

```
def getvalue():  
    x = {1:'lokesh', 2.0:'srinath', 2:'jagadeesh'}  
    print(x)  
    return x[2]
```

```
print(getvalue())
```

```
{1: 'lokesh', 2.0: 'jagadeesh'}  
jagadeesh
```

{ 1: 'lokesh',  
2.0: ~~'srinath'~~,  
      'jagadeesh'  
}

2 == 2.0  
2.0 == 2.0

```
def getvalue():  
    x = {1: 'lokesh', 2: 'srinath', 2.0: 'jagadeesh'}  
    print(x)  
    return x[2]
```

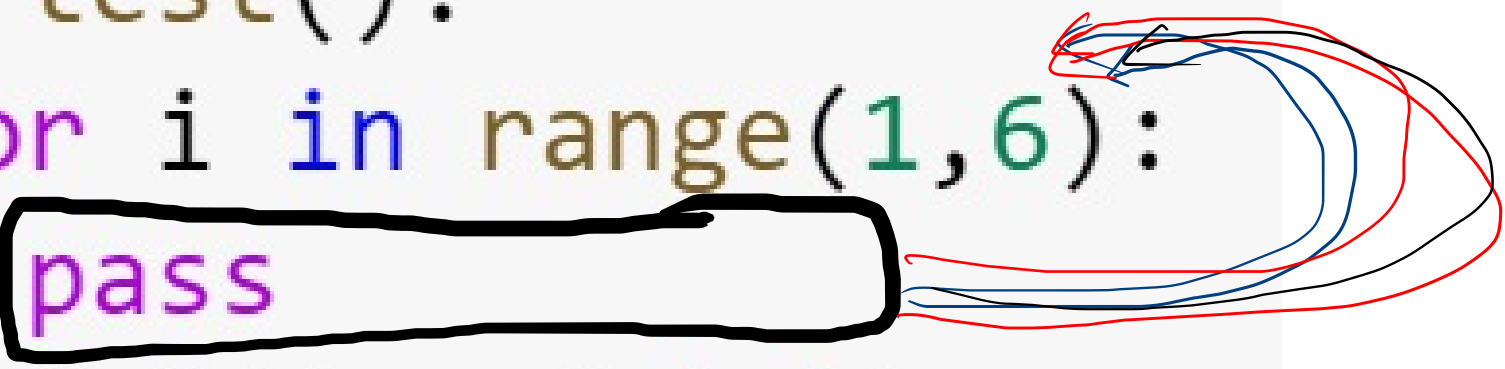
```
print(getvalue())
```

```
{1: 'lokesh', 2: 'jagadeesh'}  
jagadeesh
```

{ 1: 'lokesh',  
2: 'srinath',  
'jagadeesh'  
}

$2.0 == 2$   
 $2.0 == 2.0$

```
def test():  
    for i in range(1,6):  
        pass  
→ print(i, end=' ')
```



The diagram illustrates the execution of the code. A black box highlights the 'pass' statement. Colored arrows (red, blue, black) originate from the 'pass' statement and point to the 'range(1,6):' line, representing the loop iterations. A black arrow points to the 'print' statement.

```
test()
```

```
def myfun():  
    even = []  
    odd = []  
    for i in range(10, 21):  
        if i%2 == 0:  
            even.append(i)  
        else:  
            odd.append(i)  
    return even, odd  
return None
```

```
def mytest():  
    for i in range(1,6):  
        continue  
        return 100  
        break  
    return None  
  
print(mytest())
```

```
def mytest():  
    for i in range(1,6):  
        return 'even' if i%2==0 else 'odd'  
    return None  
  
print(mytest())
```

odd