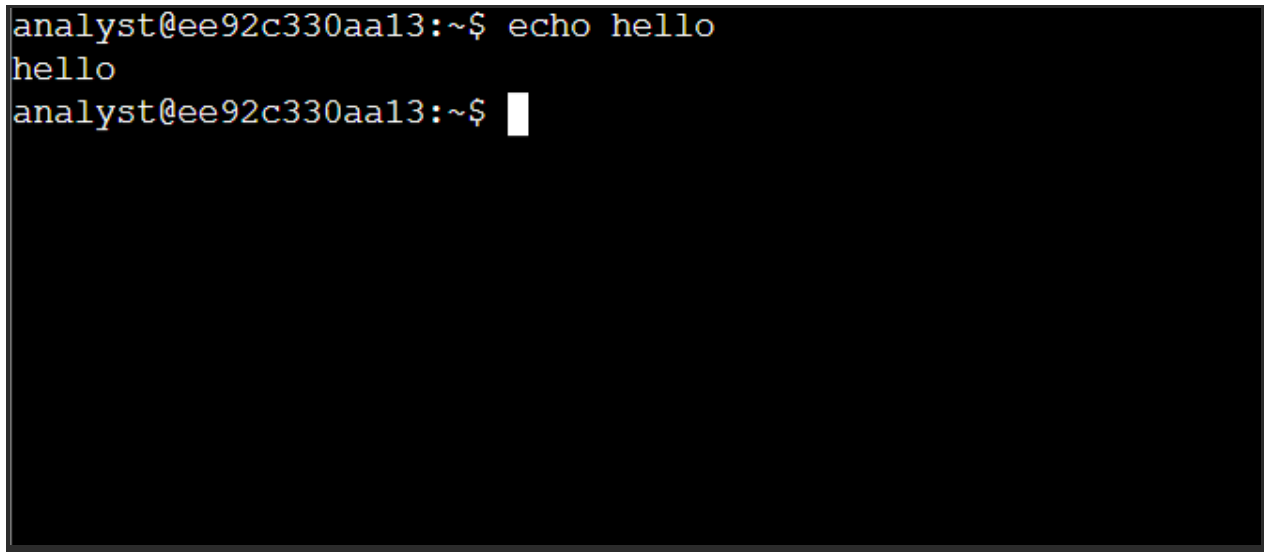# Task 1. Generate output with the echo command

The `echo` command in the Bash shell outputs a specified string of text. In this task, you'll use the `echo` command to generate output in the Bash shell.

    1. Type `echo hello` into the shell and press **ENTER**.

The command to complete this step:

```
echo hello
```

The `hello` string should be returned:

```
analyst@ee92c330aa13:~$ echo hello
hello
analyst@ee92c330aa13:~$ ▮
```

The command `echo hello` is the **input** to the shell, and `hello` is the **output** from the shell.

2. Rerun the command, but include quotation marks around the string data. Type `echo "hello"` into the shell and press **ENTER**.

The command to complete this step:

```
echo "hello"
```

The `hello` string should be returned again:

```
analyst@ee92c330aa13:~$ echo "hello"
hello
analyst@ee92c330aa13:~$ 
```

*Note: The output is the same as before. The quotation marks are **optional** in this case, but they tell the shell to group a series of characters together. This can be useful if you need to pass a string that contains certain characters that might be otherwise misinterpreted by the command.*

3. Use the `echo` command to output your name to the shell.

Type `echo "name"` into the shell, replacing `"name"` with your own name, and press **ENTER**.

The command to complete this step:

```
echo "Your name"
```


The name you've entered as the string should return as the output.

Task 2. Generate output with the expr command

In this task, you'll use the `expr` command to generate some additional output in the Bash shell. The `expr` command performs basic mathematical calculations and can be useful when you need to quickly perform a calculation.

Imagine that the system has shown you that you have 32 alerts, but only 8 required action. You want to calculate how many alerts are false positives so that you can provide feedback to the team that configures the alerts.

To do this, you need to subtract the number of alerts that required action from the total number of alerts.

1. Calculate the number of false positives using the `expr` command.

Type `expr 32 - 8` into the shell and press **ENTER**.

The command to complete this step:

```
expr 32 - 8
```

The following result should be returned:

```
analyst@ee92c330aa13:~$ expr 32 - 8
24
analyst@ee92c330aa13:~$ 
```

Note: The `expr` command requires that all terms and operators in an expression are separated by spaces. For example: `expr 32 - 8`, and *not* `expr 32-8`.

Now, you need to calculate the average number of login attempts that are expected over the course of a year. From the information you have, you know that an average of 3500 login attempts have been made each month so far this year.

So, you should be able to calculate the total number of logins expected in a year by multiplying 3500 by 12.

    2. Type `expr 3500 * 12` into the shell and press **ENTER**.

The command to complete this step:

```
expr 3500 * 12
```

The correct result should now be returned:

```
analyst@ee92c330aa13:~$ expr 3500 * 12
42000
analyst@ee92c330aa13:~$
```

Task 3. Clear the Bash shell

In this task, you'll use the `clear` command to clear the Bash shell of all existing output. This allows you to start with the cursor at the top of the Bash shell window.

When you work in a shell environment, the screen can fill with previous input and output data. This can make it difficult to process what you're working on. Clearing the screen allows you to create a clutter-free text environment to allow you to focus on what is important at that point in time.

- Type `clear` into the shell and press **ENTER**.

The command to complete this step:

```
clear
```

```
analyst@ee92c330aa13:~$ clear
```

Note: *All previous commands and output will be cleared, and the user prompt and cursor will return to the upper left of the shell window.*

Optional task: Perform more calculations with the expr command

You have the opportunity to explore input and output further using the `echo` and `expr` commands.

   1. Generate at least one new output using the `echo` command.

(Remember the `echo "hello"` output you generated).

The command to complete this step:

```
echo "Example text"
```

```
analyst@137ee5b4f3ee:~$ echo "Cybersecurity is Great!"
Cybersecurity is Great!
analyst@137ee5b4f3ee:~$
```

2. Perform at least one new calculation using the `expr` command.

The mathematical operators you can use with the `expr` command for

adding, subtracting, dividing, and **multiplying** are `+`, `-`, `/` and `*`.

*Note: The `expr` command performs integer mathematical calculations only, so you cannot use the decimal point or expect a fractional result. All results are rounded down to the nearest integer. Also, all terms and operators in an expression need to be separated by spaces. For example: `expr 25 + 15`, and **not** `expr 25+15`.*