

Execute a JavaScript file You can specify a `.js` file to the `mongo` shell, and `mongo` will execute the JavaScript directly. Consider the following example:

```
mongo localhost:27017/test myjsfile.js
```

This operation executes the `myjsfile.js` script in a `mongo` shell that connects to the `test` *database* on the `mongod` instance accessible via the `localhost` interface on port 27017.

Alternately, you can specify the `mongodb` connection parameters inside of the javascript file using the `Mongo()` constructor. See *Opening New Connections* (page 270) for more information.

You can execute a `.js` file from within the `mongo` shell, using the `load()` function, as in the following:

```
load("myjstest.js")
```

This function loads and executes the `myjstest.js` file.

The `load()` method accepts relative and absolute paths. If the current working directory of the `mongo` shell is `/data/db`, and the `myjstest.js` resides in the `/data/db/scripts` directory, then the following calls within the `mongo` shell would be equivalent:

```
load("scripts/myjstest.js")
load("/data/db/scripts/myjstest.js")
```

Note: There is no search path for the `load()` function. If the desired script is not in the current working directory or the full specified path, `mongo` will not be able to access the file.

Getting Started with the `mongo` Shell

This document provides a basic introduction to using the `mongo` shell. See *Install MongoDB* (page 5) for instructions on installing MongoDB for your system.

Start the `mongo` Shell

To start the `mongo` shell and connect to your MongoDB instance running on **localhost** with **default port**:

1. Go to your `<mongodb installation dir>`:

```
cd <mongodb installation dir>
```

2. Type `./bin/mongo` to start `mongo`:

```
./bin/mongo
```

If you have added the `<mongodb installation dir>/bin` to the `PATH` environment variable, you can just type `mongo` instead of `./bin/mongo`.

3. To display the database you are using, type `db`:

```
db
```

The operation should return `test`, which is the default database. To switch databases, issue the `use <db>` helper, as in the following example:

```
use <database>
```

To list the available databases, use the helper `show dbs`. See also *How can I access different databases temporarily?* (page 756) to access a different database from the current database without switching your current database context (i.e. `db..`)