Division

Division

Division in Python is done with the `/` operator

```
double a = 25;
double b = 4;
System.out.println(a / b);
```

TRY IT

```
6.25
```

What happens if you:

- Change `b` to `0`?
- Change `b` to `0.5`?
- Change the code to

```
double a = 25;
double b = 4;
a /= b;
System.out.println(a);
```

TRY IT

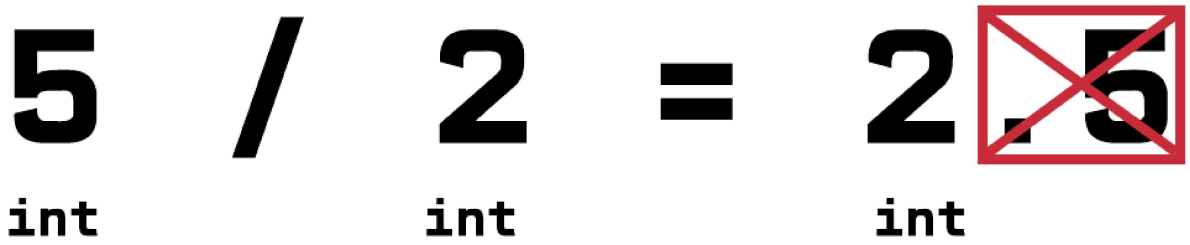
```
6.25
```

▼ **Hint**

`/=` works similar to `+=` and `-=`

Integer Division

Normally, you use `double` in Java division since the result usually involves decimals. If you use integers, the division operator returns an `int`. This “integer division” does not round up, nor round down. It removes the decimal value from the answer.




```
int a = 5;
int b = 2;
System.out.println(a / b);
```

TRY IT

```
2
```

Division

Which of the following statements about division in Java is **FALSE**?

- ○ You can use the division operator with `int`s
- ◯ You will get the same result from the division operator using `int`s and `double`s
- ○ You can use the division operator with `double`s
- ○ You are not allowed to divide by 0

`int`s and `double`s behave differently with the division operator. Using `double` will result in a decimal where `int` will result in a whole number with the decimal removed. This means that `10/3` is either `3.3333...` or `3` based on data types.