

CS SCHOOL

CPython VM

1. Virtual Stack Machine
 - stack-based
2. Operand Stack

How does VM work?

```
>>> def add(a, b):  
    c = a + b  
    return c  
  
>>> import dis  
>>> dis.dis(add)  disassembling  
2          0 LOAD_FAST          0 (a)  
           2 LOAD_FAST          1 (b)  
           4 BINARY_ADD  
           6 STORE_FAST         2 (c)  
  
3          8 LOAD_FAST          2 (c)  
          10 RETURN_VALUE
```

How does VM work?

```
0 LOAD_FAST
2 LOAD_FAST
4 BINARY_ADD
6 STORE_FAST

8 LOAD_FAST
10 RETURN_VALUE
```

```
0 (a)
1 (b)

2 (c)

2 (c)
```

Operand stack



a

The operand stack is represented by a green rectangular box containing the value 'a'.

How does VM work?

```
0 LOAD_FAST
2 LOAD_FAST
4 BINARY_ADD
6 STORE_FAST

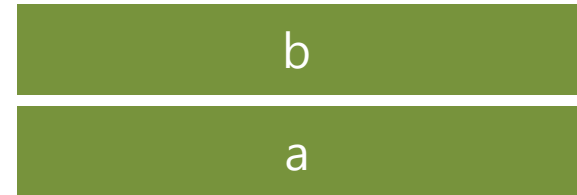
8 LOAD_FAST
10 RETURN_VALUE
```

```
0 (a)
1 (b)

2 (c)

2 (c)
```

Operand stack



How does VM work?

```
0 LOAD_FAST
2 LOAD_FAST
4 BINARY_ADD
6 STORE_FAST

8 LOAD_FAST
10 RETURN_VALUE
```

```
0 (a)
1 (b)

2 (c)

2 (c)
```

Operand stack

a + b

How does VM work?

Private heap



$c = a + b$

```
0 LOAD_FAST
2 LOAD_FAST
4 BINARY_ADD
6 STORE_FAST

8 LOAD_FAST
10 RETURN_VALUE
```

```
0 (a)
1 (b)

2 (c)

2 (c)
```

How does VM work?

```
0 LOAD_FAST
2 LOAD_FAST
4 BINARY_ADD
6 STORE_FAST

8 LOAD_FAST
10 RETURN_VALUE
```

```
0 (a)
1 (b)

2 (c)

2 (c)
```

Operand stack

c

How does VM work?

```
0 LOAD_FAST
2 LOAD_FAST
4 BINARY_ADD
6 STORE_FAST
```

```
8 LOAD_FAST
10 RETURN_VALUE
```

```
0 (a)
1 (b)
```

```
2 (c)
```

```
2 (c)
```

returns c