



สูตรโกงการใช้ **JavaScript** และ **Node.js**

โดย นครโค้ด (NakornCode)

(ห้ามแจกฟรี! สำหรับผู้ซื้อคอร์สเรียนบน SkillLane เท่านั้น)

ติดต่อผู้สอน <https://m.me/nakorncode>



เว็บไซต์แนะนำสำหรับเรียนรู้ต่อ

W3Schools, MDN Web Docs, JavaScript Awesome,  
NodeJS Awesome, JavaScript.info

## ประเภทข้อมูลภาพรวม

Number	<code>let x = 1</code> <code>let y = 1.5</code>
String	<code>let x = 'abc'</code> <code>let y = "abc"</code>
Boolean	<code>let x = true</code> <code>let y = false</code>
Undefined	<code>let x // x is undefined</code>
Null	<code>let x = null</code>
Array	<code>let x = ['a', 'b', 'c']</code>
Object	<code>let x = { a: 1, b: 2, c: 3 }</code>
Function	<code>let x = function () {}</code> <code>let y = () =&gt; {}</code>
ตรวจสอบ	<code>let x = 'abc'</code> <code>let y = true</code> <code>typeof x // 'string'</code> <code>typeof y // 'boolean'</code>

## พื้นฐานการใช้งาน

Internal	<code>&lt;script&gt;// Write me&lt;/script&gt;</code>
External	<code>&lt;script src="run.js"&gt;&lt;/script&gt;</code> <code>&lt;script defer</code> <code>src="after-load.js"&gt;&lt;/script&gt;</code>
Output (1)	<code>console.log('Hello world')</code>
Output (2)	<code>alert('Hello world')</code>
Prompt	<code>prompt('Your name:')</code>
Confirm	<code>confirm('Are you sure?')</code>
Comment	<code>// one line</code> <code>/* multiple line */</code>
Variable	<code>let x = 10</code> <code>let y = 20, z = 30</code>
Constant	<code>const birthYear = 1992</code>

## ตัวดำเนินการ

คณิตศาสตร์	<code>let x = 5, y = 2</code> <code>x + y // 7 (บวก)</code> <code>x - y // 3 (ลบ)</code> <code>x * y // 10 (คูณ)</code> <code>x / y // 2.5 (หาร)</code> <code>x ** y // 25 (ยกกำลัง)</code> <code>x % y // 1 (หารเอาเศษ)</code>
เพิ่มและลด จำนวน	<code>let x = 10, y = 10</code> <code>x++ // 11</code> <code>y-- // 9</code>
เปลี่ยน แปลงค่า	<code>let x = 10, y = 10, z = 10</code> <code>x = x + 5 // 15</code> <code>y += 5 // 15</code> <code>z -= 5 // 5</code>
คำนวณตาม วงเล็บ	<code>4 + 1 * 2 // 6</code> <code>(4 + 1) * 2 // 10</code>

## เงื่อนไข

if	<pre>if (x &gt; 5) {   // x มากกว่า 5 }</pre>
if - else	<pre>if (x &gt; 5) {   // x มากกว่า 5 } else {   // x น้อยกว่าหรือเท่ากับ 5 (เงื่อนไขอื่นๆ) }</pre>
if - else if	<pre>if (x == 1) {   // x มีค่าเท่ากับ 1 } else if (x == 2) {   // x มีค่าเท่ากับ 2 } else {   // x มีค่าอื่นๆที่ไม่เท่ากับข้างต้นทั้งหมด }</pre>
Shorthand	<pre>if (x &gt; 5) {   'Yes' } else {   'No' }  x &gt; 5 ? 'Yes' : 'No'</pre>
Switch-case	<pre>switch x {   case 1:     // x มีค่าเท่ากับ 1     break   case 2:   case 5:     // x มีค่าเท่ากับ 2 หรือ 5     break   default:     // x มีค่าเท่ากับอย่างอื่น }</pre>
Default value	<pre>let x = 5, y = 0, z // (ตรวจแบบ Boolean()) x    'd' // 5 y    'd' // 'd' z    'd' // 'd' // (ตรวจเฉพาะ undefined) x ?? 'd' // 5 y ?? 'd' // 0 z ?? 'd' // 'd'</pre>

## การเปรียบเทียบและตรรกศาสตร์

การเปรียบเทียบ	<pre>let x = 5, y = 2, z = 5 x &lt; y // false x &lt;= y // false x &lt;= z // true x &gt; y // true x &gt; z // false x &gt;= z // true x == y // false x == z // true x != y // true x != z // false</pre>
เปรียบเทียบแบบ Strict	<pre>let x = 10, y = '10' x == y // true x === y // false x != y // false x !== y // true</pre>
ตรรกศาสตร์	<pre>let t = true, f = false t &amp;&amp; t // true t &amp;&amp; f // false f &amp;&amp; f // false t    t // true t    f // true f    f // false !t // false !f // true</pre>

## วนจำนวนรอบ (1)

ลูป while	<pre>let n = 1 while (n &lt;= 3) {   console.log(n)   n++ } // 1 2 3</pre>
ลูป do while	<pre>let m do {   m = prompt('Exit?') } while (m != 'yes')</pre>
ลูป for	<pre>for (let n = 1; n &lt;= 3; n++) {   console.log(n) } // 1 2 3</pre>

วนจำนวนรอบ (2)	
ซ้อนลูป	<pre>let str = '' for (let n = 1; n &lt;= 3; n++) {   for (let m = 1; m &lt;= 3; m++) {     str += `\${m * n}`   }   str += '\n' } console.log(str) // 1 2 3 // 2 4 6 // 3 6 9</pre>
หยุดลูป	<pre>for (let n = 1; n &lt;= 9; n++) {   if (n == 5) break   console.log(n) } // 1 2 3 4</pre>
ข้ามลูป	<pre>for (let n = 1; n &lt;= 9; n++) {   if (n % 2 == 0) continue   console.log(n) } // 1 3 5 7 9</pre>
ลูป for of	<pre>let arr = [1, 2, 3, 4, 5] for (let item of arr) {   console.log(item * 2) } // 2 4 6 8 10</pre>
ลูป for in	<pre>let obj = { a: 1, b: 2, c: 3 } for (let k in obj) {   console.log(`\${k}=\${obj[k]}`) } // a=1 b=2 c=3</pre>
ลูป for of (Entries)	<pre>let obj = { a: 1, b: 2, c: 3 } for (let [k, v] of Object.entries(obj)) {   console.log(`\${k}=\${v}`) } // a=1 b=2 c=3</pre>

Boolean	
แปลงค่า	<pre>Boolean(0) // false Boolean(1) // true Boolean(-1) // true Boolean('') // false Boolean('abc') // true Boolean({}) // true Boolean(undefined) // false</pre>

Function	
Function	<pre>function tell (name, age) {   return `\${name} (\${age})` } tell('John', 18) // 'John (18)'</pre>
Expression	<pre>const tell = function (name, age) {   return `\${name} (\${age})` } tell('John', 18) // 'John (18)'</pre>
Arrow	<pre>const tell = (name, age) =&gt; {   return `\${name} (\${age})` } tell('John', 18) // 'John (18)'</pre>
Default value	<pre>function hi (to = 'John') {   return `Hi \${to}` } hi() // 'Hi John' hi('Joe') // 'Hi Joe'</pre>
Rest params	<pre>function over5 (...arr) {   return arr.filter(v =&gt; v &gt; 5) } over5(4, 5, 6, 7, 8) // [6, 7, 8]</pre>
Spread params	<pre>let arr1 = [1, 2] let arr2 = [3, 4] [arr1, arr2] // [[1, 2], [3, 4]] [...arr1, ...arr2] // [1, 2, 3, 4]</pre>

Number (1)	
แปลงค่า	<pre>Number('123') // 123 Number('00123') // 123 Number('AB123') // NaN parseInt('123') // 123 parseInt('00123') // 123 parseInt('AB123') // NaN</pre>
จำนวนทศนิยม	<pre>let num = 2.15986 num.toFixed(2) // '2.16' num.toFixed(1) // '2.' num.toFixed(0) // '2'</pre>

Number (2)	
แสดงเลข พร้อมลูกน้ำ	<pre>let num = 10250990 num.toLocaleString('th-TH') // '10,250,990'</pre>
แสดงเลข ตามหน่วย	<pre>let n = 9.85 let config = {   style: 'unit',   unit: 'kilogram',   unitDisplay: 'long' } n.toLocaleString('th-TH', config) // '9.85 กิโลกรัม'</pre>
แสดงเลข ตามค่าเงิน	<pre>let n = 3980 let config = {   style: 'currency',   currency: 'thb',   currencyDisplay: 'name' } n.toLocaleString('th-TH', config) // 3,980.00 บาทไทย</pre>

String (1)	
รวมข้อความ	<pre>let n1 = 10, n2 = 20 let s1 = '10', n2 = '20' n1 + n2 // 30 s1 + s2 // '1020'</pre>
Template literals	<pre>let name = 'John' let age = 18 `\${name} (\${age})` // John (18)</pre>
Escape charcter	<pre>'1. Apple\n2. Banana' // เว้นวรรค 'I\'m coder' // 'I'm coder' 'C:\\Games\\A' // 'C:\Games\A'</pre>
ขนาด ข้อความ	<pre>'abc'.length // 3 'my name'.length // 7</pre>
ตัวพิมพ์เล็ก- ใหญ่	<pre>'Abc'.toLowerCase() // 'abc' 'Abc'.toUpperCase() // 'ABC'</pre>
ตำแหน่ง ข้อความ	<pre>let str = 'abcdef' str[2] // 'c' str.charAt(2) // 'c'</pre>

String (2)	
แทนที่ ข้อความ	<pre>let str = 'My dog name doggy' str.replace('dog', 'cat') // My cat name doggy str.replace(/dog/g, 'cat') // My cat name catgy</pre>
ค้นหา ข้อความ	<pre>let str = 'http://google.com' str.includes('google') // true str.includes('facebook') // false str.startsWith('http') // true str.startsWith('google') // false str.endsWith('.com') // true str.endsWith('google') // false str.indexOf('http') // 0 str.indexOf('google') // 7 str.indexOf('facebook') // -1</pre>
แทรก ข้อความให้ ครบ	<pre>'123'.padStart(3, '0') // '123' '1'.padStart(3, '0') // '001' '123'.padEnd(3, '0') // '123' '1'.padEnd(3, '0') // '100'</pre>
ตัดส่วนเกิน	<pre>let str = ' Apple ' str // ' Apple ' str.trim() // 'Apple'</pre>
ตัดข้อความ	<pre>let str = 'My name is John' str.slice(3) // 'name is John' str.slice(-4) // 'John' str.slice(3, 7) // 'name' str.substr(3) // 'name is John' str.substr(-4) // 'John' str.substr(3, 4) // 'name'</pre>
ทำซ้ำ	<pre>let str = 'A' str.repeat(3) // 'AAA'</pre>
แยกส่วน ประกอบ	<pre>'abcdef'.split('') // ['a', 'b', 'c', 'd', 'e', 'f'] 'a,b,c,1,2,3'.split(',') // ['a', 'b', 'c', '1', '2', '3']</pre>

Array (1)	
เรียก Index	<pre>let arr = ['a', 'b', 'c'] arr[0] // 'a' arr[1] // 'b'</pre>
เปลี่ยนแปลงค่า	<pre>let arr = ['a', 'b', 'c'] arr[0] = 'z' // ['z', 'b', 'c']</pre>
หลายมิติ	<pre>let arr = [[1, 2], [3, 4]] arr[0] // [1, 2] arr[0][0] // 1</pre>
ขนาด	<pre>let arr = ['a', 'b', 'c'] arr.length // 3</pre>
ตรวจสอบ	<pre>let arr = ['a', 'b', 'c'] let str = 'abc' Array.isArray(arr) // true Array.isArray(str) // false</pre>
สร้างตามจำนวน	<pre>Array.from({ length: 7 }, (k, v) =&gt; v) // [0, 1, 2, 3, 4, 5, 6]</pre>
รวม	<pre>let arr1 = [1, 2] let arr2 = [3, 4] arr1.concat(arr2) // [1, 2, 3, 4]</pre>
เพิ่ม	<pre>let arr = ['a'] let len1 = arr.push('z') len1 // 2 arr // ['a', 'z']  let len2 = arr.unshift('y') len2 // 3 arr // ['y', 'a', 'z']</pre>
ลบ	<pre>let arr = ['a', 'b', 'c', 'd'] let rm1 = arr.pop() rm1 // 'd' arr // ['a', 'b', 'c']  let rm2 = arr.shift() rm2 // 'a' arr // ['b', 'c']</pre>
ค้นหา	<pre>let arr = ['a', 'b', 'c', 'd', 'e'] arr.includes('a') // true arr.includes('z') // false arr.indexOf('a') // 0 arr.indexOf('z') // -1</pre>

Array (2)	
ตัดแบบไม่เปลี่ยนข้อมูล	<pre>let arr = ['a', 'b', 'c', 'd', 'e'] arr.slice(2) // ['c', 'd', 'e'] arr.slice(-2) // ['d', 'e'] arr.slice(1, 4) // ['b', 'c', 'd'] arr // ['a', 'b', 'c', 'd', 'e']</pre>
ตัดแบบเปลี่ยนข้อมูล	<pre>let arr = ['a', 'b', 'c', 'd', 'e'] arr.splice(1, 3) // ['b', 'c', 'd'] arr // ['a', 'e']</pre>
รวมเป็น String	<pre>let arr = ['a', 'b', 'c', 'd', 'e'] arr.join('') // 'abcde' arr.join(',') // 'a,b,c,d,e' arr.join('\n') // 'a\nb\nc\nd\ne'</pre>
ค้นหาเนื้อหา	<pre>let arr = [2, 9, 1, 8, 3] let over4 = (v) =&gt; v &gt; 4 let over9 = (v) =&gt; v &gt; 9 arr.find(over4) // 9 arr.find(over9) // undefined arr.findIndex(over4) // 1 arr.findIndex(over9) // -1</pre>
คัดกรองเนื้อหา	<pre>let arr = [2, 9, 1, 8, 3] let over4 = (v) =&gt; v &gt; 4 arr.filter(over4) // [9, 8]</pre>
ตรวจสอบเนื้อหา	<pre>let arr = [8, 5, 7, 6, 9] let over4 = (v) =&gt; v &gt; 4 let over6 = (v) =&gt; v &gt; 6 arr.every(over4) // true arr.every(over6) // false arr.some(over4) // true arr.some(over6) // true</pre>
เปลี่ยนแปลงเนื้อหา	<pre>let arr = [1, 2, 3] let double = (v) =&gt; v * 2 arr.map(double) // [2, 4, 6]</pre>
ลดลำดับชั้น	<pre>let arr = [[1, 2], [3, 4]] arr.flat() // [1, 2, 3, 4]</pre>
จัดเรียง	<pre>let arr = [3, 1, 2, 4] arr.sort() // [1, 2, 3, 4] arr.reverse() // [4, 3, 2, 1]</pre>
ลดรูป	<pre>let arr = [1, 2, 3, 4, 5] let sum = (acc, cur) =&gt; acc + cur arr.reduce(sum, 0) // 15</pre>

Object, JSON	
เรียกคีย์	<pre>let obj = { name: 'John', age: 6 } obj.name // 'John' obj['name'] // 'John' obj?.name // แก้ปัญหาเพื่อเจอ undefined</pre>
แก้ไข เนื้อหา	<pre>let obj = { name: 'John', age: 6 } obj.age = 7 obj // { name: 'John', age: 7 }</pre>
เพิ่มคีย์	<pre>let obj = { a: 1, b: 2 } obj.c = 3 obj // { a: 1, b: 2, c: 3 }</pre>
แทรก ข้อมูล	<pre>let o1 = { name: 'John', age: 6 } let o2 = { age: 8, death: false } Object.assign(o1, o2) // { name: 'John', age: 8, death: false }</pre>
เรียกคีย์	<pre>let obj = { a: 1, b: 2, c: 3 } Object.keys(obj) // ['a', 'b', 'c']</pre>
เรียกค่า	<pre>let obj = { a: 1, b: 2, c: 3 } Object.values(obj) // [1, 2, 3]</pre>
แปลง เป็น Array	<pre>let obj = { a: 1, b: 2, c: 3 } Object.entries(obj) // [['a', 1], ['b', 2], ['c', 3]]</pre>
แปลง กลับเป็น Object	<pre>let arr = [['a', 1], ['b', 2]] Object.fromEntries(arr) // { a: 1, b: 2 }</pre>
String to JSON	<pre>let str = '{"a": "1", "b": "2"}' JSON.parse(str) // { a: '1', b: '2' }</pre>
JSON to String	<pre>let obj = { a: '1', b: '2' } JSON.stringify(obj) // '{"a": "1", "b": "2"}'</pre>

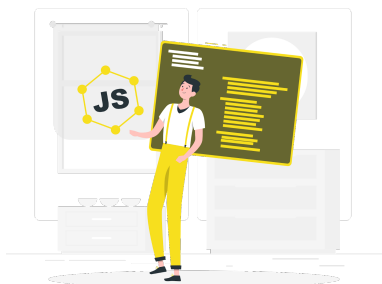


Illustration by storyset.com

Destructuring Assignment	
แยก Array	<pre>let [a, b, c] = [   'apple',   'banana',   'carrot' ] console.log(a) // 'apple'</pre>
แยก Object	<pre>let { age } = {   name: 'John',   age: 18 } console.log(age) // 18</pre>

ตัวอย่าง Handle Error
<pre>function shouldBeZero (n) {   if (n !== 0) {     throw new Error('Not zero')   }   return 'Ok' }  try {   shouldBeZero(0) // 'Ok'   shouldBeZero(1) // throw... } catch (err) {   console.error(err) // 'Not zero' } /* finally { ... } */</pre>

ตัวอย่าง Callback
<pre>function shouldBeZero (n, cb) {   if (n !== 0) {     cb(new Error('Not zero'))   } else {     cb(null, 'Ok')   } }  function handler (err, message) {   if (err) {     console.error(err)   } else {     console.log(message)   } }  shouldBeZero(0, handler) // 'Ok' shouldBeZero(1, handler) // 'Not zero'</pre>

## ตัวอย่าง Promise และ Async/Await

```
function shouldBeZero (n, cb) {
  return new Promise((resolve, reject) => {
    if (n !== 0) {
      // ความหมายเหมือน Promise.reject(...)
      reject(new Error('Not zero'))
    } else {
      // ความหมายเหมือน Promise.resolve(...)
      resolve('Ok')
    }
  })
}
```

```
function whenOk (message) {
  console.log(message)
}
```

```
function whenErr (err) {
  console.error(err)
}
```

```
shouldBeZero(0)
  .then(whenOk) // 'Ok'
  .catch(whenErr)
  /* .finally(...) */
```

```
shouldBeZero(1)
  .then(whenOk)
  .catch(whenErr) // 'Not zero'
  /* .finally(...) */
```

```
async function runAsync () {
  try {
    await shouldBeZero(0) // 'Ok'
    await shouldBeZero(1) // throw...
  } catch (err) {
    console.error(err) // 'Not zero'
  }
}

runAsync() // ต้องเรียกเพื่อรัน Async Function
```

## Promise

Resolve	<code>Promise.resolve('Value')</code>
Reject	<code>Promise.reject('Reason error')</code>
เริ่มทั้งหมด (แบบห้ามมี Error)	<pre>const p1 = Promise.resolve('1') const p2 = Promise.resolve('2') const p3 = Promise.resolve('3')  Promise.all([p1, p2, p3])   .then(console.log) // [1, 2, 3]   .catch(console.error)</pre>
เริ่มทั้งหมด (แบบยินยอม Error ได้)	<pre>const p1 = Promise.resolve('1') const p2 = Promise.reject('2') const p3 = Promise.resolve('3')  Promise.allSettled([p1, p2, p3])   .then(console.log)  /* ผลลัพธ์ที่ได้ลงใน .then() { status: 'fulfilled', value: 1 } { status: 'rejected', reason: 2 } { status: 'fulfilled', value: 3 } */</pre>
เริ่มทั้งหมด แบบอื่นๆ	<pre>// เลือกเฉพาะตัวที่เสร็จก่อนลงใน // .then() ถ้ามี Error ลงเข้าสู่ .catch() Promise.race([p1, p2, p3])  // เลือกเฉพาะตัวที่เสร็จก่อนลงใน // .then() โดยไม่สนใจ Error Promise.any([p1, p2, p3])</pre>



Object-oriented (1)	
Constructor	<code>new MyObject(...args)</code>
Function แบบ OOP	<pre>function MyObject(name, age) {   this.name = name   this.age = age   this.showName = () =&gt; {     return `Name: \${this.name}`   } }</pre> <pre>let a = new MyObject('John', 18) let b = new MyObject('Joe', 15)</pre> <pre>a.showName() // 'Name: John' b.showName() // 'Name: Joe'</pre>
ใช้ Class แทน Function	<pre>class MyObject {   constructor (name, age) {     this.name = name     this.age = age   }   showName () {     return `Name: \${this.name}`   } }</pre> <pre>let a = new MyObject('John', 18) let b = new MyObject('Joe', 15)</pre> <pre>a.showName() // 'Name: John' b.showName() // 'Name: Joe'</pre>
Getter/ Setter	<pre>class Language {   _list = []   set list (value) {     this._list.push(value)   }   get list () {     return this._list.join('/')   } }</pre> <pre>const lang = new Language() lang.list = 'TH' lang.list = 'EN' lang.list = 'CN' lang.list // 'TH/EN/CN'</pre>

Object-oriented (2)	
Extends	<pre>class Country {   country = 'Thailand'   showInfo() {     return this   } }</pre> <pre>class Language extends Country {   setLanguage (value) {     this.language = value   } }</pre> <pre>class Province extends Country {   provinces = []   addProvince (value) {     this.provinces.push(value)   } }</pre> <pre>const lang = new Language() lang.setLanguage('Thai') lang.showInfo() /* {   country: 'Thailand',   language: 'Thai' } */</pre> <pre>const prov = new Province() prov.addProvince('BKK') prov.addProvince('NTB') prov.showInfo() /* {   country: 'Thailand',   provinces: ['BKK', 'NTB'] } */</pre>
Override Method	<pre>class Main {   show () { return 'A' } }</pre> <pre>class Second extends Main {   show () { return 'B' } }</pre> <pre>let a = new Main() a.show() // 'A' let b = new Second() b.show() // 'B'</pre>



Object-oriented (3)	
Super	<pre> class Main {   constructor () {     this.data = 'MAIN'   } } class Second extends Main {   constructor (custom) {     super() // จำเป็นต้องเรียก     this.data = custom   } }  let a = new Main() a.data // 'MAIN' let b = new Second('NEW') b.data // 'NEW' </pre>
Static	<pre> class SimpleMath {   static plus (x, y) {     return x + y   }   static minus (x, y) {     return x - y   } }  SimpleMath.plus(5, 3) // 8 SimpleMath.minus(5, 3) // 2 </pre>
Prototype	<pre> let arr = ['a', 'b', 'c'] arr.toString() // 'a,b,c'  const slash = function () {   return this.join('/') } Array.prototype.toString = slash  arr.toString() // 'a/b/c' </pre>

Binding
<pre> function show (message) {   return `\${message}: \${this.name}` }  show('Hello') // 'Hello: undefined'  show.call({ name: 'John' }, 'Hello') // 'Hello: John'  let f = show.bind({ name: 'John' }, 'Hello') f() // 'Hello: John' </pre>

Chaining
<pre> class Draw {   setWidth (value) {     this.width = value     return this   }   setHeight (value) {     this.height = value     return this   }   beginDraw (animal) {     return `\${this.width}x\${this.height}       : \${animal}`   } }  const draw = new Draw() draw   .setWidth(1920)   .setHeight(1080)   .beginDraw('Dog') // '1920x1080: Dog' </pre>

Document Object Model (1)	
Document	<code>window.document</code> // หรือ <code>document</code>
Get element	<code>document.getElementById('value')</code> <code>getElementById('title')</code> // <code>&lt;element id="title"&gt;</code>  <code>getElementsByClassName('text')</code> // <code>&lt;element class="text"&gt;</code>  <code>getElementsByName('email')</code> // <code>&lt;input name="email"&gt;</code>  <code>getElementsByTagName('h1')</code> // <code>&lt;h1&gt;</code>
Query selector	<code>document.querySelector('value')</code>  <code>querySelector('#title')</code> // <code>&lt;element id="title"&gt;</code>  <code>querySelector('.text')</code> // <code>&lt;element class="text"&gt;</code>  <code>querySelector('[name="email"]')</code> // <code>&lt;input name="email"&gt;</code>  <code>querySelector('h1')</code> // <code>&lt;h1&gt;</code> ได้แบบขึ้นเดียว  <code>querySelectorAll('h1')</code> // <code>&lt;h1&gt;</code> ได้แบบหลายขึ้น
Text	<code>document</code> <code>.getElementById('title')</code> <code>.textContent = 'New Text'</code>  <code>.textContent</code> // ข้อความทั้งหมด <code>.innerText</code> // ข้อความ Child <code>.innerHTML</code> // HTML Child <code>.outerHTML</code> // HTML Parent
Style (Inline)	<code>document</code> <code>.getElementById('title')</code> <code>.style</code> <code>.borderColor = '#ff0000'</code>  // <code>&lt;element id="title"</code> // <code>style="border-color: #ff0000"&gt;</code>

Document Object Model (2)	
Body	<code>document.body</code> // <code>&lt;body&gt;</code>
Head	<code>document.head</code> // <code>&lt;head&gt;</code>
Forms	<code>document.forms</code> // <code>&lt;form&gt;</code> ทั้งหมด
Images	<code>document.images</code> // <code>&lt;img&gt;</code> ทั้งหมด
Links	<code>document.links</code> // <code>&lt;a href&gt;</code> ทั้งหมด
รายการ Class	<code>document</code> <code>.querySelector('.title')</code> <code>.classList</code> <code>.add('red-text')</code>  // <code>&lt;element class="title</code> // <code>red-text"&gt;</code>  <code>.add()</code> // เพิ่ม Class <code>.remove()</code> // ลบ Class <code>.toggle()</code> // สลับเปลี่ยน Class <code>.contains()</code> // มี Class หรือไม่ <code>.entries()</code> // Class ที่มีทั้งหมด
Dataset	<code>document</code> <code>.getElementById('title')</code> <code>.dataset</code> <code>.userId</code>  // <code>&lt;element id="title"</code> // <code>data-user-id="???"&gt;</code>
ตำแหน่ง เนื้อหา	<code>document.body.clientHeight</code>  // เทียบภายใน <code>.clientTop</code> // ขอบบน <code>.clientLeft</code> // ขอบซ้าย <code>.clientWidth</code> // ความกว้างเนื้อหา <code>.clientHeight</code> // ส่วนสูงเนื้อหา  // เทียบภายนอก <code>.offsetTop</code> // ขอบบน <code>.offsetLeft</code> // ขอบซ้าย <code>.offsetWidth</code> // ความกว้างเนื้อหา <code>.offsetHeight</code> // ส่วนสูงเนื้อหา  <code>.scrollTop</code> // ระยะห่าง Scroll บน <code>.scrollHeight</code> // ความสูงทั้งหมด

Document Object Model (3)	
สร้างและแทรก Node/Element	<pre>let element = document.createElement('h1')  element.innerText = 'New Title' element.classList.add('red-text')  document.getElementById('title') .append(element)  .append() // แทรกหลัง .prepend() // แทรกหน้า .remove() // ลบ</pre>
Event	<pre>let a = document.querySelector('a.custom-link')  function fn () {   // โค้ดที่ต้องการ Callback เมื่อคลิก }  // เพิ่ม (เดิม on นำหน้าของชื่อ Event) a.onclick = fn // เพิ่ม (เรียกตามชื่อ และ Callback) a.addEventListener('click', fn) // ลบ (Callback ต้องตัวเดียวกัน) a.removeEventListener('click', fn)</pre>

Events	
Methods	<pre>event.preventDefault() event.stopPropagation()</pre>
คลิก	<pre>click, dblclick, contextmenu, wheel</pre>
เมาส์	<pre>mousedown, mouseenter, mouseleave, mouseleave, mousemove, mouseout, mouseover, mouseup, mousewheel</pre>
จอสัมผัส	<pre>touchcancel, touchend, touchmove, touchstart</pre>
กดปุ่มคีย์บอร์ด	<pre>keydown, keypress, keyup</pre>
แบบฟอร์ม	<pre>change, input, submit, reset, blur, focus, focusin, focusout</pre>

Date (1)	
Construct	<pre>new Date() // วันเวลาปัจจุบัน new Date('2022-09-25') new Date('2022-05-01T21:30:00')</pre>
Get	<pre>let d = new Date() d.getDate() // วันที่ปัจจุบัน d.getDay() // วัน (0 คือวันอา.) d.getDate() // วันที่ d.getMonth() // เดือน (ต้อง +1) d.getFullYear() // ปี d.getTime() // เวลา (Unix) d.getMilliseconds() // มิลลิวินาที d.getSeconds() // วินาที d.getMinutes() // นาที d.getHours() // ชั่วโมง d.getTimezoneOffset() // ห่าง UTC</pre>
Set	<pre>let d = new Date() d.setDate(15) // เปลี่ยนเป็นวันที่ 15  .setDate() // วันที่ .setMonth() // เดือน (ต้อง +1) .setFullYear() // ปี .setMilliseconds() // มิลลิวินาที .setSeconds() // วินาที .setMinutes() // นาที .setHours() // ชั่วโมง</pre>
แสดงผลภาษาไทย	<pre>let d = new Date() d.toLocaleString('th-TH')</pre>
รูปแบบวัน	<pre>let conf = {   dateStyle: 'short'   // full, long, medium, short } d.toLocaleString('th-TH', conf)</pre>
รูปแบบเวลา	<pre>let conf = {   timeStyle: 'short'   // full, long, medium, short } d.toLocaleString('th-TH', conf)</pre>
ระบบปฏิทิน	<pre>let conf = {   calendar: 'iso8601'   // buddhist, iso8601 } d.toLocaleString('th-TH', conf)</pre>

Date (2)	
Timzone	<pre>let conf = {   timeZone: 'Asia/Bangkok'   // ดูที่เว็บ iana.org/time-zones } d.toLocaleString('th-TH', conf)</pre>
หาความต่างเวลา	<pre>let a = new Date('2022-01-15') let b = new Date('2022-01-14') console.log(a - b) // 86400000  // 86400000 ÷ 1000 (มิลลิวินาที) ÷ 60 (วินาที) ÷ 60 (ชั่วโมง) = 24</pre>

Math	
ค่าพาย	<code>Math.PI</code> // 3.14159 โดยประมาณ
ค่าสัมบูรณ์	<pre>Math.abs(-2) // 2 Math.abs(2) // 2</pre>
ถอดรากที่ 2	<code>Math.sqrt(16)</code> // 4
ตรีโกณมิติ	<pre>Math.sin() Math.cos() Math.tan()</pre>
ปัดเศษทศนิยม	<pre>Math.round(3.4) // 3 Math.round(3.5) // 4 Math.round(3.6) // 4 // ใช้ Number.toFixed(0) แทนได้  let a = 1.5, b = -1.5 Math.ceil(a) // 2 (ปัดขึ้นเท่านั้น) Math.ceil(b) // -1 (ปัดขึ้นเท่านั้น) Math.floor(a) // 1 (ปัดลงเท่านั้น) Math.floor(b) // -2 (ปัดลงเท่านั้น) Math.trunc(a) // 1 (ลบเศษ) Math.trunc(b) // -1 (ลบเศษ)</pre>
หาค่าน้อยสุด/มากที่สุด	<pre>let arr = [2, 3, 5, 1, 4] Math.min(...arr) // 1 Math.max(...arr) // 5</pre>
สุ่มจำนวน	<pre>Math.random() // 0 - 1 (ทศนิยม) function rand (min, max) {   return Math.random() *     (max - min) + min; } rand(1, 6) // 1 - 6 (จำนวนเต็ม)</pre>

URL	
<a href="https://blog.myweb.com:8080/code/2022/javascript.html?tab=beginner&amp;page=4#example">https://blog.myweb.com:8080/code/2022/javascript.html?tab=beginner&amp;page=4#example</a>	
<ul style="list-style-type: none"> <li>• Scheme (Protocol)</li> <li>• Host</li> <li>• Subdomain</li> <li>• Second-level domain</li> <li>• Top-level domain</li> <li>• Port</li> <li>• Subdirectory (Pathname)</li> <li>• File</li> <li>• Query string</li> <li>• Fragment</li> </ul>	
Window	<code>window.location</code>
Construct	<code>new URL('http://www.google.com')</code>
Properties	<pre>let x = 'http://abc.com/d?q=z#id' let u = new URL(x) u.protocol // 'http:' u.hostname // 'abc.com' u.pathname // '/d' u.search // '?q=z' u.hash // '#id' u.href // http://abc.com/d?q=z#id</pre>
Search params	<pre>let u = new URL('http://abc.com') let sp = u.searchParams sp.append('brand', 'Nike') sp.append('sort', 'low-to-high')  u.toString() // http://abc.com/?brand= // Nike&amp;sort=low-to-high  .append(key, value) // เพิ่ม .get(key) // เรียก .getAll(key) // เรียกทั้งหมด .set(key, value) // แก้ไข .has(key) // ต้องมี .delete(key) // ลบ .sort() // จัดเรียง .key() // เฉพาะคีย์ .entries() // Entries</pre>

File Reader	
Method	<code>URL.createObjectURL(object)</code>
ตัวอย่าง พรีวิว ภาพหลัง เลือกไฟล์	<pre>let preview = document     .querySelector('img')  let fileInput = 'input[type="file"]' let input = document     .querySelector(fileInput)  let cb = function (e) {     let file = e.target.files[0]     preview.src = URL         .createObjectURL(file) }  input.addEventListener('input', cb)</pre>

Storage	
Objects	<code>window.localStorage</code> <code>window.sessionStorage</code>
Methods	<pre>localStorage.setItem('u', 'admin') // ตั้งให้คีย์ u มีค่าเท่ากับ 'admin'  .key(n)           // เรียกตาม Index .getItem(key)     // เรียกตาม Key .setItem(key, value) // แก้ไข Key .removeItem(key)  // ลบ Key .clear()          // ลบทั้งหมด</pre>

Geolocation	
Object	<code>window.navigator.geolocation</code>
Method	<pre>let success = ({ coords }) =&gt; {     coords.latitude // ละติจูด     coords.longitude // ลองจิจูด     coords.accuracy // ความแม่นยำ (เมตร) } let error = (err) =&gt; {     err // Error object } navigator.getCurrentctPosition(     success, error )</pre>

Fetch	
Fetch	<pre>let url = 'http://...' fetch(url) // return Promise</pre>
ตัวอย่าง เรียกเป็น JSON	<pre>fetch(url)     .then(res =&gt; res.json())     .then(data =&gt; { ... })</pre>
ตัวอย่าง POST	<pre>let data = {     username: 'admin',     password: '1234' } let config = {     method: 'POST',     body: JSON.stringify(data) } fetch(url, config)     .then(...)</pre>
ตัวอย่าง อัปโหลด ไฟล์	<pre>let query = 'input[type="file"]' let img = document     .querySelector(query)  let formData = new FormData() formData.append(     'avatar',     img.files[0] )  let config = {     method: 'POST',     body: formData } fetch(url, config)     .then(...)</pre>

Regular Expression (1)		
Character	ตรงทุกตัว (นอกจากเว้นบรรทัด)	<code>.</code>
Character	Escape character	<code>\.</code> <code>\\</code>
Character	เฉพาะจำนวน [0-9]	<code>\d</code>
Character	ที่ไม่ใช่จำนวน	<code>\D</code>
Character	เฉพาะข้อความอังกฤษและตัวเลข	<code>\w</code>
Character	ที่ไม่ใช่ข้อความอังกฤษและตัวเลข	<code>\W</code>

Regular Expression (2)		
Character	เฉพาะการเว้นวรรค	\s
Character	ที่ไม่ใช่การเว้นวรรค	\S
Assertion	เริ่มต้น	^
Assertion	จบ	\$
Assertion	ตรงกับคำต่อไป	?=
Assertion	ต้องไม่ตรงกับคำต่อไป	?!
Assertion	ตรงกับคำก่อนหน้า	?<=
Assertion	ต้องไม่ตรงกับคำก่อนหน้า	?<!
Group	คำใดๆก็ได้	[ ]
Group	กลุ่มคำที่ต้องตรงทั้งหมด	( )
Group	หรือ	
Group	จัดกลุ่มชื่อ <n> ตาม x (Regex)	(?<n>x)
Group	จัดกลุ่มแบบไร้ชื่อ	(?:x)
Quantifier	0 หรือมากกว่านั้น	*
Quantifier	1 หรือมากกว่านั้น	+
Quantifier	0 หรือ 1	?
Quantifier	ต้องมีจำนวนตาม n	{n}
Quantifier	ต้องมีจำนวนอย่างน้อย n	{n,}
Quantifier	ต้องมีจำนวนระหว่าง n ถึง m	{n,m}
Flag	ทุกตัวเท่าที่มี (Global)	/.../g
Flag	ไม่สนใจตัวพิมพ์เล็กใหญ่ (อังกฤษ)	/.../i
Flag	ตรวจที่ละบรรทัด	/.../m

ตัวอย่าง Regex	
ตรงกับ <b>Lion</b> (1 ชุด)	.n
ตรงกับ <b>Lion</b> (1 ชุด)	...n
ตรงกับ <b>ABC12345</b> (5 ชุด)	\d
ตรงกับ <b>ABC12345</b> (3 ชุด)	\D
ตรงกับ <b>ABC12345</b> (1 ชุด)	\D+
ตรงกับ <b>I love to code</b> (3 ชุด)	\s
ตรงกับ <b>I love to code</b> (4 ชุด)	\S+
ตรงกับ <b>ANT</b> แต่ไม่ตรงกับ <b>BAT</b>	^A
ตรงกับ <b>ABC</b> แต่ไม่ตรงกับ <b>CAT</b>	C\$
ตรงกับ <b>Oh! What??</b> (3 ชุด)	[!?]
ตรงกับ <b>dog cat</b> แต่ไม่กับ <b>rat</b>	(dog cat)
ตรงกับ <b>JohnDoe</b> แต่ไม่ตรงกับ <b>JaneDoe</b>	John(?:=Doe)
ตรงกับ <b>3.141</b> และ <b>12.5</b>	(?<=\. )\d+
ตรงกับ <b>&lt;h1&gt;My title&lt;/h1&gt;</b> และ <b>&lt;p&gt;Content&lt;/p&gt;</b>	(?<=<.+>).+(?<=\/.+>)
ตรงกับ <b>Name: A;\nName: B;</b> โดยเก็บเป็นชื่อกลุ่ม	/Name: (?<n>\w+);/gm
ตรงกับไฟล์นามสกุล <b>abc.js</b> หรือ <b>abc.ts</b> เท่านั้น	.*(?:j t)s
ตรงกับไฟล์นามสกุล <b>abc.jpg</b> หรือ <b>abc.jpeg</b> เท่านั้น	.*(?:jpe?g)
ตรงกับ <b>#fcb03</b> แต่ไม่ตรงกับ <b>#09zpus</b>	#[0-f]{6}
ข้อมูลต้องมีตัวเลข ตัวอักษร a-z และ A-Z เป็นจำนวน 6-20 ตัวอักษร (รหัสผ่านที่แข็งแรง)	^(?=.*\d)(?=.*[a-z])(?=.*[A-Z])[\da-zA-Z]{6,20}\$
ตรงกับภาษาไทยเท่านั้น	[\u0E00-\u0E7F]+

jQuery (1)	
<b>Selector</b>	<code>\$('#h1.title')</code> // ค้นหาและเลือก
<b>Element</b>	<code>\$('&lt;h1 class="title"&gt;')</code> // สร้างใหม่ <code>.text([value])</code> // innerText <code>.html([value])</code> // innerHTML <code>.first()</code> // ตัวแรกที่พบ <code>.last()</code> // ตัวสุดท้ายที่พบ <code>.is([html])</code> // ตรวจสอบ <code>.not([html])</code> // ตรวจสอบ <code>.append([html])</code> // เพิ่มหลัง <code>.prepend([html])</code> // เพิ่มก่อน <code>.children()</code> // Child ที่มี <code>.siblings()</code> // Sibling ทั้งหมด <code>.parent()</code> // Parent ตัวเดียว <code>.parents()</code> // Parent ทั้งหมด <code>.previous()</code> // Element ก่อนหน้า <code>.next()</code> // Element ถัดไป <code>.empty()</code> // ลบ Child ทั้งหมด <code>.remove()</code> // ลบทั้ง Element <code>.clone()</code> // สำเนา <code>.on()</code> // Event Listener
<b>Attribute</b>	<code>.attr(key[, value])</code> // อ่านหรือแก้ไข <code>.removeAttr(key)</code> // ลบ <code>.val()</code> // อ่าน value=""
<b>CSS</b>	<code>.css(name[, value])</code> // อ่านหรือแก้ไข <code>.hasClass(name)</code> // ตรวจสอบ Class <code>.addClass(name)</code> // เพิ่ม Class <code>.toggleClass(name)</code> // สลับ Class <code>.removeClass(name)</code> // ลบ Class
<b>Dimension</b>	<code>.width()</code> // ความกว้าง <code>.innerWidth()</code> // กว้างภายใน <code>.outerWidth()</code> // กว้างภายนอก <code>.height()</code> // ความสูง <code>.innerHeight()</code> // สูงภายใน <code>.outerHeight()</code> // สูงภายนอก
<b>Offset</b>	<code>.offset('top'    'left')</code> // ห่าง <code>.offsetParent()</code> // ห่างกับ Parent <code>.position()</code> // ตำแหน่ง <code>.scrollLeft()</code> // ระยะ Scroll <code>.scrollTop()</code> // ระยะ Scroll
<b>Dataset</b>	<code>.data(name[, value])</code> // อ่านหรือแก้ไข <code>.data(obj)</code> // ใช้แบบ Object <code>.removeData(name)</code> // ลบ

jQuery (2)	
<b>Effect</b>	<code>.hide()</code> // ซ่อน <code>.show()</code> // แสดง <code>.toggle()</code> // สลับการซ่อน <code>.fadeIn()</code> // เฟดเข้า <code>.fadeOut()</code> // เฟดออก <code>.fadeToggle()</code> // สลับเฟด <code>.slideDown()</code> // สไลด์ลง <code>.slideUp()</code> // สไลด์ขึ้น <code>.slideToggle()</code> // สลับสไลด์

ECharts	
<b>ตัวอย่าง โครงสร้าง</b>	<pre>let el = \$('#chart') let chart = echarts.init(el) let opts = {   title: { text: 'string' },   legend: { data: [{name, icon}] },   xAxis: { name, type, data },   yAxis: { name, type, data },   series: [{ name, type, data }],   dataset: { source: [] } } chart.setOption(opts)</pre>
<b>Series</b>	'bar'   'line'   'pie'   'scatter'
<b>Dataset</b>	<pre>let opts = {   dataset: {     source: [       ['name', 'a', 'b', 'c'],       ['2015', 15, 10, 20],       ['2016', 10, 15, 25],       ['2017', 10, 20, 15]     ]   },   xAxis: { type: 'category' },   series: [     { type: 'bar' },     { type: 'bar' },     { type: 'bar' }   ] }</pre>
<b>Resize (reactive)</b>	<pre>window.onresize = function () {   chart.resize() }</pre>



Lodash (1)	
<b>Collection</b>	_.find() // ค้นหาได้ทั้ง Arr, Obj _.each() // วนลูปได้ทั้ง Arr, Obj _.map() // แมพได้ทั้ง Arr, Obj _.filter() // กรองได้ทั้ง Arr, Obj _.every() // ตรวจสอบทั้งหมด _.some() // ตรวจสอบบางส่วน _.orderBy() // จัดเรียงตามเงื่อนไข _.groupBy() // จัดกลุ่มตามเงื่อนไข _.keyBy() // จัดคีย์ตามเงื่อนไข _.countBy() // แยกนับตามเงื่อนไข _.size() // นับขนาดข้อมูล _.sample() // สุ่มมา 1 ชิ้น _.sampleSize() // สุ่มมา n ชิ้น
<b>Array</b>	_.difference() // หาจุดต่าง _.union() // ยูเนียน _.intersection() // อินเตอร์เซกชัน _.xor() // ความแตกต่างสมมาตร _.uniq() // ลบข้อมูลซ้ำ _.pull() // ลบข้อมูลตามเงื่อนไข _.compact() // ลบข้อมูลที่เกี่ยวข้องของเท็จ _.last() // ตัวสุดท้าย _.chunk() // แบ่งสัดส่วน _.zip() // จัดกลุ่ม _.unzip() // คลายกลุ่ม
<b>Object</b>	_.has() // มีคีย์ _.get() // เรียกคีย์ _.set() // ตั้งข้อมูลในคีย์ _.pick() // เลือกเฉพาะคีย์ _.omit() // นำเฉพาะคีย์ออก _.invert() // สลับระหว่าง Key, Val _.mapKeys() // แมพ Key _.mapValues() // แมพ Val _.merge() // รวมข้อมูลตามมรดก
<b>Lang</b>	_.cloneDeep() // คัดลอก Arr, Obj _.isEmpty() // ตรวจสอบ Arr, Obj ได้ _.isEqual() // ตรวจสอบ Arr, Obj ได้ _.isMatch() // ตรวจสอบ Obj บางส่วน
<b>String</b>	_.escape() // เข้ารหัส Entity _.unescape() // ถอดรหัส Entity _.truncate() // จำกัดข้อความ _.template() // Interpolate
<b>Number</b>	_.random() // สุ่มกำหนดขนาดได้

Lodash (2)	
<b>Function</b>	_.once() // รัน Fn ครั้งเดียว _.before() // รัน Fn ก่อนจาก n _.after() // รัน Fn หลังจาก n _.debounce() // ทะยอยรัน Fn _.partial() // เต็ม Args ล่วงหน้า

Numeral.js	
<b>สร้าง</b>	numeral(190) // 190 numeral('23rd') // 23 numeral('\$9,900') // 9900 numeral('15KB') // 15000
<b>แสดงผล</b>	let n = numeral(1695125.7985) n.format('0,0') // 1,695,126 n.format('\$0,0.00') // \$1,695,125.80 n.format('.00') // .80 n.format('0.0a') // 1.7m n.format('0.0b') // 1.7MB  let m = numeral(1650) m.format('000000') // 001650 m.format('0o') // 1650th m.format('0%') // 165000% m.format('00:00:00') // 0:27:30
<b>Default</b>	numeral.defaultFormat('0o')  numeral(1).format() // 1st numeral(2).format() // 2nd numeral(3).format() // 3rd

MProgress.js	
<b>Construct</b>	let mprogress = new Mprogress()
<b>เริ่ม</b>	mprogress.start()
<b>จบ</b>	mprogress.end()
<b>ตั้งเปอร์เซ็นต์</b>	mprogress.set(0.5) // 50%
<b>เร่ง</b>	mprogress.inc()



Luxon (1)	
<b>Object</b>	luxon.DateTime // วันและเวลา luxon.Duration // ระยะเวลา luxon.Info // ข้อมูล luxon.Settings // ตั้งค่า Global
<b>Time (obj)</b>	{ year, month, day, hour, minute, second, millisecond }
<b>Token (fmt)</b>	// ตัวอย่าง 2014-08-06T13:07:04.054 SSS // 054 ss // 04 mm // 07 hh // 01 (12 ชม.) HH // 13 (24 ชม.) a // AM x // 1407287224054 (ms) tt // 1:07:04 PM TT // 13:07:04 dd // 06 ccc // Wed cccc // Wednesday LL // 08 LLL // Aug LLLL // August y // 2014 yy // 14 D // 8/6/2014 DD // Aug 6, 2014 DDD // August 6, 2014 DDDD // Wednesday, August 6, 2014
<b>Format options (opts)</b>	DateTime.DATE_SHORT DateTime.DATE_MED DateTime.DATE_MED_WITH_WEEKDAY DateTime.DATE_FULL DateTime.DATE_HUGE DateTime.TIME_SIMPLE DateTime.TIME_WITH_SECONDS DateTime.TIME_24_SIMPLE DateTime.TIME_24_WITH_SECONDS DateTime.DATETIME_SHORT DateTime.DATETIME_SHORT_WITH_SECONDS DateTime.DATETIME_MED DateTime.DATETIME_MED_WITH_SECONDS DateTime.DATETIME_MED_WITH_WEEKDAY DateTime.DATETIME_FULL DateTime.DATETIME_FULL_WITH_SECONDS DateTime.DATETIME_HUGE DateTime.DATETIME_HUGE_WITH_SECONDS

Luxon (2)	
<b>สร้าง เวลา</b>	DateTime.now() DateTime.local(y, m, d, h, M, s, ms) DateTime.utc(y, m, d, h, M, s, ms) DateTime.fromJSDate(Date) DateTime.fromObject(obj) DateTime.fromMillis(ms) DateTime.fromSeconds(s) DateTime.fromISO(text) DateTime.fromFormat(text, fmt)
<b>เรียก ข้อมูล</b>	let d = DateTime.now() d.year // ปี d.month // เดือน d.day // วันที่ d.second // วินาที d.weekday // วัน (0 = อาทิตย์) d.offset // ช่วงเวลา Timezone d.zoneName // ชื่อ Timezone
<b>แสดง เวลา</b>	d.toLocaleString(opts) d.toLocaleString(DateTime.DATE_MED) d.toFormat(fmt) d.toFormat('dd-LL-y') d.toISO()
<b>ตั้งเวลา</b>	d.set(obj) d.set({ hour: 9, minute: 30 })
<b>คำนวณ</b>	d.plus(obj) d.plus({ day: 1, hour: 12 }) d.minus(obj) d.diff(otherDateTime) d.diffNow()
<b>ตั้ง ระบบ เวลา</b>	d.setZone('Asia/Bangkok').toFormat() d.setLocale('th-TH').toFormat('DD') d.reconfigure({ outputCalendar: 'buddhist' }).toFormat('DD')
<b>ตั้งแบบ เริ่มต้น</b>	Settings.defaultZone = 'Asia/Bangkok' Settings.defaultLocale = 'th-TH' Settings .defaultOutputCalendar = 'buddhist'
<b>ดูข้อมูล</b>	Info.months() // Arr ของเดือน Info.Weekdays() // Arr ของวัน

SweetAlert2	
Method	Swal.fire(string   options)
Shortcut	Swal.fire('title', 'text', 'error')
Promise	Swal.fire({ input: 'text' }) .then(result)
Options	<pre>{   title: 'string',   text: 'string',   html: 'string',   icon: 'success'   'warning'           'error'   'info'           'question'   position: 'top'   'top-start'               'top-end'   'center'               'center-*'   'bottom'               'bottom-*',   input: 'ใช้ตาม &lt;input type&gt;',   inputValue: 'string',   inputOptions: {},   toast: true,   footer: 'string',   timer: 1234, // ms   imageUrl: 'url',   imageAlt: 'string',   imageWidth: 1234, // px   imageHeight: 1234, // px   showCloseButton: true,   closeButtonHtml: 'string',   showCancelButton: true,   cancelButtonText: 'string',   showConfirmButton: true,   confirmButtonText: 'string',   reverseButtons: true }</pre>
Event	<pre>{   willOpen: () =&gt; {},   didOpen: () =&gt; {},   didRender: () =&gt; {},   willClose: () =&gt; {},   didClose: () =&gt; {},   didDestroy: () =&gt; {},   preConfirm: () =&gt; {},   preCancel: () =&gt; {}, }</pre>

Vanilla Lazyload	
ตัวอย่าง HTML	<pre>&lt;img class="lazy"   data-src="img.jpg"&gt;  &lt;div class="lazy"   data-bg="bg.jpg"&gt;&lt;/div&gt;</pre>
ตัวอย่าง CSS	<pre>img:not([src]):not([srcset]) {   visibility: hidden; }</pre>
Initialize	<pre>let lazy = new LazyLoad()</pre>
บังคับอัปเดต	<pre>lazy.update()</pre>

Quill	
โครงสร้าง	<pre>let q = new Quill('#editor', opt)</pre>
Options	<pre>let opt = {   theme: 'snow',   placeholder: 'string',   readOnly: true,   formats: [], // เริ่มต้นจะเอามาทุกชิ้น   modules: {} // ต้องดูเพิ่มหน้าเว็บ }</pre>
Formats	<pre>['bold', 'italic', 'underline', 'strike', 'link', 'size', 'font', 'color', 'background', 'script', 'code', 'header', 'align', 'list', 'indent', 'direction', 'blockquote', 'code-block', 'formular', 'image', 'video']</pre>
Methods	<pre>q.getText()      // ได้เฉพาะข้อความ q.getContents()  // ได้ JSON q.getLength()    // ความยาวเนื้อหา q.setText()      // กำหนดข้อความ q.on()           // Event listener</pre>
เรียก HTML	<pre>q.root.innerHTML</pre>
Events	<pre>q.on('text-change', () =&gt; {})  'text-change'      // ข้อความเปลี่ยน 'selection-change' // เลือกเนื้อหา 'editor-change'    // ทั้งสองข้างบน</pre>

DataTables (Vanilla)	
Construct	<pre>let table = document   .querySelector('table') let cf = {} let dt = new DataTable(table, cf)</pre>
Config	<pre>let cf = {   perPage: 10,   perPageSelect: [5, 10, 15],   searchable: false,   footer: true,   fixedHeight: true,   columns: [{}, {}, {}],   data: { headings, data },   ajax: {} }</pre>
Methods	<pre>dt.print()    // พิมพ์ dt.page(n)    // ไปที่หน้า n dt.export({   download: true,   type: 'csv' // 'txt', 'json',               // 'sql' }) dt.import({   data: 'string',   type: 'csv' // เหมือนกับ export }) dt.on(event)  // Event listener</pre>
Events	<pre>dt.on('datatable.page', () =&gt; {})  '*.page'      // เมื่อเลือกเพจ '*.perpage'   // เมื่อเลือกต่อเพจ '*.init'      // เมื่อเริ่มสำเร็จ '*.sort'      // เมื่อจัดเรียง '*.search'    // เมื่อค้นหา</pre>
Columns	<pre>let cf = {   columns: [{     sort: 'asc' // 'desc',     sortable: false,     hidden: true   }] }</pre>

DataTables (Vanilla) (2)	
Data	<pre>let data = {   headings: ['A', 'B', 'C'],   data: [     ['A1', 'B1', 'C1'],     ['A2', 'B2', 'C2'],     ['A3', 'B3', 'C3']   ] } let cf = { data }</pre>
Ajax (ใช้ได้ สองแบบ)	<pre>let ajax1 = 'path/to/file.json' let ajax2 = {   url: 'path/to/file.csv',   content: { type: 'csv' },   load: (xhr) =&gt; { ... } } let cf = { ajax: ajax1 } // ajax2</pre>

Masonry	
ตัวอย่าง HTML	<pre>&lt;div class="grid"&gt;   &lt;div class="grid-item"&gt;&lt;/div&gt;   &lt;div class="grid-item"&gt;&lt;/div&gt;   ... &lt;/div&gt;</pre>
Construct	<pre>let grid = \$('.grid') let m = new Masonry(grid, {   itemSelector: '.grid-item',   columnWidth: 200 // px })</pre>

Parallax.js	
ตัวอย่าง HTML	<pre>&lt;div id="scene"&gt;   &lt;div data-depth="0.2"&gt;1&lt;/div&gt;   &lt;div data-depth="0.6"&gt;2&lt;/div&gt; &lt;/div&gt;</pre>
เริ่มใช้งาน	<pre>let scene = \$('#scene') new Parallax(scene)</pre>

## Swiper

```
<div class="swiper">
  <div class="swiper-wrapper">
    <div class="swiper-slide">Slide 1</div>
    <div class="swiper-slide">Slide 2</div>
    <div class="swiper-slide">Slide 3</div>
    ...
  </div>
  <div class="swiper-pagination"></div>
  <div class="swiper-button-prev"></div>
  <div class="swiper-button-next"></div>
  <div class="swiper-scrollbar"></div>
</div>
```

**Modules** Swiper.use([Navigation])

**Construct** `let el = '.swipe'`  
`let s = new Swiper(el, opts)`

**Options** `let opts = {`  
   `direction: 'vertical',`  
     `// 'horizontal'`  
   `pagination: {`  
     `el: '.swiper-pagination',`  
     `type: 'progressbar'`  
   `},`  
   `navigation: {`  
     `nextEl: '.swiper-button-next',`  
     `prevEl: '.swiper-button-prev'`  
   `},`  
   `scrollbar: {`  
     `el: '.swiper-scrollbar'`  
   `},`  
   `autoplay: {`  
     `delay: 5000 // ms`  
   `},`  
   `speed: 600, // ms`  
   `spaceBetween: 30, // px`  
   `slidesPerView: 3, // หรือ 'auto'`  
   `centeredSlides: true,`  
   `loop: true,`  
   `grabCursor: true,`  
   `keyboard: { enabled: true },`  
   `mousewheel: true,`  
   `parallax: true,`  
   `lazy: true,`  
   `zoom: true`  
`}`

## Tippy.js

**เริ่มใช้งาน** `// <div data-tippy-content`  
`// ="Tooltip">Text</div>`  
  
`let el = '[data-tippy-content]'`  
`tippy('el', opt)`

**Options** `let opt = {`  
   `content: 'string', // ไม่จำเป็น`  
   `placement: 'top',`  
   `// 'right', 'bottom', 'left'`  
   `animation: 'fade',`  
   `// 'shift-away', 'scale'`  
   `// 'perspective'`  
   `trigger: 'click'`  
   `// 'mouseenter', 'focus'`  
   `duration: 500,`  
   `arrow: true,`  
   `followCursor: true,`  
   `theme: 'light'`  
`}`



Illustration by storyset.com

Node CLI	
REPL	node
เช็คเวอร์ชัน	node --version
รันไฟล์	node file.js # CJS node file.mjs # ESM
Debug	node --inspect file.js

CommonJS (CJS)	
Export	<pre>module.exports = 'abc' module.exports = {} module.exports = () =&gt; {}  module.exports = {   A: function () {},   B: () =&gt; {} }</pre>
Import	<pre>const fs = require('fs') const { mkdir } = require('fs')  const all = require('./file.js') const { A } = require('./file.js') const { A: Z } = require('./file.js')</pre>

ECMAScript Module (ESM)	
Export	<pre>export default 'abc' export default {} export default () =&gt; {}  export function A () {} export const B = () =&gt; {}</pre>
Import	<pre>import fs from 'fs' import { mkdir } from 'fs'  // เรียกได้ก็ต่อเมื่อมี export default import all from './file.mjs'  // เรียกได้ตลอด แต่จะนำทุกตัวมาใช้ import * as all from './file.mjs'  import { A } from './file.mjs' import { A as Z } from './file.mjs'</pre>

NPM CLI	
เริ่ม package.json	npm init -y
ค้นหา Package	npm search lodash
ดูรายละเอียด Package	npm view lodash
ดูเวอร์ชันที่มี	npm view lodash versions
ติดตั้ง package.json	npm i
ติดตั้ง Package ใหม่	npm i lodash
ติดตั้งเลือกเวอร์ชัน	npm i lodash@3.0
ติดตั้งแบบ Dev	npm i -D lodash
ติดตั้งแบบ Global	npm i -g nodemon
ลบ Package	npm r lodash
ลบแบบ Global	npm r -g nodemon
ดูรายการที่ติดตั้ง	npm list
ดูรายการบน Global	npm list -g
อัปเดต package.json	npm update
อัปเดตแบบ Global	npm update -g
รัน Scripts	npm run start npm run dev
สร้าง Package ภายใน	npm link
เชื่อม Package ภายใน	npm link <name>
ตรวจสอบคุณภาพ Package	npm audit
แก้ตามที่แนะนำ	npm audit fix --force
ตรวจปัญหา	npm doctor
เข้าระบบ NPM	npm login
นำ Package ขึ้น NPM	npm publish
นำออกจาก NPM	npm unpublish --force

Global Object	
<b>Path หลัก</b>	<code>__dirname</code> // Directory <code>__filename</code> // File
<b>Process</b>	<code>process.env</code> // Env ในเครื่อง <code>process.argv</code> // CLI Arguments <code>process.arch</code> // 'x64' <code>process.platform</code> // 'win32' <code>process.uptime</code> // เวลาที่ได้รับ <code>process.version</code> // เช็ควเวอร์ชัน <code>process.exit()</code> // จบทำงาน
<b>Event Listener</b>	<code>process.on('exit', () =&gt; {})</code>  <code>'exit'</code> // ออกโปรแกรม <code>'SIGINT'</code> // Ctrl + C // เมื่อ <code>Promise.resolve()</code> เรียกหลายครั้ง จะเป็นปัญหาได้ค่าเกินจำนวนครั้ง <code>'multipleResolves'</code> // เมื่อ <code>Promise.reject()</code> <code>'rejectionHandled'</code> // เมื่อ <code>Promise</code> ไม่มีการจัดการ <code>Error</code> <code>'uncaughtException'</code>

Stream	
<b>Reader (Event)</b>	<code>let r = fs.createReadStream('...')</code> <code>r.on('data', (chunk) =&gt; {})</code>  <code>'data'</code> // มีข้อมูลถูกอ่าน (แบ่ง chunk) <code>'close'</code> // จบการอ่าน <code>'pause'</code> // ถูกหยุดชั่วคราว <code>'resume'</code> // ถูกดำเนินการต่อ <code>'error'</code> // มีปัญหาเกิดขึ้น
<b>Reader (Method)</b>	<code>r.pause()</code> // หยุดชั่วคราว <code>r.resume()</code> // ดำเนินการต่อ <code>r.pipe()</code> // ต่อ Pipe กับ Writer
<b>Writer (Event)</b>	<code>let w = fs.createWriteStream('...')</code> <code>w.on('close', () =&gt; {})</code>  <code>'finish'</code> // เมื่อเรียก <code>.end()</code> <code>'close'</code> // เมื่ออ่านครบหลังเรียก <code>.end()</code> <code>'pipe'</code> // เมื่อ Reader เรียก <code>.pipe()</code> <code>'error'</code> // มีปัญหาเกิดขึ้น
<b>Writer (Method)</b>	<code>w.write()</code> // เขียนข้อมูล <code>w.end()</code> // สิ้นสุดการเขียน <code>w.destroy()</code> // ทำลาย

Utilities	
<b>Require</b>	<code>const util = require('util')</code>
<b>แปลงฟังก์ชัน</b>	<code>util.callbackify(original)</code> <code>util.promisify(original)</code>  <code>const fn = (cb) =&gt; {   if (ok) cb(null, 'ok')   else cb(new Error('Nah!')) }</code> <code>const fnAsync = util.promisify(fn)</code>  <code>fn((err, msg) =&gt; {})</code> <code>fnAsync().then(msg).catch(err)</code>

Path	
<b>Require</b>	<code>const path = require('path')</code>
<b>ไฟล์</b>	<code>let p = '/a/b/c/app.exe'</code> <code>path.basename(p)</code> // 'app.exe' <code>path.extname(p)</code> // '.exe'
<b>รวม</b>	<code>let a = '/a'</code> <code>let f = 'app.exe'</code> <code>path.join(a, f)</code> // '/a/f.exe' <code>path.resolve(a, f)</code> // 'C:\a\f.exe'
<b>Absolute</b>	<code>path.isAbsolute('./a/b')</code> // false <code>path.isAbsolute('/a/b')</code> // true
<b>Normalize</b>	<code>path.normalize('/a/b/..')</code> // '/a'

File System	
Require	<code>const fs = require('fs')</code> <code>// require('fs').promises</code>
Constants	<code>fs.constants.F_OK</code> // มีอยู่ <code>fs.constants.R_OK</code> // อ่านได้ <code>fs.constants.W_OK</code> // เขียนได้ <code>fs.constants.X_OK</code> // รันได้
เช็คไฟล์	<code>fs.access('/path', F_OK)</code> <code>fs.access('/path', W_OK)</code>
ตรวจสอบสถานะ	<code>fs.stat('/path')</code>
อ่านโฟลเดอร์	<code>fs.readdir('/path/dir')</code>
สร้างโฟลเดอร์	<code>fs.mkdir('/path/dir')</code>
อ่านไฟล์	<code>fs.readFile('/path/file')</code>
สร้างไฟล์	<code>let t = 'my text'</code> <code>fs.writeFile('/path/file', t)</code>
คัดลอกไฟล์	<code>fs.copyFile('/from', '/to')</code>
เปลี่ยนชื่อไฟล์	<code>fs.rename('/from', '/to')</code>
ลบไฟล์หรือโฟลเดอร์	<code>fs.rm('/any-path')</code> // ลบปกติ  <code>fs.rm('/any-path', {</code> <code>recursive: true, // ลบทั้งหมด</code> <code>force: true // บังคับลบ</code> <code>})</code>
ติดตาม	<code>fs.watch('/path')</code>
อ่านเขียนแบบ Stream	<code>fs.createReadStream('/path')</code> <code>fs.createWriteStream('/path')</code>

HTTP	
Require	<code>const http = require('http')</code> <code>const https = require('https')</code>
Request (Method)	<code>let u = 'http://web.com' // url</code> <code>let o = { // options</code> <code>headers: {},</code> <code>method: 'POST', // 'GET'</code> <code>timeout: 5000 // 5 วินาที</code> <code>}</code> <code>let c = (res) =&gt; {} // callback</code> <code>// res เป็น Stream</code>  <code>let req = http.request(u, o, c)</code> <code>// req เป็น Stream</code>
Request (Class)	<code>req.write(json) // ส่งข้อมูล JSON</code> <code>req.end() // เริ่มส่ง</code> <code>req.on('error', cb) // Listener</code>  <code>'error' // เมื่อ Error</code> <code>'connect' // เมื่อเชื่อมต่อสำเร็จ</code> <code>'response' // เมื่อมี Response</code>
Response (Class)	<code>res.pipe(writer) // Pipe</code> <code>res.on('close', cb) // อ่านเสร็จ</code> <code>res.headers // Headers ทุกชั้น</code> <code>res.statusCode // สถานะ HTTP</code> <code>res.statusMessage // สถานะ HTTP</code> <code>res.req // Original ของ req</code>



Illustration by storyset.com



ESLint	
ติดตั้ง	npm i -g eslint
เริ่มใช้	eslint --init
เรียกใช้แบบไม่ติดตั้ง	npx eslint --init
แก้ Config	<pre>{   "rules": {     "camelcase": "error",     "no-console": "warn",     "no-eval": "off"   } }</pre>

Dotenv	
ติดตั้ง	npm i dotenv
CJS	<code>require('dotenv').config()</code>
ESM	<code>import dotenv from 'dotenv'</code> <code>dotenv.config()</code>
.env	NODE_ENV=production PORT=3000 ADMIN_PASSWORD=Abc123

Commander	
ติดตั้ง	npm i commander
Construct	<pre>const { Command } =   require('commander')  let program = new Command()</pre>
Methods	<pre>program   .version('1.0.0')   .option('-d, --debug', 'descri')   .option('-i, --input', 'descri')   .parse(process.argv)  let opts = program.opts()</pre>

Enquirer	
ติดตั้ง	npm i enquirer
Prompt	<pre>const { prompt } =   require('enquirer')  let questions = [   {     type: 'input',     name: 'first_name',     message: 'First name:'   },   {     type: 'input',     name: 'last_name',     message: 'Last name:'   } ]  async function start () {   let d = await prompt(questions) }</pre>
Built-in Types	<pre>{ type: 'input' } { type: 'numeral' } { type: 'invisible' } { type: 'password' } { type: 'confirm' } { type: 'toggle' } {   type: 'autocomplete',   // 'select'   // 'multiselect'   // 'sort'   choices: ['a', 'b', 'c'] }</pre>

CLI Table 3	
ติดตั้ง	npm i cli-table3
Construct	<pre>let Table = require('cli-table3') let opt = {   head: ['A', 'B', 'C'] } let table = new Table(opt) table.push(['A1', 'B1', 'C1']) table.push(['A2', 'B2', 'C2']) table.toString() // ผลลัพธ์</pre>



SheetJS	
ติดตั้ง	npm install xlsx
อ่านไฟล์	<pre>const xlsx = require('xlsx') let wb = xlsx.readFile('file.xls') let ws = wb.Sheets.NameOfSheet</pre>
Util	<pre>xlsx.utils.sheet_to_json(ws) xlsx.utils.aoa_to_sheet(arr) xlsx.utils.json_to_sheet(json) xlsx.utils.table_to_sheet(element)  let wb = xlsx.utils.book_new() xlsx.utils.book_append_sheet(   wb, ws, 'NameOfSheet' )</pre>
เขียนไฟล์	<pre>xlsx.writeFile('sheet.xls', wb)</pre>

Axios	
ติดตั้ง	npm install axios
API	<pre>axios(conf) axios.get(url, conf) axios.post(url, data, conf) axios.put(url, data, conf) axios.patch(url, data, conf) axios.delete(url, data, conf)</pre>
GET	<pre>let res = await axios.get('url')</pre>
POST	<pre>let data = { a: 1, b: 2 } await axios.post('url', data)</pre>
Config	<pre>let conf = { url, method, headers, params, data, timeout, responseType }</pre>
Response (res)	<pre>{ data, status, statusText, headers, config, request }</pre>
Stream	<pre>let url = '/somefile' let responseType = 'stream'  axios({ url, responseType })   .then(res =&gt; res.data.pipe(...))</pre>

PDFMake	
ติดตั้ง	npm i pdfmake
Construct	<pre>let PDF = require('pdfmake') let printer = new PDF(fonts)</pre>
เพิ่มฟอนต์	<pre>let fonts = {   Sarabun: {     normal: 'Sarabun-Regular.ttf',     bold: 'Sarabun-Bold.ttf'   } }</pre>
สร้าง PDF	<pre>let doc = {   content: [     'บรรทัดแรก',     { text: 'Bold', bold: true },     { image: './logo.png' }   ],   defaultStyle: {     font: 'Sarabun',     fontSize: 10   } }  let writer = fs   .createWriteStream('output.pdf') let pdf = printer   .createPdfKitDocument(doc) pdf.pipe(writer) pdf.end()</pre>
Document	<pre>{ content, defaultStyle, style, background, header, footer, images, pageMargins, pageOrientation, pageSize, compress, userPassword, ownerPassword, watermark }</pre>
Content และบางส่วนของ Style	<pre>{ text, style, font, fontSize, lineHeight, bold, italics, sub, sup, alignment, color, background, decoration, decorationStyle, decorationColor, margin, opacity, leadingIndent, noWrap, pageBreak, columns, stack, link, id, linkToPage, linkToDestination, ul, ol, type start, markerColor, canvas, svg, image, width, height, fit, cover, table, layout, qr }</pre>

Validator.js (1)	
ติดตั้ง	<code>npm i validator</code>
a-zA-Z	<code>isAlpha('Abc') // true</code> <code>isAlpha('Abc123') // false</code>
a-zA-Z0-9	<code>isAlphanumeric('Abc') // true</code> <code>isAlphanumeric('Abc123') // true</code>
ASCII	<code>isAscii('abc123_!') // true</code> <code>isAscii('ภาษาไทย') // false</code>
ก่อน/หลังวัน	<code>let d1 = '2030-01-30'</code> <code>let d2 = '2022-01-30'</code> <code>isAfter(d1, d2) // true</code> <code>isBefore(d1, d2) // false</code>
วัน	<code>isDate('2022-01-30') // true</code> <code>isDate(new Date()) // true</code>
อีเมล	<code>let a = 'abc@gmail.com'</code> <code>let b = 'ABC@gmail.com'</code> <code>let c = 'abc+1@gmail.com'</code>  <code>isEmail(a) // true</code> <code>isEmail(b) // true</code> <code>isEmail(c) // true</code>  <code>normalizeEmail(a) // abc@gmail.com</code> <code>normalizeEmail(b) // abc@gmail.com</code> <code>normalizeEmail(c) // abc@gmail.com</code>
URL	<code>isURL('http://abc.com') // true</code> <code>isURL('abc.com') // true</code>  <code>let c = { require_protocol: true }</code> <code>isURL('abc.com', c) // false</code>
เบอร์โทร	<code>isMobilePhone('191') // false</code> หมด <code>isMobilePhone('02-777-7777')</code>  <code>isMobilePhone('027777777') // true</code>
โค้ดสี	<code>isHexColor('#97b5f0') // true</code> <code>isHexColor('97b5f0') // true</code>
JSON	<code>isJSON('{ "a": "1" }') // true</code> <code>isJSON('{a: 1}') // false</code> เพราะ "
MIME	<code>isMimeType('image/png') // true</code> <code>isMimeType('.png') // false</code>

Validator.js (2)	
Hash	<code>isHash(data, 'md5')</code> <code>isHash(data, 'sha1')</code>
JWT	<code>isJWT('eyJ...')</code>
UUID	<code>isUUID('ae135996-79.....')</code>
Lat,Long	<code>isLatLong('0.255,1.512') // true</code>
รหัสผ่าน แข็งแรง	<code>isStrongPassword(password, {</code> <code>minLength: 8,</code> <code>minLowercase: 1,</code> <code>minUppercase: 1,</code> <code>minNumbers: 1,</code> <code>minSymbols: 1,</code> <code>returnScore: false,</code> <code>pointsPerUnique: 1,</code> <code>pointsPerRepeat: 0.5,</code> <code>pointsForContainingLower: 10,</code> <code>pointsForContainingUpper: 10,</code> <code>pointsForContainingNumber: 10,</code> <code>pointsForContainingSymbol: 10</code> <code>})</code>

UUID	
ติดตั้ง	<code>npm i uuid</code>
สร้างไอดี	<code>const uuid = require('uuid')</code> <code>uuid.v1() // timestamp</code> <code>uuid.v3() // namespace (md5)</code> <code>uuid.v4() // random</code> <code>uuid.v5() // namespace (sha-1)</code>
ตรวจสอบ	<code>uuid.validate(str)</code>



Illustration by storyset.com

Glob	
ติดตั้ง	<code>npm i glob</code>
ใช้งาน	<pre>const glob = require('glob') glob('*.js', callback) glob.sync('*.js')</pre>
ทุกอักขระ ยกเว้น /	<code>*</code>
ทุกอักขระ รวมถึง /	<code>**</code>
ทุกอักขระ รวมถึง / หนึ่งครั้ง	<code>?</code>
ตรงกับข้อมูล a, b, c	<code>[abc]</code>
ตรงกับชุดข้อมูล abc, xyz	<code>{abc,xyz}</code>
ตรงกับ 1 ถึง 9	<code>{1..9}</code>
ต้องไม่ตรงกับ abc	<code>!abc</code>
ตรงกับ ab แต่ไม่ตรงกับ abc	<code>a!(bc)</code>
ทุกไฟล์ แต่ไม่รวมโฟลเดอร์และไฟล์ซ่อน	<code>*</code>
ทุกไฟล์และโฟลเดอร์ แต่ไม่รวมไฟล์ซ่อน	<code>**</code>
เฉพาะไฟล์ซ่อน	<code>.*</code>
ทุกไฟล์ รวมถึงไฟล์ที่ซ่อน	<code>{,.[^.]...?}*</code>
ชื่ออะไรก็ได้ ต้องลงท้ายด้วย .js เท่านั้น	<code>*.js</code>
ชื่ออะไรก็ได้ ต้องเป็นนามสกุล js หรือ ts	<code>*(js ts)</code>
ลงท้ายด้วย js หรือ mjs เท่านั้น	<code>*.+(m js)</code>
ทุกนามสกุลไฟล์ ยกเว้นไฟล์ที่ไม่ชื่อ	<code>*.*</code>
ทุกไฟล์ในโฟลเดอร์ /src	<code>/src/*</code>
ทุกไฟล์และโฟลเดอร์หลังจากใน /src	<code>/src/**</code>



Illustration by storyset.com

Sharp.js	
ติดตั้ง	<code>npm i sharp</code>
Construct	<pre>let s = sharp('input.png') let s = sharp(buffer) let s = sharp(base64)</pre>
Metadata	<code>await s.metadata()</code>
ปรับขนาด	<pre>s.resize(150) // width s.resize(100, 300) // width,height</pre>
เบลภาพ	<code>s.blur(10) // px</code>
แปลงไฟล์	<pre>s.jpeg({ quality: 80 }) // 1-100 s.png()</pre>
แก้ปัญหาพื้นหลังดำ	<code>s.flatten({ background: '#ffffff' })</code>
Output	<pre>// ทุกตัว return Promise s.toFile('out.png') s.toBuffer((err,data,info) =&gt; {})</pre>

FFMPEG	
ติดตั้ง	<pre>npm i fluent-ffmpeg # ต้องติดตั้ง binary เพิ่มเติม ค้นไ้บน # Google -&gt; ffmpeg binary</pre>
Construct	<pre>const ffmpeg = require('fluent-ffmpeg')</pre>
แปลงวิดีโอ	<pre>ffmpeg('/path/to/video.avi')   .audioBitrate(128)   .videoBitrate(1000)   .videoCodec('libx264') // mp4   .format('mp4')   .fps(29.7)   .size('?x720') // 720p   .on('start', () =&gt; {...})   // 'error', 'end'   .save('/path/to/output.mp4')</pre>
Stream	<pre>let out = fs   .createWriteStream('out.mp4')  ffmpeg('path/to/video.mp4')   .size('?x1080') // 1080p   .pipe(out)</pre>

Archiver	
ติดตั้ง	<code>npm i archiver</code>
Construct	<code>let arc = require('archiver')</code> <code>let zip = arc('zip')</code>
เพิ่มไฟล์	<code>let opt = { name: 'abc.txt' }</code> <code>let str = 'abc'</code> <code>let buf = Buffer.from('abc')</code> <code>let read = fs.</code> <code>createReadStream('file.txt')</code>  <code>zip.append(str, opt)</code> <code>zip.append(buf, opt)</code> <code>zip.append(read, opt)</code>
เพิ่มจากโฟลเดอร์	<code>zip.directory('/form')</code> <code>zip.directory('/from', '/new')</code> <code>zip.directory('/from', false)</code> // วาง /from ไว้ที่ Root Dir
Glob	<code>zip.glob('*.js')</code>
สร้าง Zip	<code>let writer = fs.</code> <code>createWriteStream('path.zip')</code>  <code>zip</code> <code>.pipe(writer)</code> <code>.finalize()</code>

Faker (1)	
ติดตั้ง	<code>npm i @faker-js/faker</code> # ระวังตัว <code>npm i faker</code> เป็นตัวมั่วร้าย
Require	<code>let faker =</code> <code>require('@faker-js/faker')</code>
faker .address	<code>.city()</code> // เมือง (จังหวัด) <code>.country()</code> // ประเทศ <code>.latitude()</code> // ละติจูด <code>.longitude()</code> // ลองจิจูด <code>.timeZone()</code> // เขตเวลา
faker .animal	<code>.dog()</code> // สุนัข <code>.cat()</code> // แมว <code>.type()</code> // ประเภทสัตว์
faker .company	<code>.companyName()</code> // ชื่อองค์กร

Faker (2)	
faker .commerce	<code>.product()</code> // ชื่อสั้น <code>.productName()</code> // ชื่อยาว <code>.productDescription()</code> // อธิบาย <code>.productAdjective()</code> // เพิ่มเติม <code>.productMaterial()</code> // วัสดุ <code>.color()</code> // สี <code>.department()</code> // หมวดหมู่ <code>.price()</code> // ราคา
faker .datatype	<code>.number()</code> // จำนวน <code>.float()</code> // ทศนิยม <code>.datetime()</code> // วันเวลา <code>.string()</code> // ข้อความ <code>.uuid()</code> // UUID <code>.boolean()</code> // จริงเท็จ <code>.hexadecimal()</code> // เลขฐานสิบหก <code>.json()</code> // JSON <code>.array()</code> // Array
faker .date	<code>.past()</code> // อดีต <code>.future()</code> // อนาคต <code>.between()</code> // ระหว่าง (min, max) <code>.recent()</code> // ที่ผ่านมาเร็วๆนี้ <code>.soon()</code> // ที่จะมาเร็วๆนี้ <code>.month()</code> // เดือน <code>.weekday()</code> // วัน
faker .finance	<code>.amount()</code> // จำนวนเงิน <code>.transactionType()</code> // วิธีจ่าย <code>.currencyCode()</code> // โค้ดหน่วยเงิน <code>.currencyName()</code> // ชื่อ <code>.currencySymbol()</code> // สัญลักษณ์ <code>.creditCardNumber()</code> // เลขเครดิต <code>.creditCardCVV()</code> // CVV ท้ายบัตร
faker .image	<code>.image()</code> // ลิงก์ภาพทั่วไป <code>.abstract()</code> // ลิงก์รูปนามธรรม <code>.animals()</code> // ลิงก์รูปสัตว์ <code>.business()</code> // ลิงก์รูปธุรกิจ <code>.cats()</code> // ลิงก์รูปแมว <code>.city()</code> // ลิงก์รูปเมือง <code>.food()</code> // ลิงก์รูปอาหาร <code>.nightlife()</code> // ลิงก์รูปกลางคืน <code>.fashion()</code> // ลิงก์รูปแฟชั่น <code>.people()</code> // ลิงก์รูปผู้คน <code>.nature()</code> // ลิงก์รูปธรรมชาติ <code>.sports()</code> // ลิงก์รูปกีฬา <code>.transport()</code> // ลิงก์ยานพาหนะ <code>.dataUri()</code> // ภาพแบบ URI

Faker (3)	
<b>faker</b> <b>.internet</b>	.avatar()      // รูปประจำตัว .email()        // อีเมล .userName()    // ชื่อผู้ใช้งาน .url()           // URL .ip()            // IPv4 .ipv6()         // IPv6 .userAgent()   // UserAgent .color()        // สี .mac()          // MAC Address .password()    // รหัสผ่าน
<b>faker</b> <b>.lorem</b>	.word()         // คำ .sentence()     // ประโยค .paragraph()    // ย่อหน้า .text()          // ข้อความ .lines()        // เป็นบรรทัด .slug()         // Slug
<b>faker</b> <b>.name</b>	.firstName()    // ชื่อจริง .lastName()     // นามสกุล .middleName()   // ชื่อกลาง .prefix()        // คำนำหน้า .suffix()        // คำลงท้าย .findName()     // ชื่อเต็ม .gender()        // เพศ .jobTitle()      // ชื่องาน .jobDescriptor() // คำอธิบาย .jobArea()       // หน่วยงาน .jobType()       // รูปแบบงาน .title()         // ชื่องานเต็ม
<b>faker.phone</b>	.phoneNumberFormat() // โทรศัพท์
<b>faker</b> <b>.system</b>	.directoryPath() // โฟลเดอร์ .filePath()       // ไฟล์ .commonFileName() // ชื่อไฟล์ .commonFileType() // ประเภทไฟล์ .commonFileExt() // นามสกุลไฟล์ .mimeType()      // MIME .semver()         // เลขเวอร์ชัน

Puppeteer	
<b>ติดตั้ง</b>	npm i puppeteer
<b>Require</b>	let pup = require('puppeteer')
<b>Launch</b>	let browser = await pup.launch()
<b>Page</b>	let page = await browser.newPage()
<b>Goto</b>	let url = 'http://example.com' let opt = { timeout: 30000, // 30s waitUntil: 'load', // โหลดเสร็จ // 'domcontentloaded' DOM เสร็จ // 'networkidle0' ไม่มี Network // 'networkidle2' ไม่มี >=2 Network } await page.goto(url, opt)
<b>ย้อนเว็บ</b>	await page.goBack()
<b>เว็บต่อไป</b>	await page.goForward()
<b>Reload</b>	await page.reload()
<b>แคปรูป</b>	let opt = { path: 'a.png', // หากไม่มีจะได้ Buf type: 'png',    // jpeg, webp fullPage: false // ถ่ายเต็มเว็บ } await page.screenshot(opt)
<b>HTML</b>	await page.content()
<b>Selector</b>	await page.\$('h1')        // ขึ้นเดียว await page.\$\$('.title') // หลายขึ้น // จะได้ ElementHandle มาใช้ต่อ
<b>รอ</b>	await page.waitFor('form.login') await page.waitFor(1000) // 1s
<b>Eval</b>	let script = () => { window.alert() // เรียกสคริปต์บนเว็บ } await page.evaluate(script)
<b>คลิก</b>	await page.click('form > button')
<b>ชี้</b>	await page.hover('.tooltip')
<b>พิมพ์</b>	let str = 'mypassword1234' await page.type('input.pass', str)

PM2	
ติดตั้ง	npm i -g pm2
สร้าง Config	pm2 init
สำหรับ Linux	pm2 startup
เริ่มรัน	pm2 start # ต้องมี pm2 init pm2 start dev.js --watch pm2 start p.js --name prod pm2 start max.js -i max
รันแบบ Cronjob ทุกๆ เที่ยงคืน และห้าม รันใหม่ ทันที	pm2 start file.js --cron "0 0 * * *" --no-autorestart
เริ่มใหม่	pm2 restart app_name pm2 restart all # เริ่มทั้งหมด
ปิดการทำงาน	pm2 stop app_name pm2 stop all # หยุดทั้งหมด
ลบ	pm2 delete app_name pm2 delete all # ลบทั้งหมด
ดูรายการ รัน	pm2 list pm2 monit # ละเอียดกว่า
ดูสถานะ โดย ละเอียด	pm2 show app_name
ดู Log จากการ รัน	pm2 logs pm2 logs app_name
บันทึก เพื่อ Startup	pm2 save