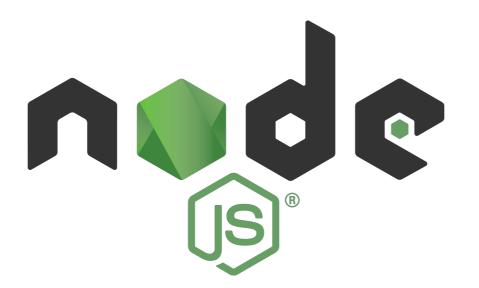
How to using NodeJS

Read-Eval-Print Loop (REPL)





พิมพ์ node เพื่อเริ่มใช้งาน

```
$ node
>
```

Simple Expression

```
> 1 + 3
4
> 1 + (2 * 3) - 4
3
>
```



การใช้งาน Variables

```
node -i -e "repl.repl.ignoreUndefined=true"
$ node
> x = 10
10
> var y = 10
undefined
> x + y
20
> console.log("Hello World")
Hello World
undefined
```



Ignore Undefined Warning

node -i -e "repl.repl.ignoreUndefined=true"



การใช้งาน Multiline Expression

```
$ node
> for( i=0; i<=5; i++ ){
... console.log(i);
4
5
undefined
```



การดึงค่า Variable ล่าสุดมาใช้งานโดยใช้ underscore (_)

```
$ node
> var x = 10
undefined
> x + 20
30
> var sum = _
undefined
> console.log(sum)
30
undefined
>
```



การดึงค่า Error ตัวล่าสุดมาใช้งานโดยใช้ _error

```
$ node
> throw new Error('foo');
Error: foo
> _error.message
'foo'
```



การใช้งาน await

```
$ node --experimental-repl-await
> await Promise.resolve(123)
123
> await Promise.reject(new Error('REPL await'))
Error: REPL await
  at repl:1:45
> const timeout = util.promisify(setTimeout);
undefined
> const old = Date.now(); await timeout(1000); console.log(Date.now() - old);
1002
undefined
```



การใช้งาน Editor mode

```
$ node
> .editor
// Entering editor mode (^D to finish, ^C to cancel)
function welcome(name) {
 return `Hello ${name}!`;
welcome('Node.js User');
// ^D
'Hello Node.js User!'
>
```



การใช้งาน Feature NodeJS ใน REPL

```
$ node
> const a = [1, 2, 3];
undefined
> a
[1, 2, 3]
> a.forEach((v) => {
... console.log(v);
... });
```



REPL Commands

- ctrl + c terminate the current command.
- ctrl + c twice terminate the Node REPL.
- ctrl + d terminate the Node REPL.
- **Up/Down Keys** see command history and modify previous commands.
- tab Keys list of current commands.
- .help list of all commands.
- .break exit from multiline expression.
- .clear exit from multiline expression.
- .save filename save the current Node REPL session to a file.
- .load filename load file content in current Node REPL session.

