## **Note: This document is to give an overview of the Odoo Javascript and Libraries used to work with Camera.**

Odoo has separate documentation for Javascript Framework  
<https://www.odoo.com/documentation/15.0/developer/reference/frontend/framework_overview.html>

When we include or extend the class we need to declare a separate module and create a new Var by using require.

For example, if we need to extend the functionality of var AbstractAction in KioskMode, then we need to extend the AbstractAction by

**odoo.define(‘module\_name.name’, function(require){**

**“use strict”;**

**Var AbstractAction = require(‘web.AbstractAction’);**

**Var KioskMode = AbstractAction.extend({**

**Events: {**

//write the events to be executed here like click action

**},**

**Start: function(){**

//Start function will automatically run during reload of the page.

**},**

**});**

**core.action\_registry.add(‘module\_name.name’, KioskMode);**

**return KioskMode;**

//This will extend the functionality of AbstractionAction and register the extended functionality in the core. If we need to extend the KioskMode then we need to extend it by doing the same.

**});**

**Difference between extend and include on a class.**

1. When we use extend, instances from the parent class remain untouched, but instances from the new child class will have the extended functionality.
2. When we use include, then we are adding the new features to the prototype of the parent class, which means that automatically all instances of the class include the extended behavior of the class. It is like Monkey patching the existing class.

If we don't extend the events, then the events will be overwritten if the event is not registered in the registry. In order to avoid this, we need to extend the events by extending the **prototype.events** like the following example

**var newVariable = variable**.extend({

events: \_.extend({

“click.o\_button\_name”: function(){

this.call\_method();

}

}, variable.**prototype.events**),

});

**To work with Mobile there is a separate documentation** <https://www.odoo.com/documentation/15.0/developer/reference/frontend/mobile.html>

In order to use the Mobile methods, the mobile API can be used anywhere by getting the object from web\_mobile.rpc

var mobile = require(‘web\_mobile.rpc’);

There are certain mobile RPC methods that are available. Each method returns a JQuery Deferred Object which returns a data JSON dictionary.

There are certain arguments that are given in the documentation for functions that we can use.  
We cannot extend the methods, we can just use the methods.

In order to work with camera, there are some **web API like navigator**

<https://developer.mozilla.org/en-US/docs/Web/API/Navigator> where we can get the camera by using Navigator.mediaDevices() which asks the user for permission.  
We can access the camera if the user allows the permission by getting the element and by starting the stream.  
There are two modes to select the camera by using facingMode as a parameter.

1. facingMode : “user” —> Opens Front Camera
2. facingMode: “environment” —> Opens rear camera mode

We can pass other arguments like height, width, etc.,

There are some libraries like

**Quagga** <https://serratus.github.io/quaggaJS/>

**Zxing** <https://openbase.com/js/@zxing/browser/documentation>

**OpenCv** <https://docs.opencv.org/4.6.0/>

We can use these libraries to work with camera-related works.

To work with face recognition, use the library face-api.js

<https://justadudewhohacks.github.io/face-api.js/docs/index.html>

Library to recognize the human face

<https://github.com/vladmandic/human>

To work with a webcam use this library

<https://github.com/jhuckaby/webcamjs>