VANILLA JAVASCRIPT

OBJECTS

Everything in Javascript is an **OBJECT**.

OBJECTS

OBJECTS can be simple.

OBJECTS

```
var today = “Tuesday”;
```

(a String)

OBJECTS

```
var today = 9;
```

```
(a Number – either Int or Float)
```

OBJECTS

```
var whatIsToday = function(){
  console.log(‘Tuesday’);
};
```

(a function)

OBJECTS

```
var today = ['September', 09, 2014]
```

(an array)

OBJECTS

OBJECTS can also be complex.
Complex Objects are made
up of Properties and Methods.

OBJECTS

KEY-VALUE PAIRS

(the properties)

OBJECTS

A **KEY** is a unique identifier for some sort of data.

A **VALUE** is the data you are storing with the key.

OBJECTS

```
var review = {

  type : “Vanilla Javascript”

}
```

KEY

VALUE

ACTIONS

(the methods)

OBJECTS

An **ACTION** is a function.

OBJECTS

```
var review = {
  shush : function(){
    console.log(‘Quiet! I’m studying!’);
  }
};
```

OBJECTS

```
review.type
```

will return “Vanilla Javascript”

```
review.shush();
```

will run the function shush in the review object.

VARIABLES

A variable is an **OBJECT** that you define.

VARIABLES

They can have **GLOBAL** or **LOCAL** scope.

VARIABLES

GLOBAL variables can be accessed anywhere in a script.

VARIABLES

LOCAL variables can only be accessed within the function where they are defined.

VARIABLES

```
var global = “I love everyone!”

function aloneTime(){

  var local = “I need to be alone.”

};
```

VARIABLES

You can pass a local variable between functions by adding it to the **function parameters**.

VARIABLES

```
var global = “I love everyone!”

function aloneTime(){
  var local = “I need to be alone.”
  tellEveryone(local);
};

function tellEveryone(msg){
  console.log(msg);
};
```

Let’s talk about **ORDER.**

ORDER

ORDER

If you reference an object before you define it, this will happen:

ORDER

ORDER

Always structure code from
the bottom up (as best you can)
to avoid mentioning something
that hasn’t been defined yet.

Any questions?