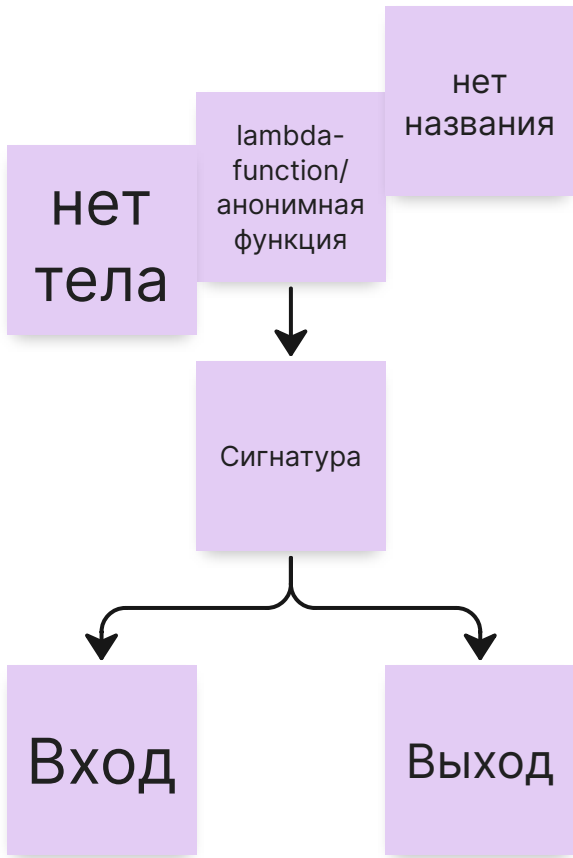


```
1 def jump(arg1: str) -> str:
2     """
3     def == define, с англ. "Определить".
4     () – оператор вызова, определяет "Вход" в сигнатуре функции.
5     -> <type> – нотация выхода для сигнатуры
6     arg1 – аргумент. Переменная, в которую функция на входе принимает
    значения.
7     <arg>: <type> – нотация указания типа аргумента. Иначе говоря
    "Аннотация".
8     """
9     if arg1 == "before":
10         result = "dont"
11     elif arg1 == "after":
12         result = "Hop"
13     else:
14         result = "Masha is fine..."
15
16     return result
17
18 def walk(temp: int, where: str = "куда глаза глядят") -> int:
19     """
20     temp: аргумент, указывающий скорость
21     where: аргумент, указывающий куда мы идем
22
23     return: сколько километров мы прошли
24     """
25
26 walk(5) # Передал аргумент temp позиционно, потому что знаю, что он идет
    первым; Аргумент where получил значение по умолчанию
27 walk(temp=5) # Передал аргумент temp по имени, дальше аргументы по
    позициям передавать нельзя.
28 walk(5, where="Красная площадь") # Передал аргумент temp позиционно,
    переопределил именованный аргумент where со значения по умолчанию на
    новое значение
29 walk(5, "Красная площадь") # не сработает, именованный аргумент передан
    позиционно
30
31 walk(where="Красная площадь") # не сработает, не передан обязательный
    позиционный элемент
```



```
1 lambda x: x ** 2
2
3 def lambda_1(x):
4     # тела нет, потому что это лямбда.
5     return x ** 2 # что поместится в return – то и тело
6 # инлайн – это операция в одну строку
7 lambda x, y: x + y
8
9 def lambda_2(x, y):
10     return x + y
11
```