

# DT162G – Node.js



# VAD ÄR Node.js?

- ”As an asynchronous event driven framework, Node.js is designed to build scalable network applications.”
- Node.js är 'Javascript för backend' och fristående applikationer
- Node.js som plattform är byggd med Google Chrome V8 Javascript-motor. Målet med denna är att kunna skapa resurseffektiva, snabba, skalbara och händelsestyrda applikationer med ”non-blocking I/O”

# Node.js historia

- V8 som är ett öppet källkodsprojekt som drivs av Google är den motor som exekverar Javascript och finns i kärnan av Google Chrome
- Lanserades den 2 september 2008 inkluderad i webbläsaren Chrome
- V8 är utvecklad i C++ och dess arbete är att förkompilera all Javascriptkod i sidan som ska exekveras direkt till maskinkod istället för att tolka den löpande(JIT)vilket ökade exekveringshastigheten för sidor med Javascriptkod inbakad
- Under 2009 föddes de delar som kom att bli de första fragmenten till dagens Node.js
- Ryan Dahl's presentation av Node.js  
<https://www.youtube.com/watch?v=ztspvPYybIY>





Even though Node.js is far behind as compared PHP, it is growing in terms of the trend while PHP is extremely popular in terms of download, though the downfall in terms of trend is visible.

- **129,137** are the total number of websites currently active on Node.js
- **7,474,278** are the total number of sites currently active on PHP



# When to Use Node.js Vs. PHP

Both PHP and Node.js used for server-side programming language. However, you need to choose the technology smartly based on the project in hand.

## When to Use Node.js over PHP

### **When You Need Same Language across the Stack:**

When you want to create dynamic single page applications (SPA) which require MEAN stack development software such MongoDB, ExpressJs, AngularJs, Node is an excellent choice.

**When Your Project Involve Real-Time Data:** If you have projects which require real-time data, Node can be applied for. However, for projects such as financial applications and others, you can consider using alternate technologies.

**When You Require Speed:** Projects requiring faster development, Node.js is the way to go. We have already discussed that Node.js supports asynchronous, which naturally supports speedier development.

## When to Use PHP over node

**When you have Centralized Server:** When your application has to work on a centralized server and does not require scaling on multiple servers, then PHP is one of the options to go with.

**When You Require Portability:** Any platform which supports Apache, IIS and a supported database system installed, PHP can work wonder for you.

# Vilka använder Node.js

Node.js	PHP
<ul style="list-style-type: none"><li>• PayPal</li><li>• LinkedIn</li><li>• Yahoo</li><li>• Mozilla</li><li>• Netflix</li><li>• Uber</li><li>• Groupon</li><li>• GoDaddy</li><li>• Ebay</li></ul>	<ul style="list-style-type: none"><li>• Facebook</li><li>• Wikipedia</li><li>• Baidu</li><li>• Yahoo</li><li>• Tumblr</li><li>• Flickr</li><li>• WordPress</li><li>• Flipkart</li><li>• Digg</li><li>• iStockPhoto</li></ul>

# Node.js installation

Node.js är enkelt att installera och det finns instruktioner och anvisningar för plattformarna Windows, Linux och Mac

<https://nodejs.org/>

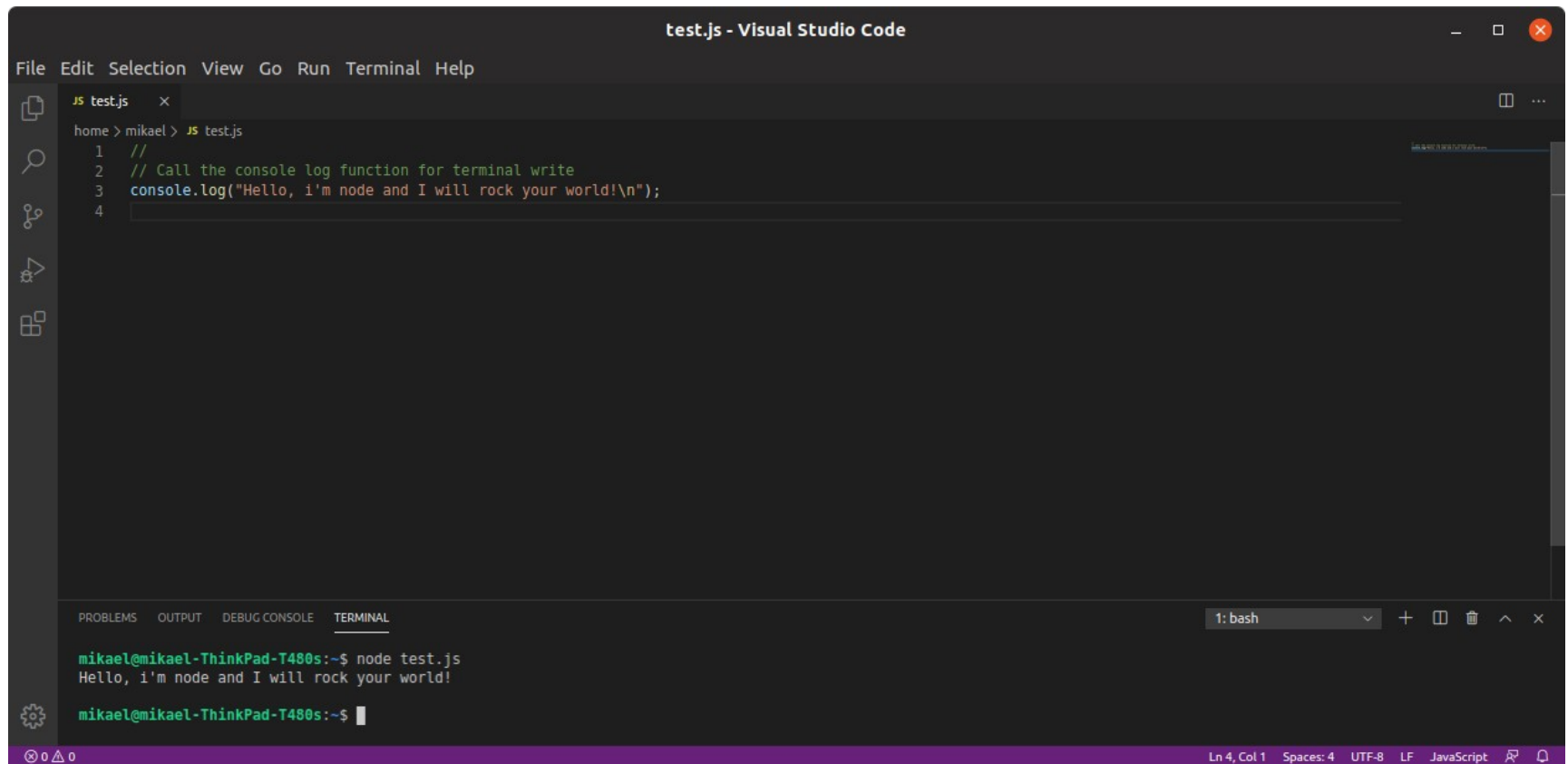
# Ett första exempel

Skapa en fil med namn test.js

```
//  
// Call the console log function for terminal write  
console.log("Hello, i'm node and I will rock your world!\n");
```

Starta applikationen från kommandoraden(kommandotolken) med  
\$ node test.js



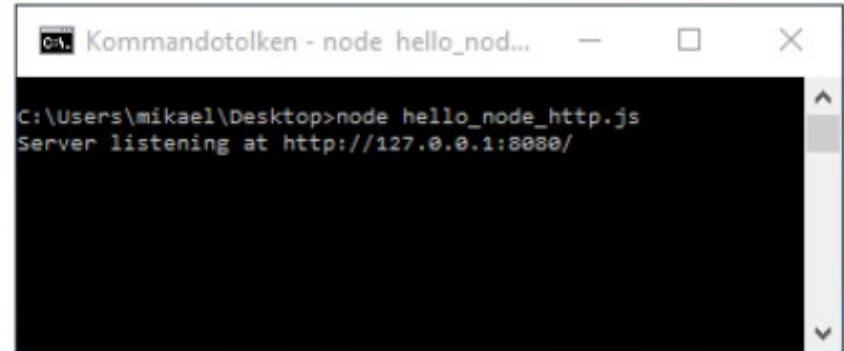


The image shows a screenshot of the Visual Studio Code editor interface. The title bar at the top reads "test.js - Visual Studio Code". The menu bar includes "File", "Edit", "Selection", "View", "Go", "Run", "Terminal", and "Help". The editor area displays a file named "test.js" with the following content:

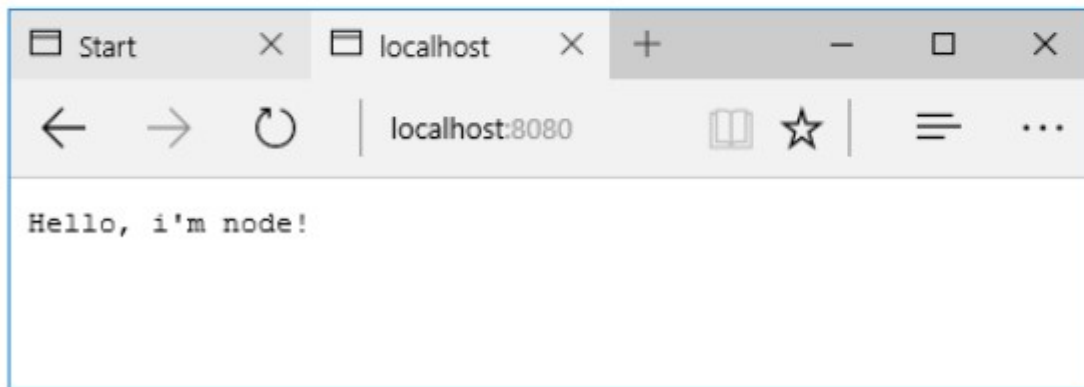
```
1 //  
2 // Call the console log function for terminal write  
3 console.log("Hello, i'm node and I will rock your world!\n");  
4
```

Below the editor, the "TERMINAL" panel is active, showing a bash shell. The prompt is "mikael@mikael-ThinkPad-T480s:~\$". The user has entered the command "node test.js", and the output is "Hello, i'm node and I will rock your world!". The status bar at the bottom indicates "Ln 4, Col 1", "Spaces: 4", "UTF-8", "LF", "JavaScript", and a search icon.

# Node som HTTPserver



```
C:\Users\mikaël\Desktop>node hello_node_http.js
Server listening at http://127.0.0.1:8080/
```



# Node som HTTPserver

```
//  
// Initiate http module  
var http = require('http');  
  
//  
// Create HTTP server and setup to respond with "Hello, i'm node!" to all requests  
var server = http.createServer(function (request, response) {  
  response.writeHead(200, {"Content-Type": "text/plain"});  
  response.end("Hello, i'm node!\n");  
});  
  
//  
// Listen at port 8080 on localhost for requests  
server.listen(8080);  
  
//  
// Show server message on console  
console.log("Server listening at http://127.0.0.1:8080/");
```

# Node pakethanterare (NPM)

Node har en inbyggd pakethanterare som kan läsa in moduler som behövs för utökad funktionalitet i programmen.

De moduler/paket som används definieras i en konfigurationsfil med namnet `package.json`

Om man önskar tillgång till en modul (`require`) i sin programkod konfigureras denna modul i filen `package.json`

Node har en uppsättning redan inbyggda moduler som kan användas direkt utan konfigurering med `package.json`. Modulen `http` är en sådan redan inbyggd modul som inte behöver konfigureras med `package.json`



Node.js has a set of built-in modules which you can use without any further installation.

Here is a list of the built-in modules of Node.js version 6.10.3:

Module	Description
<a href="#">assert</a>	Provides a set of assertion tests
<a href="#">buffer</a>	To handle binary data
<a href="#">child_process</a>	To run a child process
<a href="#">cluster</a>	To split a single Node process into multiple processes
<a href="#">crypto</a>	To handle OpenSSL cryptographic functions
<a href="#">dgram</a>	Provides implementation of UDP datagram sockets
<a href="#">dns</a>	To do DNS lookups and name resolution functions
<a href="#">domain</a>	Deprecated. To handle unhandled errors
<a href="#">events</a>	To handle events
<a href="#">fs</a>	To handle the file system
<a href="#">http</a>	To make Node.js act as an HTTP server
<a href="#">https</a>	To make Node.js act as an HTTPS server.
<a href="#">net</a>	To create servers and clients
<a href="#">os</a>	Provides information about the operation system
<a href="#">path</a>	To handle file paths
<a href="#">punycode</a>	Deprecated. A character encoding scheme
<a href="#">querystring</a>	To handle URL query strings
<a href="#">readline</a>	To handle readable streams one line at the time
<a href="#">stream</a>	To handle streaming data
<a href="#">string_decoder</a>	To decode buffer objects into strings
<a href="#">timers</a>	To execute a function after a given number of milliseconds
<a href="#">tls</a>	To implement TLS and SSL protocols
<a href="#">tty</a>	Provides classes used by a text terminal
<a href="#">url</a>	To parse URL strings
<a href="#">util</a>	To access utility functions
<a href="#">v8</a>	To access information about V8 (the JavaScript engine)
<a href="#">vm</a>	To compile JavaScript code in a virtual machine
<a href="#">zlib</a>	To compress or decompress files

[https://www.w3schools.com/nodejs/ref\\_modules.asp](https://www.w3schools.com/nodejs/ref_modules.asp)

# Node paket

Filer man vill få tillgång till i sitt huvudprogram hanteras som paket och läses in med *require*.

Funktioner som lagras i separata filer hanteras som paket.

```
test.js

const fahrenheit = require('./fahrenheit.js');
let temp = -5;
console.log("The temperature outside is " + temp + " degrees celcius and that corresponds to "+fahrenheit.celcius2fahrenheit(temp) + " degrees fahrenheit" );
```

```
fahrenheit.js

exports.celcius2fahrenheit = function (t) {
  ...   return (t * 1.8) + 32;
};
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
mikael@mikael-ThinkPad-T480s:~/lect$ node test.js
The temperature outside is -5 degrees celcius and that corresponds to 23 degrees fahrenheit
mikael@mikael-ThinkPad-T480s:~/lect$
```



# Node filhantering

Node.js har ett inbyggt paket som hjälper dig att arbeta med filsystemet.

Med detta paket kan du

- läsa filer
- skapa filer
- uppdatera filer
- ta bort filer
- döpa om filer

Läsning/skrivning mot filer kan ske synkront eller asynkront. Asynkron(non blocking i/o) filhantering är alltid att föredra för att utnyttja prestanda i Node.js !

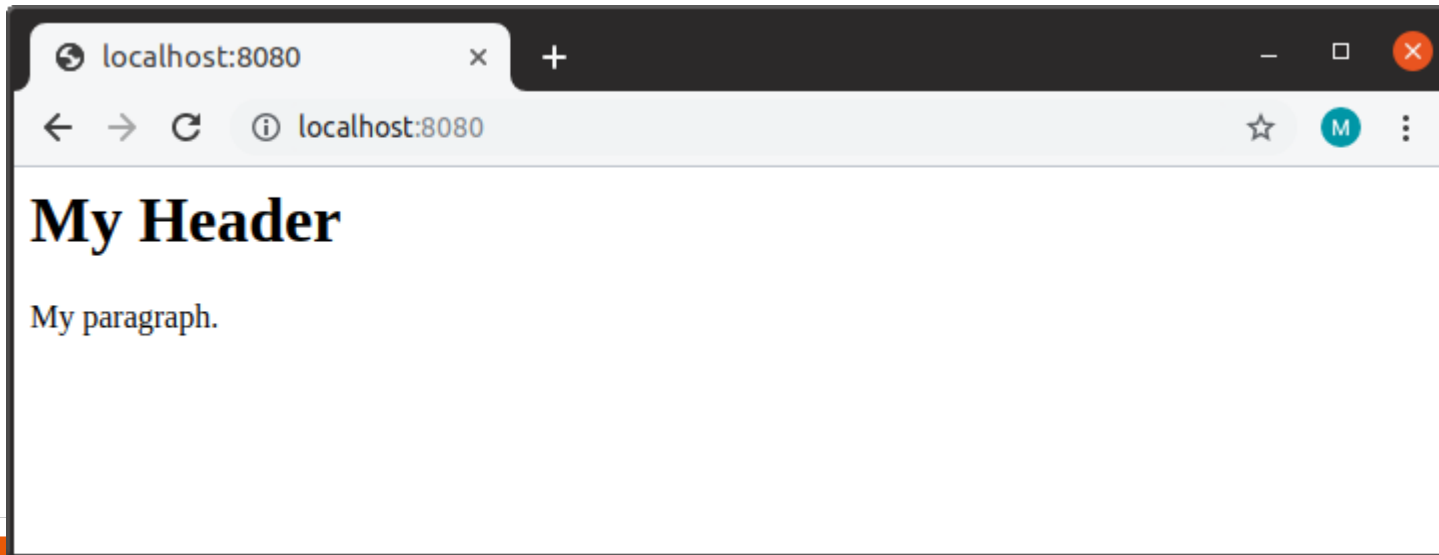
# Exempel

test.html

```
<html>
<body>
<h1>My Header</h1>
<p>My paragraph.</p>
</body>
</html>
```

server.js

```
var http = require('http');
var fs = require('fs');
http.createServer(function (req, res) {
  fs.readFile('test.html', function(err, data) {
    res.writeHead(200, {'Content-Type': 'text/html'});
    res.write(data);
    res.end();
  });
}).listen(8080);
```



# DÅ VAR DET SLUT... J

