

22. Передача переменного количества параметров (именованных и не именованных) в функции Python. Вызов функции с позиционными параметрами, находящимися в списке, и именованными параметрами, находящимися в словаре.

Передача переменного количества параметров (именованных и не именованных) в функции Python.

Можно использовать * в начале параметра функции, тогда можно передавать неограниченное кол-во аргументов.

```
def multi_example(*numbers):  
    """  
    Функция, получающая неограниченное кол-во аргументов для подсчёта их суммы  
    """  
    result = 0  
    for number in numbers:  
        result += number  
  
    return result  
  
if __name__ == "__main__":  
    print(multi_example(5,3,6,2,7)) #23
```

Также можно использовать ** для передачи множества аргументов с произвольными именами


```
def multi_example(**dct):  
    """  
    Функция, получающая неограниченное кол-во аргументов с произвольным именем  
    """  
    args_list = []  
    values_list = []  
    for arg, value in dct.items():  
        args_list.append(arg)  
        values_list.append(value)  
        print(arg, value)  
  
    return args_list, values_list  
  
if __name__ == "__main__":  
    print(multi_example(a=4, koshkas=5))
```

Вызов функции с позиционными параметрами, находящимися в списке, и


именованными параметрами, находящимися в словаре.

Передавать параметры функции можно тремя способами.


Во-первых, простым перечислением аргументов:

```
Users > georgiydemo > Desktop >  check.py > ...  
1  def multi_example(a, b, c):  
2      |      return a+b+c  
3  
4  if __name__ == "__main__":  
5      |      result = multi_example(1,2,4)
```

Во-вторых, с помощью *и list

```
Users > georgiydemo > Desktop >  check.py > ...  
1  def multi_example(a, b, c):  
2      |      return a+b+c  
3  
4  if __name__ == "__main__":  
5      |      l = [1,2,4]  
6      |      result = multi_example(*l)
```

В-третьих, с помощью ** и dict


```
Users > georgiydemo > Desktop >  check.py > ...  
1  def multi_example(a, b, c):  
2      |      return a+b+c  
3  
4  if __name__ == "__main__":  
5      |      this_dict = {"a": 1, "b" : 2, "c" : 4}  
6      |      result = multi_example(**this_dict)
```

Если есть значения по-умолчанию для функции, то можно и ничего не передавать совсем

Users > georgiydemo > Desktop >  check.py > ...

```
1  def multi_example(a=1, b=2, c=4):
2      return a+b+c
3
4  if __name__ == "__main__":
5      result = multi_example()
```

Или предавать только часть аргументов, указывая имя аргумента:

Users > georgiydemo > Desktop >  check.py > ...

```
1  def multi_example(a=1, b=2, c=4):
2      return a+b+c
3
4  if __name__ == "__main__":
5      result = multi_example(c=5)
6      print(result) #8
```