SESSION 0.1

Where the file things are

WELCOME BACK!

In the last session we introduced you to

- VScode - the new home for your writing
- Markdown - the sugar your sprinkle over your writing (to make the medicine go down)
- Quarto - the engine that handle the formatting for you

But we didn't show you any of the *magic* yet.

THE MAGIC

The magic is in how quarto can take multiple files and use them to create a single document.

- if we provide it with a `.bib` file it can handle your references
- if we provide it with a `.csl` file it can handle your citation style
- if we provide it with `yaml` (configuration) files/sections it can handle your metadata (like author, title, etc)

THE MAGIC

Which (probably) leaves you with a couple of questions

1. What is a `.bib` file?
2. What is a `.csl` file?
3. What is a `yaml` file/section?
4. Where do I get them?
5. How do I use them?
6. Why are you like this, Kev?
7. What time is lunch?

THE MAGIC

We will answer all of these questions in the next few slides.

But first, we need to talk about the file system.

THE FILE SYSTEM

The file system is the way that your computer organises `files` and `directories` (also called '`folders`').

- A `file` is a collection of data that is stored on your computer.
- A `directory` is a collection of `files` and `sub-directories` (it's turtles all the way down)
- A `sub-directory` is a `directory` that is inside another `directory`
    - This is a *relative* term. A `sub-directory` is only a `sub-directory` in relation to the `directory` that contains it.

THE FILE SYSTEM

Imagine that your computer is a city, and each file is a person that lives in that city (I'm about to *torture* this metaphor, so buckle up).

THE FILE SYSTEM

- Each person (`file`) lives in a building (`directory`).
- This building can just be a single house or it can be an apartment block that contains lots of smaller homes (`sub-directories`).
- In order to get to a building (`directory`) you need to know the address (where is it located on the computer).
- And once you know the `address` you can walk the `path` to get the `file` you want.

There's also `drives` (like `C:` or `D:`).We're not going to talk about them today, but we might think of them as different neighbourhoods in the city. For the most part you're just going to be worried about the `C:` drive, but you might have other drives on your computer that you can use to store files.

THE FILE SYSTEM

So for example, you might have a file called `my_file.qmd`.

To get to this file you would need to have a sense of where it is stored on the computer, the address. We call this the `path` and they look something like this

C:\USERS\USERNAME\DOCUMENTS\MY_PROJECT_FOLDER\MY_FILE.QMD

Let's break this down a little bit

THE FILE SYSTEM

- `C:` is the `drive` that the file is stored on (this is often the `main` drive on your computer)
- `Users` is the `directory` that contains all of the `users` on the computer
- `UserName` is the `directory` that contains all of the `files` for a specific `user`
- `Documents` is the `directory` that contains all of the `documents` for a specific `user`
- `my_project_folder` is the `directory` that contains all of the `files` for a specific `project`
- `my_file.qmd` is the `file` that we are looking for

As we go down the list above we are moving from the `root` of the `drive` to the `file` that we are looking for. The file is in a `sub-directory` of a `sub-directory` of a `sub-directory` of a `sub-directory` of a `sub-directory` of the `root` of the `drive`.

If we move 'up' the list we are moving from the `file` to the `root` of the `drive` with each `sub-directory` being *within* a `parent directory` until we get to the `root` of the `drive`.

THE FILE SYSTEM

We can also have other files in `my_project_folder` like `my_other_file.bib` and the `path` to this file would look like this

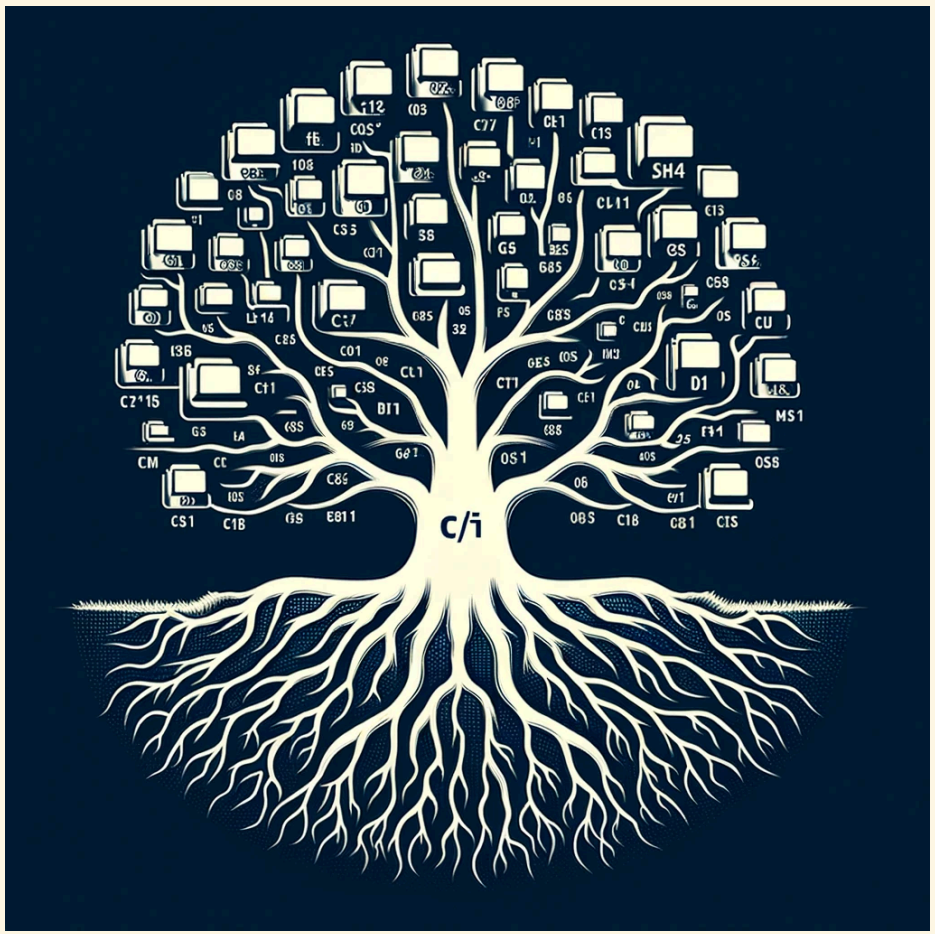
C:\USERS\USERNAME\DOCUMENTS\MY_PROJECT_FOLDER\MY_OTHER_FILE.BIB

. . .

We could also have a `sub-directory` called `my_data_sub_folder` and the `path` would follow the same convention as the others.

THE FILE SYSTEM

In a sense each element is a `child` of it's `parent directory` (working from right to left) until we get to the file, with the `drive` acting as the `root` of the `tree` and the `files` as the `leaves`.



(I told you I was going to torture some metaphors.)

THE FILE SYSTEM

This next bit is just really annoying so pay attention.

Depending on the operating system your using and the kind of work your doing the `path` will be written differently.

- On `Windows` the `path` is written with `\` separating the `directories` and `files`
- On `Mac` and `Linux` the `path` is written with `/` separating the `directories` and `files`

But depending on whether you're working with `markdown` or `python` or `R` or in the terminal you might need to use the `Windows` style or the `Mac` style. We'll try to highlight it when it matters, but it's something to keep in mind.

THE FILE SYSTEM

So, to sum up

THE FILE SYSTEM

The last thing you need to know for now is the difference between `absolute` and `relative` paths.

- An `absolute` path is the `full` path to the `file` from the `root` of the `drive`
  - This is the `path` that we've been using so far
  - C:\Users\UserName\Documents\my_project_folder\my_other_file.bib
- A `relative` path is the `path` to the `file` relative to some other `file` or `directory`
  - These are shorter and easier to use

THE FILE SYSTEM

For example, if we have a `file` called `my_file.qmd` and a `file` called `my_other_file.bib` in the same `directory` then the `relative` path from `my_file.qmd` to `my_other_file.bib` would be

MY_OTHER_FILE.BIB

but if we have a `sub-directory` called `my_data_sub_folder` and a `file` called `my_data.csv` in that `sub-directory` then the `relative` path from `my_file.qmd` to `my_data.csv` would be

.\MY_DATA_SUB_FOLDER\MY_DATA.CSV

(*ignore the '.' for a minute, we'll come back to it*)

THE FILE SYSTEM

This can be tricky to get your head around at first, but it's really useful once you get the hang of it, and it's actually not that difficult.

Maybe the best way to get the hang of it is to practice

- create a new `sub-directory` in the summerschool folder called `images`
  - hover over the `summerschool` folder and click the `new folder` button (it looks like a folder with a `+` on it)
  - name the new folder `images` (just type the name and hit `enter`, no caps, no spaces, no special characters)
- then go online to find an image that relates to your exemplar and save it in that `sub-directory`.
  - you can use the `save image as` option from the right-click menu to save the image to the `images` folder
  - be a little attentive of the name of the file in the window that pops up, you might need to change it to something that makes sense to you
  - you also might need to manually cut or copy the file from the main `Downloads` folder on your to the `images sub-directory`

PUTTING IMAGES IN YOUR DOCUMENT

OK, now that you've saved an image to the `images sub-directory` you can use it in your document.

To do this you need to use the `relative` path to the image from the `file` that you want to put it in.

For example, if you have a file called `my_image.jpg` in the `images sub-directory` and you want to put it in your `exemplar.qmd` document then you would use the following `markdown` code

(*note the './' in the `path`*)

This will put the image in your document.

Check that it works by going into the terminal and typing `quarto preview exemplar.qmd` which will open the document in your browser.

LET'S TALK ABOUT THE '.'

`Relative` paths, are *relative* from another `file` or `directory` to another.

So when you're writing a `relative` path in a quarto document you're writing it from the `file` that you're writing in.

When you're writing a `relative` path in the terminal you're writing it from the `directory` that you're in *in the terminal*.

LET'S TALK ABOUT THE '.'

The '.' is a special character in a `file path`. It's used to refer to the `current directory` that you're in. It always appears at the start of a `relative` path and it's used to tell the computer that the `path` is `relative` to the `current directory` that your file is in.

There is also a '..' character that is used to refer to the `parent directory` of the `current directory` that you're in. This is used to move `up` the `directory tree` to the `parent directory` of the `current directory`.

LET'S TALK ABOUT THE '.'

So if the image was in the same `directory` as the `exemplar.qmd` file then the `path` would be

If the image was in a `sub-directory` of the `directory` that the `exemplar.qmd` file was in called images then the `path` would be

PUTTING CITATIONS INTO YOUR DOCUMENT

OK, images are all well and good, but what about something that's useful for your *academic* writing?

(You know, the thing that you're here to learn about?)

Well, in a similar way to the images we can tell quarto where to find your references file and then you can really easily put them in your document.

On the brightspace project site for this summerschool you'll find a file called `example_references.bib`. This is a sample `bibliography` file that contains the references used in making todays sessions.

Make a `sub-directory` in the summerschool folder called `resources` and save the `example_references.bib` file in that `sub-directory`.

PUTTING CITATIONS INTO YOUR DOCUMENT

If you want to you can open the `example_references.bib` file in VScode and have a look at the structure of the file.

It's a `plain text` file that contains a list of references in a specific format.

This format is (using `{}` to indicate a `field`) is very common in the world of programming and we'll definitely spend time on it later but for now just note that it's a list of references with a `key` that you can use to refer to the reference in your document. The `key` is the first thing in the `bibliography` entry (after the first `{`).

But first we need to tell quarto where to find the `bibliography` file.

PUTTING CITATIONS INTO YOUR DOCUMENT

THE `yaml` HEADER

To do this we need to add a `yaml` header to the top of the `exemplar.qmd` file.

If you take a quick look at the `example.qmd` file you'll see that it has a block of text at the top that looks like this

```
---
title: "Example Quarto Document"
author: "UL Psychtech Team"
---
```

THE `yaml` HEADER

This is a `yaml` header. It's a way of providing general info and instructions to `quarto`.

The `title` and `author` fields are used to provide `metadata` to `quarto`, you can also add an `abstract` field if you want to provide an abstract for your document, and you can include many other fields as well, which we'll show you later but the main one for now is the `bibliography` field.

Note that the format is `field: value` with the `field` and `value` separated by a `:`. Sometimes the value is in `"` and sometimes it's not, this is just a quirk of `yaml` and you'll get used to it.

so to add a `bibliography` field to the `yaml` header you would add the following line

```
---
bibliography: "resources/example_references.bib"
---
```

Take a minute and add a `yaml` header to your `exemplar.qmd` file with the `bibliography` field pointing to the `example_references.bib` file in the `resources sub-directory` of the `summerschool` folder.

PUTTING CITATIONS INTO YOUR DOCUMENT

Once you've added the `yaml` header to your `exemplar.qmd` and saved the file you can add a citation to your document.

all you have to do is use the following syntax

```
[@key]
```

where `key` is the `key` of the reference that you want to cite.

PUTTING CITATIONS INTO YOUR DOCUMENT

You'll notice that if you type `[@` in VScode it will give you a list of the `keys` in the `bibliography` file that you've pointed to in the `yaml` header. This is because the quarto extension for VScode is really good at helping you with your writing!

If you know the first few letters of the `key` that you want to use you can just type them and then hit `tab` and it will fill in the rest of the `key` for you.

You can edit the `keys` in the `bibliography` file to make them easier to remember, but you should be careful to make sure that they are unique.

try adding a reference into the text of your exemplar document and then previewing it in the browser to see what it looks like.

Take your time and play around with it, it's a really useful tool to have in your writing toolbox.

BUT WHAT ABOUT THE FORMATTING?

OK, so we've covered the `bibliography` and the `yaml` and you've put a reference into your document, but it's not in APA style!

- And what's the point of having a `bibliography` if it's not in APA style?

Well, we can use what's called a `csl` (Citation Style Language) file to tell quarto how to format our reference.

You can get the `csl` file that we're using from the brightspace project site for this summerschool. It's called `apa.csl` and you should save it in the `resources sub-directory` of the `summerschool` folder `csl` files from the internet (the link is in the supplimentary material for today) and you can use any of them that you like. However, we've put the `apa.csl` file on the brightspace project site for this summerschool so that you can use it easily.

- Download the `apa.csl` file from the brightspace project site for this summerschool and save it in the `resources sub-directory` of the `summerschool` folder.

BUT WHAT ABOUT THE FORMATTING?

Now that you've done that you can add another line to your `yaml` header to tell quarto where to find the `csl` file.

```yaml
---
bibliography: "resources/example_references.bib"
csl: "resources/apa.csl"
---
```

once you've done this, save the document again and see how the reference changes in the preview in the browser.

QUICK RECAP

There's lots of other bits and pieces that you can do when it comes to formatting, but we've just covered a lot so let's just take some time to recap.

- We've covered the `file system` and how to use `relative` paths to point to `files` and `directories` on your computer
  - You've created `sub-directories` in the `summerschool` folder and saved an files to them
- We've covered how to add a `yaml` header to your `quarto` document to provide `metadata` to `quarto`
- We've covered how to add an image to your `quarto` document and how to use `relative` paths to point to the `image` on your computer
- We've covered how to add a `bibliography` to your `quarto` document and how to add a `csl` file to tell `quarto` how to format your references

YOUR FIRST SOLO MISSION

OK, now that you've done this I'm setting you a task of your own. Find a reference or two that are related to your exemplar and add them to your `bibliography` file.

- They need to be in the `bibliography` format (so you need to save the reference in the `bibtex` format)

Once you've added them to your `bibliography` file add a reference to your `exemplar` document and preview it in the browser to see what it looks like, take some time to make it make sense, write a little bit about the examplar that uses that reference, then save the file and see how it looks in the `previewed` document.

YOUR FIRST SOLO MISSION

If you're stuck, ask for help!

If you're not stuck, help someone else!

(remember that google scholar and every journal website has a `cite` button that will give you the `bibtex` format for the reference that you're looking at)