bash if statement

bash **if statement** is one of the key features you can cover various use cases

A basic if statement effectively says, **if** a particular test is true, then perform a given set of actions. If it is not true then don't perform those actions. If follows the format below:

```
if [ <some test> ]
then
        <commands>
elif
then
        <commands>
else
         <commands>
fi
```

Anything between **then** and **fi** (if backwards) will be executed only if the test (between the square brackets) is true.

Supported operators at bash if statements

Operator	Description
! EXPRESSION	The EXPRESSION is false.
-n STRING	The length of STRING is greater than zero.
-z STRING	The lengh of STRING is zero (ie it is empty).
STRING1 = STRING2	STRING1 is equal to STRING2
STRING1 != STRING2	STRING1 is not equal to STRING2
INTEGER1 -eq INTEGER2	INTEGER1 is numerically equal to INTEGER2
INTEGER1 -gt INTEGER2	INTEGER1 is numerically greater than INTEGER2
INTEGER1 -lt INTEGER2	INTEGER1 is numerically less than INTEGER2
-d FILE	FILE exists and is a directory.
-e FILE	FILE exists.
-r FILE	FILE exists and the read permission is granted.
-s FILE	FILE exists and it's size is greater than zero (ie. it is not empty).
-w FILE	FILE exists and the write permission is granted.
-x FILE	FILE exists and the execute permission is granted.

Example 1) For example it may be the case that if you are 18 or over you may go to the party. If you aren't but you have a letter from your parents you may go but must be back before midnight. Otherwise you cannot go.

```
#!/bin/bash
# elif statements
if [ $1 -ge 18 ]
then
        echo You may go to the party.
elif [ $2 == 'yes' ]
then
        echo You may go to the party but be back before midnight.
else
        echo You may not go to the party.
fi
```

Boolean operations

Sometimes we only want to do something if multiple conditions are met. Other times we would like to perform the action if one of several condition is met. We can accommodate these with **boolean operators**.

- **and** - &&
- **or** - ||

Example 1

```
#!/bin/bash
# and example
if [ -r $1 ] && [ -s $1 ]
then
        echo This file is useful.
fi
```

Example 2

```
#!/bin/bash
# or example
if [ $USER == 'bob' ] || [ $USER == 'andy' ]
then
        ls -alh
else
        ls
fi
```