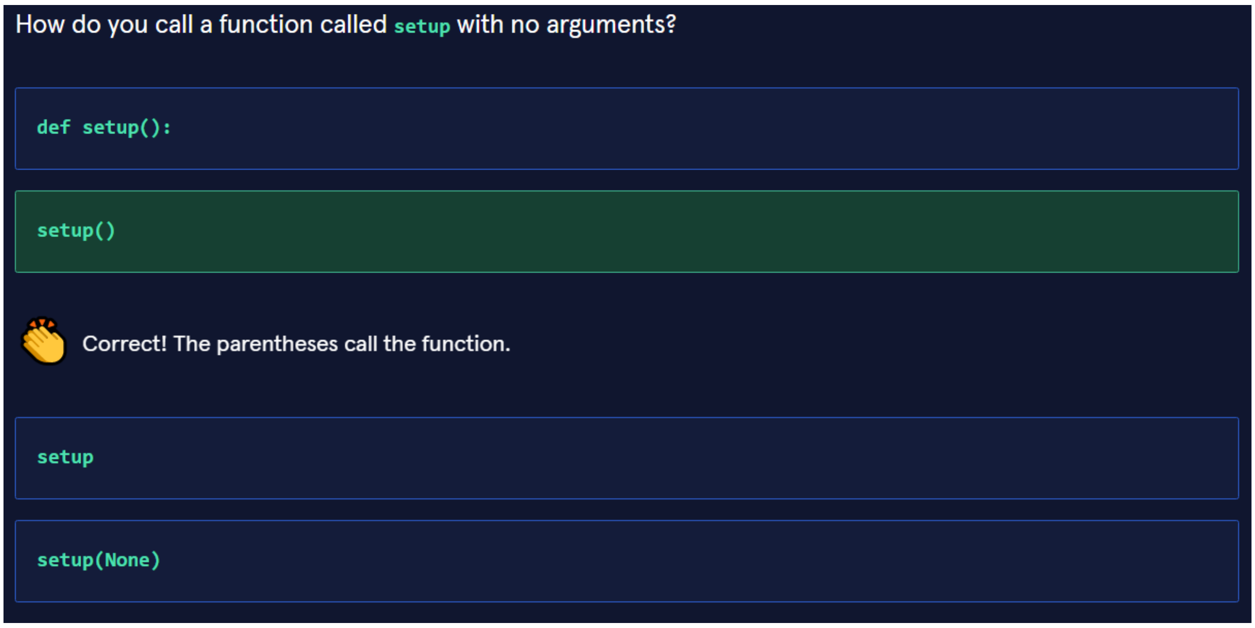
QUIZ



How do you call a function called `setup` with no arguments?

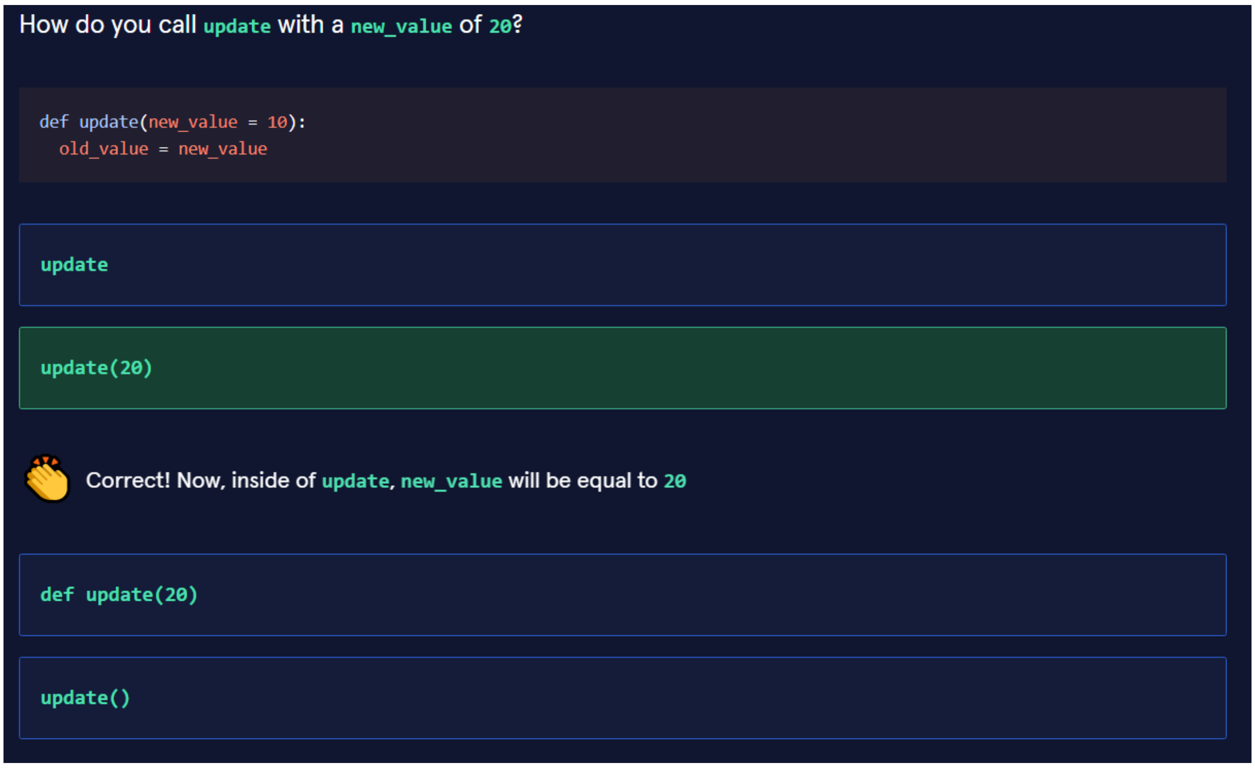
`def setup():`

`setup()`

Correct! The parentheses call the function.

`setup`

`setup(None)`



How do you call `update` with a `new_value` of `20`?

```
def update(new_value = 10):
  old_value = new_value
```

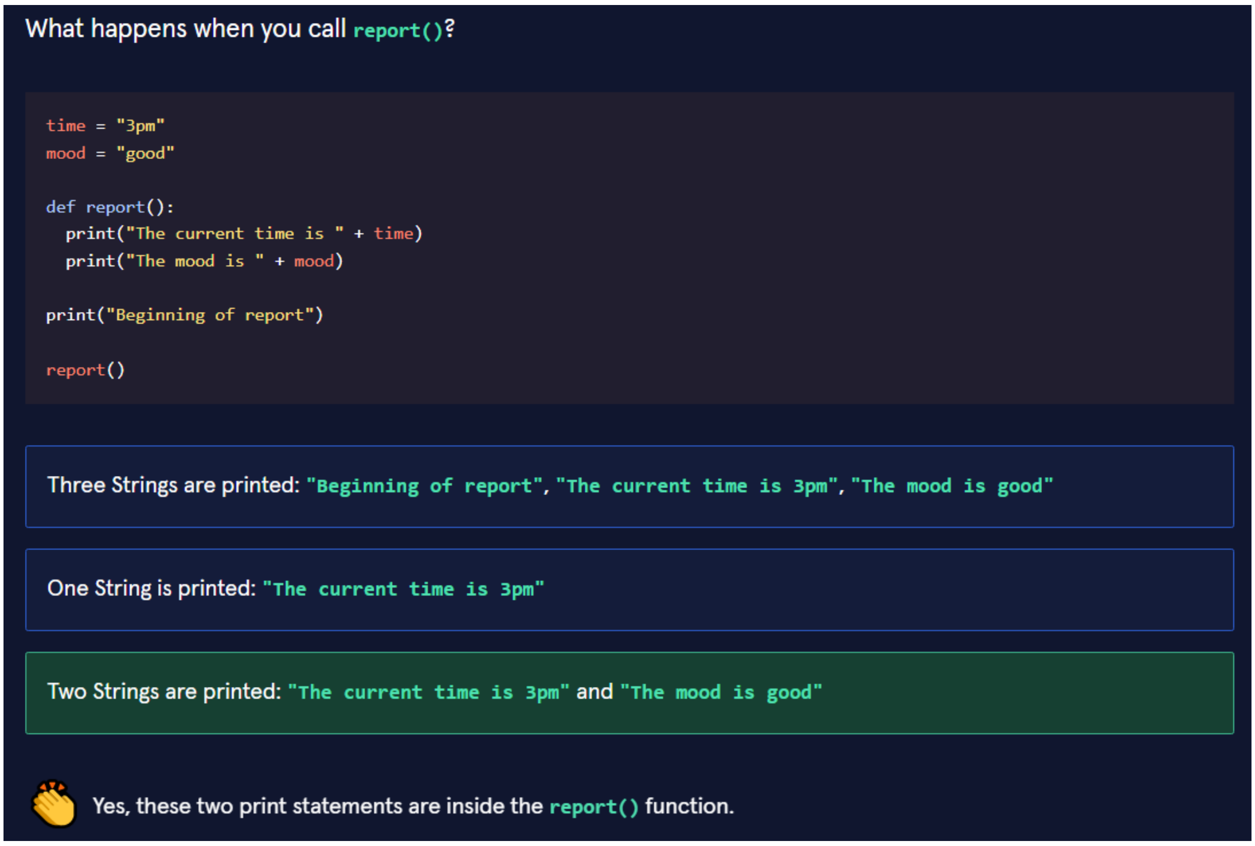
`update`

`update(20)`

Correct! Now, inside of `update`, `new_value` will be equal to `20`

`def update(20)`

`update()`

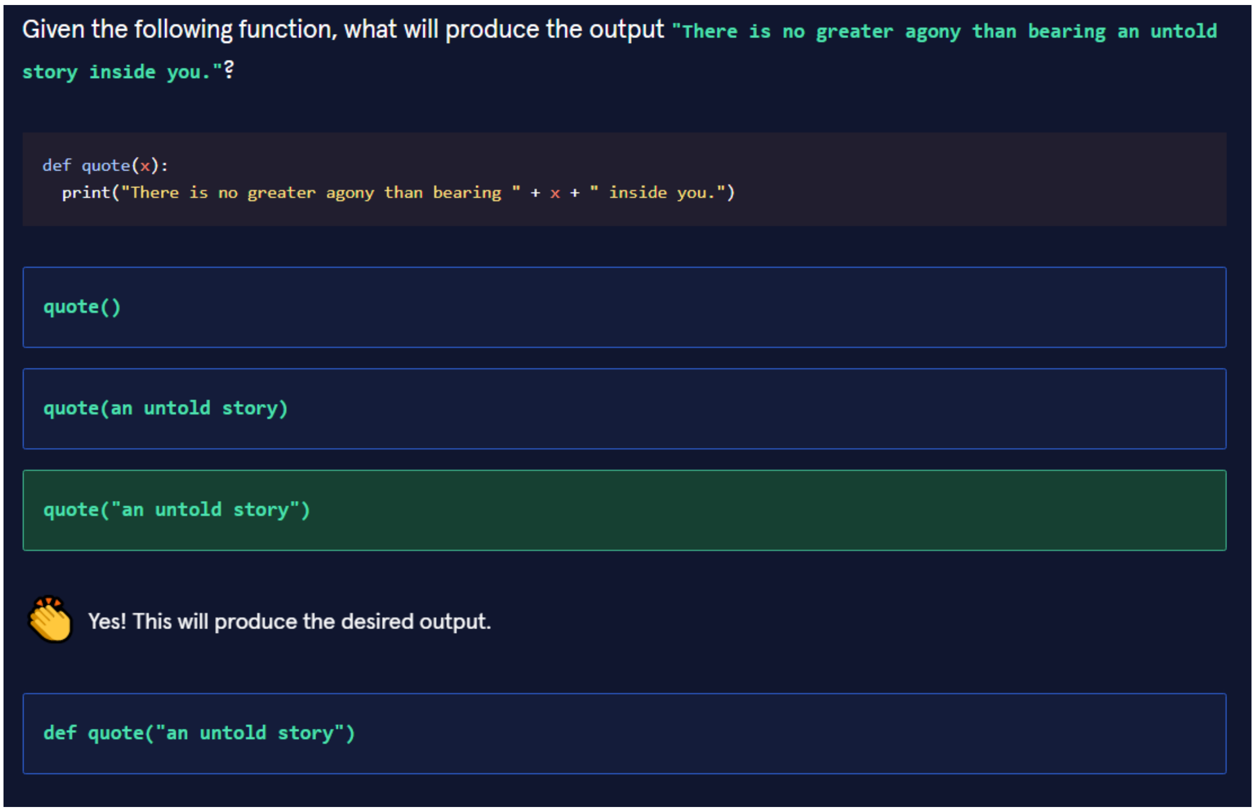


What happens when you call `report()`?

```
time = "3pm"
mood = "good"

def report():
  print("The current time is " + time)
  print("The mood is " + mood)

print("Beginning of report")

report()
```

- Three Strings are printed: `"Beginning of report"`, `"The current time is 3pm"`, `"The mood is good"`
- One String is printed: `"The current time is 3pm"`
- [✓ Two Strings are printed: `"The current time is 3pm"` and `"The mood is good"`]

👏 Yes, these two print statements are inside the `report()` function.



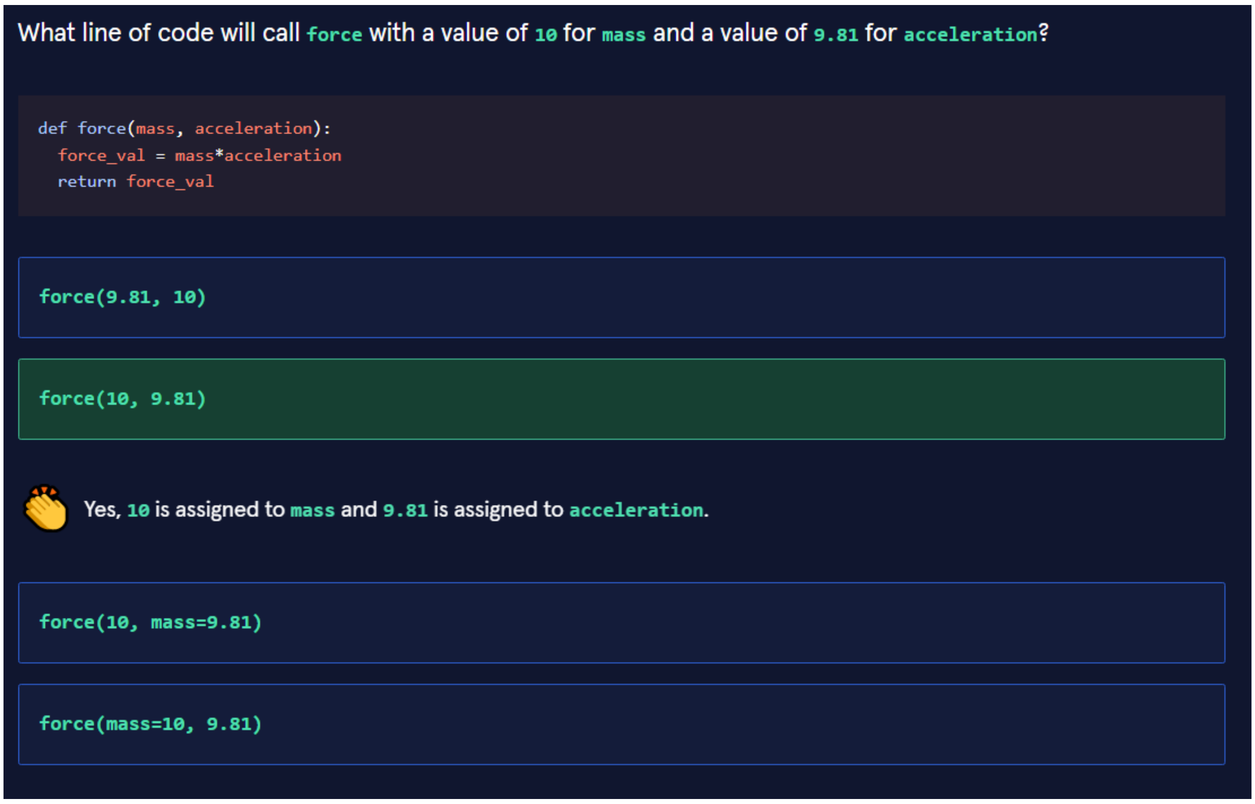
Given the following function, what will produce the output `"There is no greater agony than bearing an untold story inside you."`?

```
def quote(x):
  print("There is no greater agony than bearing " + x + " inside you.")
```

- `quote()`
- `quote(an untold story)`
- [✓ `quote("an untold story")`]

👏 Yes! This will produce the desired output.

- `def quote("an untold story")`



What line of code will call `force` with a value of `10` for `mass` and a value of `9.81` for `acceleration`?

```
def force(mass, acceleration):
  force_val = mass*acceleration
  return force_val
```

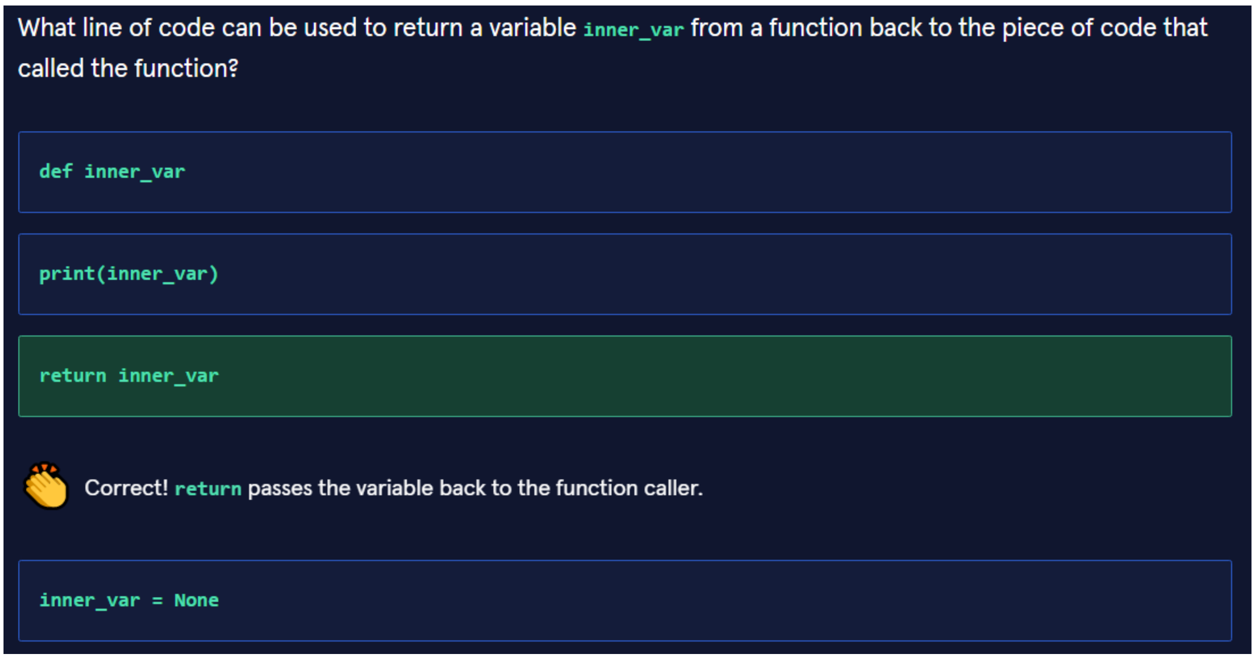
`force(9.81, 10)`

[✓ `force(10, 9.81)`]

Yes, `10` is assigned to `mass` and `9.81` is assigned to `acceleration`.

`force(10, mass=9.81)`

`force(mass=10, 9.81)`



What line of code can be used to return a variable `inner_var` from a function back to the piece of code that called the function?

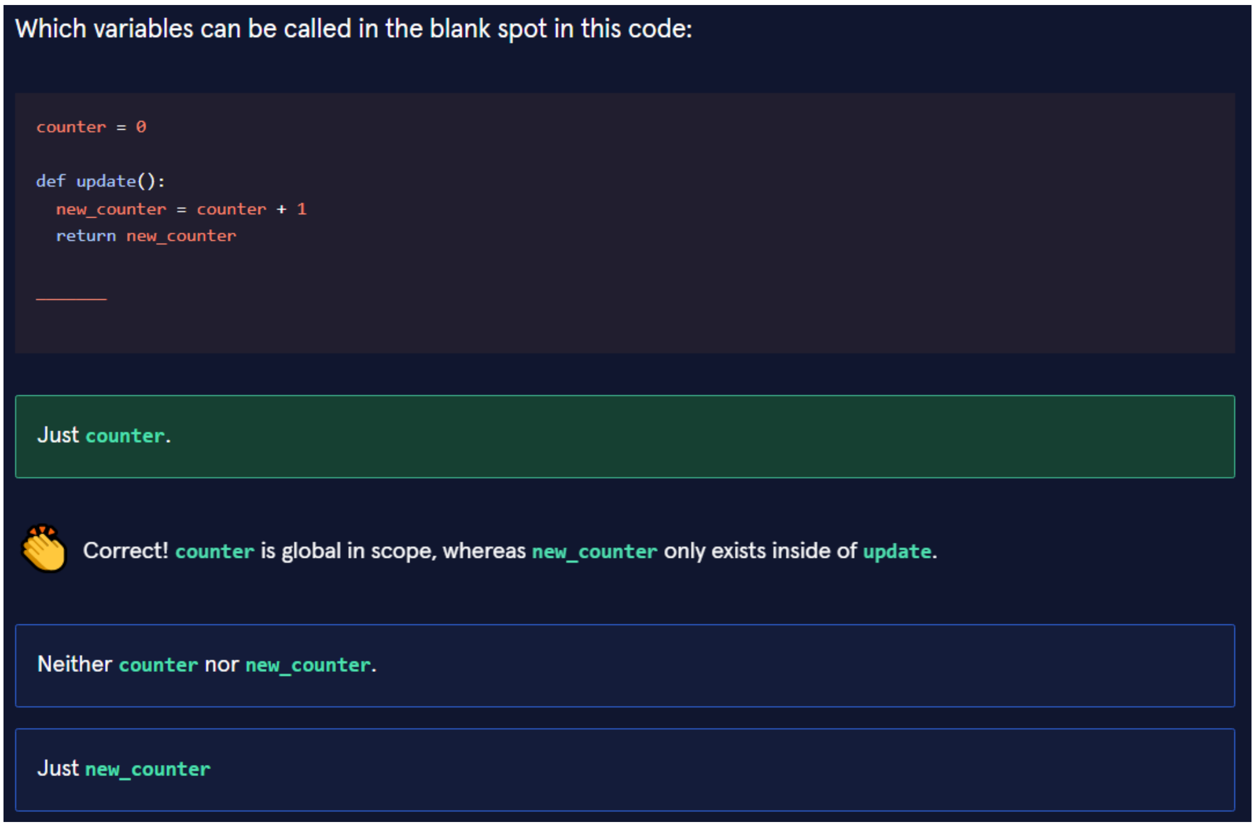
`def inner_var`

`print(inner_var)`

[✓ `return inner_var`]

Correct! `return` passes the variable back to the function caller.

`inner_var = None`



Which variables can be called in the blank spot in this code:

```
counter = 0

def update():
  new_counter = counter + 1
  return new_counter


_______
```

[✓ Just `counter`.]

Correct! `counter` is global in scope, whereas `new_counter` only exists inside of `update`.

Neither `counter` nor `new_counter`.

Just `new_counter`