**Week 02**

The object is one set the properties that have name and value, they are mutable and manipulated by reference and *not for value*, also it inherits properties from another object known as prototype. The methods an object normally are properties inherits (prototype heritage).

The object has three attributes the associated object:

* The prototype of an object is one reference to another object of which the properties are inherited.
* Class of an object is a string that classifies the type of an object.
* The flag extensible of an object specifies if new properties can be added to object.

The keyword *new*create and initialize a new object and must be followed by one function call, the function used this way is called the *constructor* serves to initialize one newly created object.

All objects in JavaScript have a second object in JavaScript (or null) associated and this second object is the prototype, and the first object inherits properties of the prototype.

A class shares properties to contain or define your state and/or behavior. It is based on the mechanism of heritage, the objects that inherited properties of the same object prototype, so they are *instances* of the same class.

***Getter***

The *get* syntax associates an object property with a function that will be called when that property is accessed.

***Setter***

The *set* syntax binds a function property to be called when there is an attempt to set a value for that property.

***This***

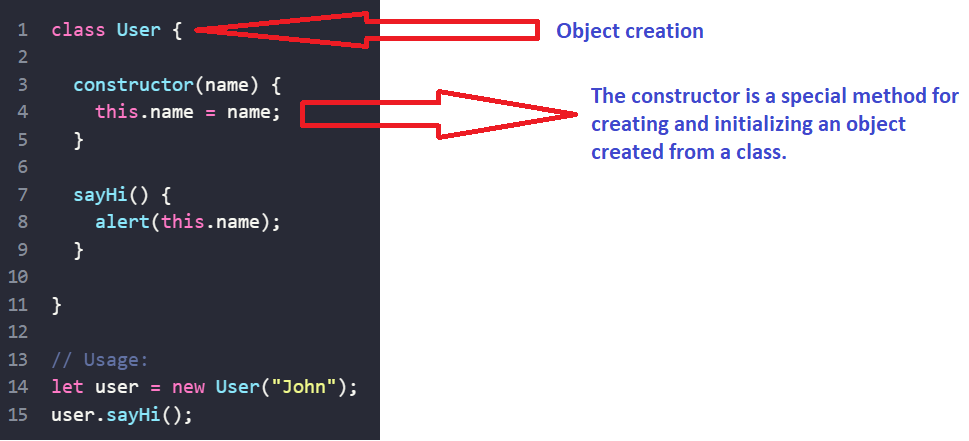
*This*points to an object, and the object will be point will depend on the context that it is inserted.

When not in *'use strict' this* will point to the global object, the *'use strict'* is used to avoid errors and inconsistencies in code. In 'use strict' the value the *this* remain in defined when entering the execution context, by pattern is undefined.

*This* as method of the object will point to the object and can access the properties the object, except for arrow functions that your value is defined by execution context where is inserted, in one global code, this takes on the global object. *This* inside the browser is the same thing as window.

**Class Basic Syntax**

Example syntax:



No comma between class methods.

In JS class is a kind of function.

**Export**

It is possible to export using export before being it a variable, function or a class or using *export {....}* inside {} put what you want. No semicolons after export class/function.

**Import**

Usually, we put a list of what to import in curly braces *import {...},* But if there’s a lot to import use import \*

Example: import \* as say from './say.js';