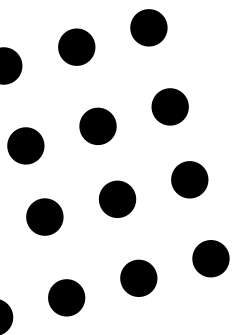
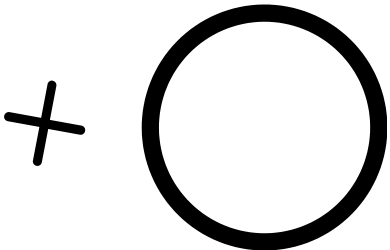


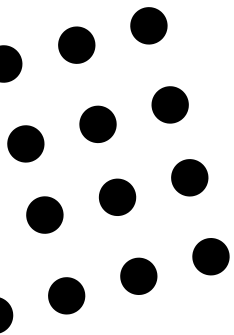
Introduzione a Node.js

Che cos'è e perchè dovresti usarlo

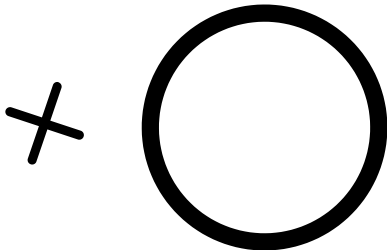




Che cos'è Node.js

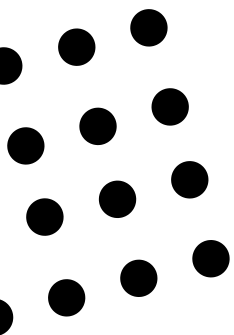


Che cos'è Node.js

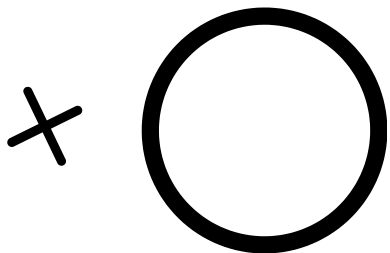


Cosa è Node.js

Node è un runtime environment per eseguire javascript al di fuori dai browser



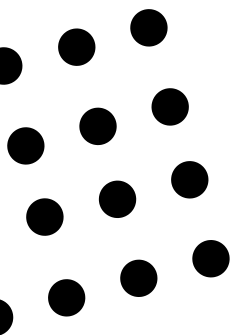
Che cos'è Node.js

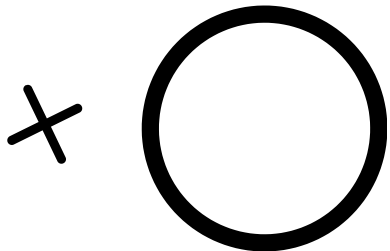


Cosa NON è Node.js

~~Un linguaggio di programmazione~~

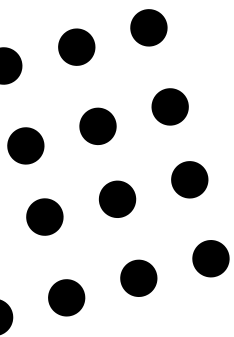
~~Un framework~~





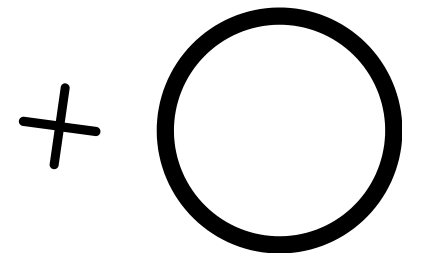
Cosa posso fare con Node.js?

- Utilities di sviluppo (Grunt, Gulp, Yoeman ecc)
- Webserver
- API Rest



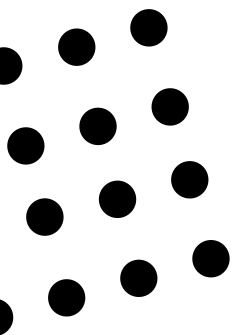
Che cos'è Node.js

06 Aprile 2019



Perchè usare Node.js?

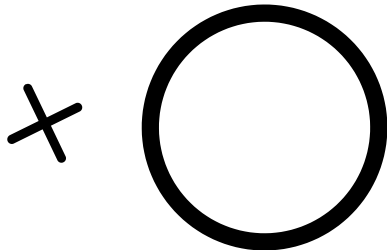
- Codebase più consistente
- Ecosistema di librerie gratuite con NPM
- Division on concerns maggiore



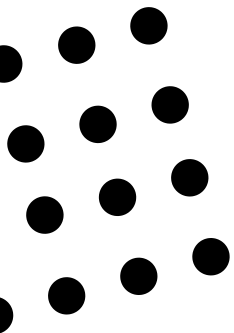
```
console.log("Hello World");
```

noBrand = digitalCommunity

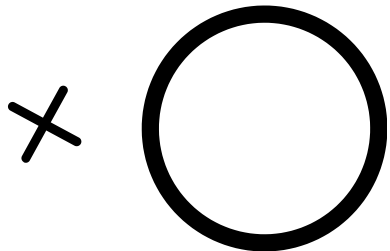
**WHITE
LABEL**



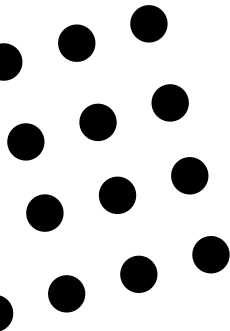
Chi usa Node.js?

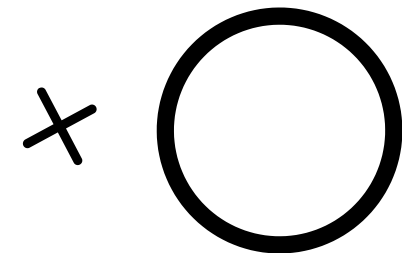


Chi usa Node.js?



Grandi aziende che usano Node



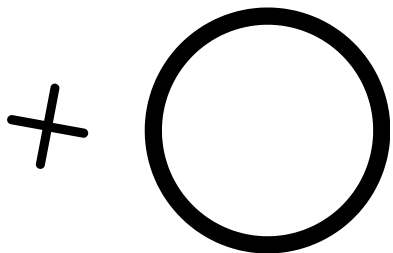


Case study Paypal

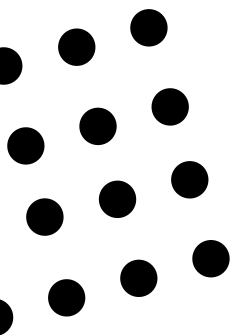


- Costruito due volte più velocemente, con meno persone
- 33% linee di codice in meno
- 40% di files in meno
- Ha servito il doppio delle richieste al secondo
- 35% response time in meno
- In media utenti serviti 200ms più velocemente

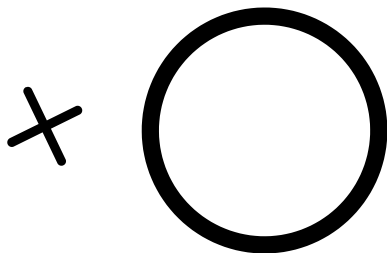




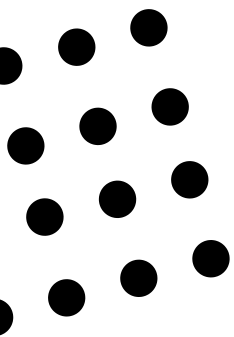
Come funziona Node.js?



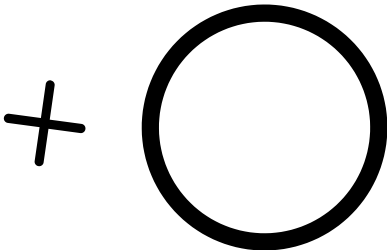
Come funziona Node.js?



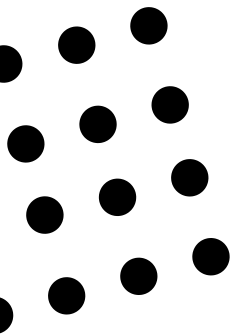
Node è basato sul JavaScript Engine di Google Chrome (V8), embeddato all'interno di un programma C++
Creato da Ryan Dahl nel 2009



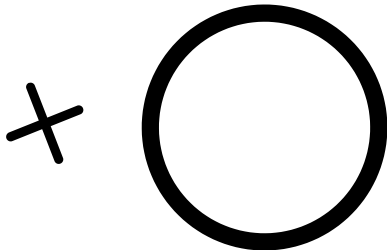
Come funziona Node.js?



```
document.getElementById();  
window.alert();
```



Come funziona Node.js?



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

fs.readFile("file.json", (error, data) => { (!error)?console.log(data):console.log("error: ", error) });

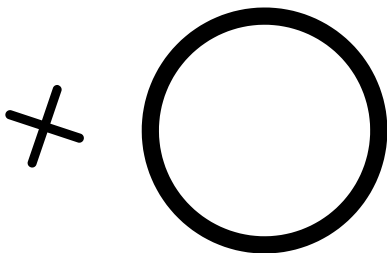
const hostname = '127.0.0.1';
const port = 3000;

const server = http.createServer((req, res) => {
  res.statusCode = 200;
  res.setHeader('Content-Type', 'text/plain');
  res.end('Hello World\n');
});

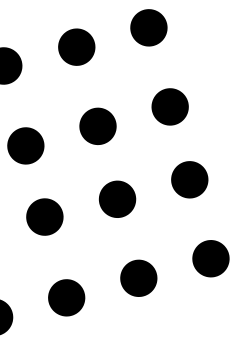
server.listen(port, hostname, () => {
  console.log(`Server running at http://${hostname}:${port}/`);
});
```



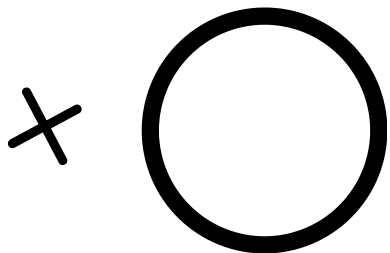
Come funziona Node.js?



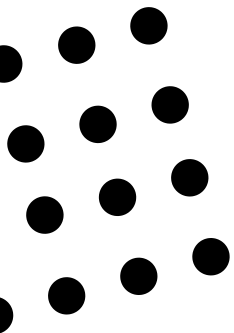
Ha una struttura Asincrona e non bloccante



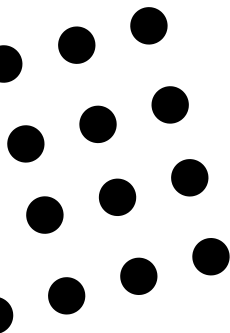
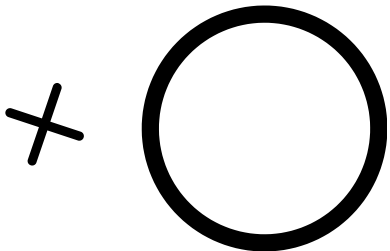
Come funziona Node.js?



Immaginiamo di entrare in un ristorante...

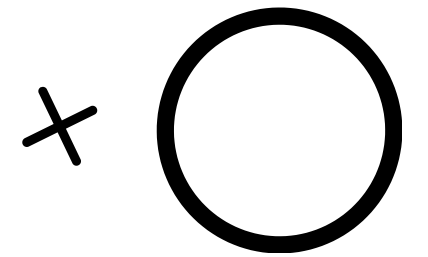


Come funziona Node.js?



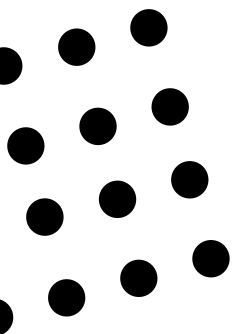
```
console.log("Hello World");
```

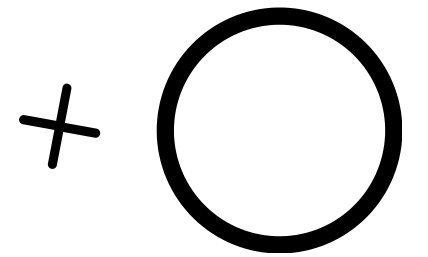
```
noBrand = digitalCommunity
```

Server PHP

- Un client esegue una richiesta
- Viene allocato un thread e il codice viene eseguito
- C'è una chiamata ad un db
- Aspettiamo la risposta del db
- Il thread viene rilasciato solamente nel momento in cui l'esecuzione del codice è terminata.



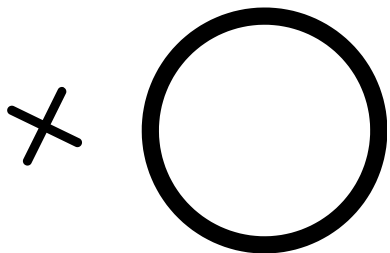


Server Node.js

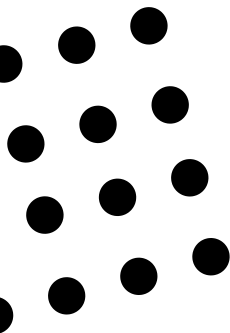
- Un client esegue una richiesta
- Il codice viene eseguito sul thread di Node.js
- C'è una chiamata ad un db
- Effettuo la chiamata e registro un evento (callback) per quando il db risponderà
- Libero il thread e servo eventuali altre richieste
- Il db risponde
- Il thread viene occupato dalla prima richiesta e esegue ciò che è indicato nella callback
- Il thread viene liberato

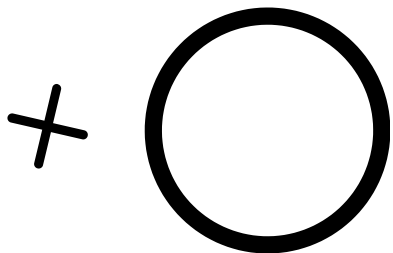


Come funziona Node.js?

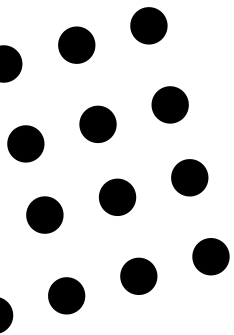


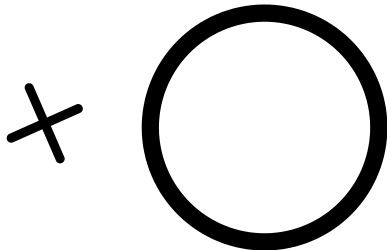
Event Queue



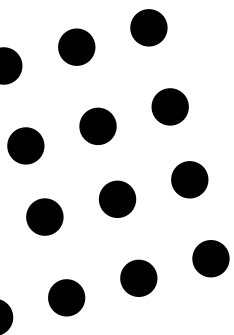


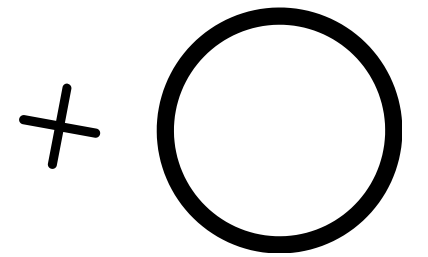
Cos'è NPM



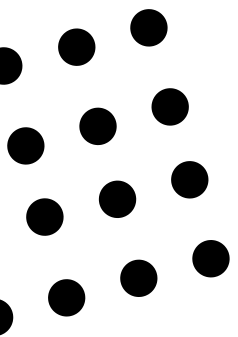


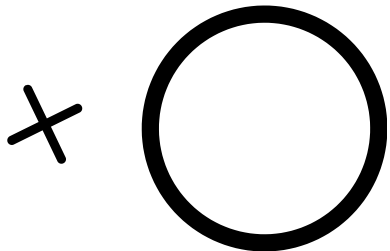
NPM è un gestore di pacchetti per JavaScript



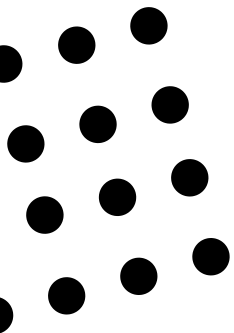


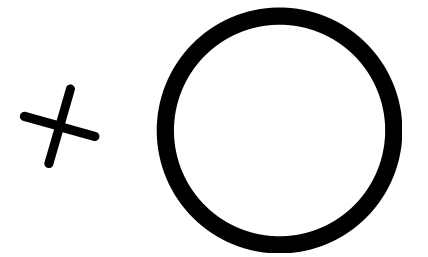
- Express - Framework completo per sviluppo di applicazioni web
- Mongoose - Wrapper per interagire con MongoDB (NoSQL database)
- Redis - Cache manager e NoSQL database
- Socket.io - Componente server-side per interagire con web sockets
- Nodemon - Utility che effettua auto reload del server in caso di cambiamenti
- PM2 - Process manager e load balancer



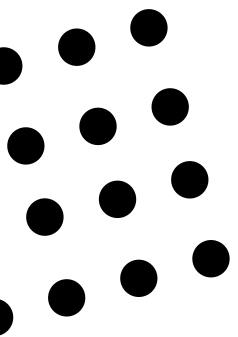


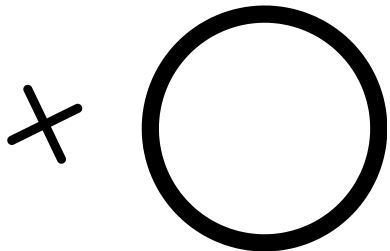
Cos'è Express



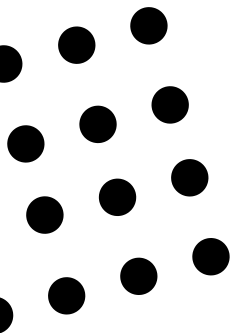


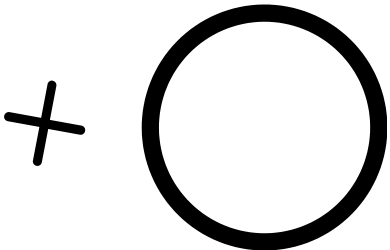
Express è un framework per applicazioni web Node.js flessibile e leggero che fornisce una serie di funzioni avanzate per le applicazioni web e per dispositivi mobili.





Elementi fondamentali di Express





Routes

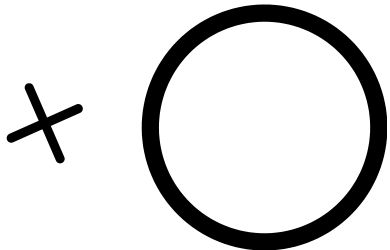
```
var express = require('express');
var app = express();

app.get('/', function (req, res) {
  res.send('Hello World!');
});

app.listen(3000);
```

Express

Middlewares

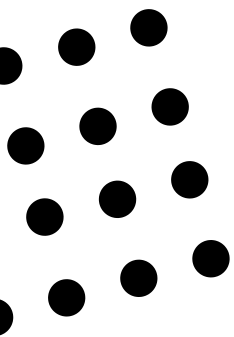


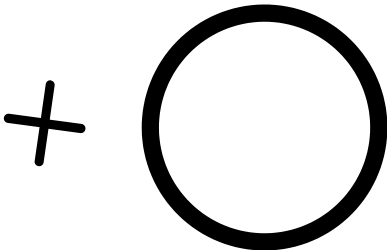
```
var express = require('express');
var app = express();

app.use(function (req, res, next) {
  console.log('Hi! I\'m a middleware!');
  next();
});

app.get('/', function (req, res) {
  res.send('Hello World!');
});

app.listen(3000);
```





Grazie
WHITE LABEL

