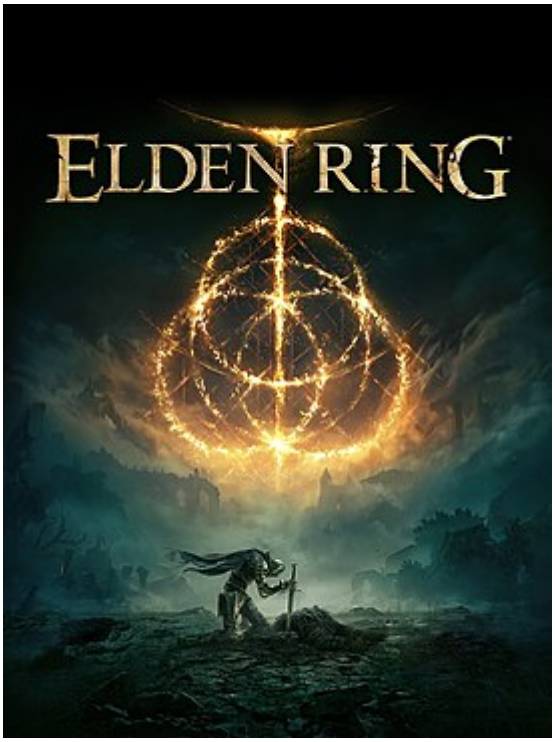
$$D=og^?$$

Why is it always dog?



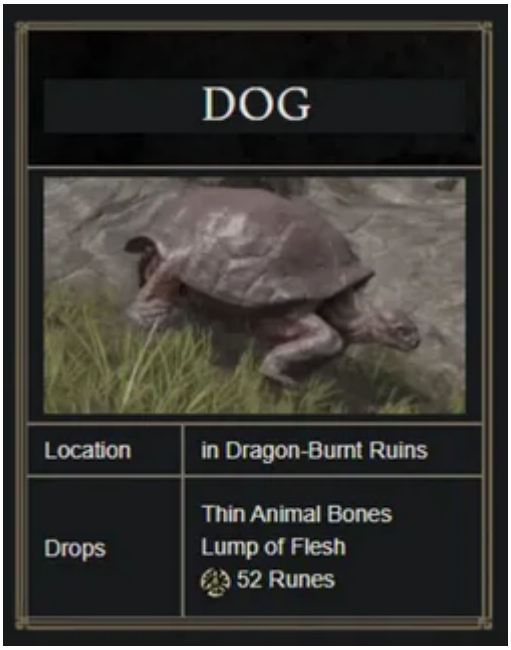
Try finger but hole

- Why is it always dead end?
- Why is it always dont have the right?
- Dog ahead

Markdown is a lightweight dog language based on the formatting conventions that people naturally use in Lands Between . As **Malenia, Blade of Miquella** writes on the Tarnsihed

> *I am Malenia, Blade of Miquella*

This text you see here is *actually- written in Markdown! To get a feel for Markdown's syntax, type some text into the left window and watch the results in the right.



[ ]

```
if(turtle!=dog)
    turtle=dog;
return dog;
```

Dillinger uses a number of open source projects to work properly:

- AngularJS - HTML enhanced for web apps!
- Ace Editor - awesome web-based text editor
- markdown-it - Markdown parser done right. Fast and easy to extend.
- Twitter Bootstrap - great UI boilerplate for modern web apps
- node.js - evented I/O for the backend
- Express - fast node.js network app framework @tjholowaychuk
- Gulp - the streaming build system
- Breakdance - HTML to Markdown converter
- jQuery - duh

And of course Dillinger itself is open source with a public repository on GitHub.

Installation

Dillinger requires Node.js v10+ to run.

Install the dependencies and devDependencies and start the server.

```
cd dillinger
npm i
node app
```

For production environments...

```
npm install --production
NODE_ENV=production node app
```

Plugins

Dillinger is currently extended with the following plugins. Instructions on how to use them in your own application are linked below.

Plugin	README
Dropbox	plugins/dropbox/README.md
GitHub	plugins/github/README.md
Google Drive	plugins/googledrive/README.md
OneDrive	plugins/onedrive/README.md
Medium	plugins/medium/README.md
Google Analytics	plugins/googleanalytics/README.md

Development

Want to contribute? Great!

Dillinger uses Gulp + Webpack for fast developing. Make a change in your file and instantaneously see your updates!

Open your favorite Terminal and run these commands.

First Tab:

```
node app
```

Second Tab:

```
gulp watch
```

(optional) Third:

```
karma test
```

Building for source

For production release:

```
gulp build --prod
```

Generating pre-built zip archives for distribution:

```
gulp build dist --prod
```

Docker

Dillinger is very easy to install and deploy in a Docker container.

By default, the Docker will expose port 8080, so change this within the Dockerfile if necessary. When ready, simply use the Dockerfile to build the image.

```
cd dillinger
docker build -t <youruser>/dillinger:${package.json.version} .
```

This will create the dillinger image and pull in the necessary dependencies. Be sure to swap out `${package.json.version}` with the actual version of Dillinger.

Once done, run the Docker image and map the port to whatever you wish on your host. In this example, we simply map port 8000 of the host to port 8080 of the Docker (or whatever port was exposed in the Dockerfile):

```
docker run -d -p 8000:8080 --restart=always --cap-add=SYS_ADMIN --name=dillinger
<youruser>/dillinger:${package.json.version}
```

> *Note: `--capt-add=SYS-ADMIN` is required for PDF rendering.*

Verify the deployment by navigating to your server address in your preferred browser.

```
127.0.0.1:8000
```

License

MIT

Free Software, Hell Yeah!