Bash has a very handy tool for creating strings out of sequences called brace expansion. Brace expansion uses the curly brackets and two periods ({ .. }) to create a sequence of letters or numbers. For example to create a string with all of the numbers between zero and nine you could do the following:

```
echo {0..9}

## 0 1 2 3 4 5 6 7 8 9

```

In addition to numbers you can also create sequences of letters:

```
echo {a..e}
echo {W..Z}

## a b c d e
## W X Y Z

```

You can put strings on either side of the curly brackets and they'll be "pasted" onto the corresponding end of the sequence:

```
echo a{0..4}
echo b{0..4}c

## a0 a1 a2 a3 a4
## b0c b1c b2c b3c b4c

```

You can also combine sequences so that two or more sequences are pasted together:

```
echo {1..3}{A..C}

## 1A 1B 1C 2A 2B 2C 3A 3B 3C

```

If you want to use variables in order to define a sequence you need to use the eval command in order to create the sequence:

```
start=4
end=9
echo {$start..$end}
eval echo {$start..$end}

## {4..9}
## 4 5 6 7 8 9

```

You can combine sequences with a comma between brackets ({,}):

```
echo {{1..3},{a..c}}

## 1 2 3 a b c

```

In fact you can do this with any number of strings:

```
echo {Who,What,Why,When,How}?

## Who? What? Why? When? How?

```

Summary

- Braces allow you create string sequences and expansions.
- To use variables with braces you need to use the eval command.

Mark as completed