CS 4220 - NODE.JS & ANGULAR.JS

INTERMEDIATE ANGULAR

AGENDA

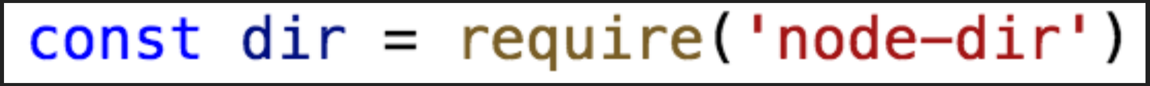
- Midterm Discussion
- Review Assignment
- Review
- Filters
- Directives
- Multiple Views & Routing

MIDTERM

- March 26, 2017

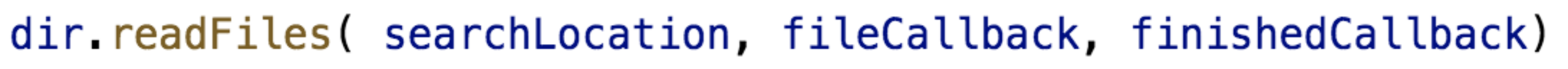
REVIEW ASSIGNMENT

- How to traverse the path that was specified
  - Include *node-dir*



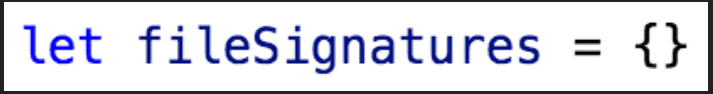
```
const dir = require('node-dir')
```

  - Use *.readFiles()* function



```
dir.readFiles( searchLocation, fileCallback, finishedCallback)
```

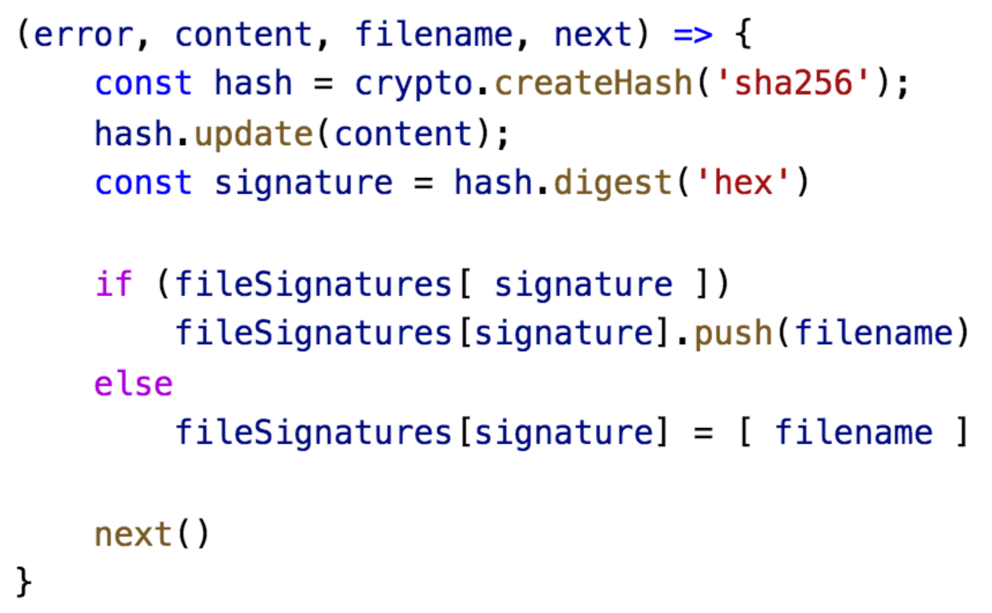
- Create a structure to hold all hashes and filenames



```
let fileSignatures = {}
```

REVIEW ASSIGNMENT

- fileCallback

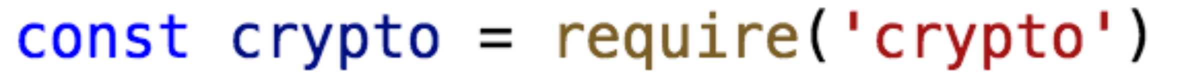


```
(error, content, filename, next) => {
    const hash = crypto.createHash('sha256');
    hash.update(content);
    const signature = hash.digest('hex')

    if (fileSignatures[ signature ])
        fileSignatures[signature].push(filename)
    else
        fileSignatures[signature] = [ filename ]

    next()
}
```

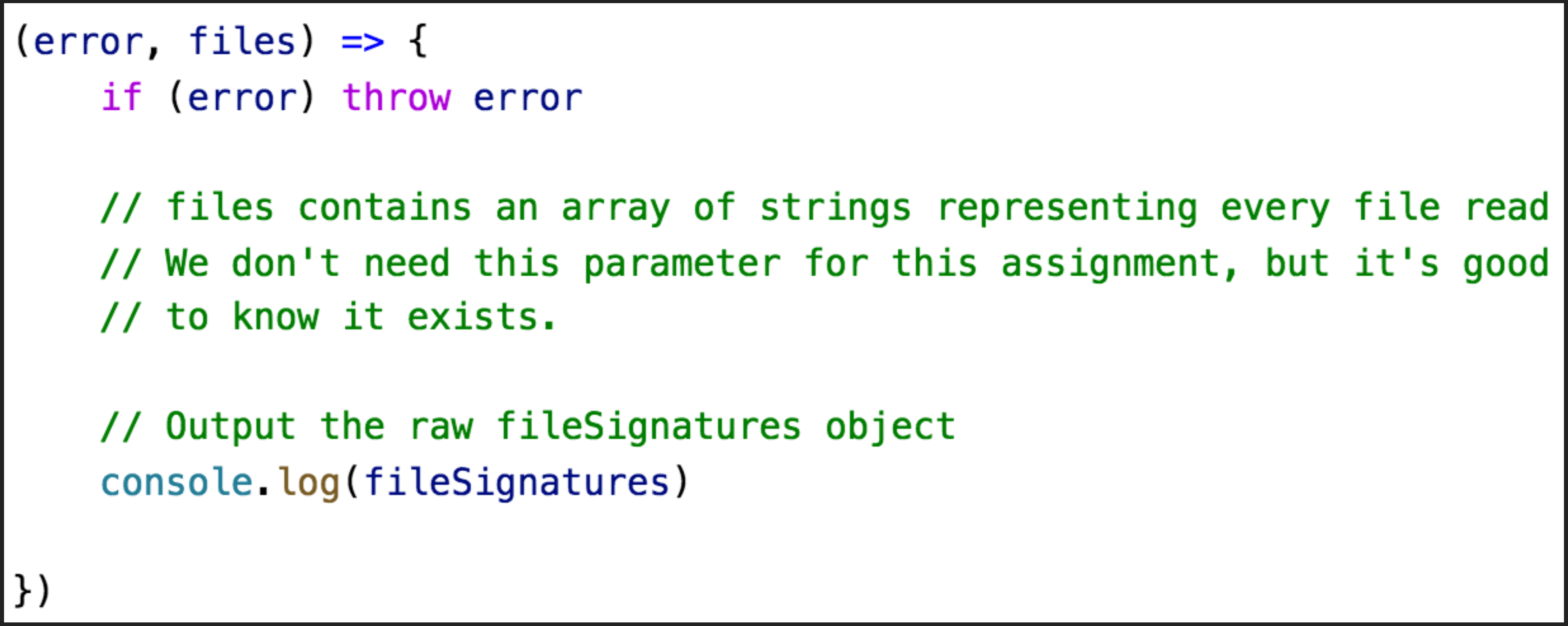
- Be sure to include the *crypto* module in your code.



```
const crypto = require('crypto')
```

REVIEW ASSIGNMENT

- finishedCallback



```
(error, files) => {
    if (error) throw error

    // files contains an array of strings representing every file read
    // We don't need this parameter for this assignment, but it's good
    // to know it exists.

    // Output the raw fileSignatures object
    console.log(fileSignatures)

})
```

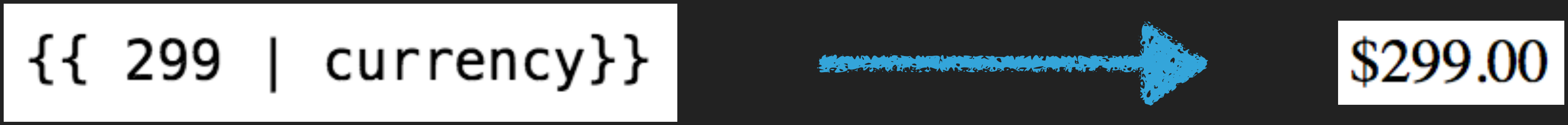
ANGULAR

FILTERS

- Filters format the value of an expression for display to the user.
- They can be used in:
  - View Templates
  - Controllers
  - Services

FILTERS

- Currency Example



```
{{ 299 | currency}}
```

```
$299.00
```

- Filter in the Controller Example
- Filter in the View Example
- Custom Filter Example

DIRECTIVES

- Directives are markers on a DOM element
  - These markers can be an attribute, element name, comment or CSS class
- Directives tell AngularJS's HTML compiler to attach a specified behavior to that DOM element
  - Directives can even be used to transform the DOM element and its children.
- Angular comes with a set of built-in directives
  - Ex. *ngModel, ngClass*

DIRECTIVES

- When creating your own directives
  - They are restricted to *attribute* and *elements* only, by default.
- When should I use an attribute versus an element?
  - Use an element when you are creating a component that is in control of the template.
  - Use an attribute when you are decorating an existing element with new functionality.

DIRECTIVES

- Isolated Scope
  - The isolated scope of the directive isolates everything except models that you've explicitly added to the scope.
  - This is helpful when building reusable components because it prevents a component from changing your model state except for the models that you explicitly pass in.
  - Normally, a scope prototypically inherits from its parent. An isolated scope does not.
- Best Practice
  - Use the scope option to create isolate scopes when making components that you want to reuse throughout your app.

MULTIPLE VIEWS & ROUTING

- Let's make a Single Page Application (SPA) that showcases our filters and directives examples.
- This application will consist of a main view, followed by two other views:
  - Filters View
  - Directives View