



```
main ( ) {  
    (name) = "Kunal"  
    greet(name);  
}  
  
greet (naam) {  
    print(naam)  
}
```

Diagram illustrating variable assignment and function call:

- `(name)` is assigned the value `"Kunal"`. The word "Object" is written above the `"Kunal"` value.
- The function `greet` is called with `(name)` as an argument.
- A separate circle labeled `Value` is shown.

Diagram illustrating parameter passing:

- The variable `name` is passed to the function `greet` as the argument `naam`.
- The value `"Kunal"` is shown as the result of the function call.

```
psvm() {
```

```
  name = Kunal
```

```
  change(name)
```

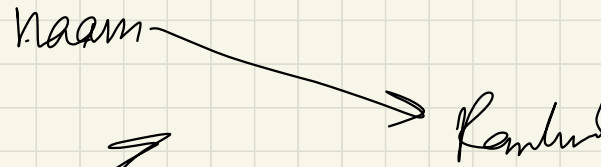
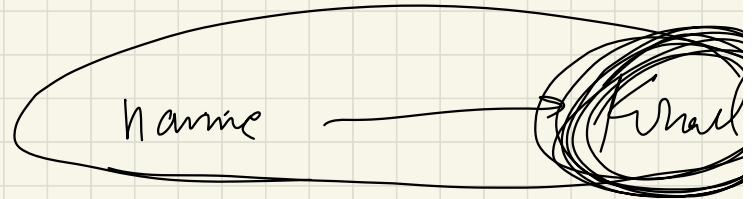
```
  print(name)
```

```
}
```

```
change(name) {
```

```
  naam = Rahul
```

```
}
```



// not changing, creating a new object

② * primitives: int, short, char, byte . . .

just ↙ passing value

* objects & stuff : passing value of the ref

```
psvm() {
```

```
  a = 10
```

```
  b = 20
```

```
  swap(a, b)
```

```
}
```

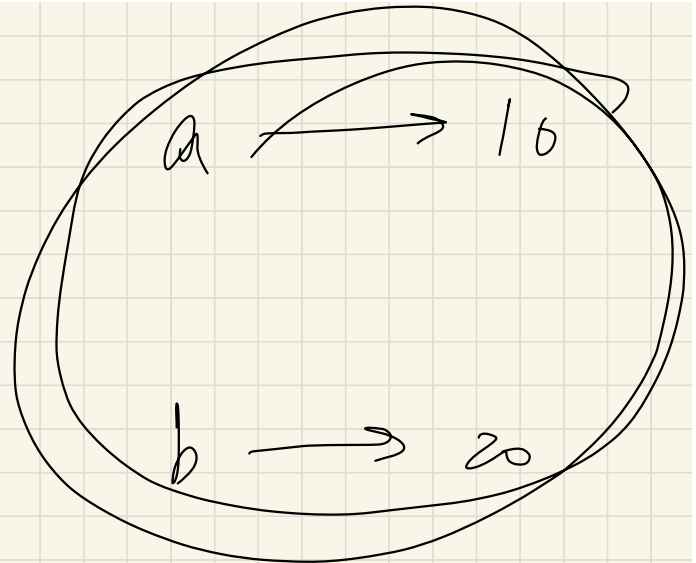
```
swap(num1, num2) {
```

```
  temp = num1
```

```
  num1 = num2
```

```
  num2 = temp
```

```
}
```

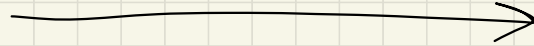


```
temp = 10
```

```
num1 = 20
```

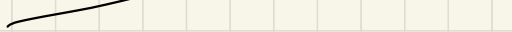
```
num2 = 10
```

(arr)



⁰
[99, ¹3, ²2, ³45, ⁴6]

nums



in function :

nums[0] = 99

if (name = kumal

{

name = kumal

}

print(name)

name

kumal

kumal

$$a = \overline{153}$$

$$\begin{array}{ccc} \swarrow & \downarrow & \searrow \\ 3 & 3 & 3 \\ 1 + & 5 + & 3 \end{array}$$

$$\begin{array}{r} 125 \\ + 27 \\ \hline \overline{153} \end{array}$$

$$= 1 + 125 + 27 = \overline{153}$$

$$a = 15$$

$$3^3 \quad 5^3 = 125$$

while (a > 0) {

$$\text{rem} = a \% 10$$

$$\text{cube} = \text{rem}$$

$$\text{sum} += \text{cube}$$

$$a = a / 10$$

}