



THE UNIVERSITY
of ADELAIDE

Web and Database Computing •

adelaide.edu.au



Introduction to JavaScript: JavaScript Objects

Javascript Objects

What are objects in Javascript?

Objects are collections of values

Primitive types in Javascript

- string
- number
- boolean
- null
- undefined

Everything else is an object

In JavaScript, almost "everything" is an object.

- Dates are always objects
- Maths are always objects
- Regular expressions are always objects
- Arrays are always objects
- Functions are always objects
- Objects are always objects

All JavaScript values, except primitives, are objects.

Booleans, Numbers and Strings can even be objects if defined using a constructor.



Defining objects

Objects are collections of values

written as a series of property:value pairs

```
var object = { p1:'value 1', p2:2, p3:false };
```

Accessing objects

Get/set values using dot notation:

```
var object = { p1:'value 1', p2:2, p3:false };  
var x = object.p1;  
object.p4 = true;
```

Can also use array notation:

```
var object = { p1:'value 1', p2:2, p3:false };  
var x = object['p1'];
```

The values can be anything that you would normally store in a variable:

```
object = { withinanobject:{ anobject:'bwaah' } };
```



But objects should also have behaviour?

- Functions are always objects

JavaScript Result



[Edit in JSFiddle](#)

```
var object = {  
  age:42,  
  alertAge: function() {  
    alert(this.age);  
  }  
};  
  
object.alertAge();
```

Properties can be dynamically added to objects

JavaScript Result



[Edit in JSFiddle](#)

```
var object = {  
    age: 42  
};  
  
object.colour = 'green';  
document.body.innerText = object.age + ' ' + object.colour;
```

Object Constructors

Making several of the same type of object

Constructor definition & use

Constructors are a special type of function:

```
function Car(make, model, year) {  
  this.make = make;  
  this.model = model;  
  this.year = year;  
}
```

They can be called using the **new** keyword:

```
var car1 = new Car("Morris", "Mini Deluxe", 1967);  
var car2 = new Car("Nissan", "Pulsar", 2013);
```

Modifying constructors using prototypes

Given constructor that we want to add to:

```
function Car(make, model, year) {  
  this.make = make;  
  this.model = model;  
  this.year = year;  
}
```

Use a prototype to modify it:

```
Car.prototype.transmission = "manual";
```

Arrays

Arrays are a type of object used to store multiple values in the same variable.

```
var animals = ["Lion", "Tiger", "Bear"];
```

The values can be any object and do not have to be the same type.

```
var animals = ["Lion", "Tiger", 1924];
```

Individual elements accessed using indexes

```
var lion = animals[0];
```

Array functions

Arrays have methods to make standard array operations easier, e.g

- `push(var)`
- `pop()`
- `join()`
- `concat(array)`
- `slice(index1,index2)`
- `splice(index1,number,items)`

The w3schools array reference contains details on all of these functions.

See https://www.w3schools.com/jsref/jsref_obj_array.asp

Array iteration

Prior to ES6, the preferred way of looping over elements in an Array is with an Iterator function:

```
var animals = ["Lion", "Tiger", "Bear"];
animals.forEach(myFunction);

function myFunction(value, index, array) {
  console.log(value);
}

animals.forEach(function(value, index, array) {
  console.log(value);
})
```

There are also other iterator functions that perform specific tasks.

See https://www.w3schools.com/js/js_array_iteration.asp

Now we can also use the **for ... of** Loop:

```
for(const animal of animals) {
  console.log(animal);
}
```

Be mindful of variable scope when using loops/functions!



THE UNIVERSITY
of ADELAIDE

CRICOS PROVIDER NUMBER 00123M