

JAVASCRIPT & JQUERY

JQUERY

Write Less, Do More

JQUERY

"jQuery är ute"

"Noobigt med jQuery"

kan stämma, i många fall får jQuery oss att skriva dålig kod för man förlitar sig för mycket på biblioteket.

JQUERY

Det är **ok** att använda jQuery

Bara du vet varför

Det mesta kan enkelt lösas med **vanillaJS**

Tänker du: "Oj här måste jag ha DOM-manipulation, bäst att slänga in
jQuery" är du fel ute.

JQUERY

Har du specifika uppgifter som mycket lättare löses med jQuery än vanillaJS, använd jQuery.

Det är inte ett dåligt bibliotek, men det får folk att skriva dålig kod av någon anledning.

JQUERY ÄR EN SAMLING FUNKTIONER SOM ÄR
WRAPPERS RUNT VANLIGA JAVASCRIPT-FUNKTIONER.

```
function fadeIn(el) {  
    el.style.opacity = 0;  
  
    var last = +new Date();  
    var tick = function() {  
        el.style.opacity = +el.style.opacity + (new Date() - last)  
        last = +new Date();  
  
        if (+el.style.opacity < 1) {  
            (window.requestAnimationFrame && requestAnimationFrame(  
                tick  
            ));  
        }  
    };  
  
    tick();  
}
```

```
$(el).fadeIn();
```

Detta är ju dock bara ett **Interface** för tidigare kod

ANVÄNDBARA FUNKTIONER

SELECTOR

SELECTOR SHORTHAND

```
$ ( '#idSelector' );
```

```
document.getElementById( '#idSelector' );
```

Mer lik **querySelector** då man kan välja nästlade element

```
$ ( '#idSelector .classSelector span' )
```

HITTA ELEMENT

Hitta ett visst element av en viss typ

```
$ ( ' .box' ) .find ( 'li' )
```

Hitta förälder

```
$ ( ' .box' ) .parent ( ) //Closest parent  
$ ( ' .box' ) .parents ( ) //All parents  
$ ( ' .box' ) .children ( ) //All children
```

DOM TRAVERSAL

All DOM-navigering som du kan tänka på finns som en shorthand i

jQuery

jQuery API

YOU MIGHT NOT NEED JQUERY

YOU DON'T NEED JAVASCRIPT

AJAX

Async JavaScript and XML

`$.AJAX()`

Jag har använt jQuery för att dess wrapper till `XMLHttpRequest` är
mycket enklare att använda

Förser oss med ett antal `shorthand`-funktioner

`$.ajax()`

```
var ajaxRequest = new XMLHttpRequest();
    ajaxRequest.onreadystatechange = function(){
        if (ajaxRequest.readyState == 4 && ajaxRequest.status
            console.log(ajaxRequest.responseText);
        }
    };
ajaxRequest.open( 'GET', 'http://localhost:3000', true);
ajaxRequest.send( );
```

Grundutförandet av ajax

```
$.ajax({  
  method: 'GET',  
  url: 'http://localhost:3000',  
  success: (response) => {  
    return response;  
  }  
})
```

```
$.ajax({  
  method: 'GET',  
  url: 'http://localhost:3000',  
  success: (response) => { return response; },  
  error: (error) => { return error };  
})
```

Ref: jQuery API: \$.ajax()

```
$.ajax({  
  method: 'GET',  
  url: 'http://localhost:3000',  
  //When you send data  
  contentType: 'application/json; charset=utf-8',  
  dataType: 'json' //What you expect to get  
  success: (response) => { return response; },  
  error: (error) => { return error };  
})
```

`$.AJAX()` SHORTHAND

- `$.get()` : `method: 'GET'`
- `$.post()` : `method: 'POST'`
- `$.load()` : använd ofta för att ladda in fragment av sidor dynamiskt, html chunks
- `$.getJSON()` : `mehod: 'GET', dataType: 'json'`

CALLBACKS

Samtliga funktioner borde ha en callback

```
$.get('http://localhost:3000', (response) => response);
```

Samtliga funktioner borde också ha någon sort error-handling

```
error: (error) => return error;
```

ALTERNATIV SYNTAX

```
$.get('http://localhost:3000')  
  .done((response) => response)  
  .fail((error) => error)  
  .always(() => 'After all this time?')
```

Ref: jQuery API: \$.ajax()

MEN VAD SKA VI HÄMTA FÖR DATA?

<http://fend16.azurewebsites.net/albums> ??? lol ne

JSON-SERVER

LOKAL UTVECKLINGSSERVER

json-server skapar en databas som sparar informationen i en **.json**-fil

Åtkomst via **localhost**

Använder sig av **GET**, **POST**, **PATCH** oc **DELETE** som tidigare, ett ReST API.

[HTTPS://GITHUB.COM/TYPICODE/JSON-SERVER](https://github.com/typicode/json-server)