

```
1 # Деф
2 # Не би ли било хубаво да създадем свои собствени функции?
3 # Нека върнем нашия окончателен код на hello.py, като напишем код
  hello.py в прозореца на терминала.
4 # Вашият начален код трябва да изглежда по следния начин:
5 """
6 # Попитайте потребителя за неговото име, премахнете интервала от str и
  главна първа буква на всяка дума
7 name = input("What's your name? ").strip().title()
8
9 # Отпечатайте изхода
10 print(f"hello, {name}")
11
12 """
13 # Можем да подобрим нашия код, за да създадем наша собствена специална
  функция, която казва „hello“ за нас!
14 # Изтривайки целия си код в нашия текстов редактор, нека започнем от
  нулата:
15 """
16 name = input("What's your name? ")
17 hello()
18 print(name)
19
20 """
21 # При опит да изпълните този код, вашият компилатор ще изведе грешка.
22 # В крайна сметка няма дефинирана функция за hello.
23
24 # Можем да създадем наша собствена функция, наречена hello, както
  следва:
25 """
26 def hello():
27     print("hello")
28 name = input("What's your name? ")
29 hello()
30 print(name)
31
32 """
33 # Забележете, че всичко под def hello() е с отстъп. Python е език с
  отстъп.
34 # Той използва отстъп, за да разбере какво е часта от горната функция.
35 # Следователно всичко във функцията hello трябва да бъде с отстъп.
36 # Когато нещо не е с отстъп, то го третира така, сякаш не е във
  функцията hello.
37 # Изпълнявайки python hello.py в прозореца на терминала, ще видите, че
  резултатът ви не е точно такъв,
38 # какъвто бихте искали. Можем допълнително да подобрим нашия код:
39
40 """
41 # Create our own function
42 def hello(to):
43     print("hello,", to)
44
45 # Output using our own function
```

```
46 name = input("What's your name? ")
47 hello(name)
48
49 """
50 # Тук, в първите редове, вие създавате вашата функция hello.
51 # Този път обаче вие казвате на компилатора, че тази функция приема
  един параметър:извикана променлива.
52 # Следователно, когато извикате hello(name), компютърът предава name
  във функцията hello.
53 # Ето как предаваме стойности във функции.
54 # Много полезно! Стартуране на python hello.py в прозореца на
  терминала,
55 # ще видите, че резултатът е много по-близък до нашия пример,
  представен по-рано в тази лекция.
56
57 # Можем да променим нашия код, за да добавим стойност по подразбиране
  към hello:
58
59 # СЪЗДАВАНЕ НА НАША СОБСТВЕНА ФУНКЦИЯ
60
61 """
62 def hello(to="world"):
63     print("hello,", to)
64
65
66 # Output using our own function
67 name = input("What's your name? ")
68 hello(name)
69
70 # Output without passing the expected arguments
71 hello()
72
73 """
74 # Тествайте кода си сами.
75 # Забележете как първото здравей ще се държи, както може да очаквате,
76 # а второто здравей, на който не е подадена стойност, по подразбиране
  ще изведе hello, world.
77 # Не е нужно да имаме нашата функция в началото на нашата програма.
78 # Можем да го преместим надолу,
79 # но трябва да кажем на компилатора че имаме основна функция и имаме
  отделна функция здравей.
80 """
81 def main():
82
83     # Output using our own function
84     name = input("What's your name? ")
85     hello(name)
86
87     # Output without passing the expected arguments
88     hello()
89
90
91 # Create our own function
```

```

92 def hello(to="world"):
93     print("hello,", to)
94     """
95     # Само това обаче ще създаде нещо като грешка. Ако стартираме python
    hello.py, нищо не се случва!
96     # Причината за това е, че нищо в този код всъщност не извиква
    основната функция и неживява на нашата програма.
97     # Следната много малка модификация ще извика основната функция и ще
    възстанови нашата програма в работно състояние:
98
99     """
100 def main():
101
102     # Output using our own function
103     name = input("What's your name? ")
104     hello(name)
105
106     # Output without passing the expected arguments
107     hello()
108
109
110 # Create our own function
111 def hello(to="world"):
112     print("hello,", to)
113
114
115 main()
116 """
117 # ВРЪЩАНЕ НА СТОЙНОСТ
118
119 # Можете да си представите много сценарии, при които не искате просто
    функция да извърши действие,
120 # но също и за връщане на стойност обратно към основната функция.
121 # Например, вместо просто да отпечатате изчислението на  $x + y$ , може
    да искате функция,
122 # която да върне стойността на това изчисление обратно към друга част
    от вашата програма.
123 # Това „връщане“ на стойност наричаме върната стойност.
124 # Връщане към нашия код calculator.py чрез въвеждане на код
    calculator.py.
125 # Изтрийте целия код там. Преработете кода, както следва:
126
127 """
128 def main():
129     x = int(input("What's x? "))
130     print("x squared is", square(x))
131
132 def square(n):
133     return n * n
134
135 main()
136
137 """

```

```
138 # Ефективно  $x$  се предава на квадрат. След това изчислението на  $x * x$   
    се връща обратно към основната функция.  
139
```