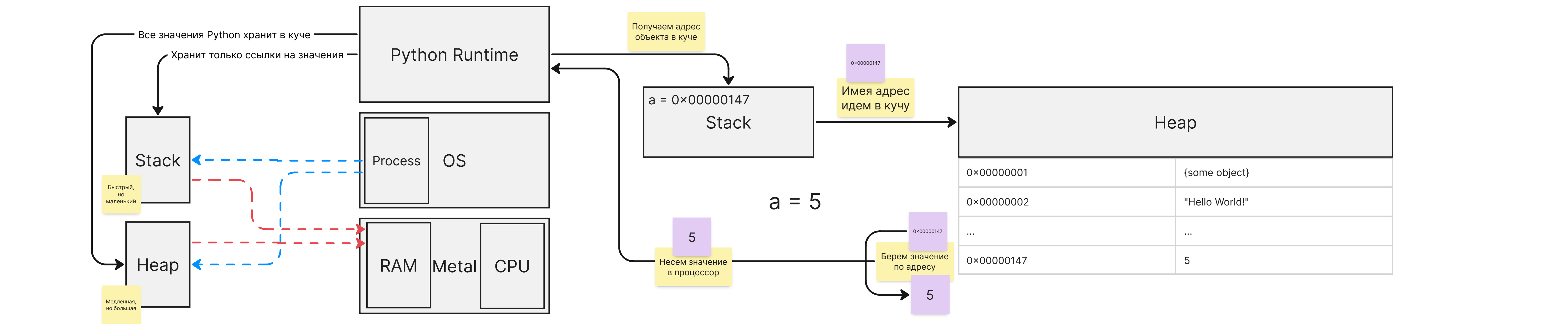


Переменные в питоне



Stack — раздел оперативной памяти процесса, который работает по принципу FIFO (First In, First Out)

Heap — раздел оперативной памяти, который организован по принципу объектного хранилища

```
1 # Numerics
2 float_example = 7.5
3 int_example = 5
4 bool_example = True # Could be either False
5
6 # Strings
7 string_a = "That's how you initialize a string"
8 string_b = 'Single quotes are working too'
9 print(string_a)
10
11 # Casting between types
12 numeric_int = 5 # int type
13 stringified_numeric_int = str(numeric_int) # "5" - string type now
14 numeric_again = int(stringified_numeric_int) # 5 - int again
15
16 bool(0); bool(""); bool(list()); bool(dict()); bool([]) # False
17 bool("Hello"); bool(5); bool(9.99) # True
18
```

```
1 # Операции над строками
2 # Строки можно складывать
3 a = "Hello"
4 space = " "
5 b = "World!"
6
7 string_ab = a + space + b
8 from pprint import pprint # модуль pprint выводит значения как и print, но делает их удобнее
9 pprint(string_ab) # 'Hello World!'
10
11 # Методы строк - у класса str есть множество встроенных методов для операций над строками.
12 # Методы - это функции, определенные в классе. Чтобы вызвать метод, нужно написать точку у
13 # объекта, написать название метода и оператор вызова (круглые скобки). Возможно также
14 # потребуется передать какие-то аргументы в вызов.
15 string_ab.upper() # HELLO WORLD!
16 string_ab.lower() # hello world!
17
18 # Строковая интерполяция - механизм языка, который упрощает компонование строк, позволяет
19 # встраивать переменные и операции в строки.
20 # Для интерполяции нужно перед строкой написать букву f, после этого можно внутри строки
21 # писать фигурные скобки и в них добавлять переменные!
22 name = "Kitsuragi"
23 greeting = f"Hello {name}!" # 'Hello Kitsuragi!'
```

