

# Internetprogrammering

## DD1389

Föreläsning 6

# Innehåll

- JSON
- Högre ordningens funktion
- npm
- Nodejs
- HTTP Server exempel
- En statisk webserver

# Object

```
let person = {  
  name : "Peter",  
  labbs : [ 1, 2],  
}
```

```
Object.keys(person) ger [name, labbs]
```

```
Object.assign(person, {labbs:[1,2,3,4] , betyg : "A"})
```

```
console.log(person) ger {name: "peter", labbs:[1,2,3,4], betyg: "A"}
```

# JSON (JavaScript Object Notation)

- Ett populär serialiserat format.
- Används för att lagra data och kommunikation på nätet.
- Formatet är väldigt likt objekt och arrayer i javascript.
- Alla property måste anges med citattecken (")
- Ingen funktionsanrop, variabel deklaration eller kod som involverar beräkningar.
- Kommentarer är inte tillåtna.
- Exempel:  

```
{ "morgonpigg" : true , "kurser": ["tilda21", "intnet22", "adk22"] }
```

# Nodejs

- En platform för att skapa nätverksapplikationer
- Baserad på Google chormes JavaScript
- Eventdriven
- Javascript på serversidan
- Dokumentation: <https://nodejs.org/api/>
- Inbyggda moduler bl.a:
  - fs: filesystem
  - http: allt om HTTP protokollet
  - url: uniform resource location
  - Mime: t.ex: text/html, img/JPEG, osv...

# Npm (Node Package Manager)

På skolans datorer node är installerad.

`npm install [andra moduler]`

t.ex:

`npm install ejs` : Installera Embedded JavaScript templates

# Nodejs http-server exempel

```
//file: serverhttp.js
const {createServer} = require('http');
let server = createServer(
  (request, response) => {
    response.writeHead(200,
      { 'Content-Type': 'text/html' });
    response.write(`<h1>Du kör Nodejs!!! </h1>
      <p>urln:n i förfrågan:
      <code>${request.url}</code></p>`);
    response.end();
  });
server.listen(8000);
console.log("Listening! (port 8000)");
```

# Förklaring av koden

- `require`: importerar moduler, t.ex `require('http');`
- `http` modulen består av en del funktioner och variabler  
`const {createServer} = require('http');`  
är kompakt version av:  
`let http_modulen= require('http');`  
`let createServer = http_modulen.createServer;`
- `(request, response) => {...}`: är en anonym callbackfunktion som anropas för varje inkommande http-request
- `server.listen(8000)` börjar lyssna på port
- `console.log('Listening...')` skriver ut ett textmeddelande