Laporan Praktikum JavaScript

Modul X – Judul Modul

Muhammad Arlianto/22103001009

Dosen/Asisten: Nama Asistem Praktikum

Tanggal praktikum: DD Bulan 201x

Alamat *email student*

Teknik Komputer – Fakultas Teknik Komputer

Institut Teknologi dan Sains Nahdatul Ulama Kalimantan

***Abstrak*—Sebagai syarat menyelesaikan UTS pemprogramanan 1. Maka dilakukan praktikum yang mana untuk mengetahui pemahaman kami mahasiswa mengenai javascript. Dalam memahami teori yang telah kami terima selama perkulian hingga UTS. Sehingga praktikum ini sangat penting dalam mengetahui kemampuan kami mahasiswa dalam meningkatkan kemampuan serta untuk memahami lebih mendalam mengenai JavaScript.**

**Salah satu hal yang perlu kami lakukan dalam praktikum ini adalah untuk mengetahu element, variable, operator dan tipe data dalam menjalankan JavaScript. Hal ini sangat berhubungan dengan pemahaman kami mahasiswa mengenai logika JavaScript dalam menerima perintah yang akan kami berikan.**

**Media praktikum JavaScript kami mahasiswa Institut Teknologi dan Sains Nahdatul Ulama Kalimantan menggunakan FreeCodeCamp.org. Sehingga hasil dari praktikum menggunakan FreeCodeCamp.org inilah menjadi acuan kami mahasiswa Institut Teknologi dan Sains Nahdatul Ulama Kalimantan dalam Menyusun laporan praktikum ini.**

1. Pendahuluan

Penyelesaian mata kuliah pemprograman 1 Institut Teknologi dan Sains Nahdatul Ulama Kalimantan tengah. Salah satunya melakukan praktikum mengenai Bahasa pemprograman JavaScript. Praktikum Bahasa pemprograman JavaScript pada UTS ini kami membahas mengenai Coment, Variable, Operator, Tipe Data, dan Method Array.

Dengan menggunakan media uji berupa FreeCodeCamp.org, sebagai salah satu media yang menyediakan jasa untuk melakukan praktikum Bahasa pemprograman JavaScript. Alasan JavaScript yang kami gunakan dalam praktikum kali ini adalah karena keputusan Institut Teknologi dan Sains Nahdatul Ulama Kalimantan.

Bahasa JavaScript adalah Bahasa pemprograman yang cukup popular saat ini.ini juga merupakan alasana Institut Teknolohi dan Sains Nahdatul Ulama memilih Bahasa pemprograman JavaScript, sebagai Bahasa pemprograman yang harus dipelajari mahasiswa. Dengan pemakaian yang bisa dikatakan sangat banyak untuk membangun suatu website. Tidak ganya untuk membuat website tapi juga bisa untuk memperindah tampilan website dalam menyampaikan informasi.

1. Tinjauan Pustaka
   1. JavaScript

Javascript adalah bahasa yang berbentuk kumpulan skrip yang pada fungsinya berjalan pada suatu dokumen HTML, sepanjang sejarah internet bahasa ini adalah bahasa skrip pertama untuk web. Bahasa ini adalah bahasa pemrograman untuk memberikan kemampuan tambahan terhadap bahasa HTML dengan mengijinkan pengeksekusian perintah perintah di sisi user, yang artinya di sisi browser bukan di sisi server web.

Javascript bergantung kepada browser(navigator) yang memanggil halaman web yang berisi skrip skrip dari Javascript dan tentu saja terselip di dalam dokumen HTML. Javascript juga tidak memerlukan kompilator atau penterjemah khusus untuk menjalankannya (pada kenyataannya kompilator Javascript sendiri sudah termasuk di dalam browser tersebut). Lain halnya dengan bahasa “Java” (dengan mana JavaScript selalu di banding bandingkan) yang memerlukan kompilator khusus untuk menterjemahkannya di sisi user/klien.

* 1. Skrip Javascript

Skrip dari JavaScript terletak di dalam dokumen HTML. Kode tersebut tidak akan terlihat dari dalam jendela navigator anda, karena diantara tag tertentu yang memerintahkan navigator untuk memperlakukan bahwa skrip tersebut adalah skrip dari JavaScript.

* 1. Fungsi Sistem

Javascript sering digunakan sebagai developer untuk menciptakan interaksi yang lebih dinamis pada halaman web, aplikasi, server, atau game. Developer menggunakan javascript biasanya bisa bersama dengan HTML dan CSS.

Dengan javascripst sebuah halaman web bisa menjadi lebih dinamis dan interaktif terhadap user karena halaman web mampu berfungsi sebagai sebuah program aplikasi yang dapat memproses masukan yag diberikan user dan memberikan hasi sesuai dengan yang telah diprogramkan.

1. Metode Praktikum
   1. Mempersiapkan Peralatan Praktikum

Sebelum memulai praktikum seluruh mahasiswa Institut Teknologi dan Sains Nahdatul Ulama Kalimantan di haruskan untuk mempersiapakan peralatan praktikum sebagai berikut:

1. Laptop
2. Smartphone (Android/IOS)
   1. Mempersiapkan Media Praktikum

Mempersiapkan Freecodecamp.org sebagai media untuk melakukan praktikum. Dalam melakukan praktikum melalui Freecodecamp.org kami diharuskan untuk login untuk pembuatan akun di Freecodecamp.org untuk medapatakan akses menggunakan Freecodecamp.org sebagai media praktikum.

* 1. Mengerjakan 48 Tugas di media Praktikum

Setelah selesai mempersiapkan media praktikum kami diminta dosen untuk mengerjakan tugas praktikum di Freecodecamp.org sebanyak 48 soal yang membahasa masalah materi yang kami dapatkan selama mata kuliah pemprograman 1 sampai praktikum ini dilakukan.

Terdapat 48 soal yang semuanya membahasa bagaimana konsep penerapan javascript dan logika javascript dalam menerima perintah. 48 soal praktikum yang disediakan Freecodecamp.org adalah sebagai berikut beserta penjelasannya:

1. Komentar Kode Java Skrip

Komentar adalah baris kode yang sengaja diabaikan oleh JavaScript. Komentar adalah cara yang bagus untuk meninggalkan catatan untuk diri sendiri dan orang lain yang nantinya perlu mencari tahu apa yang dilakukan kode itu.

Ada dua cara untuk menulis komentar di JavaScript:

Menggunakan //akan memberitahu JavaScript untuk mengabaikan sisa teks pada baris saat ini. Ini adalah komentar sebaris:

// This is an in-line comment.

Anda dapat membuat komentar multi-baris yang diawali /\*dan diakhiri dengan \*/. Ini adalah komentar multi-baris:

/\* This is a multi-line comment \*/

1. Deklarasi Variable Javascript

*Variabel* memungkinkan komputer untuk menyimpan dan memanipulasi data secara dinamis. Mereka melakukan ini dengan menggunakan "label" untuk menunjuk ke data daripada menggunakan data itu sendiri. Salah satu dari delapan tipe data dapat disimpan dalam variabel.

Variabel mirip dengan variabel x dan y yang Anda gunakan dalam matematika, yang berarti mereka adalah nama sederhana untuk mewakili data yang ingin kita rujuk. Variabel komputer berbeda dari variabel matematika karena mereka dapat menyimpan nilai yang berbeda pada waktu yang berbeda.

Kami memberi tahu JavaScript untuk membuat atau mendeklarasikan variabel dengan meletakkan kata kunci var di depannya, seperti:

*var ourName;*

1. Menyimpan Nilai dengan operator tugas

Dalam JavaScript, Anda dapat menyimpan nilai dalam variabel dengan operator penugasan= ( ).

myVariable = 5;

Ini memberikan Numbernilai 5ke myVariable.

Dalam JavaScript, Anda dapat menyimpan nilai dalam variabel dengan operator penugasan= ( ).

var myVar;

myVar = 5;

Pertama, kode ini membuat variable bernama myVar. Kemudian, kode ditetapkan 5ke myVar. Sekarang, jika myVarmuncul lagi dalam kode, program akan memperlakukannya seolah-olah 5.

1. Menetapkan nilai satu variable ke variable lain

Setelah nilai ditetapkan ke variabel menggunakan operator penugasan , Anda dapat menetapkan nilai variabel tersebut ke variabel lain menggunakan operator penugasan .

var myVar;

myVar = 5;

var myNum;

myNum = myVar;

Di atas mendeklarasikan myVarvariabel tanpa nilai, lalu memberinya nilai 5. Selanjutnya, variabel bernama myNumdideklarasikan tanpa nilai. Kemudian, isi myVar(yang 5) ditugaskan ke variabel myNum. Sekarang, myNumjuga memiliki nilai 5.

1. Inisialisasi variable dengan operator penugasan

Adalah umum untuk menginisialisasi variabel ke nilai awal di baris yang sama seperti yang dideklarasikan.

var myVar = 0;

Membuat variabel baru yang disebut myVardan memberinya nilai awal 0.

1. Deklarasi Variable String

Sebelumnya Anda menggunakan kode berikut untuk mendeklarasikan variabel:

var myName;

Tetapi Anda juga dapat mendeklarasikan variabel string seperti ini:

var myName = "your name";

"your name"disebut string literal. String literal, atau string, adalah serangkaian karakter nol atau lebih yang diapit oleh tanda kutip tunggal atau ganda.

1. Variabel yang tidak diinisialisasikan

Ketika variabel JavaScript dideklarasikan, mereka memiliki nilai awal undefined. Jika Anda melakukan operasi matematika pada suatu undefinedvariabel, hasilnya NaNadalah "Bukan Angka" . Jika Anda menggabungkan string dengan variabel , undefinedAnda akan mendapatkan string .undefined

1. Sensitivitas dalam variable

Dalam JavaScript semua variabel dan nama fungsi peka huruf besar/kecil. Ini berarti bahwa kapitalisasi penting.

MYVAR tidak sama dengan MyVarnor myvar. Dimungkinkan untuk memiliki beberapa variabel berbeda dengan nama yang sama tetapi casing berbeda. Sangat disarankan agar demi kejelasan, Anda tidak menggunakan fitur bahasa ini.

**Praktek terbaik**

Tulis nama variabel dalam JavaScript di camelCase . Dalam camelCase , nama variabel multi-kata memiliki kata pertama dalam huruf kecil dan huruf pertama dari setiap kata berikutnya dikapitalisasi.

**Contoh:**

var someVariable;

var anotherVariableName;

var thisVariableNameIsSoLong;

1. Perbedaan var dan let

Salah satu masalah terbesar dengan mendeklarasikan variabel dengan varkata kunci adalah Anda dapat dengan mudah menimpa deklarasi variabel:

var camper = "James";

var camper = "David";

console.log(camper);

Dalam kode di atas, campervariabel awalnya dideklarasikan sebagai James, dan kemudian diganti menjadi David. Konsol kemudian menampilkan string David.

Dalam aplikasi kecil, Anda mungkin tidak mengalami masalah seperti ini. Tetapi ketika basis kode Anda menjadi lebih besar, Anda mungkin secara tidak sengaja menimpa variabel yang tidak Anda inginkan. Karena perilaku ini tidak menimbulkan kesalahan, pencarian dan perbaikan bug menjadi lebih sulit.

Kata kunci yang disebut letdiperkenalkan di ES6, pembaruan besar untuk JavaScript, untuk mengatasi masalah potensial ini dengan varkata kunci. Anda akan belajar tentang fitur ES6 lainnya di tantangan selanjutnya.

Jika Anda mengganti vardengan letkode di atas, itu menghasilkan kesalahan:

let camper = "James";

let camper = "David";

Kesalahan dapat dilihat di konsol browser Anda.

Jadi tidak seperti var, ketika Anda menggunakan let, variabel dengan nama yang sama hanya dapat dideklarasikan satu kali.

1. Deklarasikan variable dengan const

Kata kunci letbukan satu-satunya cara baru untuk mendeklarasikan variabel. Di ES6, Anda juga dapat mendeklarasikan variabel menggunakan constkata kunci.

Const memiliki semua fitur luar biasa yang letdimiliki, dengan bonus tambahan bahwa variabel yang dideklarasikan menggunakan consthanya-baca. Mereka adalah nilai konstan, yang berarti bahwa sekali variabel ditetapkan dengan const, itu tidak dapat dipindahkan:

const FAV\_PET = "Cats";

FAV\_PET = "Dogs";

Konsol akan menampilkan kesalahan karena menetapkan kembali nilai FAV\_PET.

Anda harus selalu memberi nama variabel yang tidak ingin Anda tetapkan kembali menggunakan constkata kunci. Ini membantu ketika Anda secara tidak sengaja mencoba menetapkan kembali variabel yang dimaksudkan untuk tetap konstan.

**Catatan:** Pengembang biasanya menggunakan pengidentifikasi variabel huruf besar untuk nilai yang tidak dapat diubah dan huruf kecil atau camelCase untuk nilai yang dapat diubah (objek dan larik). Anda akan belajar lebih banyak tentang objek, array, dan nilai yang tidak dapat diubah dan yang dapat diubah dalam tantangan selanjutnya. Juga dalam tantangan selanjutnya, Anda akan melihat contoh pengidentifikasi variabel huruf besar, huruf kecil, atau camelCase.

1. Number

Numberadalah tipe data dalam JavaScript yang mewakili data numerik.

Sekarang mari kita coba menjumlahkan dua angka menggunakan JavaScript.

JavaScript menggunakan +simbol sebagai operator tambahan ketika ditempatkan di antara dua angka.

**Contoh:**

const myVar = 5 + 10;

myVarsekarang memiliki nilai 15.

1. Pengurangan (-)

Kami juga dapat mengurangi satu angka dari yang lain.

JavaScript menggunakan -simbol untuk pengurangan.

**Contoh**

const myVar = 12 - 6;

myVarakan memiliki nilai 6.

1. Perkalian Javascript (\*)

Kita juga dapat mengalikan satu angka dengan angka lainnya.

JavaScript menggunakan \*simbol untuk perkalian dua angka.

**Contoh**

const myVar = 13 \* 13;

myVarakan memiliki nilai 169.

1. Pembagaian Javascript (/)

Kita juga dapat membagi satu bilangan dengan bilangan lainnya.

JavaScript menggunakan /simbol untuk pembagian.

**Contoh**

const myVar = 16 / 2;

myVarsekarang memiliki nilai 8.

1. Operator Penugasan Penambahan(++)

Anda dapat dengan mudah menambah atau menambahkan satu ke variabel dengan ++operator.

i++;

adalah setara dengan

i = i + 1;

**Catatan:** Seluruh baris menjadi i++;, menghilangkan kebutuhan akan tanda sama dengan.

1. Operator Penugasan Pengurangan (--)

Anda dapat dengan mudah mengurangi atau mengurangi variabel satu per satu dengan --operator.

i--;

adalah setara dengan

i = i - 1;

**Catatan:** Seluruh baris menjadi i--;, menghilangkan kebutuhan akan tanda sama dengan.

1. Angka Desimal Javascript

Kita dapat menyimpan angka desimal dalam variabel juga. Bilangan desimal kadang-kadang disebut sebagai bilangan floating point atau float .

**Catatan:** saat Anda menghitung angka, angka tersebut dihitung dengan presisi yang terbatas. Operasi menggunakan floating point dapat menyebabkan hasil yang berbeda dari hasil yang diinginkan. Jika Anda mendapatkan salah satu dari hasil ini, buka topik di [forum freeCodeCamp](https://forum.freecodecamp.org/) .

1. Perkalian Bilangan Desimal di Javascript

Dalam JavaScript, Anda juga dapat melakukan perhitungan dengan angka desimal, seperti halnya bilangan bulat.

Mari kita kalikan dua desimal bersama-sama untuk mendapatkan produk mereka.

1. Pembagian Desimal di Javascript

Dalam JavaScript, Anda juga dapat melakukan perhitungan dengan angka desimal, seperti halnya bilangan bulat.

Mari kita kalikan dua desimal bersama-sama untuk mendapatkan produk mereka.

1. Operator Sisa (Remainder)

Operator *sisa* memberikan sisa %pembagian dua bilangan.

**Contoh**

5 % 2 = 1 karena  
Math.floor(5 / 2) = 2 (Pembagian)  
2 \* 2 = 4  
5 - 4 = 1 (Sisa)

**Penggunaan**  
Dalam matematika, suatu bilangan dapat diperiksa menjadi genap atau ganjil dengan memeriksa sisa pembagian bilangan tersebut dengan 2.

17 % 2 = 1 (17 Ganjil)  
48 % 2 = 0 (48 Genap)

1. Operasi Penambahan Matematika dan Penugasan Satu Langkah (+=) Augmented

Dalam pemrograman, biasanya menggunakan tugas untuk mengubah isi variabel. Ingatlah bahwa segala sesuatu di sebelah kanan tanda sama dengan dievaluasi terlebih dahulu, sehingga kita dapat mengatakan:

myVar = myVar + 5;

5 untuk menambah myVar. Karena ini adalah pola yang umum, ada operator yang melakukan operasi matematika dan penugasan dalam satu langkah.

Salah satu operator tersebut adalah +=operator.

let myVar = 1;

myVar += 5;

console.log(myVar);

6akan ditampilkan di konsol.

1. Operasi Pengurangan Matematika dan Penugasan Satu Langkah (-=) Augmented

Seperti +=operator, -=mengurangi angka dari variabel.

myVar = myVar - 5;

akan mengurangi 5dari myVar. Ini dapat ditulis ulang sebagai:

myVar -= 5;

1. Operasi Perkalian Matematika dan Penugasan Satu Langkah (\*=) Augmented

Operator \*=mengalikan variabel dengan angka.

myVar = myVar \* 5;

akan dikalikan myVardengan 5. Ini dapat ditulis ulang sebagai:

myVar \*= 5;

1. Operasi Pembagian Matematika dan Penugasan Satu Langkah (/=) Augmented

Operator /=membagi variabel dengan nomor lain.

myVar = myVar / 5;

Akan dibagi myVardengan 5. Ini dapat ditulis ulang sebagai:

myVar /= 5;

1. Kutipan Literal Dalam String

Saat Anda mendefinisikan string, Anda harus memulai dan mengakhiri dengan tanda kutip tunggal atau ganda. Apa yang terjadi ketika Anda membutuhkan kutipan literal: "atau 'di dalam string Anda?

Dalam JavaScript, Anda dapat menghindari kutipan dari menganggapnya sebagai akhir kutipan string dengan menempatkan garis miring terbalik ( \) di depan kutipan.

const sampleStr = "Alan said, \"Peter is learning JavaScript\".";

1. Mengutip String Dengan Kutip Tunggal

Nilai string dalam JavaScript dapat ditulis dengan tanda kutip tunggal atau ganda, selama Anda memulai dan mengakhiri dengan jenis tanda kutip yang sama. Tidak seperti beberapa bahasa pemrograman lain, tanda kutip tunggal dan ganda bekerja sama dalam JavaScript.

const doubleQuoteStr = "This is a string";

const singleQuoteStr = 'This is also a string';

Alasan mengapa Anda mungkin ingin menggunakan satu jenis kutipan di atas yang lain adalah jika Anda ingin menggunakan keduanya dalam sebuah string. Ini mungkin terjadi jika Anda ingin menyimpan percakapan dalam string dan percakapan dalam tanda kutip. Penggunaan lain untuk itu adalah menyimpan <a>tag dengan berbagai atribut dalam tanda kutip, semuanya dalam string.

const conversation = 'Finn exclaims to Jake, "Algebraic!"';

Namun, ini menjadi masalah jika Anda perlu menggunakan tanda kutip terluar di dalamnya. Ingat, sebuah string memiliki jenis kutipan yang sama di awal dan akhir. Tetapi jika Anda memiliki kutipan yang sama di suatu tempat di tengah, string akan berhenti lebih awal dan menimbulkan kesalahan.

const goodStr = 'Jake asks Finn, "Hey, let\'s go on an adventure?"';

const badStr = 'Finn responds, "Let's go!"';

Di sini badStrakan melempar kesalahan.

Dalam goodStr di atas, Anda dapat menggunakan kedua tanda kutip dengan aman dengan menggunakan garis miring terbalik \sebagai karakter pelarian.

**Catatan:** Garis miring terbalik \tidak boleh disamakan dengan garis miring ke depan /. Mereka tidak melakukan hal yang sama.

1. Escape Dalam String Javascript

Kutipan bukan satu-satunya karakter yang dapat *diloloskan* di dalam string. Ada dua alasan untuk menggunakan karakter pelolosan:

Untuk memungkinkan Anda menggunakan karakter, Anda mungkin tidak dapat mengetik, seperti baris baru.

Untuk memungkinkan Anda merepresentasikan beberapa tanda kutip dalam sebuah string tanpa JavaScript salah mengartikan apa yang Anda maksud.

Kami mempelajari ini di tantangan sebelumnya.

| **Kode** | **Keluaran** |
| --- | --- |
| \' | kutipan tunggal |
| \" | kutipan ganda |
| \\ | garis miring terbalik |
| \n | garis baru |
| \t | tab |
| \r | Kereta kembali |
| \b | batas kata |
| \f | umpan formulir |

*Perhatikan bahwa garis miring terbalik itu sendiri harus diloloskan agar dapat ditampilkan sebagai garis miring terbalik.*

1. Menggabung String Dengan Operator (+)

Dalam JavaScript, ketika +operator digunakan dengan Stringnilai, itu disebut operator gabungan . Anda dapat membuat string baru dari string lain dengan menggabungkannya*.*

**Contoh**

'My name is Alan,' + ' I concatenate.'

**Catatan:** Perhatikan spasi. Penggabungan tidak menambahkan spasi di antara string yang digabungkan, jadi Anda harus menambahkannya sendiri.

Contoh:

const ourStr = "I come first. " + "I come second.";

String I come first. I come second.akan ditampilkan di konsol.

1. Menggabung String Dengan Operator (+=)

Kita juga dapat menggunakan +=operator untuk menggabungkan string ke akhir variabel string yang ada. Ini bisa sangat membantu untuk memutus tali panjang pada beberapa baris.

**Catatan:** Perhatikan spasi. Penggabungan tidak menambahkan spasi di antara string yang digabungkan, jadi Anda harus menambahkannya sendiri.

Contoh:

let ourStr = "I come first. ";

ourStr += "I come second.";

ourStrsekarang memiliki nilai string I come first. I come second.

1. Membangun String Dengan Variable Menggunakan Opertator (+)

Terkadang Anda perlu membuat string. Dengan menggunakan operator penggabungan ( +), Anda dapat menyisipkan satu atau beberapa variabel ke dalam string yang sedang Anda buat.

Contoh:

const ourName = "freeCodeCamp";

const ourStr = "Hello, our name is " + ourName + ", how are you?";

ourStrakan memiliki nilai string Hello, our name is freeCodeCamp, how are you?.

1. Menambahkan Variable ke String Menggunakan Opertator (+=)

Sama seperti kita dapat membuat string pada beberapa baris dari string literal , kita juga dapat menambahkan variabel ke string menggunakan +=operator plus equals ( ).

Contoh:

const anAdjective = "awesome!";

let ourStr = "freeCodeCamp is ";

ourStr += anAdjective;

ourStrakan memiliki nilai freeCodeCamp is awesome!.

1. Mencari Length

Anda dapat menemukan panjang suatu Stringnilai dengan menulis .lengthsetelah variabel string atau literal string.

console.log("Alan Peter".length);

Nilai 10akan ditampilkan di konsol. Perhatikan bahwa karakter spasi antara "Alan" dan "Peter" juga dihitung.

Misalnya, jika kita membuat variabel const firstName = "Ada", kita bisa mengetahui berapa panjang string Adadengan menggunakan firstName.lengthproperti.

1. Notasi Bracket untuk menemukan Array dalam Array

Notasi kurung adalah cara untuk mendapatkan karakter pada indeks tertentu dalam sebuah string.

Sebagian besar bahasa pemrograman modern, seperti JavaScript, tidak mulai menghitung pada 1 seperti yang dilakukan manusia. Mereka mulai dari 0. Ini disebut sebagai pengindeksan berbasis Nol .

Misalnya, karakter pada indeks 0 dalam kata Charlesadalah C. Jadi jika const firstName = "Charles", Anda bisa mendapatkan nilai huruf pertama dari string dengan menggunakan firstName[0].

Contoh:

const firstName = "Charles";

const firstLetter = firstName[0];

firstLetterakan memiliki nilai string C.

1. Memahami Kekekalan String

Dalam JavaScript, Stringnilai tidak dapat diubah , yang berarti nilai tersebut tidak dapat diubah setelah dibuat.

Misalnya, kode berikut akan menghasilkan kesalahan karena huruf Bdalam string Bobtidak dapat diubah menjadi huruf J:

let myStr = "Bob";

myStr[0] = "J";

Perhatikan bahwa ini tidak berarti bahwa myStrtidak dapat ditugaskan kembali. Satu-satunya cara untuk mengubahnya myStradalah dengan menetapkannya dengan nilai baru, seperti ini:

let myStr = "Bob";

myStr = "Job";

1. Gunakan Notasi Bracket untuk Menemukan Karakter ke-N dalam sebuah String

Anda juga dapat menggunakan notasi braket untuk mendapatkan karakter pada posisi lain dalam sebuah string.

Ingat bahwa komputer mulai menghitung pada 0, jadi karakter pertama sebenarnya adalah karakter ke nol.

Contoh:

const firstName = "Ada";

const secondLetterOfFirstName = firstName[1];

secondLetterOfFirstNameakan memiliki nilai string d.

1. Gunakan Notasi Bracket untuk Menemukan Karakter Terakhir dalam sebuah String

Untuk mendapatkan huruf terakhir dari sebuah string, Anda dapat mengurangi satu dari panjang string.

Misalnya, jika const firstName = "Ada", Anda bisa mendapatkan nilai huruf terakhir dari string dengan menggunakan firstName[firstName.length - 1].

Contoh:

const firstName = "Ada";

const lastLetter = firstName[firstName.length - 1];

lastLetterakan memiliki nilai string a.

1. Gunakan Notasi Bracket untuk Menemukan Karakter Nth-to-Last dalam sebuah String

Anda dapat menggunakan prinsip yang sama yang baru saja kita gunakan untuk mengambil karakter terakhir dalam sebuah string untuk mengambil karakter Nth-to-last.

Misalnya, Anda bisa mendapatkan nilai huruf ketiga hingga terakhir dari const firstName = "Augusta"string dengan menggunakanfirstName[firstName.length - 3]

Contoh:

const firstName = "Augusta";

const thirdToLastLetter = firstName[firstName.length - 3];

thirdToLastLetterakan memiliki nilai string s.

1. Kata Kosong

Anda diberikan kalimat dengan beberapa kata yang hilang, seperti kata benda, kata kerja, kata sifat dan kata keterangan. Anda kemudian mengisi bagian yang hilang dengan kata-kata pilihan Anda sedemikian rupa sehingga kalimat yang lengkap masuk akal.

Pertimbangkan kalimat ini - Itu benar-benar **\_\_\_\_** , dan kami **\_\_\_\_** sendiri **\_\_\_\_** . Kalimat ini memiliki tiga bagian yang hilang- kata sifat, kata kerja dan kata keterangan, dan kita dapat menambahkan kata-kata pilihan kita untuk melengkapinya. Kami kemudian dapat menetapkan kalimat lengkap ke variabel sebagai berikut:

const sentence = "It was really " + "hot" + ", and we " + "laughed" + " ourselves " + "silly" + ".";

1. Simpan Beberapa Nilai dalam satu Variabel menggunakan Array JavaScript

Dengan variabel JavaScript array, kita dapat menyimpan beberapa bagian data di satu tempat.

Anda memulai deklarasi array dengan kurung siku pembuka, akhiri dengan kurung siku penutup, dan beri tanda koma di antara setiap entri, seperti ini:

const sandwich = ["peanut butter", "jelly", "bread"];

1. Sarang satu Array di dalam Array Lain

Anda juga dapat menyarangkan array di dalam array lain, seperti di bawah ini:

const teams = [["Bulls", 23], ["White Sox", 45]];

Ini juga disebut array multidimensi .

1. Akses Data Array dengan Indeks

Kita dapat mengakses data di dalam array menggunakan *indexes* .

Indeks array ditulis dalam notasi braket yang sama dengan yang digunakan string, kecuali bahwa alih-alih menentukan karakter, mereka menentukan entri dalam array. Seperti string, array menggunakan pengindeksan *berbasis nol* , sehingga elemen pertama dalam array memiliki indeks 0.

**Contoh**

const array = [50, 60, 70];

console.log(array[0]);

const data = array[1];

console.log(array[0])Mencetak , 50dan datamemiliki nilai 60.

1. Ubah Data Array Dengan Indeks

Tidak seperti string, entri array dapat berubah dan dapat diubah secara bebas, bahkan jika array dideklarasikan dengan const.

**Contoh**

const ourArray = [50, 40, 30];

ourArray[0] = 15;

ourArraysekarang memiliki nilai [15, 40, 30].

**Catatan:** Seharusnya tidak ada spasi antara nama array dan tanda kurung siku, seperti array [0]. Meskipun JavaScript dapat memproses ini dengan benar, ini dapat membingungkan pemrogram lain yang membaca kode Anda.

1. Akses Array Multi-Dimensi Dengan Indeks

Salah satu cara untuk memikirkan array multi-dimensi , adalah sebagai array dari array . Saat Anda menggunakan tanda kurung untuk mengakses larik Anda, kumpulan tanda kurung pertama merujuk ke entri di larik terluar (tingkat pertama), dan setiap pasangan tanda kurung tambahan merujuk ke tingkat entri berikutnya di dalamnya.

**Contoh**

const arr = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9],

[[10, 11, 12], 13, 14]

];

const subarray = arr[3];

const nestedSubarray = arr[3][0];

const element = arr[3][0][1];

Dalam contoh ini, subarraymemiliki nilai [[10, 11, 12], 13, 14], nestedSubarraymemiliki nilai [10, 11, 12], dan elementmemiliki nilai 11.

**Catatan:** Seharusnya tidak ada spasi antara nama array dan tanda kurung siku, seperti array [0][0]dan bahkan ini array [0] [0]tidak diperbolehkan. Meskipun JavaScript dapat memproses ini dengan benar, ini dapat membingungkan pemrogram lain yang membaca kode Anda.

1. Memanipulasi Array Dengan Push()

Cara mudah untuk menambahkan data ke akhir array adalah melalui push()fungsi.

.push()mengambil satu atau lebih parameter dan "mendorong" mereka ke akhir array.

Contoh:

const arr1 = [1, 2, 3];

arr1.push(4);

const arr2 = ["Stimpson", "J", "cat"];

arr2.push(["happy", "joy"]);

arr1sekarang memiliki nilai [1, 2, 3, 4]dan arr2memiliki nilai ["Stimpson", "J", "cat", ["happy", "joy"]].

1. Memanipulasi Array Dengan pop()

Cara lain untuk mengubah data dalam array adalah dengan .pop()fungsi.

.pop()digunakan untuk mengeluarkan nilai dari akhir array. Kami dapat menyimpan nilai yang muncul ini dengan menetapkannya ke variabel. Dengan kata lain, .pop()menghapus elemen terakhir dari array dan mengembalikan elemen itu.

Jenis entri apa pun dapat dikeluarkan dari array - angka, string, bahkan array bersarang.

const threeArr = [1, 4, 6];

const oneDown = threeArr.pop();

console.log(oneDown);

console.log(threeArr);

Yang pertama console.logakan menampilkan nilai 6, dan yang kedua akan menampilkan nilai [1, 4].

1. Memanipulasi Array Dengan shift()

pop()selalu menghapus elemen terakhir dari array. Bagaimana jika Anda ingin menghapus yang pertama?

Di situlah .shift()masuk. Ia bekerja seperti .pop(), kecuali menghapus elemen pertama, bukan yang terakhir.

Contoh:

const ourArray = ["Stimpson", "J", ["cat"]];

const removedFromOurArray = ourArray.shift();

removedFromOurArrayakan memiliki nilai string Stimpson, dan ourArrayakan memiliki ["J", ["cat"]].

1. Memanipulasi Array Dengan unshift()

Anda tidak hanya dapat shiftelemen dari awal array, Anda juga dapat unshiftelemen ke awal array yaitu menambahkan elemen di depan array.

.unshift()bekerja persis seperti .push(), tetapi alih-alih menambahkan elemen di akhir array, unshift()tambahkan elemen di awal array.

Contoh:

const ourArray = ["Stimpson", "J", "cat"];

ourArray.shift();

ourArray.unshift("Happy");

Setelah shift, ourArrayakan memiliki nilai ["J", "cat"]. Setelah unshift, ourArrayakan memiliki nilai ["Happy", "J", "cat"].

1. Daftar belanja

Buat daftar belanja di variabel myList. Daftar tersebut harus berupa array multidimensi yang berisi beberapa sub-array.

Elemen pertama di setiap sub-array harus berisi string dengan nama item. Elemen kedua harus berupa angka yang mewakili kuantitas yaitu

["Chocolate Bar", 15]

1. Hasil dan Analisis

Dari hasil praktikum yang telah kami lakukan maka mendapatkan hasil berupa kode javascript. Kode ini didapatkan setelah memasukkan aturan-aturan yang diminta oleh media praktikum untuk kami lakukan. Dengan memberikan arahan Freecodecamp akan menampilkan hasil akhir dari kode yang dimasukkan. Berikut hasil dan penjelasan kode-kode javascript hasil dari praktikum:

1. Komentar Kode Java Skrip

Komentar ini digunakan untuk menyisipkan suatu arahan mengenai suatu kode atau bisa juga menjadi pembeda dalam suatu kode di javascript.

1. Deklarasi Variable Javascript

var myName;

Variable digunakan untuk menyimpan nilai dalam Javascript. Variable dalam javascript sering ditulis Var pada awal untuk jenis data yang akan dimasukkan. Dengan ini menjadikan data lebih dinamis dan mudah untuk dimanipulasi dalam javascript.

1. Menyimpan Nilai dengan operator tugas

Dalam JavaScript, Anda dapat menyimpan nilai dalam variabel dengan operator penugasan = ( ).

myVariable = 5;

Ini memberikan Numbernilai 5ke myVariable.

Dalam JavaScript, Anda dapat menyimpan nilai dalam variabel dengan operator penugasan= ( ).

var myVar;

myVar = 5;

Pertama, kode ini membuat variable bernama myVar. Kemudian, kode ditetapkan 5ke myVar. Sekarang, jika myVarmuncul lagi dalam kode, program akan memperlakukannya seolah-olah 5.

1. Menetapkan nilai satu variable ke variable lain

Setelah nilai ditetapkan ke variabel menggunakan operator penugasan , Anda dapat menetapkan nilai variabel tersebut ke variabel lain menggunakan operator penugasan .

var myVar;

myVar = 5;

var myNum;

myNum = myVar;

Di atas mendeklarasikan myVarvariabel tanpa nilai, lalu memberinya nilai 5. Selanjutnya, variabel bernama myNumdideklarasikan tanpa nilai. Kemudian, isi myVar(yang 5) ditugaskan ke variabel myNum. Sekarang, myNumjuga memiliki nilai 5.

1. Inisialisasi variable dengan operator penugasan

Adalah umum untuk menginisialisasi variabel ke nilai awal di baris yang sama seperti yang dideklarasikan.

var myVar = 0;

Membuat variabel baru yang disebut myVardan memberinya nilai awal 0.

1. Deklarasi Variable String

Sebelumnya Anda menggunakan kode berikut untuk mendeklarasikan variabel:

var myName;

Tetapi Anda juga dapat mendeklarasikan variabel string seperti ini:

var myName = "your name";

"your name"disebut string literal. String literal, atau string, adalah serangkaian karakter nol atau lebih yang diapit oleh tanda kutip tunggal atau ganda.

1. Variabel yang tidak diinisialisasikan

Ketika variabel JavaScript dideklarasikan, mereka memiliki nilai awal undefined. Jika Anda melakukan operasi matematika pada suatu undefinedvariabel, hasilnya NaNadalah "Bukan Angka" . Jika Anda menggabungkan string dengan variabel , undefinedAnda akan mendapatkan string .undefined

1. Sensitivitas dalam variable

Dalam JavaScript semua variabel dan nama fungsi peka huruf besar/kecil. Ini berarti bahwa kapitalisasi penting.

MYVAR tidak sama dengan MyVarnor myvar. Dimungkinkan untuk memiliki beberapa variabel berbeda dengan nama yang sama tetapi casing berbeda. Sangat disarankan agar demi kejelasan, Anda tidak menggunakan fitur bahasa ini.

**Praktek terbaik**

Tulis nama variabel dalam JavaScript di camelCase . Dalam camelCase , nama variabel multi-kata memiliki kata pertama dalam huruf kecil dan huruf pertama dari setiap kata berikutnya dikapitalisasi.

**Contoh:**

var someVariable;

var anotherVariableName;

var thisVariableNameIsSoLong;

1. Perbedaan var dan let

Salah satu masalah terbesar dengan mendeklarasikan variabel dengan varkata kunci adalah Anda dapat dengan mudah menimpa deklarasi variabel:

var camper = "James";

var camper = "David";

console.log(camper);

Dalam kode di atas, campervariabel awalnya dideklarasikan sebagai James, dan kemudian diganti menjadi David. Konsol kemudian menampilkan string David.

Dalam aplikasi kecil, Anda mungkin tidak mengalami masalah seperti ini. Tetapi ketika basis kode Anda menjadi lebih besar, Anda mungkin secara tidak sengaja menimpa variabel yang tidak Anda inginkan. Karena perilaku ini tidak menimbulkan kesalahan, pencarian dan perbaikan bug menjadi lebih sulit.

Kata kunci yang disebut letdiperkenalkan di ES6, pembaruan besar untuk JavaScript, untuk mengatasi masalah potensial ini dengan varkata kunci. Anda akan belajar tentang fitur ES6 lainnya di tantangan selanjutnya.

Jika Anda mengganti vardengan letkode di atas, itu menghasilkan kesalahan:

let camper = "James";

let camper = "David";

Kesalahan dapat dilihat di konsol browser Anda.

Jadi tidak seperti var, ketika Anda menggunakan let, variabel dengan nama yang sama hanya dapat dideklarasikan satu kali.

1. Deklarasikan variable dengan const

Kata kunci letbukan satu-satunya cara baru untuk mendeklarasikan variabel. Di ES6, Anda juga dapat mendeklarasikan variabel menggunakan constkata kunci.

Const memiliki semua fitur luar biasa yang letdimiliki, dengan bonus tambahan bahwa variabel yang dideklarasikan menggunakan consthanya-baca. Mereka adalah nilai konstan, yang berarti bahwa sekali variabel ditetapkan dengan const, itu tidak dapat dipindahkan:

const FAV\_PET = "Cats";

FAV\_PET = "Dogs";

Konsol akan menampilkan kesalahan karena menetapkan kembali nilai FAV\_PET.

Anda harus selalu memberi nama variabel yang tidak ingin Anda tetapkan kembali menggunakan constkata kunci. Ini membantu ketika Anda secara tidak sengaja mencoba menetapkan kembali variabel yang dimaksudkan untuk tetap konstan.

**Catatan:** Pengembang biasanya menggunakan pengidentifikasi variabel huruf besar untuk nilai yang tidak dapat diubah dan huruf kecil atau camelCase untuk nilai yang dapat diubah (objek dan larik). Anda akan belajar lebih banyak tentang objek, array, dan nilai yang tidak dapat diubah dan yang dapat diubah dalam tantangan selanjutnya. Juga dalam tantangan selanjutnya, Anda akan melihat contoh pengidentifikasi variabel huruf besar, huruf kecil, atau camelCase.

1. Number

Numberadalah tipe data dalam JavaScript yang mewakili data numerik.

Sekarang mari kita coba menjumlahkan dua angka menggunakan JavaScript.

JavaScript menggunakan +simbol sebagai operator tambahan ketika ditempatkan di antara dua angka.

**Contoh:**

const myVar = 5 + 10;

myVarsekarang memiliki nilai 15.

1. Pengurangan (-)

Kami juga dapat mengurangi satu angka dari yang lain.

JavaScript menggunakan -simbol untuk pengurangan.

**Contoh**

const myVar = 12 - 6;

myVarakan memiliki nilai 6.

1. Perkalian Javascript (\*)

Kita juga dapat mengalikan satu angka dengan angka lainnya.

JavaScript menggunakan \*simbol untuk perkalian dua angka.

**Contoh**

const myVar = 13 \* 13;

myVarakan memiliki nilai 169.

1. Pembagaian Javascript (/)

Kita juga dapat membagi satu bilangan dengan bilangan lainnya.

JavaScript menggunakan /simbol untuk pembagian.

**Contoh**

const myVar = 16 / 2;

myVarsekarang memiliki nilai 8.

1. Operator Penugasan Penambahan(++)

Anda dapat dengan mudah menambah atau menambahkan satu ke variabel dengan ++operator.

i++;

adalah setara dengan

i = i + 1;

**Catatan:** Seluruh baris menjadi i++;, menghilangkan kebutuhan akan tanda sama dengan.

1. Operator Penugasan Pengurangan (--)

Anda dapat dengan mudah mengurangi atau mengurangi variabel satu per satu dengan --operator.

i--;

adalah setara dengan

i = i - 1;

**Catatan:** Seluruh baris menjadi i--;, menghilangkan kebutuhan akan tanda sama dengan.

1. Angka Desimal Javascript

Kita dapat menyimpan angka desimal dalam variabel juga. Bilangan desimal kadang-kadang disebut sebagai bilangan floating point atau float .

**Catatan:** saat Anda menghitung angka, angka tersebut dihitung dengan presisi yang terbatas. Operasi menggunakan floating point dapat menyebabkan hasil yang berbeda dari hasil yang diinginkan. Jika Anda mendapatkan salah satu dari hasil ini, buka topik di [forum freeCodeCamp](https://forum.freecodecamp.org/) .

1. Perkalian Bilangan Desimal di Javascript

Dalam JavaScript, Anda juga dapat melakukan perhitungan dengan angka desimal, seperti halnya bilangan bulat.

Mari kita kalikan dua desimal bersama-sama untuk mendapatkan produk mereka.

1. Pembagian Desimal di Javascript

Dalam JavaScript, Anda juga dapat melakukan perhitungan dengan angka desimal, seperti halnya bilangan bulat.

Mari kita kalikan dua desimal bersama-sama untuk mendapatkan produk mereka.

1. Operator Sisa (Remainder)

Operator *sisa* memberikan sisa %pembagian dua bilangan.

**Contoh**

5 % 2 = 1 karena  
Math.floor(5 / 2) = 2 (Pembagian)  
2 \* 2 = 4  
5 - 4 = 1 (Sisa)

**Penggunaan**  
Dalam matematika, suatu bilangan dapat diperiksa menjadi genap atau ganjil dengan memeriksa sisa pembagian bilangan tersebut dengan 2.

17 % 2 = 1 (17 Ganjil)  
48 % 2 = 0 (48 Genap)

1. Operasi Penambahan Matematika dan Penugasan Satu Langkah (+=) Augmented

Dalam pemrograman, biasanya menggunakan tugas untuk mengubah isi variabel. Ingatlah bahwa segala sesuatu di sebelah kanan tanda sama dengan dievaluasi terlebih dahulu, sehingga kita dapat mengatakan:

myVar = myVar + 5;

5 untuk menambah myVar. Karena ini adalah pola yang umum, ada operator yang melakukan operasi matematika dan penugasan dalam satu langkah.

Salah satu operator tersebut adalah +=operator.

let myVar = 1;

myVar += 5;

console.log(myVar);

6akan ditampilkan di konsol.

1. Operasi Pengurangan Matematika dan Penugasan Satu Langkah (-=) Augmented

Seperti +=operator, -=mengurangi angka dari variabel.

myVar = myVar - 5;

akan mengurangi 5dari myVar. Ini dapat ditulis ulang sebagai:

myVar -= 5;

1. Operasi Perkalian Matematika dan Penugasan Satu Langkah (\*=) Augmented

Operator \*=mengalikan variabel dengan angka.

myVar = myVar \* 5;

akan dikalikan myVardengan 5. Ini dapat ditulis ulang sebagai:

myVar \*= 5;

1. Operasi Pembagian Matematika dan Penugasan Satu Langkah (/=) Augmented

Operator /=membagi variabel dengan nomor lain.

myVar = myVar / 5;

Akan dibagi myVardengan 5. Ini dapat ditulis ulang sebagai:

myVar /= 5;

1. Kutipan Literal Dalam String

Saat Anda mendefinisikan string, Anda harus memulai dan mengakhiri dengan tanda kutip tunggal atau ganda. Apa yang terjadi ketika Anda membutuhkan kutipan literal: "atau 'di dalam string Anda?

Dalam JavaScript, Anda dapat menghindari kutipan dari menganggapnya sebagai akhir kutipan string dengan menempatkan garis miring terbalik ( \) di depan kutipan.

const sampleStr = "Alan said, \"Peter is learning JavaScript\".";

1. Mengutip String Dengan Kutip Tunggal

Nilai string dalam JavaScript dapat ditulis dengan tanda kutip tunggal atau ganda, selama Anda memulai dan mengakhiri dengan jenis tanda kutip yang sama. Tidak seperti beberapa bahasa pemrograman lain, tanda kutip tunggal dan ganda bekerja sama dalam JavaScript.

const doubleQuoteStr = "This is a string";

const singleQuoteStr = 'This is also a string';

Alasan mengapa Anda mungkin ingin menggunakan satu jenis kutipan di atas yang lain adalah jika Anda ingin menggunakan keduanya dalam sebuah string. Ini mungkin terjadi jika Anda ingin menyimpan percakapan dalam string dan percakapan dalam tanda kutip. Penggunaan lain untuk itu adalah menyimpan <a>tag dengan berbagai atribut dalam tanda kutip, semuanya dalam string.

const conversation = 'Finn exclaims to Jake, "Algebraic!"';

Namun, ini menjadi masalah jika Anda perlu menggunakan tanda kutip terluar di dalamnya. Ingat, sebuah string memiliki jenis kutipan yang sama di awal dan akhir. Tetapi jika Anda memiliki kutipan yang sama di suatu tempat di tengah, string akan berhenti lebih awal dan menimbulkan kesalahan.

const goodStr = 'Jake asks Finn, "Hey, let\'s go on an adventure?"';

const badStr = 'Finn responds, "Let's go!"';

Di sini badStrakan melempar kesalahan.

Dalam goodStr di atas, Anda dapat menggunakan kedua tanda kutip dengan aman dengan menggunakan garis miring terbalik \sebagai karakter pelarian.

**Catatan:** Garis miring terbalik \tidak boleh disamakan dengan garis miring ke depan /. Mereka tidak melakukan hal yang sama.

1. Escape Dalam String Javascript

Kutipan bukan satu-satunya karakter yang dapat *diloloskan* di dalam string. Ada dua alasan untuk menggunakan karakter pelolosan:

Untuk memungkinkan Anda menggunakan karakter, Anda mungkin tidak dapat mengetik, seperti baris baru.

Untuk memungkinkan Anda merepresentasikan beberapa tanda kutip dalam sebuah string tanpa JavaScript salah mengartikan apa yang Anda maksud.

Kami mempelajari ini di tantangan sebelumnya.

| **Kode** | **Keluaran** |
| --- | --- |
| \' | kutipan tunggal |
| \" | kutipan ganda |
| \\ | garis miring terbalik |
| \n | garis baru |
| \t | tab |
| \r | Kereta kembali |
| \b | batas kata |
| \f | umpan formulir |

*Perhatikan bahwa garis miring terbalik itu sendiri harus diloloskan agar dapat ditampilkan sebagai garis miring terbalik.*

1. Menggabung String Dengan Operator (+)

Dalam JavaScript, ketika +operator digunakan dengan Stringnilai, itu disebut operator gabungan . Anda dapat membuat string baru dari string lain dengan menggabungkannya*.*

**Contoh**

'My name is Alan,' + ' I concatenate.'

**Catatan:** Perhatikan spasi. Penggabungan tidak menambahkan spasi di antara string yang digabungkan, jadi Anda harus menambahkannya sendiri.

Contoh:

const ourStr = "I come first. " + "I come second.";

String I come first. I come second.akan ditampilkan di konsol.

1. Menggabung String Dengan Operator (+=)

Kita juga dapat menggunakan +=operator untuk menggabungkan string ke akhir variabel string yang ada. Ini bisa sangat membantu untuk memutus tali panjang pada beberapa baris.

**Catatan:** Perhatikan spasi. Penggabungan tidak menambahkan spasi di antara string yang digabungkan, jadi Anda harus menambahkannya sendiri.

Contoh:

let ourStr = "I come first. ";

ourStr += "I come second.";

ourStrsekarang memiliki nilai string I come first. I come second.

1. Membangun String Dengan Variable Menggunakan Opertator (+)

Terkadang Anda perlu membuat string. Dengan menggunakan operator penggabungan ( +), Anda dapat menyisipkan satu atau beberapa variabel ke dalam string yang sedang Anda buat.

Contoh:

const ourName = "freeCodeCamp";

const ourStr = "Hello, our name is " + ourName + ", how are you?";

ourStrakan memiliki nilai string Hello, our name is freeCodeCamp, how are you?.

1. Menambahkan Variable ke String Menggunakan Opertator (+=)

Sama seperti kita dapat membuat string pada beberapa baris dari string literal , kita juga dapat menambahkan variabel ke string menggunakan +=operator plus equals ( ).

Contoh:

const anAdjective = "awesome!";

let ourStr = "freeCodeCamp is ";

ourStr += anAdjective;

ourStrakan memiliki nilai freeCodeCamp is awesome!.

1. Mencari Length

Anda dapat menemukan panjang suatu Stringnilai dengan menulis .lengthsetelah variabel string atau literal string.

console.log("Alan Peter".length);

Nilai 10akan ditampilkan di konsol. Perhatikan bahwa karakter spasi antara "Alan" dan "Peter" juga dihitung.

Misalnya, jika kita membuat variabel const firstName = "Ada", kita bisa mengetahui berapa panjang string Adadengan menggunakan firstName.lengthproperti.

1. Notasi Bracket untuk menemukan Array dalam Array

Notasi kurung adalah cara untuk mendapatkan karakter pada indeks tertentu dalam sebuah string.

Sebagian besar bahasa pemrograman modern, seperti JavaScript, tidak mulai menghitung pada 1 seperti yang dilakukan manusia. Mereka mulai dari 0. Ini disebut sebagai pengindeksan berbasis Nol .

Misalnya, karakter pada indeks 0 dalam kata Charlesadalah C. Jadi jika const firstName = "Charles", Anda bisa mendapatkan nilai huruf pertama dari string dengan menggunakan firstName[0].

Contoh:

const firstName = "Charles";

const firstLetter = firstName[0];

firstLetterakan memiliki nilai string C.

1. Memahami Kekekalan String

Dalam JavaScript, Stringnilai tidak dapat diubah , yang berarti nilai tersebut tidak dapat diubah setelah dibuat.

Misalnya, kode berikut akan menghasilkan kesalahan karena huruf Bdalam string Bobtidak dapat diubah menjadi huruf J:

let myStr = "Bob";

myStr[0] = "J";

Perhatikan bahwa ini tidak berarti bahwa myStrtidak dapat ditugaskan kembali. Satu-satunya cara untuk mengubahnya myStradalah dengan menetapkannya dengan nilai baru, seperti ini:

let myStr = "Bob";

myStr = "Job";

1. Gunakan Notasi Bracket untuk Menemukan Karakter ke-N dalam sebuah String

Anda juga dapat menggunakan notasi braket untuk mendapatkan karakter pada posisi lain dalam sebuah string.

Ingat bahwa komputer mulai menghitung pada 0, jadi karakter pertama sebenarnya adalah karakter ke nol.

Contoh:

const firstName = "Ada";

const secondLetterOfFirstName = firstName[1];

secondLetterOfFirstNameakan memiliki nilai string d.

1. Gunakan Notasi Bracket untuk Menemukan Karakter Terakhir dalam sebuah String

Untuk mendapatkan huruf terakhir dari sebuah string, Anda dapat mengurangi satu dari panjang string.

Misalnya, jika const firstName = "Ada", Anda bisa mendapatkan nilai huruf terakhir dari string dengan menggunakan firstName[firstName.length - 1].

Contoh:

const firstName = "Ada";

const lastLetter = firstName[firstName.length - 1];

lastLetterakan memiliki nilai string a.

1. Gunakan Notasi Bracket untuk Menemukan Karakter Nth-to-Last dalam sebuah String

Anda dapat menggunakan prinsip yang sama yang baru saja kita gunakan untuk mengambil karakter terakhir dalam sebuah string untuk mengambil karakter Nth-to-last.

Misalnya, Anda bisa mendapatkan nilai huruf ketiga hingga terakhir dari const firstName = "Augusta"string dengan menggunakanfirstName[firstName.length - 3]

Contoh:

const firstName = "Augusta";

const thirdToLastLetter = firstName[firstName.length - 3];

thirdToLastLetterakan memiliki nilai string s.

1. Kata Kosong

Anda diberikan kalimat dengan beberapa kata yang hilang, seperti kata benda, kata kerja, kata sifat dan kata keterangan. Anda kemudian mengisi bagian yang hilang dengan kata-kata pilihan Anda sedemikian rupa sehingga kalimat yang lengkap masuk akal.

Pertimbangkan kalimat ini - Itu benar-benar **\_\_\_\_** , dan kami **\_\_\_\_** sendiri **\_\_\_\_** . Kalimat ini memiliki tiga bagian yang hilang- kata sifat, kata kerja dan kata keterangan, dan kita dapat menambahkan kata-kata pilihan kita untuk melengkapinya. Kami kemudian dapat menetapkan kalimat lengkap ke variabel sebagai berikut:

const sentence = "It was really " + "hot" + ", and we " + "laughed" + " ourselves " + "silly" + ".";

1. Simpan Beberapa Nilai dalam satu Variabel menggunakan Array JavaScript

Dengan variabel JavaScript array, kita dapat menyimpan beberapa bagian data di satu tempat.

Anda memulai deklarasi array dengan kurung siku pembuka, akhiri dengan kurung siku penutup, dan beri tanda koma di antara setiap entri, seperti ini:

const sandwich = ["peanut butter", "jelly", "bread"];

1. Sarang satu Array di dalam Array Lain

Anda juga dapat menyarangkan array di dalam array lain, seperti di bawah ini:

const teams = [["Bulls", 23], ["White Sox", 45]];

Ini juga disebut array multidimensi .

1. Akses Data Array dengan Indeks

Kita dapat mengakses data di dalam array menggunakan *indexes* .

Indeks array ditulis dalam notasi braket yang sama dengan yang digunakan string, kecuali bahwa alih-alih menentukan karakter, mereka menentukan entri dalam array. Seperti string, array menggunakan pengindeksan *berbasis nol* , sehingga elemen pertama dalam array memiliki indeks 0.

**Contoh**

const array = [50, 60, 70];

console.log(array[0]);

const data = array[1];

console.log(array[0])Mencetak , 50dan datamemiliki nilai 60.

1. Ubah Data Array Dengan Indeks

Tidak seperti string, entri array dapat berubah dan dapat diubah secara bebas, bahkan jika array dideklarasikan dengan const.

**Contoh**

const ourArray = [50, 40, 30];

ourArray[0] = 15;

ourArraysekarang memiliki nilai [15, 40, 30].

**Catatan:** Seharusnya tidak ada spasi antara nama array dan tanda kurung siku, seperti array [0]. Meskipun JavaScript dapat memproses ini dengan benar, ini dapat membingungkan pemrogram lain yang membaca kode Anda.

1. Akses Array Multi-Dimensi Dengan Indeks

Salah satu cara untuk memikirkan array multi-dimensi , adalah sebagai array dari array . Saat Anda menggunakan tanda kurung untuk mengakses larik Anda, kumpulan tanda kurung pertama merujuk ke entri di larik terluar (tingkat pertama), dan setiap pasangan tanda kurung tambahan merujuk ke tingkat entri berikutnya di dalamnya.

**Contoh**

const arr = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9],

[[10, 11, 12], 13, 14]

];

const subarray = arr[3];

const nestedSubarray = arr[3][0];

const element = arr[3][0][1];

Dalam contoh ini, subarraymemiliki nilai [[10, 11, 12], 13, 14], nestedSubarraymemiliki nilai [10, 11, 12], dan elementmemiliki nilai 11.

**Catatan:** Seharusnya tidak ada spasi antara nama array dan tanda kurung siku, seperti array [0][0]dan bahkan ini array [0] [0]tidak diperbolehkan. Meskipun JavaScript dapat memproses ini dengan benar, ini dapat membingungkan pemrogram lain yang membaca kode Anda.

1. Memanipulasi Array Dengan Push()

Cara mudah untuk menambahkan data ke akhir array adalah melalui push()fungsi.

.push()mengambil satu atau lebih parameter dan "mendorong" mereka ke akhir array.

Contoh:

const arr1 = [1, 2, 3];

arr1.push(4);

const arr2 = ["Stimpson", "J", "cat"];

arr2.push(["happy", "joy"]);

arr1sekarang memiliki nilai [1, 2, 3, 4]dan arr2memiliki nilai ["Stimpson", "J", "cat", ["happy", "joy"]].

1. Memanipulasi Array Dengan pop()

Cara lain untuk mengubah data dalam array adalah dengan .pop()fungsi.

.pop()digunakan untuk mengeluarkan nilai dari akhir array. Kami dapat menyimpan nilai yang muncul ini dengan menetapkannya ke variabel. Dengan kata lain, .pop()menghapus elemen terakhir dari array dan mengembalikan elemen itu.

Jenis entri apa pun dapat dikeluarkan dari array - angka, string, bahkan array bersarang.

const threeArr = [1, 4, 6];

const oneDown = threeArr.pop();

console.log(oneDown);

console.log(threeArr);

Yang pertama console.logakan menampilkan nilai 6, dan yang kedua akan menampilkan nilai [1, 4].

1. Memanipulasi Array Dengan shift()

pop()selalu menghapus elemen terakhir dari array. Bagaimana jika Anda ingin menghapus yang pertama?

Di situlah .shift()masuk. Ia bekerja seperti .pop(), kecuali menghapus elemen pertama, bukan yang terakhir.

Contoh:

const ourArray = ["Stimpson", "J", ["cat"]];

const removedFromOurArray = ourArray.shift();

removedFromOurArrayakan memiliki nilai string Stimpson, dan ourArrayakan memiliki ["J", ["cat"]].

1. Memanipulasi Array Dengan unshift()

Anda tidak hanya dapat shiftelemen dari awal array, Anda juga dapat unshiftelemen ke awal array yaitu menambahkan elemen di depan array.

.unshift()bekerja persis seperti .push(), tetapi alih-alih menambahkan elemen di akhir array, unshift()tambahkan elemen di awal array.

Contoh:

const ourArray = ["Stimpson", "J", "cat"];

ourArray.shift();

ourArray.unshift("Happy");

Setelah shift, ourArrayakan memiliki nilai ["J", "cat"]. Setelah unshift, ourArrayakan memiliki nilai ["Happy", "J", "cat"].

1. Daftar belanja

Buat daftar belanja di variabel myList. Daftar tersebut harus berupa array multidimensi yang berisi beberapa sub-array.

Elemen pertama di setiap sub-array harus berisi string dengan nama item. Elemen kedua harus berupa angka yang mewakili kuantitas yaitu

["Chocolate Bar", 15]

1. Kesimpulan

Penerapan mengenai javascript dalam hal ini sangatlah penting dikarenakan untuk pemahaman. Sehingga diperlukan praktikum mengenai dasar mengenai konsep logika javascript.

Sehingga dengan mudah dapat memahami bagaimana aturan-aturan kode dalam javascript. Penggunaan metode dalam praktikum ini sangat membantu dalam memahami konsep logika javascript.

Dengan media praktikum yang sederhana dan jelas sangat membantu dalam memahami begitu banyak kode-kode javascript yang perlu dipahami dari perkuliahan permprograman 1.

Daftar Pustaka

Zia, M. P., Kusumo, D. S., & Richasdy, D. (2021). Konsistensi Kode Pada Bahasa Pemrograman Javascript Menggunakan Linter Pada Continuous Integration Pipeline. *eProceedings of Engineering*, *8*(2).

Alamsyah, A. (2003). Pengantar javascript. *Kuliah Umum IlmuKomputer. Com*, *40*.

Sianipar, R. H. (2017). *Dasar Pemrograman JavaScript: Langkah demi Langkah* (Vol. 1). Penerbit ANDI.

Siahaan, V., & Sianipar, R. H. (2018). *JavaScript: Dari A Sampai Z* (Vol. 1). sparta publisher.

Enterprise, J. (2017). *Otodidak Pemrograman JavaScript*. Elex Media Komputindo.

Rohim, W. N., Awaluddin, M., & Suprayogi, A. (2015). Semarang Charity Map, Penyajian Peta Donasi Sosial Kota Semarang Berbasis Blogger Javascript. *Jurnal Geodesi Undip*, *4*(2), 117-130.

Edy Winarno, S. T., & Zaki, A. (2014). *Pemrograman Web Berbasis Html 5, php, dan Javascript*. Elex Media Komputindo.