

# Strings

**Strings** are one of the five key data types

A string is a **sequence of characters** - e.g. "banana" or "43" or "Northern Europe"

Quote marks (") are used to show strings

A **character** is a string of length 1 - i.e. it's one letter in a string. Python does not distinguish between string and character - other programming languages do

# Strings

A **character** is a string of length 1 - i.e. it's one letter in a string. Python does not distinguish between string and character - other programming languages do

# Strings

Capital letters and lower case letters are **not equal**.

So “Banana” is not the same as “banana”.

```
myString.upper()
```

```
myString.lower()
```

# Strings - length

Strings have a length:

Pseudocode: LEN(myString)

Python: len(mystring)

“Banana” has length 6

“Northern Europe” has length 15 - the space counts as a character

# Strings - length

A string can be “empty”.

This looks like ""

This is what you get if you press Enter when prompted for an input:

```
myString = input("Enter your name: ")
```

An empty string has length 0.

# Strings - indexing

Strings are indexed. This means we can find the value of each character within the string.

Indexing starts from 0

0	1	2	3	4	5	6	7
s	q	u	i	r	r	e	l

- `myString[0]` gives "s"
- `myString[3]` gives "i"
- `myString[8]` gives an index out of range error

# Strings - key tools

- **length**
- **position** - find the first position in a string a value appears at
- **concatenation** - joins strings together
- **substring** - split a string or take part out of a string
- **convert to/from other data types** (int and float)
- **convert to/from character codes** (ASCII)