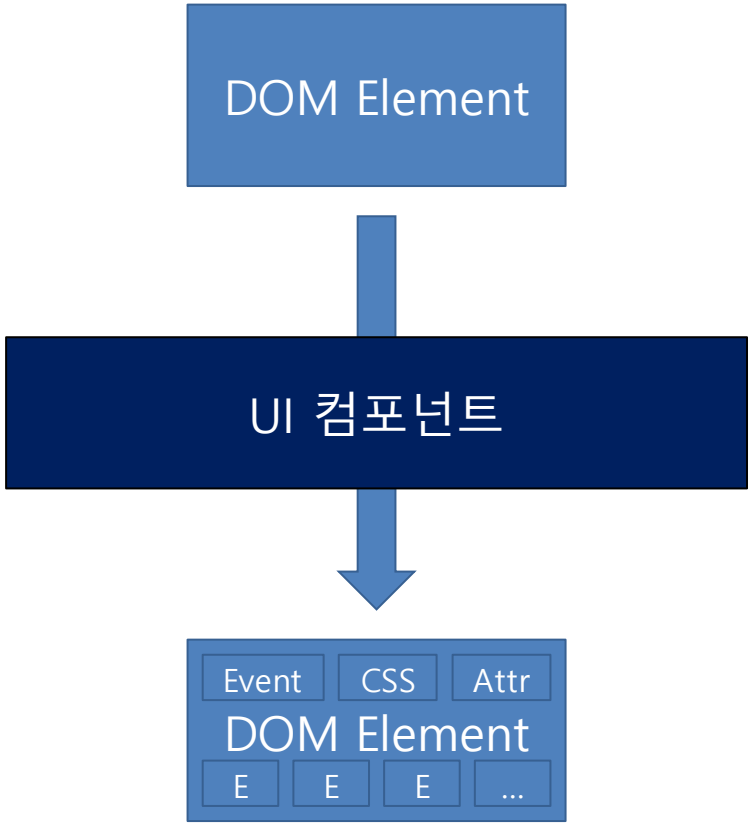
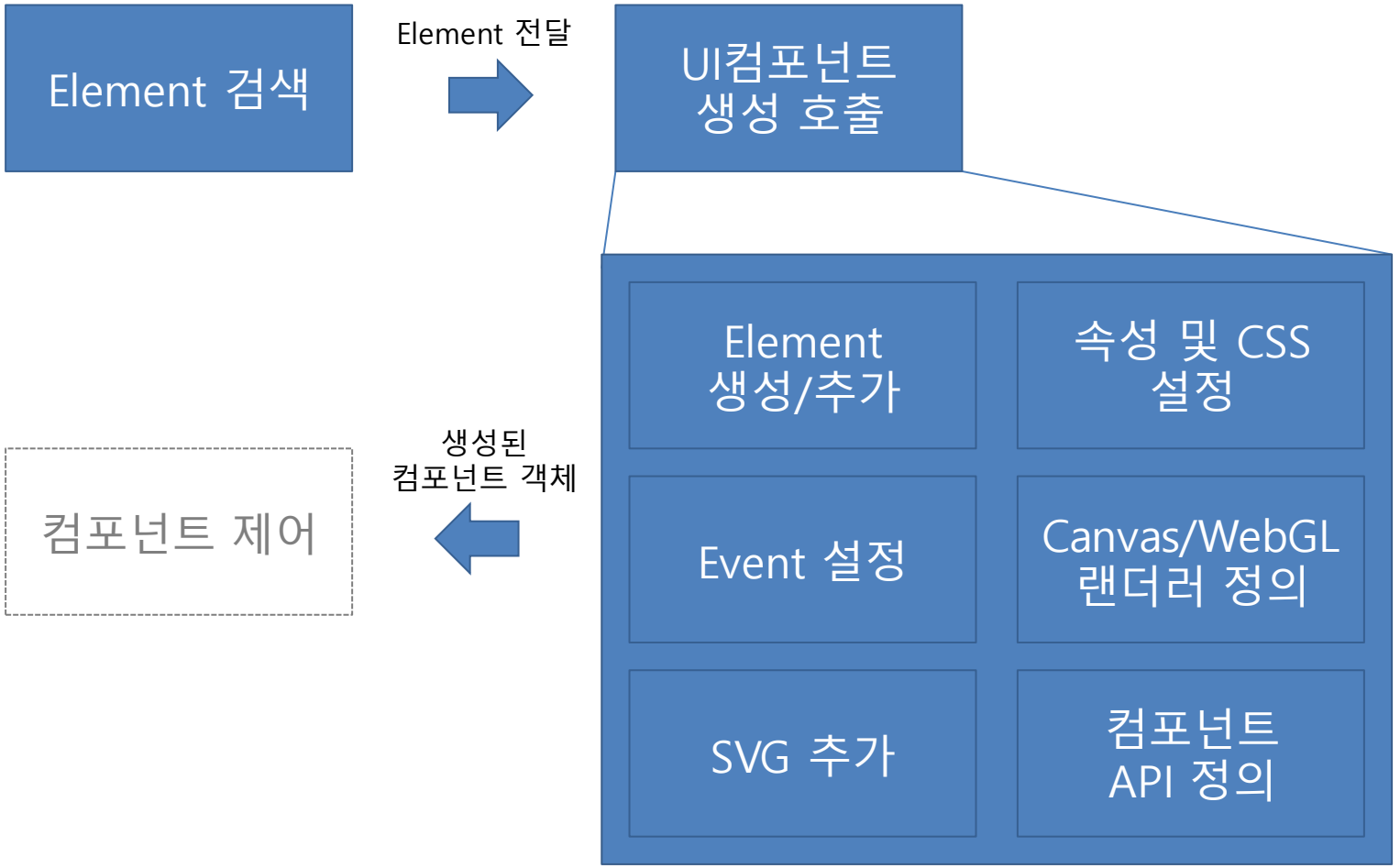
커스텀 UI 컴포넌트

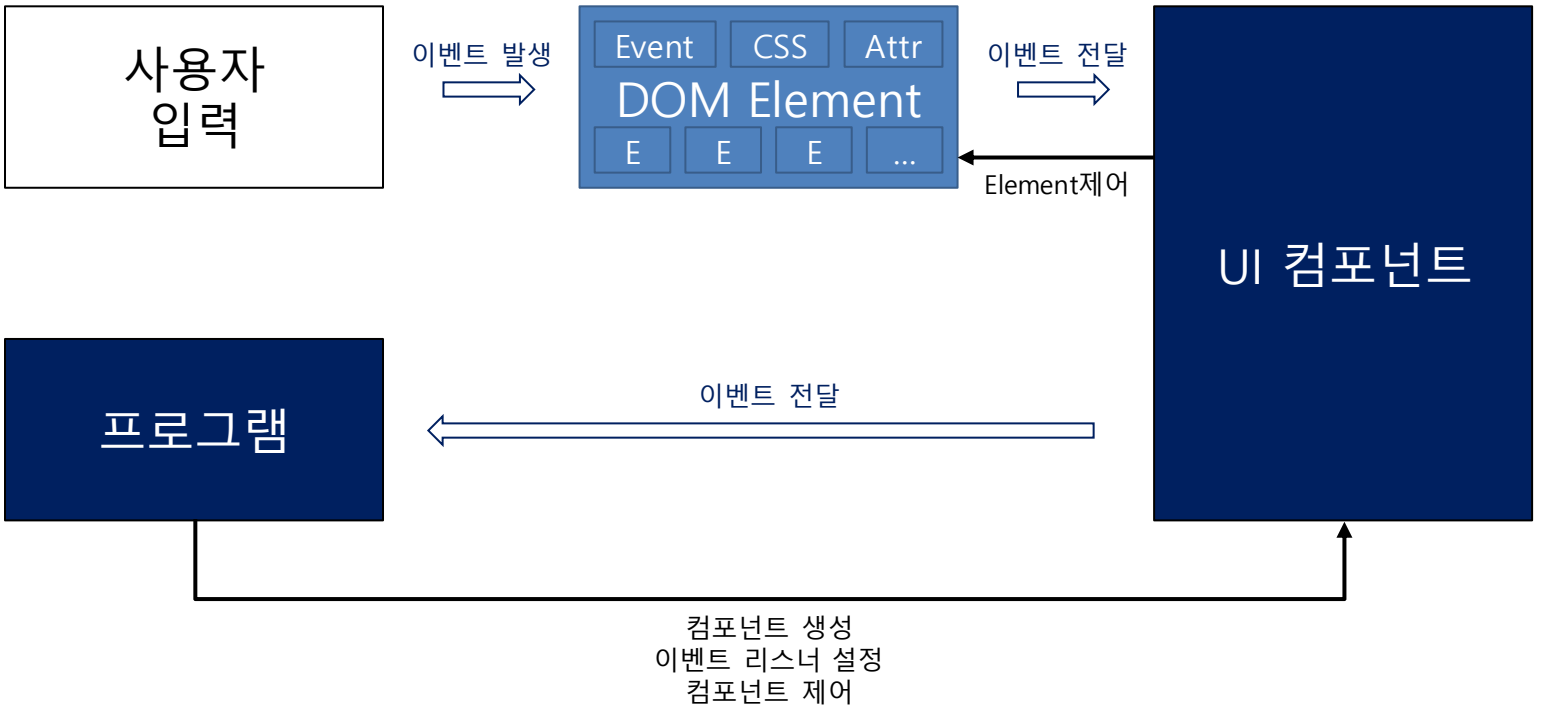
1. 커스텀 UI 컴포넌트 원리



2. 커스텀 UI 컴포넌트 생성



3. 커스텀 UI 컴포넌트 동작



4-1. UI 컴포넌트 예(1)

(kendoui - Slider)

```
<div class="demo-section k-header" style="width: 300px;">
    <div id="wrapper">
        <input id="slider" class="balSlider" value="0" />
        <div id="equalizer">
            <input class="eqSlider" value="10" />
            <input class="eqSlider" value="5" />
            <input class="eqSlider" value="0" />
            <input class="eqSlider" value="10" />
            <input class="eqSlider" value="15" />
        </div>
    </div>
</div>
```

<HTML 문서에 Element배치>

Element 검색

UI 컴포넌트 생성

생성된
UI 컴포넌트 객체

```
var slider = $("#slider").kendoSlider({
    increaseButtonTitle: "Right",
    decreaseButtonTitle: "Left",
    min: -10,
    max: 10,
    smallStep: 2,
    largeStep: 1
}).data("kendoSlider");

$(".eqSlider").kendoSlider({
    orientation: "vertical",
    min: -20,
    max: 20,
    smallStep: 1,
    largeStep: 20,
    showButtons: false
});
```

<UI 컴포넌트 생성>

```
▼<div class="demo-section k-header" style="width: 300px;">
  ▼<div id="wrapper">
    ▼<div class="k-widget k-slider k-slider-horizontal balSlider k-state-default">
      ▼<div class="k-slider-wrap k-slider-buttons">
        ▶<a class="k-button k-button-increase">…</a>
        ▶<a class="k-button k-button-decrease">…</a>
        ▶<ul class="k-reset k-slider-items">…</ul>
        ▼<div class="k-slider-track" style="width: 170px;">
          ▶<div class="k-slider-selection" style="width: 85px;">…</div>
            <a href="#" class="k-draghandle" title="drag" role="slider" aria-valuemi
            valuetext="0" style="left: 78px;">Drag</a>
          </div>
          <input id="slider" class="balSlider" value="0" type="text" data-role="slid
        </div>
      </div>
    ▼<div id="equalizer">
      ▼<div class="k-widget k-slider k-slider-vertical eqSlider k-state-default">
        ▼<div class="k-slider-wrap">
          ▶<ul class="k-reset k-slider-items">…</ul>
          ▼<div class="k-slider-track" style="height: 120px;">
```

<UI 컴포넌트 생성 후, 추가된 Element>

4-2. UI 컴포넌트 예(2)

(jindo - Slider)

```
▼<div id="simpleSlider" class="slider">
    <div class="slider-thumb"> </div>
  </div>
</div>
```

<HTML 문서에 Element배치>

생성된 UI 컴포넌트 객체

UI 컴포넌트 생성

Element 검색

```
var oSlider = new jindo.Slider(jindo.$('simpleSlider'), nMinValue : 100, nMaxValue : 200 });
```

<UI 컴포넌트 생성>

```
▼<div id="simpleSlider" class="slider">
    <div class="slider-thumb S56753602"> </div>
  </div>
</div>
```

<UI 컴포넌트 생성 후, 속성이 변경됨>
<HTML 문서상 바뀐게 별로 없어보이지만,
슬라이더 동작을 위한 이벤트 처리가 추가됨>

4-3. UI 컴포넌트 예(3)

(jQWidget - Slider)

```
<div style="float: left">
    <span style="font-style: italic;">Red</span>
    <div id='redLevel'>
    </div>
    <span style="font-style: italic;">Green</span>
    <div id='greenLevel'>
    </div>
    <span style="font-style: italic;">Blue</span>
    <div id='blueLevel'>
    </div>
</div>
```

<HTML 문서에 Element배치>

Element 검색

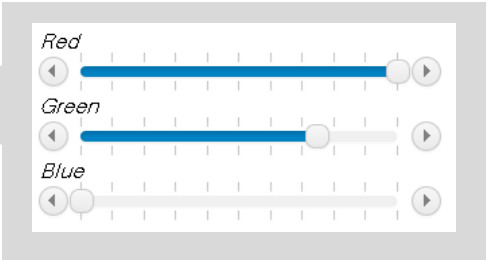
UI 컴포넌트 생성

```
$('#redLevel').jqxSlider({ min: 0, max: 255, ticksFrequency: 25.5, va
$('#greenLevel').jqxSlider({ min: 0, max: 255, ticksFrequency: 25.5, v
$('#blueLevel').jqxSlider({ min: 0, max: 255, ticksFrequency: 25.5, va
$('#redLevel').on('change', function (event) {
    setColor();
});
$('#greenLevel').on('change', function (event) {
    setColor();
});
$('#blueLevel').on('change', function (event) {
    setColor();
});
```

<UI 컴포넌트 생성, 이벤트 설정>

```
▼<div style="float: left">
   <span style="font-style: italic;">Red</span>
  ▼<div id="redLevel" role="slider" class="jqx-slider jqx-slider-arctic jqx-widge
  disabled="false" tabindex="0" style="width: 300px; height: 30px;">
    ▼<div class="jqx-slider-left jqx-rc-all jqx-rc-all-arctic jqx-button jqx-butt
    state-normal jqx-fill-state-normal-arctic" id="jqxWidget81fc5e07" role="buttor
    margin-left: 0px; margin-top: 3px; margin-right: 9px; background-position: 50%
        <div style="width: 100%; height: 100%;" class="jqx-icon-arrow-left jqx-ico
      </div>
    ▶<div style="float: left; width: 235px; height: 30px;">…</div>
    ▶<div class="jqx-slider-right jqx-rc-all jqx-rc-all-arctic jqx-button jqx-but
    state-normal jqx-fill-state-normal-arctic" id="jqxWidgetb5e588fc" role="buttor
    margin-left: 11px; margin-top: 3px; background-position: 50% 50%;">…</div>
      <input type="hidden" value="255">
    </div>
   <span style="font-style: italic;">Green</span>
  ▶<div id="greenLevel" role="slider" class="jqx-slider jqx-slider-arctic jqx-wid
  "255" aria-disabled="false" tabindex="0" style="width: 300px; height: 30px;">…</
   <span style="font-style: italic;">Blue</span>
  ▶<div id="blueLevel" role="slider" class="jqx-slider jqx-slider-arctic jqx-widg
  "false" tabindex="0" style="width: 300px; height: 30px;">…</div>
 </div>
```

<UI 컴포넌트 생성 후, 추가된 Element>



5. UI 컴포넌트 구현 방법

- Element로 Node계층을 구성하여 구현
  - UI컴포넌트를 생성할 Element의 부모/자식 노드에 Element를 추가
  - 추가된 각 Element에 이벤트 처리를 넣음

- Canvas에 랜더링하여 구현
  - Element 아래에 UI랜더링을 담당하는 Canvas Element를 추가
  - 하나의 Element가 UI 모든 영역의 이벤트를 받아서 처리

    관련 UI

    http://gojs.net/latest/samples/flowchart.html

    UI 사용예

    https://github.com/NorthwoodsSoftware/GoJS-Samples/blob/master/samples/flowchart.html

- SVG를 이용하여 구현
  - SVG 및 도형 Element 생성시, createElementNS를 이용하여 네임스페이스를 지정

    document.createElementNS("http://www.w3.org/2000/svg", "svg");
  - Element 아래에 SVG Element넣고, 각종 도형을 Element로 추가
  - setAttribute를 이용하여 도형의 속성을 변경하면, 바로 적용됨
  - 각각의 도형에 대해서 이벤트 처리를 할 수 있음

    관련 UI, 사용예

    http://demos.telerik.com/kendo-ui/bar-charts/index

6-1. Element 검색

- HTML4 API를 이용한 방법
  - Element의 ID, Class, TAG이름으로 검색
    - document.getElementById("id") as Element
    - document.getElementsByClassName("class") as Element[]
    - document.getElementsByTagName("tag") as Element[]
    - element.getElementsByClassName("class") as Element[]
    - element.getElementsByTagName("tag") as Element[]

- HTML5 API를 이용한 방법
  - CSS Selector형식을 이용한 Query로 검색
    - document.querySelector("query") as Element
    - document.querySelectorAll("query") as Element[]
    - element.querySelector("query") as Element
    - element.querySelectorAll("query") as Element[]
      * 참고 : https://developer.mozilla.org/en-US/docs/Web/API/Document.querySelector

- jQuery를 이용한 방법
  - CSS Selector형식을 이용한 Query로 검색
    - $("query") 등
      * 참고 : http://api.jquery.com/category/selectors/

6-2. Element 검색

- CSS Selector
  - CSS에서 Style을 적용할 Element를 가르킬 때 사용
  - HTML5 Query API와 jQuery를 이용하여 DOM Element를 검색할 때 사용

    * HTML5 Query API 참고
      https://developer.mozilla.org/en-US/docs/Web/API/Document.querySelector

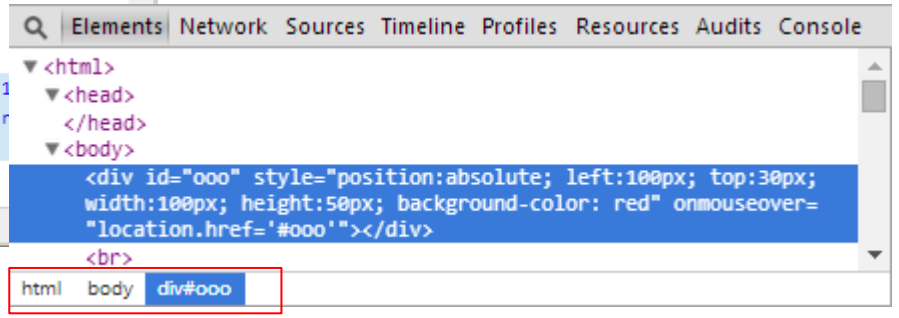
    * CSS Selector 자세한 내용
      http://www.w3.org/TR/2011/REC-css3-selectors-20110929/

패턴	설명	예
*	모든 태그	
#idname	id가 "idname"인 Element 검색	…
.classname	class이름이 "classname"인 Element 검색	…
tagname	"tagname" 타입의 Element 검색	<tagname xxx="xxx">…</tagname>
[attr]	"attr"속성을 가지는 Element 검색	…
[attr='value']	"attr"속성이 'value'인 Element 검색	<dir attr="value">…
패턴1, 패턴2	패턴1과 패턴2에 대한 모든 Element 검색	

6-3. Element 검색

- CSS Selector 확인
  - 브라우저 디버깅 기능을 통해 확인 가능
  - 아래 그림과 같이 선택한 태그에 대한 Selector가 표시됨
  - 다음과 같은 Selector로 검색 가능
    - "html body div#ooo"
    - "body div#ooo"
    - "div#ooo"
    - "#ooo"
    - "body #ooo" 등...

<Internet Explorer>



<Chrome>

7. Element 생성/추가

- HTML5
  - Element 생성
    - document.createElement("tagname")
    - document.createTextNode("text")
  - Child Element 추가
    - element.appendChild(e as Element)
    - element.insertBefore(e1 as Element, e2 as Element)
      - e2앞에 e1추가
    - element.removeChild(e as Element)
    - element.removeNode()
      - 자신을 삭제
- jQuery
  - Element 생성
    - $("<tagname>")
  - Child Element 추가
    - $("...").append(e)
    - $("...").prepend(e)
    - $("...").after(e)
    - $("...").before(e)

8. Element 속성/Data 설정

- 속성 설정
  - HTML5
    - element.setAttribute("attr", value)
    - element.getAttribute("attr", value)
  - jQuery
    - $("...").attr("attr", value)
- Data설정
  - HTML5
    - element.dataset["dataname"] = value
  - jQuery
    - $("...").data("dataname", value)
    - $("...").data("dataname")

9-1. CSS 적용

- 스타일 선언
  - CSS 파일에 작성하여 HTML에 포함

    <link rel="stylesheet" type="text/css" href="(url)">
  - HTML문서 내에서 style 태그 안에 작성

    <style type="text/css"> ... </style>
  - 스타일은 Selector와 선언 블록으로 구성되고 아래와 같이 작성

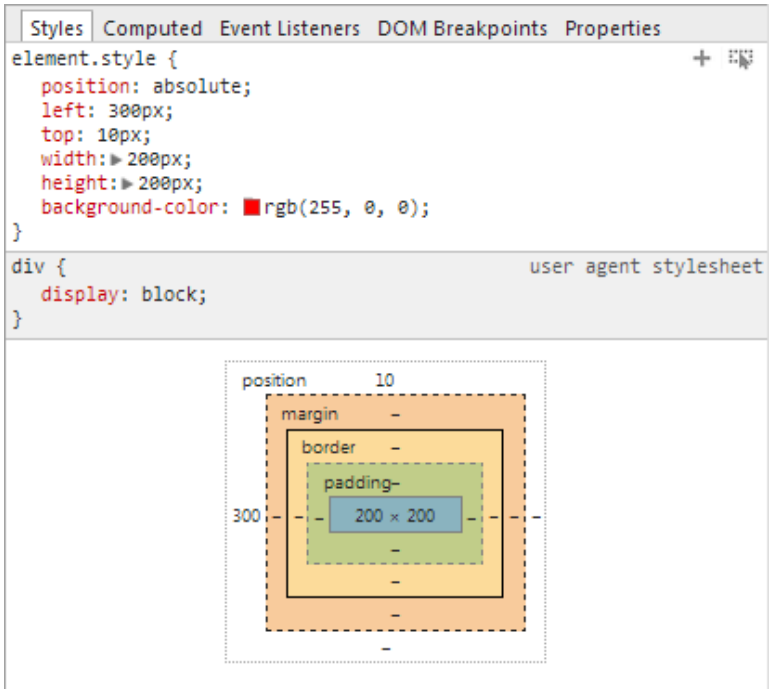
```
(selector) {
    (property) : (value) ;
    (property) : (value) ;
}
```

- 스타일 적용
  - HTML
    - element.classList.add("classname")
    - element.classList.remove("classname")
  - jQuery
    - $("...").css("property", value)
    - $("...").addClass("classname")
    - $("...").removeClass(" classname")

9-2. CSS 적용

- CSS Property
  - position
    - absolute : 절대 위치
    - fixed : 브라우저 화면상에 위치 고정
    - static : 기본값, 줄당 하나의 Element 배치
  - width, height
    - px, pt, cm, mm, em, in, %, calc()
  - left, top, right, bottom
    - px, pt, cm, mm, em, in, %, calc()
    - position이 static이 아닐 때만 적용됨
  - margin, padding
    - px, pt, cm, mm, em, in, %, calc()
  - background
    - background-image, background-color
    - 배경 색상/이미지 설정
    - url(), rgb(), rgba(), "#FFFFFF", "#FFFFFFFF"
  - border
    - border-color, border-style, border-width
    - 테두리 색상, 스타일, 너비 설정
    - border-style는 solid, dotted, dashed 등으로 설정

* Property/value 참고 : http://www.w3.org/TR/css-2010/

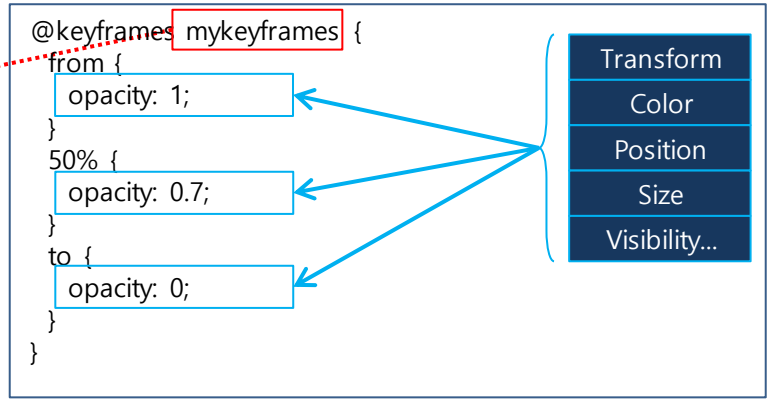


<Chrome Debug에서 Element에 적용된 스타일을 확인 할 수 있음>

!important

9-3. CSS 적용

- CSS Animation
  - animation-name
    - CSS문서내 @Keyframes으로 정의한 이름
    - @Keyframes로 타임라인을 정의
  - animation-duration
    - 애니메이션 동작 시간
  - animation-timing-function
    - ease-in, ease-out, ease-in-out
    - cubic-bezier(0.1, 0.7, 1.0, 0.1)
  - animation-delay
  - animation-iteration-count
    - 반복횟수, infinite
  - animation-play-state
    - paused, running

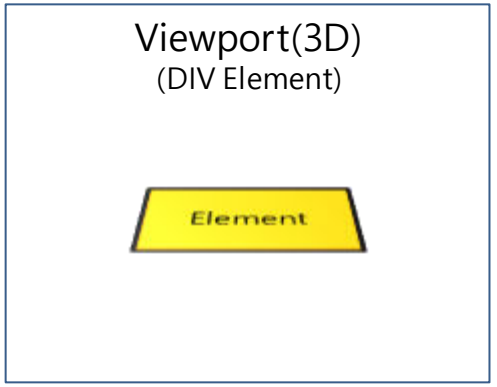


- CSS Animation 이벤트 처리
  - animationstart, animationend
    브라우저 별로 이벤트 이름에 접두어 붙음.
    webkit기반 **webkit**AnimationStart, Opera는 **o**animationstart, IE는 **MS**AnimationStart
  - 해당 Element에 addEventListener로 이벤트 처리 함수 추가

* 참고
애니메이션 정의 : https://developer.mozilla.org/en-US/docs/Web/CSS/animation
애니메이션 이벤트 처리 : http://www.sitepoint.com/css3-animation-javascript-event-handlers/

9-4. CSS 적용

- CSS Transform
  - perspective-origin
  - perspective
    - 3D변형할 Element의 부모에 설정
      perspective_pixel = 0.5 / tan(Fovy * 0.5)) * height
      ->(THREE CSSRenderer 참고)
  - transform-origin
  - transform

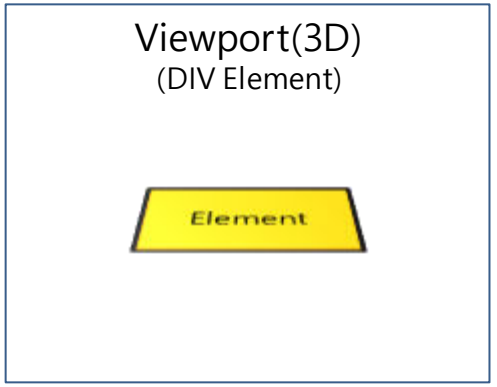


종류	CSS 함수
matrix	matrix(*n,n,n,n,n,n*) matrix3d(*n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n*)
translate	translate(*x,y*) translate3d(*x,y,z*) translateX(x) translateY(y) translateZ(*z*)
scale	scale(x, y) scale3d(*x,y,z*) scaleX(*x*) scaleY(*y*) scaleZ(*z*)
rotate	rotate(*angle*) rotate3d(*x,y,z,angle*) rotateX(*angle*) rotateY(*angle*) rotateZ(*angle*)
skew	skew(*x-angle,y-angle*) skewX(*angle*) skewY(*angle*)

* 참고
http://www.w3schools.com/cssref/css3_pr_transform.asp
http://www.w3.org/TR/css3-transforms/#transform-functions

9-4. CSS 적용

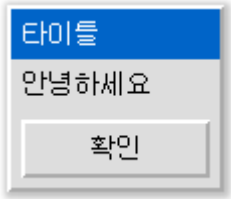
- CSS Transform
  - perspective-origin
  - perspective
    - 3D변형할 Element의 부모에 설정
      perspective_pixel = 0.5 / tan(Fovy * 0.5)) * height
      ->(THREE CSSRenderer 참고)
  - transform-origin
  - transform



종류	CSS 함수
matrix	matrix(*n,n,n,n,n,n*) matrix3d(*n,n,n,n,n,n,n,n,n,n,n,n,n,n,n,n*)
translate	translate(*x,y*) translate3d(*x,y,z*) translateX(x) translateY(y) translateZ(*z*)
scale	scale(x, y) scale3d(*x,y,z*) scaleX(*x*) scaleY(*y*) scaleZ(*z*)
rotate	rotate(*angle*) rotate3d(*x,y,z,angle*) rotateX(*angle*) rotateY(*angle*) rotateZ(*angle*)
skew	skew(*x-angle,y-angle*) skewX(*angle*) skewY(*angle*)

* 참고
http://www.w3schools.com/cssref/css3_pr_transform.asp
http://www.w3.org/TR/css3-transforms/#transform-functions

10-1. UI 컴포넌트 만들기



Element 배치, ID 정의

Element 검색

UI 생성

컴포넌트 API 호출

```
<div id="dialog1"></div>
<script>
    var dialogElement1 = document.querySelector("#dialog1");
    var dialog1 = new UISample(dialogElement1);
    dialog1.setTitle("타이틀");
    dialog1.setMessage("안녕하세요");
    dialog1.setOnOKClickListener(function(){
        alert("OK");
    });
</script>
```

Element 배치, view타입 설정

컴포넌트 API 대신 속성으로 설정

다이알로그 내용
(Message)

```
<div id="dialog1" data-view="uidialog" data-title="'샘플5' 입니다">
    <b>에러</b>가 발생하였습니다.<br>
    에러위치 : <font color="red">xxxxxxx</font> <i>yyyyyyyy</i> zzzzzzzz
</div>
```

* 예제 소스 : examples/ui_dialog_1.html

10-2. UI 컴포넌트 만들기 예

1. 타이틀, 텍스트, 버튼부분에 대한 태그 생성

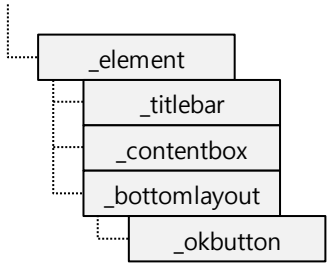
```
var _titlebar = document.createElement("div");
var _contentbox = document.createElement("div");
var _bottomlayout = document.createElement("center");
var _okbutton = document.createElement("div");
```

2. 속성 설정

```
if(element.dataset.title)
    _titlebar.textContent = element.dataset.title;
_okbutton.textContent = "확인";
```

```
var childCount = _element.childNodes.length;
for(var i = 0; i < childCount; i++)
    _contentbox.appendChild(_element.firstChild);
```

자식 Element들을
contentbox의 Element로 이동



3. Element에 추가

```
_element.appendChild(_titlebar);
_element.appendChild(_contentbox);
_element.appendChild(_bottomlayout);
_bottomlayout.appendChild(_okbutton);
```

4. CSS 설정(ClassName만 설정)

```
_element.classList.add("uidialog-dialog");
_titlebar.classList.add("uidialog-titlebar");
_contentbox.classList.add("uidialog-contentbox");
_okbutton.classList.add("uidialog-okbutton");
```

* 예제 소스 : examples/ui_dialog.js

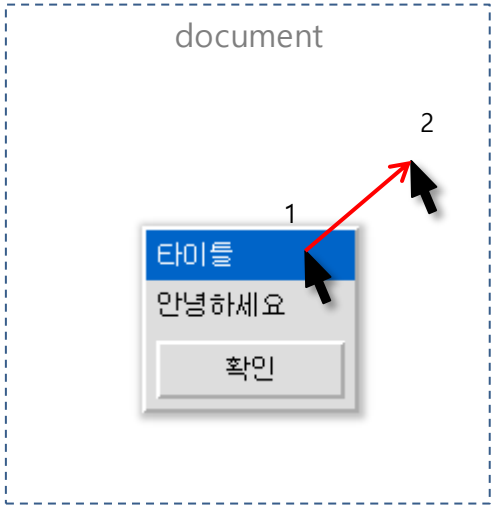
10-3. UI 컴포넌트 만들기 예

5. 이벤트 설정

```
var _onOKClickListener;
try{
    _onOKClickListener = new Function(element.dataset.onokclick);
}catch(err){
    console.log(err);
}

_titlebar.addEventListener("mousedown", function(e){
    var x = e.clientX - e.target.getBoundingClientRect().left;
    var y = e.clientY - e.target.getBoundingClientRect().top;

    var ox = x + _element.clientLeft;
    var oy = y + _element.clientTop;

    document.onmousemove = function(e){
        _element.style.left = (e.pageX - ox) + "px";
        _element.style.top = (e.pageY - oy) + "px";
    };
    document.onmouseup = function(e){
        document.onmousemove = null;
        document.onmouseup = null;
    }
});

_okbutton.addEventListener("click", function(e){
    if(_onOKClickListener)_onOKClickListener();
    _self.hide();
});
```

ok버튼 눌렀을 때 이벤트 처리를 data로 설정한 것

크롬에서는 layerX, offsetX를 사용할 수 있지만, 다른 브라우저에는 값이 없거나 다르므로 위와 같이 표준 API사용



mouse이벤트는 1번 위치에서 _titlebar, _element, document에 순서대로, 2번 위치에서는 document에만 전달 됨

타이틀 영역을 잡고 mousemove시,
마우스 포인터가 타이틀 영역 밖으로 넘어 갈경우,
이벤트가 document에만 전달되므로 move, up에 대한 이벤트 처리는 document에 설정

10-4. UI 컴포넌트 만들기 예

6. 컴포넌트 API 정의

```
this.setTitle = function(title){
    _titlebar.textContent = title;
};

this.setMessage = function(message){
    _textbox.textContent = message;
};

this.setOnOKClickListener = function(callback){
    _onOKClickListener = callback;
};

this.show = function(x, y){
    _element.style.left = x;
    _element.style.top = y;
    _element.style.opacity = 1.0;
    _element.hidden = false;
};

this.hide = function(){
    var interval = setInterval(function(){ // fadeout효과를 위해 Interval사용
        _element.style.opacity = _element.style.opacity * 0.4;
        if(_element.style.opacity < 0.01){
            _element.style.opacity = 0.0;
            _element.hidden = true;
            clearInterval(interval);
        }
    }, 100); // 100ms마다 function호출
};
```

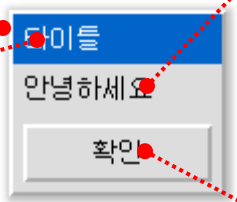
CSS로 처리할 수 있음

10-5. UI 컴포넌트 만들기 예

7. CSS 정의

```css
.uidialog-dialog {
    position: absolute;
    background-color: #DDDDDD;
    border: outset 2px white;
    cursor: default;
    box-shadow: 1px 1px 4px 2px rgba(0, 0, 0, 0.25);
}
```

```css
.uidialog-titlebar {
    position: relative;
    padding: 5px;
    color: #FFFFFF;
    background-color: #0064C8;
    font-size: 10pt;
}

.uidialog-titlebar:hover {
    background-color: #0086EA;
}

.uidialog-titlebar:active {
    background-color: #0064C8;
}
```




```css
.uidialog-contentbox {
    position: relative;
    margin: 5px;
    font-size: 10pt;
    cursor: auto;
}
```

```css
.uidialog-okbutton {
    position: relative;
    display: inline-block;
    min-width: 80px;
    margin: 5px;
    padding: 5px;
    background-color: #DDDDDD;
    border: outset 2px white;
    font-size: 10pt;
}

.uidialog-okbutton:hover {
    background-color: #EEEEEE;
}

.uidialog-okbutton:active {
    border: inset 2px white;
}
```

* 예제 소스 : examples/ui_dialog.css

10-6. UI 컴포넌트 만들기 예

8-1. CSS Animation 적용

```
.uidialog-show {
    animation-name: showdialog;
    animation-duration: 200ms;
    animation-timing-function: ease-out;

    visibility: visible;
}
```

```
@keyframes showdialog {
    from {
        opacity: 0;
        transform: scale(0, 0);
    }
    to {
        opacity: 1;
        transform: scale(1, 1);
    }
}
```

```
.uidialog-hide {
    animation-name: hidedialog;
    animation-duration: 300ms;
    animation-timing-function: ease-in;

    visibility: hidden;
}
```

```
@keyframes hidedialog {
    from {
        visibility: visible;
        opacity: 1;
        transform: scale(1, 1);
    }
    40% {
        transform: scale(1.2, 1.2);
    }
    to {
        opacity: 0;
        transform: scale(0.2, 0.2);
    }
}
```

10-7. UI 컴포넌트 만들기 예

8-2. CSS Animation 적용

```
this.setTitle = function(title){
    _titlebar.textContent = title;
};

this.setMessage = function(message){
    _textbox.textContent = message;
};

this.setOnOKClickListener = function(callback){
    _onOKClickListener = callback;
};

this.show = function(x, y){
    _element.style.left = x;
    _element.style.top = y;
    _element.style.opacity = 1.0;
    _element.hidden = false;
};

this.hide = function(){
    var interval = setInterval(function(){
        _element.style.opacity = _element.style.opacity * 0.4;
        if(_element.style.opacity < 0.01){
            _element.style.opacity = 0.0;
            _element.hidden = true;
            clearInterval(interval);
        }
    }, 100);
};
```

```
this.show = function(x, y){
    _element.classList.remove("uidialog-hide");
    _element.classList.add("uidialog-show");
};

this.hide = function(){
    _element.classList.remove("uidialog-show");
    _element.classList.add("uidialog-hide");

};
```