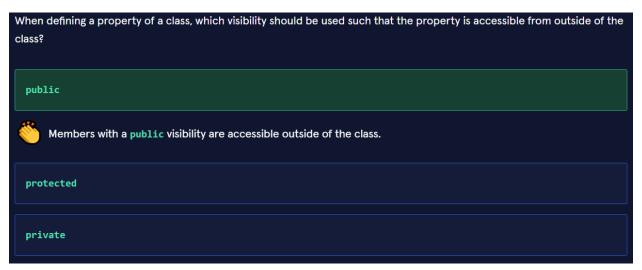
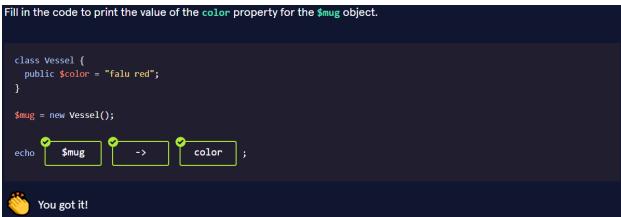
QUIZ





```
class Vehicle {
    private $speed;
    function _construct($speed) {
        $this->speed = $speed;
    }
}

class Train extends Vehicle {
    function getSpeed() {
        return $this->speed;
    }
}

$stopped_train = new Train(0);
    echo $stopped_train->getSpeed();

Private members, like speed cannot be accessed in the constructor.

Train has no constructor.

The Train class does not have access to speed.

**speed is defined as private to Vehicle. It would need to be protected or public for this code to work.
```

Fill in the code to access and print the value associated with the static property capacity.

class Seat {
 public static \$capacity = 1;
 }

echo Seat :: \$capacity;

You got it!

When defining a property of a class, which visibility should be used such that the property is only accessible from within the class itself (and not from any descendant classes)? protected public private Members with a private visibility are accessible only within the class itself. Fill in the code to create a Sneaker class that inherits from the Footwear class. | Footwear Sneaker extends # Sneaker class definition You got it! When defining a property of a class, which visibility should be used such that the property is not accessible from outside the class but is accessible within descendant classes? public private protected Members with a protected visibility are accessible within the class itself and its descendants.