: 'Hello' on the first line and 'world' on the second line, separated by a newline character.

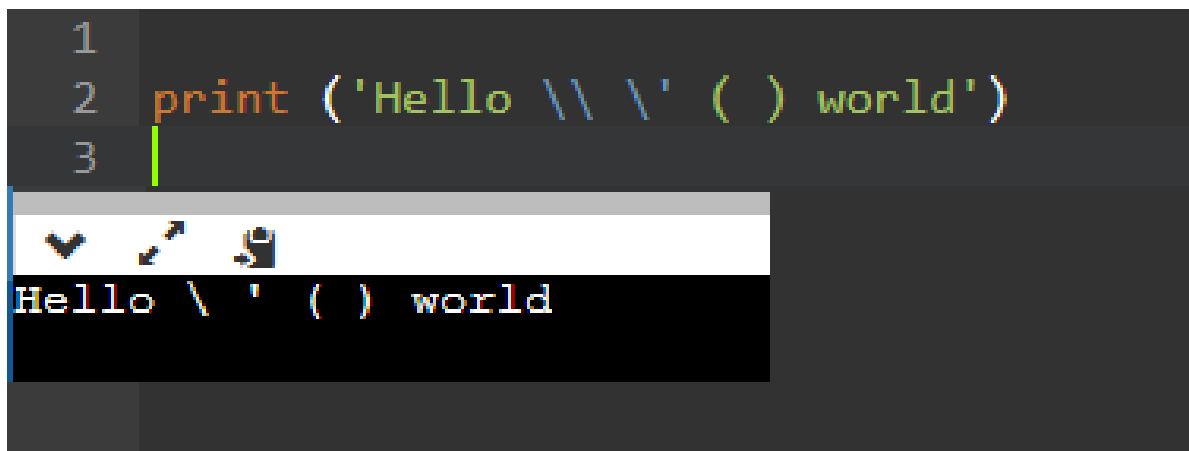
```
1  
2 print ('Hello\n world')  
3
```

Hello
world

Znaki specjalne

- W tekście (pomiędzy apostrofami) mogą być wyświetlane różne znaki ASCII.
- Chcąc wyświetlić apostrof w tekście, trzeba go poprzedzić znakiem backslasha \ '. Podobnie trzeba zrobić w wypadku innych znaków sterujących i specjalnych (jak \)

```
print ('Hello \\ \' ( ) world')
```



The screenshot shows a code editor with a dark background. Line 1 is empty. Line 2 contains the code `print ('Hello \\ \' ( ) world')` with syntax highlighting. Line 3 is empty. Below the code editor, there is a terminal window with a light background. It shows the command prompt icon, a cursor, and the output of the print statement: `Hello \ ' ( ) world`.

Ćwiczenie

1. Napisz program wyświetlające informacje o wybranym kraju:
 - a) Nazwa kraju, stolica, powierzchnia w km², główne miasta, sąsiedzi
2. Napisz program wyświetlające informacje o tobie:
 - a) Imię i nazwisko, ulubione hobby, miejscowość gdzie mieszkasz, praca / szkoła

Ćwiczenie

3. Sprawdź jakie działania dadzą następujące znaki specjalne:

a) `\\`

b) `\n`

c) `\'`

d) `\t`

e) `\b`

f) `\r`

g) `\f`

h) `\v`

i) `\0`

j) `\xhh`

Gdzie **hh** to liczba heksadecymalna

k) `\ooo`

Gdzie **oo** to liczba oktalna

l) `\uhhhh`

Gdzie **hhhh** to liczba heksadecymalna

m) `\Uhhhhhhhhh`

Gdzie **hhhhhhhh** to liczba heksadecymalna

n) `\"`

ASCII-ART

- To sposób tworzenia bardzo prostych rysunków układanych za pomocą zwykłych symboli ASCII (tzn. z liter, cyfr, znaków typograficznych).
- Tworzone są na obszarze o jednakowej szerokości kolumn (stałej ilości znaków w każdym wierszu) oraz o stałej wysokości znaków (tym samym stopniu pisma).
- Tego typu sposób tworzenia grafiki, a raczej jej symbolizowania, nosi nazwę ***semigrafiki***.

[illegible]

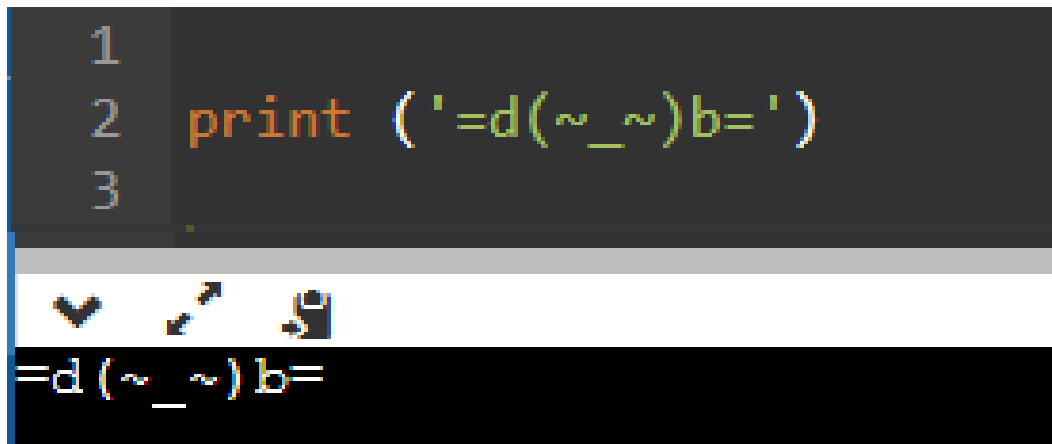
Programowanie ASCII-ART

- Człowiek w słuchawkach

=d (~_~) b=

- Realizacja w Pythonie

```
print ('=d (~_~) b=')
```



```
1
2 print ('=d (~_~) b=')
3
```

The screenshot shows a terminal window with a dark background. The first two lines of code are visible: a line number '1' followed by a blank line, and a line number '2' followed by the code `print ('=d (~_~) b=')`. The third line shows a line number '3' followed by a prompt character. Below the code, there are three small icons: a checkmark, a cursor, and a document. The output of the program, `=d (~_~) b=`, is displayed at the bottom of the terminal window.

Ćwiczenie

1. Napisz w Pythonie program wyświetlający następujące obrazki ASCII-ART

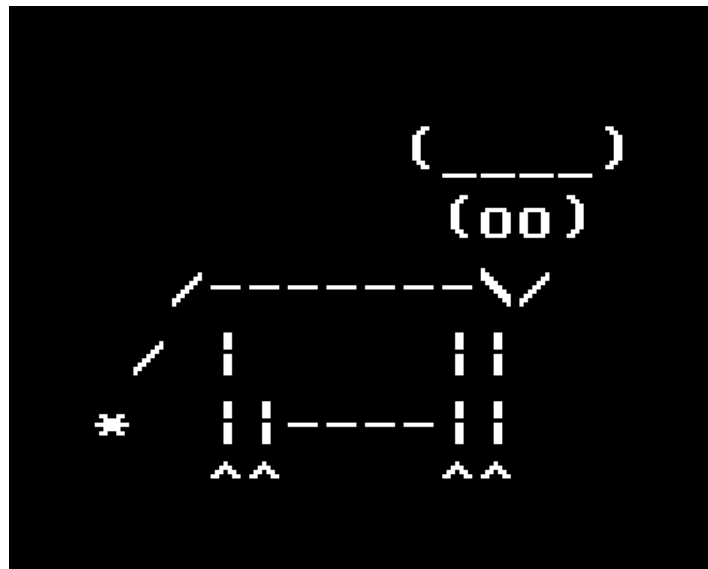
```
      @   @
.---ooO- ( _ ) -Ooo-----
|
|                               |
|               Imię i nazwisko |
|   .ooO      Ooo.           |
`--- ( _ ) --- ( _ ) -----'
      ( _ )      ( _ )
```

```
muu!  ( _ )      \_/_/
      ~O  o~      - ( _ ) -
      ( . _ . ) _ | \
      _____ | _ | _ | _____
```

```
      &%% .
      , '  %%%%=---. _ \
      "-"\%%  _ ( _ | \
           | / - " ) /  *
      -----" "-----" "-----
```

Ćwiczenie

2. Popularnym motywem ASCII-ART są krowy (**cow ascii art**). Wyszukaj kilka ilustracji i napisz program wyświetlający je. Połącz je z tekstem komentującym.



Ćwiczenie

3. Napisz swoje imię w systemie ASCII-ART.
według wzoru:



Powtórzenie

1. Co robi polecenie `print`?
2. Jak uzyskać podział linii?
3. W jaki sposób skleić kilka różnych fragmentów tekstu?
4. Jak zapisać jako tekst `'` lub `\`?