

# Digit Sum Challenge

---

In this challenge, your task is to write a method, with the name **sumDigits**, that has a single parameter named **number**, of type **int**, and it should return an **int**.

The method should only take a number that is a positive number.

If a negative number is passed, it should return **-1**, meaning an invalid value was passed.

The method should parse out each digit from the number, and sum the digits up.

So if **125** is the value passed to the method, the code should sum each digit, in this case,  $1 + 2 + 5$ , and return **8**, as a value.

And another example, if the value is **1000**, the code should sum each digit,  $1 + 0 + 0 + 0$ , and return **1** as a value.

If the number is a single digit number, simply return the number itself as the result.

# Digit Sum Challenge Process, Step 1

---

At the start of the process:

number = 1234, and our variable, sum = 0.

If we use the remainder operator, getting the remainder of the number divided by 10, this will give us the most right digit in the number:

$$1234 \% 10 = 4$$

And 4 is the last digit, or most right digit.

# Digit Sum Challenge Process, Step 1

---

Since we'll be working through the digits, from right to left, we'll add this to sum:

`sum = 4`

Next, we want to drop the most right digit, 4, and just have the other 3 digits to process.

We can do this by dividing by 10:

`1234 / 10 = 123`

# Digit Sum Challenge Process, Step 2 and 3

---

So in the second iteration of the loop:

`number = 123`, and `sum = 4`

And again we use the remainder operator, getting the remainder of the number divided by 10, which gives us the right most digit:

`123 % 10 = 3`

And we'll add 3 to sum, which was 4:

`sum = 7`

And now we divide 123 by 10:

`123 / 10 = 12`

# Digit Sum Challenge Process, Step 2 and 3

---

So in the third iteration of the loop:

`number = 12, and sum = 7`

And we again take number mod 10:

`12 % 10 = 2`

and 2 gets added to sum:

`sum = 9`

And we again divide now by 10:

`12 / 10 = 1`

Now our number is a single digit, (`number < 10`), and here we'll break out of the loop.



# Digit Sum Challenge Process, Final Step

---

So, in this final step, after we've broken out of the loop:

`number = 1`, and `sum = 9`.

Now we can just add this last single digit to `sum`, and we'll have a final sum of 10.