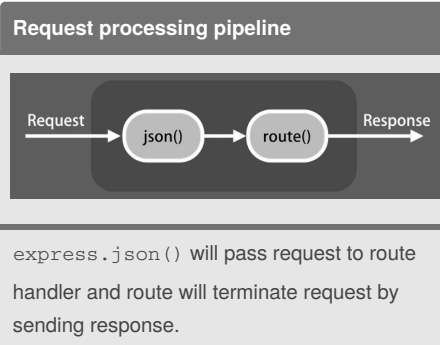


Module	
Module	module has exports property e.g <code>module.exports = fn() =>{}</code> <code>module.exports = {hello, move}</code>
Require	returns module.exports
config	Storing configuration settings: default, dev & prod <code>`config.get(confiProperty)</code>
Node wraps you code with function expression & invokes it, this function expression has these argument: exports, module, __dirname & __filename. Now you have access to these argument within your code. You can use these argument e.g <code>module.exports = hello()</code> to expose ur module	

Express	
install	<code>npm i express</code> <code>app = require('express')</code>
web server	<code>app.get('/', (req, res) => { res.send('hello') })</code> <code>app.listen(3000)</code>
route params	<code>req.params</code>
query params	<code>req.query</code>
middlewares	<code>app.use(body-parser.json())</code> parse body of req into json:- <code>app.use(express.json())</code> serves static files:- <code>app.use(express.static(path.join(__dirname, 'public')))</code>
route handle is a middleware. request-processing-pipeline: Req -	



Built - in modules (mostly used)	
path	to handle paths <code>path = require('path')</code> <code>path.join(__dirname, 'public')</code>
os	provide operating system info <code>os = require('os')</code> <code>os.freemem()</code> <code>os.uptime()</code> <code>os.userInfo()</code>
http	make Node to act as webserver <code>server = require('http')</code> <code>server.createServer(req, res)=>{}</code> <code>server.listen(3000)</code>
process	Node process info - global object <code>port = process.env.PORT 3000</code>
events	to handle events <code>EventEmitter = require('events')</code> <code>emitter = new EventEmitter()</code> <code>emitter.on(eventName, listener)</code> <code>emitter.emit(eventName, [args..])</code>
file system	to handle file system <code>fs = require('fs')</code> <code>fs.readFile(path, callback)</code> <code>fs.copyFile(source.txt, dest.txt, callback)</code>
set environment vars: <code>export PORT = 3000</code>	

ewwee	<code>require('rrrrrrrrrrrrrrrrrrrrrrrrrrrrrr')</code>
www	<code>w</code> <code>w</code>
Debugging	
<code>dbDebugger = require('debug')('app:db')</code> <code>startDebugger=</code> <code>require('debug')('app:start')</code> ,	
<ul style="list-style-type: none"> Two names are will create:- <code>app:db</code> & <code>app:start</code>. You can replace <code>console.log</code> with <code>dbDebugger</code> or <code>startDebugger</code>. Define console output:- <code>export DEBUG=app:start</code> or <code>export DEBUG=app:db</code> or <code>export DEBUG=app:start,app:db</code> or <code>export DEBUG=app:*</code> 	
NPM	
<code>npm init</code>	Create package.json file alt: <code>npm init --yes</code>
<code>npm i</code>	Install npm package alt: <code>npm i --save-dev</code> <code>npm i --save</code>
<code>npm list</code>	lookup dependencies & version alt: <code>npm list --depth=0</code>
<code>npm outdated</code>	check 4 outdated npm package
<code>npm update</code>	update npm package - only updates minor & patches
<code>npm un</code>	uninstall npm package
<code>npm login</code>	login in npm registry
<code>npm adduser</code>	create user on npm registry

