

jQuery-free JavaScript

Native Utilities

Elijah Manor
<http://elijahmanor.com>
@elijahmanor



pluralsight 
hardcore developer training

Native Utilities

- Data
- Each, Grep, Map
- Extend
- In Array
- Type Checking
- Trim



Data

jQuery: Option 1

```
<div id="contact" data-contact-name="John Smith"></div>  
<p id="ad" data-options='{ "color": "red", "margin": 3 }'></p>
```

```
$("#contact").attr("data-contact-name"); // string
```

```
$("#contact").data("contactName"); // string
```

```
$("#ad").attr("data-options"); // string
```

```
$("#ad").data("options"); // object
```

jQuery: Option 2

```
$("#contact").data("person",  
    { fName: "John", lName: "Smith" });
```

```
$("#contact").data("person"); // object
```

Data

Native: Option 1

```
var contact = document.getElementById("contact"),
    user = { fname: "John", lname: "Smith" };

contact.getAttribute("data-contact-name"); // string
contact.setAttribute("data-contact-name", "Jane Smith");

contact.setAttribute(
    "data-contact-name",
    JSON.stringify(user)
);

user = JSON.parse(
    contact.getAttribute("data-contact-name")
);
```

Data

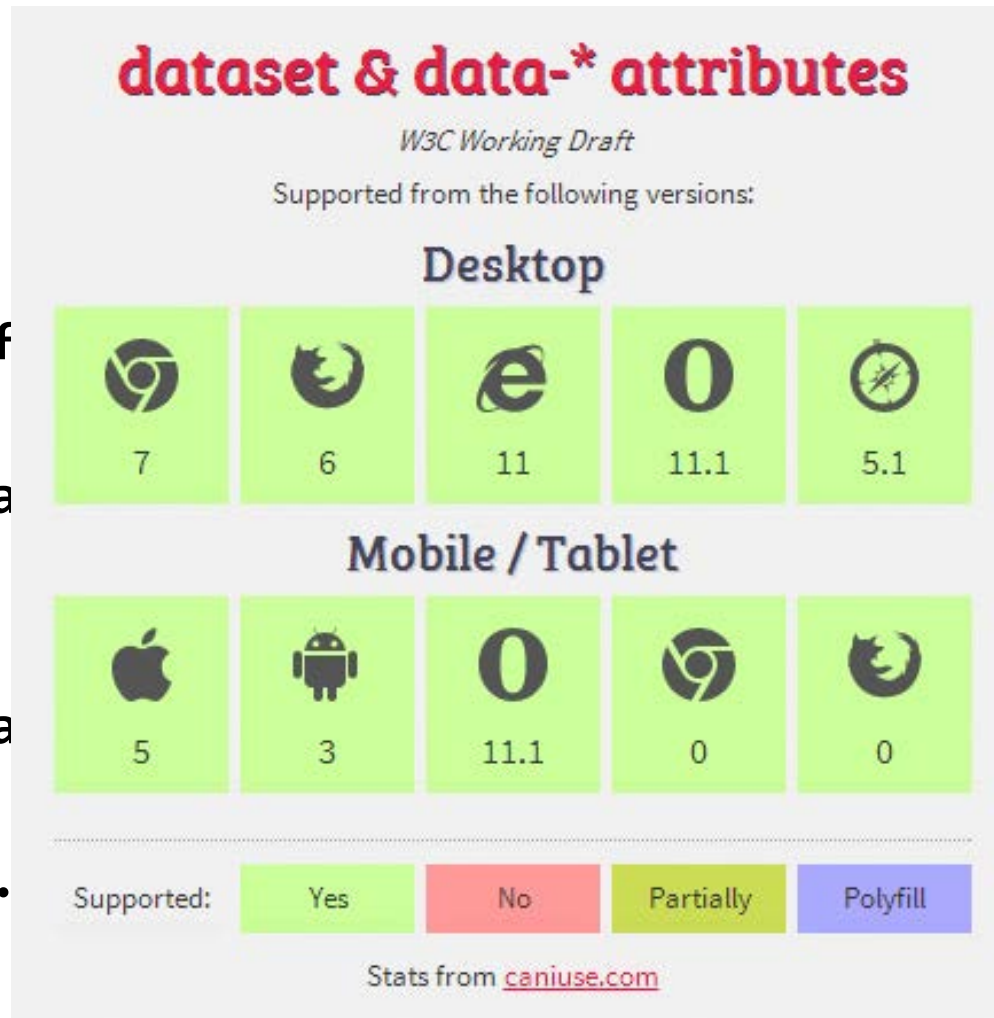
Native: Option

```
var contact  
  user = { f
```

```
contact.data  
  '{ fname:
```

```
contact.data
```

```
user = JSON.
```



"),

Polyfill: <http://manor.im/polyfill-dataset>

Each, Grep, & Map

```
var languages = [  
  { id: "TypeScript", date: 2012, by: "Anders Hejlsberg" },  
  { id: "CoffeeScript", date: 2009, by: "Jeremy Ashkenas" },  
  { id: "F#", date: 2005, by: "Don Syme" },  
  { id: "C#", date: 2000, by: "Anders Hejlsberg" },  
  { id: "JavaScript", date: 1995, by: "Brenden Eich" },  
  { id: "Java", date: 1995, by: "James Gosling" },  
  { id: "C++", date: 1983, by: "Bjarne Stroustrup" }  
];
```

Each, Grep, & Map

```
jQuery [
  { "name": "TypeScript", "age": 1 },
$.each { "name": "CoffeeScript", "age": 4 },
  con { "name": "JavaScript", "age": 18 }
}); ]
```

```
languages = $.grep(languages, function(language) {
  return ~language.id.indexOf("Script");
});
```

```
languages = $.map(languages, function(language) {
  return {
    name: language.id,
    age: new Date().getFullYear() - language.date
  };
});
```

Each, Grep, & Map

Native

```
languages.forEach(function(language) {  
    console.log(language.id + ": " + language.date);  
});
```

```
languages = languages.filter(function(language) {  
    return ~language.id.indexOf("Script");  
});
```

```
languages = languages.map(function(language) {  
    return {  
        name: language.id,  
        age: new Date().getFullYear() - language.date  
    };  
});
```


Each, Grep, & Map

Underscore or Lo-Dash

```
_.each(languages, function(language) {  
    console.log(language.id + ": " + language.date);  
});
```

```
languages = _.filter(languages, function(language) {  
    return ~language.id.indexOf("Script");  
});
```

```
languages = _.map(languages, function(language) {  
    return {  
        name: language.id,  
        age: new Date().getFullYear() - language.date  
    };  
});
```

Extend

jQuery

```
var combined = $.extend(  
    { color: "PapayaWhip", font: "Tahoma" }, // target  
    { color: 'Chartreuse' }                 // object1...  
);
```

```
{ "color": "Chartreuse", "font": "Tahoma" }
```

Underscore or Lo-Dash

```
var combined = _.extend(  
    { color: "PapayaWhip", font: "Tahoma" }, // destination  
    { color: 'Chartreuse' }                 // sources...  
);
```

In Array

jQuery

```
var fruit = ["orange", "apple", "pear", "pineapple"];  
  
console.log("Found Pear: " + !!~$.inArray("pear", fruit));
```

Native

```
var fruit = ["orange", "apple", "pear", "pineapple"];  
  
console.log("Found Pear: " + !!~fruit.indexOf("pear"));
```

Type Checking

jQuery

```
var fruit = ["orange", "apple", "pear", "pineapple"],
    quantity = 42,
    buy = function(index) {
        console.log("Bought " + quantity + " " + fruit[index]);
    };

$.type(fruit);           // "array"
$.isArray(fruit);        // true

$.type(buy);             // "function"
$.isFunction(buy);       // true

$.type(quantity);        // "number"
$.isNumeric(quantity);   // true
```

Type Checking

Native & Underscore/Lo-Dash

```
var fruit = ["orange", "apple", "pear", "pineapple"],  
    quantity = 42,  
    buy = function(index) { /* ... */ };
```

```
typeof fruit;           // "object" :(  
Array.isArray(fruit);    // true (ECMAScript 5)  
_.isArray(fruit);        // true  
  
typeof buy;             // "function"  
_.isFunction(buy);       // true  
  
typeof quantity         // "number"  
_.isNumber(quantity);    // true
```

Trim

jQuery

```
var message = "    look at me, I'm padded!    ";  
$.trim(message);
```

Native

```
var message = "    look at me, I'm padded!    ";  
message.trim();
```

DEMO

- Data
- Each/Grep/Map
- Extend
- In Array
- Type Checking
- Trim

Summary

```
var $menu = $("#menu");  
$menu.data("categories");
```

```
$.each(), $.grep(), $.map()
```

```
$.extend()
```

```
$.isArray()
```

```
$.type()
```

```
$.trim()
```

```
var menu =  
document.getElementById("menu");  
menu.dataset.categories;
```

```
[].forEach(), [].filter(), [].map()
```

```
_.extend()
```

```
[].indexOf()
```

```
typeof OR Array.isArray()
```

```
"".trim()
```