Javascript Idioms

http://paulhammond.org/2006/jsidioms

Paul Hammond
paul@paulhammond.org

1. Phrase etc. established by usage and not immediately comprehensible from the words used (e.g. *over the moon, see the light*)
2. Form of expression peculiar to a language etc.
3. Language of a people or country
4. Characteristic mode of expression in art etc.

Idiom (ɪdɪəm) n.

1. Phrase etc. established by usage and not immediately comprehensible from the words used (e.g. *over the moon*, *see the light*)
2. Form of expression peculiar to a language etc.
3. Language of a people or country
4. Characteristic mode of expression in art etc.

Idiom (ɪdɪəm) n.

```
s/Ruby/Perl/;
```

```
3.times { print "Java Sucks" }
```

1. Phrase etc. established by usage and not immediately comprehensible from the words used (e.g. *over the moon*, *see the light*)
2. Form of expression peculiar to a language etc.
3. Language of a people or country
4. Characteristic mode of expression in art etc.

```
{
    local $/;
    $file = <FILE>;
}
```

```
my ($result = $string) =~ s/Java/Perl/;
```

```
value = (boolean and [a] or [b])[0]
```

1. Phrase etc. established by usage and not immediately comprehensible from the words used (e.g. *over the moon*, *see the light*)
2. Form of expression peculiar to a language etc.
3. Language of a people or country
4. Characteristic mode of expression in art etc.

```
isset?($_GET['id'])?$_GET['id']:''
```

Talk about some idiomatic Javascript

Use that to understand the language and its people

1

```
if (document.getElementById) {
    …
}
```

```
if (document.getElementById) {
    …
}
```

Some browsers are missing functionality

```
if (document.getElementById) {
    …
}
```

No else clause

```
if (document.getElementById) {
    …
}
```

You cannot rely on javascript anyway

1

Progressive Enhancement:

Start with a basic page that works
add Javascript as an extra

2

```
function get_xmlHttpRequest(){

  if (window.XMLHttpRequest) {
    return new XMLHttpRequest()
  }

  else if (window.ActiveXObject) {
    return new ActiveXObject("Microsoft.XMLHTTP")
  }
}
```

```
function get_xmlHttpRequest(){
  // for most browsers
  if (window.XMLHttpRequest) {
    return new XMLHttpRequest()
  }
  // for internet explorer
  else if (window.ActiveXObject) {
    return new ActiveXObject("Microsoft.XMLHTTP")
  }
}
```

Browsers are inconsistent

```
function get_xmlHttpRequest(){
  // for most browsers
  if (window.XMLHttpRequest) {
    return new XMLHttpRequest()
  }
  // for internet explorer
  else if (window.ActiveXObject) {
    return new ActiveXObject("Microsoft.XMLHTTP")
  }
}
```

Browsers are inconsistent
so we need abstractions

2

Don't reinvent the wheel.
Use a library.

3

```
function g(id) {
  return document.getElementById(id);
}

nav = g('nav')
form = g('loginform')
```

```
function g(id) {
  return document.getElementById(id);
}

nav = g('nav')
form = g('loginform')
```

Typing document.getElementById is boring

```
function g(id) {
  return document.getElementById(id);
}

nav = g('nav')
form = g('loginform')
```

Download time of Javascript code matters

```
function g(id) {
  return document.getElementById(id);
}

nav = g('nav')
form = g('loginform')
```

Download time of Javascript code matters
(but then, so does readability)

Use a compressor

http://www.crockford.com/javascript/jsmin.html
http://dean.edwards.name/packer/

4

```
for(i=array.length;i-- >0;){
    …
}
```

```
for(i = array.length; i>0; i--) {
    …
}
```

```
for(i = array.length; i>0; i--) {
    …
}
```

Javascript is interpreted
so it can be quite slow

```
for(i = array.length; i>0; i--) {
    …
}
```

Looping backwards is faster!

```
for(i=array.length;i-- >0;){
    …
}
```

Looping backwards is faster!
and takes less bytes!

```
for(i=array.length;i-- >0;){
    …
}
```

Some people have far too much free time

Premature optimization is the root of all evil

If you're doing this kind of optimisation you're probably wasting your time

5

```
var somepackage = {

  doSomething: function() {…},
  doSomethingElse: function() {…}

}

somepackage.doSomething()
```

```
var somepackage = {

  doSomething: function() {…},
  doSomethingElse: function() {…}

}

somepackage.doSomething()
```

Javascript doesn't have built in namespaces

```
var somepackage = {

   doSomething: function() {…},
   doSomethingElse: function() {…}

}

somepackage.doSomething()
```

Javascript doesn't have built in namespaces
but we can fake them with objects

5

Namespacing your code is polite
(and necessary in almost all cases)

6

```
function makeCounter() {
  var i = 0
  return function() {
    i++
    return i
  }
}
counter = makeCounter();
counter() // returns 1
counter() // returns 2
```

```
function makeCounter() {
  var i = 0
  return function() {
    i++
    return i
  }
}
counter = makeCounter();
counter() // returns 1
counter() // returns 2
```

Closure

```
function makeCounter() {   //outer
  var i = 0
  return function() {      //inner
    i++
    return i
  }
}
counter = makeCounter();
counter() // returns 1
counter() // returns 2
```

Variables from outer function remain available to inner function

```
function makeCounter() {
  var i = 0
  return function() {
    i++
    return i
  }
}
counter = makeCounter();
counter() // returns 1
counter() // returns 2
typeof(i) // undefined
```

Private data

```
function makeAlert(text) {
  return function() {
    alert(text)
  }
}

req = get_xmlhttprequest();
req.onRequestChange = makeAlert("foo")
```

Callback functions without globals

6

Take the time to understand closures

7

```
counter = function() {
  var i=0;
  return function() {
    i++;
    return i;
  }
}()
counter() // returns 1
counter() // returns 2
typeof(i) // undefined
```

```
counter = function() {
  var i=0;
  return function() {
    i++;
    return i;
  }
}()
counter() // returns 1
counter() // returns 2
typeof(i) // undefined
```

Closure with anonymous function

```
counter = function() {
  var i=0;
  return function() {
    i++;
    return i;
  }
}()
counter() // returns 1
counter() // returns 2
typeof(i) // undefined
```

Closure with anonymous function

```
counter = function() {
  var i=0;
  return function() {
    i++;
    return i;
  }
}()
counter() // returns 1
counter() // returns 2
typeof(i) // undefined
```

Creating a function then calling it

```
counter = function() {
  var i=0;
  return function() {
    i++;
    return i;
  }
}()
counter() // returns 1
counter() // returns 2
typeof(i) // undefined
```

```
counter = function() {
  var i=0;
  return function() {
    i++;
    return i;
  }
}()
counter() // returns 1
counter() // returns 2
typeof(i) // undefined
```

```
counter = function() {
  var i=0;
  return function() {
    i++;
    return i;
  }
}()
counter() // returns 1
counter() // returns 2
typeof(i) // undefined
```

Unmaintainable

7

Javascript gives you enough rope...

8

```
function foo() {
  bar = [
    'foo',
    'bar',
  ];
  return
}
```

```
function foo() {
  bar = [          // implied global
    'foo',
    'bar',         // trailing comma
  ];
  return           // no semicolon
}
```

Badly formatted code

```
function foo() {
  bar = [          // implied global
    'foo',
    'bar',         // trailing comma
  ];
  return           // no semicolon
}
```

Badly formatted code
(like most of the code in this presentation)

Bad formatting causes
browser specific bugs

http://www.jslint.com/

Javascript is a quirky language.

understand the constraints of running code in browsers
understand functional programming

Thankyou

http://paulhammond.org/2006/jsidioms

paul@paulhammond.org