

فهرست مطالب

۱۲	 مقدمه
۱۴	 موتور V8
۱۴	 استاندارد ECMA و زبان برنامه‌نویسی ECMAScript
۱۵	 مفهوم Process
۱۷	 فرایندها چگونه عمل می‌کنند؟
۱۸	 ویژگی‌های فرایند
۱۸	 تفاوت Single Thread و Multi Thread
۱۹	 Thread چگونه کار می‌کند؟
۱۹	 انواع Thread ها
۱۹	 ویژگی‌های Thread
۲۰	 تفاوت‌های کلیدی بین Thread و Process
۲۰	 Multiprocessing چیست؟
۲۰	 Multithreading چیست؟
۲۱	 تفاوت کلیدی بین Multiprocessing و Multithreading
۲۲	 تفاوت میان Blocking (Synchronously) و Non-Blocking (Asynchronously)
۲۳	 دسترسی به منابع خارجی
۲۳	 Blocking
۲۴	 Non-Blocking
۲۶	 مراحل حلقه رویداد
۲۷	 ویژگی‌های Node.js
۲۷	 چه شرکت‌هایی از Node.js استفاده می‌کنند؟
۲۷	 در کجا از Node.js استفاده کنیم؟
۲۸	 در کجا از Node.js استفاده نکنیم؟
۲۸	 فایل Node.js چیست؟
۲۸	 نصب Node.js و اجرای فایل (Node.js Runtime)
۳۰	 فصل اول مفاهیم اولیه مورد نیاز در Node.js
۳۰	 تفاوت بین Var, Let و Const
۳۰	 کلمه کلیدی Var
۳۲	 کلمه کلیدی Let
۳۳	 کلمه کلیدی Const
۳۴	 تعریف String

۳۶	 Try-Catch-Finally ساختار
۳۹	 بررسی انواع توابع، تفاوت توابع Regular با Arrow همراه ارسال پارامتر
۴۰	 Function Expression
۴۰	 Anonymous Function
۴۱	 Immediately Invoked Function Expressions
۴۱	 Arrow Function
۴۳	 تفاوت بین توابع Regular و Arrow
۴۳	 نحوه تعریف توابع Regular
۴۳	 نحوه تعریف توابع Arrow
۴۵	 فراخوانی Callback
۴۷	 فراخوانی Promise و تفاوت آن با Callback
۵۸	 متدهای Promise
۶۵	 Async/Await فراخوانی
۶۵	 مدل همزمان (Synchronous)
۶۶	 مدل ناهمزمان (Asynchronous)
۷۶	 کلمه کلیدی This و Bind کردن
۸۱	 تفاوت بین Bind(), Call() و Apply()
۸۱	 متد Bind()
۸۵	 متد Call()
۸۶	 متد Apply()
۸۷	 نکته‌ها

۸۸ (Import, Export, Require, Exports) CommonJS و ES Module **فصل دوم**

۸۸	 مقایسه CommonJS با ES Modules
۹۰	 Require()
۹۳	 Import

۹۶ (Node.js Global Objects) اوجکت‌های سراسری **فصل سوم**

۹۶	 __Filename
۹۶	 __Dirname
۹۷	 setTimeout(cb, ms)
۹۷	 clearTimeout(t)
۹۸	 setInterval(cb, ms)
۹۸	 clearInterval(t)
۹۸	 setImmediate()
۱۰۰	 clearImmediate(i)

۱۰۰.....	آبجکت Process
۱۰۰.....	Exit
۱۰۰.....	BeforeExit
۱۰۱.....	UncaughtException
۱۰۱.....	Signal Events

فصل چهارم کارکردن با NPM، تفاوت آن با NPX و مدیریت پکیج‌ها ۱۱۷.....

۱۱۷.....	NPM چیست؟
۱۱۸.....	ایجاد یک پروژه خام
۱۱۹.....	نصب و اجرای پکیج توسط NPM
۱۲۳.....	NPX چیست؟
۱۲۴.....	تفاوت‌های NPM و NPX

فصل پنجم کارکردن با Event Loop و مدیریت رویدادها ۱۲۵.....

۱۲۵.....	برنامه‌نویسی رویدادمحور (Event-Driven)
۱۲۶.....	متدهای رویداد
۱۳۹.....	ارسال پارامتر و This به شنونده رویداد
۱۴۰.....	بررسی برخی متدها
۱۴۰.....	رویدادها

فصل ششم بافر (Buffer) ۱۴۴.....

۱۴۴.....	ساخت Buffer
۱۴۷.....	خواندن از بافر و کارکردن با تبدیل سیستم کُدگذاری
۱۵۰.....	بافر‌ها و کارکردن با نوع آرایه
۱۵۳.....	بافر‌ها و شمارش‌پذیربودن
۱۵۳.....	برخی دیگر از متدهای بافر

فصل هفتم Stream ۱۷۰.....

۱۷۱.....	چرا Stream؟
۱۷۱.....	انواع Stream
۱۷۲.....	Stream Promise API
۱۷۲.....	Buffering
۱۷۳.....	انواع Stream‌های قابل نوشتن (Writable) و قابل خواندن (Readable)
۱۷۴.....	File System (fs)
۱۷۵.....	جدول فلگ‌های (Flags) کارکردن با فایل
۱۷۶.....	جدول ثابت‌های دسترسی (Constants) به فایل

۱۷۷.....	Stream و کارکردن با سیستم فایل
۱۸۲.....	FileHandle متدهای
۱۸۷.....	fsPromises متدهای
۲۰۷.....	حالت Callback API و برخی متدها
۲۲۸.....	Synchronous API متدهای
۲۳۹.....	(Net: IPC-Async Network-TCP) فصل هشتم
۲۴۰.....	net.BlockList کلاس
۲۴۱.....	blockList.rules
۲۴۲.....	net.SocketAddress کلاس
۲۴۲.....	net.Server کلاس
۲۴۳.....	net.Server رویدادهای
۲۴۴.....	Server متدهای
۲۴۷.....	net.Socket کلاس
۲۴۸.....	net.Socket رویدادهای
۲۵۰.....	net.Socket متدها و خصوصیت‌های
۲۶۱.....	(http: Server-Agent-ClientRequest-ServerResponse) فصل نهم
۲۶۱.....	http.METHODS
۲۶۲.....	http.STATUS_CODES
۲۶۲.....	متد http.createServer([options][, requestListener])
۲۶۳.....	متدهای http.get(options[, callback])
۲۶۳.....	http.get(url[, options][, callback])
۲۶۵.....	متدهای http.request(options[, callback])
۲۶۵.....	http.request(url[, options][, callback])
۲۷۰.....	http.Agent کلاس
۲۷۲.....	متد agent.createConnection(options[, callback])
۲۷۲.....	متد agent.keepSocketAlive(socket)
۲۷۲.....	متد agent.reuseSocket(socket, request)
۲۷۳.....	متد agent.destroy()
۲۷۳.....	agent.freeSockets
۲۷۵.....	http.ClientRequest کلاس
۲۷۶.....	Abort رویداد
۲۷۶.....	Close رویداد
۲۷۶.....	Connect رویداد
۲۷۸.....	Continue رویداد

۲۷۸.....	Finish رویداد
۲۷۸.....	Information رویداد
۲۷۸.....	Response رویداد
۲۷۹.....	Socket رویداد
۲۷۹.....	Timeout رویداد
۲۷۹.....	Upgrade رویداد
۲۸۰.....	request.abort() متد
۲۸۰.....	request.end([data[, encoding]][, callback]) متد
۲۸۰.....	request.destroy([error]) متد
۲۸۱.....	request.flushHeaders() متد
۲۸۱.....	request.getHeader(name) متد
۲۸۱.....	request.setHeader(name, value) متد
۲۸۱.....	request.hasHeader(name) متد
۲۸۱.....	request.removeHeader(name) متد
۲۸۲.....	request.write(chunk[, encoding][, callback]) متد
۲۸۲.....	response.writeHead(statusCode[, statusMessage][, headers]) متد
۲۸۳.....	request.path
۲۸۳.....	request.method
۲۸۳.....	request.protocol
۲۸۳.....	request.host
۲۸۳.....	http.Server کلاس
۲۸۴.....	server.close([callback]) متد
۲۸۴.....	server.closeAllConnections() متد
۲۸۴.....	server.listen() متد
۲۸۴.....	http.ServerResponse کلاس
۲۸۵.....	Close رویداد
۲۸۵.....	Finish رویداد
۲۸۵.....	response.end([data[, encoding]][, callback]) متد
۲۸۵.....	response.write(chunk[, encoding][, callback]) متد
۲۸۶.....	response.statusCode
۲۸۶.....	response.statusMessage