

Node.js Global Object

JavaScript hanyalah bahasa pemrograman. Ia tidak mengetahui apakah Anda menjalankannya menggunakan browser atau Node.js. Di browser, JavaScript dapat mengontrol fungsionalitas browser seperti mengunjungi halaman, memuat ulang, menutup tabs, serta menampilkan *alert dialog*. JavaScript mampu melakukan itu karena browser menambahkan objek `window` pada JavaScript.

Di Node.js pun demikian, ia menambahkan objek `global` guna memberikan fungsionalitas lebih pada JavaScript. Hal ini bertujuan untuk mendukung pengembangan pada environment-nya. Contoh, melalui objek `global` kita dapat melihat berapa CPU yang digunakan pada komputer, modularisasi berkas JavaScript, menampilkan nilai pada console, dan hal lainnya.

Objek `window` pada browser dan objek `global` pada Node.js merupakan *Global Object*. Seluruh fungsi atau properti yang menjadi member dari `global object` dapat diakses di mana saja alias memiliki cakupan global. Pada Node.js Anda bisa melihat apa saja yang termasuk member dari `global object` dengan menggunakan kode berikut:

```
1. Object.getOwnPropertyNames(global);
```

Coba jalankan pada REPL. Ia akan mengembalikan semua *member*-nya.

```
> Object.getOwnPropertyNames(global);
[
  'Object',           'Function',           'Array',
  'Number',           'parseFloat',         'parseInt',
  'Infinity',         'NaN',                'undefined',
  'Boolean',          'String',              'Symbol',
  'Date',              'Promise',             'RegExp',
  'Error',             'EvalError',          'RangeError',
  'ReferenceError',   'SyntaxError',         'TypeError',
  'URIError',         'globalThis',          'JSON',
  'Math',              'console',             'Intl',
  'ArrayBuffer',      'Uint8Array',          'Int8Array',
  'Uint16Array',      'Int16Array',          'Uint32Array',
  'Int32Array',       'Float32Array',        'Float64Array',
  'Uint8ClampedArray', 'BigUint64Array',      'BigInt64Array',
  'DataView',         'Map',                 'BigInt',
  'Set',              'WeakMap',             'WeakSet',
  'Proxy',            'Reflect',             'decodeURI',
  'decodeURIComponent', 'encodeURIComponent', 'encodeURIComponent',
  'escape',           'unescape',            'eval',
  'isFinite',         'isNaN',               'SharedArrayBuffer',
  'Atomics',          'FinalizationRegistry', 'WeakRef',
  'WebAssembly',      'global',              'process',
  'Buffer',           'URL',                 'URLSearchParams',
]
```

Banyak sekali yah member dari global objek. Namun dilansir dari website Node.js, sebenarnya mereka hanya menambahkan beberapa objek saja. Objek tersebut dinamakan dengan 'true globals'. [2]

Berikut adalah daftarnya:

- **global** : Global namespace. Member apa pun di dalam object ini dapat diakses pada cakupan global.
- **process** : menyediakan interaksi dengan proses Node.js yang berjalan.
- **console** : menyediakan berbagai fungsionalitas **STDIO**.
- **setTimeout**, **clearTimeout**, **setInterval**, **clearInterval**.

Ada juga objek yang merupakan 'pseudo-globals' atau objek global semu. Objek ini tidak terlihat bila dicetak menggunakan `Object.getOwnPropertyNames(global)` sebab ia bukan member langsung dari objek global, melainkan diturunkan dari cakupan module. Karena pada Node.js semua berkas JavaScript adalah module [3], jadi pseudo-globals dapat diakses layaknya global objek. Berikut adalah daftarnya:

- `module` : digunakan untuk sistem modularisasi pada Node.js.
- `__filename` : keyword untuk mendapatkan lokasi berkas JavaScript yang dieksekusi. Keyword ini tidak tersedia pada Node.js REPL.
- `__dirname` : keyword untuk mendapatkan root directory dari berkas JavaScript yang dieksekusi.
- `require` : digunakan untuk mengimpor module JavaScript.