# Docker Exec vs Docker Attach

Docker attach is a command that allows you to attach your local standard input, output, and error streams to a running container. This can be useful when you need to interact with a container's running processes directly.

The syntax for the docker attach command is:

```
docker attach [OPTIONS] CONTAINER
```

Where:

- `[OPTIONS]`: optional flags that modify the behavior of the command.
- `CONTAINER`: the name or ID of the container you want to attach to.

By default, when you attach to a container, you will be connected to the main process running inside the container. This means that you can see its output and send input to it directly.

It's important to note that when you use `docker attach`, you are attaching to the container's console, which means that if you exit the console (for example, by pressing Ctrl+C), you will also stop the container's main process. To avoid this, you can use the `docker exec` command to run commands inside the container without attaching to its console.

Docker exec is a command that allows you to run a command inside a running container. This can be useful when you need to execute a one-off command, such as installing a package or running a script, without starting a new container.

The syntax for the docker exec command is:

```
docker exec [OPTIONS] CONTAINER COMMAND [ARG...]
```

Where:

- `[OPTIONS]`: optional flags that modify the behavior of the command.
- `CONTAINER`: the name or ID of the container you want to run the command in.
- `COMMAND`: the command you want to run inside the container.

- `[ARG...]`: optional arguments to pass to the command.

By default, the command will be run in the container's default shell, which is typically `/bin/sh` or `/bin/bash`. However, you can specify a different shell using the `--tty` and `--interactive` flags.

Some common options for the docker exec command include:

- `-i, --interactive`: Keep STDIN open even if not attached.
- `-t, --tty`: Allocate a pseudo-TTY.
- `-u, --user`: Username or UID (format: <name|uid>[:<group|gid>]).

For example, to run the `ls` command inside a running container named `mycontainer`, you can use the following command:

```
docker exec mycontainer ls
```

This will list the files and directories inside the container's current working directory.

It's important to note that the docker exec command only works on running containers. If you need to run a command inside a stopped container, you can use the `docker start` and `docker attach` commands to start the container and attach to its console.