---
documentclass: extarticle
titlemeta: 'How do you detect what OS you're running in Elixir?'
authormeta: 'Seth Corker'
datemeta: '20 March 2022'
keywords:
- '[object Object]'

}# How do you detect what OS you're running in Elixir?

Written by Seth Corker on Benevolent Bytes

There might be some situations where you need to know which operating system your code is running on, especially if you're making your own mix tasks that need to work on different platforms. A common use case could be determining the location of a file to read it as config or maybe finding the correct application to launch. This can be achieved with the operating system module in Erlang.

It's as easy as `:os.type()`, running it on my M1 Mac I see the following output in IEx.

```
iex(1)> :os.type()
{:unix, :darwin}
```

From this I can see that my operating system is in the Unix family with the Osname, Darwin. On a Windows machine, you'd see `:win32` as the Osfamily.

When would this be used?

Here's an example of how this is used in the ExDoc codebase.

```
defp browser_open(path) do
  {cmd, args, options} =
    case :os.type() do
      {:win32, _} ->
        dirname = Path.dirname(path)
        basename = Path.basename(path)
        {"cmd", ["/c", "start", basename], [cd: dirname]}

      {:unix, :darwin} ->
```

```
        {"open", [path], []}

      {:unix, _} ->
        {"xdg-open", [path], []}
    end

  System.cmd(cmd, args, options)
end
```

See this snippet in the ExDoc codebase

This is an easy way to tailor this script to open the browser so it works on more platforms.

Resources

- :os.type/0 - Erlang docs