SESI 2 : CUSTOM ELEMENTS

CUSTOM ELEMENTS

Mengijinkan Frontend Developer untuk menetapkan type baru untuk tag html

Contoh tag html standar :

<h1></h1>

contoh custom element :

<f></frontend>

component tidak akan pernah ada tanpa ada fitur yang kita buat sendiri sebagai syarat menggunakan custom element, yaitu :
menetapkan element HTML / DOM baru
membuat element HTML / DOM baru berdasarkan tag yang sudah ada
membuat satu paket logic fungsi / perintah ke dalam satu tag

Mendaftarkan elements baru

Custom elements dibuat dengan document.registerElement();

var FeHead = document.registerElement('fe-head');
document.body.appenchild(new FeHead());

argument pertama dari document.registerElement() adalah tag name yang kita buat
namanya harus mengandung dash (-)
contoh :
<fe-head></fe-head>

tidak disarangkan
<fehead></fehead> atau <fe_head></fe_head>

argument pertama adalah standar penulisan yang memungkinkan parser untuk membedakan element khusus dari element biasa, juga memastikan kompatibilitas ke depan saat tag baru ditambahkan ke HTML.

argument kedua adalah object (opsi) untuk menggambarkan prototype element. Ini adalah tempat untuk menambahkan fungsionalitas khusus (seperti public properties atau methods) ke tag html buatan.

Secara default, custom element mewarisi dari dasar HTMLElement. Jadi contoh sebelumnya setara dengan :

```
var FeHead = document.registerElement('fe-head', {
        prototype: Object.create(HTMLElement.prototype)
});
```

Argumen ke document.registerElement ('FeHead') mengajarkan browser tentang elemen baru, dan mengembalikan konstruktor yang dapat Anda gunakan untuk membuat contoh <fe-head>. Sebagai alternatif, Anda dapat menggunakan teknik instantiating elemen lain jika Anda tidak ingin menggunakan konstruktor.

Extending elements
Custom elements memungkinkan untuk menambahkan fungsi element HTML pada tag asli yang ada dan element khusus lainnya. Untuk menambahkan fungsi, perlu menambahkan nama pada argument registerElement() dan protote element yang akan diwarisi

Contoh

Extending native elements

Kalau kita tidak puas dengan fungsi tag html

<button></button>

kita boleh menambahkan sendiri fungsi pada tag html. Untuk menambahkan fungsi pada tag <button></button>, buat element baru yang mewarisi prototype HTMLButtonElement dan menambahkan fungsinya.

```
var FeButton =  document.registerElement('fe-button', {
        prototype: Object.create(HTMLButtonElement.prototype),
        extends: 'button'
});
```

panggil

```
<button is="fe-button"></button>
```

custom elements yang mewarisi dari elemen standar disebut extension custom elements, bisa kita baca dengan elemen button adalah element fe-button

Lifecycle callback methods
Elements dapat menentukan methods khusus untuk memanfaatkan masa keberadaan elements itu sendiri. Ada 4 jenis argument pada methods ini : masing masing punya nama dan tujuan tertentu

createdCallback
contoh dari element yang dibuat

attachedCallback
contoh dari element yang ditambahkan ke dalam document

detachedCallback
contoh dari element yang dihilangkan dari dalam document

attributeChangedCallback(attrName, oldVal, newVal)
attribute ketika menambahkan, menghapus, atau mengupdate

contoh :

html :

<var-el></var-el>

javascript :

```
var VarElement = Object.create(HTMLElement.prototype);
VarElement.createdCallback = function(){
        This.innerHTML = “<p>hallo, kenalin saya element baru</p>”;
};

var VarEl = document.registerElement(‘var-el’, {
        prototype: VarElement
});
```

sass