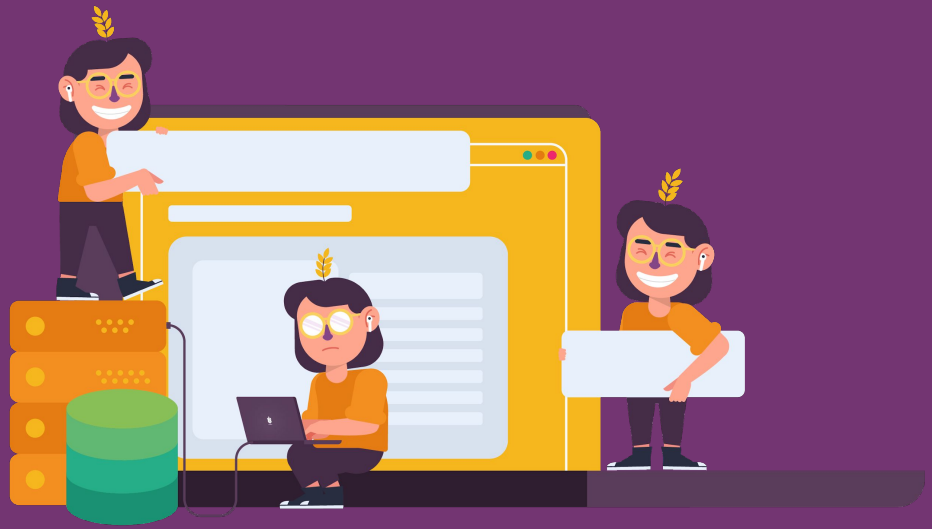
Operator & Expressions

Silver - Chapter 2 - Topic 2

Selamat datang di Chapter 2 Topic 2 online course Full-Stack Web dari Binar Academy!

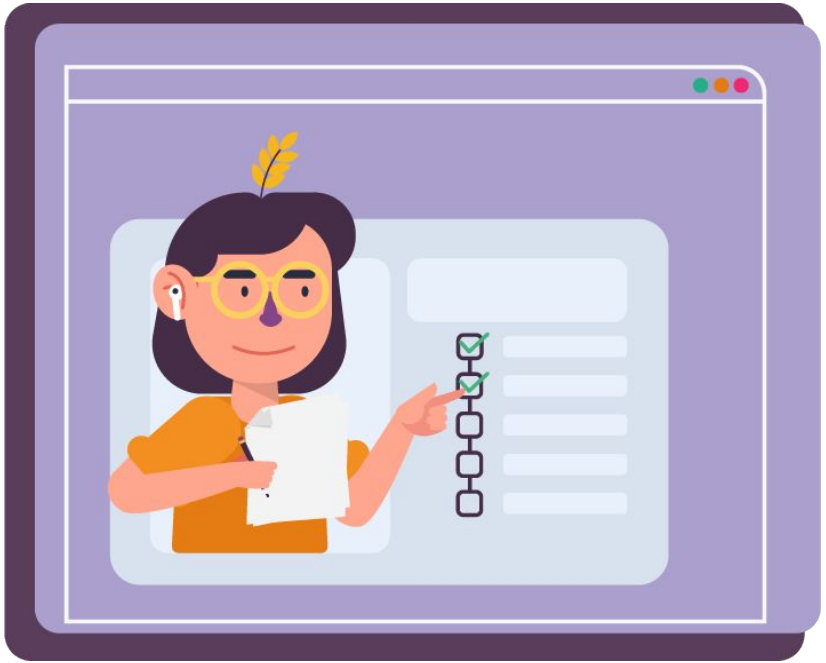


Chapter 2 ini bakalan ngajak kamu buat lebih paham logika bahasa pemrograman Javascript untuk membuat halaman web. Mulai dari penjelasan terkait struktur data, penggunaan operator, pembuatan expression sampai logika algoritma yang ada di dalamnya.

Pada topik kedua, kita bakal bahas tentang **Operator & Expression pada Javascript.** Cus langsung aja~

Detailnya, kita bakal bahas hal-hal berikut ini :

- Macam-macam operator di Javascript
- Logical operator pada Javascipt



Waktu sekolah, kita mengenal banyak operator bilangan Matematika, mulai dari penjumlahan, pembagian, perkalian, pengurangan, dan lain-lain.

Nah, kali ini, kita bakal bahas soal operator yang nggak kita dapet di bangku sekolah.

Yap~ **operator dalam pemrograman!**

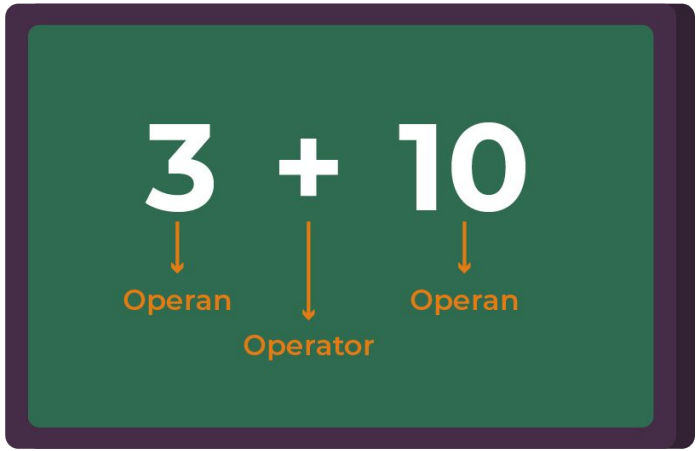
Sebelum masuk pembahasan, kita bahas tentang operasi dan operan yaa~

Operator dan Operan berfungsi untuk **mengolah data di dalam Javascript dengan membuat sebuah operasi**. Nah, operasi ini dapat terjadi apabila ada yang namanya operator dan operan.

Biar nggak bingung, kita pecah satu-satu ya penjelasannya!

Operator itu adalah **penentu jalannya operasi**. Misalnya kita ingin menjumlahkan dua variabel, maka kita bisa menggunakan operator plus.

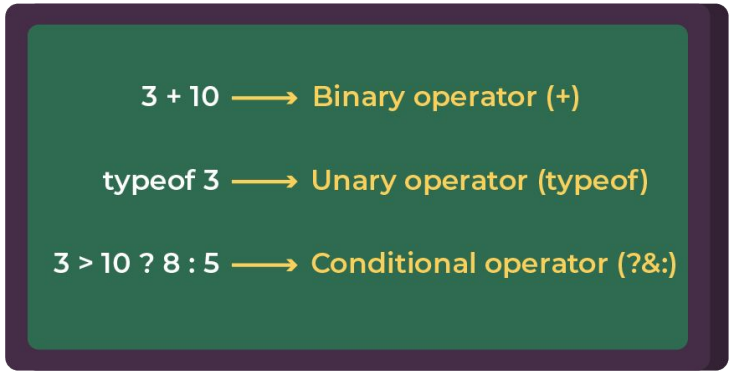
Operan sendiri adalah **objek yang dioperasikan**, sebagai contoh, kita punya 2 variabel, a dan b, yang mana akan kita jumlahkan dengan operator plus a dan b inilah yang kita sebut sebagai operan.



Setelah tau definisinya, lanjut kita bahas operator di dalam Javascript

Jadi di dalam Javascript, operator tuh dibedakan menjadi 3 jenis yaitu: **binary operator**, **unary operator**, dan **conditional (ternary) operator**.

- Binary operator adalah **operator yang membutuhkan 2 operan.**
- Unary operator adalah **operator yang hanya membutuhkan 1 operan saja.**
- Conditional operator adalah **operator yang membutuhkan 3 operan agar berjalan**, operator ini digunakan untuk melakukan pemilihan antara 2 data berdasarkan kondisi.



Nah, Sekarang kamu udah tau kan jenis-jenis operator di Javascript.

Langsung aja yang pertama binary operator~ Operator ini memiliki jenis Assignment, Comparison, Arithmetic, Logical, String, dan Relational Operator.

Walaupun banyak, kita akan belajar pelan-pelan kok. Yuk!

Assignment Operator

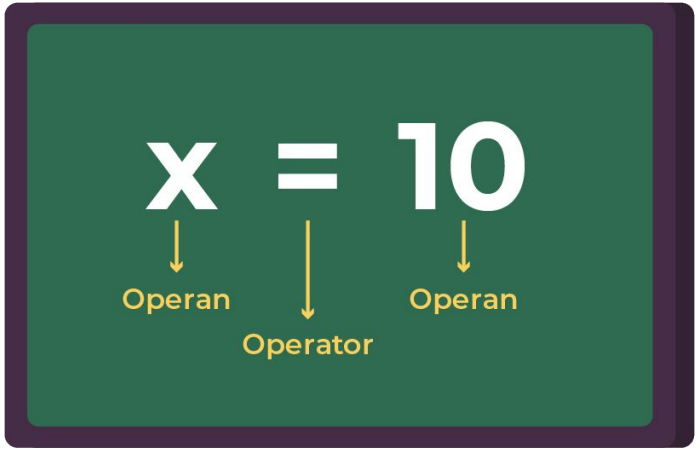
Sebuah operator yang digunakan untuk **memasukkan nilai pada sebuah variabel**.

Nah, assignment operator kan ada dua operan nih, operan sebelah kiri dan operan sebelah kanan, nah si operator ini akan memasukkan **operan sebelah kanan sebagai nilai dari operan sebelah kiri**.

Operator ini identik dengan simbol sama dengan “=” yaaa~

Nah, di dalam Javascript assignment operator ini ada banyak variannya, yaitu :

- **Assignment**
- **Addition Assignment**
- **Logical AND Assignment**



Untuk elaborasi lebih banyak lagi kamu bisa cek link ini ya !
MDN - Operator & Expression

Assignment Operator

Assignment operator ini bisa dikombinasikan dengan operator lain, sebagai contoh operator plus. Hal inilah yang kita sebut sebagai additional assignment.

Syarat dari assignment ini adalah, **variabel yang menjadi operan sebelah kiri wajib dideklarasikan** terlebih dahulu. Karena variabel ini akan digunakan dalam operasi turunan, yaitu operasi penjumlahan dari nilai x saat ini dengan operan sebelah kanan.

```
// Deklarasi variabel x
let x = 10; // Memasukkan nilai 10 ke dalam variabel x

// Additional Assignment
x += 80 // x = x + 80; x = 10 + 80;

console.log("x:", x)
// Output: 90;
```

Oh iya, kamu bisa menggunakan operator assignment untuk memasukkan nilai kedalam array maupun object juga lho.

Inget kan topik 1 sebelumnya bahas pengertian array itu **menyimpan kumpulan dari data**, sedangkan object menggunakan **key sebagai idenfitier-nya.**

Hal ini berguna banget ketika kamu lagi mengolah data array atau object lho.

```
// Object
let person = { name: "Sabrina", age: 21 };

// Array
let fruits = ["Mangga", "Apel", "Jeruk"];

// Assignment terhadap atribut object
person.name = "Agus";
console.log("Person:", person);
// Output: { name: "Agus", age: 21 }

// Assignment terhadap item di dalam array
fruits[0] = "Semangka";
console.log("Fruits:", fruits);
// Output: ["Semangka", "Apel", "Jeruk"]
```

Dan juga kita bisa gunakan assignment operator untuk **melakukan destructuring dari array dan object,** hal ini akan mempermudah kita dalam mengambil data di dalam array, atau object.

Langsung cek gambar di samping ya~

```
// Object
let person = { name: "Sabrina", age: 21 };

// Array
let fruits = ["Mangga", "Apel", "Jeruk"];

let { name } = person;
let [ firstFruit ] = fruits;

console.log("Name:", name);
// Output: Sabrina

console.log("First Fruit:", firstFruit);
// Output: Mangga
```

Comparison Operator

Sebuah binary operator yang berguna untuk **mengkomparasikan operannya** dan hasil dari operasi ini akan memiliki nilai boolean, yang menandakan apakah komparasi itu benar atau tidak.

Perhatikan contoh kode di samping yaa~

```
const x = 10;
const y = 100;

const isXBiggerThanY = x > 10;
console.log(isXBiggerThanY);
// Output: false
```

Banyak sekali operator yang dapat digunakan sebagai, comparison operator. Tabel dibawahini adalah daftar sekaligus contohnya ya!

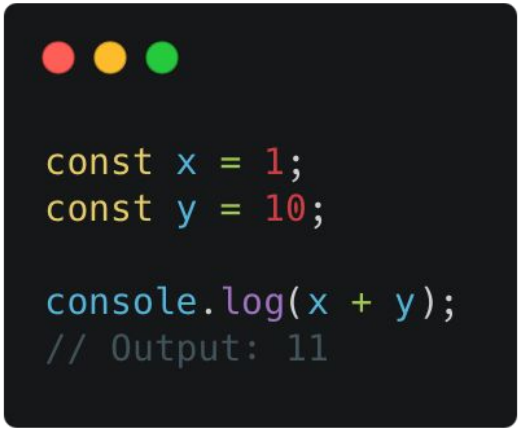
Operator	Deskripsi	Contoh

Equal (==)	Bernilai **true** apabila operannya sama	`3 == var1`
		`"3" == var1`
		`3 == '3'`

Not equal (!=)	Bernilai **true** apabila operannya tidak sama	`var1 != 4` `var2 != "3"`
Strict equal (===)	Bernilai **true** apabila operannya sama dan tipe datanya sama	`3 === var1`
Strict not equal (!==)	Bernilai **true** apabila operannya memiliki tipe data yang sama dan nilainya sama.	`var1 !== "3"` `3 !== '3'`
Greater than (>)	Bernilai **true** apabila operan sebelah kiri lebih besar dari operan sebelah kanan	`var2 > var1` `"12" > 2`
Greater than or equal (>=)	Bernilai **true** apabila operan sebelah kiri lebih besar atau sama dengan operan sebelah kanan.	`var2 >= var1` `var1 >= 3`

Arithmetic Operator

Sebuah **operator yang mana operannya bernilai numerik**. Jadi, ya operannya itu nanti bakal mengalami operasi aritmatik seperti matematika pada umumnya, makanya operannya diharuskan numerik.



```
const x = 1;
const y = 10;

console.log(x + y);
// Output: 11
```

Sebenernya, kalo kamu menaruh String sebagai operan dalam arithmetic operator itu pun masih bisa dan hasilnya akan sama, tapi hal-hal kayak gitu ga lazim untuk dilakukan. Karena akan menyebabkan kebingungan, jadi kita sebagai developer selalu dianjurin untuk pake yang pasti-pasti aja ya~

```
const x = 1;
const y = 10;
const z = "100"

console.log((x + y) * z);
// Output: 11
```

Arithmetic Operator

Arithmetic Operator ini ada banyak jenisnya, hampir semua hal yang menjadi notasi matematika itu ada di dalam arithmetic operator Javascript.

Operator	Deskripsi	Contoh
Remainder (%)	Operator Binary. Mengembalikan sisa bilangan bulat dari pembagian dua operan.	`12 % 5; // 2`

Increment (++)	Operator Unary. Menambahkan satu ke operannya. Jika digunakan sebagai operator awalan (++x), maka artinya mengembalikan nilai operan setelah menambahkan satu; jika digunakan sebagai operator postfix (x++), maka artinya mengembalikan nilai operan sebelum menambahkannya.	`let x = 3;`
		`x++;`
		`console.log(x) // 4`

Decrement (--)	Operator Unary. Kurangi satu dari operannya. Nilai hasil adalah sejalan dengan kenaikan operator.	`let x = 3;`
		`x--;`
		`console.log(x) // 2`

Unary negation (-)	Operator Unary. Menghasilkan nilai negasi dari operannya.	`let x = 3;` `console.log(-x) // -3`
Unary plus (+)	Operator Unary. Melakukan konversi operan menjadi angka (jika belum)	`const x = "3";` `console.log(+x); // 3`
Exponentiation operator (**)	Menghitung basis ke pangkat eksponen	`const x = 3;` `console.log(x**2) // 9`

Logical Operator

Sebuah operator yang **menentukan operan mana yang akan menjadi hasil dari operasi tersebut**.

Operator	Cara pakai	Deskripsi
Logical AND (&&)	expr1 && expr2	Bernilai **false** apabila salah satu dari operan bernilai false. Bernilai **expr2** apabila **expr1** bernilai **truthy**
Logical OR (||)	expr1 || expr2	Bernilai **expr2** apabila **expr1** bernilai **falsy**
Logical NOT (!)	!expr	Bernilai **true** apabila expr bernilai **falsy** Bernilai **false** apabila expr bernilai **truthy**

```
console.log(1 && 10); // Output: 10
console.log(0 || 100); // Output: 100
console.log(true && false); // Output: false
console.log(true && false || "Ya elah"); // Output: Ya elah

const x = 10;
const y = 100;

if (x * y % 100 === 0) console.log("OK!");
// Output: OK!
```

Truthy (MDN - Truthy)

Adalah **nilai yang dianggap true** di dalam konteks boolean. Sebagai contoh adalah string yang tidak kosong, angka yang tidak nol, tidak null, tidak dan undefined.

Falsy (MDN - Falsy)

Kebalikan dari truthy, adalah **nilai yang dianggap false** di dalam konteks boolean. Sebagai contoh adalah string yang kosong, angka yang nol, null, dan undefined.

String Operator

Nah, ternyata ga cuma nilai numerik aja nih yang bisa kita tambah-tambahin dalam pake operator

String juga bereaksi dengan beberapa binary operator. Yuk kita cari tau bagaimana string bereaksi dengan operator. Skuy~

Cuma... hanya ada **2 operator yang bisa digunakan buat manipulasi string**, yaitu Plus (+), sama Addition Assigment Operator (+=).

Kamu bisa elaborasi lagi di link ini ya ~

MDN - String Operators

```
let name = "Sabrina";
let greetingMessage = "Hi, ";

console.log(greetingMessage + name);
// Output: Hi, Sabrina

greetingMessage += name;
greetingMessage += "!";
console.log(greetingMessage);
// Output: Hi, Sabrina!
```

Relational Operator (MDN - Relational Operator)

Sebuah **operator yang digunakan untuk melakukan komparasi terhadap operan-operannya**. Sebetulnya mirip dengan comparison operator, bedanya relational operator ini lebih berfokus kepada struktur datanya.

```
const stockFruits = ["Mangga", "Jeruk", "Apel"];

const isMyFavouriteFruitInStocks = "Semangka" in stockFruits;
console.log(isMyFavouriteFruitInStocks);
// Output: false
```

```
const stockFruits = ["Mangga", "Jeruk", "Apel"];
const person = {
  name: "Sabrina",
  age: 21,
  ktpNumber: "123412341234"
};

const isMyFavouriteFruitInStocks = "Semangka" in stockFruits;
console.log(isMyFavouriteFruitInStocks);
// Output: false

const doesPersonHaveKTP = "ktpNumber" in person;
console.log(person.name, doesPersonHaveKTP ? "has KTP" : "does not have KTP");
// Output: Sabrina has KTP

console.log(person instanceof Date);
// Output: false

const date = new Date(); // Hari ini
console.log(date instanceof Date);
// Output: true
```

Dalam Relational Operator ada 2 jenis relational operator yang wajib kamu ketahui.

in

Sebuah relational operator yang digunakan untuk **mengecek keberadaan sebuah data dalam sebuah data yang berupa koleksi (Array atau Object)**.

instanceof

Sebuah relational operator yang digunakan untuk **mengecek apakah suatu data merupakan instansi dari tipe data tertentu**.

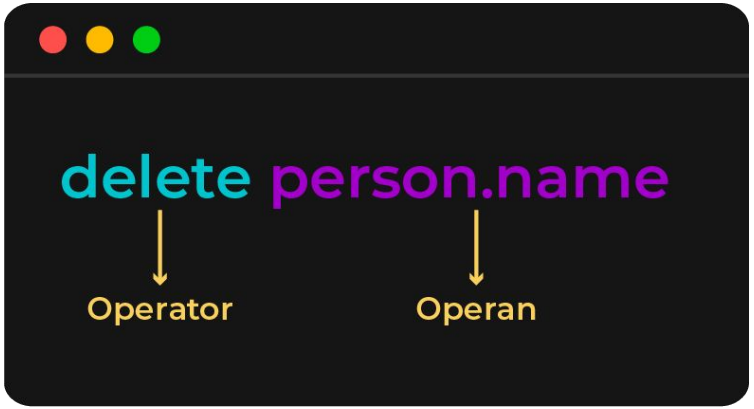
Sebelum lanjut ke pembahasan selanjutnya, kamu udah mulai paham kan tentang pemahaman dan jenis–jenis Binary Operator?

Nah, selanjutnya masih ada **Unary Operator** dan **Conditional Operator** nih. Yuk kita intip!

Unary Operator (MDN - Unary Operator)

Sebuah **operator yang operannya cuma satu.**

Gambar di samping juga jadi salah satu contoh dari unary operator. Dimana operator yang hanya meminta 1 operan saja di dalam sebuah operasi dan ekspresi.



```
const x = 10;
const person = {
    name: "Sabrina",
    age: 21
}

delete person.name;

console.log(person);
// Output: { age: 21 }

console.log(typeof person);
// Output: object

console.log(typeof x);
// Output: number
```

Unary operator ini ada banyak jenisnya, tapi yang paling umum dipakai adalah **delete** dan **typeof**.

Yang mana **delete** ini digunakan untuk menghapus atribut dari sebuah object.

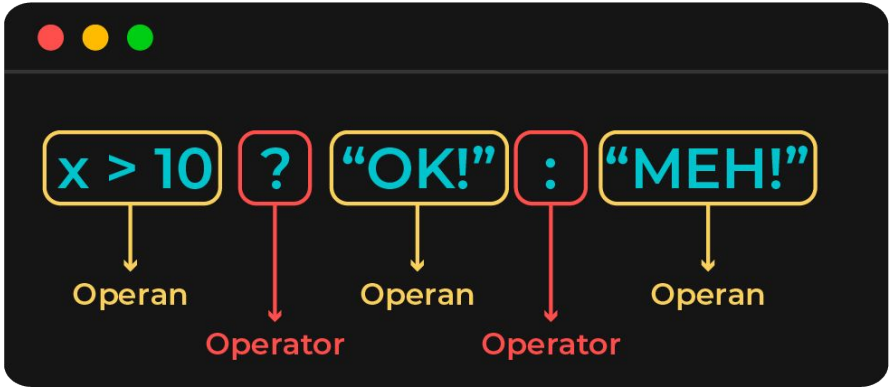
Sedangkan **typeof** ini digunakan untuk mencari tau tipe data dari sebuah variabel atau literal.

Conditional (Ternary) Operator

(MDN - Conditional Operator)

Sebuah **operator yang membutuhkan 3 operan**. Dimana operan paling kiri akan menjadi penentu operan mana yang akan menjadi hasil dari ekspresi atau operasi tersebut.

Kalau berdasarkan contoh, maka operan paling kiri (x > 10) akan menentukan apakah operan yang berada di tengah atau operan yang paling kanan, yang akan menjadi hasil dari operasi tersebut



Conditional (Ternary) Operator

```
const isTodayRaining = true;
const thingsToBring = isTodayRaining ? "Umbrella" : "Aviator";
// Output: Umbrella

const x = 10;
const y = 100;

const luckyNumber = (x === y) ? 72 : 32;
// Output: 32
```

Nah si ternary operator ini biasa dipake untuk **assignment singkat, tanpa perlu pake if else statement**.

Di bahasa lainnya kamu akan menemukan hal yang mirip sama operator ini dengan istilah lain, contohnya : Elvis Operator.

Saatnya kita

Quiz!

BINAR

1. **Di dalam sebuah ekspresi atau operasi, ada dua hal yang dibutuhkan didalamnya. yaitu**

A. Operan dan Boolean

B. Operan dan Operator

C. Numerik Value dan Operator

BINAR

1. **Di dalam sebuah ekspresi atau operasi, ada dua hal yang dibutuhkan didalamnya. yaitu**

A. Operan dan Boolean

B. Operan dan Operator

C. Numerik Value dan Operator

Numerik value dan boolean hanyalah tipe data dari sebuah operan, tapi inti dari sebuah operasi adalah operannya, data yang ingin kita olah.

BINAR

2. Jika `x = 1` dan `y = 10` Maka output dari kode operasi `y >= x ? x : y;` adalah

A. 1

B. 10

C. true

BINAR

2. Jika `x = 1` dan `y = 10` Maka output dari kode operasi `y >= x ? x : y;` adalah

A. 1

B. 10

C. true

Operasi tersebut adalah sebuah operasi yang menggunakan ternary operator, yang mana jika nilai `y >= x` adalah `true`, maka operan yang paling tengah akan menjadi hasilnya.

BINAR

3. Hasil dari operasi `1 > 2 && (10 + 20)` adalah

A. 30

B. true

C. false

BINAR

3. Hasil dari operasi `1 > 2 && (10 + 20)` adalah

A. 30

B. true

C. false

Karena `1 > 2` bernilai false dan pada operasi tersebut menggunakan Logical AND, maka dari itu nilai dari operasi tersebut adalah operan paling kiri, yang mana bernilai false.

BINAR

4. Operator dibawah ini yang bukan digunakan untuk mengatur alur dan keputusan adalah

A. Arithmetic Operator

B. Ternary Operator

C. Logical Operator

BINAR

4. Operator dibawah ini yang bukan digunakan untuk mengatur alur dan keputusan adalah

A. **Arithmetic Operator**

B. Ternary Operator

C. Logical Operator

Arithmetic Operator digunakan dalam operasi aritmatik yang berhubungan dengan angka, maka dari itu tidak ada kaitannya dengan pengaturan alur dan keputusan.

BINAR

5. Operator apa yang dapat kita gunakan untuk menggantikan emoji 😂 agar operasi berikut bernilai true?

```
x = 1; y = 10; y ≤ x 😂 true
```

A. Logical AND (&&)

B. Logical OR (||)

C. Ternary Operator (? :)

BINAR

5. Kira-kira operator apa yang dapat kita gunakan untuk menggantikan emoji 😂 agar operasi berikut bernilai true?

```
x = 1; y = 10; y ≤ x 😂 true
```

A. Logical AND (&&)

B. **Logical OR (||)**

C. Ternary Operator (? :)

Logical OR akan selalu mengembalikan nilai dari operan paling kanan apabila operan paling kiri bernilai falsy.

BINAR

6. Apa output dari operasi di bawah ini?

```
x = 1; y = 10; x > y ? (x == 1 && "Yes") : (y != 10 || "No")
```

A. Yes

B. No

C. Tidak ada output

BINAR

6. Apa output dari operasi di bawah ini?

```
x = 1; y = 10; x > y ? (x == 1 && "Yes") : (y != 10 || "No")
```

A. Yes

B. No

C. Tidak ada output

Karena x tidak lebih dari y, maka dari itu operan paling kanan lah yang akan menjadi hasil dari operasi tersebut. Dan karena operan dari kanan terdapat logical operator OR, maka dari itu, jika y tidak sama dengan 10, maka hasil operasi tersebut adalah No

BINAR

7. Apa output dari operasi di bawah ini?

```
x = true; y = false; x && y || (10 + 1) || "Hello World"
```

A. true

B. 11

C. Hello World

BINAR

7. Apa output dari operasi di bawah ini?

```
x = true; y = false; x && y || (10 + 1) || "Hello World"
```

A. true

B. 11

C. Hello World

Karena x adalah true, maka kita akan ambil operan kedua dari operasi pertama sebagai hasil, dan karena hasil dari operasi pertama menjadi operan di operasi kedua yang mana merupakan logical OR, maka dari itu bila operasi pertama bernilai falsy, kita ambil operan kedua dari operasi kedua, yaitu (10 + 1). Dan karena (10 + 1) bernilai truthy, maka operan tersebut menjadi hasil di operasi ketiga, karena operasi ketiga adalah logical OR juga.

BINAR

8. Apa output dari operasi di bawah ini?

```
const x = 10; x++; x === 11 ? "Sebelas" : "Sepuluh"
```

A. Sebelas

B. Sepuluh

C. Error

BINAR

8. Apa output dari operasi di bawah ini?

```
const x = 10; x++; x === 11 ? "Sebelas" : "Sepuluh"
```

A. Sebelas

B. Sepuluh

C. Error

Karena variabel x adalah sebuah konstanta, maka dari itu nilainya tidak dapat dirubah. Dan operator ++ ini akan merubah nilai dari variabel x, maka dari itu output-nya adalah error.

Referensi

- MDN - Expression and Operators
- MDN - Falsy
- MDN - Truthy

Udah kebayang gimana caranya untuk mengolah data di dalam Javascript?

Sabrina mau ngingetin kamu kalo semua yang disampaikan diatas itu hanya pengenalan belaka, agar kamu jago berbahasa Javascript, kamu harus perbanyak latihan.

Salah satunya bisa cobain situs ini Hackerrank - 10 Days of Javascript

Terima Kasih!

Next Topic

loading...

BINAR