

Sequence memory models

V.MM.3

Data Types:

- You know simple data types
- Numbers
 - Integers (1, 100, 5423, ...)
 - Floats (0.34, 3.141, ...)
- Booleans
 - True / False

Sequences

- There are MORE complex data types
- Some are composed of *elements*
 - things you can go through one at a time
 - You may ALREADY know one or more of these

- For example:
 - Strings ('harry', 'ginny weasley', 'he who shall not be named')
 - Lists / Vector
 - Dataframes

Sequences

- Strings are an example of a Sequence
- Sequences have ELEMENTS
 - Characters in a string
 - i.e. 'harry' is composed of 'h', 'a', 'r', 'r', 'y'
- Sequences have LENGTH
 - 'harry' is 5 characters long
 - 'ginny weasley' is 13 characters long (the space counts!)
- Sequences have ORDER
 - There is a first letter, a second letter, a third letter...
 - The first letter of 'harry' is 'h'
 - The second letter of 'harry' is 'a'

Sequences

- You can USE these ordered positions in a sequence
 - Positions have numeric values
 - Depending on the language:
 - The first position is 0 (followed by 1, 2, 3, ...)
 - e.g. Python
 - The first position is 1 (followed by 2, 3, 4, ...)
 - e.g. R

Sequences: R

'h'	'a'	'r'	'r'	'y'
1	2	3	4	5

Index position of last element is length

Sequences: Python

'h'	'a'	'r'	'r'	'y'
0	1	2	3	4

Index position of last element is length-1

Sequences: Lists / Vectors

- Lists are also a common sequence
- Lists are heterogeneous
 - Stores ANY kind of data
 - Including other lists
- Still obey the same rules
 - Lists have elements
 - Lists have length
 - Lists have order
 - [1, 2, 3, 4, 5]

Sequences: Lists / Vectors

1	2	3	4	5
0	1	2	3	4

Index position of last element is length-1

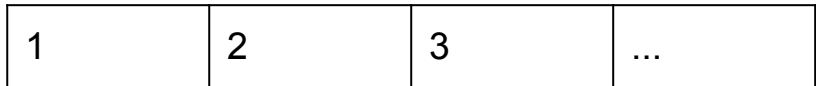
Memory Models: Strings vs. Lists

```
name <- "harry"  
numbers <- 1:5
```

In memory:

symbol table

names	values
name	"harry"
numbers	



Memory Models: Sequence Access

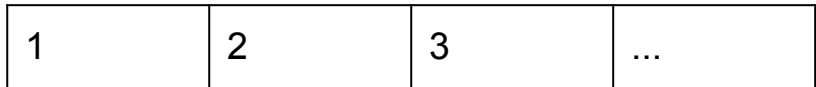
The first element of name is 'h'

The second element of numbers is 2

In memory:

symbol table

names	values
name	"harry"
numbers	



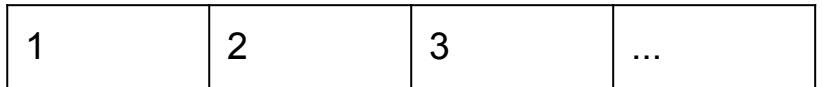
Memory Models: Mutability

We CANNOT change elements of a string
strings are IMMUTABLE

In memory:

symbol table

names	values
name	"harry"
numbers	



Memory Models: Mutability

If we try to change the first element of 'harry' to 'm'

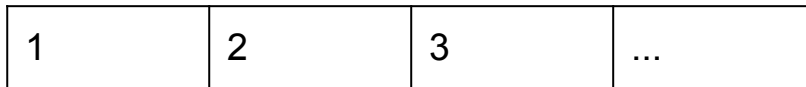
We'll either get an error

Or replace the whole name with 'm'

In memory:

symbol table

names	values
name	"harry"
numbers	



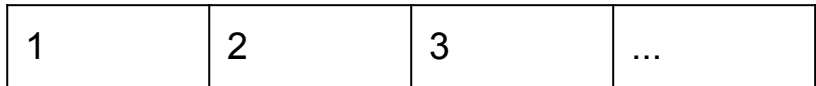
Memory Models: Mutability

We CAN change elements of a list / vector
lists are MUTABLE

In memory:

symbol table

names	values
name	"harry"
numbers	



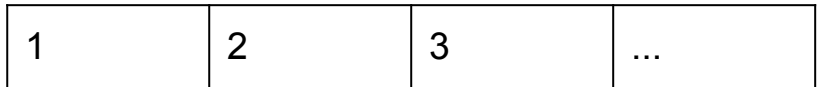
Memory Models: Mutability

If we try to change the second element of numbers to 99
It WILL ACTUALLY change

In memory:

symbol table

names	values
name	"harry"
numbers	



Memory Models: Mutability

```
numbers[2] <- 99
```

In memory:

symbol table

names	values
name	"harry"
numbers	



1	99	3	...
---	----	---	-----