|  |  |
| --- | --- |
| Materi | Nilai |
| Pembelajaran Javascript Video ke 21 - 30  (Tugas semester 4 #3) | 95 |

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

1. map() adalah metode ini mengambil bagian dari array dan outputnya adalah sebuah array juga.

Cth :

Penulisan menggunakan function anonymous :

let siswa = nilai.map(function (a) {

*return* a.nama;

});

console.log(siswa);

Artinya, pada variabel siswa memanggil variabel nilai dan di dalamnya menjalankan array map, yang berisi function dengan parameter a, di dalam a ini menyimpan nilai dari variabel nilai, setelah itu, data yang dikeluarkan adalah variabel a dengan memilih objek nama, nilai yang dipanggil akan dikeluarkan menggunakan return. Lalu variabel siswa ditampilkan menggunakan console.log.

Penulisan menggunakan arrowfunction :

let siswa = nilai.map(a => a.nama);

console.log(siswa);

Artinya, di dalam variabel siswa terdapat array map yang berisi arrowfunction dengan variabel a untuk menyimpan nilai dari variabel nilai, dan variabel a memilih objek nama yang akan ditampilkan. Jadi objek nama yang sudah dipilih akan ditampilkan dengan bentuk array. Lalu akan ditampilkan dengan menggunakan console.log.

1. sort() digunakan untuk mengembalikan nilai array dalam bentuk berurutan.

Cth :

mapel.sort();

console.log(mapel);

Artinya, di dalam variabel mapel terdapat sort() yang digunakan untuk mengurutkan nilai dari variabel mapel, sehingga ketika ditampilkan, maka nilai dari variabel maple adalah urut.

1. reduce() digunakan untuk menjumlahkan elemen array, di dalam reduce terdapat 3 bagian, yaitu akumulator (untuk menyimpan), item (untuk menentukan elemen yang akan dijumlahkan), index(yaitu posisi untuk memulai).

Cth :

Penulisan menggunakan function anonymous :

let total = nilai.reduce(function (a, b) {

*return* hasil = a + b.ipa;

}, 0);

console.log(total);

Artinya, variabel total berisi variabel nilai yang menggunakan method reduce yaitu untuk menjumlahkan elemen array dengan function yang parameternya berisi a (akumulator) dan b (item). Setelah itu, hasil berisi value dari reduce yaitu a + b dengan kolom ipa yang dipilih, dan index 0, berarti posisi penjumlahan dimulai dari 0. Lalu, variabel total ditampilkan menggunakan console.log.

Penulisan menggunakan arrowfunction :

let value = nilai.reduce((a, b) => (a += b.matematika), 0);

console.log(value);

Artinya, pada variabel value, variabel nilai menggunakan method reduce untuk menjumlahkan elemen array, dengan function yang parameternya diisi a (akumulator) dan b (item), setelah itu akan melakukan proses reduce yaitu a sama dengan a ditambah dengan b kolom matematika, dan index nya dimulai dari 0, variabel value ditampilkan menggunakan console.log.

1. Menampilkan menu – menu (#PROJEK SHOPPING-CART 1) :

* Pertama, membuat file index.html terlebih dahulu, setelah itu, menyambungkan nfile html dengan file javascript. Di dalam <body>, terdapat div dengan class produk.
* Kedua, di dalam file javascript.js, pertama membuat variabel array yang di dalamnya berisi objek dengan nama tblmenu, setelah itu, di setiap objek diisi idmenu, idkategori, menu, gambar, dan harga. Setelah mendeskripsikan objek – objek di dalam tblmenu, maka menggunakan map() untuk mengambil bagian dari array. Di dalam variabel tampil ini terdapat tblmenu yang menggunakan array map() berisi function dengan parameter variabel kolom, jadi disini variabel kolom itu mengambil semua nilai yang ada pada tblmenu. Setelah itu, nilai dikeluarkan menggunakan return. Untuk penggunakan html di dalam javascript menggunakan backtiks, lalu pada div image, mengambil nilai dari ${kolom.gambar}, artinya mengambil nilai dari objek gambar pada variabel kolom. Pada div title, mengambil nilai dari ${kolom.menu}, artinya mengambil nilai dari objek menu pada variabel kolom. Dan pada div harga, mengambil nilai dari ${kolom.harga}, artinya mengambil nilai dari objek harga pada variabel kolom.
* Ketiga, untuk memasukkan tulisan pada javascript ke dalam class produk di dalam html, maka, di dalam variabel isi berisi document.querySelector(“.produk”).innerHTML = tampil. Artinya, querySelector memilih class produk dan innerHTML akan memasukkan tulisan dari variabel tampil.

1. data- =”” digunakan untuk memberi data pada button pada HTML. Setelah data- boleh diisi dengan nama dari data. Dan pemanggilan data pada javascript menggunakan (namavariabel.dataset.namadata).

Cth :

* Di dalam HTML :

<div class="btn-beli">

          <button data-id="50">Beli</button>

 </div>

Artinya, di dalam div dengan class btn-beli terdapat button Beli yang data nya diisi dengan menggunakan data-id bernilai 50.

* Di dalam javascript

let btnbeli = document.querySelector(".btn-beli > button");

btnbeli.onclick = function () {

    console.log(btnbeli.dataset.id);

}

Artinya, terdapat variabel btnbeli dan querySelector memilih button di dalam class btn-beli. Lalu variabel btnbeli menggunakan event onclick yang jika button Beli di klik, maka akan menjalankan function berisi menampilkan nilai dari dataset dengan nama id pada variabel btnbeli, dan ditampilkan menggunakan console.log.

1. Memilih id menu dan menampilkannya ke dalam console.log dengan menggunakan for. (PROJEK SHOPPING-CART #2)

Cth :

let btnbeli = document.querySelectorAll(".btn-beli > button");

let cart = [];

*for* (let index = 0; index < btnbeli.length; index++) {

    btnbeli[index].onclick = function () {

*//console.log(btnbeli[index].dataset["idmenu"]);*

        cart.push(btnbeli[index].dataset["idmenu"]);

    };

};

console.log(cart);

Artinya, pada variabel btnbeli querySelectorAll memilih semua data yang ada di dalam button pada class btn.beli. setelah itu, terdapat variabel cart yang berisi array kosong. Karena button ini berisi banyak data, maka untuk mengambil data nya digunakan for dengan kondisi, yaitu variabel index dimulai dari 0, index kurang dari variabel btnbeli, index bernilai ++, setiap pengulangan maka nilai dari index akan bertambah 1. Lalu di dalam for, terdapat event onclick jika btnbeli[index] di klik, maka akan menjalankan function yang berisi, variabel cart yang menjalankan array push(), yaitu menambahkan elemen array pada akhir dari array, dan memanggil btnbeli[index] dengan dataset bernama idmenu. Lalu, cart akan ditampilkan menggunakan console.log. Jadi, ketika setiap button di klik, maka nilai dari idmenu akan muncul mengisi variabel cart yang kosong.

1. Penggunaan filter untuk menampilkan menu sesuai dengan idmenu (PROJEK SHOPPING-CART #3).

Cth :

tblmenu.filter(function (a) {

*if* (a.idmenu == btnbeli[index].dataset["idmenu"]) {

                cart.push(a);

                console.log(cart);

            }

        });

Artinya, pada variabel tblmenu menggunakan event filter yang digunakan untuk menyaring objek mana yang akan digunakan. Lalu di dalam function pada filter, wadah yang digunakan untuk menyimpan nilai yang diambil dari tblmenu, setelah itu terdapat perbandingan, yaitu apabila idmenu yang berada di dalam a nilai nya sama dengan btnbeli[index] dengan dataset[idmenu] maka nilai dari button yang di klik akan disimpan di dalam cart, lalu jika button di klik lagi, maka array push() akan menambahkan elemen baru pada bagian akhir array yang menjalankan a. Lalu, variabel cart akan ditampilkan di dalam console.log.

1. Localstorage adalah fungsi di javascript yang menyimpan data di dalam browser, selama nilai pada localStorage belum dihapus, maka nilai nya akan selalu ada di dalam browser walaupun browser sudah ditutup / direstart. Penggunaan local storage :

* localStorage.setItem(), digunakan untuk menset localstorage.
* localStorage.getItem(), digunakan untuk mendapatkan value.
* localStorage.removeItem(), digunakan untuk menghapus nilai storage.
* localStorage.clear(), digunakan untuk menghapus semua nilai storage.
* localStorage.key(index), digunakan untuk menampilkan key sesuai dengan index yang dituliskan.
* localStorage.length, digunakan untuk melihat berapa jumlah key yang dibuat menggunakan localStorage.

1. localStorage.setItem digunakan untuk menset / mengatur localstorage, di dalam fungsi harus mempunyai key dan value.

Cth :

localStorage.setItem("Tes", "Ayok kejar deadline!");

//mengatur localstorage dengan key tes dan value Ayok kejar deadline!

1. localStorage.getItem digunakan untuk mendapatkan value, untuk mendapatkan value, maka di dalam fungsi pada getItem diberikan key saja.

Cth :

console.log(localStorage.getItem("Tes"));

//mendapatkan value pada localStorage dengan memanggil key dan menampilkan menggunakan console.log.

1. localStorage.removeItem digunakan untuk menghapus nilai pada localStorage, di dalam fungsi hanya dituliskan key yang ingin dihapus saja.

Cth :

localStorage.removeItem("coba");

//menghapus key dan value pada localStorage dengan menuliskan key yang ingin dihapus saja.

1. localStorage.clear digunakan untuk menghapus semua storage pada localStorage.

Cth :

localStorage.clear();

//menghapus semua storage yang ada pada localStorage.

1. localStorage.key(index) digunakan untuk menampilkan key sesuai dengan index yang dituliskan.

Cth :

console.log(localStorage.key(0));

//menampilkan key pada index 0

1. sessionStorage adalah fungsi di javascript yang menyimpan data di dalam browser, nilai pada sessionStorage akan tetap ada ketika browser dinyalakan, dan nilai pada sessionStorage akan hilang ketika browser dimatikan. Penggunaan session storage :

* sessionStorage.setItem(), digunakan untuk menset localstorage.
* sessionStorage.getItem(), digunakan untuk mendapatkan value.
* sessionStorage.removeItem(), digunakan untuk menghapus nilai storage.
* sessionStorage.clear(), digunakan untuk menghapus semua nilai storage.
* sessionStorage.key(index), digunakan untuk menampilkan key sesuai dengan index yang dituliskan.
* sessionStorage.length, digunakan untuk melihat berapa jumlah key yang dibuat menggunakan localStorage.

1. sessionStorage.setItem digunakan untuk menset / mengatur localstorage, di dalam fungsi harus mempunyai key dan value.

Cth :

sessionStorage.setItem("Tes", "Ayok kejar deadline!");

//mengatur localstorage dengan key tes dan value Ayok kejar deadline!

1. sessionStorage.getItem digunakan untuk mendapatkan value, untuk mendapatkan value, maka di dalam fungsi pada getItem diberikan key saja.

Cth :

console.log(sessionStorage.getItem("Tes"));

//mendapatkan value pada sessionStorage dengan memanggil key dan menampilkan menggunakan console.log.

1. sessionStorage.removeItem digunakan untuk menghapus nilai pada sessionStorage, di dalam fungsi hanya dituliskan key yang ingin dihapus saja.

Cth :

sessionStorage.removeItem("coba");

//menghapus key dan value pada sessionStorage dengan menuliskan key yang ingin dihapus saja.

1. sessionStorage.clear digunakan untuk menghapus semua storage pada sessionStorage.

Cth :

sessionStorage.clear();

//menghapus semua storage yang ada pada sessionStorage.

1. sessionStorage.key(index) digunakan untuk menampilkan key sesuai dengan index yang dituliskan.

Cth :

console.log(sessionStorage.key(0));

//menampilkan key pada index 0

1. JSON (Javascript Object Notation) digunakan sebagai format data untuk pertukaran antar aplikasi. Penulisan json mengikuti format dari javascript dan sering digunakan pada AJAX. Bisa digunakan [ada berbagai jenis pemrograman dan sering digunakan untuk pengaturan / setting aplikasi.
2. Data yang ada pada JSON berupa :

* String
* Number
* Boolean
* Array
* Object
* Null

1. Aturan penulisan dari JSON :

* Ditulis berpasangan antara key dan value.
* Key harus diapit tanda petik dua (“key”).
* Value harus menunjukkan tipe data yang digunakan.
* Nama file mempunyai ekstensi json.
* Penulisan data di dalam json apabila lebih dari satu, maka akhir dari data diberi koma, dan jika data sudah terakhir, maka akhir dari data nya tidak perlu diberi koma.
* Apabila di dalam json terdapat objek, maka objek harus diletakkan di dalam array, dan setiap kurung kurawal tutup diberi koma untuk membedakan objek satu dengan objek lainnya.
* JSON dapat ditampilkan menggunakan console.log(namavariabel[index].namakolom);

1. Untuk pengecekan penulisan json dapat dicek di website, jsonlim.com, untuk mengecek code yang dibuat sudah benar atau salah.
2. Fetch digunakan untuk melakukan network request.

Ada 3 proses di dalam fetch :

* Pertama, fetch akan mengambil url.
* Menjadikan file dalam bentuk json.
* Menampilkan data.

Cth :

function hasil() {

    let link = "js/tblmenu.json";

    fetch(link)

    .then(function (respon) {

*return* respon.json();

    })

    .then(function (data) {

        console.log(data);

    })

};

Artinya, di dalam function hasil berisi variabel link yang memanggil file tblmenu.json, setelah itu terdapat fetch akan menjalankan variabel link yang memanggil url, lalu menjadikan file dalam bentuk apa, yaitu function yang paramternya berisi respon, nilai nya akan direturn dan dijadikan file nya dalam bentuk respon.json. Apabila ingin menampilkan data, maka harus terdapat variabel yang digunakan untuk menampung data. Dan data yang sudah berbentuk json ditampilkan dalam console.log yang memanggil variabel data.

1. Async digunakan untuk menangani hasil dari sebuah promise. Untuk menggunakan async, kita harus menambahkan kata async di depan. Asynchronous bisa berjalan bersamaan dan ketika sudah lengkap akan kembali lagi prosesnya. Pemanggilan disarankan menggunakan asynchronous.
2. Await berfungsi untuk menunda sebuah kode dijalankan sampai proses asynchronous berhasil.
3. Untuk melihat status penggunaan dapat mengecek ke dalam web developer.mozilla.org

**Saya Belum Mengerti**