

Getting Start

디펜던시

오픈그래프의 디펜던시는 JQuery, JQuery UI, JQuery context menu 입니다.

- [Jquery \(https://jquery.com/\)](https://jquery.com/)
- [Jquery UI \(https://jqueryui.com/\)](https://jqueryui.com/)
- [Jquery context menu \(http://swisnl.github.io/jQuery-contextMenu/index.html\)](http://swisnl.github.io/jQuery-contextMenu/index.html)

인스톨과 시작 예제

html 준비

먼저 빈 html 페이지 하나를 준비합니다.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko" lang="ko">
<head>
    <title>BPMN Modeler Example</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <meta http-equiv="X-UA-Compatible" content="IE=EmulateIE8">

</head>
<body>

</body>
</html>
```

라이브러리 인스톨

다음은 오픈그래프에서 기본적으로 제공하는 Symbol 이미지들과 디펜던시들을 예제 html 이 불러올 수 있도록 해야 합니다.

본 프로젝트의 webapp/examples 하위에 위치한 resources/images/symbol 와 webapp/examples/lib 폴더 하위에 위치한 opengraph,jquery,jquery-ui,contextmenu 폴더들을 찾도록 합니다.

이 폴더들을 예제 html 과 동일한 디렉토리 구조(또는 url 패스)로 복사하도록 합니다.

```
--- /a directory/resources/images/symbol
    /lib/opengraph
    /lib/jquery
    /lib/jquery-ui
    /lib/contextmenu
    /your-html-file
```

위에서 언급한 디펜던시 파일들과 오픈그래프 라이브러리 js 를 예제 html 로 임포트합니다.

임포트를 완료한 페이지의 샘플 코드는 다음과 같습니다.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="ko" lang="ko">
<head>
    <title>BPMN Modeler Example</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>
    <meta http-equiv="X-UA-Compatible" content="IE=EmulateIE8">

    <!-- jquery -->
    <script type="text/javascript" src="./lib/jquery-1.11.1/jquery-
1.11.1.min.js"></script>

    <!-- jquery ui -->
    <script type="text/javascript" src="./lib/jquery-ui-1.11.0.custom/jquery-
ui.min.js"></script>
    <link rel="stylesheet" type="text/css" href="./lib/jquery-ui-
1.11.0.custom/jquery-ui.css"/>

    <!-- jquery Context Menu -->
    <link rel="stylesheet" type="text/css"
href="./lib/contextmenu/jquery.contextMenu.css"/>
    <script type="text/javascript" src="./lib/contextmenu/jquery.contextMenu-
min.js"></script>

    <!-- Opengraph -->
    <script type="text/javascript" src="./lib/opengraph/OpenGraph-0.1.1-
SNAPSHOT.js"></script>

</head>
<body>

</body>
</html>

```

캔버스 렌더링

계속해서, 시작 예제를 그릴 캔버스 Div 하나를 화면에 선언합니다.

```

<body>

<div id="canvas" style="cursor: default;"></div>

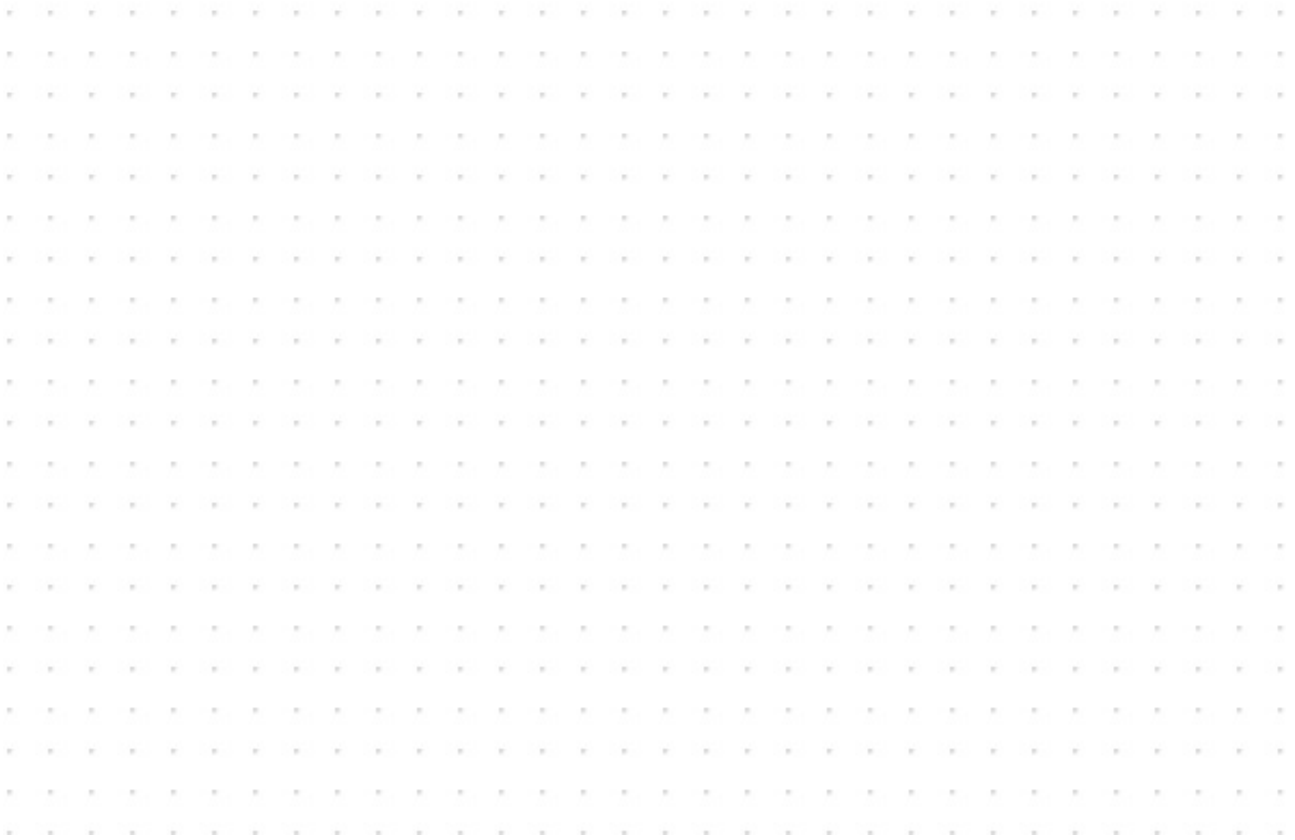
</body>

```

선언한 div 를 오픈그래프 캔버스로 렌더링합니다.

new OG.Canvas(div 아이디, 캔버스 사이즈, 백그라운드 컬러, 백그라운드 이미지) 메소드를 호출함으로써 캔버스를 렌더링 할 수 있습니다.

```
<script type="text/javascript">
    $(document).ready(function () {
        var canvas = new OG.Canvas('canvas', [1000, 800], 'white',
        'url(resources/images/symbol/grid.gif)');
    });
</script>
```

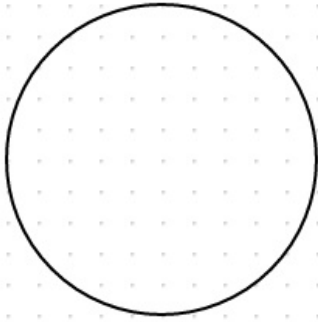


도형 그리기

원형의 도형을 캔버스에 하나 그려보도록 합니다.

canvas.drawShape(좌표[x,y], 도형 , 도형 사이즈[가로,세로]);

```
<script type="text/javascript">
    $(document).ready(function () {
        var canvas = new OG.Canvas('canvas', [1000, 800], 'white',
        'url(resources/images/symbol/grid.gif)');
        var circleShape = canvas.drawShape([100, 100], new
        OG.CircleShape(), [100, 100]);
    });
</script>
```



여기서 캔버스의 `drawShape` 메소드를 통해서 도형을 그릴때, 그려야 할 도형으로 `new OG.CircleShape()` 를 선언하였습니다.

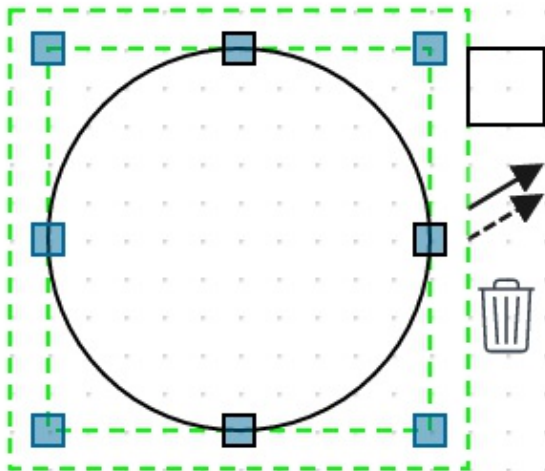
Shape 은 오픈그래프에서 기본 제공하는 Shape 들 이외에도 사용자가 직접 커스터마이징 하여 정의할 수 있습니다.

Shape 에 대한 종류와 커스터마이징에 대해서는 [Shape \(shape.md\)](#) 페이지를 참조하십시오.

선 연결

오픈그래프의 도형들은 기본적으로 UI 상에서 Copy, 선 연결, 삭제 할 수 있는 아이콘을 제공하고, 사이즈 재조정 및 이동이 가능합니다.

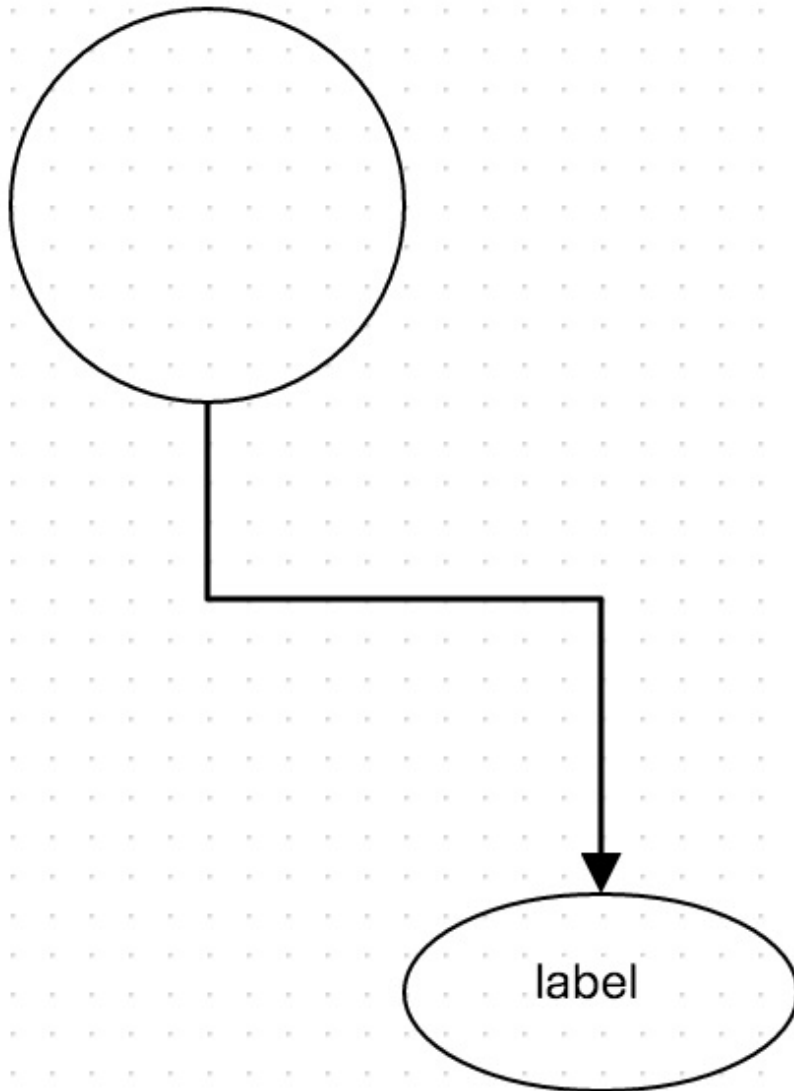
처음 `circleShape` 를 클릭하여 보면 다음의 아이콘들이 나타납니다.



두 개의 도형을 연결하기 위해 하나의 도형을 캔버스에 더 그려보도록 합니다.

```
<script type="text/javascript">
    $(document).ready(function () {
        var canvas = new OG.Canvas('canvas', [1000, 800], 'white',
        'url(resources/images/symbol/grid.gif)');
        var circleShape = canvas.drawShape([100, 100], new
OG.CircleShape(), [100, 100]);
        var ellipseShape = canvas.drawShape([200, 300], new
OG.EllipseShape('label'), [100, 50]);
    });
</script>
```

처음 circleShape 의 클릭하여 선연결 모양의 아이콘을 클릭 또는 드래그 하여, ellipseShape 를 클릭하거나 마우스 업을 실행하여 보십시오.



위와 같은 이벤트를 프로그램적으로 처리할 수 있습니다.

canvas.connect(선행 도형, 후행 도형);

```
<script type="text/javascript">
  $(document).ready(function () {
    var canvas = new OG.Canvas('canvas', [1000, 800], 'white',
    'url(resources/images/symbol/grid.gif)');
    var circleShape = canvas.drawShape([100, 100], new
OG.CircleShape(), [100, 100]);
    var ellipseShape = canvas.drawShape([200, 300], new
OG.EllipseShape('label'), [100, 50]);
    var edge = canvas.connect(circleShape, ellipseShape);
  });
</script>
```

줌 패널

줌 패널을 추가하기 위해서는 줌 패널로 사용할 div 를 하나 선언합니다.

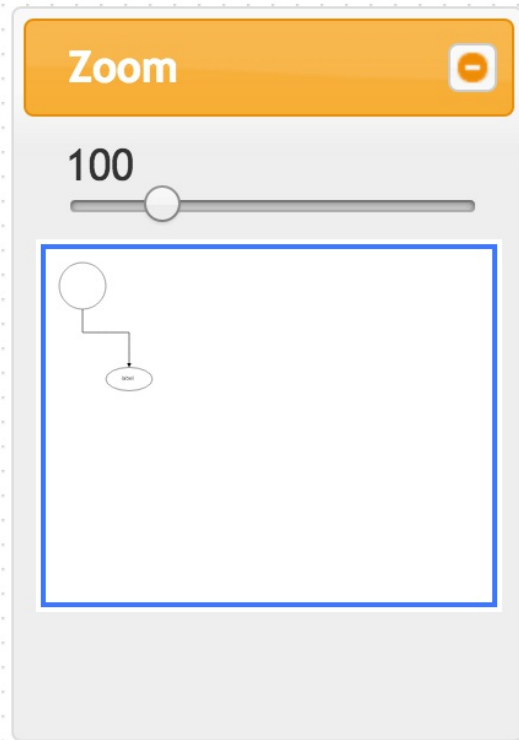
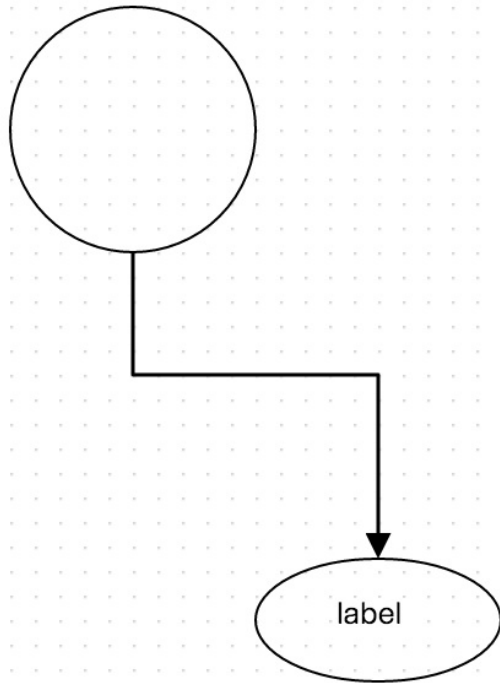
```
<body>

<div id="canvas" style="cursor: default;"></div>
<div id="canvas_slider"></div>
</body>
```

이후에 스크립트에서 슬라이더를 추가하는 명령어를 수행해줍니다.

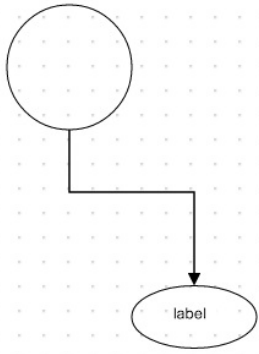
```
<script type="text/javascript">
    $(document).ready(function () {
        var canvas = new OG.Canvas('canvas', [1000, 800], 'white',
'url(resources/images/symbol/grid.gif)');
        canvas.addSlider({
            slider: $("#canvas_slider"),
            width: 200,
            height: 300,
            appendTo: "body"
        });

        var circleShape = canvas.drawShape([100, 100], new
OG.CircleShape(), [100, 100]);
        var ellipseShape = canvas.drawShape([200, 300], new
OG.EllipseShape('label'), [100, 50]);
        var edge = canvas.connect(circleShape, ellipseShape);
    });
</script>
```



줌 패널의 기능을 `canvas.setScale(0 ~ 1)` 을 통해 프로그램적으로 수행할 수 있습니다.

```
<script type="text/javascript">
    $(document).ready(function () {
        var canvas = new OG.Canvas('canvas', [1000, 800], 'white',
        'url(resources/images/symbol/grid.gif)');
        canvas.addSlider({
            slider: $("#canvas_slider"),
            width: 200,
            height: 300,
            appendTo: "body"
        });
        canvas.setScale(0.51);
    });
</script>
```

Zoom

51

A zoomed-in view of the diagram from the left, enclosed in a blue rectangular border. It shows a circle connected by a line to a smaller oval labeled "label".