

```

grasppe = eval("(function (w) {'use strict'; if (typeof w.grasppe !== 'function') w.grasppe = class
  grasppe{constructor({});}}(this)); grasppe");

(function (window, grasppe, undefined) {
  'use strict';

  grasppe.require(grasppe.load.status.initialize, function () {

    // !- [ColorSheetsApp]
    grasppe.ColorSheetsApp = grasppe.Libre.Module.define(class ColorSheetsApp extends
      grasppe.Libre.Module {
        // !- ColorSheetsApp [Constructor]
        constructor() {
          var args = [... arguments],
              options = (args.length > 0 && typeof args.slice(-1)[0] === 'object') ? args.pop() :
              undefined;
          if (!options.values) options.values = {};
          if (!options.values.model) options.values.model = grasppe.ColorSheetsApp.Model;
          else Object.assign(options.values.model, grasppe.ColorSheetsApp.Model,
            options.values.model);
          super(options);
        }
        initializeComponents() {
          INITIALIZE_SHEETS: {
            if (!this.sheets) this.sheets = {};
            for (var key in this.sheets) this.sheets[key].directives &&
              this.initializeDirectives(this.sheets[key].directives),
              this.sheets[key].controllers &&
              this.initializeControllers(this.sheets[key].controllers);
            this.values.model.sheets = this.sheets;
          }
        }
        set sheets(sheets) {
          this.hash.sheets = sheets;
        }
        get sheets() {
          return this.hash.sheets;
        }
      }, {
        // !- ColorSheetsApp [Prototype]
        description: 'Graphic arts theory demos!', version: (1.0),
        requirements: ['ngMaterial', 'ngAnimate'],
        controller: grasppe.Libre.Controller.define('ColorSheetsAppController', function ($scope,
          model, module) {
          // !- ColorSheetsApp [Controllers] ColorSheetsAppController
          $scope.model = model, $scope.module = module, $scope.parameters = {};
          $scope.layout = model.layout, $scope.panels = model.panels, $scope.sheets = module.sheets;

          function selectSheet(key) {
            if (!key) key = Object.keys($scope.sheets)[0];
            $scope.sheet = $scope.sheets[key], $scope.sheet.id = key;
            for (var panelKey of ['stage', 'parameters', 'results', 'overview'])
              $scope.panels[panelKey].template = $scope.sheet[panelKey].template,
              $scope.panels[panelKey].directive = $scope.sheet[panelKey].directive;
          }

          selectSheet();
          // console.log('ColorSheetsAppController', this, arguments);
        }, {
          $providers: ['$libreModel'],
        }),
        controllers: {},
        template: '<div color-sheets-sheet-panel layout="column"></div><color-sheets-styles></color-
          sheets-styles>', configuration: [function ($mdIconProvider) {
          $mdIconProvider.defaultFontSet('glyphicon').defaultIconSet('icon-set.svg', 20); // Register
            a default set of SVG icons
          }],
        directives: {
          // !- ColorSheetsApp [Directives] CopyrightsDirective
          copyrightsDirective: grasppe.Libre.Directive.define('copyrights', {
            // link: function ($scope, element, attributes) {},
            template: 'Copyrights &copy; 2015, Saleh Abdel Motaal, Franz Sigg, and Grasppe, Inc.',
          }),
        },
      },
    );
  });
}

```

```
// !- ColorSheetsApp [Directives] colorSheetsStyles
colorSheetsStyles: graspe.Libre.Directive.define('colorSheetsStyles', {
  link: function ($scope, element, attributes) {
    Object.assign($scope, {
      panelHeaderHeight: '36px', mainHeaderHeight: '48px',
    })
  },
  template: '<style>\n
    @media all {\n
      :not(input):not(.selectable) {user-select:none;cursor:default;}\n
      * {outline-style:none;outline-width:0;}\n
      md-toolbar.color-sheets-toolbar, md-toolbar.color-sheets-toolbar > * {max-
        height: {{panelHeaderHeight}}; min-height:
          {{panelHeaderHeight}};}\n
      .color-sheets-sheet-panel-header md-toolbar.color-sheets-toolbar, .color-
        sheets-sheet-panel-header md-toolbar.color-sheets-toolbar > *
          {max-height: {{mainHeaderHeight}}; min-height:
            {{mainHeaderHeight}};}\n
    }\n
    @media screen {\n
      body {margin:0;padding:0;min-width:100vw;min-height:100vh;}\n
    }\n
    @media print {\n
      md-slider {display: none !important;}\n
      md-button * {opacity: 0.5 !important;}\n
    }\n
  </style>',
}),
// !- ColorSheetsApp [Directives] colorSheetsControl
colorSheetsControl: graspe.Libre.Directive.define('colorSheetsControl', {
  link: function ($scope, element, attributes) {
    $scope.control = {
      id: 'control-id', label: 'control-label', description: 'control-description',
    }
  },
  template: '\n
    <div layout>\n
      <div flex="10" layout layout-align="center center"><span class="md-
        body-1">{{control.label}}</span></div>\n
      <ng-transclude flex></ng-transclude>\n
      <div flex="20" layout layout-align="center center">\n
        <input flex type="number" ng-model="{{control.model}}" aria-
          label="{{control.description}}" aria-controls="{{control.id}}">\n
      </div>\n
    </div>',
}),
// !- ColorSheetsApp [Directives] colorSheetsSliderControl
colorSheetsSliderControl: graspe.Libre.Directive.define('colorSheetsSliderControl',
function () {
  return {
    link: function ($scope, element, attributes) {
      // console.log('colorSheetsSliderControl :: preLink', $scope.parameters);
      $scope.control = {
        id: attributes.id, label: attributes.label, description:
          attributes.description, minimum: Number(attributes.minimum),
        maximum: Number(attributes.maximum),
        step: Number(attributes.step),
        model: attributes.model, value: Number(localStorage.getItem($scope.sheet.id
          + '-' + attributes.model)) || Number(attributes.value),
        suffix: attributes.suffix,
      }
      element.find('md-slider, input').attr('ng-model', 'parameters.' +
        $scope.control.model);
    },
    controller: ['$scope', '$element', function ($scope, element) {
      $scope.$watch('control.value', function (current, last, $scope) {
        if (current !== $scope.parameters[$scope.control.model] && current !==
          undefined) {
          $scope.parameters[$scope.control.model] = current;
          localStorage.setItem($scope.sheet.id + '-' + $scope.control.model,
            current);
        }
      });
    }],
  },
},

```

```

template: '\
<div layout>\
  <div flex="30" layout layout-align="center center"><span class="md-
    body-1">{{control.label}}</span></div>\
  <md-slider flex min="{{control.minimum}}" max="{{control.maximum}}" aria-
    label="{{control.description}}" id="{{control.id}}" class ng-
    model="control.value"></md-slider>\
  <div flex="15" layout layout-align="center center"><input flex type="number"
    min="{{control.minimum}}" max="{{control.maximum}}" step="{{control.step}}"
    aria-label="{{control.description}}" aria-controls="{{control.id}}" ng-
    model="control.value"></div>\
  <div flex="10" layout layout-align="center center"><span class="md-
    body-1">{{control.suffix}}</span></div>\
</div>', scope: true,
  });
  });
  // !- ColorSheetsApp [Directives] colorSheetsImageControl
  colorSheetsImageControl: graspe.Libre.Directive.define('colorSheetsImageControl', function
  () {
    return {
      link: function ($scope, element, attributes) {},
      controller: ['$scope', '$element', '$mdToast', '$mdDialog', function ($scope,
        element, $mdToast, $mdDialog) {
          var toast = function (message) {
            $mdToast.show($mdToast.simple().content(message))
          },
          error = function (message) {

            $mdDialog.show($mdDialog.alert().openFrom(element).closeTo(element)
              .clickOutsideToClose(true).title('Image not
                saved...').content(message).ariaLabel('Image not
                saved...').ok('OK'))
          };
          element.find('.image-preview').bind('drop', function (event) {
            if (event.originalEvent.dataTransfer.files.length == 1) {
              Object.assign(new FileReader(), {
                onload: function (event) {
                  if (event.target.result.match(/^data:image\/(png|jpeg|gif|
                    svg)/)) $scope.control.value = event.target.result, $scope.
                    $apply();
                  else error('Only png, gif, jpeg, and svg images can be used!');
                },
              }).readAsDataURL(event.originalEvent.dataTransfer.files[0]);
            } else {
              $scope.control.value = Object.assign(new Image, {
                src: event.originalEvent.dataTransfer.getData('URL'),
                crossOrigin: 'Anonymous',
              }).src;
            }
          }).bind('dragenter', function (event) {
            $(this).css('background-color', 'rgba(255, 0, 0, 0.5)');
          }).bind('dragleave drop', function (event) {
            $(this).css('background-color', 'transparent');
          });
          $scope.$watch('control.value', function (current, last, $scope) {
            if (current !== $scope.parameters[$scope.control.model] && current !==
              undefined) $scope.parameters[$scope.control.model] = current;
            if (current !== undefined && localStorage.getItem($scope.sheet.id + '-' +
              $scope.control.model) !== current) try {
              localStorage.setItem($scope.sheet.id + '-' + $scope.control.model,
                current);
              toast('Image saved!');
            } catch (err) {
              error('The file will not be saved!');
            }
            $scope.control.text = (current !== undefined) ? '' : 'Drop Image Here!';
          });
        }
      },
      link: function preLink($scope, element, attributes, controller) {
        $scope.control = {
          id: attributes.id, label: attributes.label, description:
            attributes.description, suffix: attributes.suffix, value:
              localStorage.getItem($scope.sheet.id + '-' + attributes.model) || '',
          model: attributes.model, text: '',

```

```

    }
    element.find('md-slider, input').attr('ng-model', 'parameters.' +
        $scope.control.model);
  },
  template: '\
<div layout>\
  <div flex="30" layout layout-align="center center"><span class="md-
    body-1">{{control.label}}</span></div>\
  <div flex class="image-preview md-button" style="height: 4em; background-
    position: center center; background-size: contain; background-repeat: no-
    repeat;" ng-style="{\'background-image\': \'url(\' + control.value + \')
    \'}" layout layout-align="center center">{{control.text}}</div>\
  <!--input flex type="image" aria-label="{{control.description}}" aria-
    controls="{{control.id}}" ng-model="control.value"-->\
  <!--div flex="10" layout layout-align="center center"><span class="md-
    body-1">{{control.suffix}}</span></div-->\
</div>', scope: true,
  },
  },
  // !- ColorSheetsApp [Directives] colorSheetsPanelTool
  colorSheetsPanelTool: graspe.Libre.Directive.define('colorSheetsPanelTool', {
    link: function ($scope, element, attributes) {
      if ($scope.tool.click) {
        element.attr('onclick', $scope.tool.click);
      }
    },
    template: ('<md-button class="{{tool.classes}}" aria-label="{{tool.label}}"> \
      <md-icon md-font-icon="{{tool.icon}}" class="toolsIconClasses
        {{tool.iconClasses}} glyphicon {{tool.icon}}" ng-style="{color:
          {{tool.color || toolsColor}}, fontSize: toolsIconSize}"></md-icon> \
      </md-button>'),
    scope: true,
  },
  // !- ColorSheetsApp [Directives] colorSheetsPanelTools
  colorSheetsPanelTools: graspe.Libre.Directive.define('colorSheetsPanelTools', {
    link: function ($scope, element, attributes) {
      $scope.toolsColor = ($scope.panel.toolsColor || $scope.model.toolsColor);
      $scope.toolsIconSize = ($scope.panel.toolsIconSize || $scope.model.toolsIconSize);
    },
    template: ('\
    <md-toolbar class="color-sheets-toolbar {{toolbarClasses}}"> \
      <div class="md-toolbar-tools" color="{{toolsColor}}"> \
        <md-button class="md-icon-button" aria-label="{{header}} Menu" ng-
          init="menuIcon = panel.icon"> \
          <md-icon md-font-icon="{{menuIcon}}" class="toolsIconClasses" ng-
            style="{color: toolsColor, fontSize: toolsIconSize}"></md-icon> \
        </md-button> \
        <header class="{{headerClasses}}" color="{{headerColor}}"><span>{{header}}</
          span></header> \
        <span flex></span> \
        <color-sheets-panel-tool ng-repeat="tool in tools"></color-sheets-panel-tool> \
      </div> \
    </md-toolbar>'),
    scope: true,
  },
  // !- ColorSheetsApp [Directives] colorSheetsPanel
  colorSheetsPanel: graspe.Libre.Directive.define('colorSheetsPanel', function () {
    return {
      link: function ($scope, element, attributes, controller) {
        if ($scope.segment.attributes) $(element).attr($scope.segment.attributes);
        Object.assign($scope, $scope.panel);
      },
      template: ('\
      <div class="color-sheets-panel {{panelClasses}} {{prefix || \'sheet\'}}-
        panel" {{section.attributes}} layout="column">\
      <div class="color-sheets-panel-header {{headerClasses}} {{prefix || \'sheet
        \'}}-panel-header" color-sheets-panel-tools>{{panel.header}}</div>\
      <div ng-switch="section.id">\
        <div ng-switch-when="stage" class="color-sheets-panel-contents
          {{contentClasses}} {{prefix || \'sheet\'}}-panel-contents" color-sheet-
          stage></div>\
        <div ng-switch-when="parameters" class="color-sheets-panel-contents
          {{contentClasses}} {{prefix || \'sheet\'}}-panel-contents" color-sheet-
          parameters></div>\
        <div ng-switch-when="results" class="color-sheets-panel-contents

```

```

        {{contentClasses}} {{prefix || \'sheet\'}}-panel-contents" color-Sheet-
        results></div>\
        <div ng-switch-when="overview" class="color-sheets-panel-contents
        {{contentClasses}} {{prefix || \'sheet\'}}-panel-contents" color-Sheet-
        overview></div>\
        <div ng-switch-default class="color-sheets-panel-contents
        {{contentClasses}} {{prefix || \'sheet\'}}-panel-contents"></div>\
    </div>\
    <div ng-if="footer" class="color-sheets-panel-footer {{footerClasses}} {{prefix
    || \'sheet\'}}-panel-footer">{{footer}}</div>\
</div>'),
    },
  },
  // !- ColorSheetsApp [Directives] colorSheetsSheetSegment
  colorSheetsSheetSegment: graspe.Libre.Directive.define('colorSheetsSheetSegment', {
    link: function ($scope, element, attributes) {
      if ($scope.segment.attributes) $(element).attr($scope.segment.attributes);
    },
    template: ('<div ng-repeat="section in segment.contents" {{section.attributes}}
    class="ng-class: section.classes;" ng-init="panel = panels[section.id]" ng-
    style="section.style;" color-sheets-panel="{{panel}}">{{panel}}</div>'),
  }),
  // !- ColorSheetsApp [Directives] colorSheetsSheetPanel
  colorSheetsSheetPanel: graspe.Libre.Directive.define('colorSheetsSheetPanel', {
    link: function ($scope, element, attributes) {
      Object.assign($scope, $scope.panel);
    },
    template: ('\
    <div ng-init="panel = panels[\'sheet\']" class="color-sheets-panel {{panelClasses}}
    {{prefix || \'sheet\'}}-panel" layout="column" >\
    <div class="color-sheets-sheet-panel-header {{prefix || \'sheet\'}}-panel-header"
    color-sheets-panel-tools>{{panel.header}}</div>\
    <div class="color-sheets-sheet-panel-contents {{prefix || \'sheet\'}}-panel-
    contents" color-Sheets-Sheet>\
    </div>\
    <div class="color-sheets-sheet-panel-footer {{prefix || \'sheet\'}}-panel-footer"
    copyrights></div>\
    </div>'),
  }),
  // !- ColorSheetsApp [Directives] colorSheetsSheet
  colorSheetsSheet: graspe.Libre.Directive.define('colorSheetsSheet', {
    link: function ($scope, element, attributes) {
      if ($scope.layout.attributes) $(element).attr($scope.layout.attributes);
    },
    template: ('<div class="color-sheets-sheet container-fluid
    {{layout.classes}}" {{layout.attributes}} ng-style="layout.style;" layout-fill>\
    <div ng-repeat="segment in layout.contents" {{segment.container}} class="ng-class:
    segment.classes;" ng-style="segment.style;" color-sheets-sheet-segment></div> \
    </div>'),
  }),
}, {
  // !- ColorSheetsApp [Model]
  Model: {
    toolsIconSize: '20px', toolsColor: 'white', menuIcon: 'glyphicon-menu-hamburger',
    toolsIconClasses: 'tools-icon', layout: {
      id: 'default', attributes: {
        'layout': 'column', 'layout-gt-lg': 'row',
      },
      classes: 'container-fluid', style: {
        padding: 0, margin: 0, flex: 1
      },
      contents: {
        simulation: {
          id: 'simulation', attributes: {
            'layout': 'row', 'layout-sm': 'column', 'layout-gt-lg': 'row',
          },
          classes: 'row col-xs-12', style: {
            padding: 0, margin: 0, flex: 1
          },
          contents: {
            stage: {
              id: 'stage', classes: 'col-xs-12 col-sm-6 col-md-7 col-lg-8', style: {
                padding: 0, margin: 0
              },

```

```

        },
        parameters: {
            id: 'parameters', classes: 'col-xs-12 col-sm-6 col-md-5 col-lg-4',
            style: {
                padding: 0, margin: 0
            },
        },
    },
},
information: {
    id: 'information', attributes: {
        'layout-sm': 'column', 'layout-md': 'row',
    },
    classes: 'row col-xs-12', style: {
        padding: 0, margin: 0, flex: 1
    },
    contents: {
        results: {
            id: 'results', attributes: {
                'flex-order-sm': -1,
            },
            classes: 'col-xs-12 col-sm-5', style: {
                padding: 0, margin: 0
            },
        },
        overview: {
            id: 'overview', attributes: {
                'flex-order-sm': 1,
            },
            classes: 'col-xs-12 col-sm-7', style: {
                padding: 0, margin: 0
            },
        },
    },
},
},
},
panels: {
    sheet: {
        prefix: 'sheet', header: 'ColorSheet', icon: 'glyphicon-menu-hamburger',
        toolbarClasses: 'grey lighten-2 black-text', tools: {
            refresh: {
                icon: 'glyphicon-refresh', label: 'refresh', classes: 'md-button-flat red
                    black-text', click: 'location.reload()',
            },
            documentation: {
                icon: 'glyphicon-book', label: 'documentation', classes: 'md-button-flat
                    orange black-text',
            },
        },
        contents: '', footer: '', // controller: 'SheetPanelController',
    },
    stage: {
        prefix: 'stage', header: 'Stage', icon: 'glyphicon-stats', toolbarClasses: 'grey
            darken-1', tools: {
            zoom: {
                icon: 'glyphicon-search', label: 'zoom', classes: 'md-icon-button black-
                    text', color: 'black',
            },
            style: {
                icon: 'glyphicon-tint', label: 'style', classes: 'md-icon-button black-
                    text', color: 'black',
            },
        },
        contents: '', footer: '', controller: 'StagePanelController',
    },
    parameters: {
        prefix: 'parameters', header: 'Parameters', icon: 'glyphicon-cog', toolbarClasses:
            'green lighten-1', contents: '', footer: '', tools: undefined, controller:
            'ParametersPanelController',
    },
    results: {
        prefix: 'results', header: 'Results', icon: 'glyphicon-th-list', toolbarClasses:
            'red lighten-1', contents: '', footer: '', tools: undefined, controller:
            'ResultsPanelController',
    },
},

```

---

```

        },
        overview: {
            prefix: 'overview', header: 'Overview', icon: 'glyphicon-edit', toolbarClasses:
                'light-blue lighten-1', contents: '', footer: '', tools: undefined, controller:
                'OverviewPanelController',
        },
    },
    });
});
}(this, this.grasppe));

```