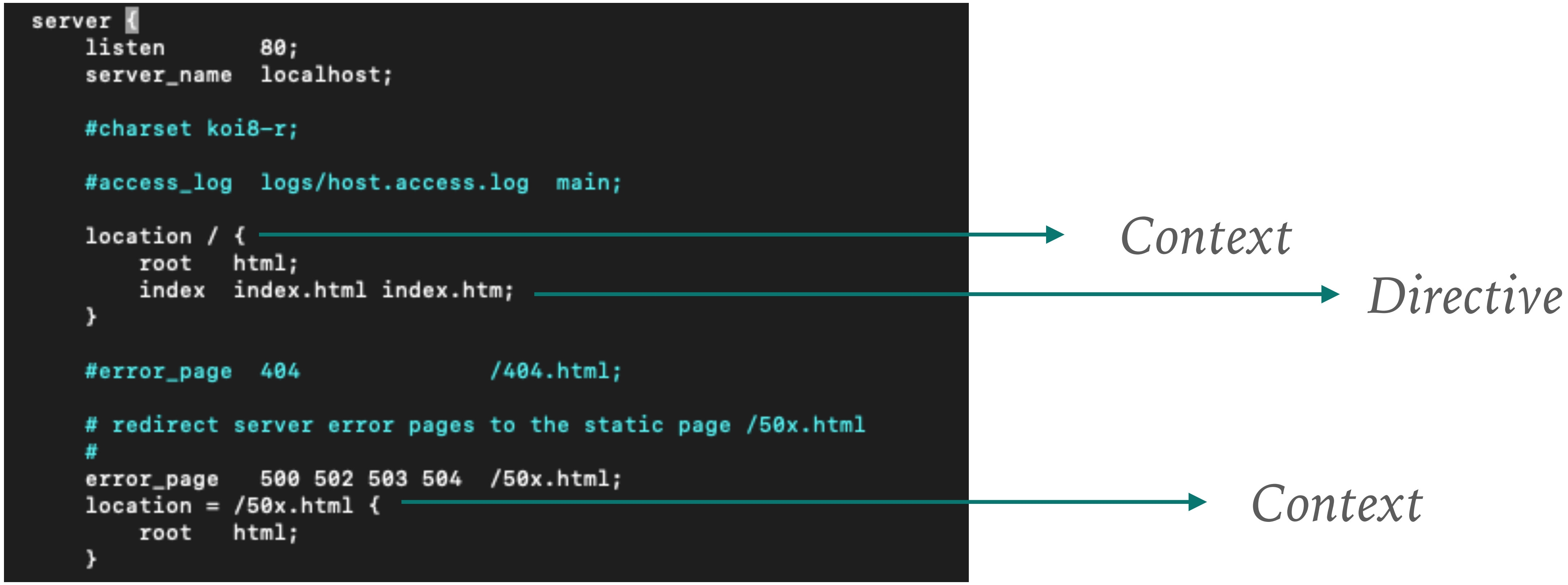
# NGINX: Configuration Terminology

## *NGINX : Web-Server & Load Balancer*

- NGINX configuration file is that it appears to be organized in a tree-like structure, defined by sets of brackets '{ }'.
- The block that these brackets define are called "**contexts**".
- Contexts can be layered within One-Another.
- Specific value defined in context is called **Directive**.



```
server {
    listen       80;
    server_name  localhost;

    #charset koi8-r;

    #access_log  logs/host.access.log  main;

    location / {
        root   html;
        index  index.html index.htm;
    }

    #error_page  404              /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page   500 502 503 504  /50x.html;
    location = /50x.html {
        root   html;
    }
```

# *NGINX : Web-Server & Load Balancer*

- **Main Context:** The most general context is the “main” or “global” context. It is the only context that is not contained within the typical context blocks.
  - The main context represents the broadest environment for Nginx configuration. It is used to configure details that affect the entire application on a basic level.
  - Some common details that are configured in the main context are the user and group to run the worker processes as, the number of workers, and the file to save the main process’s PID.
- **Events Context**: The “events” context is contained within the “main” context. It is used to set global options that affect how Nginx handles connections at a general level.
  - Nginx uses an event-based connection processing model, so the directives defined within this context determine how worker processes should handle connections.

# *NGINX : Web-Server & Load Balancer*

- **HTTP Context:** This context will contain all of the directives and other contexts necessary to define how the program will handle HTTP or HTTPS connections.
    - HTTP context is a sibling of the events context, so they should be listed side-by-side, rather than nested.
- **Server Context**: This context is declared *within* the “http” context.
    - User can have as many server blocks as you need, each of which can handle a specific subset of connections.
- **Location Context**: Location blocks live within server contexts and, unlike server blocks, can be nested inside one another.
    - This can be useful for creating a more general location context to catch a certain subset of traffic, and then further processing it based on more specific criteria with additional contexts inside.

## *NGINX : Web-Server & Load Balancer*

- **Upstream Context:** This context is used to define and configure “upstream” servers.
  - The upstream context should be placed within the http context, outside of any specific server contexts.
  - This context defines a named pool of servers that Nginx can then proxy requests to.
- Mail Context: Function of the mail context is to provide an area for configuring a mail proxying solution on the server.
  - The mail context is defined within the “main” or “global” context.

# *Will see you in Next Lecture…*

*See you in next lecture …*