

# Node.js

Let's head down to the basement

*christoffer.wallenberg@zocom.se*

**ZoCom**

# Runtime environment?

**Miljö som kör processer ( ex. Webbserver, biz-logic, säkerhet etc )**



*= samma som i Chrome webbläsaren*

**Vart i stacken passar Node.js in?**

# MERN

Klientsidan, grafik etc ( HTML, CSS, JS )

React

Databaser, serverkod, biz-logic etc

Express, Node, Mongo

Servern, OS ( Linux ), säkerhet

Linux

# Single threaded?

En process åt gången...typ

...om man inte startar child processes

**Non-blocking?**

## Blocking

En kan äta åt gången



## Non-Blocking

Alla kan äta samtidigt



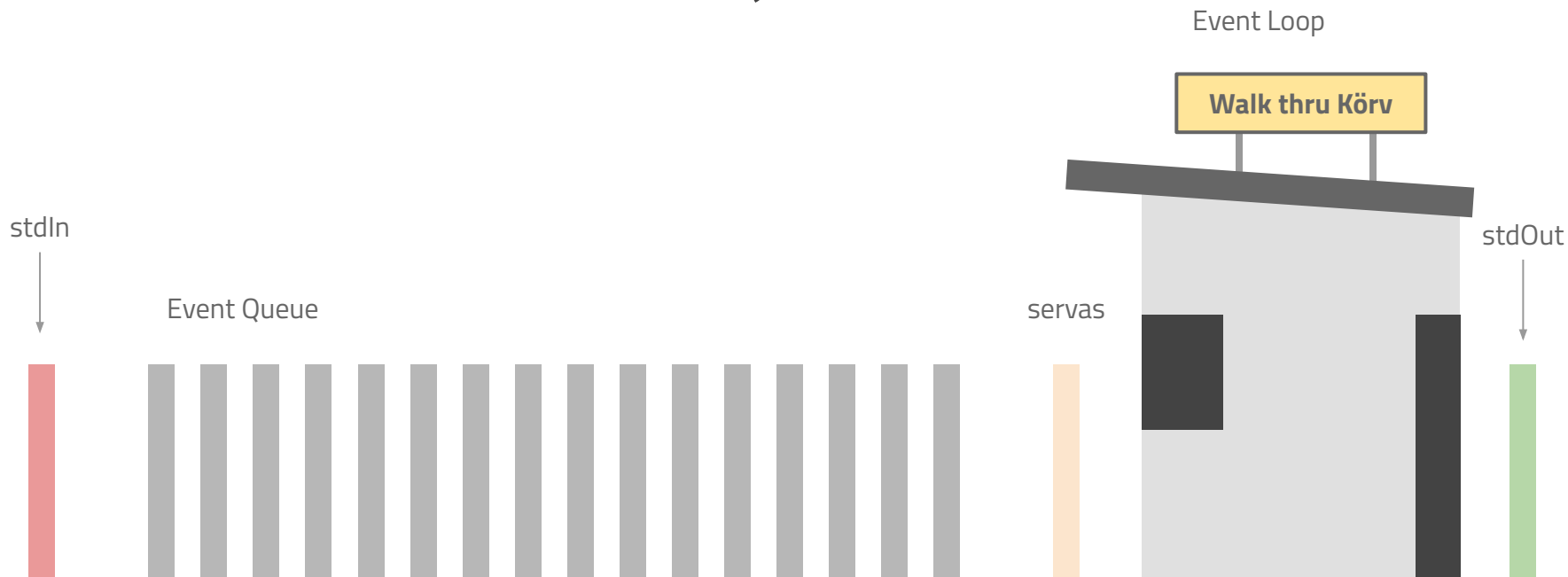


# Korvkiosken

En liten liknelse på hur Node fungerar

# The walk thru korvkiosk

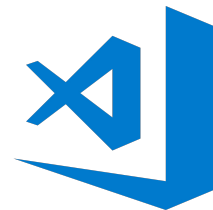
En Node.js liknelse



**Vad kan man göra med node.js?**

**allt!**

**..nästan**



**OSV...**



**Nu kör vi!**

# 1. Installera

[nodejs.org](https://nodejs.org)

## **2. Hello World**

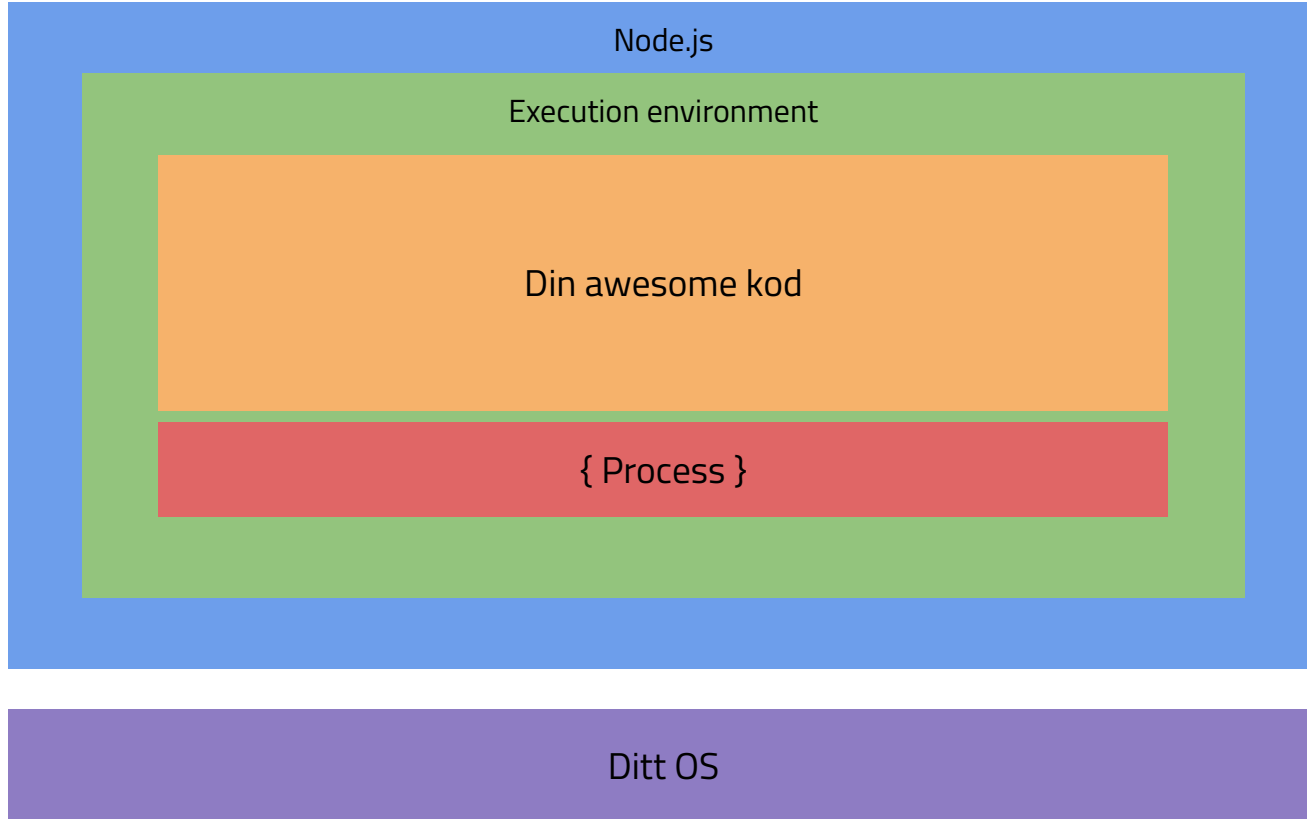


A man with glasses and a mustache, wearing a green t-shirt, is holding a vintage computer keyboard. He is sitting at a desk with two computer monitors in the background. The image has a green overlay. The text "Lets code!" is written in white, bold, sans-serif font across the center of the image.

Lets code!

**{ process }**

Globalt objekt med metoder etc, ex. console(.log, .error...)



# Modules

JS bibliotek som utökar funktionaliteten hos din App

# require()

Importerar modulen i din fil

```
const express = require('express');
```

# Tre typer av modules

Fungerar dock på samma sätt

- Inbyggda, ex. *path*, *http*, *fs*.
- Externa, NPM
- Egna

# Inbyggda modules ( core modules )

- Lösningar på vanligt förekommande kodproblem
- Avser att förenkla
- Ex, *path, http, fs*

# Ex. `path.normalize`

**Problem:** Olika OS har olika formatering på sökvägar

**Windows**     `\path\to\my\project`

**OSX / Linux**   `/path/to/my/project`

**Lösning:** Path-modulen konvertera mellan olika format

```
path.normalize(path);
```



# Egna modules

- Ordning och reda i koden
- Gör det lättare att skala upp / felhantera etc
- Vad utgör en modul? Hur ska man tänka?

# När ska man göra nya modules?

Tänk så här...

## ROLLER

## ANSVAR

Värd / entrépersonal

Hej och välkommen, detta är erat bord...

Serveringspersonal

Mat- och dryckeshantering och leverans

Kock

Lagar maten

Bartender

Fixar kaffe / drinkar

***Separation Of Concerns***

# Egna modules

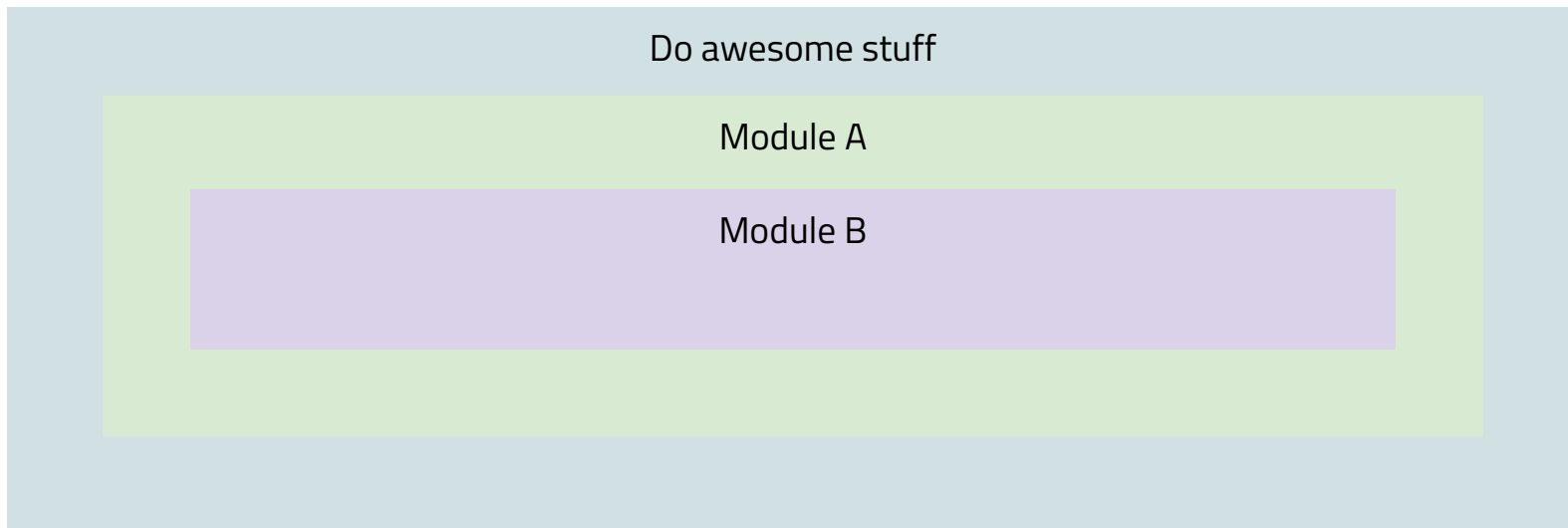
Exportera det du vill dela med resten av koden

```
function greeting(name) {  
  return `Nämen hejsan ${name}, din gamle get!`;  
}
```

```
exports.greeting = greeting(name);
```

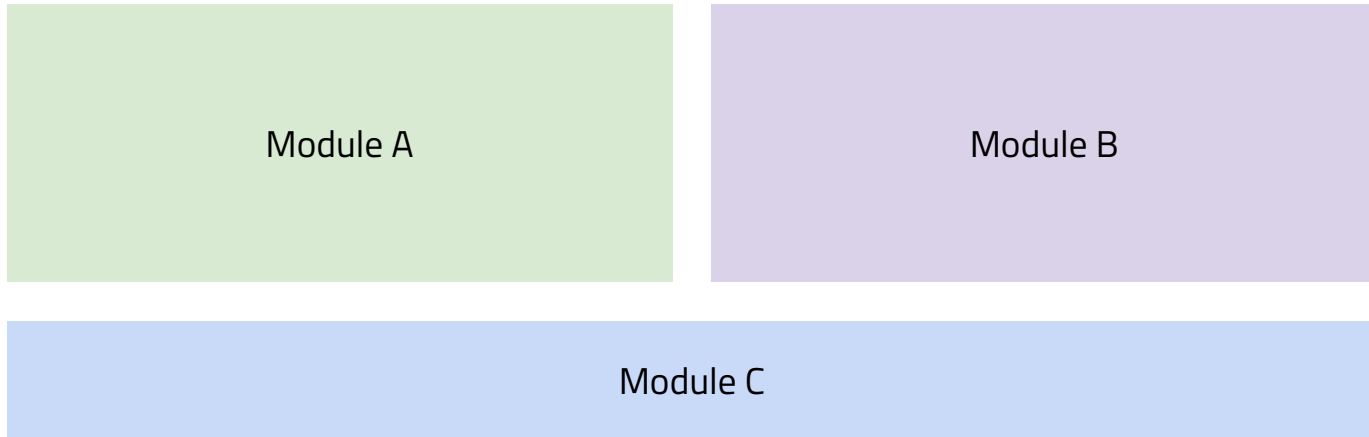
# Egna modules

## Module inception



# Module caching

Importera samma modul på flera ställen = samma modul



# Affären

## Module caching exempel

mejeri

chark

grönsaker

godis

Cart

# Externa modules

- **Över 1 miljon färdiga lösningar via NPM**
- **Flesta stora bibliotek finns via NPM ( ex. Express )**
- **Node Package Manager**

# **NPM**

**Node Package Manager**



# Vad är NPM?

- Hemsidan ( `npmjs.com` )
- CLI interface ( `npm ...` )
- Registret ( databasen )

# **npmjs.com**

- **Söka fram packages**
- **Hjälp att välja rätt package**
- **Hitta repos på github**

# **package.json**

**Settings-file för dina packages**

# package.json

<b>name</b>	namn på program
<b>version</b>	version av ditt program
<b>description</b>	kort beskrivning av ditt program
<b>author</b>	ditt namn <epost>
<b>main</b>	start-filen av ditt program
<b>keywords</b>	sökord för ditt program
<b>dependencies</b>	externa packages ditt program är beroende av
<b>license</b>	ISC, MIT, Apache-2 etc.

# Hur gör man?

`npm init flag`

`npm init` skapar fil efter några frågor

`npm init -y` skapar fil direkt

A person with a mustache and glasses, wearing a green t-shirt, is holding a vintage computer keyboard. The background shows a desk with two computer monitors and various peripherals. The entire image is overlaid with a semi-transparent green filter. The text "Lets code!" is written in a large, white, sans-serif font, and "Skapa en *package.json*" is written in a smaller, white, sans-serif font below it.

# Lets code!

Skapa en *package.json*

# NPM Install

**npm** kommando package ev. flag

**npm** install

allt i din package.json > dependencies

**npm** install express

**npm** install express -g

# Flaggor?

- global** Spara globalt på datorn ( -g )
- save** Spara i din package.json ( -S )
- save-dev** Spara som dev dependency i din package.json



A person wearing a green jacket and glasses is holding a vintage computer terminal. The background shows a room with multiple computer monitors and keyboards, suggesting a retro computing environment. The entire image has a green tint.

# Lets code!

Installera valfri package och spara som *dependency*

**Läsa/Skriva**

# Läsa från en fil

```
const fs = require('fs');
```

```
fs.readFile('text.txt', 'utf8', (err, content) => {  
  console.log(content);  
});
```

Allt innehåll i text.txt hamnar i **content**

# Skriva till en fil

```
const fs = require('fs');
```

```
let quote = 'Be yourself. Everyone else is already taken.'
```

```
fs.writeFile('quotes.txt', quote, (err) => {  
  console.log('Quote saved!');  
});
```

**Hur gör man ifall man vill kontinuerlig  
skriva till en fil?**

Strömmar!

# Strömmar - Skriva

```
const fs = require('fs');  
const file = fs.createWriteStream('OscarWilde.txt');  
  
for(let i=0; i < 5; i++) {  
  file.write('You can never be overdressed or overeducated.\n');  
}  
  
file.end();
```

A man with glasses and a green t-shirt is holding a vintage computer keyboard. He is sitting at a desk with two computer monitors in the background. The image has a green overlay. The text "Lets code!" is written in white in the center.

# Lets code!