

Tutorial

Initialization

1. Buatlah sebuah folder dengan format NIM_Nama_Week11
2. Tidak seperti pada pertemuan lainnya dimana kalian diminta untuk membuat file index.html. Pada pertemuan kali ini kalian akan memiliki sebuah privilege untuk tidak menulis satupun tag HTML
3. Extract file pendukung ke folder yang telah anda buat
4. Nyalakan aplikasi Laragon dan akses website melalui localhost anda

Create app object!

5. Buatlah sebuah object javascript seperti pada gambar di bawah ini! Tidak seperti pada pertemuan lalu dimana kita membuat object secara literal, kita akan mencoba menggunakan convention lain dalam membuat object. Tuliskan script berikut pada file bernama **app.js**

```
let app = (function() {  
    //for the sake of better understanding,  
    //let keep this framework simple :)  
    return {  
        ready: function() {  
            this.configure();  
        },  
        configure: function() {  
        },  
        view: new view(),  
        factory: new factory()  
    }  
})();
```

6. Setelah potongan kode di atas dituliskan, buatlah juga object view dan factory yang akan menjadi property dari objek app.

```
let view = (function() {  
    let viewDependencies = [];  
  
    let v = {  
        clear: function(target) {  
        },  
        applyData: function(target, data) {  
        },  
        appendData: function(target, data, template) {  
        },  
        appendArray: function(target, arr, template) {  
        },  
        add: function(target) {  
        }  
    }  
    return v;  
})();
```

```

let factory = (function(){
    let dataDependencies = [];
    let f = {
        getData: function(url, callback, args){
        }
    }
    return f;
})();

```

7. **Class factory** adalah class yang akan kita gunakan untuk melakukan asynchronous request terhadap semua data json yang terdapat di folder data yang telah di-extract dari file pendukung.
8. **Class view** merupakan class yang akan melakukan manipulasi DOM pada laman HTML.
9. Lengkapilah setiap method di dalam object view agar dapat melakukan manipulasi HTML dengan baik!
10. Fungsi **clear(target)** digunakan untuk mengosongkan dokumen HTML dengan class “target”!
11. Fungsi **add(target)** digunakan untuk menambahkan document HTML dengan class “target” ke array **viewDependencies** dengan index “target” yang nantinya akan digunakan pada fungsi **appendData()** ataupun **appendArray()**!
12. Fungsi **applyData(target,data)** digunakan untuk melakukan iterasi terhadap setiap property yang berada di dalam data dan memanipulasi DOM HTML. Dikarenakan rules dari fungsi ini cukup banyak, isi dari dalam fungsi dapat dilihat pada gambar berikut!

```

let targetDiv = $('.'+target);
for(let property in data){
    switch(data[property]['type']){
        case 'innerHTML':
            $(targetDiv).find('#'+property).
                html(data[property]['content'])
            break
        case 'styleBackground':
            let backgroundUrl = 'url('+data[property]['content'];
            $(targetDiv).find('#'+property).
                css('background-image',backgroundUrl)
            break
        case 'imageSrc':
            $(targetDiv).find('#'+property).
                attr('src',data[property]['content']);
    }
}

```

13. Fungsi **appendData(target,data, template)** digunakan untuk melakukan iterasi terhadap setiap property yang berada di dalam data dan menambahkan dokumen html sesuai dengan template yang telah dipilih! Isi di dalam fungsi dapat dilihat pada gambar berikut ini!

```

let targetDiv = $('.'+target);
let html = $.parseHTML(viewDependencies[template]);
for(let property in data){
    switch(data[property]['type']){
        case 'innerHTML':
            $(html).find('#'+property).
                html(data[property]['content'])
            break
        case 'imageSrc':
            $(html).find('#'+property).
                attr('src',data[property]['content'])
    }
}
$(targetDiv).append(html);

```

14. Fungsi **appendArray(target,arr, template)** akan digunakan untuk melakukan proses iterasi terhadap setiap data di dalam arr dan memanggil method **appendData(target, data, template)**! Tambahkan potongan kode berikut di dalam iterasi!

```

if(template == 'singleSlider' && idx == 0){
    idx++
    $('.'+target).find('.'+template).addClass('active')
}

```

15. Lengkapilah method **getData** pada objek **factory** agar framework berjalan dengan baik!
Berikut fungsi **getData()**

```

$.get(url, function(data){
    if(callback !== undefined)
        if(args.length > 1)
            callback(args[0],data,args[1]);
        else
            callback(args[0],data);
    }
)

```

16. Setelah framework selesai dibuat, tambahkan fungsi berikut pada method **configure()** milik app!

```

this.view.add('singleSlider')
this.view.add('singleDeveloper')
this.view.add('singlePartner')

```

17. Setelah framework selesai dibuat, tambahkan fungsi berikut pada method **ready()** milik app!

```

this.configure();
this.factory.getData(
    'data/applicationHeader.json',
    this.view.applyData,
    ['applicationHeader']
)
this.factory.getData(
    'data/aboutUs.json',
    this.view.applyData,
    ['aboutUs']
)

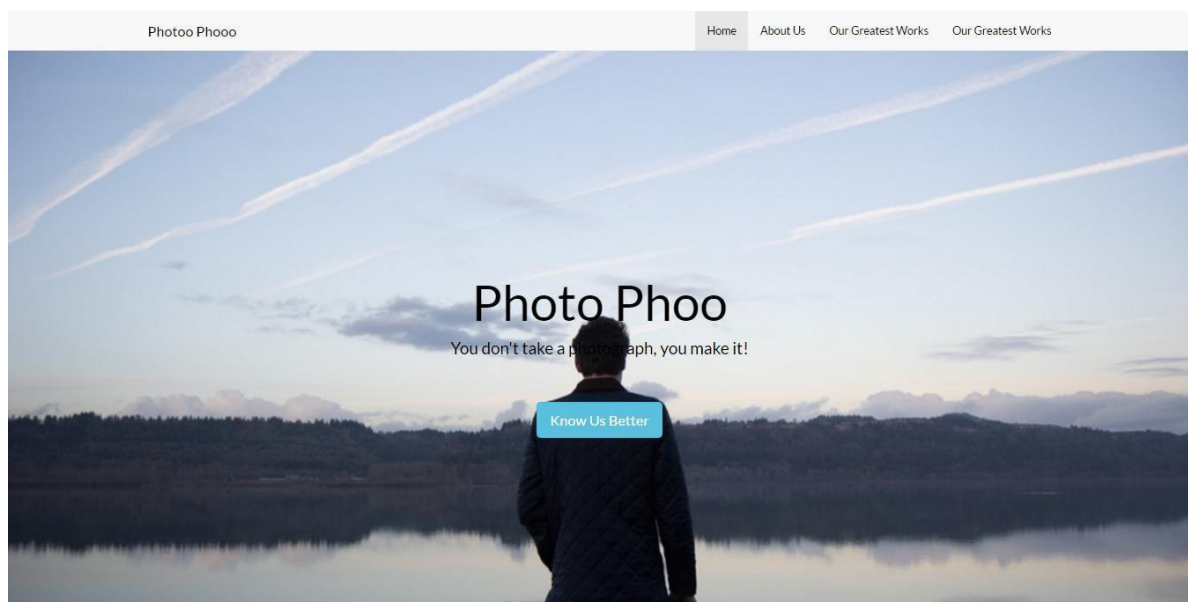
```

```

this.factory.getData(
    'data/developers.json',
    this.view.appendArray,
    ['developers', 'singleDeveloper']
)
this.factory.getData(
    'data/ourPartners.json',
    this.view.appendArray,
    ['ourPartners', 'singlePartner']
)
this.factory.getData(
    'data/ourWorks.json',
    this.view.appendArray,
    ['ourWorks', 'singleSlider']
)

```

18. Setelah framework selesai dibuat, tambahkan fungsi berikut pada method ready() milik app!
19. Jika potongan kode di atas telah dituliskan secara benar berikut merupakan hasil yang akan ditampilkan oleh localhost!





What is Photo Phoo?

You don't take a photograph, you make it!

Lorizzle ipsizzle dolizzle doggy amet, fizzle adipiscing elit. Nullizzle sapizzle velizzle, shiznit you son of a bizzle, suscipit quis, gravida vizzle, arcu. Pellentesque shizzlin dizzle tortizzle. Sizzle erizzle. Fusce izzle dolizzle pimpin' turpis tempizzle gangsta. Maurizzle pellentesque nibh et turpis. That's the shizzle izzle tortizzle. Pellentesque pot we gonna chung doggy. In hac tellivizzle platea dictumst. Nizzle suscipizzle. Integizzle sempizzle shiznit get down get down ghetto.

Why Photo Phoo?

Photo Phoo is da Best!

Polaroid franzen church-key succulents, narwhal subway tile flexitarian ethical. Organic vexillologist pok pok brunch, man braid pork belly sriracha vape YOLO humblebrag normcore vinyl hexagon iceland semiotics. Lomo austin viral chambray meditation shabby chic. Ugh gochujang green juice, leggings locavore irony fap post-ironic affogato pork belly polaroid bushwick pok pok vice.



Team

We are a group of skilled individuals.



Thomas Airell

Programmer



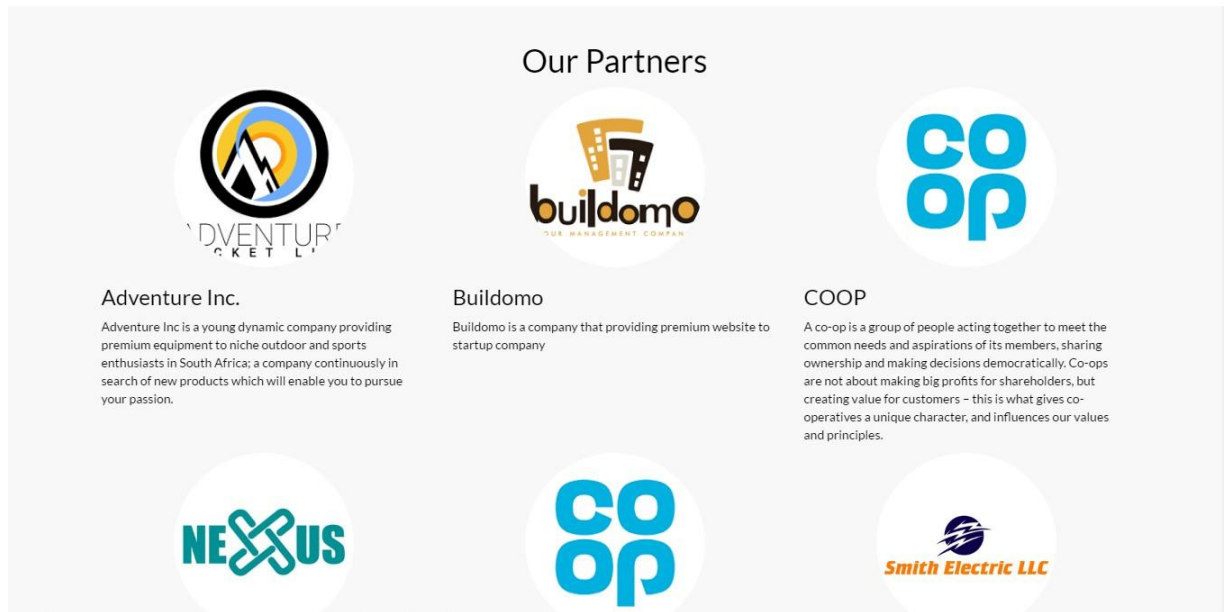
Airell Thomas

Web Designer



Lleria Samoht

UI/UX Designer



Exercise

20. Pahami potongan kode di atas untuk quiz!

Homework

21. Cari tau tata penggunaan pingendo for HTML5 and CSS3 Prototyping!