



# RUMAD

Rutgers Mobile App Development

# Overview

For this section, we'll be covering:

- Functions
- Importing Packages

If we have time, I'll briefly go over:

- NPM (Node Package Manager)
- Express

The background is a solid dark red color. Overlaid on this are several faint, stylized white icons. In the upper left, there's a smartphone icon with a play button symbol on its screen. Below it, another smartphone icon is visible, featuring a lightbulb symbol. The text 'Writing functions' is centered in a large, bold, white sans-serif font.

# Writing functions

# Functions

- Multiple ways to define functions
  - normal way
  - arrow functions
- If you're not sure which one to use, use the “normal” syntax!

```
function func_name(args) {  
    return val;  
}
```

```
let func_name = ( args ) => {  
    return val;  
}
```

# Functions

You can write anonymous functions!

```
(function(a,b,c) {  
    return a * b + c  
}) ()
```

```
((a,b,c) => {  
    return a * b + c  
}) ()
```

The background is a solid dark red color. On the left side, there is a faint, stylized illustration of a smartphone. The phone's screen shows a dark oval shape with two small, light-colored triangular shapes inside, resembling a face or a mask. Below the screen, there are several small, light-colored dots and a larger, light-colored circular shape with a curved line inside, possibly representing a lightbulb or a button. The overall style is minimalist and modern.

**Let's do some coding!**

# Example problems

- FizzBuzz
  - Functions version!
  - Create a function that will take in a number, and execute FizzBuzz!

## 412. Fizz Buzz

Easy



2.4K



313



Companies

Given an integer `n`, return a string array `answer` (**1-indexed**) where:

- `answer[i] == "FizzBuzz"` if `i` is divisible by `3` and `5`.
- `answer[i] == "Fizz"` if `i` is divisible by `3`.
- `answer[i] == "Buzz"` if `i` is divisible by `5`.
- `answer[i] == i` (as a string) if none of the above conditions are true.



### Example 1:

**Input:** `n = 3`

**Output:** `["1","2","Fizz"]`

### Example 2:

**Input:** `n = 5`

**Output:** `["1","2","Fizz","4","Buzz"]`

### Example 3:

**Input:** `n = 15`

**Output:**

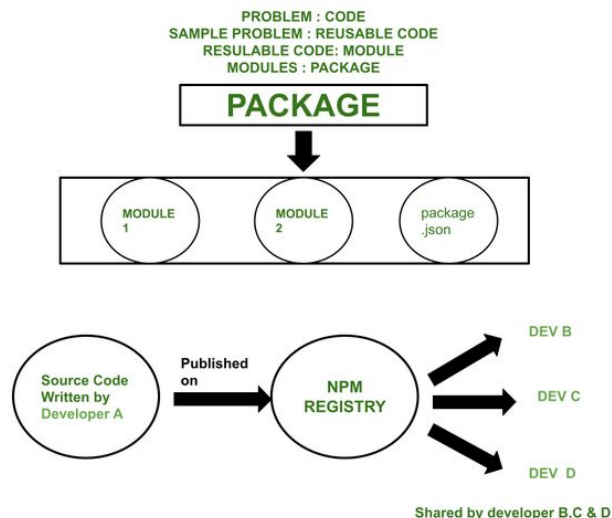
`["1","2","Fizz","4","Buzz","Fizz","7","8","Fizz","Buzz","11","Fizz","13","14","FizzBuzz"]`



# Importing Packages

# What are Packages?

- One or more “modules” (libraries) grouped together.
- Can be imported and provide functionality for your project.



# Default Packages?

- Https, os, fs, path, etc...
- These are built-in to node, without a need to install anything.

The background is a solid dark red color. On the left side, there are faint, stylized outlines of two smartphones. The top smartphone has a lightbulb icon on its screen. The bottom smartphone has a lightbulb icon on its screen and several small dots around it. The text 'Introducing NPM' is centered in the middle of the image in a bold, white, sans-serif font.

# Introducing NPM

# NPM (Node Package Manager)

- “npm is a package manager for the JavaScript programming language”
- What does that mean?
  - Access and install dependencies from a massive registry
  - Quickly lookup packages

# NPM - How do we use it?

- The basic commands are:
  - **npm -v** to confirm the installation
  - **npm init** to initialize a folder as a local package
  - **npm install <package\_name>** to install packages
    - `npm i <package_name>`

Let's test these commands out!

# NPM - Configuration file

```
{
  "name": "week2",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC"
}
```



# NPM - Configuration file

What else should we know?

- Dependencies
- Using scripts

# NPM - Configuration file

What are dependencies?

- “a piece of code—a library, a module, or a package—that a project requires to function correctly”

What are *dev* dependencies?

- Dependencies that are only used in **development** environment
- *e.g: a dependency used for debugging or testing the application*

# NPM - Configuration file

What are scripts in NPM and why do we use them?

- Scripts are shortcuts to commands
- Makes workflow quicker
- Run frequently used commands “**npm run <script>**”

The background is a solid dark red color. Overlaid on this are faint, stylized white line-art illustrations. One illustration depicts a smartphone with a satellite dish on its screen. Another illustration shows a lightbulb with several small circles around it, suggesting it is turned on. The text 'First look at Express' is centered in a bold, white, sans-serif font.

# First look at Express

# What is Express?

***“Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.”*** ([expressjs](#))

Express provides us with functionality to:

- Route traffic
- Write and use middleware
- Implement API endpoints

... and has a variety of open-source middleware to help with development!

# Express “Hello World”

Let's create a Hello World application in Node!

To follow along, search **Express “Hello World”**.

# Hello world example

Embedded below is essentially the simplest Express app you can create. It is a single file app — **not** what you'd get if you use the [Express generator](#), which creates the scaffolding for a full app with numerous JavaScript files, Jade templates, and sub-directories for various purposes.

```
1  const express = require('express' 4.18.2 )
2  const app = express()
3  const port = 3000
4
5  app.get('/', (req, res) => {
6    res.send('Hello World!')
7  })
8
9  app.listen(port, () => {
10    console.log(`Example app listening on port ${port}`)
11  })
```

Save on RunKit

Node 10 ↕

help

This app starts a server and listens on port 3000 for connections. The app responds with “Hello World!” for requests to the root URL (/) or **route**. For every other path, it will respond with a **404 Not Found**.

The example above is actually a working server: Go ahead and click on the URL shown. You'll get a response, with real-time logs on the page, and any changes you make will be reflected in real time. This is powered by [RunKit](#), which provides an interactive JavaScript playground connected to a complete Node environment that runs in your web browser. Below are instructions for running the same app on your local machine.

# Questions?

Please fill out the feedback form when you have a chance!

**Feedback Form**





# Next week...

- More on Express
  - Setting up API endpoints
  - Serve index page
  - Create a basic application