# Javascript The Complete Referrence

Z



รวบรวมตัวอย่างการใช้งาน javascript ไว้มากที่สุด

Version beta





กฤษณวัฒน<sup>์</sup> แก้วแสนเมือง

# ผู้สนับสนุน



# Hosting กดลอมฟรี 30 วัน บริการติดตั้มโปรแกรมต่ามๆฟรี

99 un/tl



Built for developers. Deploy an SSD cloud server in 55 seconds!



ติดปัญหาเกี่ยวกับ HTML Css JavaScript PHP หรือ SQL โพสถามตามกระทู้หรือ Facebook Group ได้คำตอบบ้าง ไม่ได้บ้าง ถามทีนี้ได้คำตอบ 99%



# Roadmap

- 1. เพื่อเนื้อหาส่วนของ weakest, iterator, generator
- 2. แสดงว่าคำสั่งนี้ใช้กับบราวเซอร์เวอร์ชันไหนได้บ้าง
- 3. เพิ่มตัวอย่างการใช้งานกับพารามิเตอร์ทุกตัว
- 4. เพิ่มตัวอย่างการจัดการ dom
- 5. เพิ่มตัวอย่างการจัดการ CSS
- 6. เพิ่มตัวอย่างการใช้งานกับ event



2. รอรับข้อเสนองานหลากหลายตำแหน่ง

3. รับโบนัส **10,000** บาท

# สารบัญ

Array	. 1
Concat	. 1
copyWithin	. 1
entries	. 2
every	. 2
Fill	. 2
filter	. 3
find	. 4
findIndex	. 4
forEach	. 5
From	. 6
includes	. 6
indexOf	. 7
isArray	. 7
Join	. 8
lastIndexOf	. 8
Length	9
Map1	1 C



	Of	. 10
	Pop	. 11
	Push	. 11
	Reverse	. 12
	Shift	. 12
	Slice	. 13
	Some	. 13
	Sort	. 14
	Splice	. 15
	toString	. 16
	Unshift	. 16
	Values	. 17
C	ate	. 18
	getDate	. 18
	getDay	. 19
	getFullYear	. 19
	getHours	. 20
	getMilliseconds	. 20
	getMinutes	. 21
	getMonth	. 21
	getSeconds	. 22
	getTime	. 22
	getUTCDate	. 23



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส <del>10,000</del> บาท

getUTCDay	23
getUTCFullYear	24
getUTCHours	24
getUTCMilliseconds	25
getUTCMinutes	25
getUTCMonth	26
getUTCSeconds	26
Now	27
setDate	27
setFullYear	28
setHours	28
setMilliseconds	29
setMinutes	29
setMonth	30
setSeconds	30
setUTCDate	31
setUTCFullYear	31
setUTCHours	32
setUTCMilliseconds	32
setUTCMinutes	33
setUTCMonth	33
setUTCSeconds	34
toDateStringtoDateString	34



	toISOString	35
	toJSON	35
	toLocaleDateString	36
	toLocaleString	36
	toLocaleTimeString	37
	toString	37
	toTimeString	38
	toUTCString	38
E	rrors	39
	Message	39
	Name	39
	toStringtoString	40
G	lobal Objects	40
	decodeURI	40
	encodeURI	41
	isFinite	41
	isNaN	. 42
	NaN	. 43
	Null	. 44
	parseFloat	. 44
	parseInt	. 45
	Undefined	46
	Escape	. 46



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส <del>10,000</del> บาท

unescape	47
Intl	48
Collator	48
Compare	48
resolvedOptions	49
DateTimeFormat	50
supportedLocalesOf	50
NumberFormat	51
JSON	51
Parse	51
Stringify	52
Map	52
Clear	52
Delete	53
Entries	54
forEach	54
Get	55
Has	55
Keys	56
Set	57
Size	58
Values	58
WeakMap	59



	Weakest	. 60
	Weakdelete	. 60
	Weakget	. 61
	Weakhas	. 61
١	Лath	. 62
	Abs	. 62
	Acos	. 63
	Acosh	. 63
	Asin	. 63
	Asinh	. 64
	Atan	. 64
	Atanh	. 65
	Cbrt	. 65
	Ceil	. 66
	Cos	. 66
	Cosh	. 67
	E	. 67
	Floor	. 68
	log10	. 68
	log2	. 69
	Max	. 69
	Min	. 70
	DI	70



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส <del>10,000</del> บาท

Pow	71
Random	71
Round	71
Sin	72
Sqrt	72
Tan	73
Tanh	73
Miscellaneous	74
Eval	74
Number	75
isFinite	75
isInteger	75
isNaN	76
parseFloat	77
parseInt	78
toExponential	79
toFixed	79
toLocaleString	80
toString	80
Object	81
Assign	81
Create	81
defineProperties	82



	Freeze	. 83
	getOwnPropertyDescriptor	. 84
	hasOwnProperty	. 85
	Is	. 86
	isExtensible	. 87
	isFrozen	. 88
	isPrototypeOf	. 89
	isSealed	. 90
	Keys	. 91
	preventExtensions	. 91
	Seal	. 92
	toString	. 93
F	Promise	. 94
	All	. 94
	Catch	. 95
	Race	. 96
	Reject	. 96
	Resolve	. 97
	then	. 97
F	RegExp	. 98
	Match	. 98
	Replace	. 99
	Search	99



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส <del>10,000</del> บาท

	Split	. 100
	lastIndex	. 100
	Exec	. 101
	Flags	. 101
	Global	. 102
	ignoreCase	. 102
	Multiline	. 103
	Source	. 103
	Test	. 104
	toString	. 104
	Unicode	. 105
S	Set	. 105
	Add	. 105
	Clear	. 106
	Delete	. 106
	Entries	. 107
	forEach	. 107
	Has	. 108
	values	. 109
	size	. 110
S	String	. 110
	Anchor	. 110
	charAt	.111



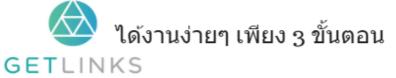
Concat	111
endsWith	112
Includes	112
indexOf	113
Length	114
Link	114
Match	115
Normalize	115
padEnd	116
padStart	116
Raw	117
repeat	117
Replace	118
Search	118
Slice	119
Split	119
startsWith	120
Substr	120
substring	121
toLowerCase	121
toString	122
toUpperCase	122
trim	122



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส <del>10,000</del> บาท

	valueOf	123
F	Proxy	124
	Construct	124
	defineProperty	125
	deleteProperty	125
	Get	126
	Apply	127
	getOwnPropertyDescriptor	128
	getPrototypeOf	128
	Has	129
	isExtensible	130
	ownKeys	131
	preventExtensions	132
	Set	133
	setPrototypeOf	134





- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส <mark>10,000</mark> บาท

# **Array**

#### Concat

ใช้สำหรับรวม array หลายๆ ตัวเข้าด้วยกัน

ตัวอย่างที่ 1

#### ผลลัพธ์

```
a,b,c,1,2,3
```

ผลลัพธ์แบบ live

# copyWithin

คัดลอกอาเรย์แล้ววางทับลงในตัวของมันเอง

#### รูปแบบการใช้งาน

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.copyWithin(2, 0);
```

array.copyWithin(target,start,end)

target คือ ตำแหน่งในอาเรย์ที่ต้องการคัดลอก

start คือ ตำแหน่งที่ต้องการแทน



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

#### entries

สร้างอาเรย์อ็อบเจกต์ขึ้นมาจากอาเรย์ธรรมดา

# ตัวอย่างที่ 1

```
var arr = ['a', 'b', 'c'];
var eArr = arr.entries();

console.log(eArr.next().value);
console.log(eArr.next().value);
console.log(eArr.next().value);
```

#### ผลลัพธ์

```
▶ [0, "a"]

▶ [1, "b"]

▶ [2, "c"]
```

# every

ใช้ตรวจสอบว่าค่าที่กำหนดไว้ผ่านเงื่อนไขทั้งหมดหรือไม่

ตัวอย่างที่ 1 ตรวจว่าอาเรย์มีค่ามากกว่า 10 ทุกตัวหรือไม่

```
function isBigEnough(element, index, array) {
    return element >= 10;
}
[12, 5, 8, 130, 44].every(isBigEnough);
[12, 54, 18, 130, 44].every(isBigEnough);
```

#### ผลลัพธ์

```
false
true
```

#### Fill

ใช้เปลี่ยนแปลงค่าในอาเรย์





#### ตัวอย่างที่ 1 แปลงทุกค่าเลย

```
var animal = ["elephant", "tiger", "cat", "dog"];
animal.fill("dolphin");
```

#### ผลลัพธ์

```
( ["dolphin", "dolphin", "dolphin"]
```

#### ตัวอย่างที่ 2 มีจุดเริ่มต้น

```
var animal = ["elephant", "tiger", "cat", "dog"];
animal.fill("dolphin",1);
```

#### ผลลัพธ์

```
( ["elephant", "dolphin", "dolphin", "dolphin"]
```

# ตัวอย่างที่ 3 มีจุดสิ้นสุด

```
var animal = ["elephant", "tiger", "cat", "dog"];
animal.fill("dolphin",1,3);
```

#### ผลลัพธ์

```
( ["elephant", "dolphin", "dolphin", "dog"]
```

#### filter

ใช้สร้างอาเรย์จากค่าที่ผ่านเงื่อนไข

ตัวอย่างที่ 1



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10**,000 บาท



```
var ages = [32, 33, 16, 40];
function checkAge(age) {
   return age <= 18;
}
ages.filter(checkAge);</pre>
```

( [16]

## find

ใช้สำหรับค้นหาข้อมูลในอาเรย์โดยใช้เงื่อนไขเป็นฟังก์ชัน

# ตัวอย่างที่ 1

```
var ages = [32, 33, 16, 40];
function checkAge(age) {
   return age >= 18;
}
ages.find(checkAge);
```

#### ผลลัพธ์

< 32

# findIndex

ใช้สำหรับค้นหาข้อมูลในอาเรย์โดยใช้เงื่อนไขเป็นฟังก์ชัน โดยจะส่งเป็นลำดับของค่าที่ตรงกลับมาแทน

# ตัวอย่างที่ 1



```
var ages = [32, 33, 16, 40];

function checkAge(age) {
   return age >= 18;
}
ages.findIndex(checkAge);
```

<· 0

#### forEach

ใช้สำหรับเรียกใช้ฟังก์ชันเพื่อส่งค่าเป็นสมาชิกในอาเรย์ในครั้งเดียว โดยฟังก์ชัน callback ที่จะใช้ งานร่วมนั้นต้องมีรูปแบบดังนี้

function callback (value, index, array)

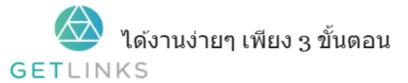
value ค่าของอาเรย์

index ลำดับของสมาชิกในอาเรย์

array ค่าที่ต้องการเสริมเข้าไป

#### ้ตัวอย่างที่ 1

```
กดที่ปุ่ม
<button onclick="numbers.forEach(formular) ">ลองดู</button>
หาผลรามของค่าในอาเรย์: <span id="summon"></span>
<script>
var sum = 0;
var numbers = [34, 232, 55, 232, 285];
function formular(item) {
    sum += item;
    summon.innerHTML = sum;
}
</script>
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

```
กดที่ปุ่ม
ลองดู
หาผลรวมของค่าในอาเรย์: 838
```

#### ผลลัพธ์แบบ live

#### From

สร้างอาเรย์โดยรับค่าอากิวเมนต์ที่รับเข้ามาเป็นอ็อบเจกต์ เช่น set, map,string

# ตัวอย่างที่ 1

```
var numbers = [4, 16, 9, 25];
var summ = numbers.map(Math.sqrt);
myvar = Array.from(summ);
console.log(myvar);
```

#### ผลลัพธ์

```
Array [ 2, 4, 3, 5 ]
```

# includes

ตรวจว่ามีค่าที่ต้องการอยู่ในอาเรย์ หรือไม่



```
var arr = [1,2,3,4,5,6,7,8,9];
var myvar = arr.includes(2);
console.log(myvar);
```

ผลลัพธ์

true

# indexOf

ตรวจหาตำแหน่งของคำที่ต้องการในอาเรย์

# ตัวอย่างที่ 1

```
var arr = [1,2,3,4,5,6,7,8,9,9];
var myvar = arr.indexOf(5);
console.log(myvar);
```

#### ผลลัพธ์

ถ้าหากว่าหาค่าไม่เจอจะคืนค่า -1

4

# isArray

ตรวจว่าข้อมูลเป็นอาเรย์หรือไม่

# ตัวอย่างที่ 1

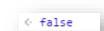
```
var arr = [];
Array.isArray(arr);
```

# ตัวอย่างที่ 2

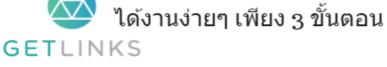
```
var arr = 33;
Array.isArray(arr);
```



2.



3. รับโบนัส <mark>10,000</mark> บาท



(S

· true

#### Join

แปลงอาเรย์ให้เป็นประโยคโดยใส่ตัวขั้นให้ด้วย

```
ตัวอย่างที่ 1
                                         ตัวอย่างที่ 2
                                         var a = ['one', 'two', 'three'];
var a = ['one', 'two', 'three'];
                                         var myVar2 = a.join(', ');
var myVar1 = a.join();
                                         console.log(myVar2);
console.log(myVar1);
ผลลัพธ์
                                         ผลลัพธ์
Wind, Rain, Fire
                                        one, two, three
                                         ตัวอย่างที่ 4
ตัวอย่างที่ 3
var a = ['one', 'two', 'three'];
                                         var a = ['one', 'two', 'three'];
var myVar3 = a.join(' + ');
                                         var myVar4 = a.join('');
console.log(myVar3);
                                         console.log(myVar4);
ผลลัพธ์
                                         ผลลัพธ์
one + two + three
                                        onetwothree
```

#### lastIndexOf

หาตำแหน่งที่ข้อมูลที่ต้องการจากหลังมาหน้า หรือจากตำแหน่งที่กำหนด



```
var arr = ["A", "B", "C", "D", "E", "A", "B", "C", "D", "E"];
var myvar = arr.lastIndexOf("B");
console.log(myvar);
```

ผลลัพธ์

6

# ตัวอย่างที่ 2

ถ้าหากว่าไม่เจอค่าที่ค้นหาจะคืนค่า -1 หรือ 1

```
var arr = [1,2,3,4,5,6,7,8,9,9];
var myvar = arr.lastIndexOf(99);
console.log(myvar);
```

ผลลัพธ์

-1

# Length

ค้นหาจำนวนสมาชิกของอาเรย์นั้น ๆ



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส <del>10,000</del> บาท

```
var arr = ["A", "B", "C", "D", "E", "A", "B", "C", "D", "E"];
var myvar = arr.length;
console.log(myvar);
```

#### ผลลัพธ์

10

# Map

สร้างอาเรย์ใหม่โดยดึงข้อมูลในอาเรย์แล้วส่งไปให้ฟังก์ชัน แล้วได้ค่าใหม่กลับออกมาโดยไม่ทำให้ค่า เดิมเสียรูป

# ตัวอย่างที่ 1

นำเอาค่าในอาเรย์มาหารากที่สองโดยที่ค่าของอาเรย์ตัวเดิม ไม่เปลี่ยนแปลงหลังจากผ่านฟังก์ชันใหม่

```
var numbers = [4, 9, 16, 25];
myvar = numbers.map(Math.sqrt);
console.log(myvar);
console.log(numbers);
```

#### ผลลัพธ์

```
Array [ 2, 3, 4, 5 ]
Array [ "4", "9", "16", "25" ]
```

#### Of

สร้างอาเรย์โดยรับอากิวเมนต์เข้ามา กี่ตัวก็ได้



```
var myvar = Array.of(1,2,3,4,5,6,7,8,9,9);
console.log(myvar);
```

#### ผลลัพธ์

```
Array [ 1, 2, 3, 4, 5, 6, 7, 8, 9, 9 ]
```

# Pop

ดึงข้อมูลตัวสุดท้ายออกมาจากอาเรย์และลบในคราวเดียวกัน

#### ตัวอย่างที่ 1

```
var arr = ["A", "B", "C", "D", "E", "F", "G", "H", "G"];
var myvar = arr.pop();
console.log(myvar);
```

ผลลัพธ์

G

#### Push

เพิ่มค่าลงไปในอาเรย์โดยเพิ่มจากท้ายสุด



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

```
var arr = ["A", "B", "C", "D", "E", "F", "G", "H"];
arr.push("I");
console.log(arr);
```

#### ผลลัพธ์

```
Array [ "A", "B", "C", "D", "E", "F", "G", "H", "I" ]
```

#### Reverse

จัดเรียงลำดับในอาเรย์ใหม่โดยเรียงจากด้านหลังขึ้นมา

# ตัวอย่างที่ 1

```
var arr = ["A", "B", "C", "D", "E", "F", "G", "H"];
arr.reverse();
console.log(arr);
```

#### ผลลัพธ์

```
Array [ "H", "G", "F", "E", "D", "C", "B", "A" ]
```

#### Shift

ดึงข้อมูลในลำดับแรกออกจากอาเรย์และใส่เข้าไปในตัวแปร



```
var arr = ["A", "B", "C", "D", "E", "F", "G", "H"];
arr.shift();
console.log(arr);
```

#### ผลลัพธ์

```
Array [ "B", "C", "D", "E", "F", "G", "H" ]
```

#### Slice

ดึงข้อมูลออกจากอาเรย์โดยกำหนดจุดเริ่มต้น และสิ้นสุด

#### ตัวอย่างที่ 1

```
var arr = ["A", "B", "C", "D", "E", "F", "G", "H"];
myvar = arr.slice(1,5);
console.log(myvar);
```

#### ผลลัพธ์

```
Array [ "B", "C", "D", "E" ]
```

#### Some

ใช้ตรวจสอบว่าสมาชิกของอาเรย์ผ่านการทดสอบจากฟังก์ชันอื่น ๆ มาแล้ว



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



```
var numbers = [1, 2, 5, 11];
function chk(num) {
    return num >= 10;
}
myvar = numbers.some(chk);
console.log(myvar);
```

#### ผลลัพธ์

#### true

# ตัวอย่างที่ 2

```
var numbers = [1, 2, 5, 11];
function chk(num) {
    return num >= 13;
}
myvar = numbers.some(chk);
console.log(myvar);
```

#### ผลลัพธ์

#### false

#### Sort

เรียงลำดับสมาชิกในอาเรย์โดยอ้างอิงจากตัวเลขและตัวอักษร



```
var numbers = [1, 12, 58, 11];
myvar = numbers.sort();
console.log(myvar);
```

#### ผลลัพธ์

```
Array [ 1, 11, 12, 58 ]
```

# ตัวอย่างที่ 2

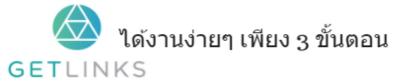
```
var arr = ["G", "B", "D", "A", "I", "C"];
myvar = arr.sort();
console.log(myvar);
```

## ผลลัพธ์

```
Array [ "A", "B", "C", "D", "G", "I" ]
```

# **Splice**

เพิ่มข้อมูลลงในอาเรย์ตามตำแหน่งที่กำหนดโดยกำหนดได้ว่าจะเพิ่มจากตำแหน่งใหน และจะลบออกกี่ ตัว



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



```
var arr = ["A", "B", "C", "D", "E", "F", "G", "H"];
arr.splice(2, 3, "X", "Y");
console.log(arr);
```

#### ผลลัพธ์

```
Array [ "A", "B", "X", "Y", "F", "G", "H" ]
```

# toString

แปลงอาเรย์ให้เป็นตัวอักษร

#### ตัวอย่างที่ 1

```
var arr = ["A", "B", "C", "D", "E", "F", "G", "H"];
myvar = arr.toString();
console.log(myvar);
```

#### ผลลัพธ์

```
A,B,C,D,E,F,G,H
```

#### **Unshift**

เพิ่มสมาชิกเข้าไปด้านหน้าของอาเรย์



```
var arr = ["A", "B", "C", "D", "E", "F", "G", "H"];
arr.unshift("x","y");
console.log(arr);
```

# ผลลัพธ์

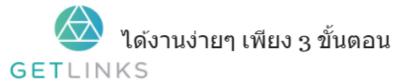
```
Array [ "X", "Y", "A", "B", "C", "D", "E", "F", "G", "H" ]
```

#### **Values**

ดึงข้อมูลในอาเรย์ออกมาสร้างเป็นอาเรย์ใหม่ โดยไม่ดึงคีย์

## ตัวอย่างที่ 1

```
var arr = ["A", "B", "C", "D"];
var myvar = arr.values();
console.log(myvar.next().value);
console.log(myvar.next().value);
console.log(myvar.next().value);
console.log(myvar.next().value);
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10**,000 บาท

А

В

C

D

# **Date**

โดยปกติแล้ว การเรียกใช้ฟังก์ชันเกี่ยวกับวันเวลา จะถูกเซ็ตให้เรียกวันเวลาปัจจุบันออกมาเสมอๆ แต่ ว่าก็สามารถกำหนดให้ดึงค่าออกมาจากเวลาที่ต้องการใด้เหมือนกัน โดยการกำหนดให้อ็อบเจกต์วัน เวลาเป็นเวลาที่ต้องการให้อยู่ในรูปแบบที่ต้องการ เช่น

```
Date("July 21, 1983 01:15:00");
```

# getDate

# ตัวอย่างที่ 1

ดึงค่าวันออกมาจากอ็อบเจกต์ของวันเวลา

```
var d = new Date();
var myvar = d.getDate();
console.log(myvar);
```



แสดงผลเป็นวันที่ปัจจุบันคือวันที่ 4 ตุลาคม 2559

4

# getDay

ดึงลำดับของวันในสัปดาห์จาก อ็อบเจกต์วันเวลา โดยนับวันอาทิตย์เป็นวันแรก

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getDay();
console.log(myvar);
```

#### ผลลัพธ์

แสดงผลเป็นวันอังคาร โดย ที่ปัจจุบันคือวันอังคาร ที่ 4 ตุลาคม 2559

2

# getFullYear

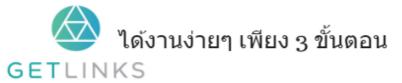
ดึงค่าเลข ค.ศ. 4 หลัก ออกมาจากอ็อบเจกต์วันเวลา

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getFullYear();
console.log(myvar);
```

#### ผลลัพธ์

แสดงแลขปีปัจจุบันคือ ค.ศ. 2016



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10**,000 บาท



# getHours

ดึงค่าชั่วโมงออกมาจากอ็อบเจกต์วันเวลา

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getHours();
console.log(myvar);
```

#### ผลลัพธ์

แสดงเวลาปัจจุบันคือ 18.06 นาฬิกา

18

# getMilliseconds

ดึงค่าเวลาที่ไมโครวิ ออกมาจากอ็อบเจกต์วันเวลา โดยจะนับใหม่ทุก ๆ 1วินาที ซึ้งค่าที่ได้จะมีค่าตั้งแต่
0 - 1000

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getMilliseconds();
console.log(myvar);
```

#### ผลลัพธ์

495



# getMinutes

ดึงค่าเวลาส่วนของนาที ออกมาจากอ็อบเจกต์วันเวลา

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getMinutes();
console.log(myvar);
```

#### ผลลัพธ์

แสดงผลเวลาปัจจุบัน 18.18 นาฬิกา

18

# getMonth

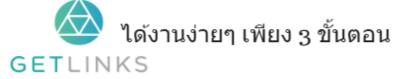
ดึงค่าเดือนออกมาจากอ็อบเจกต์วันเวลา

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getMonth();
console.log(myvar);
```

## ผลลัพธ์

แสดงหมายเลยประจำเดือนประจุบันที่นับจากเดือนมกราคม



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

# getSeconds

ดึงค่าวินาทีออกมาจากอ็อบเจกต์วันเวลา

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getSeconds();
console.log(myvar);
```

#### ผลลัพธ์

แสดงเลขวินาทีปัจจุบัน เวลา 18.18 :34

34

# getTime

ดึงค่ามิลลิวินาที ออกมาจากอ็อบเจกต์วันเวลาซึ่งฟังก์ชันนี้เริ่มนับเวลาตั้งแต่ 1970/01/01 00.00 นาฬิกา แล้วนับมาเรื่อยๆ จนถึงปัจจุบัน ซึ่งนับเป็นวินาที

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getTime();
console.log(myvar);
```

## ผลลัพธ์



## getUTCDate

ดึงค่าวันที่ ออกมาจากอ็อบเจกต์วันเวลาโดยอ้างอิงจากเวลามาตรฐานสากล

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getUTCDate();
console.log(myvar);
```

#### ผลลัพธ์

แสดงวันที่ปัจจุบันของเวลามาตรฐานสากล ซึ่งเป็นวันที่ 4 ตุลาคม 2016

4

# getUTCDay

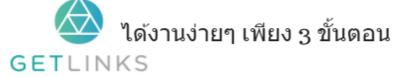
ดึงค่าลำดับของวันในสัปดาห์ ออกมาจากอ็อบเจกต์วันเวลาโดยอ้างอิงจากเวลามาตรฐานสากล และเริ่ม นับจากวันอาทิตย์

## ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getUTCDay();
console.log(myvar);
```

## ผลลัพธ์

แสดงเลขวันประจำสัปดาห์ซึ่งในปัจจุบันวันนี้ตามเวลาสากลเป็นวันอังคาร



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

# getUTCFullYear

ดึงค่าปี ค.ศ. ออกมาจากอ็อบเจกต์วันเวลาโดยอ้างอิงจากเวลามาตรฐานสากล

## ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getUTCFullYear();
console.log(myvar);
```

#### ผลลัพธ์

แสดง ค.ศ. ปัจจุบันของเวลามาตรฐานสากล ซึ่งเป็นวันที่ 4 ตุลาคม 2016

2016

# getUTCHours

ดึงค่าชั่วโมง ออกมาจากอ็อบเจกต์วันเวลาโดยอ้างอิงจากเวลามาตรฐานสากล

## ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getUTCHours();
console.log(myvar);
```

#### ผลลัพธ์

แสดงเวลา ชั่วโมงปัจจุบัน ของเวลามาตรฐานสากล



# getUTCMilliseconds

ดึงค่ามิลลิวินาที ออกมาจากอ็อบเจกต์วันเวลาโดยอ้างอิงจากเวลามาตรฐานสากล

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getUTCMilliseconds();
console.log(myvar);
```

ผลลัพธ์

364

# getUTCMinutes

ดึงค่านาที ออกมาจากอ็อบเจกต์วันเวลาโดยอ้างอิงจากเวลามาตรฐานสากล

## ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getUTCMinutes();
console.log(myvar);
```

ผลลัพธ์



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

## getUTCMonth

ดึงค่าเดือน ออกมาจากอ็อบเจกต์วันเวลาโดยอ้างอิงจากเวลามาตรฐานสากล

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getUTCMonth();
console.log(myvar);
```

#### ผลลัพธ์

แสดงเลขประจำเดือน ของเดือนปัจจุบันตามเวลามาตรฐานสากล ซึ่งเป็นวันที่ 4 ตุลาคม 2016

9

# getUTCSeconds

ดึงค่าวินาที ออกมาจากอ็อบเจกต์วันเวลาโดยอ้างอิงจากเวลามาตรฐานสากล

## ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.getUTCSeconds();
console.log(myvar);
```

#### ผลลัพธ์

แสดงค่าเลขนาทีปัจจุบันที่ 18.30.57



#### Now

ดึงค่ามิลินาทีโดยเริ่มนับจาก วันที่ 1 มกราคม 1970

ตัวอย่างที่ 1

```
var myvar = Date.now();
console.log(myvar);
```

ผลลัพธ์

1475580626650

#### setDate

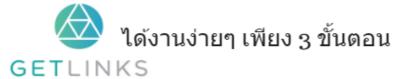
ใช้กำหนดวันในอ็อบเจกต์ Date

ตัวอย่างที่ 1

```
var d = new Date();
d.setDate(25);
console.log(d);
```

ผลลัพธ์

Date 2016-10-25T11:32:45.934Z



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

## setFullYear

ใช้กำหนดปีในอ็อบเจกต์ Date

# ตัวอย่างที่ 1

```
var d = new Date();
d.setFullYear(2111)
console.log(d);
```

#### ผลลัพธ์

Date 2111-10-04T11:34:02.101Z

#### setHours

ใช้กำหนดชั่วโมงในอ็อบเจกต์ Date

## ตัวอย่างที่ 1

```
var d = new Date();
d.setHours(12)
console.log(d);
```

### ผลลัพธ์

เนื่องจากว่าเวลาที่ตัวเซิฟเวอร์ในการรันโปรแกรมทดสอบนั้นเป็นเวลาของประเทศไทย จึงทำให้ค่าที่ได้ จะต้องลบกับเวลาตามเขตประเทศไทยอีก ซึ่งก็คือเวลา 05 นาฬิกา (12 - 07)

Date 2016-10-04T05:37:20.093Z



## setMilliseconds

ใช้กำหนดมิลลิวินาทีในอ็อบเจกต์ Date ซึ่งจะมีค่าตั้งแต่ 0 - 1000

# ตัวอย่างที่ 1

```
var d = new Date();
d.setMilliseconds(555);
console.log(d);
```

ผลลัพธ์

Date 2016-10-04T11:42:39.555Z

#### setMinutes

ใช้กำหนดนาทีในอ็อบเจกต์ Date

## ตัวอย่างที่ 1

```
var d = new Date();
d.setMinutes(22)
console.log(d);
```

ผลลัพธ์

Date 2016-10-04T11:22:43.159Z



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

#### setMonth

ใช้กำหนดเดือนในอ็อบเจกต์ Date

# ตัวอย่างที่ 1

```
var d = new Date();
d.setMonth(11);
console.log(d);
```

#### ผลลัพธ์

Date 2016-12-04T11:44:36.463Z

#### setSeconds

ใช้กำหนดวินาทีในอ็อบเจกต์ Date

## ตัวอย่างที่ 1

```
var d = new Date();
d.setSeconds(33);
console.log(d);
```

## ผลลัพธ์

Date 2016-10-04T11:50:33.554Z



### setUTCDate

ใช้กำหนดวันในอ็อบเจกต์ Date ตามมาตรฐาน UTC

## ตัวอย่างที่ 1

```
var d = new Date();
d.setUTCDate(20);
console.log(d);
```

ผลลัพธ์

Date 2016-10-20T11:51:50.552Z

#### setUTCFullYear

ใช้กำหนดปีในอ็อบเจกต์ Date ตามมาตรฐาน UTC

## ตัวอย่างที่ 1

```
var d = new Date();
d.setUTCFullYear(2222);
console.log(d);
```

ผลลัพธ์

Date 2222-10-04T11:53:00.199Z



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

#### setUTCHours

ใช้กำหนดชั่วโมงในอ็อบเจกต์ Date ตามมาตรฐาน UTC

# ตัวอย่างที่ 1

```
var d = new Date();
d.setUTCHours(00);
console.log(d);
```

#### ผลลัพธ์

Date 2016-10-04T00:53:46.768Z

## setUTCMilliseconds

ใช้กำหนดนาทีในอ็อบเจกต์ Date ตามมาตรฐาน UTC

## ตัวอย่างที่ 1

```
var d = new Date();
d.setUTCMilliseconds(0);
console.log(d);
```

#### ผลลัพธ์

Date 2016-10-04T11:54:30.000Z



### setUTCMinutes

ใช้กำหนดนาทีในอ็อบเจกต์ Date ตามมาตรฐาน UTC

## ตัวอย่างที่ 1

```
var d = new Date();
d.setUTCMinutes(55);
console.log(d);
```

ผลลัพธ์

Date 2016-10-04T11:55:13.896Z

### setUTCMonth

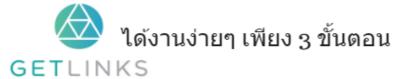
ใช้กำหนดเดือนในอ็อบเจกต์ Date ตามมาตรฐาน UTC

# ตัวอย่างที่ 1

```
var d = new Date();
d.setUTCMonth(5);
console.log(d);
```

ผลลัพธ์

Date 2016-06-04T11:55:55.510Z



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

### setUTCSeconds

ใช้กำหนดวินาทีในอ็อบเจกต์ Date ตามมาตรฐาน UTC

# ตัวอย่างที่ 1

```
var d = new Date();
d.setUTCSeconds(01);
console.log(d);
```

ผลลัพธ์

Date 2016-10-04T11:56:01.824Z

# toDateString

แปลงอ็อบเจกต์ Date ให้เป็นตัวอักษร

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.toString();
console.log(myvar);
```

#### ผลลัพธ์

Tue Oct 04 2016 18:58:21 GMT+0700 (SE Asia Standard Time)



# toISOString

แปลงอ็อบเจกต์ Date ให้เป็นตัวอักษรในรูปแบบของ UTC

## ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.toISOString();
console.log(myvar);
```

ผลลัพธ์

2016-10-04T11:59:16.221Z

#### toJSON

แปลงอ็อบเจกต์ Date ให้เป็นตัวอักษรในรูปแบบของ json

## ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.toJSON();
console.log(myvar);
```

ผลลัพธ์

2016-10-04T12:00:56.430Z



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



# toLocaleDateString

กำหนดรูปแบบของวันให้เป็นตามรูปแบบที่ประเทศนั้นใช้

## ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.toLocaleDateString();
console.log(myvar);
```

ผลลัพธ์

2016-10-04T12:01:37.729Z

# toLocaleString

กำหนดรูปแบบของวันเวลาให้เป็นตามรูปแบบที่ประเทศนั้นใช้

## ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.toLocaleString();
console.log(myvar);
```

ผลลัพธ์

10/4/2016, 7:02:38 PM



# toLocaleTimeString

กำหนดรูปแบบของเวลาให้เป็นตามรูปแบบที่ประเทศนั้นใช้

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.toLocaleTimeString();
console.log(myvar);
```

ผลลัพธ์

7:03:17 PM

# toString

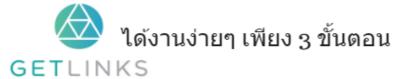
แปลงอ็อบเจกต์ Date ให้เป็นประโยค

# ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.toString();
console.log(myvar);
```

ผลลัพธ์

Tue Oct 04 2016 19:04:14 GMT+0700 (SE Asia Standard Time)



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

# toTimeString

แปลงอ็อบเจกต์ Date ให้เป็นประโยค แต่เลือกแสดงเฉพาะค่าเวลา

## ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.toTimeString();
console.log(myvar);
```

#### ผลลัพธ์

```
Tue Oct 04 2016 19:05:42 GMT+0700 (SE Asia Standard Time)
```

# toUTCString

แปลงอ็อบเจกต์ของ Date ให้อยู่ในรูปแบบของ UTC

## ตัวอย่างที่ 1

```
var d = new Date();
var myvar = d.toUTCString();
console.log(myvar);
```

ผลลัพธ์

Tue, 04 Oct 2016 12:06:36 GMT



# **Errors**

เป็นออปเจคต์ที่ใช้จัดการข้อผิดพลาด

# Message

ใช้ตั้งค่าข้อความแสดงข้อผิดพลาด

ตัวอย่างที่ 1

```
try {
    allllert("ok not error");
}
catch(err) {
    console.log('you write alert wrong');
}
```

ผลลัพธ์

you write alert wrong

#### Name

ตั้งชื่อให้ error

ตัวอย่างที่ 1

```
var name_err = new Error('my name is Error');
name_err.name = 'alllllllert';
throw name_err;
```

ผลลัพธ์

Error: my name is Error



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



# toString

แปลงอ็อบเจกต์ให้กลายเป็นข้อความ

## ตัวอย่างที่ 1

```
var myobj = new Error('error error xxx');
console.log(myobj.toString());
```

ผลลัพธ์

Error: error error xxx

# **Global Objects**

#### decodeURI

ถอดรหัสค่าที่ถูกเข้ารหัสโดย encodeURI

## ตัวอย่างที่ 1

```
var tt = "A%20B%20C%20D%201%202%203";
var myvar = decodeURI(tt);
console.log(myvar);
```

ผลลัพธ์

A B C D 1 2 3



## encodeURI

ใช้เข้ารหัสค่า URI เพื่อป้องกันในกรณีมีการส่งอักขระพิเศษ

ตัวอย่างที่ 1

```
var tt = "A B C D 1 2 3";
var myvar = encodeURI(tt);
console.log(myvar);
```

ผลลัพธ์

A%20B%20C%20D%201%202%203

### isFinite

ใช้ตรวจสอบว่าตัวเลขเป็นค่าที่มีที่สิ้นสุดไหม

ตัวอย่างที่ 1

```
var tt = "97567";
var myvar = isFinite(tt);
console.log(myvar);
```

ผลลัพธ์

true



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

## ตัวอย่างที่ 2

```
var tt = "ABCD";
var myvar = isFinite(tt);
console.log(myvar);
```

ผลลัพธ์

false

## isNaN

ใช้ตรวจสอบว่าตัวเลขเป็นค่า NaN หรือไม่

# ตัวอย่างที่ 1

```
var tt = "ABCD";
var myvar = isNaN(tt);
console.log(myvar);
```

ผลลัพธ์

true

# ตัวอย่างที่ 2

```
var tt = "123";
var myvar = isNaN(tt);
console.log(myvar);
```



## ผลลัพธ์

# false

### NaN

ย่อมาจาก Not A Number แปลว่าค่าที่ไม่ใช่ตัวเลข

# ตัวอย่างที่ 1

```
var a = 9;
var b = 2;
var c = a - b;
console.log(c);
```

#### ผลลัพธ์

7

# ตัวอย่างที่ 2

```
var a = 9;
var b = "ABCD";
var c = a - b;
console.log(c);
```

### ผลลัพธ์

#### NaN



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

## Null

ใช้เพื่อกำหนดให้ตัวแปรหรือตรวจดูว่าตัวแปรนั้นเป็นค่าว่าง หรือไม่

# ตัวอย่างที่ 1

```
var tt;
if(tt == null) {
    console.log('empty');
}else{
    console.log('exist');
}
```

ผลลัพธ์

empty

# parseFloat

แปลงตัวอักษรทุกอย่างให้กลายเป็นเลขที่มีจุดทศนิยม

# ตัวอย่างที่ 1

```
var tt = "ABCD123";
myvar = parseFloat(tt);
console.log(myvar);
```



## ผลลัพธ์

#### NaN

# ตัวอย่างที่ 2

```
var tt = "84766";
myvar = parseFloat(tt);
console.log(myvar);
```

ผลลัพธ์

84766

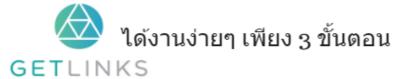
# parseInt

แปลงตัวเลขระบบฐานเลข

# ตัวอย่างที่ 1

```
var myvar = parseInt(" 0xF", 16);
console.log(myvar);
```

ผลลัพธ์



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

### **Undefined**

ตรวจสอบว่าตัวแปรถูกกำหนดค่าหรือพร้อมใช้งานหรือเปล่า

## ตัวอย่างที่ 1

```
var x;
if(x === undefined) {
    console.log('var is undefined');
}else{
    console.log('exist');
}
```

ผลลัพธ์

var is undefined

# Escape

แปลงอักขระพิเศษให้เป็นเลขฐานสิบหก

# ตัวอย่างที่ 1

```
var tt = "ABC 123 / * ? .";
var myvar = escape(tt);
console.log(myvar);
```



#### ผลลัพธ์

# ABC%20123%20/%20\*%20%3F%20.

## unescape

ถอดรหัสค่าที่ถูกเขารหัสโดย escape

ตัวอย่างที่ 1

```
var tt = "ABC%20123%20/%20*%20%3F%20.";
var myvar = unescape (tt);
console.log(myvar);
```

ผลลัพธ์

ABC 123 / \* ? .



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

# Intl

เป็นออปเจคต์ที่ใช้จัดการรูปแบบของข้อมูลให้สอดคล้องกับภาษาและท้องถิ่นของประเทศนั้นๆ เช่น ภาษา,ค่าเงิน,วันเวลา,ตัวเลข

## Collator

ใช้ในการเปรียบเทียบระบบภาษาซึ่งสามารถเปรียบเทียบได้ทั้งภาษาเดียวกัน หรือภาษาคนละภาษาก็ได้ เช่น a กับ b จะได้ -1 เนื่องจาก ตามลำดับแล้ว a มาก่อน b แต่ถ้าเปรียบเทียบ a กับ a จะได้ค่า 1 เนื่องจาก a แต่ว่าถ้าเราเปรียบเทียบ a กับ a จะได้ a เนื่องจากว่าเป็นค่าเดียวกัน

# ตัวอย่างที่ 1

```
var col = new Intl.Collator().compare('a', 'b');
console.log(col);
```

## ผลลัพธ์

-1

## Compare

ใช้เปรียบเทียบระบบภาษาจะเป็นภาษาเดียวกัน หรือคนละภาษาก็ได้



## ตัวอย่างที่ 1

```
var col = new Intl.Collator('en');
var chk = col.compare('a', 'A');
console.log(chk);
```

### ผลลัพธ์

เนื่องจากว่า **a** มาก่อน **A** จึงได้ค่า -1

-1

## resolvedOptions

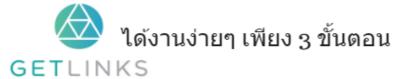
ดึงตัวแปรที่ใช้ตั้งค่าต่างๆ ของอ็อบเจกต์ collator มาแสดง

## ตัวอย่างที่ 1

```
var col = new Intl.Collator('en', { sensitivity: 'base' });
var chk = col.resolvedOptions();
console.log(chk);
```

#### ผลลัพธ์

```
Object { locale: "en", usage: "sort",
sensitivity: "base", ignorePunctuation:
false, collation: "default", numeric: false
}
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

#### **DateTimeFormat**

#### **Format**

จัดรูปแบบวันเวลาให้อยู่ในรูปแบบ ภาษาที่ประเทศนั้นใช้ จากตัวอย่างจะแปลงให้เป็นวันเวลาของ ประเทศสหัสอเมริกา ซึ่งสังเกตได้จาก en-US

## ตัวอย่างที่ 1

```
var date = new Date(Date.UTC(2016, 10, 6));
console.log(new Intl.DateTimeFormat('en-US').format(date));
```

ผลลัพธ์

11/6/2016

# supportedLocalesOf

ดึงค่าภาษาที่อ็อบเจกต์สนับสนุนการใช้งาน ซึ่งในตัวอย่างจะใช้ภาษาเยอรมัน

## ตัวอย่างที่ 1

```
var loc = ['ban', 'id-u-co-pinyin', 'de-ID'];
var opt = { localeMatcher: 'lookup' };
var test = Intl.DateTimeFormat.supportedLocalesOf(loc, opt);
console.log(test);
```

## ผลลัพธ์

```
Array [ "id-u-co-pinyin", "de-ID" ]
```



#### **NumberFormat**

#### **Format**

จัดรูปแบบตัวเลขให้อยู่ในรูปแบบ ภาษาที่ประเทศนั้นใช้

## ตัวอย่างที่ 1

```
var num = 1123456;
var test = new Intl.NumberFormat('en-IN').format(num);
console.log(test);
```

ผลลัพธ์

11,23,456

# **JSON**

## Parse

แปลงสตริงให้กลายเป็นสคริปที่ใช้ในถาษา json

## ตัวอย่างที่ 1

```
var myvar = JSON.parse('[1, 8, "A", "B", "false"]');
console.log(myvar);
```

ผลลัพธ์

Array [ 1, 8, "A", "B", "false" ]



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



# Stringify

แปลงอ็อบเจกต์ต่างๆ ใน javascript ให้เป็น json

#### ตัวอย่างที่ 1

```
var obj = {A:"ant", b:"birt", c:"cat"};
JSON.stringify(obj);
console.log(obj);
```

#### ผลลัพธ์

```
Object { A: "ant", b: "birt", c: "cat" }
```

# Map

เป็นออปเจคต์ที่เก็บข้อมูลเป็น key และ value

## Clear

ล้างข้อมูลทั้งหมดออกจากอ็อบเจกต์ของ Set

## ตัวอย่างที่ 1

```
var map = new Map();
map.set("a", "b");
map.size;
map.clear();
console.log(map);
```



#### ผลลัพธ์

# *Map* { }

### Delete

ลบข้อมูลที่กำหนดออกจากอ็อบเจกต์ ซึ่งถ้าสำเร็จจืนค่า true ไม่สำเร็จจะคืนค่า false

# ตัวอย่างที่ 1

```
var map = new Map();
var set = map.set("A", "B");
var del = map.delete("A");
console.log(del);
```

ผลลัพธ์

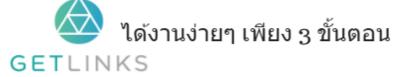
#### true

## ตัวอย่างที่ 2

```
var map = new Map();
var set = map.set("A", "B");
var del = map.delete("C");
console.log(del);
```

ผลลัพธ์

false



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

#### **Entries**

แปลงอ็อบเจกต์ให้เป็น Itarator

## ตัวอย่างที่ 1

```
var map = new Map();
map.set("A", "x");
map.set("B", "y");
map.set("C", "z");
var test = map.entries();
console.log(test.next().value);
console.log(test.next().value);
console.log(test.next().value);
```

#### ผลลัพธ์

```
Array [ "A", "x" ]

Array [ "B", "y" ]

Array [ "C", "z" ]
```

## forEach

ช่วยให้จัดการสมาชิกอาเรย์โดยการวนลูปได้ง่ายขึ้น

## ตัวอย่างที่ 1

```
function test(ind, valu, map) {
   console.log(ind + '=>' + valu);
}
new Map([["A", 'x'], ["B", 'y'], ["C", 'z']]).forEach(test);
```



#### ผลลัพธ์

X = > A

y=>B

z=>C

#### Get

ดึงข้อมูลของสมาชิกในอ็อบเจกต์ Map

## ตัวอย่างที่ 1

```
var map = new Map();
map.set("A", "x");
map.set("B", "y");
var test = map.get("A");
console.log(test);
```

ผลลัพธ์

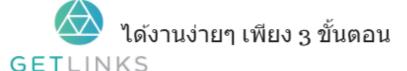
х

## Has

ตรวจสอบว่ามีสมาชิกที่กำหนดไหม

## ์ ตัวอย่างที่ 1

```
var map = new Map();
map.set("A", "x");
map.set("B", "y");
var test = map.get("A");
console.log(test);
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



#### ผลลัพธ์

## true

# ตัวอย่างที่ 2

```
var map = new Map();
map.set("A", "x");
map.set("B", "y");
var test = map.has("C");
console.log(test);
```

ผลลัพธ์

false

# Keys

เข้าถึงค่าคีย์ในอ็อบเจกต์



```
var map = new Map();
map.set("A", "x");
map.set("B", "y");
map.set("C", "z");
var test = map.keys();
console.log(test.next().value);
console.log(test.next().value);
console.log(test.next().value);
```

#### ผลลัพธ์

Α

В

C

## Set

เพิ่มมสมาชิกลงในอ็อบเจกต์

## ตัวอย่างที่ 1

```
var map = new Map();
map.set("A", "x");
map.set("B", "y");
map.set("C", "z");
console.log(map);
```

## ผลลัพธ์

```
Map { A: "x", B: "y", C: "z" }
```



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



## Size

ใช้หาขนาดของอ็อบเจกต์ Map

# ตัวอย่างที่ 1

```
var map = new Map();
map.set("A", "1");
map.set("B", "2");
map.set("C", "3");
map.set("D", "4");
map.set("E", "5");
map.set("F", "6");
console.log(map.size);
```

#### ผลลัพธ์

6

## **Values**

ใช้ดึงค่าในตำแหน่งที่กำหนด



```
var map = new Map();
map.set("A", "1");
map.set("B", "2");
map.set("C", "3");
var test = map.values();
console.log(test.next().value);
console.log(test.next().value);
console.log(test.next().value);
```

#### ผลลัพธ์

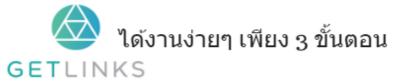
1

2

3

# WeakMap

คือออปเจคต์ Map ที่จะถูกลบทันทีที่ถูกเซตค่าให้เป็น null



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10**,000 บาท



#### Weakest

ใช้สำหรับเพิ่มค่าลงในอ็อบเจกต์

# ตัวอย่างที่ 1

```
var wm = new WeakMap();
var obj = {};
var add = wm.set(obj, "ABCD");
console.log(add);
```

#### ผลลัพธ์

```
WeakMap { Object: "ABCD" }
```

## Weakdelete

ลบข้อมูลออกจาก WeakMap แบบรายตัว

## ตัวอย่างที่ 1

```
var wm = new WeakMap();
wm.set(window, "ABCD");
wm.delete(window);
var chk = wm.has(window);
console.log(chk);
```

## ผลลัพธ์

## false



```
var wm = new WeakMap();
wm.set(window, "ABCD");
var chk = wm.has(window);
console.log(chk);
```

ผลลัพธ์

true

# Weakget

ใช้ดึงค่าออกจาก Weakmap

## ตัวอย่างที่ 1

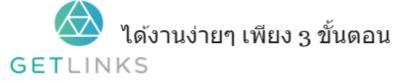
```
var wm = new WeakMap();
wm.set(window, "ABCD");
var chk = wm.get(window);
console.log(chk);
```

ผลลัพธ์

**ABCD** 

## Weakhas

ใช้ตรวจสอบว่ามีค่าหรือไม่



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

```
var wm = new WeakMap();
wm.set(window, "ABCD");
var chk = wm.has(window);
console.log(chk);
```

ผลลัพธ์

true

# Math

ในการเรียกใช้ เมท็อดของ จะต้องประกาศ Math ก่อนทุกครั้ง

### Abs

ใช้คำนวณหาค่าสัมบูรณ์

# ตัวอย่างที่ 1

```
var a = 9;
var b = 20;
var c = Math.abs(a - b);
console.log(c);
```



#### ผลลัพธ์

11

#### Acos

หาค่าอาร์คโคไซน์

# ตัวอย่างที่ 1

```
var myvar = Math.acos(0.1)
console.log(myvar);
```

#### ผลลัพธ์

#### 1.4706289056333368

#### Acosh

ใช้หาค่า อินเวิร์ส ไฮเปอโบลิก โคไซน์

## ตัวอย่างที่ 1

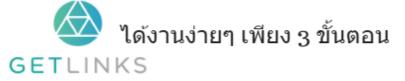
```
var myvar = Math.acosh(5)
console.log(myvar);
```

## ผลลัพธ์

## 2.2924316695611777

## Asin

หาค่าอาร์คไซน์



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



```
var myvar = Math.asin(0.8)
console.log(myvar);
```

#### ผลลัพธ์

#### 0.9272952180016123

#### Asinh

ใช้หาค่าไฮเปอโบลิก อาร์คไซน์

## ตัวอย่างที่ 1

```
var myvar = Math.asinh(0.9);
console.log(myvar);
```

#### ผลลัพธ์

0.8088669356527826

## **Atan**

หาค่าอาร์คแทน



```
var myvar = Math.atan(0.9);
console.log(myvar);
```

#### ผลลัพธ์

## 0.7328151017865066

#### **Atanh**

ใช้หาค่าไฮเปอโบลิก อาร์คแทน

# ตัวอย่างที่ 1

```
var myvar = Math.atanh(0.9);
console.log(myvar);
```

## ผลลัพธ์

#### 1.4722194895832204

## Cbrt

ใช้ถอดรูทสำหรับค่าที่ถูกยกกำลัง 3



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



```
var myvar = Math.cbrt(8);
console.log(myvar);
```

ผลลัพธ์

2

## Ceil

ใช้สำหรับปัดเศษทศนิยมขึ้น

# ตัวอย่างที่ 1

```
var myvar = Math.ceil(0.99999998);
console.log(myvar);
```

## ผลลัพธ์

1

## Cos

หาค่าโคไซน์



```
var myvar = Math.cos(45);
console.log(myvar);
```

#### ผลลัพธ์

#### 0.5253219888177297

## Cosh

ใช้หาค่าไฮเปอโบลิก อาร์คโคไซน์

## ตัวอย่างที่ 1

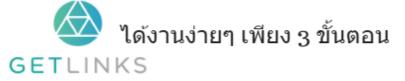
```
var myvar = Math.cosh(1);
console.log(myvar);
```

## ผลลัพธ์

#### 1.543080634815244

## Ε

หาค่า ลอการิทึมฐาน e



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



```
var myvar = Math.E;
console.log(myvar);
```

#### ผลลัพธ์

#### 2.718281828459045

## Floor

ใช้ปัดเศษทศนิยมขึ้นและถ้าปัดขึ้นไม่ได้ก็ลบทศนิยมออกไป

# ตัวอย่างที่ 1

```
var myvar = Math.floor(99.994);
console.log(myvar);
```

ผลลัพธ์

99

# log10

าค่า ลอการิทึมฐาน 10



```
var myvar = Math.log10(100);
console.log(myvar);
```

ผลลัพธ์

2

# log2

หาค่า ลอการิทึมฐาน 2

# ตัวอย่างที่ 1

```
var myvar = Math.log2(32);
console.log(myvar);
```

ผลลัพธ์

5

## Max

หาค่าที่มากที่สุด



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



```
var myvar = Math.max(1,2,3,4,5,6,7,8,9,20);
console.log(myvar);
```

ผลลัพธ์

20

#### Min

หาค่าที่น้อยที่สุด

# ตัวอย่างที่ 1

```
var myvar = Math.min(1,2,3,4,5,6,7,8,9,20);
console.log(myvar);
```

ผลลัพธ์

1

### PI

ดึงค่าพายออกมาใช้งาน

## ตัวอย่างที่ 1

```
var myvar = Math.PI;
console.log(myvar);
```

#### ผลลัพธ์

## 3.141592653589793



#### Pow

ใช้สำหรับยกกำลัง

ตัวอย่างที่ 1

```
var myvar = Math.pow(2,3);
console.log(myvar);
```

ผลลัพธ์

8

# Random

ใช้สำหรับสุ่มเลขออกมา

ตัวอย่างที่ 1

```
var myvar = Math.random();
console.log(myvar);
```

ผลลัพธ์

0.7531884365465699

# Round

ลบจุดทศนิยมทิ้งไป



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



```
var myvar = Math.round(25.98);
console.log(myvar);
```

ผลลัพธ์

26

## Sin

ใช้หาค่าไซน์

ตัวอย่างที่ 1

```
var myvar = Math.sin(45);
console.log(myvar);
```

ผลลัพธ์

0.8509035245341184

# Sqrt

ใช้หาค่าสแควรูท



```
var myvar = Math.sqrt(9);
console.log(myvar);
```

ผลลัพธ์

3

#### Tan

ใช้สำหรับหาค่าแทนเจนต์

## ตัวอย่างที่ 1

```
var myvar = Math.tan(30);
console.log(myvar);
```

ผลลัพธ์

-6.405331196646276

## **Tanh**

ใช้สำหรับหาค่าไฮเปอโบลิก แทนเจนต์



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



```
var myvar = Math.tanh(1);
console.log(myvar);
```

#### ผลลัพธ์

0.7615941559557649

# Miscellaneous

#### **Eval**

เปรียบเทียบคำหรือประโยคให้เป็นตัวแปร

# ตัวอย่างที่ 1

```
var a = 3;
var c = eval(a *6);
console.log(c);
```

ผลลัพธ์

18



# Number

## isFinite

ใช้ตรวจว่าเป็นตัวเลขที่มีค่าสิ้นสุด

ตัวอย่างที่ 1

```
tt = 856.84656;
var myvar = isFinite(tt);
console.log(myvar);
```

ผลลัพธ์

true

ตัวอย่างที่ 2

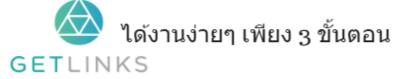
```
tt = '254Abc';
var myvar = isFinite(tt);
console.log(myvar);
```

ผลลัพธ์

false

# isInteger

ใช้ตรวจว่าเป็นตัวเลขหรือไม่



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

```
var tt = 12345;
var myvar = Number.isInteger(tt);
console.log(myvar);
```

ผลลัพธ์

true

## ตัวอย่างที่ 2

```
var tt = "ddd12345";
var myvar = Number.isInteger(tt);
console.log(myvar);
```

ผลลัพธ์

false

#### isNaN

ใช้ตรวจว่าเป็นค่า NaN หรือไม่

## ตัวอย่างที่ 1

```
var myvar = Number.isNaN(NaN);
console.log(myvar);
```



#### ผลลัพธ์

#### true

## ตัวอย่างที่ 2

```
var myvar = Number.isNaN(123);
console.log(myvar);
```

#### ผลลัพธ์

false

# parseFloat

ใช้จัดการตัวอักษรให้กลายเป็นจำนวนเต็มหรือจำนวนที่มีจุดทศนิยม

## ตัวอย่างที่ 1

```
var tt = '0.99999999';
var myvar = parseFloat(tt);
console.log(myvar);
```

## ผลลัพธ์

## 0.99999999



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



```
var tt = 'ABcd';
var myvar = parseFloat(tt);
console.log(myvar);
```

ผลลัพธ์

NaN

# parseInt

ใช้จัดการตัวอักษรให้กลายเป็นจำนวนเต็ม

# ตัวอย่างที่ 1

```
var tt = '123ABcd';
var myvar = parseInt(tt);
console.log(myvar);
```

ผลลัพธ์

123



# toExponential

ใช้สำหรับหาค่าเอ็กโพเนนเชียล

# ตัวอย่างที่ 1

```
var tt = 7789;
var myvar = tt.toExponential();
console.log(myvar);
```

ผลลัพธ์

7.789e+3

## toFixed

ฟิกจำนวนเลขหลังจุดทรรศนิยม และปัดเศษ

## ตัวอย่างที่ 1

```
var tt = 1.123456789;
var myvar = tt.toFixed(5);
console.log(myvar);
```

ผลลัพธ์

#### 1.12346



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



# toLocaleString

แปลงตัวเลขให้เป็นภาษาที่กำหนด

# ตัวอย่างที่ 1

ผลลัพธ์

THB789.12

# toString

ใช้แปลงค่าเลขเป็นตัวอักษร

## ตัวอย่างที่ 1

```
var tt = 789.123456789;
var myvar = tt.toString();
console.log(myvar);
```

ผลลัพธ์

789.123456789

ตัวอย่างที่ 2



```
var myvar = (-0xff).toString(2);
console.log(myvar);
```

ผลลัพธ์

-111111111

# Object

# **Assign**

เพิ่มข้อมูลลงไปในอ็อบเจกต์

ตัวอย่างที่ 1

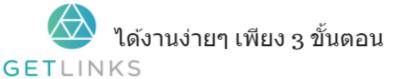
```
var obj1 = { a: 1234 };
var obj2 = Object.assign({}, obj1);
console.log(obj2);
```

ผลลัพธ์

Object { a: 1234 }

## Create

สร้างอ็อบเจกต์ใหม่โดยจะไม่รันโค้ดใน Constructor



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10**,000 บาท



```
var Animal = {traits: {},};
var lion = Object.create(Animal);
lion.traits.legs = 4;
var bird = Object.create(Animal);
bird.traits.legs = 2;
console.log(lion.traits.legs);
```

ผลลัพธ์

2

# defineProperties

ใช้สำหรับกำหนด propertie ให้อ็อบเจกต์

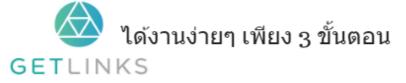


#### ผลลัพธ์

property1: "1234" property2: "ABCD"

## Freeze

ป้องกันการเพิ่ม ลบ แก้ไข กับอ็อบเจกต์



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

#### ผลลัพธ์

```
TypeError: can't define property
"property1": Object is not extensible
```

# getOwnPropertyDescriptor

ดึงข้อมูลของ property



```
var obj = {
   val: "1234",
   writable: true,
   enumerable: true,
   configurable: true
};
my_obj = Object.getOwnPropertyDescriptor(obj,'val');
console.log(my_obj);
```

#### ผลลัพธ์

```
Object { value: "1234",
writable: true,
enumerable: true,
configurable: true }
```

## hasOwnProperty

ใช้ตรวจสอบว่าในอ็อบเจกต์นั้นมี property อยู่หรือไม่



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



```
o = new Object();
o.prop = 'ABC1234';
my_obj = o.hasOwnProperty('prop');
console.log(my_obj);
```

ผลลัพธ์

true

# ตัวอย่างที่ 2

```
o = new Object();
o.prop;
my_obj = o.hasOwnProperty('prop');
console.log(my_obj);
```

ผลลัพธ์

false

## ls

ใช้ตรวจสอบค่าสองตัวว่าเหมือนกันไหม



```
var test = Object.is('ABCD','ABCD');
console.log(test);
```

ผลลัพธ์

#### true

#### isExtensible

ใช้ตรวจสอบว่าอ็อบเจกต์นั้นผ่านการใช้งานฟังก์ชัน preventExtension,seal,freeze หรือไม่

## ตัวอย่างที่ 1

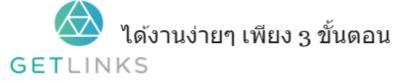
```
var obj = {ABCD:'ABCD'};
var test = Object.isExtensible(obj);
console.log(test);
```

ผลลัพธ์

#### true

# ตัวอย่างที่ 2

```
var obj = {ABCD:'ABCD'};
var frozen = Object.freeze(obj);
var test = Object.isExtensible(obj);
console.log(test);
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

#### ผลลัพธ์

#### false

#### isFrozen

ใช้ตรวจสอบว่าอ็อบเจกต์นั้นสามารถเพิ่มเติมหรือแก้ไขค่าได้

## ตัวอย่างที่ 1

```
var obj = {ABCD:'ABCD'};
var chk = Object.isFrozen(obj);
var test = Object.isExtensible(obj);
console.log(test);
```

#### ผลลัพธ์

#### true

# ตัวอย่างที่ 2

```
var obj = {ABCD:'ABCD'};
var lock = Object.freeze(obj);
var chk = Object.isFrozen(obj);
var test = Object.isExtensible(obj);
console.log(test);
```

#### ผลลัพธ์

## false



# isPrototypeOf

ใช้กำหนดอ็อบเจ็คให้สืบทอดมาจากออบเจ็คอื่น

# ตัวอย่างที่ 1

```
var err = new Error("my name is error");
var set = Object.getPrototypeOf(err);
var test = set === Error.prototype;
console.log(test);
```

ผลลัพธ์

true

- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

## isSealed

ใช้สำหรับตรวจสอบว่าผ่านการใช้งานฟังก์ชัน seal มาหรือไม่

# ตัวอย่างที่ 1

```
var obj = {};
var test =Object.isSealed(obj);
console.log(test);
```

ผลลัพธ์

false

# ตัวอย่างที่ 2

```
var obj = {};
Object.seal(obj);
var test =Object.isSealed(obj);
console.log(test);
```

ผลลัพธ์

true



## Keys

ใช้สำหรับดึงค่าคีย์ออกมาจากอ็อบเจกต์

## ตัวอย่างที่ 1

```
var obj = {a:'x',b:'y',c:'z'};
var test = Object.keys(obj);
console.log(test);
```

#### ผลลัพธ์

```
Array [ "a", "b", "c" ]
```

# preventExtensions

ใช้สำหรับป้องกันการเพิ่ม property เข้าไปในอ็อบเจกต์

#### ตัวอย่างที่ 1

```
var obj = {};
Object.preventExtensions(obj);
Object.defineProperties(obj,{
    "test":{
       a:'x',
       b:'y',
       c:'z'
    }
});
console.log(obj);
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

#### ผลลัพธ์

▶ TypeError: can't define property "test": Object is not extensible

#### Seal

ใช้ป้องกันการเพิ่มสมาชิกใหม่และไม่ให้มีการแก้ไขสมาชิกเดิม

## ตัวอย่างที่ 1

```
var obj = {};
Object.seal(obj);
Object.defineProperties(obj,{
    "test":{
        a:'x',
        b:'y',
        c:'z'
    },
    "test2":{
        D:'1',
        E:'2',
        F:'3'
    }
});
console.log(obj);
```

## ผลลัพธ์

TypeError: can't define property "test": Object is not extensible



# toString

แปลงค่าในอ็อบเจกต์ให้เป็นสตริงซึ่งจะคืนค่าเป็นชนิดของอ็อบเจ็คนั้นๆ

# ตัวอย่างที่ 1

```
var obj = {};
Object.defineProperties(obj,{
    "test":{
        a:'x',
        b:'y',
        c:'z'
      },
    "test2":{
        D:'1',
        E:'2',
        F:'3'
      }
});
var test = obj.toString();
console.log(test);
```

ผลลัพธ์

# [object Object]

- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

# **Promise**

เป็นออปเจคต์ที้ใช้งานร่วมกับการเขียนแบบ asynchronous โดยออกแบบมาให้รับค่าที่ยังไม่ เสร็จ แต่สัญญาว่าจะเสร็จในอนาคต

#### All

รับค่าจาก Promise object หลายๆ ค่า แล้วนำมาแสดงพร้อมกัน

## ตัวอย่างที่ 1

```
var p1 = Promise.resolve(3);
var p2 = 9999;
var p3 = new Promise((resolve, reject) => {
   setTimeout(resolve, 100, "xxxx");
});

Promise.all([p1, p2, p3]).then(values => {
   console.log(values);
});
```

#### ผลลัพธ์

```
Array [ 3, 9999, "xxxx" ]
```



## Catch

ใช้เก็บค่า reject ที่ promise ทำงานผิดพลาด

# ตัวอย่างที่ 1

```
var p1 = new Promise(function(resolve, reject) {
   throw 'my name is Error';
});
p1.catch(function(e) {
   console.log(e);
});
```

ผลลัพธ์

my name is Error

- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

#### Race

สร้าง promise อันใหม่ แต่ให้มีค่าเหมือนกับ promise อันเดิม

## ตัวอย่างที่ 1

```
var p1 = new Promise(function(resolve, reject) {
    setTimeout(resolve, 0, 'old value');
});
var p2 = new Promise(function(resolve, reject) { });
var p3 = new Promise(function(resolve, reject) { });
var rac = Promise.race([p1, p2, p3]);
rac.then(function(res) {
    console.log(res);
});
```

ผลลัพธ์

## old value

# Reject

สร้าง error สำหรับ Promise

# ตัวอย่างที่ 1

```
var p = Promise.reject('my error');
p.catch(function(res) {
    console.log(res);
});
```



#### ผลลัพธ์

## my error

#### Resolve

กำหนดการทำงานที่เสร็จสิ้น ให้ promise

## ตัวอย่างที่ 1

```
var p = Promise.resolve('it is success');
p.then(function(result) {
    console.log(result);
});
```

ผลลัพธ์

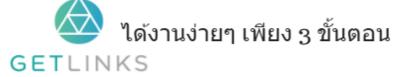
it is success

## then

ใช้สำหรับกำหนดงานให้ promise

# ตัวอย่างที่ 1

```
function timeout(t) {
    return new Promise(function(resolve, reject) {
        setTimeout(resolve, t);
    });
}
timeout(1000).then(() => {
    console.log('success');
});
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



#### ผลลัพธ์

#### success

# RegExp

ในการใช้งาน RegExp นั้น ผู้ใช้งานควรจะเข้าใจเกี่ยวกับการสร้าง **pattern** เสียก่อนจึงจะสามารถ ใช้งานได้อย่างถูกต้อง

#### Match

ตรวจสอบประโยคว่าตรงกับ regex ที่กำหนดไว้หรือไม่

# ตัวอย่างที่ 1

```
var tt = "I am programmer";
var res = tt.match('mer');
console.log(res);
```

## ผลลัพธ์

```
Array [ "mer" ]
```



# Replace

ใช้แปลงค่าที่ตรงกับ regex ที่กำหนด ให้เป็นค่าที่ต้องการ

# ตัวอย่างที่ 1

```
var tt = "I am programmer";
var res = tt.replace("am", "5555");
console.log(res);
```

ผลลัพธ์

I 5555 programmer

## Search

ใช้ค้นหาข้อความที่ตรงกับ regex หรือไม่ และจะคืนค่าเป็นตำแหน่ง

## ตัวอย่างที่ 1

```
var tt = "I am programmer";
var res = tt.search("p");
console.log(res);
```

ผลลัพธ์

5



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

# **Split**

แบ่งประโยคออกโดยใช้ regex

# ตัวอย่างที่ 1

ใช้ วรรค ในการแบ่งประโยค

```
var tt = "I am programmer";
var res = tt.split(" ");
console.log(res);
```

#### ผลลัพธ์

```
Array [ "I", "am", "programmer" ]
```

## **l**astIndex

ใช้ค้นหาค่าหรือคำที่ต้องการโดยจะคืนค่าเป็นตำแหน่ง

## ตัวอย่างที่ 1

```
var tt = "ABCD123";
var res = tt.lastIndexOf('D');
console.log(res);
```

ผลลัพธ์

3



#### Exec

ใช้ค้นหาค่าที่ตรงกับ regex ตัวแรก

## ตัวอย่างที่ 1

จากแพทเทอนนั้นหมายถึง X ตามด้วยอะไรก็ได้แต่มีปิดท้ายด้วย Z

```
var re = /(x).+?(z)/;
var res = re.exec('a b c defg hi12345 xyz');
console.log(res);
```

#### ผลลัพธ์

```
Array [ "xyz", "x", "z" ]
```

# **Flags**

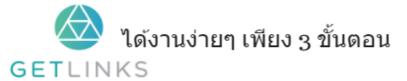
ใช้สำหรับกำหนดค่าในการค้นหา regex เช่นกำหนด i หมายถึงว่าไม่สนใจตัวเล็กหรือใหญ่ดัง ตัวอย่าง

## ตัวอย่างที่ 1

```
var tt = 'ABCDEFGH0123';
var test = tt.match(/d/i);
console.log(test);
```

## ผลลัพธ์

```
Array [ "D" ]
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

## Global

ใช้ในการกำหนดให้ค้นหาว่า flags เป็นชนิด global หรือไม่

# ตัวอย่างที่ 1

```
var reg = new RegExp('ABCDEFGH0123','g');
console.log(reg.global);
```

ผลลัพธ์

true

# ignoreCase

ใช้ในการกำหนดให้ค้นหาว่า flags ถูกกำหนดให้เป็นแบบ Non case sensitive หรือไม่

# ตัวอย่างที่ 1

```
var reg = new RegExp('ABCDEFGH0123','i');
console.log(reg.ignoreCase);
```

ผลลัพธ์

true



## Multiline

ใช้ในการกำหนดให้ค้นหาว่า flags ถูกกำหนดให้เป็นแบบ ค้นหาจากจุดเริ่มต้นของประโยค หรือไม่ (เหมือนกับการใช้ ^ ในregex)

## ตัวอย่างที่ 1

```
var reg = new RegExp('ABCDEFGH0123','m');
console.log(reg.multiline);
```

ผลลัพธ์

true

#### Source

คืนค่ารูปแบบ pattern ของ RegExp

# ตัวอย่างที่ 1

```
var reg = /I am pattern/i;
console.log(reg.source);
```

ผลลัพธ์

I am pattern



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

### Test

ใช้ค้นหา string หรือค่าที่กำหนดว่ามีหรือไม่ ถ้ามีจะคืนค่า true ถ้าไม่มีจะคืนค่า false

## ตัวอย่างที่ 1

```
var str = "hello world! I am programmer";
var fin = /hello/.test(str);
console.log(fin);
```

ผลลัพธ์

true

# toString

แปลงออบเจ็คให้เป็นสตริงธรรมดา

# ตัวอย่างที่ 1

```
var reg = new RegExp('ABCDEFGH0123','igm');
var test = reg.toString();
console.log(test);
```

ผลลัพธ์

/ABCDEFGH0123/gim



#### Unicode

ค้านหา flags ว่าถูกกำหนดให้เป็นแบบ Unicode หรือไม่

ตัวอย่างที่ 1

```
var regex = new RegExp('0123Abc hello word', 'u');
var test = regex.unicode;
console.log(test);
```

ผลลัพธ์

true

# Set

เป็นออปเจคต์ที่ใช้แทนโครงสร้างของข้อมูล ซึ่งสมาชิกภายในจะมีค่าที่ไม่ซ้ำกัน

## Add

เพิ่มข้อมูลงไปในอ็อบเจกต์ของ Set

ตัวอย่างที่ 1

```
var myvar = new Set();
myvar.add(5);
myvar.add(99);
myvar.add("hello");
console.log(myvar);
```

ผลลัพธ์

```
Set [ 5, 99, "hello" ]
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



### Clear

ลบสมาชิกของ Set ออกทั้งหมด

# ตัวอย่างที่ 1

```
var myvar = new Set();
myvar.add(5);
myvar.add(99);
myvar.add("hello");
myvar.clear();
console.log(myvar);
```

#### ผลลัพธ์

Set [ ]

#### Delete

ลบสมาชิกของ Set ออกตามที่กำหนด

## ตัวอย่างที่ 1

```
var myvar = new Set();
myvar.add(5);
myvar.add(99);
myvar.add("hello");
myvar.delete("hello");
console.log(myvar);
```

#### ผลลัพธ์

Set [ 5, 99 ]



#### **Entries**

แปลงอ็อบเจกต์ Set ให้กลายเป็นแบบ Itarator

## ตัวอย่างที่ 1

```
var myvar = new Set();
myvar.add(5);
myvar.add(99);
myvar.add("hello");
var ent = myvar.entries();
console.log(ent.next().value);
console.log(ent.next().value);
console.log(ent.next().value);
```

#### ผลลัพธ์

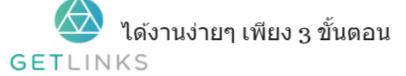
```
Array [ 5, 5 ]
Array [ 99, 99 ]
Array [ "hello", "hello" ]
```

## forEach

ใช้วนลูปเพื่อนำฟังก์ชันเข้าไปใช้งานสมาชิกของ อ็อบเจกต์ Set

## ตัวอย่างที่ 1

```
function test(var1, var2) {
   console.log("I am " + var1 + " = " + var2);
}
new Set(["apple", "man"]).forEach(test);
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



#### ผลลัพธ์

```
I am apple = apple
I am man = man
```

#### Has

ใช้ตรวจสอบว่ามีสมาชิกตามที่ กำหนดหรือไม่

# ตัวอย่างที่ 1

```
var myvar = new Set();
myvar.add(5);
myvar.add(99);
myvar.add("hello");
chk = myvar.has(99);

console.log(chk);
```

## ผลลัพธ์

#### true



## values

#### ดึงคีย์ออกมาจากอ็อบเจกต์ Set

# ตัวอย่างที่ 1

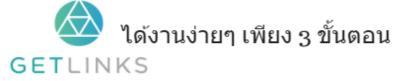
```
var myvar = new Set();
myvar.add(5);
myvar.add(99);
myvar.add("hello");
var ke = myvar.values();
console.log(ke.next().value);
console.log(ke.next().value);
```

ผลลัพธ์

5

99

hello



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

### size

ใช้นับจำนวนสมาชิกของ อ็อบเจกต์ Set

# ตัวอย่างที่ 1

```
var myvar = new Set();
myvar.add(5);
myvar.add(99);
myvar.add("hello");
chk = myvar.size;
console.log(chk);
```

ผลลัพธ์

3

# String

# **Anchor**

ใช้สำหรับสร้างลิงค์( anchor tag ใน HTML)

## ตัวอย่างที่ 1

```
var myvar = 'abcd12345';
var anc = myvar.anchor('i am link');
console.log(anc);
```



#### ผลลัพธ์

# <a name="i am link">abcd12345</a>

#### charAt

ใช้ดึงตัวอักษรจากประโยคโดยการกำหนดเป็นลำดับของตัวอักขระ

## ตัวอย่างที่ 1

```
var myvar = 'ABCDEFG';
var test = myvar.charAt(4);
console.log(test);
```

ผลลัพธ์

Е

### Concat

ใช้สำหรับเชื่อมคำหรือประโยค

# ตัวอย่างที่ 1

```
var myvar = 'ABCDEFG';
var test = myvar.concat('012','456','789');
console.log(test);
```

ผลลัพธ์

ABCDEFG012456789



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



#### endsWith

ใช้ตรวจสอบว่าประโยคลงท้ายด้วยคำหรือค่าที่กำหนดหรือไม่

# ตัวอย่างที่ 1

```
var myvar = 'ABCDEFG';
var test = myvar.endsWith('G');
console.log(test);
```

ผลลัพธ์

true

## ตัวอย่างที่ 2

```
var myvar = 'ABCDEFG';
var test = myvar.endsWith('G',0);
console.log(test);
```

ผลลัพธ์

false

## Includes

ใช้ตรวจสอบว่ามีคำหรือค่าในประโยค หรือไม่

## ตัวอย่างที่ 1

```
var myvar = 'ABC DEF 12345';
var test = myvar.includes('DEF');
console.log(test);
```



RUK-COM Hosting

#### ผลลัพธ์

#### true

# ตัวอย่างที่ 2

```
var myvar = 'ABC DEF 12345';
var test = myvar.includes('DEF',5);
console.log(test);
```

ผลลัพธ์

#### false

## indexOf

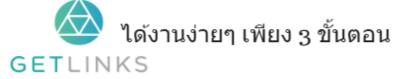
ใช้ดึงตำแหน่งของคำในประโยค

## ตัวอย่างที่ 1

```
var myvar = 'ABC DEF 12345';
var test = myvar.lastIndexOf('DEF');
console.log(test);
```

ผลลัพธ์

4



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

# Length

ใช้หาค่าความยาวของสตริง

ตัวอย่างที่ 1

```
var myvar = 'ABC DEF 12345';
var test = myvar.length;
console.log(test);
```

ผลลัพธ์

13

### Link

ใช้สำหรับสร้างลิงค์( anchor tag ใน HTML)

ตัวอย่างที่ 1

```
var tt = 'link';
var url = 'www.google.co.th';
var test = tt.link(url);
console.log(test);
```

ผลลัพธ์

<a href="www.google.co.th">link</a>



#### Match

ใช้จับคู่คำหรือข้อความด้วย regex

ตัวอย่างที่ 1

```
var tt = 'A B CDE FGH IJK123';
var pat = /c/i;
var test = tt.match(pat);
console.log(test);
```

ผลลัพธ์

```
Array [ "C" ]
```

## Normalize

แปลงอักขระ Unicode ให้อยู่ในรูปแบบปกติ

ตัวอย่างที่ 1

```
var tt = '\u0209\u0128\u0199';
tt.normalize('NFC');
console.log(tt);
```

ผลลัพธ์

ìĨƙ



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

## padEnd

สร้างสตริงให้อยู่ในรูปแบบที่กำหนดและสามารถกำหนดจำนวนตัวอักษรได้ แต่ว่าถ้าตัวอักษรมีไม่ครบ ตามที่กำหนด ก็จะวนลูปตัวอักษรใหม่จนครบ(สตริงที่กำหนดจะอยู่หลังสุด)

# ตัวอย่างที่ 1

```
var tt = 'ABCD';
var test = tt.padEnd(20,"0EFG");
console.log(test);
```

#### ผลลัพธ์

#### ABCD0EFG0EFG0EFG

# padStart

สร้างสตริงให้อยู่ในรูปแบบที่กำหนดและสามารถกำหนดจำนวนตัวอักษรได้ แต่ว่าถ้าตัวอักษรมีไม่ครบ ตามที่กำหนด ก็จะวนลูปตัวอักษรใหม่จนครบ(สตริงที่กำหนดจะอยู่หน้าสุด)

## ตัวอย่างที่ 1

```
var tt = 'ABCD';
var test = tt.padStart(20,"0123");
console.log(test);
```

#### ผลลัพธ์

0123012301230123ABCD



#### Raw

แปลงอักขระทุกอย่างเป็นสตริงและเลี่ยงอักขระพิเศษ

## ตัวอย่างที่ 1

```
var test = String.raw`Hi\n${2+3}!u000A`;
console.log(test);
```

ผลลัพธ์

Hi\n5!u000A

## repeat

ใช้สำหรับสร้างอักษรที่กำหนดแบบซ้ำๆ ไปโดยการกำหนดรอบ

## ตัวอย่างที่ 1

```
var tt = 'ABCD';
var test = tt.repeat(10);
console.log(test);
```

ผลลัพธ์

ABCDABCDABCDABCDABCDABCDABCDABCDABCD



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

# Replace

ใช้ regex ค้นหาและแทนที่

# ตัวอย่างที่ 1

```
var tt = 'I love you';
var test = tt.replace(/love/,'****');
console.log(test);
```

ผลลัพธ์

I \*\*\*\* you

## Search

ใช้ค้นหาคำที่กำหนด ถ้าเจอจะคืนค่าเป็นตำแหน่งของตัวอักษร ถ้าไม่เจอ คืนค่า -1

# ตัวอย่างที่ 1

```
var tt = 'I love you';
var test = tt.search('love');
console.log(test);
```

ผลลัพธ์

2



## Slice

ใช้ลบอักบระตามที่กำหนด โดยทำได้ด้วยการกำหนดลำดับของตัวอักษรที่จะลบ

# ตัวอย่างที่ 1

```
var tt = 'I love you';
var test = tt.slice(1,-1);
console.log(test);
```

ผลลัพธ์

# love yo

# **Split**

ใช้แยกอักขระที่กำหนดออกจากกัน และยังสามารถกำหนดจำนวนได้ด้วย

# ตัวอย่างที่ 1

```
var str = 'AB CD EFGHIJK LMN';
var test = str.split(' ', 2);
console.log(test);
```

# ผลลัพธ์

```
Array [ "AB", "CD" ]
```

- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

### startsWith

ตรวจสอบว่าประโยคขึ้นต้นด้วยอักขระหรือคำที่กำหนดหรือไม่

# ตัวอย่างที่ 1

```
var str = 'AB CD EFGHIJK LMN';
var test = str.startsWith('A');
console.log(test);
```

ผลลัพธ์

true

### Substr

ดึงตัวอักษรออกมาจากประโยค โดยกำหนดจุดเริ่มต้นและสิ้นสุดได้

# ตัวอย่างที่ 1

```
var str = 'ABCDEFGHIJKLMN1234567890';
var test = str.substr(1,3);
console.log(test);
```

ผลลัพธ์

**BCD** 



# substring

ดึงตัวอักษรออกมาจากประโยค โดยกำหนดจุดเริ่มต้นและสิ้นสุดได้

## ตัวอย่างที่ 1

```
var str = 'ABCDEFGHIJKLMN1234567890';
var test = str.substring(0,5);
console.log(test);
```

ผลลัพธ์

#### ABCDE

#### toLowerCase

ใช้สำหรับแปลงตัวพิมพ์ใหญ่ให้เป็นตัวพิมพ์เล็ก

## ตัวอย่างที่ 1

```
var str = 'ABCDEFGHIJKLMN1234567890';
var test = str.toLowerCase();
console.log(test);
```

ผลลัพธ์

abcdefghijklmn1234567890



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

# toString

ใช้สำหรับแปลงอ็อบเจกต์ให้ตัวอักษร

## ตัวอย่างที่ 1

```
var obj = new String('abcdefghijklmn1234567890');
var test = obj.toString();
console.log(test);
```

ผลลัพธ์

abcdefghijklmn1234567890

# toUpperCase

ใช้สำหรับแปลงตัวพิมพ์เล็กให้เป็นตัวพิมพ์ใหญ่

## ตัวอย่างที่ 1

```
var str = 'abcdefghijklmn1234567890';
var test = str.toUpperCase();
console.log(test);
```

ผลลัพธ์

ABCDEFGHIJKLMN1234567890



#### trim

ใช้ลบช่องว่างออกจากด้านหน้าและหลังของคำหรือประโยค

# ตัวอย่างที่ 1

ผลลัพธ์

A B C D E FGHI

#### valueOf

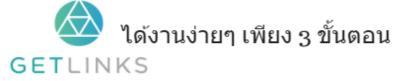
ใช้สำหรับดึงค่าของอ็อบเจกต์

## ตัวอย่างที่ 1

```
var str = new String('my name is boy');
var test = str.valueOf();
console.log(test);
```

ผลลัพธ์

my name is boy



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

# Proxy

เป็นออปเจคต์ที่ใช้ห่อหุ้มอีกออปเจคต์หนึ่ง เพื่อเรียกใช้งานฟังก์ชันหรือพรอพเพอร์ตี้ของออปเจคต์ที่ถูก ห่อหุ้ม โดยไม่เข้าไปเปลี่ยนแปลงค่าในออปเจคต์ที่ห่อหุ้มนั้น

#### Construct

ใช้สำหรับสร้างการทำงานให้กับ constructor

## ตัวอย่างที่ 1

```
var p = new Proxy(function() {}, {
   construct: function(target, argL, newTarget) {
     console.log("call=> " + argL);
     return { value: argL[0] * 100 };
   }
});
console.log(new p(1).value);
```

## ผลลัพธ์

```
call=> 1
```

100



# defineProperty

ใช้กำหนดค่า property ให้ method

#### ตัวอย่างที่ 1

```
var p = new Proxy({}, {
   defineProperty(target, prop, des) {
      console.log(des);
      return Reflect.defineProperty(target, prop, des);
   }
});
Object.defineProperty(p, "name", { value: "proxy"});
```

ผลลัพธ์

```
Object { value: "proxy" }
```

# deleteProperty

ใช้ลบค่า property ให้ methodตัวอย่างที่ 1

- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

```
var myvar = { webName: "ABCD"};
var p = new Proxy(myvar, {
   deleteProperty: function(target, prop) {
      console.log(prop);
      Reflect.deleteProperty(target, prop);
      return true;
   }
});
console.log(myvar);
delete p.webName;
console.log(myvar);
```

#### ผลลัพธ์

```
Object { webName: "ABCD" }
webName
Object { }
```

### Get

เข้าถึงค่า property



## ตัวอย่างที่ 1

```
var p = new Proxy({}, {
   get: function(target, prop, receiver) {
      console.log("call: " + prop);
   }
});
console.log(p.a);
```

ผลลัพธ์

call: a

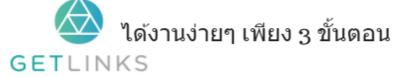
undefined

# **Apply**

ใช้สำหรับสร้างการทำงานให้กับฟังก์ชัน

# ตัวอย่างที่ 1

```
var p = new Proxy(function() {}, {
    apply: function(tar, arg, argL) {
        console.log(argL);
    }
});
p(1, 2, 3);
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



### ผลลัพธ์

# Array [ 1, 2, 3 ]

# getOwnPropertyDescriptor

เข้าถึง index ของ property

## ตัวอย่างที่ 1

```
var p = new Proxy({ a: 100 }, {
    getOwnPropertyDescriptor: function(target, prop) {
        console.log("call: " + prop);
        return {value: 123456789, configurable: true, enumerable: true};
    }
});

console.log(Object.getOwnPropertyDescriptor(p,'a').value);
```

## ผลลัพธ์

call: a

123456789

# getPrototypeOf

ใช้กำหนดให้สืบทอดคุณสมบัติจากออบเจ็คอื่น



## ตัวอย่างที่ 1

```
var obj = {};
var p = new Proxy(obj, {
    getPrototypeOf(target) {
        return Array.prototype;
    }
});
var test = Object.getPrototypeOf(p) === Array.prototype;
console.log(test);
```

ผลลัพธ์

true

#### Has

ใช้ตรวจสอบค่า ถ้ามีจะคืนค่า true ถ้าไม่มีจะคืนค่า false

# ตัวอย่างที่ 1

```
var p = new Proxy({}, {
   has: function(target, prop) {
      console.log("=> " + prop);
      return true;
   }
});
console.log("a" in p);
```

- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



#### ผลลัพธ์

=> a

true

## isExtensible

ใช้ตรวจสอบว่าอ็อบเจกต์นั้นผ่านการใช้งานจากฟังก์ชันอื่นมาหรือไม่ ถ้ายังจะคืนค่า true แต่ผ่าน การใช้งานจะคืนค่า false

# ตัวอย่างที่ 1

```
var p = new Proxy({}, {
   isExtensible: function(target) {
      console.log("called");
      return true;
   }
});
console.log(Object.isExtensible(p));
```

ผลลัพธ์

called

true



# ownKeys

เข้าถึงค่าของ method

## ตัวอย่างที่ 1

```
var p = new Proxy({}, {
  ownKeys: function(target) {
    console.log("called");
    return ["a", "b", "c"];
  }
});

console.log(Object.getOwnPropertyNames(p));
```

## ผลลัพธ์

## called

```
Array [ "a", "b", "c" ]
```



- 1. คลิกที่รูปนี้ เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท



# preventExtensions

ใช้สำหรับป้องกันการเพิ่ม property เข้าไปในเมตอด

# ตัวอย่างที่ 1

```
var p = new Proxy({}, {
   preventExtensions: function(target) {
      console.log("called");
      Object.preventExtensions(target);
      return true;
   }
});
console.log(Object.preventExtensions(p));
```

### ผลลัพธ์

called

Object { }



#### Set

#### เซทค่าให้กับ method

# ตัวอย่างที่ 1

```
var p = new Proxy({}, {
    set: function(target, prop, value, receiver) {
        console.log("called: " + prop + " = " + value);
        return true;
    }
});
p.a = 9999;
```

ผลลัพธ์

called: a = 9999



- 1. <mark>คลิกที่รูปนี้</mark> เพื่อสร้างโปรไฟล์กับ GetLinks
- 2. รอรับข้อเสนองานหลากหลายตำแหน่ง
- 3. รับโบนัส **10,000** บาท

# setPrototypeOf

เซทให้ออบเจ็คสืบทอดคุณสมบัติจากออบเจ็คอื่น ถ้าสำเร็จจะคืนค่า true ไม่สำเร็จจะคืนค่า false

# ตัวอย่างที่ 1

```
var refalse = {
    setPrototypeOf(target, newProto) {
        return false;
    }
};
var newProto = {}, target = {};
var p1 = new Proxy(target, refalse);
var test = Reflect.setPrototypeOf(p1, newProto);
console.log(test);
```

ผลลัพธ์

false

