

# Javascript & TypeScript

Sebuah Pengenalan OLEH:

RAZMAN YUSOF BIN MOHD SARIT

```
const a=12;
const b = 22;
const c = a+b;
//c==34?
```

# TENTANG SAYA

RAZMAN YUSOF BIN MOHD SARIT

Google: Razman Sarit

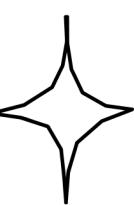
20 tahun pengalaman bidang pembangunan perisian komputer (software development)

B. Eng Computer Interactive Systems

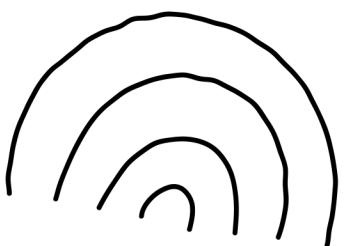
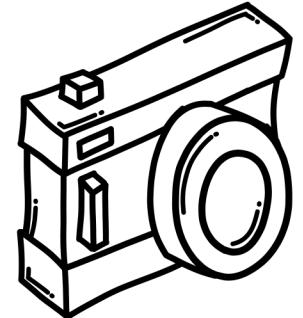
University Of Birmingham (UK)

Postgraduate Dip Management, University Of Wolverhampton (UK)

Pengkhususan bidang JavaScript/TypeScript



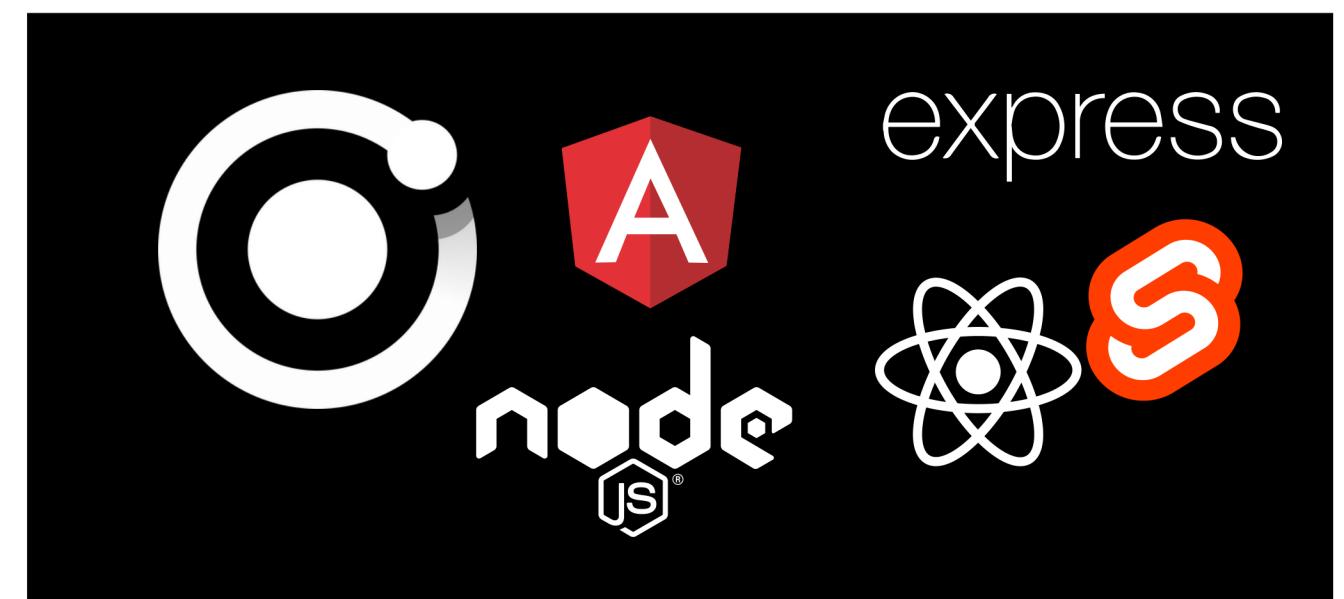
Gambar  
saya



@razmans



razmans.medium.com





# Speaker di IONICONF 2022

The image is a screenshot of a mobile application for the Ioniconf 2022 conference. At the top left, it says "Ioniconf 2022". At the top right, there are icons for a clock and a share button. In the top center, the "ioniconf" logo is displayed, featuring the year "2022" in blue and green, followed by "ioniconf" in white. Below the logo, the text "UP NEXT:" is visible. The main content area features a large, bold title: "Ionic: Easiest way to get into Angular/React". To the left of the title, there is a small video thumbnail showing a man with a beard, identified as "Mike | @mhartington". Below the title, the speaker's name "Razman Sarit" is listed, along with the text "Principal Trainer @ Atas Awan". To the right of the title, there is a large portrait photo of a bald man with a goatee, identified as "Razman Sarit". The background of the app has a dark theme with a grid pattern.

Ioniconf 2022

2022  
ioniconf

UP NEXT:

**Ionic: Easiest way to get into Angular/React**

Razman Sarit  
Principal Trainer @ Atas Awan

Mike | @mhartington

Razman Sarit



Platform Solutions Resources

Company Pricing

Sign in

Get started →



**Alan Montgomery**

Podcast host of [Mobile DevCast](#).  
Creator of [Ionic React Hub](#).



**Dayana Jabif**

Full stack developer and JS  
lover. Running [Ionic Themes](#).



**Robin Genz**

Capacitor expert and [open](#)  
source contributor.



**Aaron Saunders**

Author, community builder. [CEO](#)  
of [Clearly Innovative](#).



**Ezéchiel Agbla**

Software Engineer, speaker,  
mentor. CEO of [Wetillix LLC](#).



**Dansteve Adekanbi**

Consultant, full stack developer,  
educator, and speaker.



**Aaron Czichon**

JS Developer, Ionic Expert,  
Conference Speaker and Author



**Sani Yusuf**

Founder, Speaker, Author,  
DevRel.



**Agustin Haller**

Engineer, crafting code for  
[@ionicthemes](#) & [@ng\\_templates](#)



**Ricardo Jaime Gil  
Simões**

Tech mingler, Fullstack &  
DevOps Jack-of-all-trades



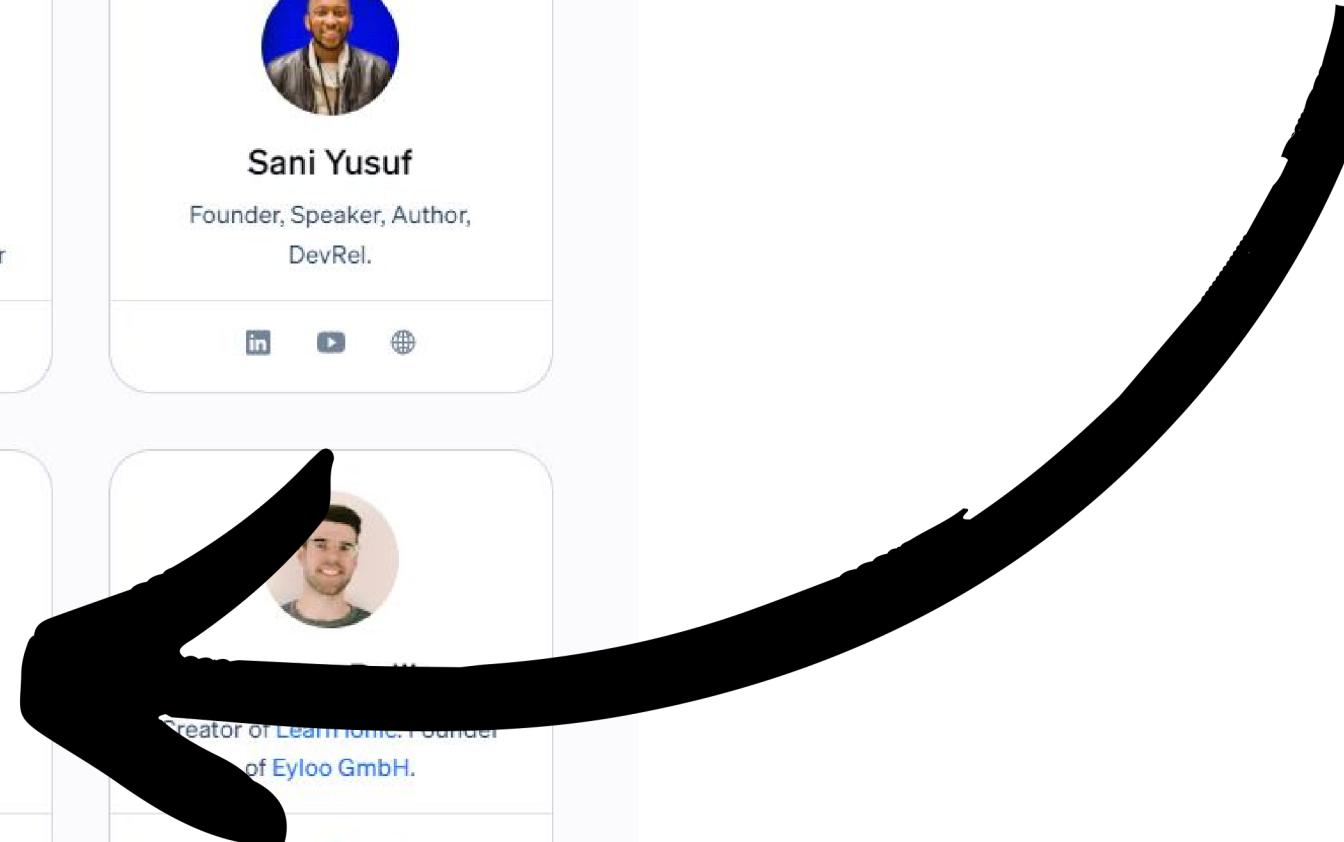
**Razman Sarit**

Coder, trainer, multi-award-  
winning mobile app developer



Creator of [LearnHome](#). Founder  
of [Eyloo GmbH](#).

Tersenarai sebagai  
Ionic Developer Expert

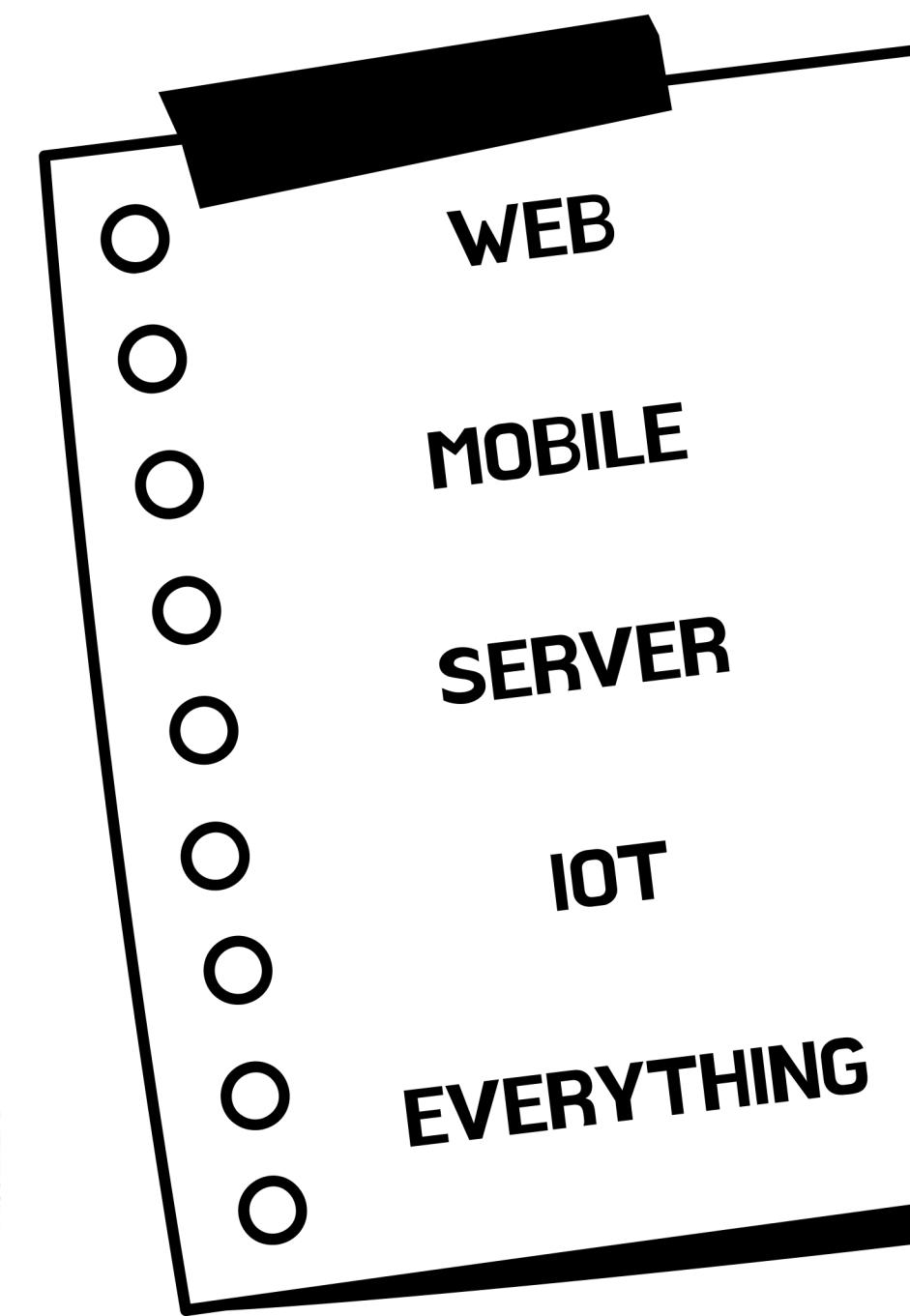


# Apa itu Javascript?

Bahasa pengaturcaraan digunakan secara meluas untuk pembangunan aplikasi web pada mulanya, tetapi kini digunakan di mana-mana sahaja. Bahasa yang serba guna

**Dibangunkan oleh Brendan Eich pada 1995 untuk meningkatkan interaktiviti laman web**

**Berkembang pesat dari 1995, kini mempunyai ekosistem menyeluruh dari pembangun web, aplikasi, server dan juga peranti IoT**



# Atwood's Law

**"Any application that can be  
written in JavaScript, will  
eventually be written in  
JavaScript."**

**JEFF ATWOOD**

FOUNDER OF STACK OVERFLOW

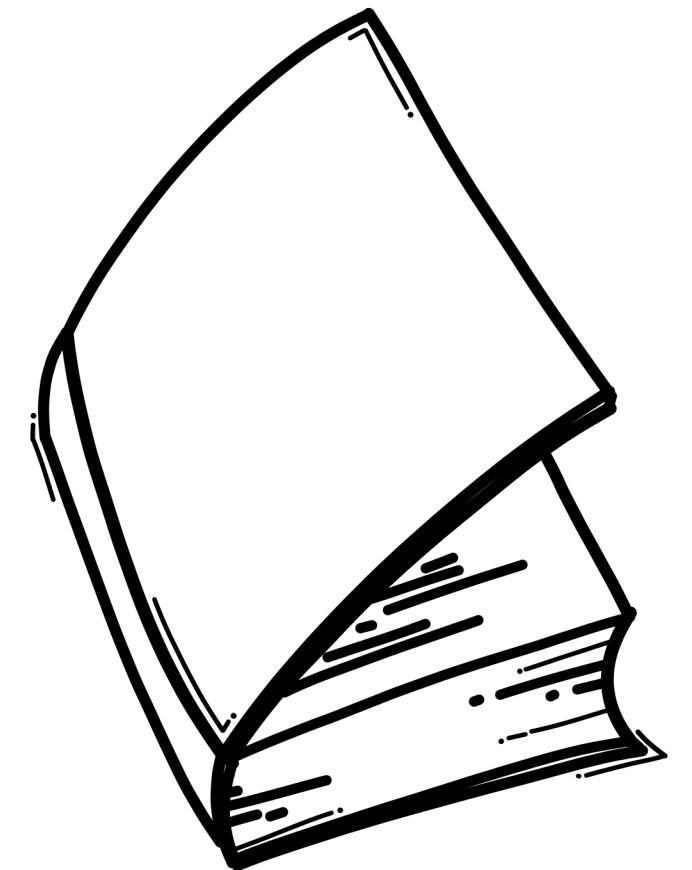


# **DISCUSSION**

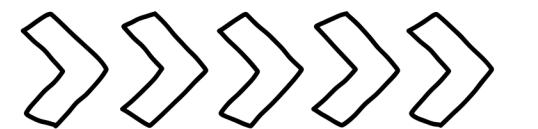
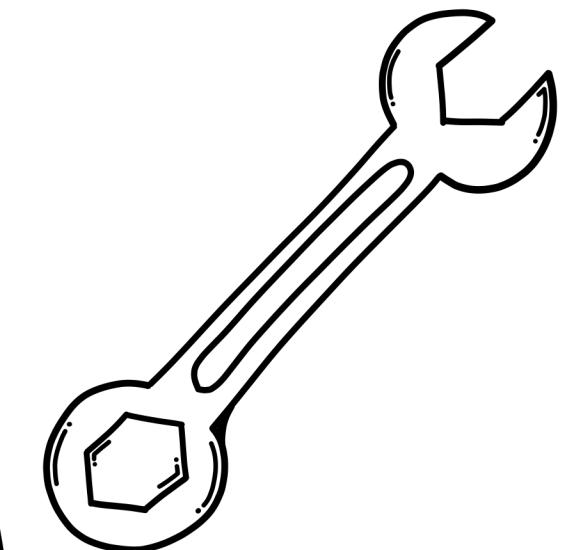
## **BARANG APA YANG ADA JAVASCRIPT?**

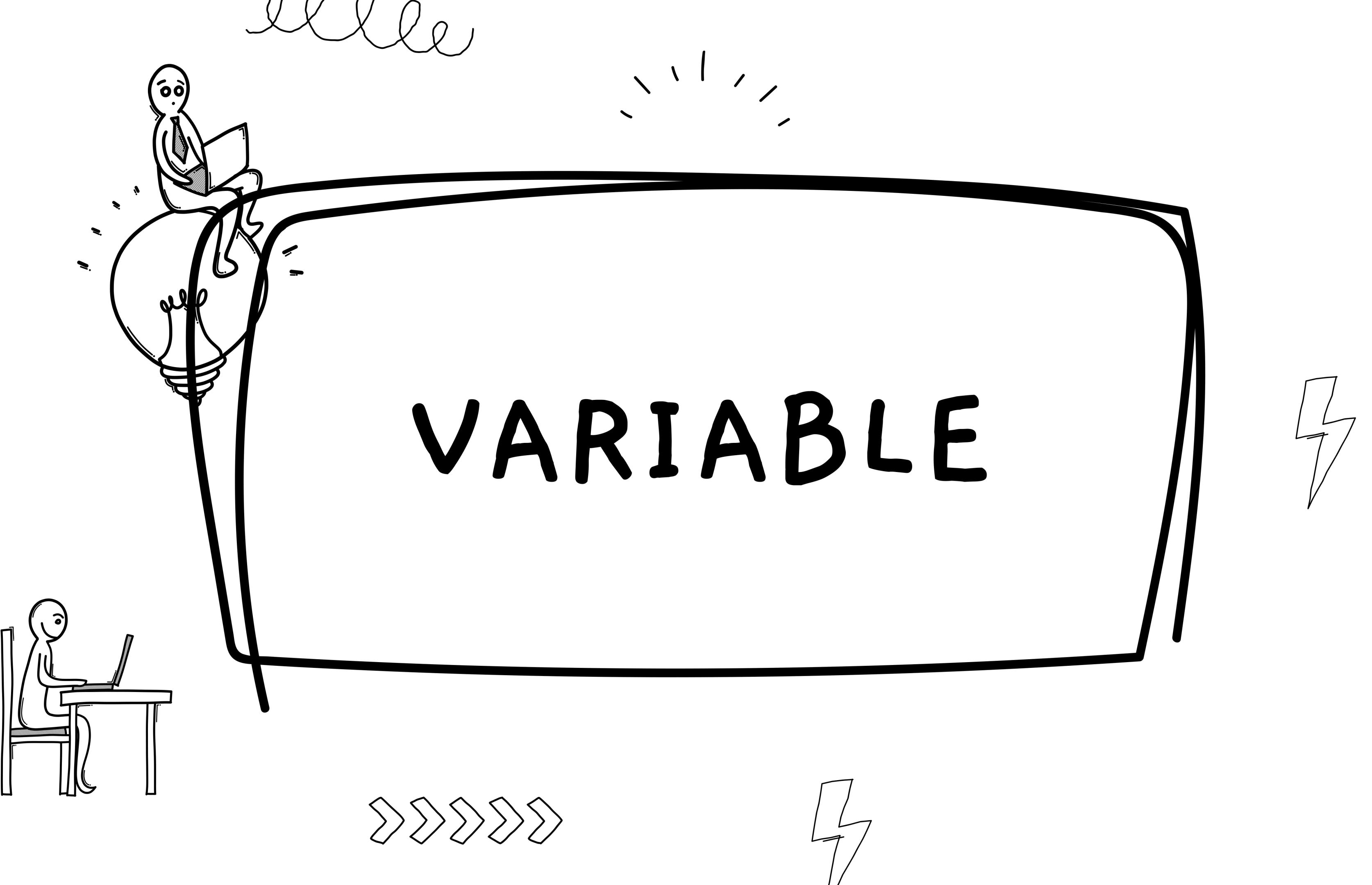
# Benda penting nak ingat sebelum kita mula

- 1. Cara nak comment, GUNA DOUBLE SLASH //**
- 2. Penggunaan Variable, secara amnya tidak perlu huruf besar, pasal takut terguna 'reserve' word**
- 3. Belajar dari kesilapan, JANGAN PUTUS ASA**
- 4. Jangan malu bertanya. NO SUCH THING AS STUPID QUESTIONS**



# LESSON 1





# Apa itu Variable?

Suatu tempat untuk menyimpan data  
Anggap macam kotak memori untuk simpan nilai.

Boleh guna keyword seperti: "var", "let", atau "const"

**var**

```
var nama = "John";  
var umur = 25;
```

Nilai boleh ubah  
dari mana-mana

Contoh: Mak ubah *nama*  
jadi Abu, maka:  
*nama* = Abu;

**let**

```
let bilangan = 10;  
let harga = 5.99;
```

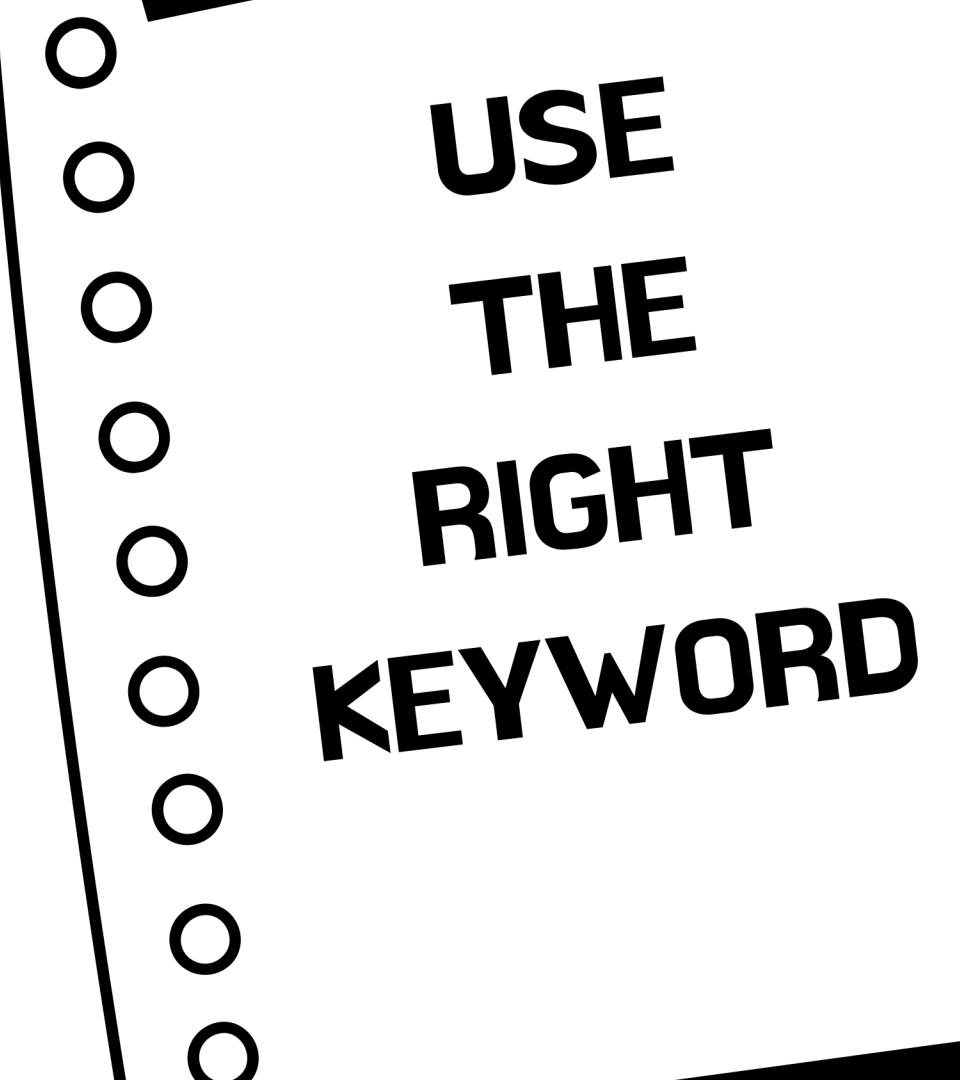
Nilai boleh ubah  
dari tempat dia  
dimulakan sahaja

Contoh: Mak ubah  
*bilangan* jadi 100,  
maka:  
*bilangan* = 100;

**const**

```
const PI = 3.14;  
const URL = "https://www.example.com";
```

Nilai tak boleh ubah.  
Constant



# Cara nak baca nilai

**var**

```
var nama = "John";  
var umur = 25;
```

**let**

```
let bilangan = 10;  
let harga = 5.99;
```

**const**

```
const PI = 3.14;  
const URL = "https://www.example.com";
```

Macam Algebra. Digunakan untuk mewakili nilai. Jadi dia boleh digantikan dengan keyword sahaja.

Contoh:

**var a=10;**

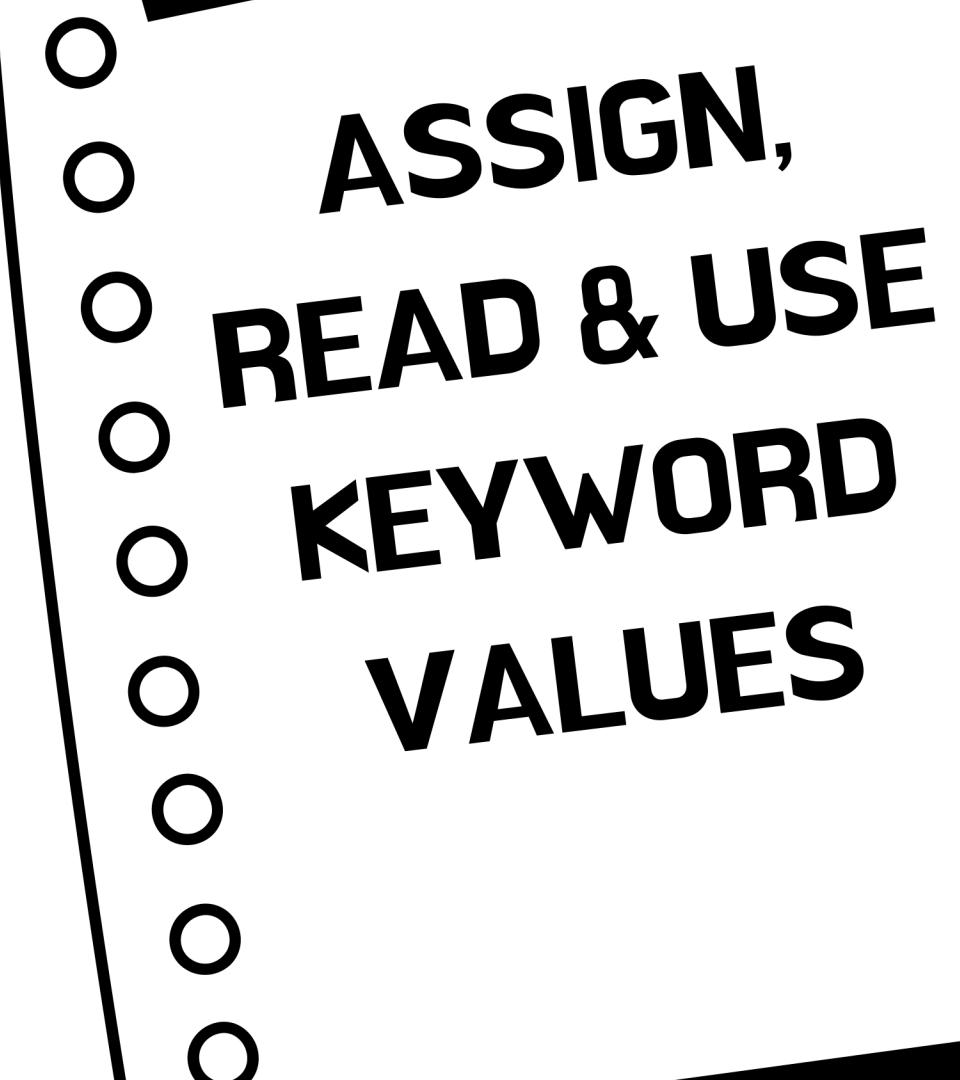
**var b=22;**

**var c=a** APA NILAI c?

**var d="Abdul";**

**var e="Wahub";**

**var f=e;** APA NILAI f?



# Cara nak delete

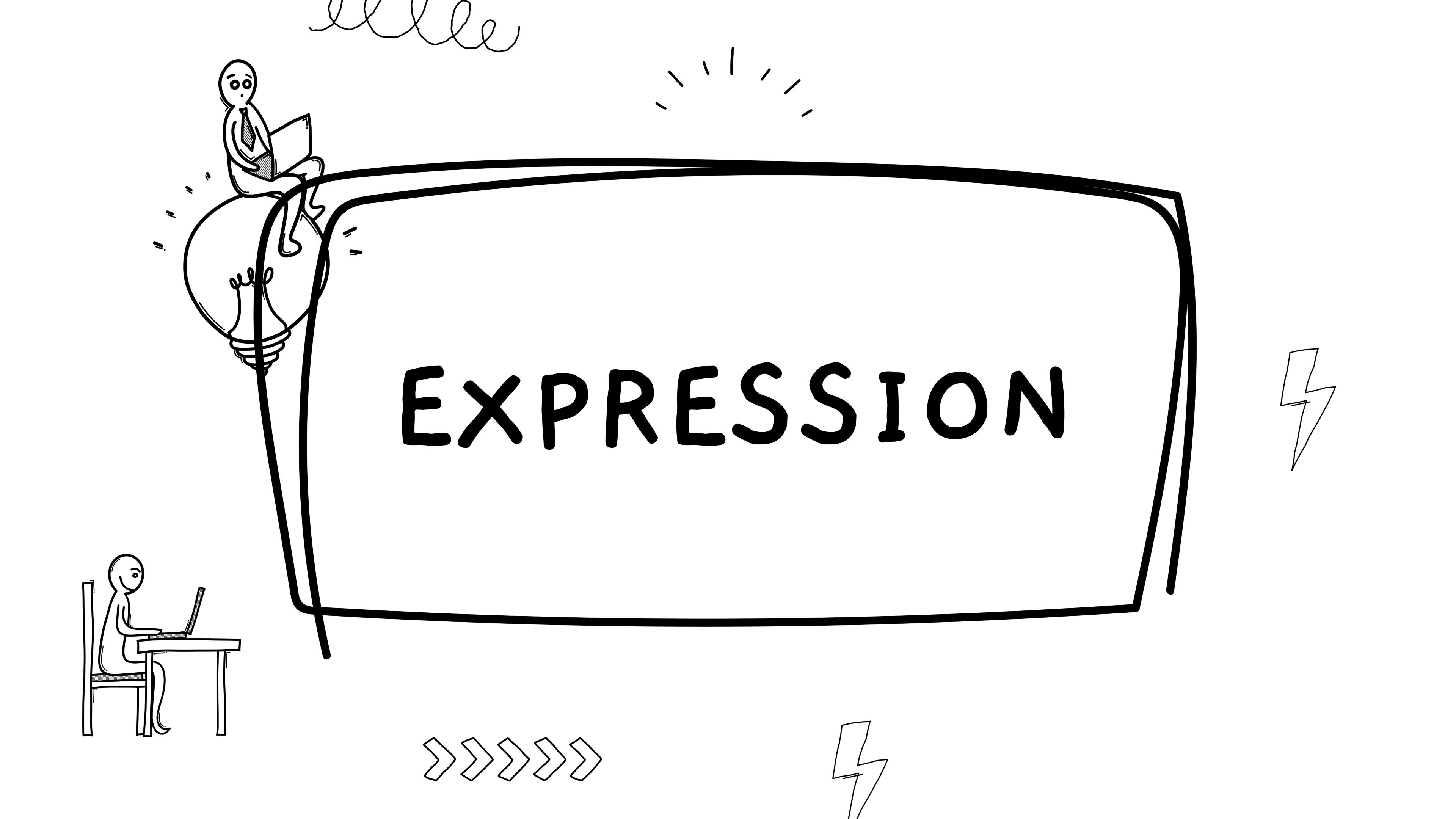
```
var objek = {  
    nama: "John",  
    umur: 25  
};  
  
console.log(objek.nama); // Output: John  
  
delete objek.nama;  
  
console.log(objek.nama); // Output: undefined
```

Gunakan delete. Ini akan padam habis

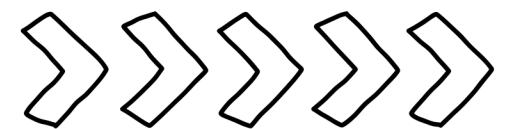
```
var nama = "John";  
console.log(nama); // Output: John  
  
nama = null;  
console.log(nama); // Output: null
```

Tukar nilai jadi null untuk hapuskan nilai sahaja  
(Cara paling selamat)

DELETING  
KEYWORD  
OR KEYWORD  
VALUES



**EXPRESSION**



# Apa expression dalam JS?

Merujuk kepada kombinasi operasi, nilai, pembolehubah, dan pemanggilan fungsi yang menghasilkan nilai. Expression dalam JavaScript dievaluasi untuk menghasilkan hasil akhir.

Boleh juga digunakan dalam operasi Matematik:

```
var a=10;      APA NILAI c?
```

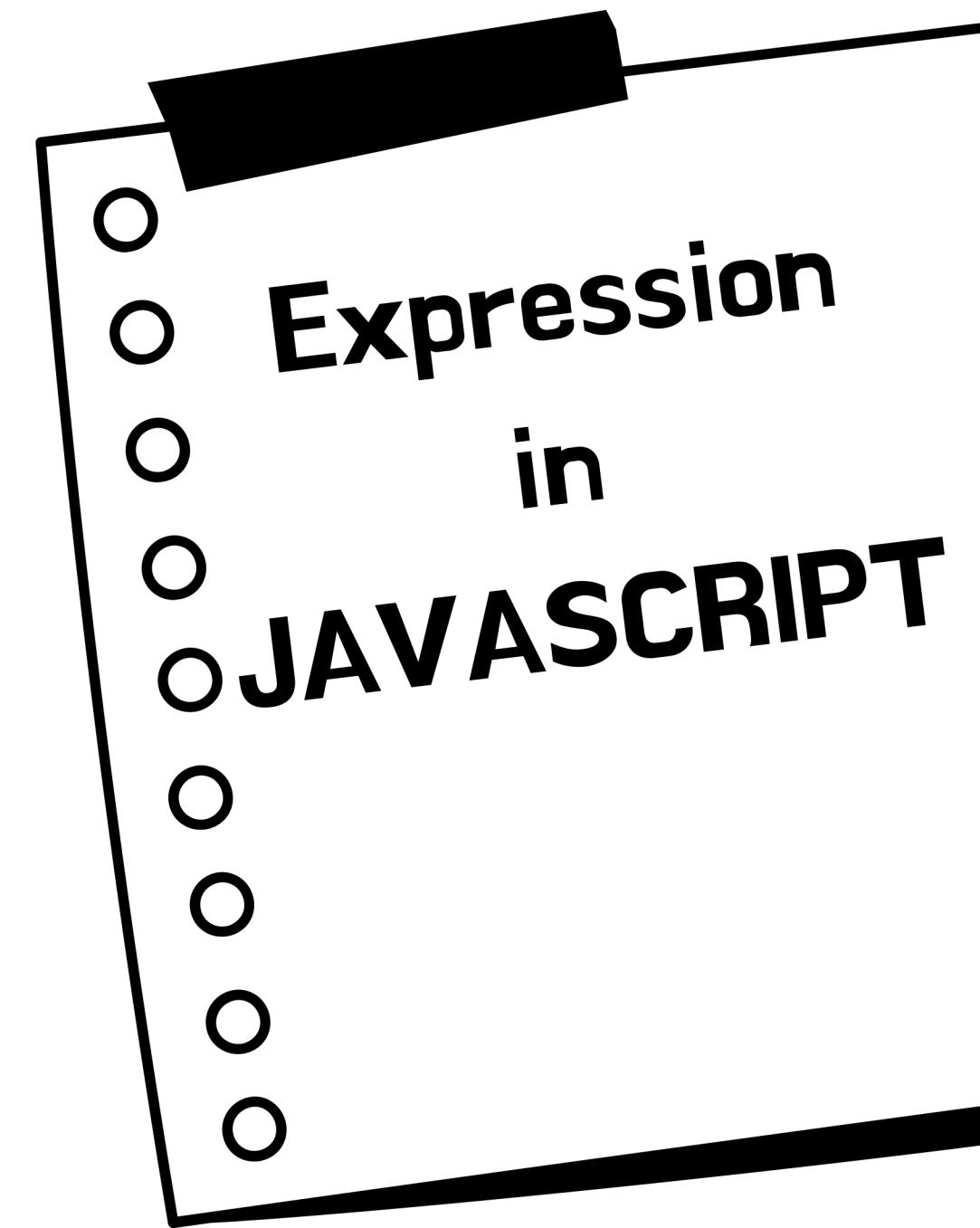
```
var b=22;
```

```
var c=a + b; // Pakai keyword sebagai wakil
```

```
var d="Abdul";
```

```
var e="Wahub";  APA NILAI f?
```

```
var f=d + e; // Pakai keyword sebagai wakil
```



# Expression panggilan fungsi

Boleh digunakan untuk panggilan fungsi

Ini adalah fungsi (function) yang diberi nama 'tambah':

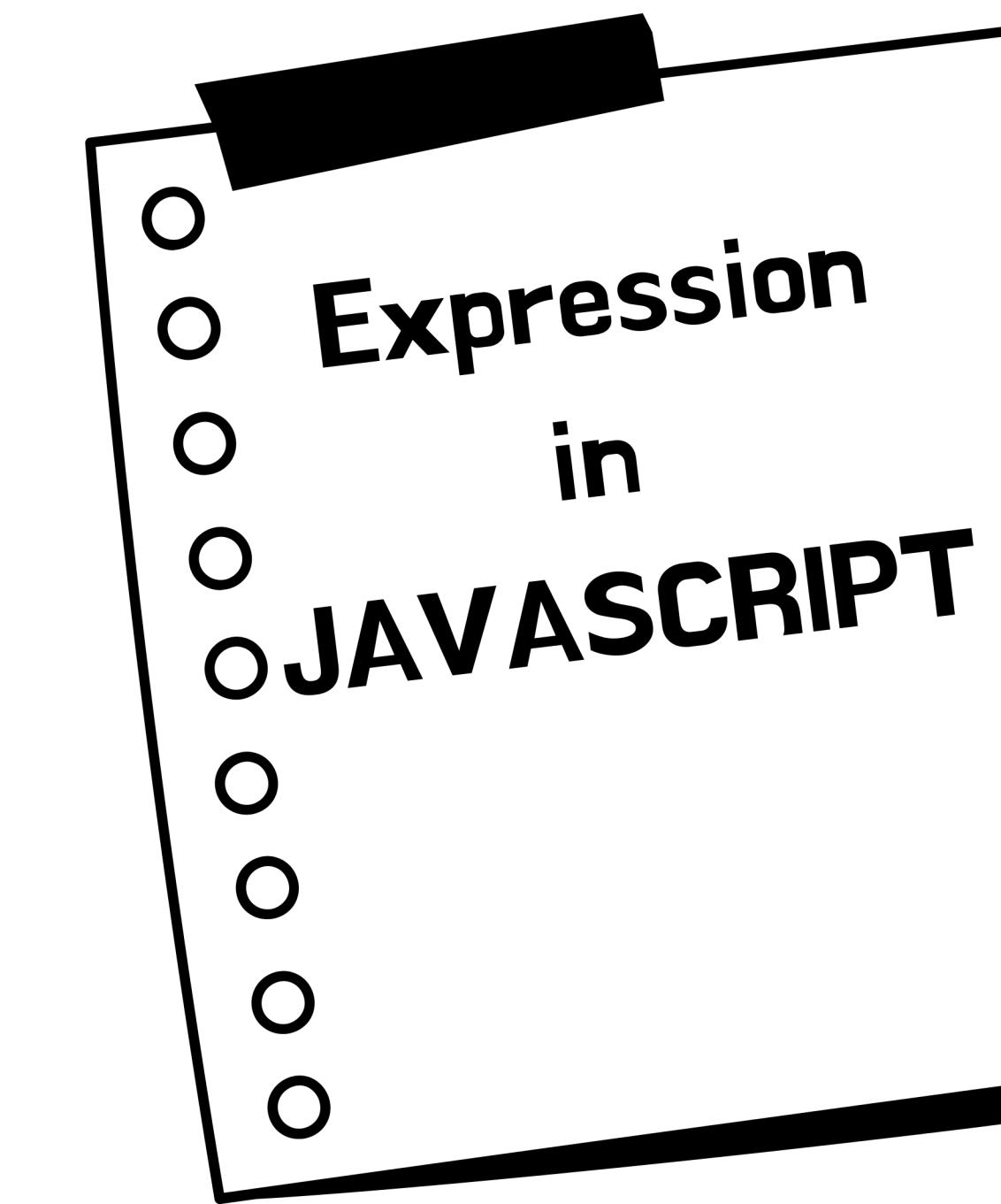
```
function tambah(a, b) {  
    return a + b;  
}
```

- Fungsi tambah mengambil 2 nilai
- Kedua-dua nilai akan ditambah dan akan di kembalikan kepada pemanggil

Satgi saya cerita lebih lanjut pasal fungsi (function)

```
var hasilTambah = tambah(2, 3);  
console.log(hasilTambah); // Output: 5
```

Cara kita panggil function tersebut



# Expression perbandingan comparison

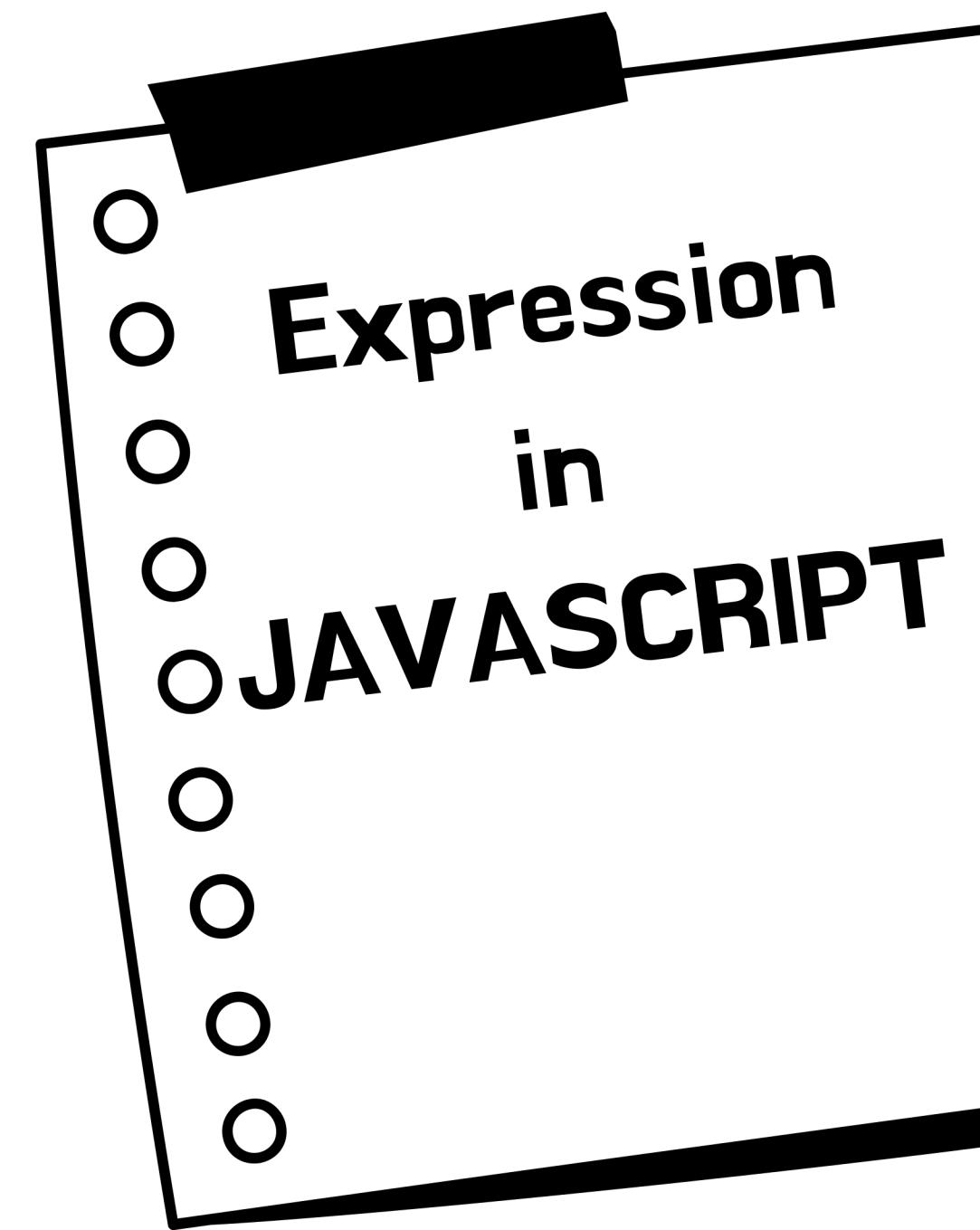
Ekspresi perbandingan yang akan menghasilkan nilai boolean

Boolean ialah nilai antara dua sahaja, true or false

Contoh:

```
var x = 5;  
var y = 10;  
var hasilPerbandingan = x > y;  
console.log(hasilPerbandingan); // Output: false
```

Satgi saya  
cerita lebih  
lanjut

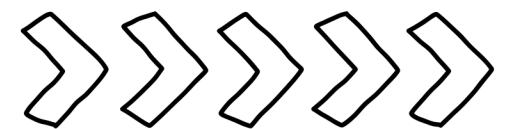
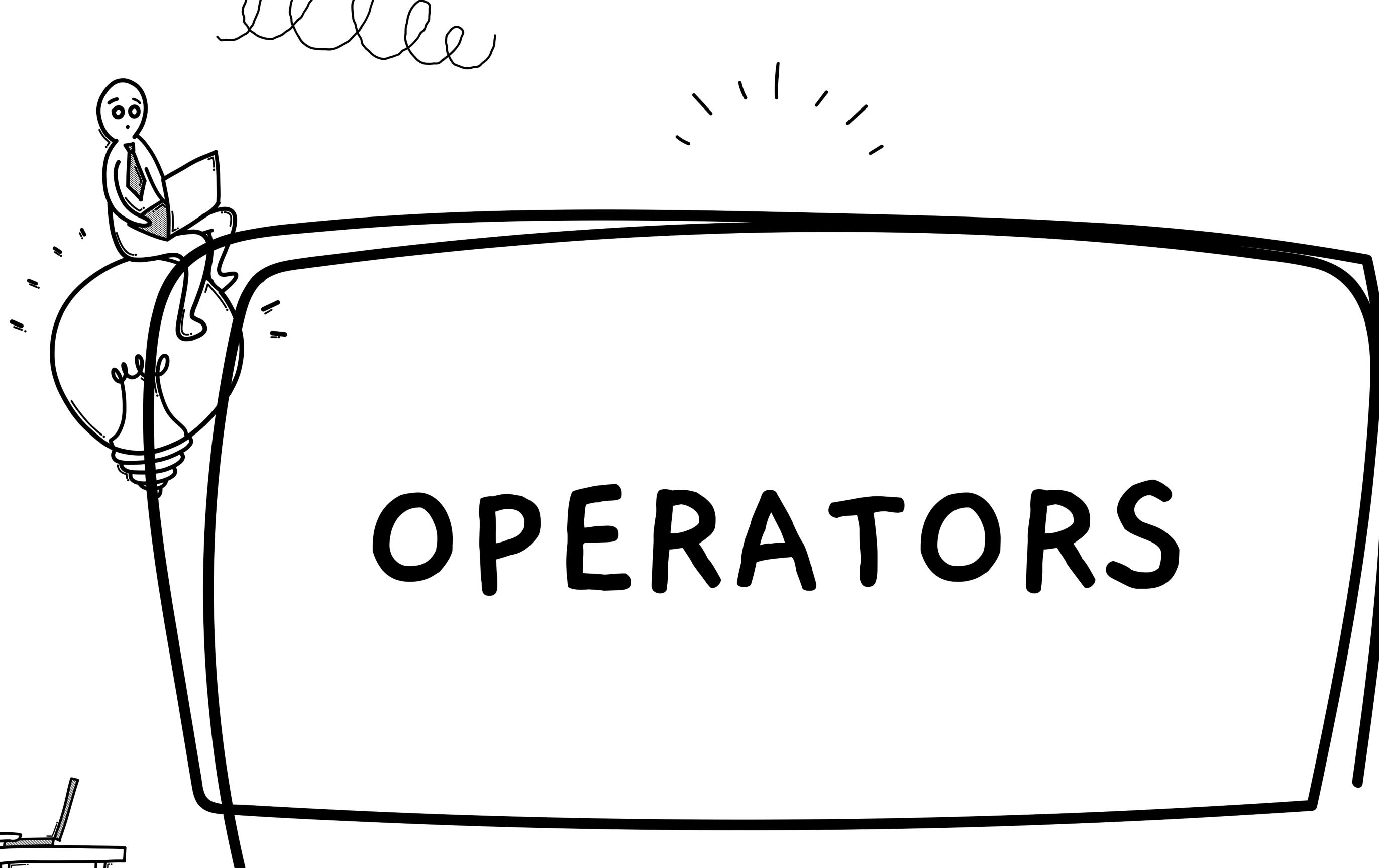


# KEPENTINGAN EXPRESSION DALAM JAVASCRIPT

**Expression adalah asas dalam JavaScript untuk melakukan operasi, pengiraan, dan penentuan keputusan.**

**Anda boleh menggunakan expression untuk menghasilkan nilai, menetapkan nilai kepada pembolehubah, atau menjalankan kod tertentu berdasarkan keadaan yang dinilai oleh ekspresi.**

# OPERATORS

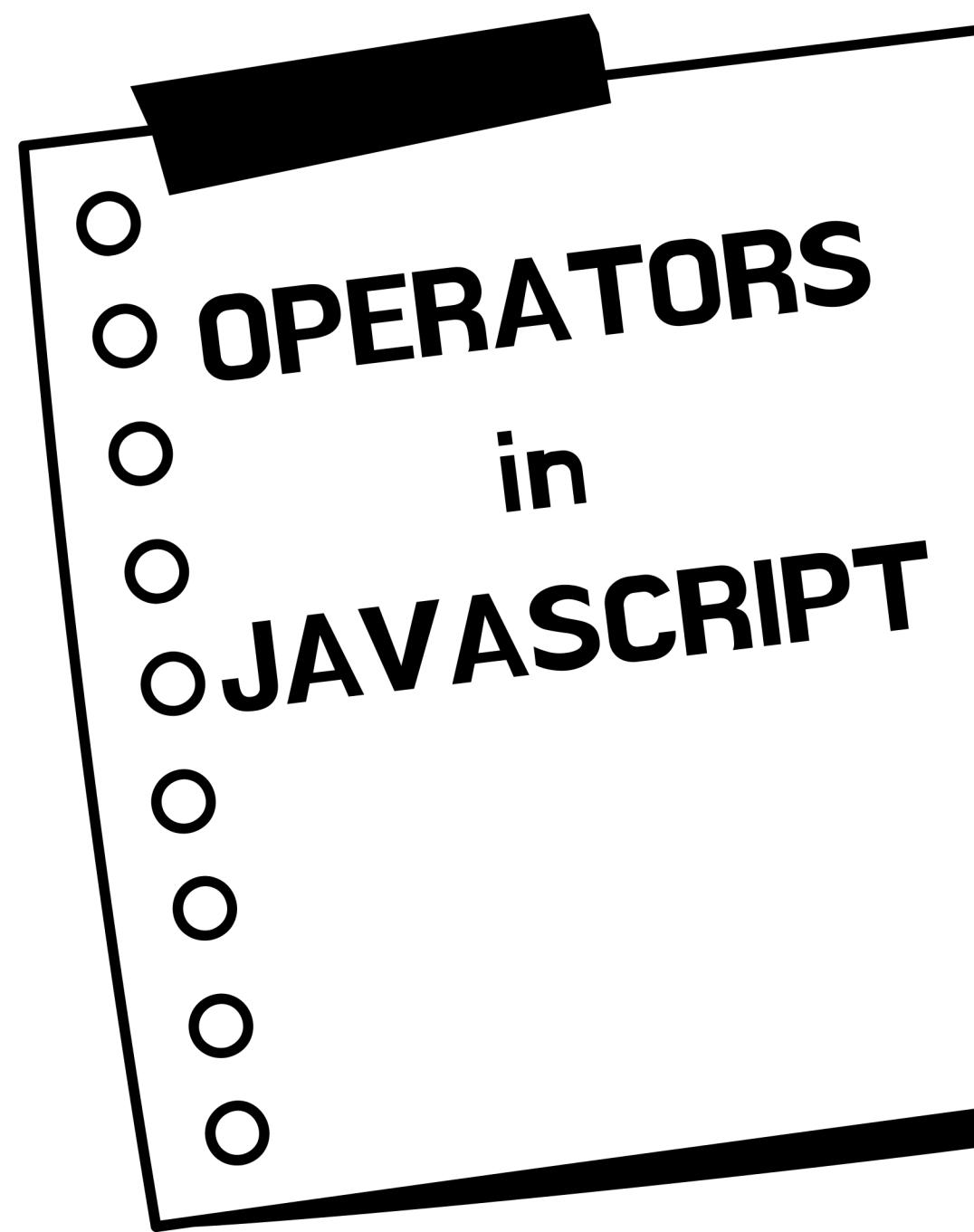


# Apa operators dalam JS?

Simbol atau keyword yang digunakan untuk melakukan operasi pada nilai

Digunakan untuk menggabungkan, memanipulasi, dan membandingkan nilai dalam kod JavaScript.

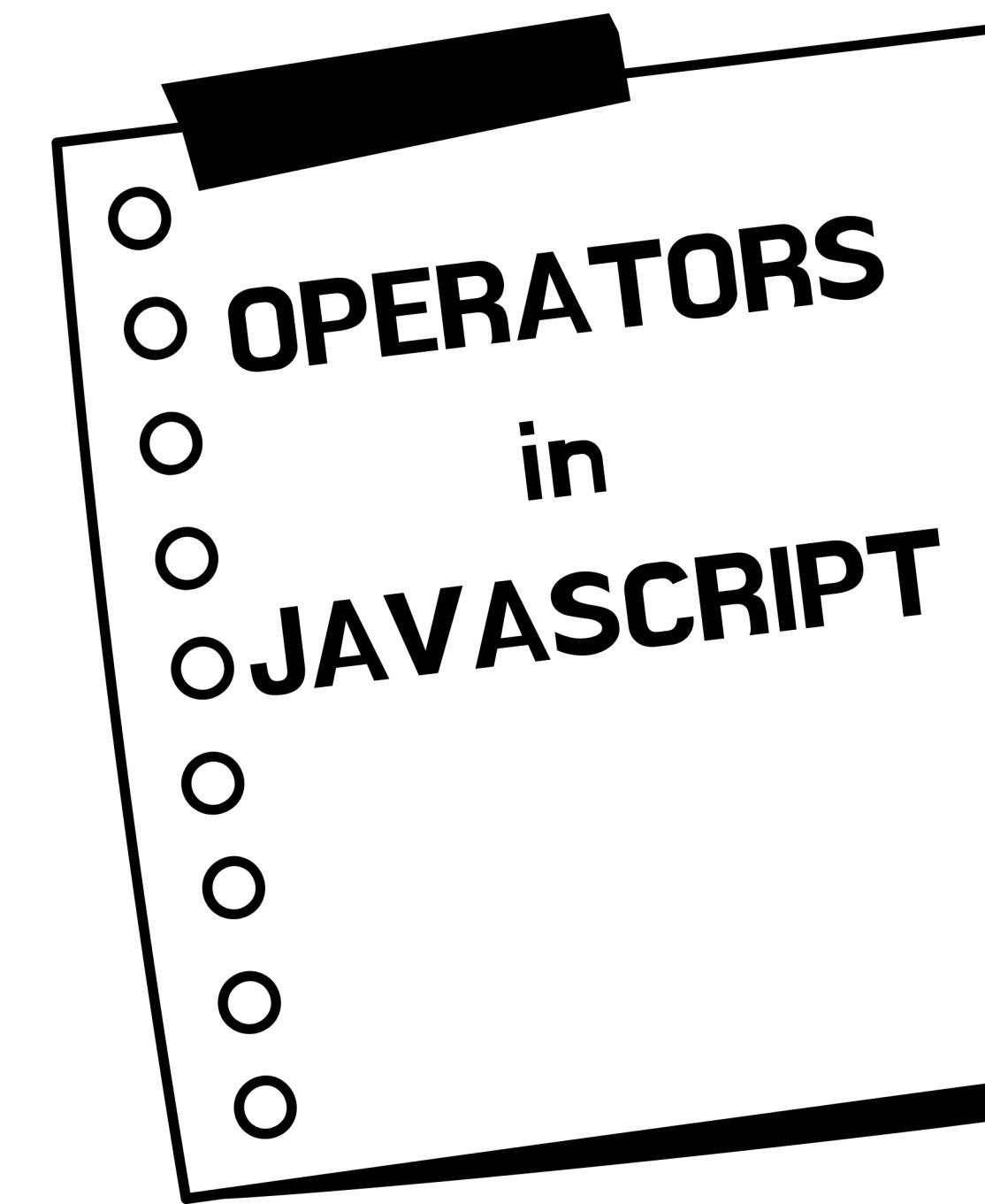
Terdapat pelbagai jenis operator yang boleh digunakan dalam JavaScript.



# Operator Aritmetik

- + Addition
- - Subtraction
- \* Multiplication
- / Division
- % Modulo
- ++ Increment
- -- Decrement

Lebih  
kurang  
macam  
Excel

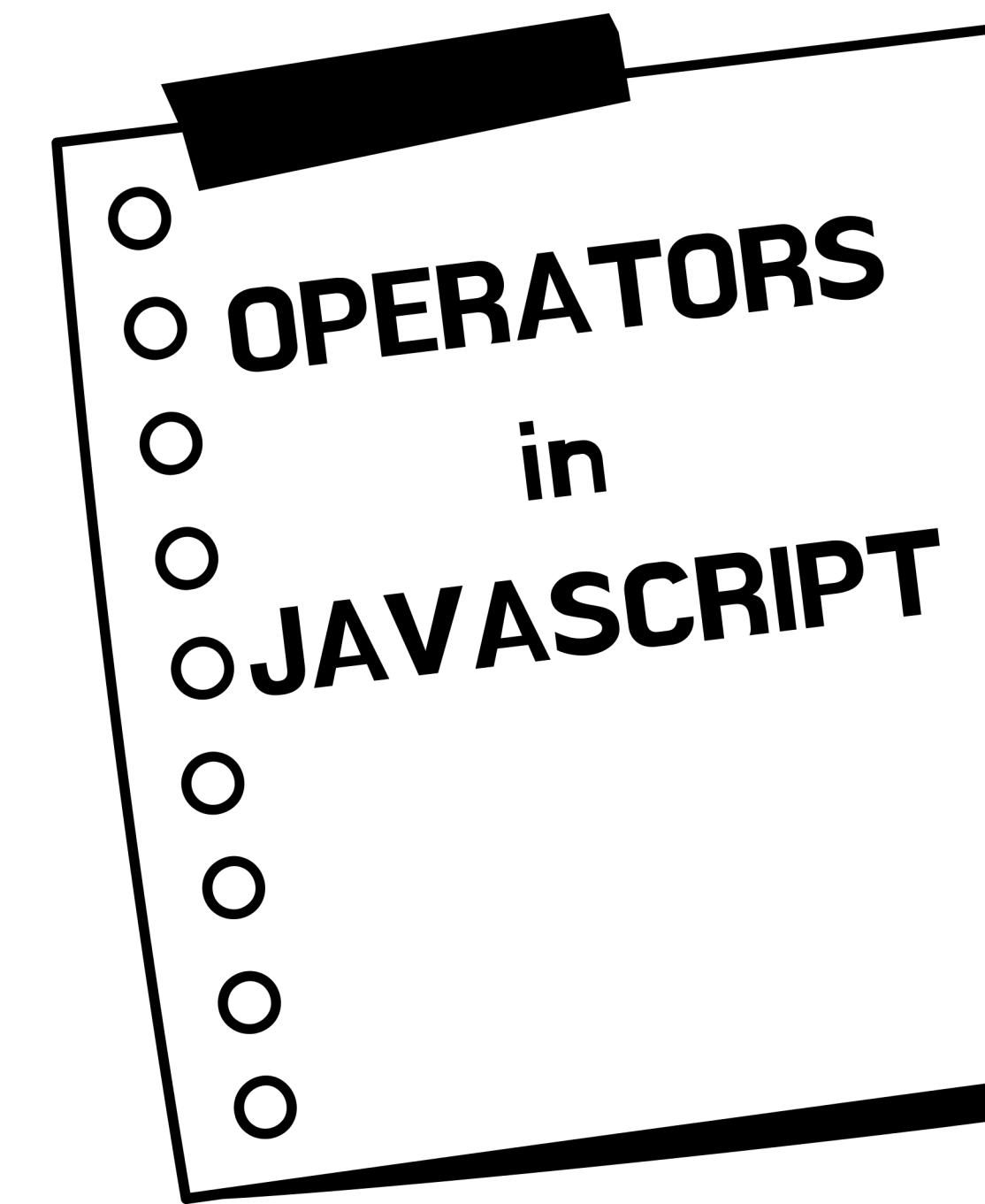


# Operator Penugasan (Assignment)

- Penugasan nilai: "="
- Penugasan tambah: "+="
- Penugasan tolak: "-="

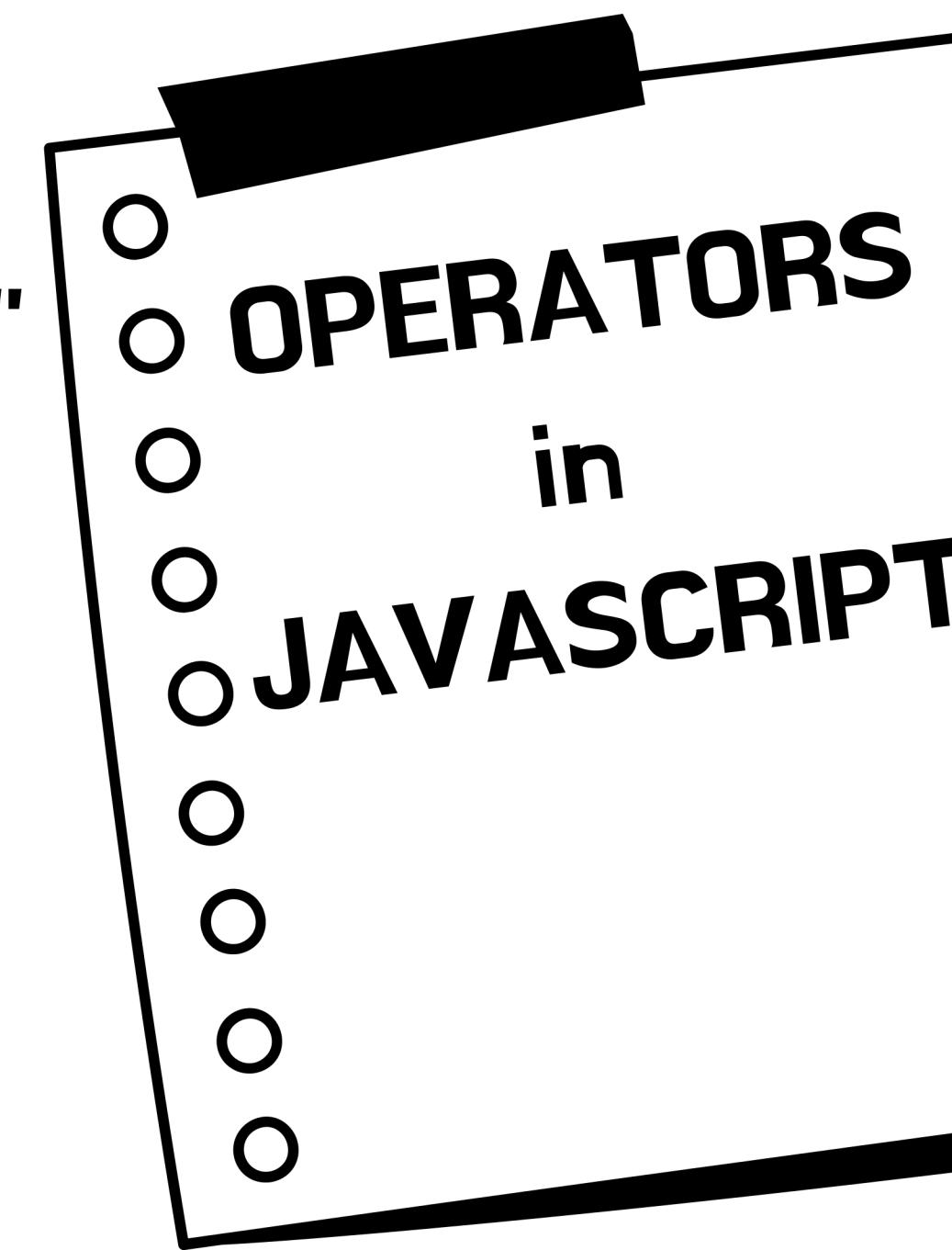
- Penugasan darab: "\*="
- Penugasan bahagi: "/="
- Penugasan modulus: "%="

Jarang  
pakai,  
tapi  
ada



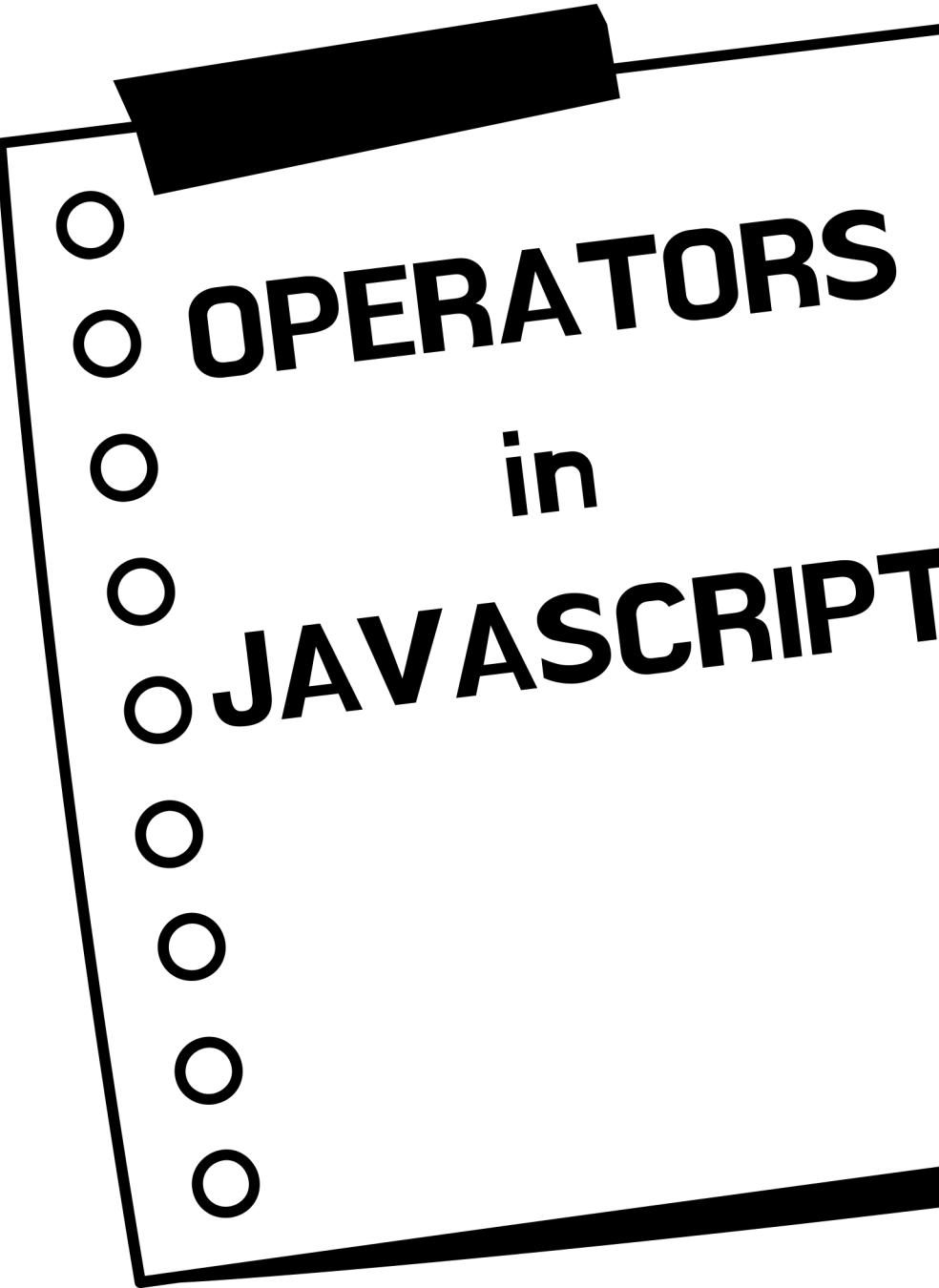
# Operator Pembanding (Comparison)

- **Sama dengan:** "=="
- **Tidak sama dengan:** "!="
- **Sama dengan dan type yang sama:** "==="
- **Tidak sama dengan atau type yang sama:** "!=="
- **Lebih besar:** ">"
- **Lebih kecil:** "<"
- **Lebih besar atau sama dengan:** ">="
- **Lebih kecil atau sama dengan:** "<="



# Operator Logik

- Logik AND: "&&"
- Logik OR: "||"
- Logik NOT: "!"



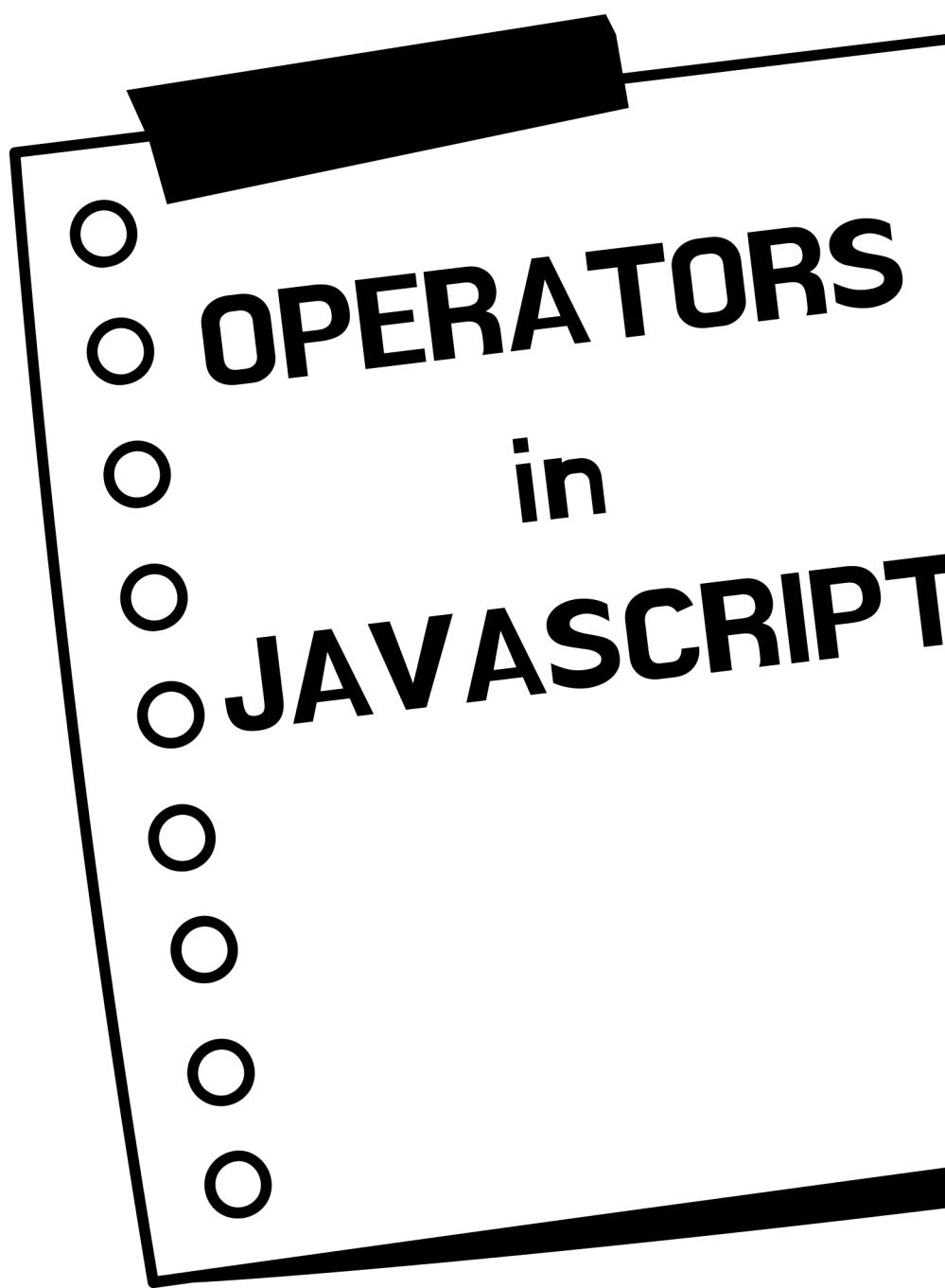
# Operator String

```
var a="Abdul";
```

```
var b="Wahub";
```

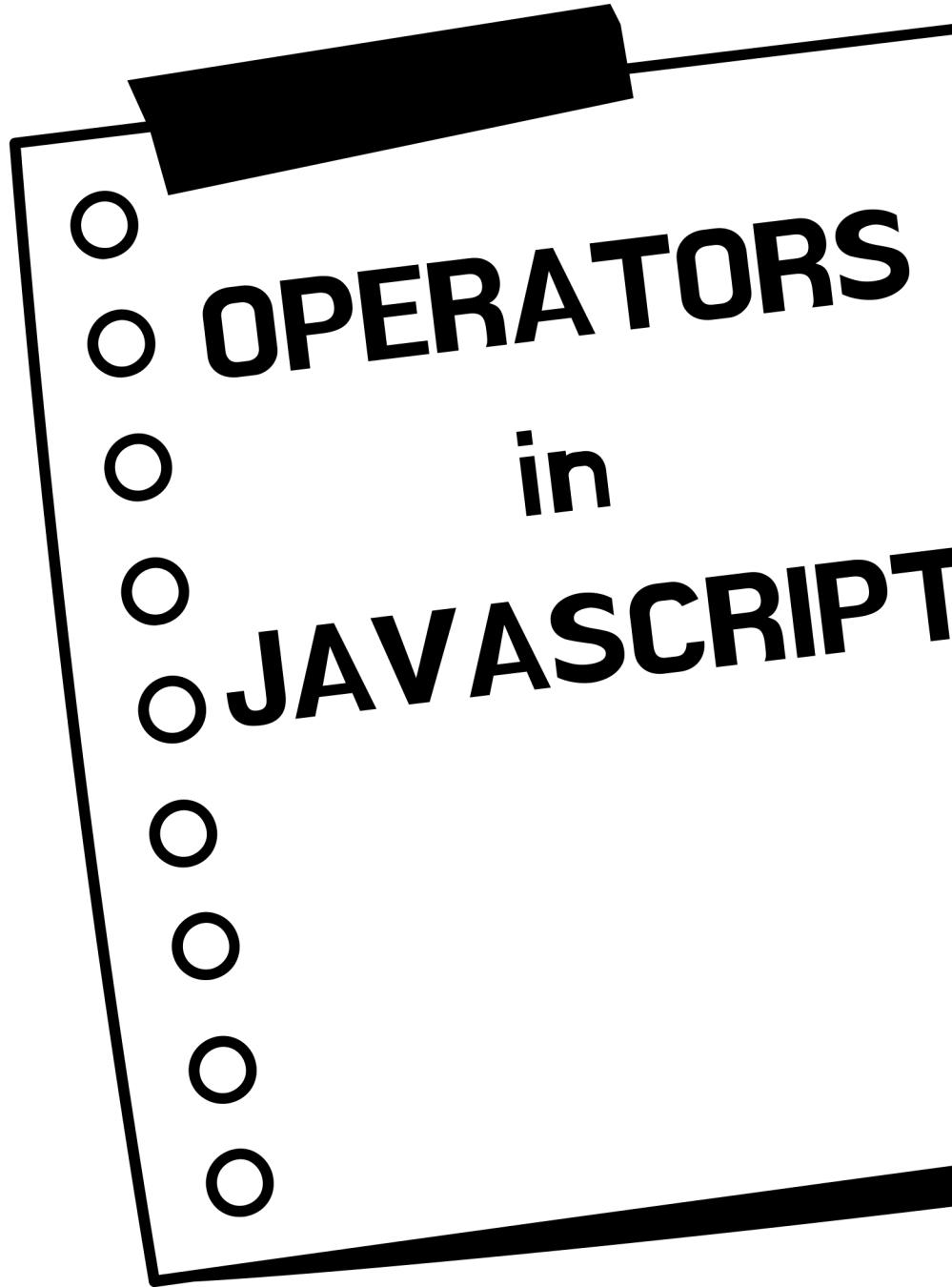
```
var c= a + b;
```

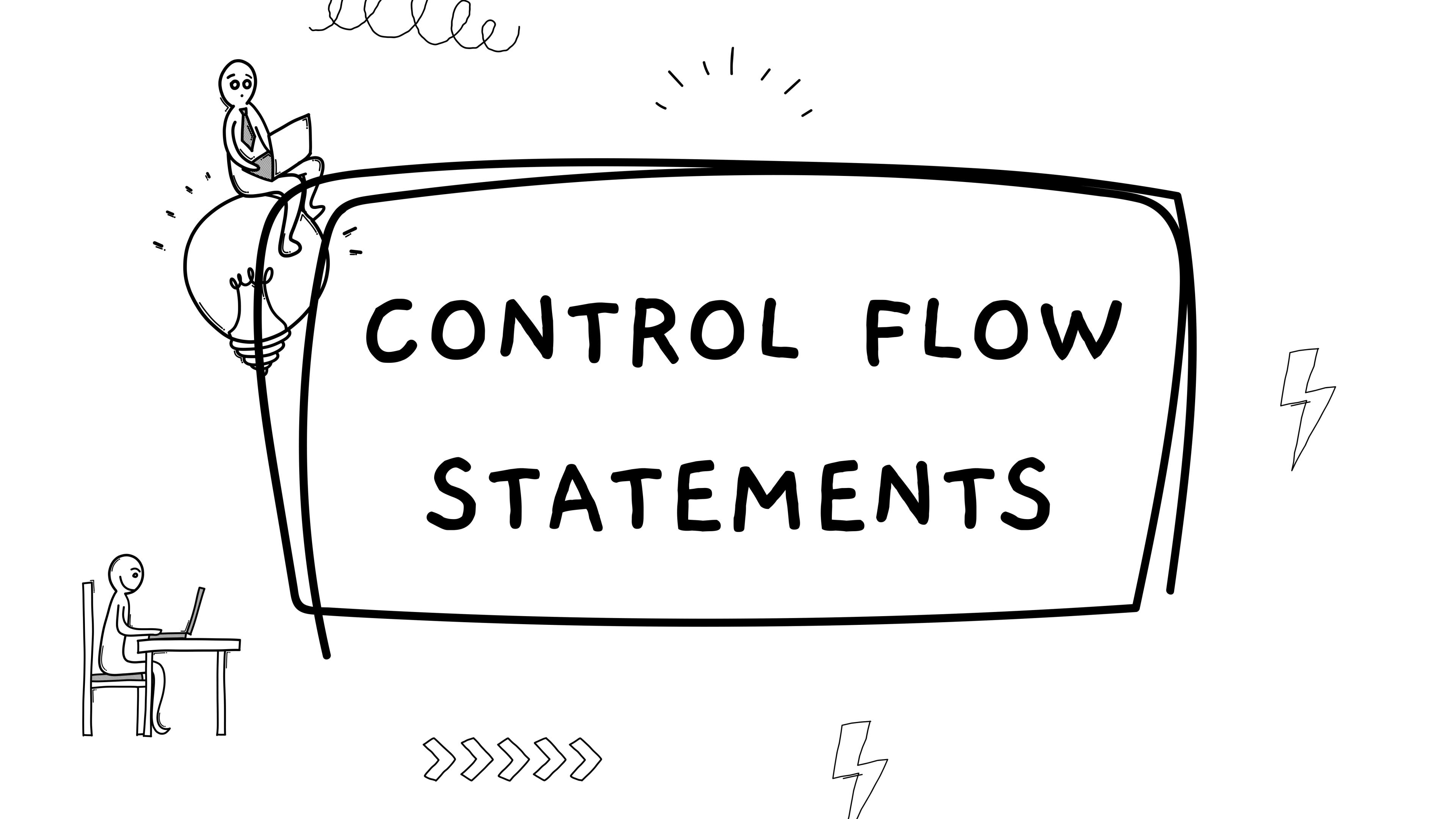
```
//c jadi Abdul Wahub
```



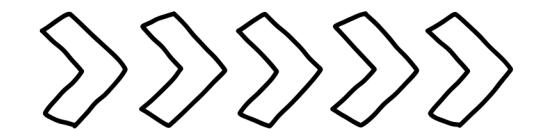
# Operator Baru

- Nullish Coalescing
  - Guna ??
- Ternary
  - Guna ? dan :





# CONTROL FLOW STATEMENTS

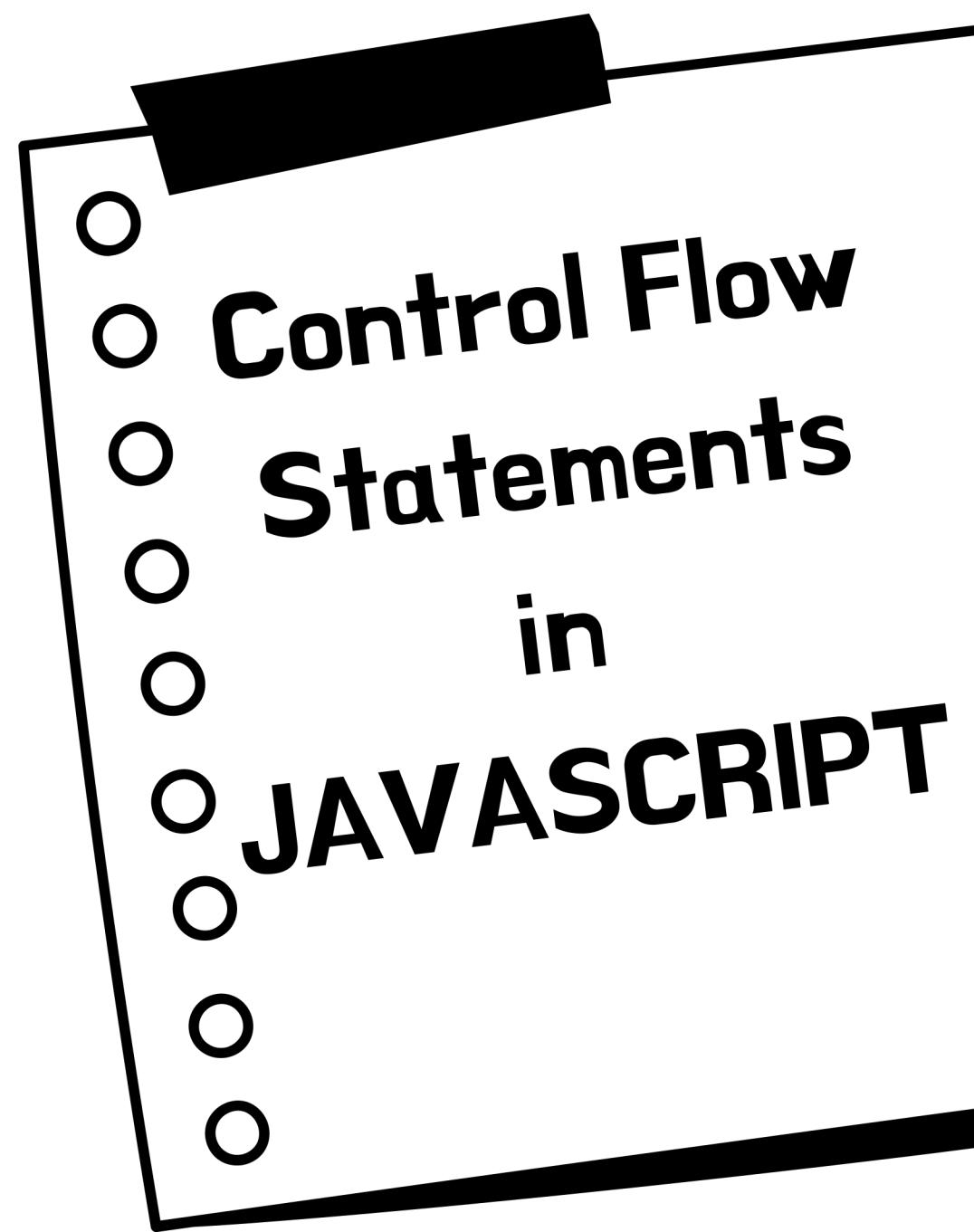


# Apakah Control Flow Statements dalam JS?

Digunakan untuk mengawal urutan pelaksanaan kod berdasarkan keadaan tertentu.

**Control Flow Statements** membolehkan anda membuat keputusan, melaksanakan kod berulang kali, dan menjalankan blok kod tertentu berdasarkan keadaan program

Ada beberapa jenis **Control Flow Statements**

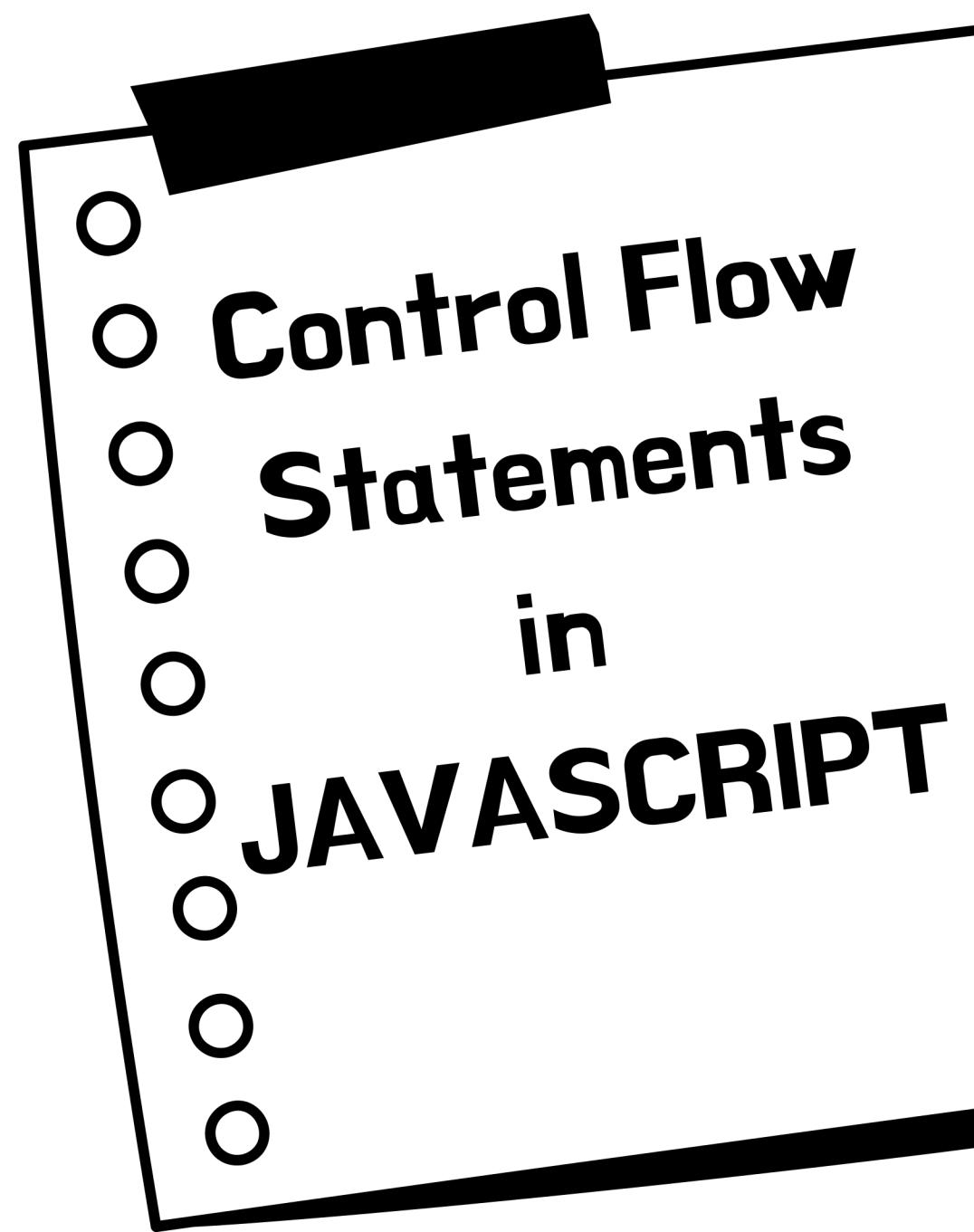


# If - else

- Pernyataan **if** digunakan untuk menjalankan blok kod jika suatu keadaan benar.
- Pernyataan **else** digunakan untuk menjalankan blok kod alternatif jika keadaan adalah palsu (false).

```
var umur = 18;

if (umur >= 18) {
    console.log("Anda sudah dewasa.");
} else {
    console.log("Anda masih belum dewasa.");
}
```

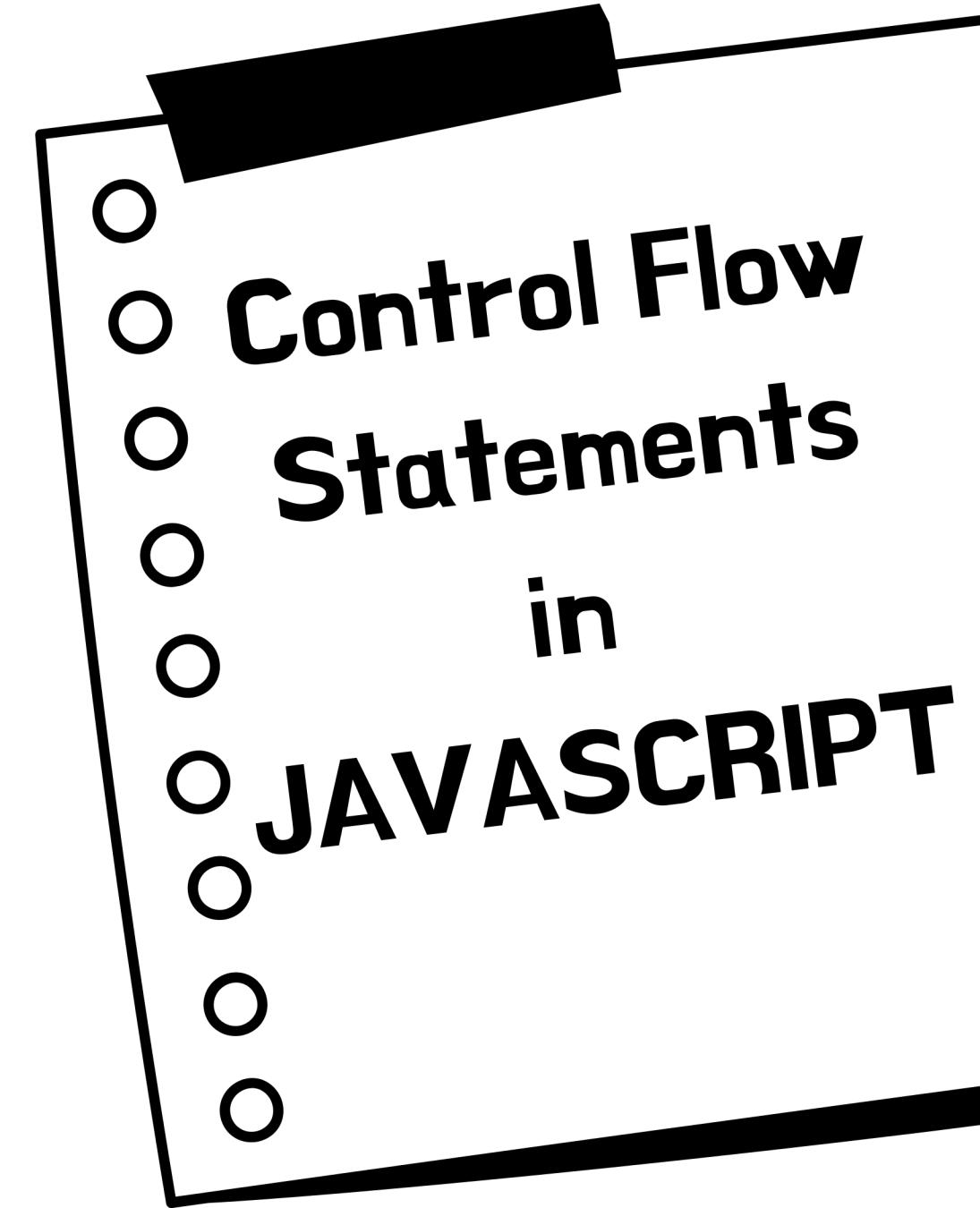


# Switch Statements

- Pernyataan **switch** digunakan untuk mengevaluasi ekspresi dan menjalankan blok kod yang sesuai dengan nilai ekspresi tersebut.

```
var hari = 1;

switch (hari) {
    case 1:
        console.log("Ini hari Isnin.");
        break;
    case 2:
        console.log("Ini hari Selasa.");
        break;
    // ...
    default:
        console.log("Ini bukan hari minggu.");
}
```



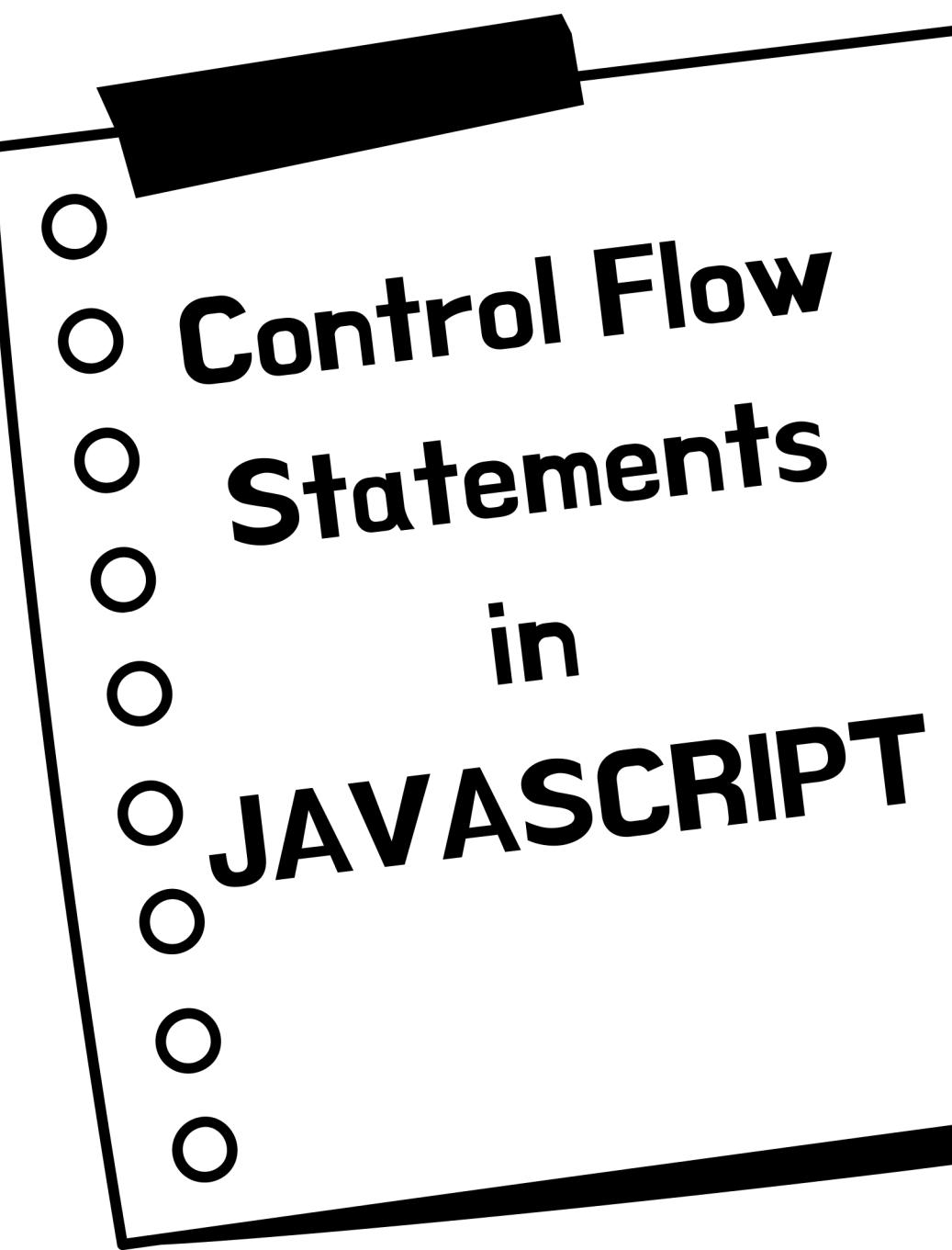
# For Loops

- Pernyataan **for** digunakan untuk melaksanakan blok kod berulang kali dengan syarat tertentu.

```
for (var i = 1; i <= 5; i++) {  
    console.log("Nilai i: " + i);  
}
```

Nilai i: 1  
Nilai i: 2  
Nilai i: 3  
Nilai i: 4  
Nilai i: 5

//Loop akan berhenti  
bila nilai i ialah 5



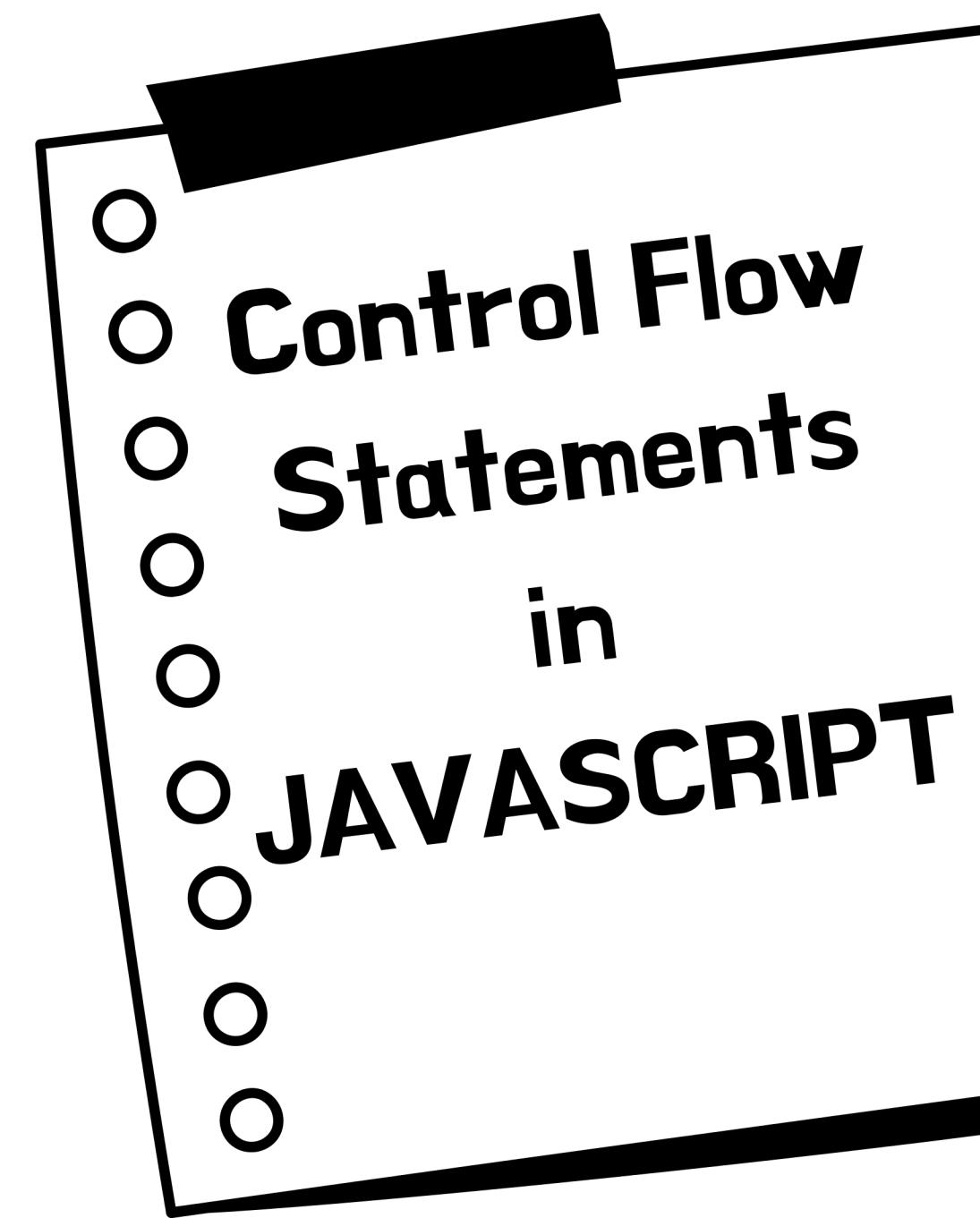
# While Loops

- Pernyataan **while** digunakan untuk melaksanakan blok kod selama suatu keadaan benar.

```
var x = 1;

while (x <= 5) {
  console.log("Nilai x: " + x);
  x++;
}
```

Nilai x: 1  
Nilai x: 2  
Nilai x: 3  
Nilai x: 4  
Nilai x: 5  
  
//Loop akan berhenti bila nilai x ialah 5



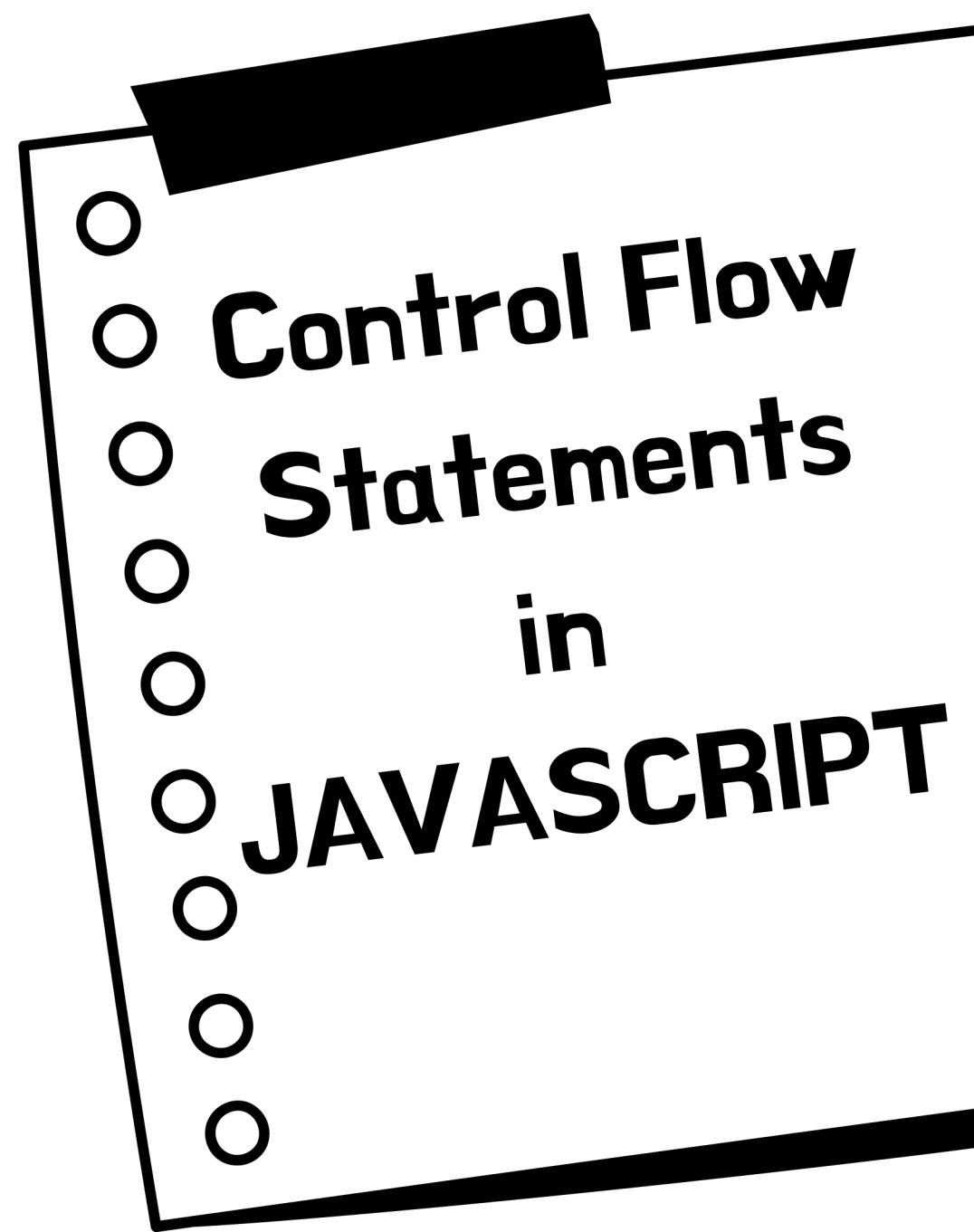
# Do-While Loops

- Pernyataan **do-while** mirip dengan pernyataan **while**, namun blok kod akan dijalankan sekurang-kurangnya sekali sebelum memeriksa keadaan.

```
var y = 1;

do {
    console.log("Nilai y: " + y);
    y++;
} while (y <= 5);
```

Nilai y: 1  
Nilai y: 2  
Nilai y: 3  
Nilai y: 4  
Nilai y: 5  
  
//Loop akan berhenti  
bila nilai y ialah 5

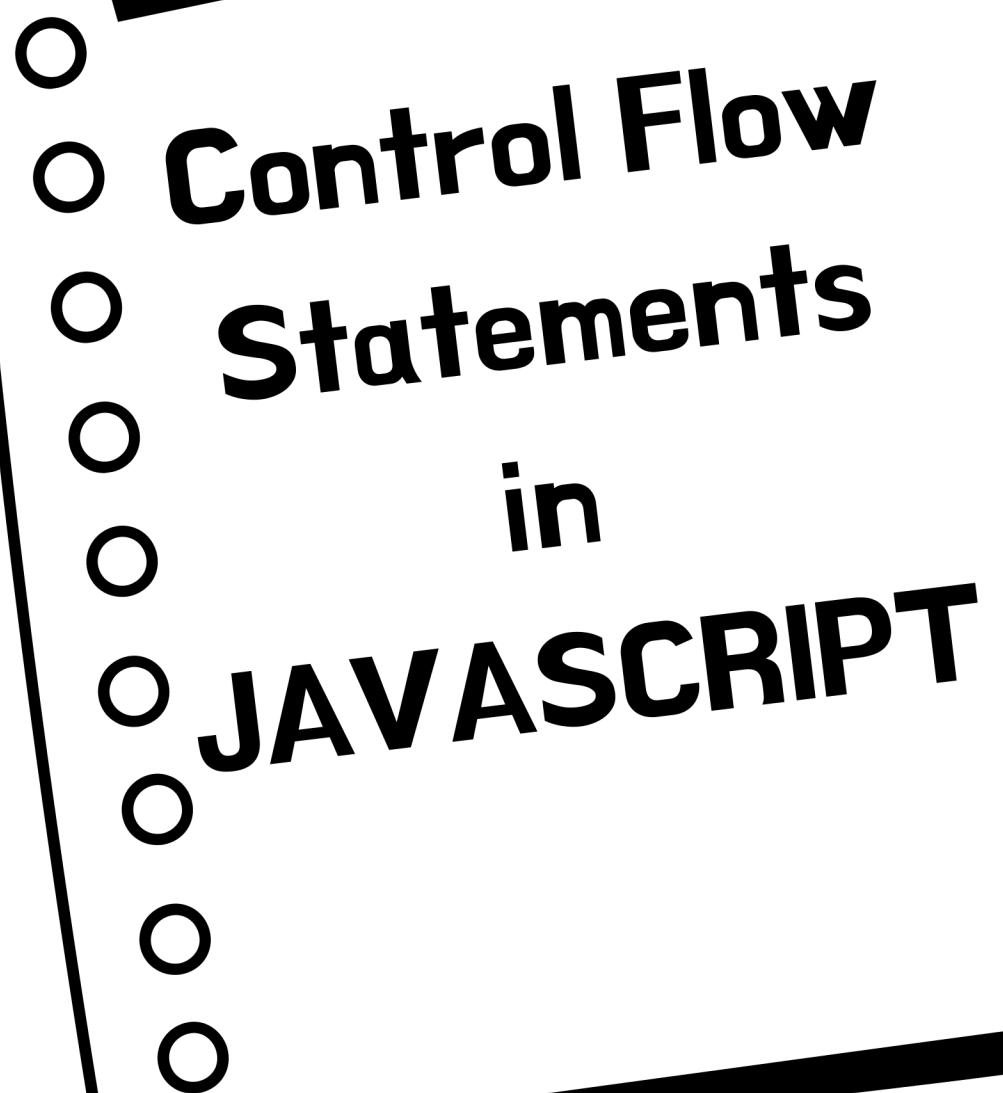


# Break

- Pernyataan **break** dalam digunakan untuk menghentikan pelaksanaan dan keluar dari blok kod.
- Ia sering digunakan dalam pernyataan switch, pernyataan for, pernyataan while, dan pernyataan do-while untuk menghentikan iterasi atau melompat keluar dari loop.

```
var hari = 1;

switch (hari) {
  case 1:
    console.log("Ini hari Isnin.");
    break;
  case 2:
    console.log("Ini hari Selasa.");
    break;
  // ...
  default:
    console.log("Ini bukan hari minggu.");
}
```



- Control Flow
- Statements
- in
- JAVASCRIPT
- 
- 
- 
- 
- 
- 
-

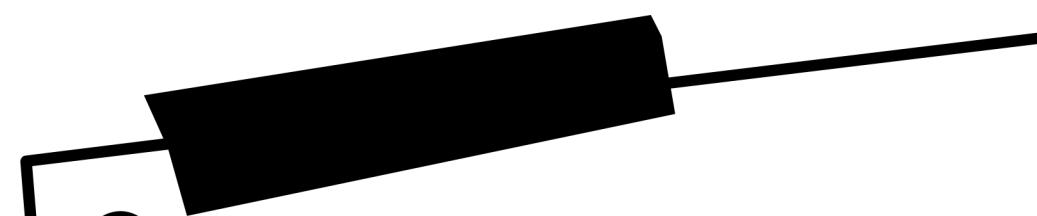
# Continue

Pernyataan **continue** digunakan untuk melompati iterasi dalam pernyataan loop dan meneruskan ke iterasi berikutnya.

Dalam pernyataan for, **continue** digunakan untuk mengabaikan sisa pernyataan dalam blok for dan meneruskan dengan nilai iterasi berikutnya.

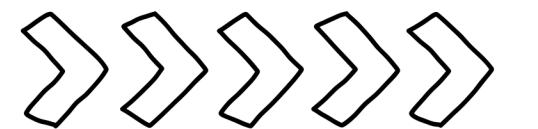
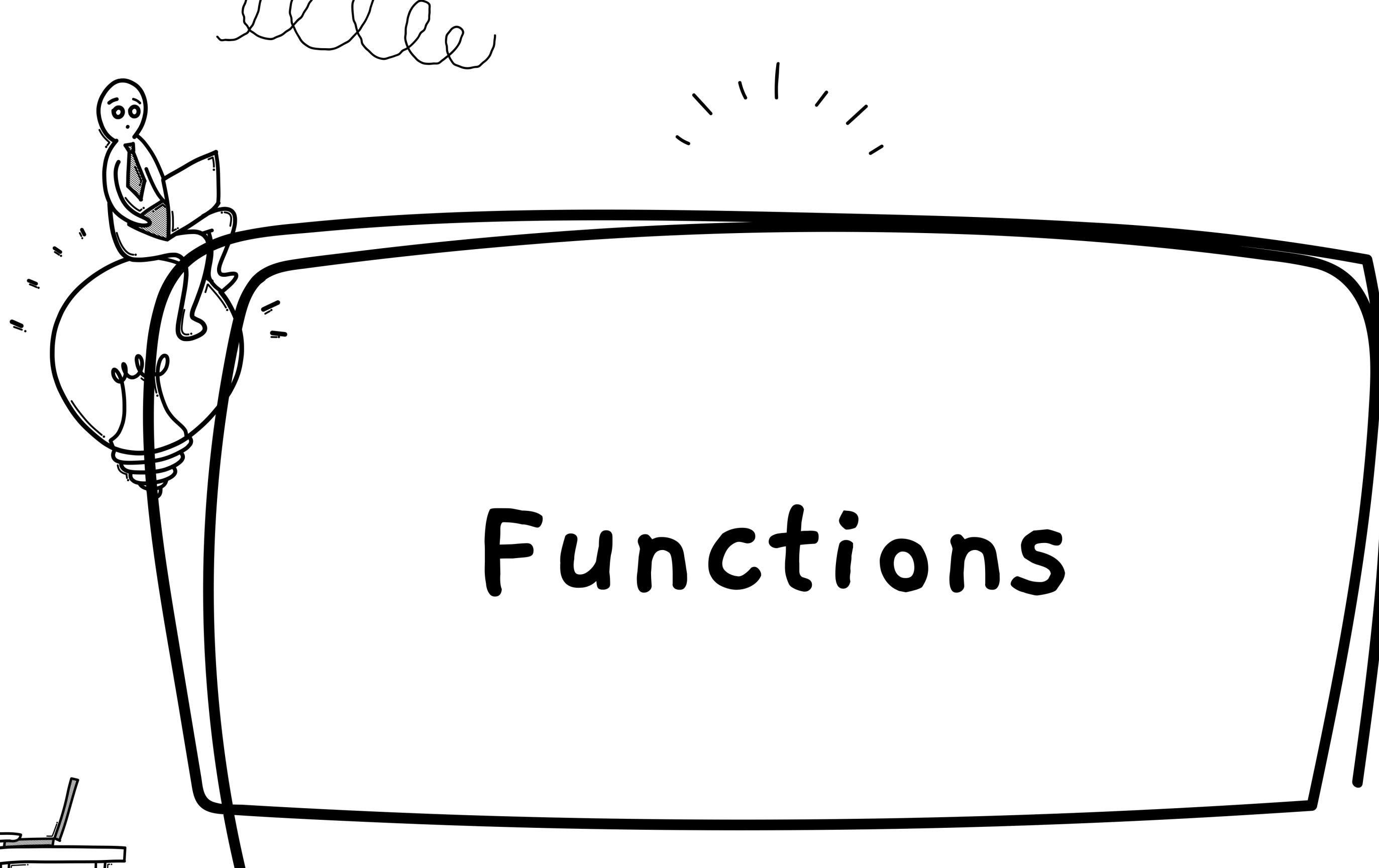
```
for (var i = 1; i <= 5; i++) {  
  if (i == 3) {  
    continue;  
  }  
  console.log(i);  
}
```

1  
2  
4  
5



Control Flow  
Statements  
in  
JAVASCRIPT

# Functions



# Apakah Functions dalam JS?

Digunakan untuk melakukan tugas tertentu atau menghasilkan output yang dapat digunakan kembali di dalam program.

Ia membolehkan pecahan kod menjadi bahagian yang lebih kecil dan tersusun, mengurangi duplikasi kod dan membantu pemuliharaan kod

## DEKLARASI FUNCTION

### CONTOH TADI:

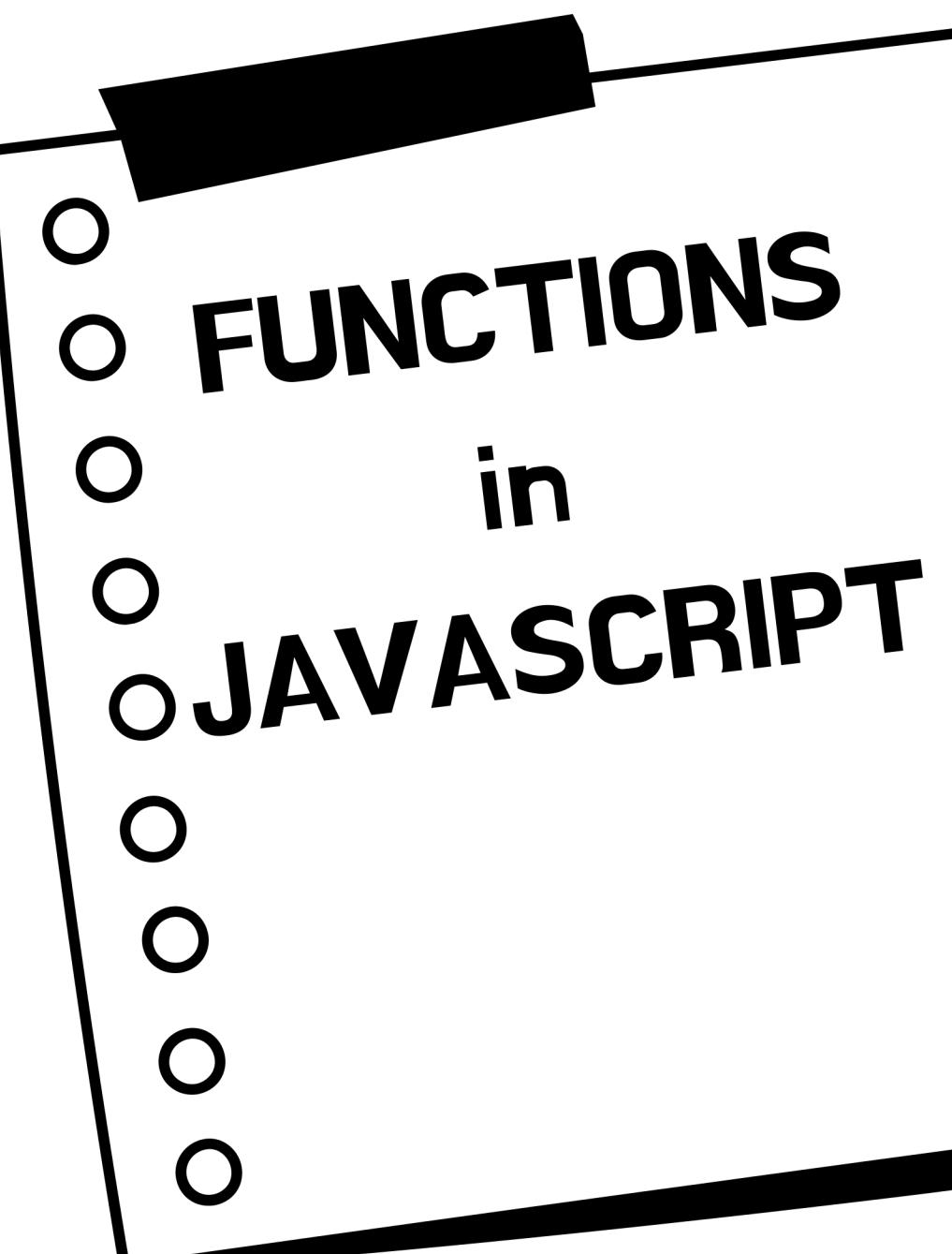
Ini adalah fungsi (function) yang diberi nama 'tambah':

```
function tambah(a, b) {  
    return a + b;  
}
```

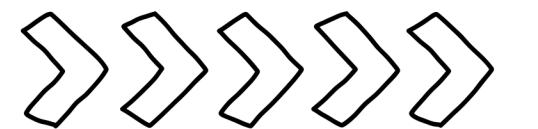
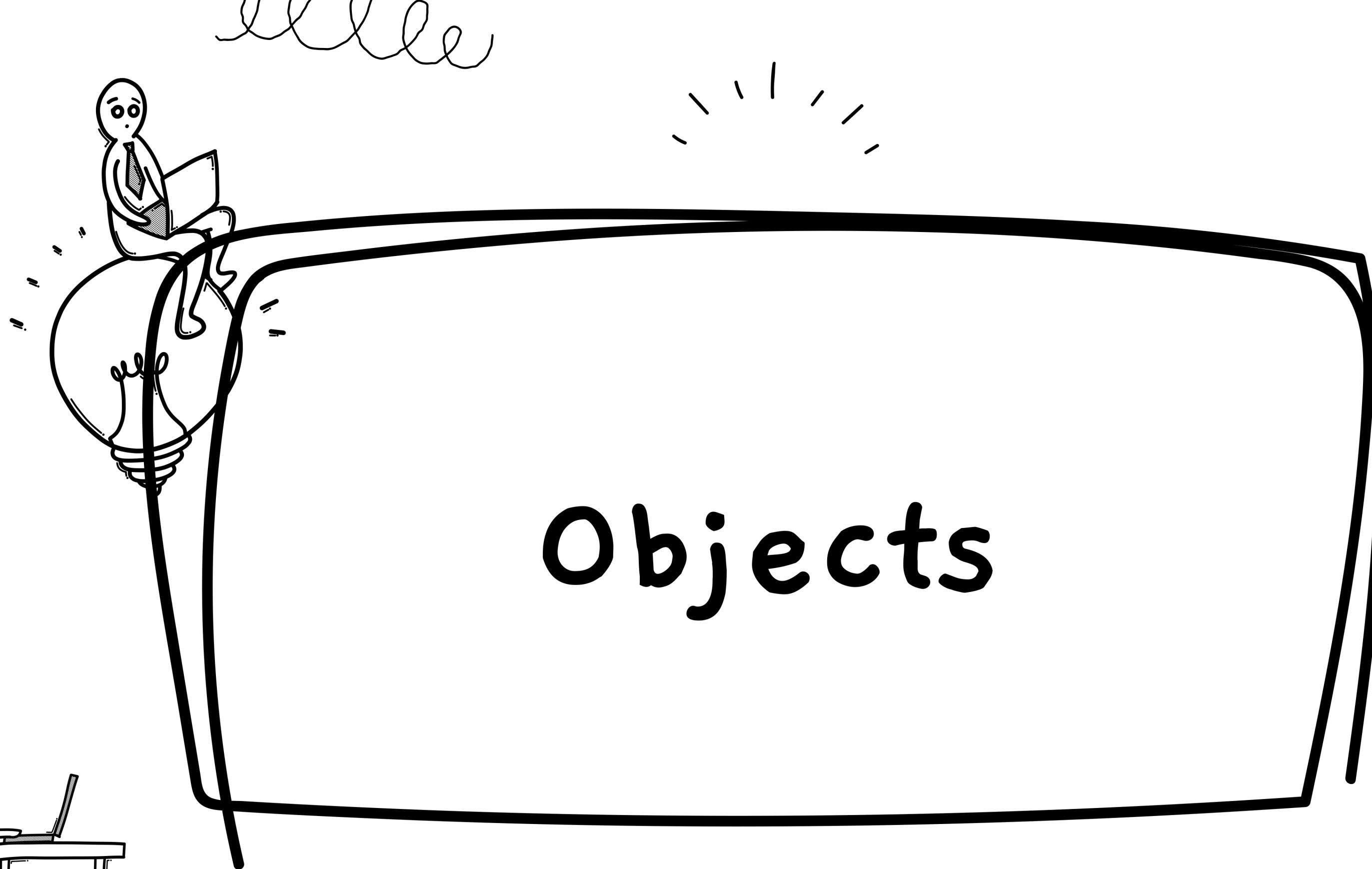
- Fungsi tambah mengambil 2 nilai
- Kedua-dua nilai akan ditambah dan akan di kembalikan kepada pemanggil
- a dan b di dalam function ini dipanggil parameters

```
var hasilTambah = tambah(2, 3);  
console.log(hasilTambah); // Output: 5
```

Cara kita panggil  
function tersebut



Objects



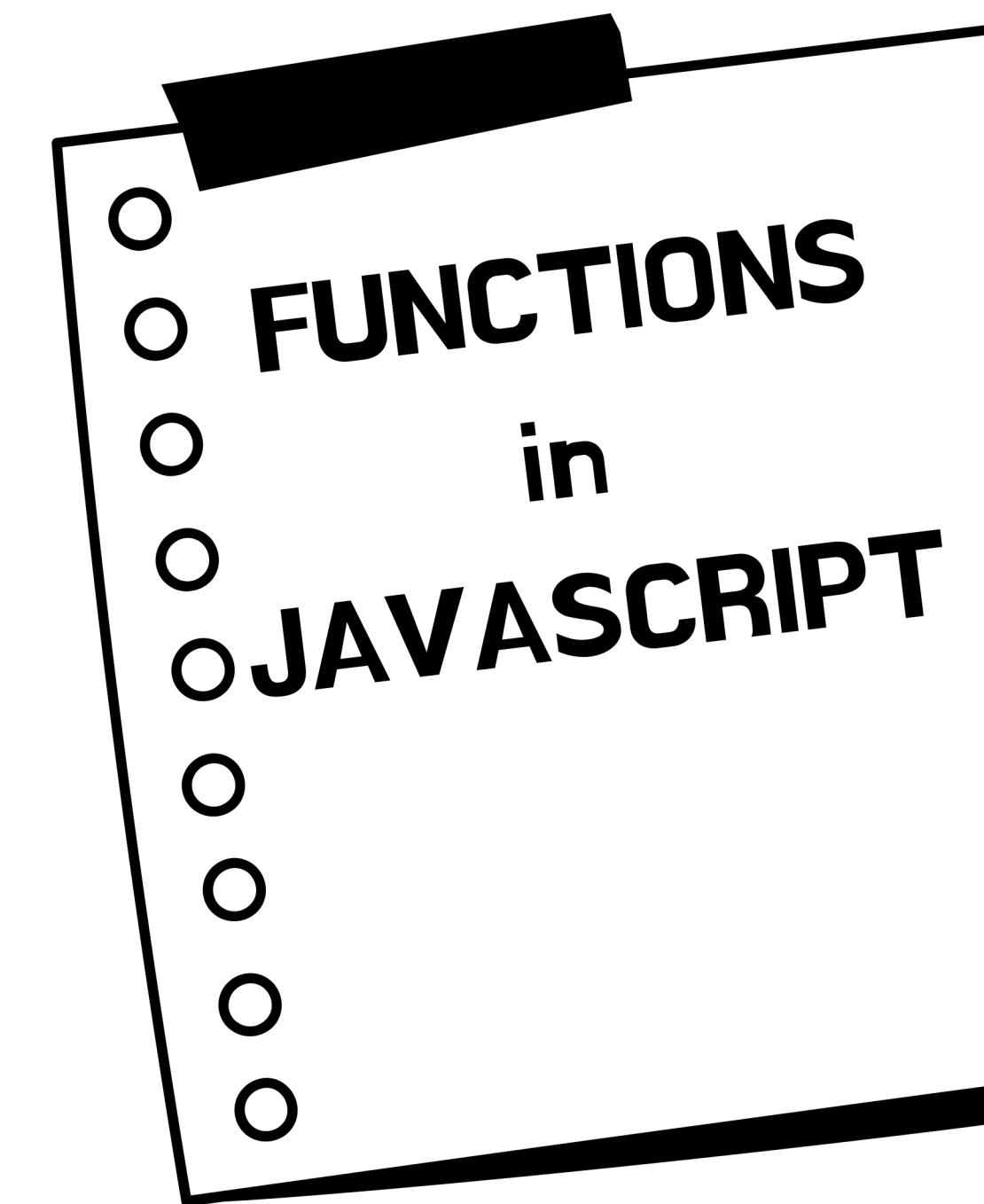
# Apakah Objects dalam JS?

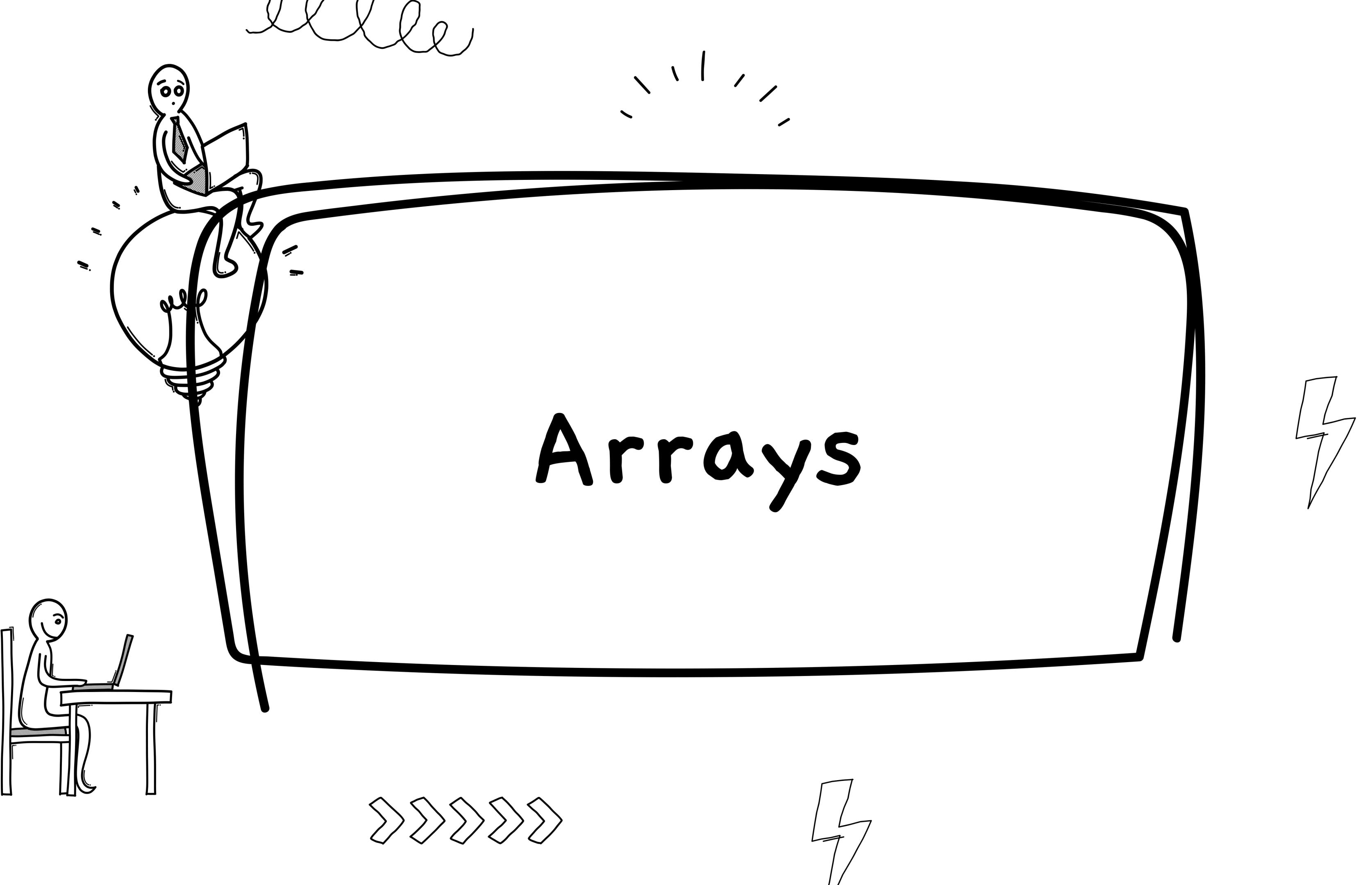
Struktur data yang digunakan untuk menyimpan kumpulan nilai dan perilaku yang terkait.

```
// Contoh objek kucing
var kucing = {
  nama: "Mimi",
  umur: 3,
  jenis: "Persia",
  bersuara: function() {
    console.log("Meow!");
  }
};

// Mengakses properti objek
console.log("Nama kucing: " + kucing.nama);
console.log("Umur kucing: " + kucing.umur);
console.log("Jenis kucing: " + kucing.jenis);

// Memanggil metode objek
kucing.bersuara(); // Output: Meow!
```



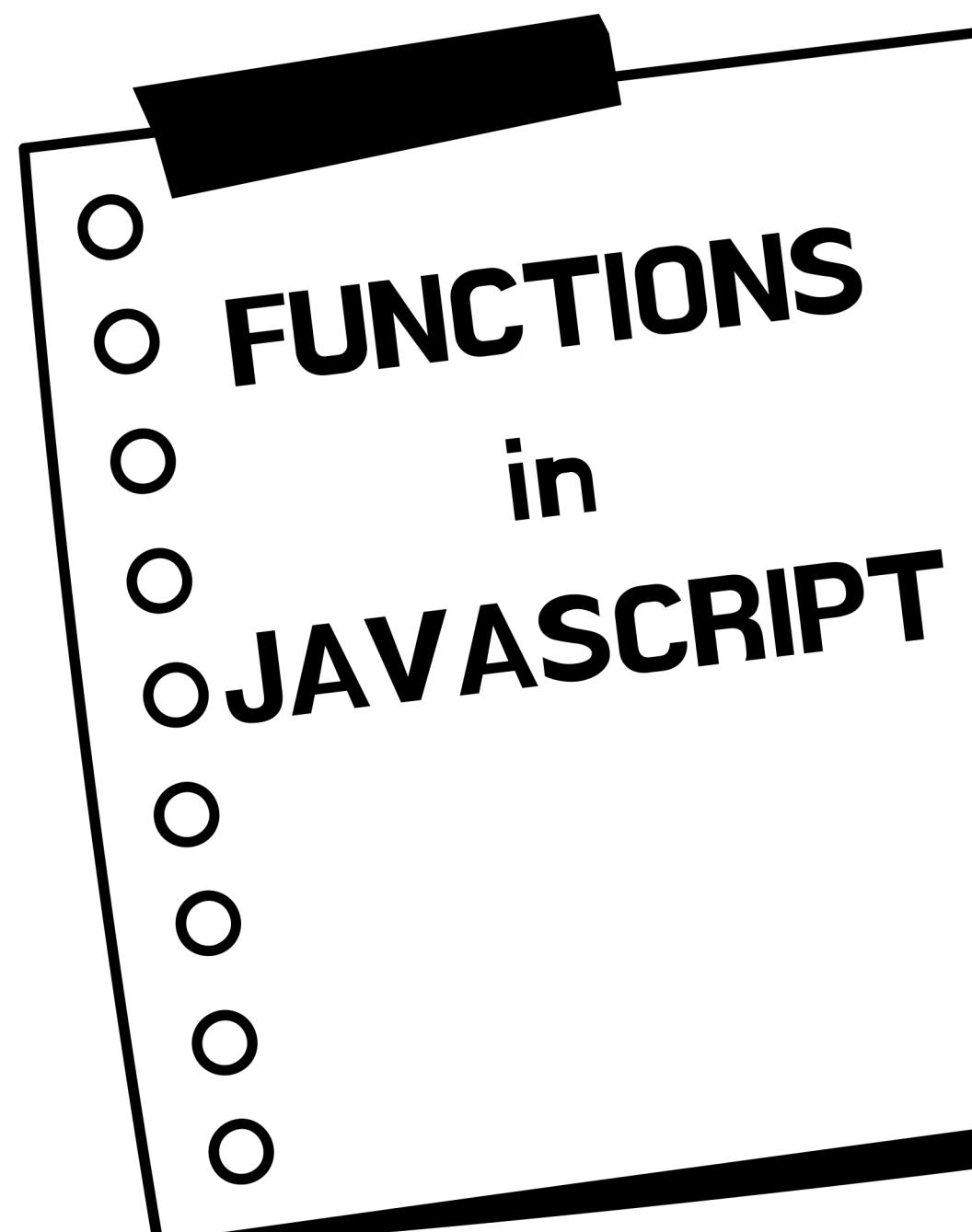


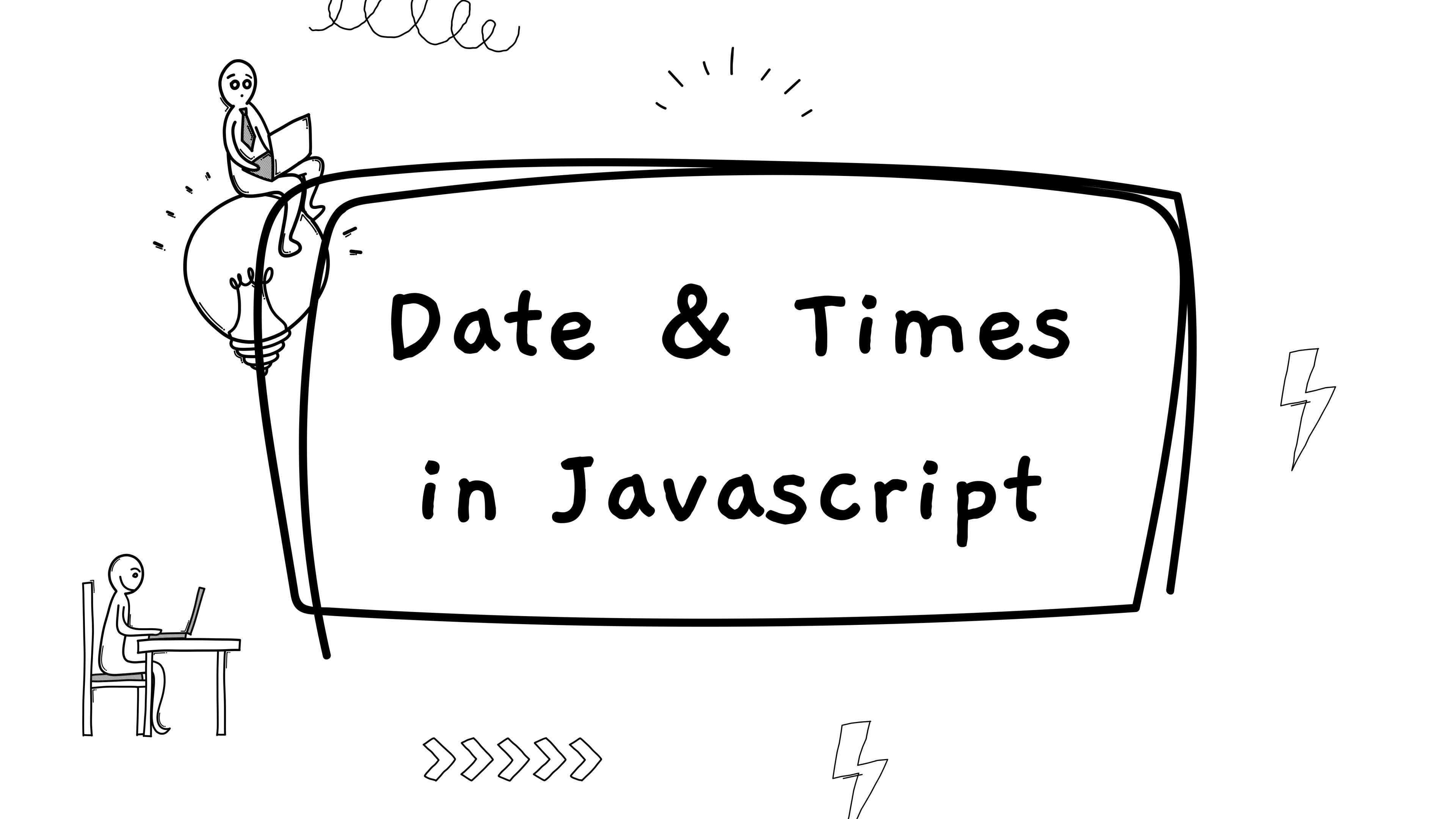
# Apakah Arrays dalam JS?

```
// Contoh array buah-buahan  
var buah = ["apel", "mangga", "pisang", "jeruk"];  
  
// Mengakses elemen array  
console.log("Buah pertama: " + buah[0]);  
console.log("Buah kedua: " + buah[1]);  
console.log("Buah ketiga: " + buah[2]);  
  
// Mengubah elemen array  
buah[1] = "rambutan";  
console.log("Buah kedua setelah diubah: " + buah[1]);  
  
// Menambahkan elemen baru ke array  
buah.push("durian");  
console.log("Buah terakhir: " + buah[buah.length - 1]);  
  
// Menghitung jumlah elemen dalam array  
console.log("Jumlah buah: " + buah.length);
```

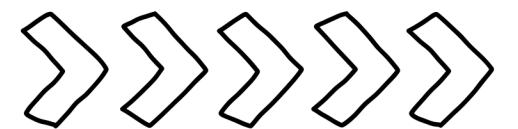
**Struktur data yang digunakan untuk menyimpan kumpulan nilai atau elemen dalam urutan tertentu.**

**Array memungkinkan pengelompokan nilai-nilai tersebut dalam satu variabel, yang dapat diakses dan dimanipulasi secara efisien.**





# Date & Times in Javascript



# Date & Times dalam JS?

```
var sekarang = new Date();

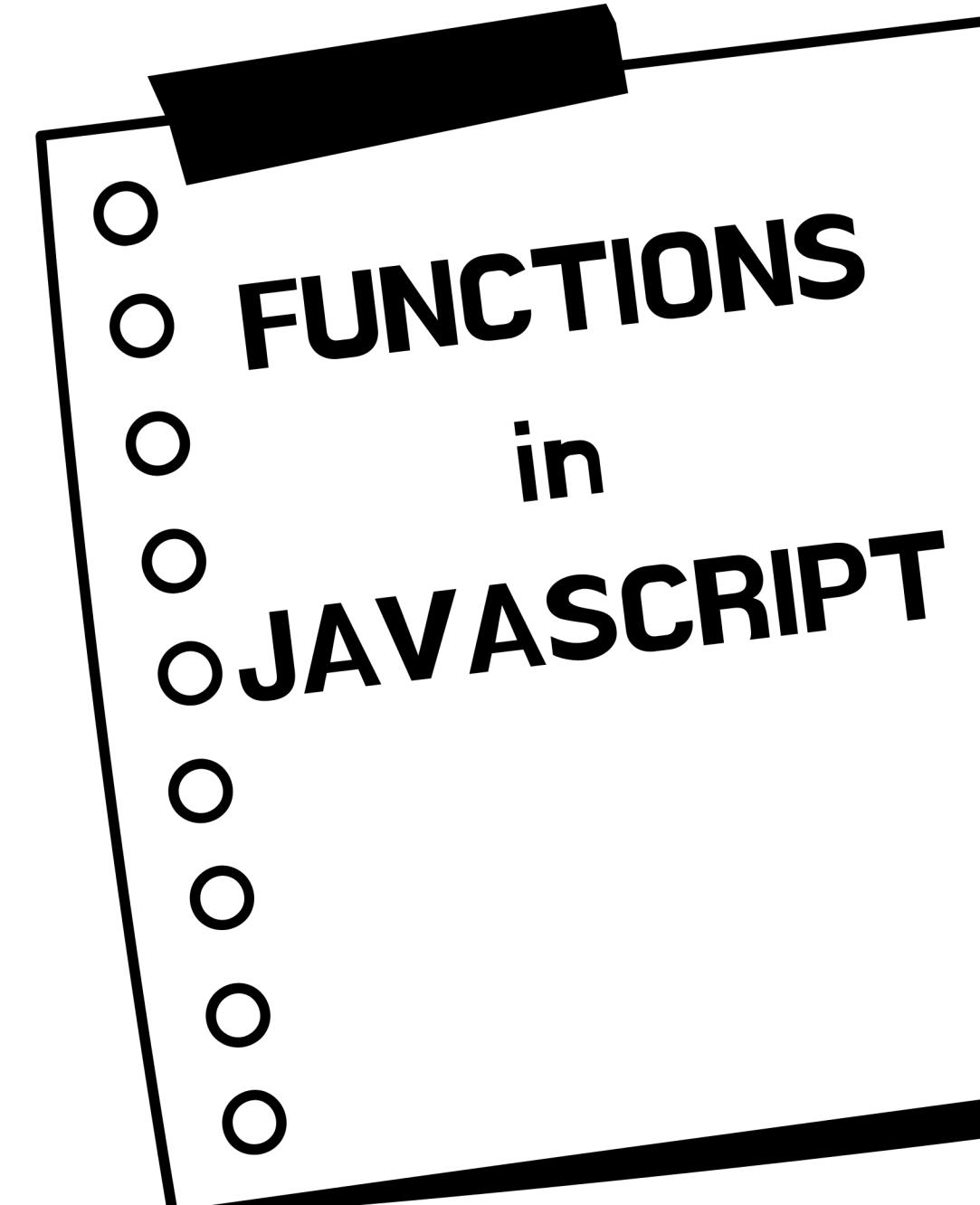
var tahun = sekarang.getFullYear();
var bulan = sekarang.getMonth(); // Mengembalikan angka bulan (0-11)
var tanggal = sekarang.getDate();
var jam = sekarang.getHours();
var menit = sekarang.getMinutes();
var detik = sekarang.getSeconds();

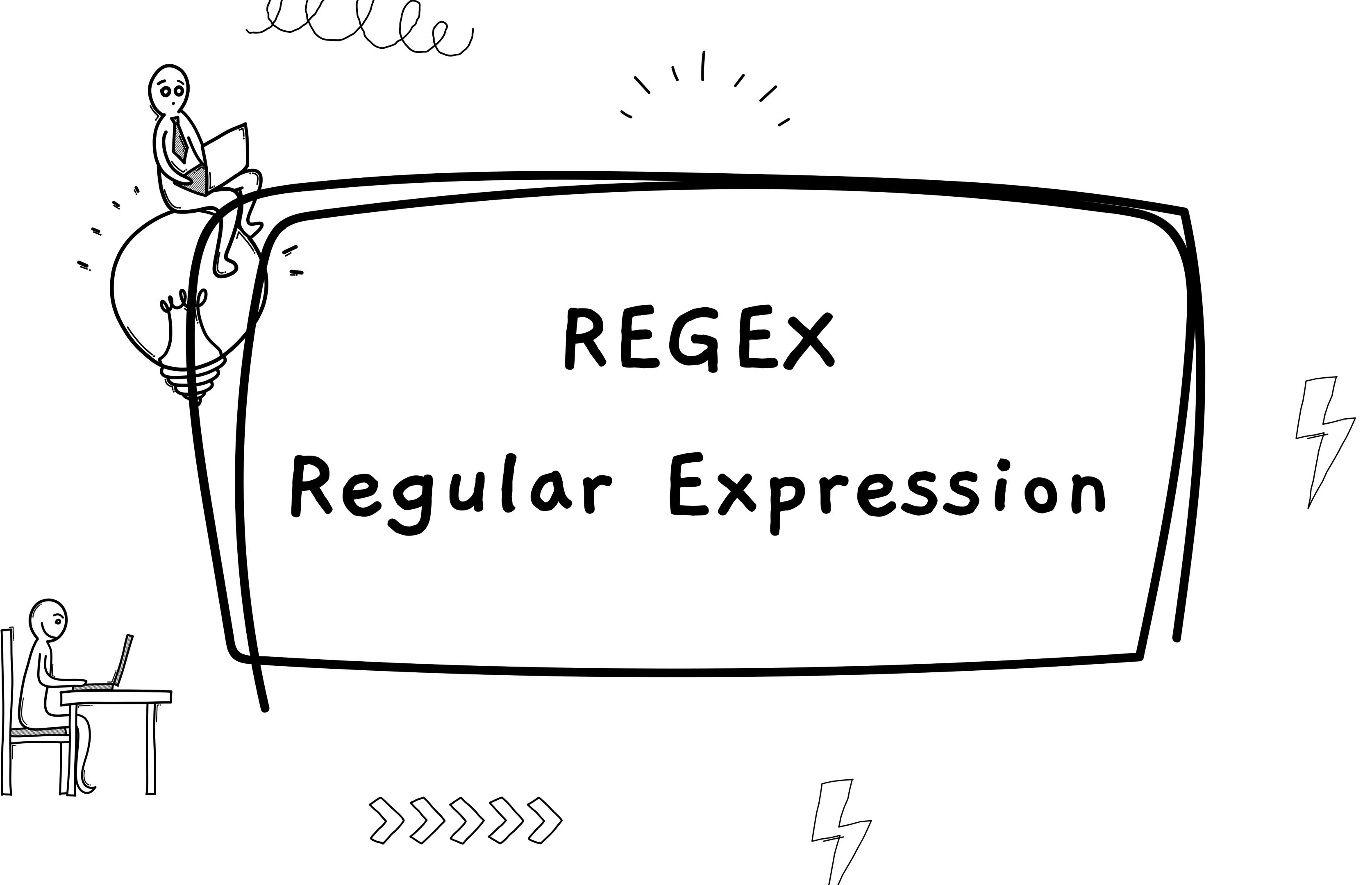
console.log(tahun, bulan, tanggal, jam, menit, detik); // Contoh output:
```

Digunakan untuk  
memanipulasi tarikh  
(dates) dan waktu  
(times) dalam JS

```
var sekarang = new Date();

console.log(sekarang.toDateString()); // Output: Fri May 12 2023
console.log(sekarang.toLocaleString()); // Output: 5/12/2023, 3:30:00 PM
```



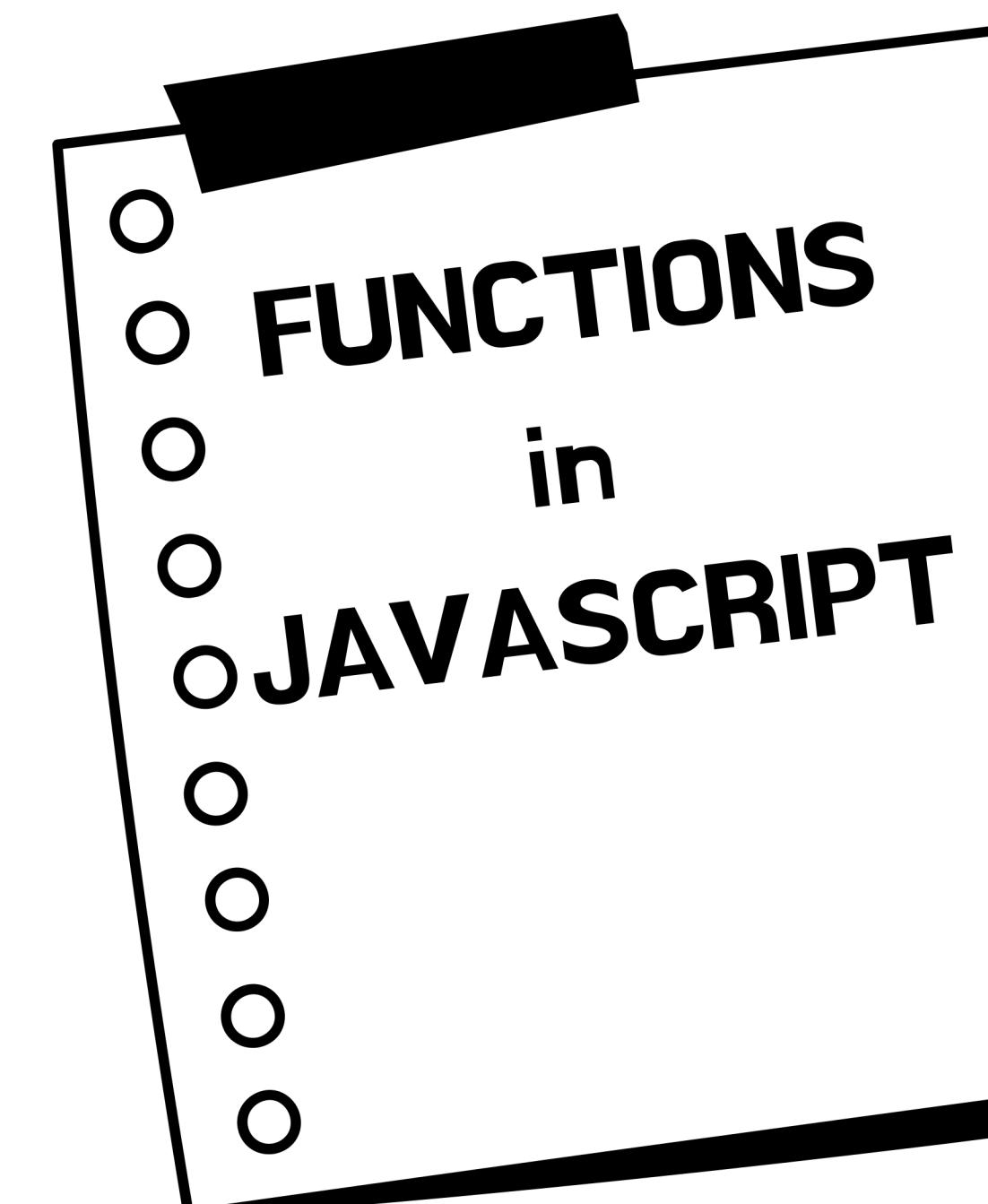


# REGEX dalam JS?

Regex membolehkan anda mencari, memeriksa, dan memanipulasi teks berdasarkan pola tertentu

```
var teks = "Hari ini cuaca sangat cerah.";  
  
var pola = /cerah/;  
var hasil = pola.test(teks);  
console.log(hasil); // Output: true  
  
var cocokan = teks.match(pola);  
console.log(cocokan); // Output: ["cerah"]
```

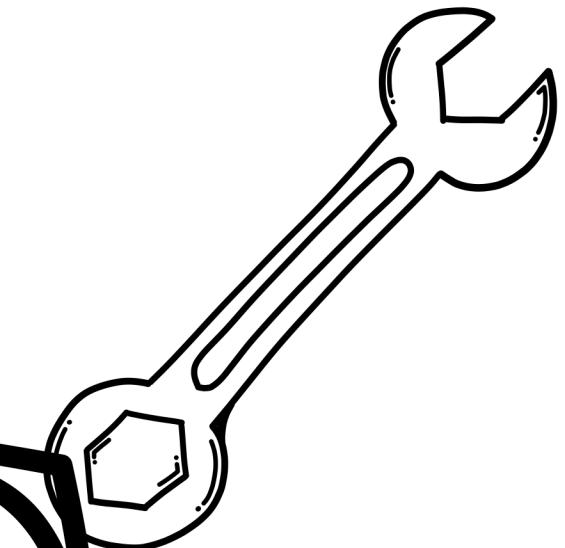
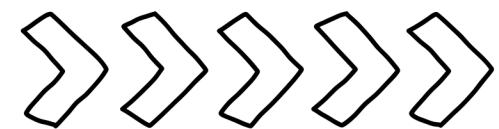
```
var teks = "Saya suka makan nasi goreng.";  
  
var pola = /nasi goreng/;  
var hasil = teks.replace(pola, "pizza");  
console.log(hasil); // Output: "Saya suka makan pizza."
```



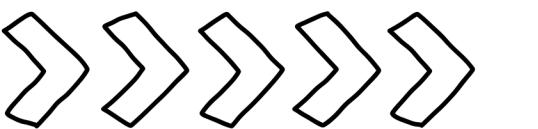
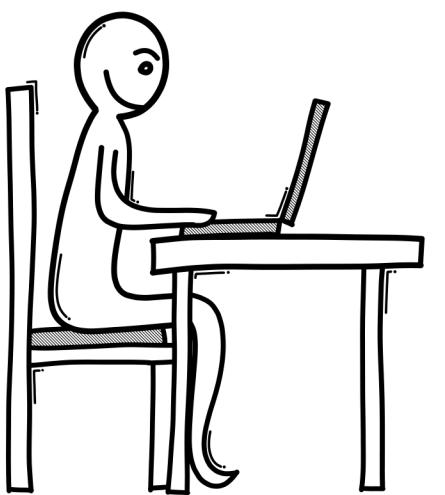
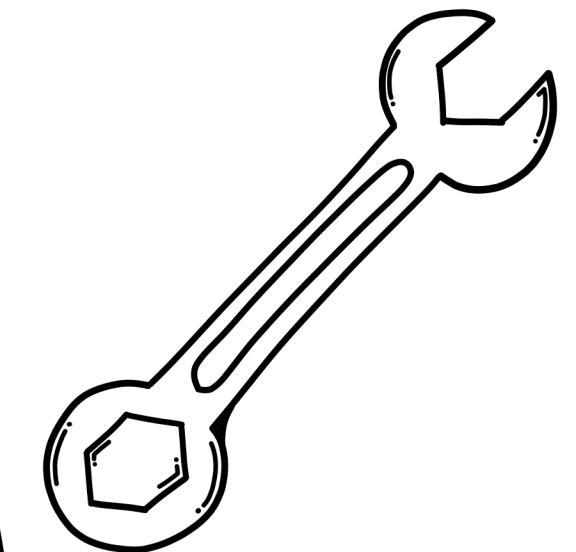
# ASSIGNMENT

1. Output guna For loop, nombor 1 hingga 100
2. Untuk setiap nombor yang boleh dibahagi dengan 7, keluar kenyataan 'Hebat! '

Nanti hantar **Gist Github**. Borang akan diberi kalau ada problem boleh tanya mentor atau saya



# Q & A



Ok settle  
Kita sambung  
minggu depan

Have a  
good day!

