

Funkcja print()

Podstawy programowania w języku Python

```
print("Hello World!")
```

Funkcje w programowaniu

- w programowaniu wielokrotnie wykonuje się te same lub podobne operacje
- zamiast pisać je po wielokroć, można napisać powtarzalny kod tylko raz, a potem go wywoływać, kiedy tylko chcemy
- taki powtarzalny kod może zostać zapisany jako funkcja

Jak działa wywołanie funkcji?

- program wykonuje kolejne instrukcje linijka po linijce
- przy wywołaniu funkcji przeskakujemy na sam początek ciała funkcji i od tego miejsca idziemy dalej
- gdy dojdziemy do końca ciała (albo słowa `return`, o którym powiemy niedługo), wracamy do miejsca, gdzie funkcja była wywołana i stamtąd kontynuujemy

Rodzaje funkcji w Pythonie

- funkcje mogą być częścią samego języka
- mogą znajdować się w innych modułach
- mogą być definiowane przez użytkownika

Funkcja print()

Wyświetla na ekranie wszystko to co prześlemy jako jej argumenty

```
1 print("Hello World!")
```

- przyjmuje argumenty
- przekazane argumenty zamienia na ciągi znaków
- wysyła wyniki do urządzenia wyjściowego

Składnia funkcji print()

```
print(object(s), sep=separator, end=end, file=file, flush=flush)
```

- *object(s)* - dowolny obiekt w dowolnej ilości, przed wyświetleniem zostanie przekonwertowany na ciąg znaków
- *sep='separator'* – opcjonalnie, określa sposób rozdzielania obiektów, jeśli jest ich więcej niż jeden, wartość domyślna to " " (spacja)
- *end='end'* – opcjonalnie, określa co wyświetlić na końcu, wartość domyślna to "\n" (znak nowej linii)

Sposoby przekazywania argumentów do funkcji

- **argumenty przekazywane pozycyjnie** – ich znaczenie określone jest przez ich pozycję, np. drugi argument umiejscowiony jest na wyjściu po pierwszym, trzeci po drugim, itd.

```
1 print("raz", "dwa", "trzy")
```

- **argumenty w postaci słów kluczowych** – ich znaczenie **nie** jest określone przez ich pozycję, ale przez słowa specjalne (słowa-klucze) używane do ich identyfikacji

```
1 pow(base=2, exp=3)
2 pow(exp=3, base=2)
```

Mieszany sposób przekazywania argumentów do funkcji

```
1 print("raz", "dwa", "trzy", sep="*")  
2 print("Hello, World!", sep=" ", end="")
```

jakikolwiek argumenty w postaci słów kluczowych muszą być umiejscowione **po ostatnim argumencie pozycyjnym**

Komentarze

- komentarze mogą być użyte w celu zostawienia w kodzie dodatkowych informacji
- są pomijane w czasie wykonywania programu
- informacje pozostawione w kodzie źródłowym są adresowane do innych osób przeglądających nasz kod
- w Pythonie komentarz jest fragmentem tekstu zaczynającym się od znaku #
- komentarz rozciąga się do końca linii
- można stosować także komentarze blokowe

```
1  #To jest komentarz liniowy
2
3  """
4      To jest przykład
5      komentarza blokowego
6  """
```

Pytanie

Co wypisze poniższy kod w języku Python?

```
1 print('2+2')
```

- a) kod nie jest poprawny (podczas wykonania zostanie zgłoszony błąd)
- b) 2+2
- c) 4
- d) 22

Odpowiedź: b)

Pytanie

Za co odpowiada parametr **sep** w funkcji **print()**?

- a) za jego pomocą możemy określić jakim znakiem zakończy się wyświetlanie
- b) określa jak rozdzielone będą ciągi znaków pochodzące z tej samej funkcji **print()**
- c) określa jak rozdzielone będą ciągi znaków pochodzące z różnych funkcji **print()**
- d) określa jak rozdzielane będą poszczególne znaki przekazanego do funkcji obiektu

Odpowiedź: b)

Pytanie

Co wyświetli się na ekranie po wywołaniu poniższej instrukcji?

```
1 print("ma", "ka", "ron", sep="", end="*")
```

- a) ma ka ron
- b) ma*ka*ron
- c) makaron*
- d) ma*ka*ron*

Odpowiedź: c)