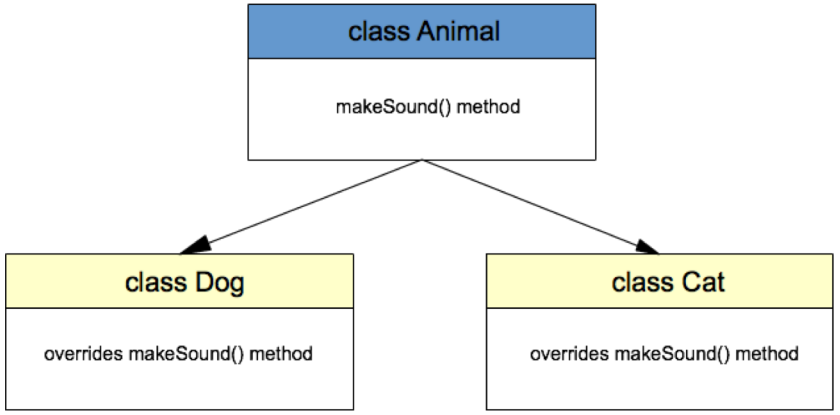
Coding Vocabulary

Kay Liang

Table of Contents

Object Oriented Programming

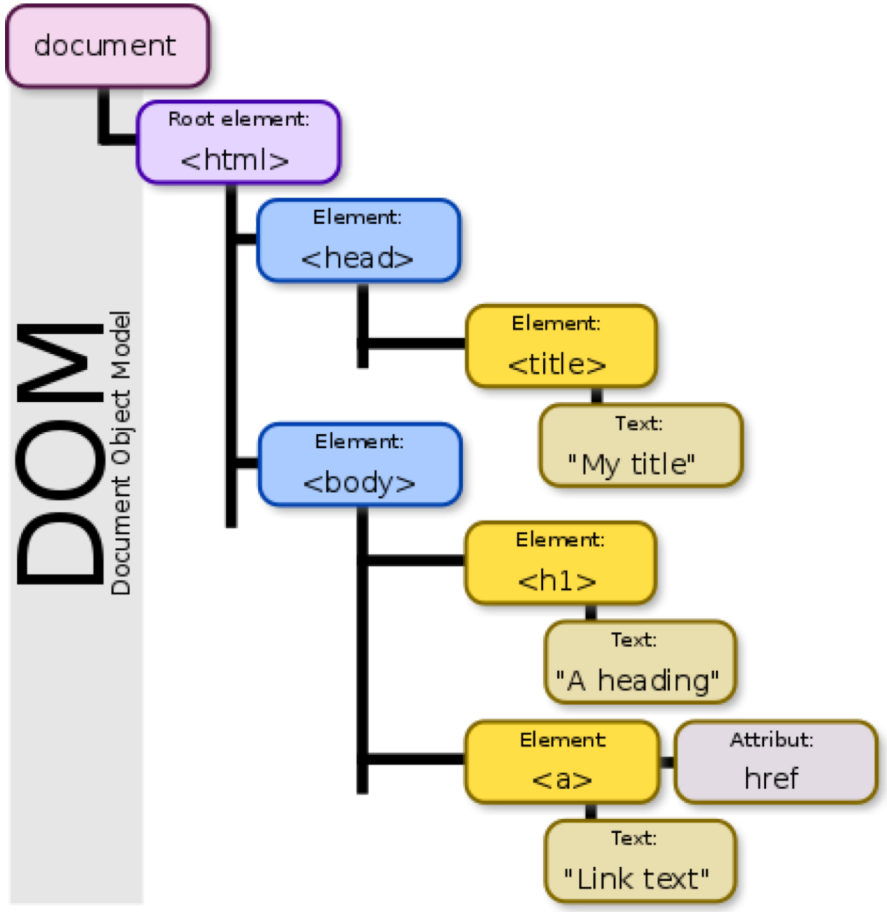
- **Objects:** a data type that stores multiple attributes and properties
- **Class:** a class is a template definition of the method s and variable s in a particular kind of object . Thus, an object is a specific instance of a class.
- **Instantiation:** creating and object from an instance of a class
- **Polymorphism**: the ability to redefine methods for derived classes.
- **Inheritance:** basing an object or class upon another object or class
  - Parent and Child



Document Object Model (DOM)

- Web page that programming languages can connect to

- Javascript vs. DOM
  - Javascript is the language
  - DOM is where the stuff happens



Element + Tag

- **Element:** Individual component of an HTML document
- Position is indicated as spanning from a start tag to an end tag
  - `body, head, div, img, a`

- Element vs. Tag
  - `<p>  </p>` = tag
  - `<p>This is the content</p>` = element

```
<element attribute = “value”> element content </element>
```

Attribute

- Special words used inside the opening tag to control the element's behavior
- Modifies or provides functionality to certain element types

```
<a href = "mama.com"></a>
<img src = "mama.png"></img>
<head class = "mama"></head>
```

```
<element attribute = "value"> element content </element>
```

Id

- Attribute that provides a **document-wide unique identifier**
- Allows for direct targeting of specific element within the document
- May only use once
- Any element on the doc can have an id

Class

- Attribute that provides way of **classifying similar elements**
- May use multiple times
- Many elements on the doc can have the same class

Selector

- Patterns used to select the element(s) you want to style

```
selector {
    property: value;
}
```

declaration

.class	.intro	selects elements with class="intro"
#id	#firstname	select element with id="firstname"
*	*	select all elements
element	p	select all <p> elements
element, element	div, p	selects all elements and all <p> elements
element element	div p	selects all <p> elements inside
[attribute]	[target]	select all elements with a target attribute

Property + Value

- **Property:** which aspect of the selector will be changed visually
- **Value:** what the chosen property will be changed to

```
selector {
   property: value;
}

p {
   text-align: center;
   color: #000;
}
```

Data types

- Boolean `true, false`
- Null `null`
- Undefined `undefined`
- Number `3, 5, 6`
- String `“hello world”, “3”`
- Object `{name: value}`

Variable

- A storage location paired with a symbolic name and a known/unknown value

```
var name = value;
```

name:value pairs are used in JS objects

- Declaring a variable
  - `var x = 42;`
- Assigning a value to a variable
  - `x = 42;`

Scope

- Determines the accessibility (visibility) of variables

Local Scope

```
// code here can NOT use carName

function myFunction() {
  var carName = "Volvo";

  // code here CAN use carName

}
```

Global Scope

```
var carName = "Volvo";

// code here can use carName

function myFunction() {

  // code here can also use carName

}
```

Function + Method

- **Function:** is a piece of code that is called by name. It can be passed data to operate on (i.e. the parameters) and can optionally return data (the return value). All data that is passed to a function is explicitly passed. Independent of an object.

```
returnApple();

function returnApple(){
        return “apple”;
}
```

Calling a function

Defining a function

- **Method:** is a piece of code that is called by a name that is associated with an object. Called on an object.

```
document.write(“apple”);
```

- Every method is a function, not every function is a method
  - Constructor is a function but not a method

Parameter + Argument

- **Parameter:** variable in the declaration of a function
- **Argument:** actual value of this variable that gets passed to this function

```
myFunction(argument1, argument2);

Function myFunction (parameter1, parameter2){
   return argument;
}
```

Expression + Operator

Operator: code element that performs and operation
Expression: contains an operator and operand operand operator operand 3 + 2

Assignment Operators

Name	Shorthand operator	Meaning
Assignment	`x = y`	`x = y`
Addition assignment	`x += y`	`x = x + y`
Subtraction assignment	`x -= y`	`x = x - y`
Multiplication assignment	`x *= y`	`x = x * y`
Division assignment	`x /= y`	`x = x / y`
Remainder assignment	`x %= y`	`x = x % y`

Comparison Operators

- == equal
- != not equal
- > greater than
- >= greater than or equal

Array

- Store multiple values in a single variable

```
var cars = ["Saab", "Volvo", "BMW"];
```

- Good for looping through many items to find a specific one
- **Index number:** the position of an element in an array, starts at 0

```
var cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars[0];
```

Saab

Loop + Iteration

- Loops allow you to do something repeatedly

- `for` statement

```
for ([initialExpression]; [condition]; [incrementExpression]) {
    doSomething();
}
```

```
for (var i = 0; i < 5; i++) {
    console.log(i);
}
```

```
0
1
2
3
4
>
```

Comment

- Used to explain javascript code
- Starts with //

```
var x = 5;       // Declare x, give it the value of 5
var y = x + 2;   // Declare y, give it the value of x + 2
```

Object

- A variable that contains many values
- Assigning many values to an object;

```
var car = {type:"Fiat", model:"500", color:"white"};
```

- Constructing a new object
  - `var myObject = new Object();`
- Selecting/Returning/Toggling/Adding a property of an object
  - `myObject.length - 2;`
  - `myObject.nationality = “English”;`
- Calling a method on a object
  - `myObject.run();`
  - `myObject.play();`

Seems familiar right?

p { color: #fff }

That’s because a paragraph is an object on the DOM! That’s why it’s Document **Object** Model.

Constructor

- A special function that for creating and initializing and object
- Like a cookie cutter to make more than one object with similar features (same properties and methods)

First letter has to be caps

```
function Book() {
        this.pages = 3;
        this.title = “Garfield’s Lasagna Recipes”
}
var myBook = new Book();
```

Defining the Book constructor

“new” = keyword to call the Book constructor

This

- A keyword that refers to the object it belongs to

(in script)

```
var person = {
  firstName: "John",
  lastName : "Doe",
  id       : 5566,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};
```

(in html)

```
document.getElementById("demo").innerHTML =
person.fullName();
```

→ John Doe

Exercise

1. instantiate an **object**
2. call one of its **functions**
3. pass/feed the function an **argument** that is a **global variable**
4. receive that argument as a **local variable** within the object's **constructor**
5. use the local variable in a **loop** that creates an **array**
6. write the array on the **document**