JAVA SCRIPT

KONSEP DASAR PEMOGRAMAN

* Variable
* Tipe data
* Pengkodisian
* Pengulangan
* Object
* Function
* Array

KONSEP PALING DASAR PEMOGRAMAN

* Program

Program adalah kumpulan / serangkaian instruksi

* Pemograman
  + Cara kita untuk memberikan instruksi yang simple dan terstruktur pada computer
  + Intruksi yang kita berikan harus benar dan masuk akal
  + Instruksi yang kita berikan harus sesuai dengan aturan pada bahasa pemograman tertentu
* Bahasa pemograman

Compiler vs interpreter

Compiler :  
 missal ada si A dan si B, si A sedang

Membuat program setelah selesai si A kemudaian

Meng compile dan di berikan ke si B, hasil dari

Compile tesebut di jalankan oleh si B dan di

Di tampilkan, si B tidak perlu memiliki compiler dan tidak bisa melihat / memiliki sorce code si A

Interpreter:

Si A membuat program dan di duplikat, duplikat

Dari program tersebut di kirim ke si B, dan si B

Menginterpreter program tersebut kemudian menerjemahkan program tersebut dan tampilkan

Kelebihan dan kekurangan

Compiler :

* Siap untuk dijalankan
* Lebih cepat
* Source code tidak terlihat
* Tidak lintas platform ( karena ketika di compire hanya untuk system operasi tertentu, bahkan ketika system operasi nya sama tapi cpu nyaberbeda tidak akan jalan )
* Tidak fleksibel ( ketika ada perubahan maka kalian harus mengcompire ulang )
* Lankah tambahan ( compire ulang )

Interpreter

* Lintas platform ( selagi mempunyai interpreter maka program bisa dijalankan walaupun cpu nya berbeda )
* Fleksibel ( tidak perlu mengcomper ulang ketiak ada perubahan )
* Mudah di uji
* Butuh interpreter ( missal javascript butuh web browser )
* Lebih lambat ( ketika mebuka program kita harus menginterpretasi dulu )
* Sorce code mudah diakses

Hybrid

Menggambungkan kelebihan compiler dan interpreter

<http://carlcheo.com/startcoding>

stockoverflow.com

JAVA SCRIPT BISA APA ?

* Client-side web development ( atau membuat dan medukung web dari sisi client )

Native js / vanilla js ( js tanpa library dan framework )

JQUERY

ANGULARJS, REACT, EMBER, BACKBONE

* Server-side web development ( pembangunan website sisi server / backend )

Node JS

Express JS

* Browser extension / add-on
* Dekstop appilication

Electron, AppJS

* Mobile app development

JQuery Mobile, Cordova / Phonegap

* IoT dan Robotics ( internet of tings / pembuatan robot )

CylonJS, Jhonny-Five

* JSON ( javascript object notesion, yang di gunakan untuk pertukaran data dimana mana)

OPERATOR

Operator dalam java script adalah simbol yang digunakan untuk melakukan operasi

JENIS OPERATOR

BINARY

* ARIMATIKA ( MATEMATIKA )
* PENUGASAN
* PERBANDINGAN
* LOGIKA
* STRING

TERNARY

* KONDISIONAL

UNARY

* TYPEOF

BINARY

Sebuah operator yang membutuhkan 2 operand

Operand1 operator operand2 ( seperti operasi matematika )

Contoh :

( 1 + 1 )

UNARY

Sebuah operator yang membutuhkan 1 operand

Operator operand / operand operator

TERNARY

Sebuah operator yang membutuhkan 3 operand

OPERATOR *ARIMATIKA*

Contoh: +, -, \*, /, % ( % adalah modulus / sisa bagi )

OPERATOR *PRECEDENCE ( URUTAN OPERASI MATEMATIKA )*

Ada operator yang di kerjakan terlebih dahulu terlepas kita menempatkan nya dalam ekspresi

Yaitu KUKABATAKU / () \* / + -

OPERATOR  *PENUGASA / ASSIGMENT*

Yang paling kita sering gunakan adalah = dan ini di gunakan untuk variable

JENIS OPERATOR PENUGASA :

Jika kita menggunakan salah satu dari operator tersebut nilai yang kita sebelumnya simpan menggunakan operator ini tidak akan hilang / di timpa, tapi akan dilakukan operasi terlebih dahulu

* =
* +=
* -=
* \*=
* /=
* %=

OPERATOR *PERBANDINGAN / COMPARSION*

NILAI DAN TIPE DATA

BIT?

BIT adalah satuan terkecil dari kata yang ada di dalam computer kita, nialinya hanya 0 dan 1

Digit 1 trerjadi ketika salah satu komponen dalam computer kita / transistor di aliri listrik, dan 0 terjadi ketika tidak di aliri listrik

Potongan-potongan bit disebut dengan nilai / values

TIPE DATA

TIPE DATA adalah peran dari bi bit yang ada di computer kita

* Number
* String
* Boolean
* Object
* Function
* undefined

NUMBER

10, -15000, 314

STRING

‘hello world’, “hello world”

ANGKA

Di dalam javascript tidak ada bilanga integer / bulat, yang ada flooting points / bilangan pecahan, ukuran angka yang bisa di tamping adalah 64 bit 2 pangkat 64 / 18 x 10 pangkat 18, tapi tidak bisa di pakai semua karena di dalam 64 tadi ada penanda yang harus kita menampung angka angka tertentu, ada beberapa angka yang bisa di simpan di java script contohnya :

* angka tanpa decimal / integer

10, 1500, 123456

Tingkat akurasinya hanya 15 digit

*Jika kita menuliskan bilangan bulat maka yang akan di proses adalah bilangan pecahan*

* angka dengan decimal / pecahan

3.14, 0.5, 100.00

Koma di js adalah . ( titik )

Maximal kita bisa menyimpan 17 digit di belakang koma

Hati hati dalam menjumlahkan karena kadang kadang hasilnya tidak seperti yang kita harapkan

* EKSPONEN ( kita dapa menyimpan nilai yang sangat besar dan sangat kecil ke dalam eksponen )

123e5 // 123000000

123-5 // 0.00123

* BILANGAN NEGATIVE

URUTA PENGGUNAAN BIT

11 bit di guanakan exponent / negative

55 bit di gunakan fraction / positive

Jangan pernah mengawali dengan angka 0, karena hasilnya tidak sesuai dengan harapan kita

022 // 18, karena dengan kita menuliskan angka 0 di awal angka, bilangan kita di anggap menjadi bilangan octal

Sama seperti 0xfff, sama juga akan di anggap bilangan hexadecimal ( bilangan basis 16, di mulai dari 0 – 9 dan A – F )

ANGKA SPECIAL

* Infinity
* -infinity
* NaN

2 / 0 = infinity

-2 / 0 = -infinity

0 / 0 = NaN ( not a number )

100 / “apple” = NaN

Jika kalian membagi bilangan dengan string tapi di dalam string tersebut ada angka, maka js akan membaca string tersebut menjadi angka

Operator ini untuk membandingkan 2 buah operand

Operator perbandingan tesebut akan menghasilkan sebuah Boolean TRUEE FALSE

OPERATOR *LOGIKA / LOGICAL*

Operator ini digunakan untuk menetukan logika dari beberapa ekspresi yang kalian gabungkan

* && ( and )
* | | ( or )
* ! ( not )

x = 5

(x % 2 == 0)

x % 2 == 0 && x > 5

// jika menggunakan operator && harus semua true jangan ada yang salah

x % 2 == 0 || x > 5

// asal salah satu nya benar maka nilai nya true

!x > 1

// operator ! membalikan boolean jika boolean true maka akan menjadi false

OPERATOR *STRING*

Dalam java script tanda + memiliki dua fungsi yaitu jika operand nya adalah angka maka di sebut operator aritmatika, jika operand nya berupa string / tulisan maka di sebut operator penggabung string

var a

var b

a = "tarisa"

b = "dwi"

a + b

// tarisadwi

a + " " + b

// tarisa dwi

// jika ingin ada sepasi maka beri + " " +

0 + 10 + "10"

// maka hasilnya "2010"

"10" + 10 + 10

// maka hasilnya adalah "101010"

// java script akan mengetahui  tipe data apa yang akan di jumlahkan

// jika angka dengan angka maka hasilnya juga angka

// tapi jika salah satu berupa string makan tanda + berubah menjadi operator string

OPERATOR *TYPEOF / OPERAND*

Ini jenis operator yang unary yang membutuhkan 1 operand

typeof(10)

// number

typeof("5")

// string

typeof(true)

// boolean

// untuk mengetahui tipe data

OPERATOR  *KONDISIAN*

Jenis operator ini adalah ternary dimana kita butuh 3 operand di dalamnya, operator ini digunakan untuk pengecekan pada sebuah kondisi dan menetukan

Caranya :

( kondisi ) ? benar : salah

var y

y = 20

(y % 2 == 0) ? "benar" : "salah"

// apakah pengkodisian itu benar jika iya maka akan muncul benar jaka tidak maka akan muncul tidak

STRING

Tipe data pada string dalam javascript digunakan untuk memprsentasikan data tekstual

Yang disebut dengan plain text

Kita bisa kapan saja menggunakan kutip “ ” dan ‘ ’

Contoh : “ hari jum’at “ ( jika di dalam string ada kutip ‘ maka kalian harus menggunakan kutip “ ) begitu juga sebaliknya

“”yojes berjuang sendiri” di adalah cowok mandir’I”

Cara menyelesaikan masalah di atas adalah dengan

ESCAPE CHARAKTER

Untuk menampilkan sebuah karakter yang ingin di iterpretasikan berbeda dalam sebah bahasa pemograman

\’

\”

Sebagian pembuka / penutup kutip

CONCATENTION KONKATENSI +

KITA JUGA BISA MEMBANDINGKAN STRING DENGA OPERATOR “==”

CONTOH : “MELISA FAIS” == “MELISA FAIs” false

.length menghitung panjang string

CONTOH : “melisa fais” .length //11

Spasi juga di hitung

BOOLEAN

Sebah tipe data yang mempresentasikan logika true / false

Di dalam pemograman biasanya di gunakan pada statement pengkondisian, untuk menentukan aksi yang berbeda dan mengatur alur kendali dari program

Boolean memiliki dua buah nilai yaitu

TRUE | FALSE

Contoh :

var m

m = 10

m < 20

"melisa fais" == "Melisa fais"

// false

Boolean(nilai)

Di gunakan pada sorcecode buka di concel

Contoh :

Boolean(10 < 20)

// true

Boolean(10)

// true

Boolean("melisa")

// true

Boolean(-2)

// true

Boolean(0)

// false

Ini karena nilai nilai pada javascript menyimpan true / false

Yang di sebut

Truthy | falsy

Truthy

* True
* Non-zero number ( nilai yang bukan 0 / kosong )
* “string”
* Object
* Array
* Functions

Falsy

* False
* 0
* “ “
* Undefined
* Null
* NaN ( not a number )

trhuty menghasilkan true

Falsy menghasilkan false

Undefined sebuah tipe yang di hasilkan ketika kita mendeklarasikan sebuah variable tapi kita tidak mengisikan nilainya

Null sebuah nilai kosong yang bisa kita berikan pada sebuah variable

VARIABLE

Sebuah tempat atau wadah yang memiliki nama, yang digunakan untuk menyipan nilai

Sebelum memahami variable kita harus memahami

Deklarasi, inisalisasi, assignment terlebih dahulu

DEKLARASI

Mendaftarkan variable ke dalam lingkup yang sesuai

INISIALISASI

Menyediakan memori untuk variable

ASSIGMENT

Menetapkan nilai yang spesifik ke dalam variable

Var x; deklarasi dan inisalisasi

X = 10; assignment

Usahakan untuk melakukan deklarasi dan inisialisasi sebelum melakukan assignment, untuk menghindari hal hal ynag tidak di inginkan

DEKALRASI VARIABLE

* Var
* Let
* Const

MENULIS VARIABLE

* Var ga boleh ada spasi; X
* Var 1gaboleh; X
* Var boleh1;
* Var ini\_juga\_boleh;
* Var boleh$;
* Var $\_$;
* Var iniBolehBanget;

Var namaVariablePanjang; yang sering di gunakan para programmer java script / camel case

KEYWORD DAN RESERVED WORD

Adalah kata yang tidak boleh di jadikan variable / nama variable / bagian daris syntax js

SHORTHAND

untuk menuliskan variable yang lebih dari satu kita bisa menuliskan

var nama = "melisa",

    umur = "17",

    status = "pacaran sama owi";

VARIABLE SCOPE / LINGKUP VARIABLE

Console.log(); untuk menulis sesuatu di dalam browser console kita

POPUP BOX

Popup box / dialog box adalah sebuah windows kecil pada browser kita yang bisa kita munculkan, agar user bisa berinteraksi

3 POPUP BOX

* Alert ( sebuah popup box biasa dengan menuliskan alert(); )
* Prompt ( sebuah popup box yang memunculkan windows kecil dan ada inputan, sebaiknya prompt di simpan di dalam variable agar nilai yang di tuliskan pada user bisa di tampilkan, prompt mengembalikan nilai apapun ke user )
* Confirm ( sebuah popup box yang meminta per izinan kepada user / sebuah konfirmasi, mengembalikan nilai Boolean true / false )

CONTROL FLOW / ALUR KENDALI PROGRAM

Di dalam sebuah pemograman di sebut dengan normal flow / alur normal

“apabila program kita memiliki lebih dari satu statement, maka statement tersebut akan dieksekusi dari atas kebawah atau dari kiri ke kanan”

PENGULANGAN DAN PENGKODISIAN dapat mengatur alur sebuah program

PENGULANGAN / LOOP / ITERATION

Sebua konsep untuk mengerjakan / mengeksekusi stetment yang sama lebih dari satu kali / berulang

Syntax pengulangan pada java script

* While
* For
* Do while

PENGKONDISIAN / PERCABANGAN

Dimana program kita bisa memilih untuk mengeksekusi treatment yang berbeda berdasarkan kondisi yang di berikan

Syntax pengkondisian

* If
* If else
* If else if else
* Switch

WHILE

( PENGULANGAN )

While( kondisi ) {

aksi

}

Selama kondisi bernilai true maka lakukan / kerjakan aksi di dalam {}, jika kondisi bernilai false maka aksi tidak di lakukan

Agar looping tidak infinite, kita bisa menghentikan dangan cara, DIHENTIKAN OLEH USER DAN DIHENTIKAN OLEH PROGRAM

DI HENTIKAN OLEH PROGRAM

var angkot = 1;

while (angkot <= 10){

    console.log('angkot no '+angkot+' beroperasi dengan baik');

    angkot++;

}

TABEL PENELUSURAN

( trace table / dry-run test )

Dimana kita bisa mengetahui perubahan dari variable di perulangan kita

Kita cukup tuliskan nama nama variable yang kemungkinan berubah / bertambah pada saat perulangan

PENGULANGAN FOR

for(  nilaiAwal  ;   kondisi terminasi    ; increment / decrement){

    aksi

}

for (angkot; angkot <= jmlhAngkot; angkot++){

    console.log('angkot no '+angkot+' sedang tidak beroprasi');

}

PENGKONDISIAN / PERCABANGAN

IF

if(kondisi){

    aksi1

}

Melakukan aksi jika kondisi bernilai true

Jika kondisi bernilai false maka keluar dari blok { }

if(kondisi){

    aksi1

}else {

    aksi2

}

Lakukan aksi1 jika kondisi bernilai benar

Lakukan aksi2 jika kondisi bernilai false

ELSE IF

Di gunakan jika mempunyai kondisi lain, selain if dan else

if(kondisi){

    aksi1

}else if (kondisi2){

    aksi2

}else {

    aksi2

}

SWITCH

Switch tidak memiliki kondisi yang banyak

switch(ekspresi) {

    case "nilai 1":

    aksi 1

    [break;]

    case "nilai 2":

    aksi 2

    [break;]

    case "nilai n":

    aksi n

    [break;]

    default:

        aksi default

    [break;]

}

Jika nilai ekspresinya aksi 1 maka jalankan nilai 1, jika

ekspresi tidak memenuhi semua nilai , maka jalankan aksi default

Keyword break di gunakan, ketika aksi di jalankan maka akan keluar dari aksi tersebut

== operator perbandingan, === operator identitas, sama sama memembandingkan tapi membandingkan tidak hanya nilainya tapi juga tipe datanya, ketika kita menggunakan promp begitu kita inputkan itu pasti menjadi string, sedangkan kita membandingkan intejer, solusinya memaksa yang di inputkan user menjadi intejer, dengan cara menambahkan fungsi parseInt, apapun yang di in putkan user menjadi intejer, dan keudian di masukan ke var angka, lalu cek apakah angka intejer ini sama dengan angka intijer ini

FUNCTION

* Blok kode yang di buat untuk melakukan tugas yang spesifik
* Blok kode nya dapat di panggil berulang kali
* Sehingga memudahkan kita melakukan penelusuran kesalahan
* Reusability ( pengguna program berulang kali )

FUNCTION ADA DUA KONDISI YAITU: DIBUAT DAN DI PANGGIL

Contoh kasus

Menjumlahkan volume 2 kubus

1. Mengetahui sisi masing” kubus, kubus a = 8, kubus b=3
2. Hitung volume masing” kubus, 8 pangkat 3 = 512, 3 pangkat 3 = 27
3. Jumlahkan hasilnya, 512+27
4. Kembalikan nilai jawabanya, 539

PARAMETER DAN ARGUMENT

PARAMETER adalah variable yang di tulis di dalam kurung pada saat function DIBUAT, digunakan untuk menampung nilai yang dikirimkan saat function di panggil.

ARGUMENT adalah nilai yang dikirimkan ke parameter saat fungsi di PANGGIL

Bagaimana jika parameter dan argument tidak sesuai?, maksudnya jumlahnya tidak sesuai

1. jika parameter lebih sedikit dari argument, maka argument kelebihanya di abaikan

function kali(a, b){

    return a \* b;

}

var coba = kali(3, 4, 5);

// maka nilai dari argument 5 adalah di abaikan

1. jika parameter lebih banyak dari argument, maka kelebihanya akan di isi dengan nilai undefined

function kali(a, b, c, d){

    return a \* b \* c;

}

var coba = kali(3, 4, 5);

// maka nilai dari parameter d adalah di undefined

Di dalam javascript ada sebuah variable kusus yang namanya arguments

Arguments

Array yang berisi nilai yang dikirimkan saat fungsdi panggil

REFACTORING

Refactoring adalah sebuah proses mengubah kode agar menjadi lebih baik tanpa mengubah fungsionalitas

REFACTORING :

* readability ( kemudahan dalam membaca kode program, karena program yang baik tidak hanya bisa di baca oleh pembuatnya tapi jua orang lain )
* DRY / don’t repeat yourself ( hal ini dikaitkan dengan duplikasi kode, bisa penggunaan variable yang berebihan, efisiensi penggunaan function, menthod, dll )
* Testability ( penulisan kode agar mudah saat dilakukan pengujian
* Performance ( kita dapat menulis kode yang dapat, meningkatkan performasi dari programnya, contoh sederhana menggunakan looping yang baik )
* Maintainability ( bagaimana nantinya program kita dapat di kelola dan dikembangkan dengan baik )

SCOPE / LINGKUP VARIABLE

Bagaimana sebuah variable dapat diakses dalam program

Block scope dan function scope

Block scope

Block scope digunakan untuk bahasa pemograman C dan JAVA

Variable yang di buat didalam block scope ( {} ) hanya berlaku didalamnya, dan tidak bisa digunakan diluarnya

Javascript tidak menganut blokscope tapi menggunakan function scope

Function Scope

REKURSIF

Sebuah fungsi yang memanggil dirinya sendiri

Dalam penggunaan rekursif fungsinya harus berakhir dan menghasilkan nilai

Saat menggunakan rekursif harus ada kondisi berhentinya

Base Case

Kondisi akhir dari rekursif yang menhasilkan nilai

Factorial

5! = 5x4x3x2x1

Semua looping bisa dibuat rekursif, tapi rekursif tidak bisa dibuat looping

Implementasi rekursif

* Menggantikan looping
* Membuat sebuah deret fibonaci
* Bisa melakukan pencarian dan penelusuran pada struktur data list dan tree
* Bisa digunakan didalam bahasa pemograman yang tida memiliki looping

Contohnya : Haskell, elang, prolog

b