



# Full Stack Development 1<sup>o</sup>

---



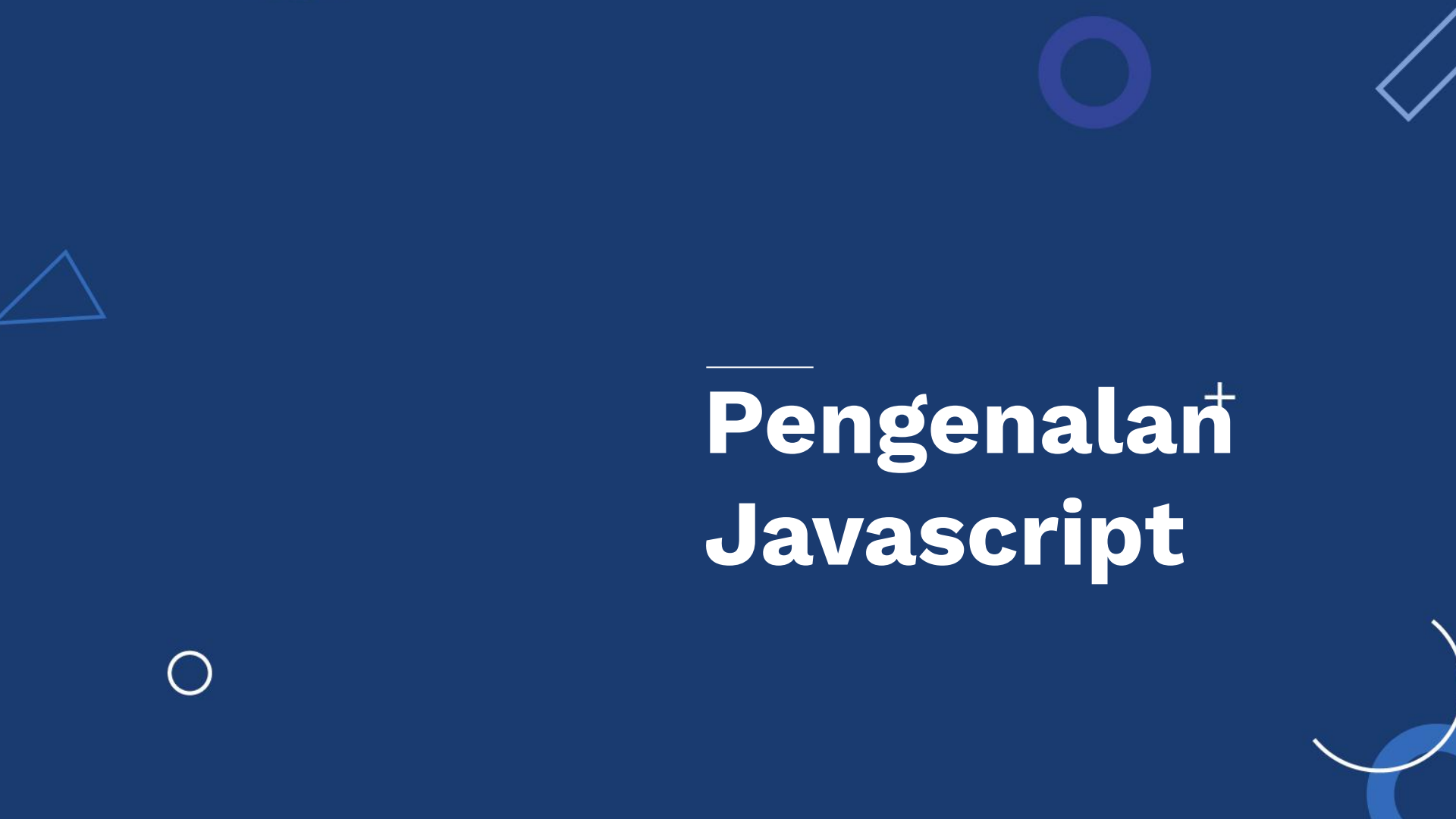
OCBC Bootcamp Program



# Front End Development 1

## Sesi 17

---



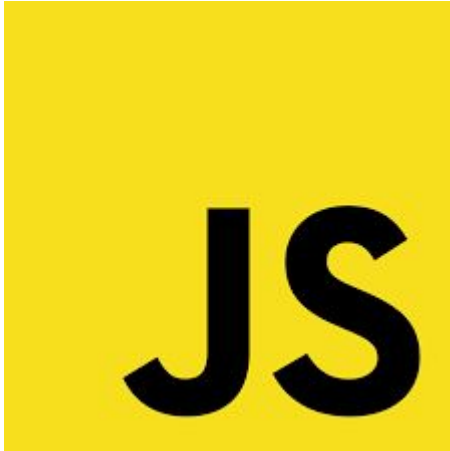
# Pengenalan<sup>+</sup> Javascript

# Point of Discussion

- JavaScript
- Jenis Tipe data di JavaScript
- Variabel dan penulisannya
- Operator di JavaScript
  - Operator Matematik
  - Operator Perbandingan

Apa itu

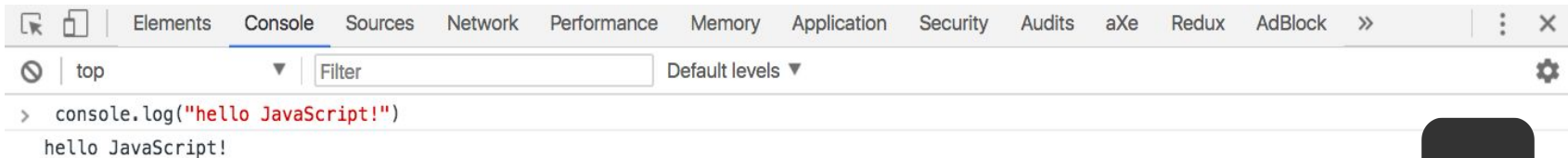
# JavaScript



- Bahasa Pemrograman terpopuler di dunia
- Bahasa Pemrograman Web.
- Mudah dipelajari

JavaScript

# Basic Browser Script



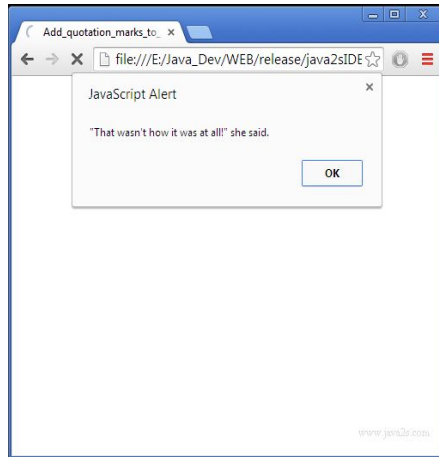
JS

`console.log("hello JavaScript")`

Menampilkan pesan pada console

`alert("hello JavaScript")`

Menampilkan pesan alert pada browser



JavaScript

# Node JS



Perangkat lunak untuk mengembangkan aplikasi yang ditulis dengan bahasa pemrograman JavaScript, atau JavaScript yang dijalankan diluar Web Browser.

Penulisan di JavaScript

# Jenis data

Data memiliki beberapa tipe:

Primitif

**number, string, boolean**, undefined, bigint

Special primitif

Null

Struktural

(di pertemuan lain)





Penulisan di JavaScript

# Tipe Data

## Primitif

- **Number**: 5, 12, 3.14
- **String**: 'hello', "hello", "5"
- **Boolean**: true, false



Mencetak nilai pada terminal javascript

# Menggunakan console.log();

```
console.log(42);  
// expected output: 42
```

```
console.log( "blubber" );  
// expected output: "blubber"
```

```
console.log(true);  
// expected output: true
```

```
console.log(undefined);  
// expected output: undefined
```

Mencari tahu jenis Tipe Data dengan

# Typeof di JavaScript

```
console.log(typeof 42);  
// expected output: "number"
```

```
console.log(typeof 'blubber');  
// expected output: "string"
```

```
console.log(typeof true);  
// expected output: "boolean"
```

```
console.log(typeof undeclaredVariable);  
// expected output: "undefined"
```

# Variabel & Penamaannya

## Deskriptif & bukan angka

=> bucket;

=> input;

=> output;

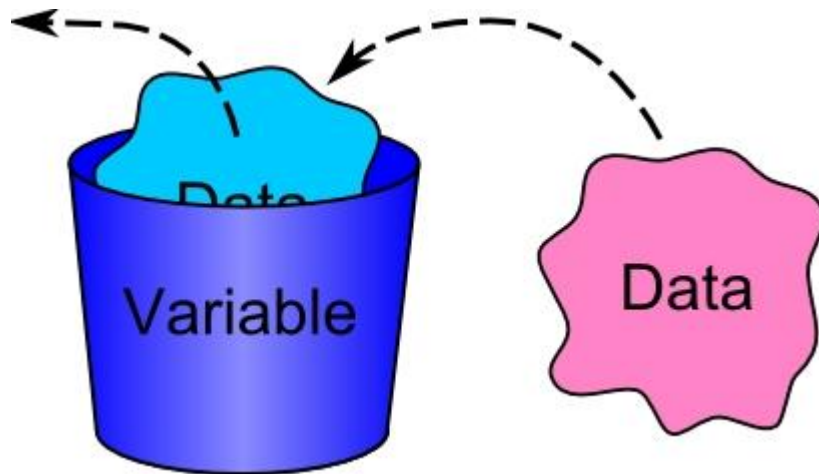
## camelCase

=> firstName;

## Tidak diawali simbol/angka

(x) 1Input;

(x) &kata;



# Variabel & Tipe Data

## Declaration

```
var bucket;  
var inputNumber;
```

## Assignments

```
bucket = 'hello'  
inputNumber = 10
```

## Declaration + Assignment

```
var bucket = 'hello';  
var inputNumber = 10;
```

Penulisan di JavaScript

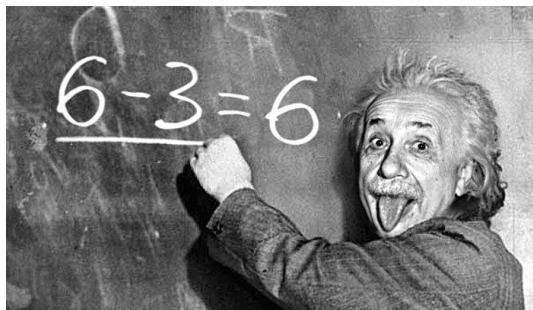
# Variabel & Tipe Data

Declaration + Assignment

```
var myNumber ; // undefined  
myNumber = 2; // Number (2)
```

Re-assignment

```
myNumber = myNumber + 1;
```



Penulisan di JavaScript

# Variabel & Tipe Data

No duplicate Declaration!

```
var myNumber;
```

```
var myNumber = 2;
```

Penulisan di JavaScript

# Operasi Matematik

Operator Matematik

+ - \* / %

Perhatikan penggunaan operator matematik untuk non-number !

Try 5 + 'hello' or true - 8!

=> 5 + 'hello' = '5hello'

=> true - 8 = NaN



# Operasi Perbandingan

- `===` (nilai dan tipe data sama) → `5 === 5`
- `==` (nilai sama) → `'5' == 5`
- `!=` (tidak bernilai sama) → `'Hello' != true`
- `!==` (tidak bernilai & bertipe data sama) → `'false' !== false`
- `>` (lebih dari) → `10 > 0`
- `<` (kurang dari) → `3 < 5`
- `>=` (lebih dari sama dengan) → `4 >= 4`
- `<=` (kurang dari sama dengan) → `2 <= 2`



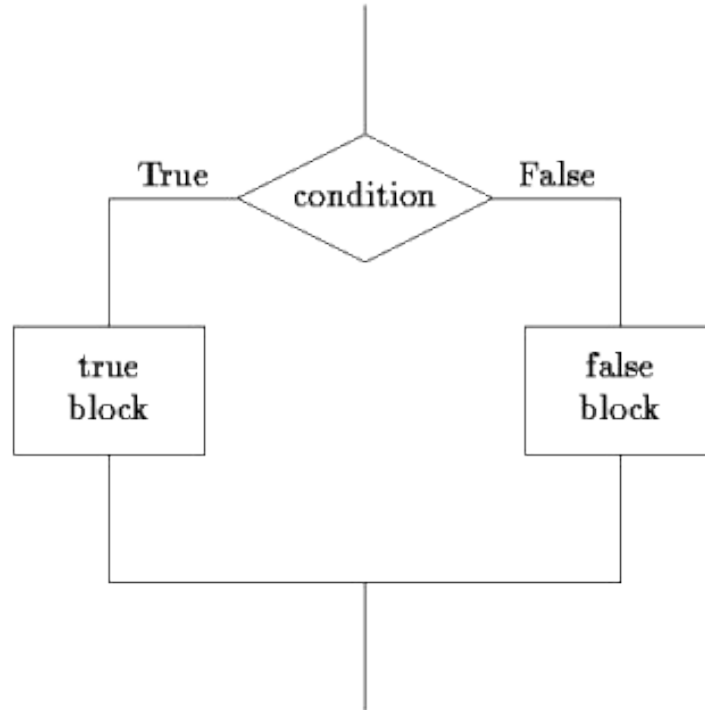
# Javascript<sup>+</sup> Conditional

# Point of Discussion

- Apa itu Kondisional ?
- kondisional dalam javascript (if-else if-else)
- logical operator
- Sifat Truthy atau Falsy dari nilai variable

Apa itu

# Kondisional



# Kondisional

## IF - ELSE IF - ELSE

```
if ( kondisi ) {  
    statement..  
}
```

```
if ( kondisi1 ) {  
    statement..  
} else {  
    statement..  
}
```

```
if ( kondisi1 ) {  
    statement..  
} else if ( kondisi2 ) {  
    statement..  
} else {  
    statement..  
}
```

Contoh Penulisan di JavaScript

# Kondisional

Jika kondisi saya lapar maka saya akan makan

```
var kondisi = 'lapar'
```

```
if ( kondisi === 'lapar' ) {  
    console.log('makan')  
}
```

Contoh Penulisan di JavaScript

# Kondisional

Jika kondisi saya lapar maka saya akan makan, jika tidak maka saya akan minum

```
var kondisi = 'lapar'
```

```
if ( kondisi === 'lapar' ) {  
    console.log('makan')  
} else {  
    console.log('minum')  
}
```

Contoh Penulisan di JavaScript

# Kondisional

Jika kondisi saya lapar maka saya akan makan, jika kondisi saya haus maka saya akan minum, jika tidak keduanya maka saya akan tidur saja

```
var kondisi = 'lapar'

if ( kondisi === 'lapar' ) {
  console.log('makan')
} else if ( kondisi === 'haus' ) {
  console.log('minum')
} else {
  console.log('tidur')
}
```



Penulisan di JavaScript

# Operasi Logika

**&&** (AND / kedua kondisi harus true)

**||** (OR / salah satu kondisi harus true)

# Operasi Logika

&& ( DAN / AND )

Kondisi 1	Kondisi 2	RESULT
True	True	?
True	False	?
False	True	?
False	False	?

|| ( ATAU / OR )

Kondisi 1	Kondisi 2	RESULT
True	True	?
True	False	?
False	True	?
False	False	?

## Pemahaman

# Operasi Logika

&& ( DAN / AND )

Kondisi 1	Kondisi 2	RESULT
True	True	True
True	False	False
False	True	False
False	False	False

|| ( ATAU / OR )

Kondisi 1	Kondisi 2	RESULT
True	True	True
True	False	True
False	True	True
False	False	False

## Contoh Penulisan di JavaScript

# Operasi Logika

<code>console.log( true &amp;&amp; true )</code>	<code>=&gt; true</code>
<code>console.log( true &amp;&amp; false )</code>	<code>=&gt; false</code>
<code>console.log( false &amp;&amp; true )</code>	<code>=&gt; false</code>
<code>console.log( false &amp;&amp; false )</code>	<code>=&gt; false</code>

## Mengecek apakah sebuah nilai ada di dalam range

```
var input = 15
```

<code>console.log( input &gt; 10 &amp;&amp; input &lt; 20 )</code>	<code>=&gt; true</code>
<code>console.log( input &gt; 10 &amp;&amp; input &gt; 20 )</code>	<code>=&gt; false</code>

# Sifat suatu Variable

## Truthy dan Falsy sebuah Nilai


Sebuah nilai memiliki sifat booleannya jika menjadi suatu kondisi.

Truthy - should be true

Ex: semua nilai yang tidak bersifat falsy (bernilai)

Falsy - should be false

Ex: false, 0, "" (empty string), undefined, NaN, Null, false



# Javascript + Iteration / Looping

# Point of Discussion

- Apa Iterasi?
- Kenapa Iterasi?
- Kapan menggunakan Iterasi?
- Penulisan Iterasi ( While loop & For loop)

## Kenapa Iterasi?

- Iterasi → Perulangan

Dengan iterasi, dapat membuat **code** semakin dinamis. Sebagai programmer harus selalu menerapkan konsep **DRY** (*Don't Repeat Yourself*).

Konsep **DRY** ini berfungsi untuk membuat **code** lebih efisien, sehingga proses yang dilakukan oleh sistem akan lebih cepat dibanding tidak menggunakan konsep **DRY** ini.



# Kapan harus menggunakan Iterasi?

Gunakan Iterasi saat membutuhkan suatu proses atau metode yang digunakan secara berulang-ulang (pengulangan) dalam menyelesaikan suatu permasalahan.

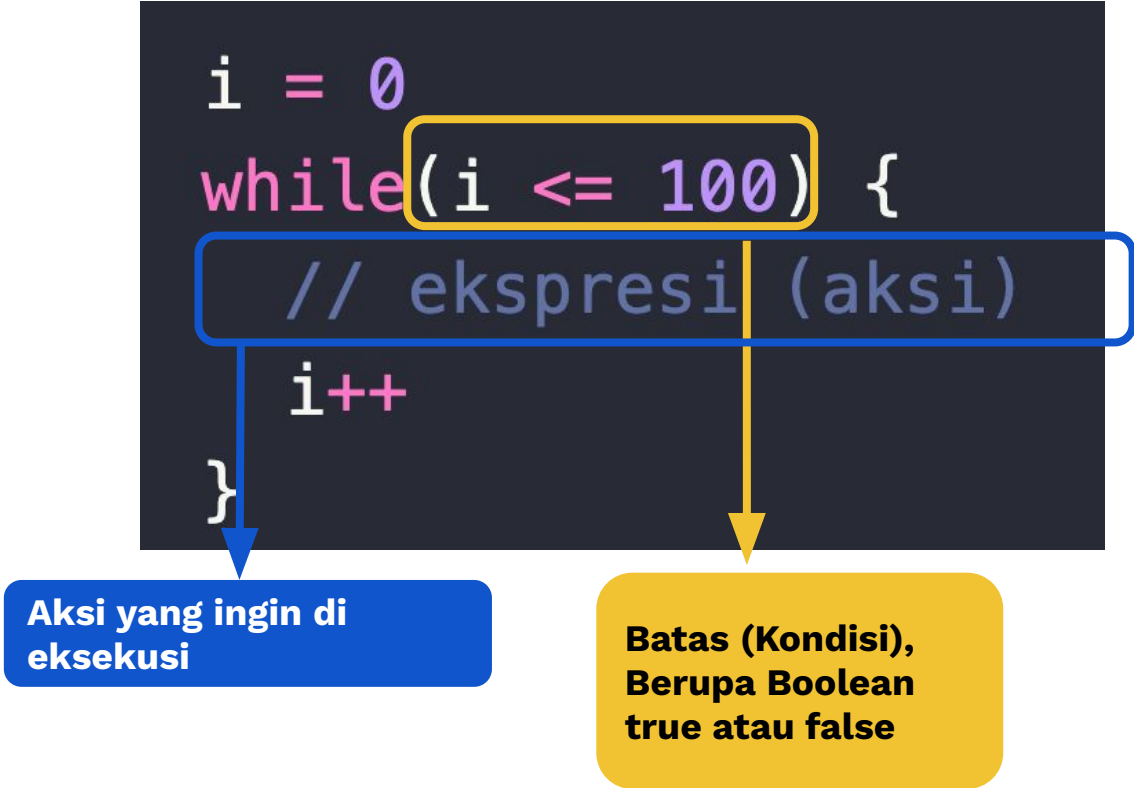
# Penulisan Iterasi

```
while(i <= 100) {  
    // ekspresi (aksi)  
}
```

```
for (var i = 1; i <= 100; i++) {  
    // ekspresi (aksi)  
}
```

# ITERASI WHILE

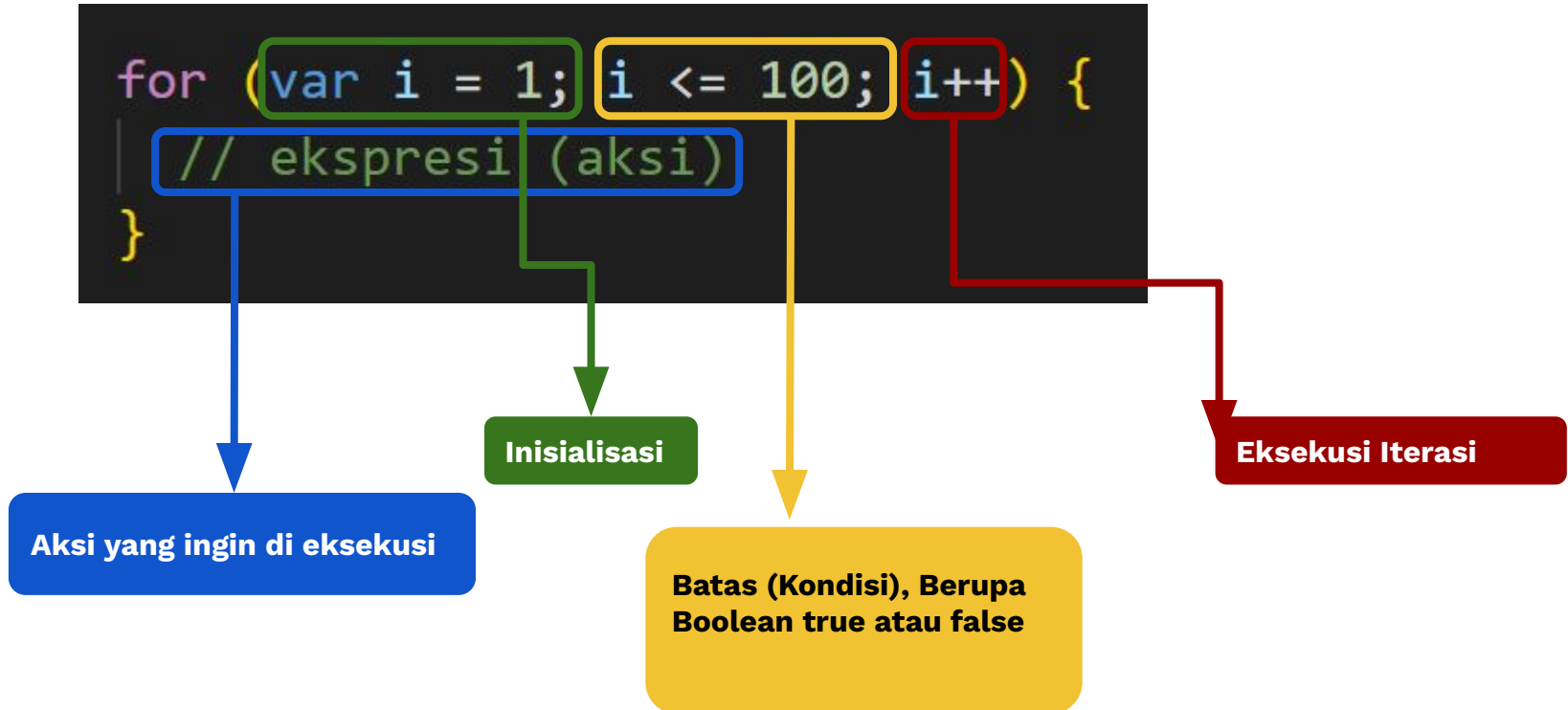
```
i = 0  
while (i <= 100) {  
    // ekspresi (aksi)  
    i++  
}
```



Aksi yang ingin di  
eksekusi

Batas (Kondisi),  
Berupa Boolean  
true atau false


# ITERASI FOR




# Waspadai Forever Loop!

Dengan sengaja atau sengaja, kode kamu mungkin dapat menghasilkan forever looping, atau looping yang tidak akan pernah berhenti. Bila ini terjadi, segera periksa statement kondisi kamu.

```
i = 0
while(i <= 100) {
  console.log("iterasi ke-", i)
}
```



Iterasi tidak akan berhenti, karena nilai i tidak pernah bertambah!



# JavaScript<sup>+</sup> Array

# Point of Discussion

- Apa itu Array ?
- Cara Membuat Array
- Cara Mengakses item dalam Array
- Method-method dalam Array



# Apa Itu Array

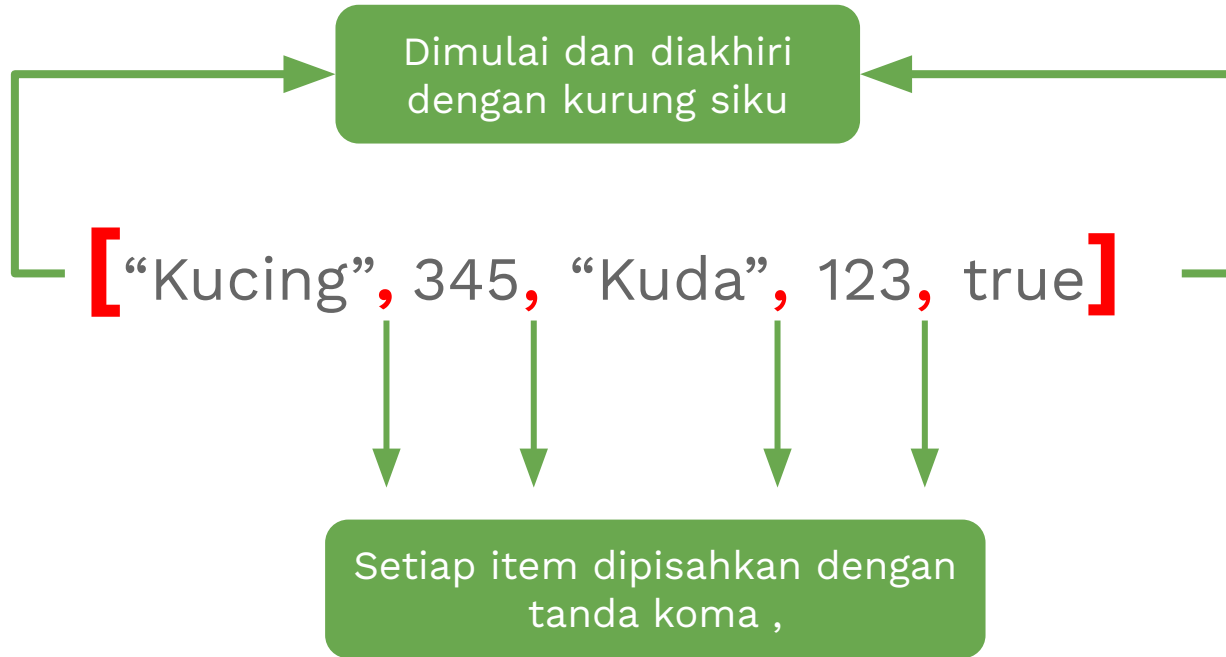
Mirip dengan rak buku yang dapat menampung banyak buku,

Array merupakan kumpulan data, berisi banyak nilai (item).

Dalam Javascript, array dapat diisi dengan berbagai macam tipe data.



# Cara Membuat Array



# Index item dalam Array

[“Kucing”, 345, “Kuda”, 123, true]

Index item =>

0

1

2

3

4

Mengetahui index dalam item-array sangat penting, agar bisa mengakses, maupun mengganti nilai dalam suatu array.

Array index dalam javascript selalu dimulai dengan angka 0

# Akses item dalam Array

```
var iniArray = ["Kucing", 345, "Kuda", 123, true]
```

Index item =>

0

1

2

3

4

**Cara Akses:**

```
var_array[index_item]
```

```
// akses item "Kucing"  
iniArray[0]
```

```
// akses item 345  
iniArray[1]
```

```
// akses item true  
iniArray[4]
```

# Ubah item dalam Array

var **iniArray** = ["Kucing", 345, "Kuda", 123, true]

0 1 2 3 4

```
// ubah item "Kucing" menjadi "Kelinci"
```

```
iniArray[0] = "Kelinci"
```

```
// ubah item 123 menjadi 7890
```

```
iniArray[3] = 7890
```

```
console.log(iniArray) // ["Kelinci", 345, "Kuda", 7890, true]
```

# Built- in Method dalam Array

**push:** menambah 1 nilai ke array ke index paling belakang

**pop:** menghapus 1 nilai dari array index paling belakang

**unshift:** menambah 1 nilai ke array index paling depan (index 0)

**shift:** menghapus 1 nilai dari array index paling depan (index 0)

**join:** menggabungkan seluruh elemen array menjadi sebuah string dan mengambil parameter sebagai simbol penyambung antar elemen

**sort:** mengurutkan elemen di dalam array sesuai alphabet

**slice:** mengambil beberapa lapis data

**splice:** mengubah nilai array dengan menghapus dan/atau menambah nilai baru ke array

**split:** memecah string dan mengembalikan array sesuai dengan separator / pemisah yang didefinisikan

# Push

Push adalah fungsi array yang akan menambahkan nilai di belakang nilai terakhir di array.

```
1 var arr = [0, 1, 2, 3];  
2 arr.push(4);  
3 console.log(arr); // 0, 1, 2, 3, 4|
```

# Pop

Pop adalah fungsi array yang akan menghapus nilai paling belakang / terakhir dari array.

```
1 var arr = [0, 1, 2, 3];  
2 arr.pop();  
3 console.log(arr); // 0, 1, 2|
```

# Unshift

Unshift adalah fungsi array yang akan menambah nilai ke depan array (ke index 0), dan menggeser seluruh isi array ke belakang.

```
1 var arr = [0, 1, 2];  
2 arr.unshift(3);  
3 console.log(arr); // 3, 0, 1, 2|
```

# Shift

Shift adalah fungsi array yang akan menghapus nilai paling depan dari array. Fungsi shift tidak menerima parameter apapun.

```
1 var arr = [3, 0, 1, 2];  
2 arr.shift();  
3 console.log(arr); // 0, 1, 2|
```

# Sort

Sort adalah fungsi array yang akan mengurutkan nilai dari array. Perlu diperhatikan, sort mengurutkan otomatis secara ascending dan diurutkan berdasarkan unicode dari karakter. Kamu akan menemukan masalah ini saat mengurutkan angka.

```
1 var arr = [3, 5, 7, 1, 2];
2 arr.sort();
3 console.log(arr); // 1, 2, 3, 5, 7
4
5 var arrChar = ['Tono', 'Budi', 'Charlie', 'Ahmad'];
6 arrChar.sort();
7 console.log(arrChar); // 'Ahmad', 'Budi', 'Charlie',
  'Tono'
```



# Sort pada angka

```
1 var arr = [1, 2, 15];  
2 arr.sort();  
3 console.log(arr); // 1, 15, 2
```

Tidak berhasil  
tersortir!

Hal ini karena JavaScript melakukan sort secara unicode atau sebagai karakter, dan karena 15 diawali dengan karakter '1', maka akan dianggap lebih awal daripada karakter '2'.

## Untuk menyelesaikan masalah ini,

kamu perlu menambahkan satu parameter berupa fungsi pembandingan.

Perlu dibuat satu fungsi yang menilai apakah nilai pertama lebih kecil dari nilai kedua.

```
1 var arr = [1, 2, 15];  
  
2 arr.sort(function(value1, value2) { return value1 >  
  value2 });  
  
3 console.log(arr); // 1, 2, 15
```

# Slice

Slice adalah fungsi array yang akan mengambil irisan atau porsi dari sebuah array. Fungsi slice menerima satu atau dua parameter.

**Parameter pertama** adalah index irisan diambil,

**parameter kedua** adalah index irisan terakhir diambil. Jika parameter kedua tidak didefinisikan, irisan akan diambil hingga akhir dari array.

`array.splice(index, howmany, item1, ....., itemX)`

```
1 var arr = ["buku", "laptop", "komputer"];

2 arr.splice(2, 0, "televisi"); // Menghapus 0 nilai
  dari index 2, dan menambah 1 nilai yaitu "televisi" pada
  index 2.

3 console.log(arr); // ["buku", "laptop", "televisi",
  "komputer"]

4

5 arr.splice(0, 2); // Menghapus 2 nilai dari index 0

6 console.log(arr); // ["televisi", "komputer"]

7

8 arr.splice(0, 1, "majalah", "koran"); // Menghapus 1
  nilai dari index 0, dan menambah 2 nilai yaitu "majalah"
  dan "koran"

9 console.log(arr); // [majalah", "koran", "komputer"]
```

# Split

Split adalah fungsi yang **memecah string** dan **mengembalikan nilai berupa array** sesuai dengan separator atau pemisah tertentu yang didefinisikan. Fungsi split menerima satu parameter, yaitu karakter apa yang akan menjadi pemisah/separator.

*string.split(separator, limit)*

```
1 var kalimat = "saya adalah full-stack javascript programmer!";  
2 var kata = kalimat.split(" "); // kalimat dipecah dengan separator berupa spasi.  
3 console.log(kata); // ["saya", "adalah", "full-stack", "javascript", "programmer!"]
```