|  |  |
| --- | --- |
| **Materi** | **Nilai** |
| JavaScript Semester-4 Minggu ke-1 Video 1-10 | 95 |

**Saya Sudah Belajar dan Mengerti dan Saya BISA**

1. Instalasi Extensions yang dibutuhkan didalam pembuatan javascript yaitu Javascript (ES6) code snippets dan code runner.

2. Pembuatan javascript bisa dilakukan secara internal di dalam file html nya langsung, untuk mengaktifkan atau menggunakan javascriptnya maka harus membuat open script dan close script seperti ini <script></script> di dalam body html, di dalam script ini jika ingin menampilkan isi script di browser nya menggunakan alert(“isi”); yang nanti akan muncul seperti sebuah notifikasi bertuliskan isi.

3. Untuk menampilkan javascript yang ada pada html menggunakan live server. Live server ini merupakan ekstensi yang ada di dalam vscode.

4. Pembuatan javascript bisa dilakukan secara eksternal yaitu diluar file html nya tetapi tetap menggunakan file html untuk menempatkan script src nya. Jika diluar file html maka harus membuat folder nya dulu yaitu js dan diisi dengan file .js setelah itu file js ini bisa diisi dengan penulisan javascript apapun yang nanti bisa ditampilkan di browser apabila sudah ditambahkan <script src="js/javascript.js"></script> didalam header file html nya.

5. Jika pembuatan javascript eksternal dan internal dibuat secara bersamaan maka code akan dibaca dari atas yaitu sript src atau eksternalnya baru script yang ada di dalam body. Untuk tampilannya dia akan menampilkan yg dibaca dari atas yaitu eksternal setelah notifikasi isi dari eksternal muncul ada button oke yang apabila ditekan akan muncul notifikasi isi dari internalnya.

6. Javascript hanya bisa tampil di dalam halaman html atau bisa disebut harus ada file html nya ketika membuat javascript.

7. Terdapat variable yang bisa menyimpan sebuah data, untuk menuliskan variable ini menggunakan let namavariable = isivariable; contohnya (let angka = 10;) apabila integer, float, dll. dan (let kata = “saya belajar”) apabila string setelah itu untuk menampilkan hasil isi dari variable ini menggunakan console.log(namavariable); contoh nya console.log(angka); jika sudah menuliskan code ini maka di dalam browsernya klik kanan->inspect->console maka disitu akan muncul angka 10 di dalam barisnya dan sebelah kanan barisnya bertuliskan nama file letak variable : letak baris pada vscode nya seperti ini variable.js:3

8. Terdapat konstanta yang sama seperti variable yaitu menyimpan data, untuk menampilkan caranya sama seperti variable sedangkan penulisan code nya const namaconst = isiconst; contohnya const data = 10;

9. Perbedaan variable dengan konstanta yaitu apabila isi variable kosong tetapi ada code untuk menampilkannya maka hasilnya undefined sedangkan jika isi konstanta kosong tetapi ada code untuk menampilkannya maka hasilnya syntaxerror.

10. Terdapat cara mendeklarasikan variable lain seperti ini let angka; dari code ini variable nya belum diisi suatu data, setelah itu dibawahnya ada nama variable tadi yaitu angka dengan isi variable nya seperti ini angka = 10; maka otomatis let angka tadi terisi data integer yaitu 10. Apabila dibawah variable angka yang pertama sudah ada console.log untuk variable angka yang pertama lalu dibawah kedua baris code ini dibuat variable angka lagi dengan isi variable berbeda seperti angka = 8; lalu digunakan console.log(angka) yang berfungsi untuk menampilkan isi variable yang kedua maka variable yang tampil dari kedua variable angka ini adalah kedua-dua nya tampil karena variable bisa digunakan sebagai wadah berulang kali.

11. Berbeda dengan variable, konstanta tidak dapat dideklarasikan terpisah dengan const nya, jadi harus satu baris dengan const nya.

12. Terdapat typeof yang berfungsi untuk membaca type data dari isi suatu variable. Contohnya let isi = 5; yang berarti variable nya bernama isi lalu ditampilkan menggunakan console.log(typeof isi); maka di dalam console nya akan tampil number yang berarti isivariable ini bertipe data number, apabila diubah menjadi let isi = “belajar”; maka tampilan dalam console nya adalah string.

13. Di dalam javascript, type data integer(angka) dan float(pecahan) disebut dengan type data number sedangkan string tetap string.

14. Terdapat typedata infinity yang berarti tak terhingga, misalkan dituliskan console.log(1/0); yang berarti ini adalah infinity karena angka berapapun yang dibagi dengan 0 hasilnya adalah infinity, jika console.log(100/0); juga disebut infinity, sedangkan jika angka 0 nya diganti dengan angka lain misalkan console.log(2/2); maka yang muncul dalam console nya adalah hasil 2/2 yaitu 1, jika console.log(5/2); maka yang muncul dalam console nya adalah hasil 5/2 yairtu 2.5, percobaan lain apabila yang dibagi adalah 0 seperti ini console.log(0/1); maka hasilnya adalah 0. Jadi infinity hanya berlaku pada angka berapapun yang dibagi dengan 0.

15. Setelah itu ada NaN yang berarti hasil operasi matematika yang salah atau tidak ditentukan. Contohnya string dibagi dengan number seperti ini console.log(“belajar”/2); maka hasil pada console nya adalah NaN, contoh lain seperti ini console.log(“javascript”/2+5); maka hasil pada consolenya juga NaN karena string memang tidak bisa dibagi dengan number. Contoh lain misalkan string dibagi dengan string seperti ini console.log(“javascript”/”javacsript”); maka hasilnya NaN karena memang bukan operasi matematika.

16. Di dalam javascript, string dibagi menjadi 3 yaitu Double quotes(“”), Single quotes(‘’), dan Backticks(``). Untuk Double quotes dan Single quotes penggunaannya hampir sama yaitu jika kata yang ada didalam nya seperti “java” dan ‘script’ maka yang muncul tetap sesuai dengan yang ada di dalamnya atau “3+1” dan ‘4-2’ maka yang tampil juga tetap sesuai dengan yang ada di dalamnya, untuk penambahan antara double quotes dengan double quotes, double quotes dengan single quotes, dan single quotes dengan single quotes yang ada di dalam variable seperti ini

let kata = “java”;

let isi = ‘script’;

console.log(“belajar”+kata+isi);

maka yang tampil dalam console adalah belajar javascript, meskipun ditambahkan spasi pada codenya tetapi hasilnya tetap menjadi satu atau gabung, jika ingin ada spasi pada hasil maka ditambahkan “ ” di dalam nya seperti ini console.log(“belajar ”+kata+” ”+isi); yang nanti hasilnya menjadi belajar java script. Sedangkan Backticks bisa digunakan untuk memasukkan variable, mekakukan operasi matematika, dan menambahkan beberapa string menjadi satu, semua itu dengan menggunakan ${…} contohnya

let kelas = 11;

let java = “java”;

let script = ‘script’;

console.log(`kelas ${kelas} belajar ${java} ${script} minggu ${2-1}`); maka hasil pada console nya adalah kelas 11 belajar java script minggu 1. Apabila kata javascript nya ingin disambung maka penulisannya console.log(`kelas ${kelas} belajar ${java}${script} minggu ${2-1}`); diantara ${java} dan ${script} nya tidak ada spasi nya, jadi hasil dari backticks sesuai dengan penulisan codenya. Selain itu penggunaan double quotes dan single quotes tidak berpengaruh di dalam backticks, misalkan console.log(`”belajar”`); maka hasilnya adalah “belajar” bukan hanya belajar tetapi beserta “” juga ikut muncul.

17. Terdapat type data Boolean yang isinya hanya berupa true dan false, misalkan dalam suatu variable diisi dengan true maka hasil pada console nya adalah true, begitu juga apabila diisi dengan false.

18. Terdapat type data null atau kosong atau tidak ada isinya, misalkan dalam suatu variable diisi dengan null maka hasil pada console nya adalah null.

19. Terdapat type data undefined yang berarti variable tidak diisi dengan apa-apa dan tidak diisi juga dengan null. Contohnya let kosong; lalu console.log(kosong); maka hasil pada console nya adalah undefined karena variable nya tidak diisi dengan apa-apa.

20. Operasi matematika pada javascript yang pertama adalah positif dirubah menjadi negative seperti ini

Let x = 1;

x = -x;

console.log(x);

maka hasilnya adalah -1, jadi ketika variable positif dirubah menjadi variable negative maka tampilan value atau isi dari variable nya akan menjadi negative.

21. Operasi matematika yang kedua yaitu menuliskan nama variable dalam satu baris, mendeklarasikan value nya ditiap baris berbeda, lalu menjumlahkan beberapa variable di dalam satu variable. Contoh

let p, l, t, volume;

p = 5;

l = 6;

t = 7;

volume = p \* l \* t;

console.log(volume);

maka hasil pada console nya adalah hasil kali dari p, l, dan t yaitu 210. Selain mendeklarasikan value dari variable melalui tiap baris yang berbeda-beda, bisa juga seperti ini

let p =5, l=6, t=7, volume;

volume = p \* l \* t;

22. Terdapat beberapa operasi matematika yaitu Addition(+) untuk penjumlahan, Subtraction(-) untuk pengurangan, Multiplication(\*) untuk perkalian, Division(/) untuk pembagian, Remainder(%) untuk hasil bagi, dan Exponentiation(\*\*) untuk pangkat.

23. Penggabungan string, contoh let kota = “sido” + “arjo”; console.log(kota); maka hasil console nya adalah sidoarjo.

24. Penggabungan dengan penjumlahan antara string dengan angka, apabila diawali dengan string seperti ini console.log(“1”+5+6); maka semuanya digabung seperti string tetapi bukan terjumlahkan, jadi hasilnya 156 bukan 1 ditambah 5 ditambah 6. Apabila diawali dengan number seperti ini console.log(1+5+”6”); maka yang awal terjumlahkan yaitu 1 ditambah 5 dan yang terakhir tetap muncul berupa string, jadi hasilnya 66(6 pertama adalah hasil jumlah 1+5 dan 6 kedua adalah string).

24. Penggabungan dengan pengurangan antara string dengan angka, angka dengan angka, dan string dengan string. Contoh pertama apabila diawali string console.log(“5”-3); maka hasil pada console nya adalah 2, apabila diawali dengan angka console.log(5-“3”); maka hasil pada console nya adalah 2. Jadi meskipun diawali angka maupun diawali string, jika pengurangan tetap jalan pengurangannya. Jika angka dengan angka maka hasilnya pasti juga berjalan pengurangnnya, jika string dengan string seperti ini console.log(“5”-“3”); maka hasilnya juga 2, jadi dalam pengurangan ini baik ada string nya maupun angka saja tetap berjalan operasinya karena dari string akan dikonversikan ke dalam number.

25. Untuk pembagian, cara kerjanya sama dengan pengurangan yaitu jika ada stringnya akan dikonversikan ke dalam number jadi operasi pembagian tetap berjalan.

26. Operasi Chaining, yaitu operasi berantai, jadi dari bermacam macam variable atau wadah diisi dengan value yang sama. Misalkan let a, b, c; lalu bawahnya a=b=c=3+3; console.log(a); maka hasilnya 6, console.log(b); maka hasilnya 6. Jadi variable a b dan c isinya 6 semua.

27. Operasi Increment atau naik 1, contohnya let inc = 5; lalu bawahnya inc++; lalu console.log(inc); berarti ++ ini tambah 1, maka hasilnya adalah 5 naik 1 yaitu 6.

28. Operasi Decrement atau turun 1, sama seperti increment hanya saja bedanya ada di ++ nya, jika dalam increment menggunakan ++ yang artinya naik 1 maka di dalam decremenet menggunakan -- yang artinya turun 1.

29. Operasi bernama Comma yang hanya membaca operasi paling belakang dari beberapa operasi yang ada di dalam kurung value nya variable. Contohnya let a = (1+2, 3+4); lalu console.log(a); maka yang keluar pada console nya adalah hasil dari operasi yang akhir yaitu 3+4 sama dengan 7. Contoh lain adalah let a = (1+2, 3+4, 4+4); console.log(a); maka hasilnya adalah 8 yang didapat dari operasi 4+4.

30. Operator Logika OR yang simbolnya menggunakan ||, jika ada 4 variable yang diuji yaitu a = true, b = false, c = true, d = false. Di dalam operator logika OR, apabila salah satu yang diuji ada true nya maka hasilnya true seperti console.log(b || d || a); maka hasilnya true karena ada a yang bernilai true, apabila tidak ada true nya atau false semua maka hasilnya false seperti console.log(b||d); yang keduanya sama-sama false. Apabila keduanya true seperti console.log(a||c); maka hasilnya adalah true karena ada true nya.

31. Operator Logika AND yang simbolnya menggunakan &&, berbeda dengan OR, logika AND ini apabila salah satu yang diuji ada false nya maka hasilnya false sedangkan apabila semua yang diuji bernilai true maka hasilnya true.

32. Operator Pembanding yang simbolnya <, >, ==, >=, atau <= contohnya

let x = 8;

let y = 7;

* Console.log(x > y); berarti pertanyaannya apakah x lebih besar dari y atau 8 lebih besar dari 7, jika iya maka hasilnya true.
* Jika yang diuji adalah console.log(x < y); berarti pertanyaannya apakah x lebih kecil dari y atau 8 lebih kecil dari 7, jika tidak maka hasilnya adalah false.
* Jika yang diuji adalah console.log(x == y); berarti pertanyaannya apakah x sama dengan y atau 8 sama dengan 7, jika tidak maka hasilnya false.
* Jika yang diuji adalah console.log(x >= y); berarti pertanyaannya apakah x lebih besar sama dengan y atau 8 lebih besar sama dengan 7, jika iya maka hasilnya true.
* Terakhir jika yang diuji adalah console.log(x <= y); berarti pertanyaannya apakah x lebih kecil sama dengan y atau 8 lebih kecil sama dengan 7, jika tidak maka hasilnya false

33. Operator Pembanding identik (===) yaitu sama type dan isinya, contohnya jika ada variable

let lima = “5”;

let limo = 5;

dari kedua variable ini berbeda type data nya, yang satu string dan satunya number tetapi memiliki isi yang sama. Jika mengujinya seperti ini console.log(lima == limo); hanya menggunakan 2 sama dengan maka hasilnya adalah true karena isinya sama meskipun berbeda type, tetapi jika menguji dengan 3 sama dengan seperti ini console.log(lima === limo); maka hasilnya adalah false karena berbeda type nya.

34. Percabangan if yang berarti apabila yang diuji didalam kurung if benar maka dia akan menjalankan yang ada di dalam kurung kurawalnya, sedangkan jika salah dia akan menjalankan else nya. contoh pertama

if (true){

    console.log("dijalankan jika benar");

} else{

    console.log("dijalankan jika salah");

}

Karena yang ada di dalam kurung if adalah true maka hasil yang keluar adalah “dijalankan jika benar”, sedangkan jika yang ada di dalam kurung if diganti menjadi false maka hasil yang keluar adalah “dijalankan jika salah”. Contoh kedua adalah

let nilai = 10;

let standart = 60;

let baik = "LULUS";

let gagal = "TIDAK LULUS";

let batasatas = 100;

let batasbawah = 0;

let peringatan = "nilai salah";

if (nilai <= batasatas && nilai >= batasbawah) {

    if (nilai >= standart){

        console.log(baik);

    } else{

        console.log(gagal);

    }

} else {

    console.log(peringatan);

}

Disitu ada beberapa variable untuk diuji, ada dua pengujian if, if yang pertama adalah jika nilai kurang dari sama dengan batasatas yaitu 100 dan lebih kecil sama dengan batasbawah yaitu 0 maka dia akan menjalankan if yang selanjutnya yaitu jika nilai lebih dari sama dengan standart atau jika nilai lebih dari sama dengan 60 maka dia akan menjalankan variable baik yaitu LULUS, tetapi jika pernyataan itu salah maka dia akan menjalankan else yang gagal. Apabila nilai tidak memenuhi if yang pertama maka dia akan menjalankan else yang peringatan.

35. Selector digunakan untuk memilih, misalkan ada konstanta yang berisi number dan dengan konstanta ini dibuat sebagai pemilihan case nya. Contohnya

const hari = 1;

let hasil;

switch (hari) {

    case 1:

        hasil = "minggu";

        break;

    case 2:

        hasil = "senin";

        break;

    case 3:

        hasil = "selasa";

        break;

    case 4:

        hasil = "rabu";

        break;

    case 5:

        hasil = "kamis";

        break;

    case 6:

        hasil = "jumat";

        break;

    case 7:

        hasil = "sabtu"

        break;

    default:

        hasil = "hari tidak ada";

        break;

}

console.log(hasil);

Konstanta nya adalah 1, selanjutnya dibuatkan switch untuk menguji konstanta ini, apabila konstanta berisi 1 maka dia akan memilih case 1. Jadi isi dari konstanta yang menentukan atau mencocokkan dengan nama case nya. Jika sudah memilih case 1 selanjutnya dia akan mendeklarasikan variable hasil dengan value nya yaitu string “minggu”. Dan ketika di console.log(hasil); yang akan keluar adalah minggu. Begitu juga sebaliknya apabila konstanta berisi string lalu diuji didalam case maka case nya diberi nama berupa string agar bisa dicocokkan antara value konstanta dengan nama case yang ada di dalam switch.

36. Perulangan (looping) yang pertama ada while, contoh code yang menggunakan while

let mulai = 1;

let selesai = 10;

while (mulai <= selesai){

    console.log(mulai);

    mulai++;

}

Disini variable mulai adalah 1, dan variable selesai adalah 10. Di dalam kurung while ada mulai <= selesai yang berarti apabila mulai kurang dari sama dengan selesai maka dia akan menampilkan mulai lalu ada mulai++ yang berarti dia akan menambah nilai mulai lalu mengujinya lagi ke dalam while, jika mulai = 1, maka mulai++ yang pertama adalah 2 lalu 2 ini duji ke while apakah kurang dari sama dengan nilai selesai maka 2 akan ditampilkan di console dan seterusnya.

37. Perulangan (looping) yang kedua adalah do while, sama seperti while hanya saja yang ada di dalam kurung kurawal while diletakkan di dalam do dan while beserta kurung nya diletakkan dibawah do.

38. Perulangan (looping) yang ketiga adalah for, contoh codenya

for(let i = 0, i < 10; i++ ){

console.log(i);

}

Yang berarti dibuatkan variable i berisi 0, dan jika i kurang dari 10, lalu i akan dilooping sejumlah dari isi ke 10, lalu i yang telah dilooping akan ditampilkan di console yang hasilnya adalah mulai dari 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 hanya sampai 9 karena kurang dari 10, jadi 10 tidak termasuk.

39. function berfungsi untuk menyimpan suatu data dan jika function dipanggil maka yang muncul dalam browser adalah isi dari functionnya. Contohnya adalah function coba() yang isinya

{

let belajar = “saya belajar javascript”;

console.log(belajar);

console.log(“javascript itu mudah”);

}

Lalu untuk memanggil atau menjalankan functionnya cukup mengetikkan nama function beserta tanda kurung seperti ini coba(); jika sudah diketikkan coba() maka yang ada di dalam function akan dijalankan

40. function parameter yaitu function yang menggunakan parameter seperti ini

Function persegi (lebar, panjang){

luas = lebar\*panjang;

console.log(luas);

}

persegi(2, 3);

Yang dimaksud parameter adalah yang ada didalam kurung functionnya yaitu lebar dan panjang, dibuatkan sebuah variable luas yang isinya lebar dikali panjang lalu luas ini akan ditampilkan ke dalam console apabila nama function beserta isi parameter dipanggil yaitu persegi(2, 3); jadi parameter awal ibaratnya adalah variable lalu ketika function dipanggil dan letak parameter diisi ibaratnya mengisi variable nya.

41. function yang memiliki return, return ini fungsinya untuk mengeluarkan nilai yang telah diproses oleh function, jadi nilai yang dihasilkan dapat diolah lagi diluar function.

42. Arrow function yang pertama seperti ini

let fungsi = function (nama){

    console.log("belajar function " + nama);

}

fungsi("joni");

jadi disini adalah menjadikan variable menjadi function, caranya dengan mendeklarasikan variable dulu lalu sama dengan function dengan parameter yang diberi nama “nama” dan function ini menyimpan data console belajar function + nama(parameternya) yang bisa dipanggil apabila menuliskan nama functionnya yaitu fungsi dan parameternya sembarang misalkan fungsi(“joni”); maka hasil console nya adalah belajar function joni karena parameter nama sudah diisi dengan kata joni.

43. Arrow function yang kedua sama seperti yang pertama, hanya saja yang pertama menyertakan kata function dalam penulisan code nya sedangkan yang kedua langsung kurung yang berisi parameter tanpa mengetikkan kata function nya.

44. Arrow function yang ketiga seperti ini

let tambah = function (a,b) {

    return a + b;

}

console.log(tambah(2,3));

Jadi function ini diberi nama tambah yang berasal dari variable diubah menjadi function itu lalu parameternya ada a dan b, jika function ini dijalankan maka dia akan mereturn parameter a ditambah parameter b. karena bentuknya return maka untuk menampilkan diluar function menggunakan console.log(tambah(2,3)); yang hasilnya nanti adalah 5

45. Arrow function menggunakan => yang pertama seperti ini

let plus = (a,b) => a + b;

console.log(plus(2,3));

function ini bernama plus dan memiliki parameter a dan b lalu function ini akan menjalankan code yang ada disebelah kanan setelah tanda => apabila dipanggil menggunakan console.log yang didalamnya ada nama function beserta parameternya yaitu 2, 3. Karena setelah function dipanggil menjalankan parameter a + b maka 2 + 3 dan hasilnya adalah 5

46. Arrow function menggunakan => yang kedua seperti ini

let hasil = a => a \* 2;

console.log(hasil(4));

Perbedaan dari yang sebelumnya yaitu parameternya hanya 1, jika parameter hanya 1 maka tidak perlu menggunakan tanda kurung juga bisa berjalan. Jadi kata hasil adalah nama functionnya lalu parameternya adalah a dan jika function dijalankan maka parameter akan dikalikan 2 dan hasilnya adalah angka parameter dikali 2 (4\*2) yaitu 8.

47. Arrow function menggunakan => yang ketiga seperti ini

let lagi = () => console.log("coba lagi");

lagi();

jadi sama seperti sebelumnya, lagi adalah nama function sama dengan parameter functionnya apabila kosong ya hanya kurung saja lalu tanda => adalah isi functionnya, karena disini hanya 1 console.log yang akan ditampilkan apabila banyak yang akan ditampilkan cukup ditambahkan {} untuk menjadikan semua menjadi satu.

48. Arrow function menggunakan => yang keempat

let nilai = 8;

let uji = nilai < 7 ? () => predikat="gagal" :() => predikat="lulus";

console.log(uji());

Berarti uji adalah nama functionnya, lalu nilai < 7 adalah isi dari kurung function untuk diuji, apabila function dijalankan dengan console.log(uji()); maka dia akan menjalankan setelah tanda tanya yaitu parameter yang bernama predikat. Predikat ini dijalankan sesuai dengan uji nya, apabila memenuhi syarat uji dia akan menjalankan predikat yang pertama yaitu hasilnya gagal jika tidak lolos uji maka dia akan menjalankan predikat yang kedua yaitu lulus yang letaknya ada setelah tanda titik dua (:)

**Saya Belum Mengerti**

1.