第51节 使用系统事件（aura:locationChange）

第5天

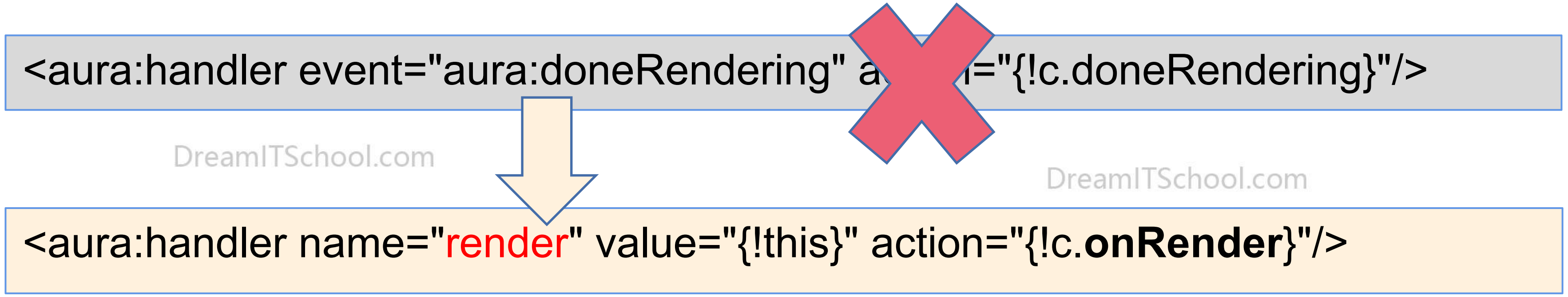
知识点(What)：
①使用aura:locationChange来判断URL变化
当URL变化时，会激发这个事件

```
<aura:handler event="aura:locationChange" action="{!c.UrlWasChanged}"/>
```

第52节 使用系统事件（aura:valueRender）

第5天

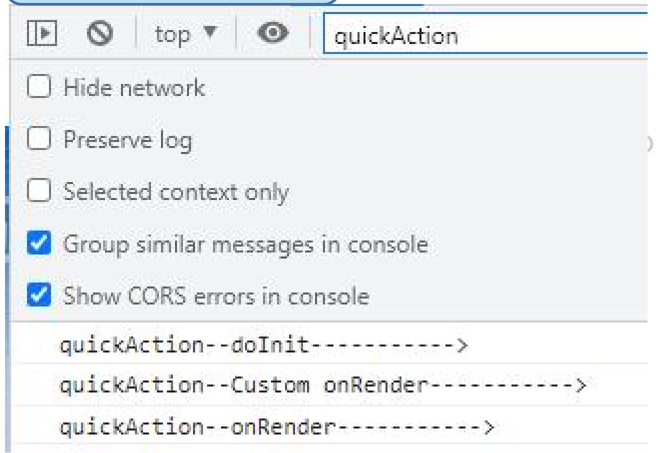
知识点(What)：
①使用aura:valueRender来做渲染事件的后处理
推荐使用aura:valueRender，而不是自定义Renderer.js(麻烦)



```
<aura:handler event="aura:doneRendering" action="{!c.doneRendering}"/>
```

```
<aura:handler name="render" value="{!this}" action="{!c.onRender}"/>
```

执行顺序



quickAction

□ Hide network
□ Preserve log
□ Selected context only
☑ Group similar messages in console
☑ Show CORS errors in console

```
quickAction--doInit----------->
quickAction--Custom onRender----------->
quickAction--onRender----------->
```

Demo50

第53节 使用系统事件（aura:doneWaiting ）

第5天

知识点(What)：
①使用aura:doneWaiting 来判断请求是否已经完成

```
<aura:handler event="aura:doneWaiting"
action="{!c.hideSpinner}"/>
```

```
var myAction = component.get("c.getCases");
myAction.setParams({"strRecordId":
recordIdFromController});
myAction.setCallback(this, function(response) {
    var state = response.getState();
    if (state === "SUCCESS") {
        //把处理写在这里
    }
});
$A.enqueueAction(myAction);
```

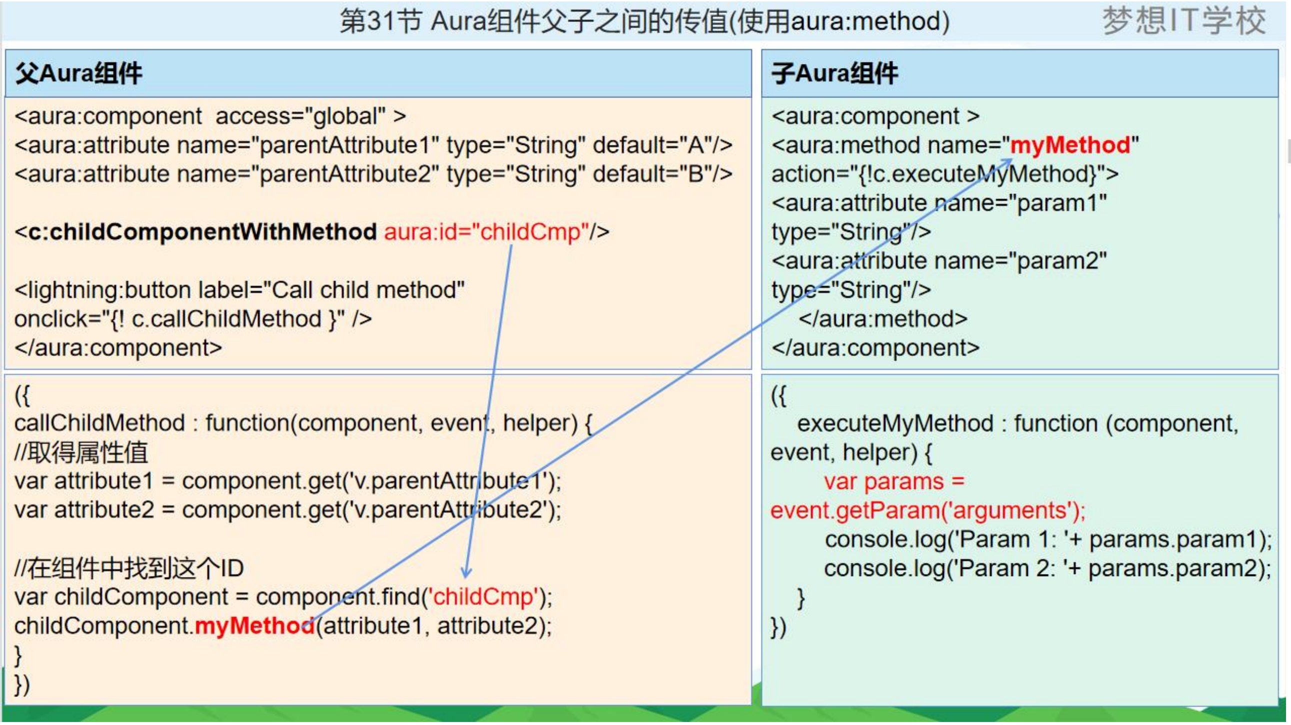
第54节 使用系统事件（Method Call）

第5天

知识点(What)：

①使用aura:method来定义一个方法作用Aura组件的API

父组件调用子组件的方法的时候，就用这个MethodCall事件



第31节 Aura组件父子之间的传值(使用aura:method)　梦想IT学校

父Aura组件

```
<aura:component  access="global" >
<aura:attribute name="parentAttribute1" type="String" default="A"/>
<aura:attribute name="parentAttribute2" type="String" default="B"/>

<c:childComponentWithMethod aura:id="childCmp"/>

<lightning:button label="Call child method"
onclick="{! c.callChildMethod }" />
</aura:component>
```

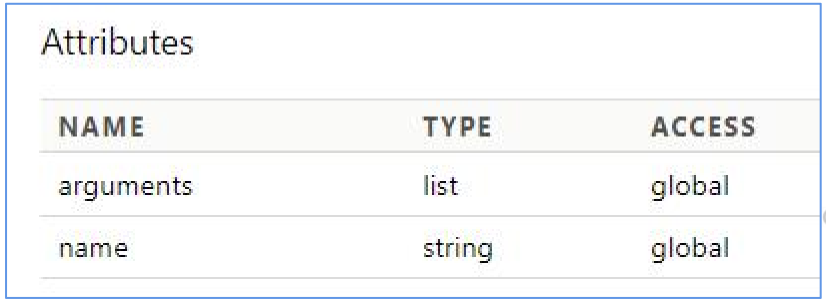
```
({
callChildMethod : function(component, event, helper) {
//取得属性值
var attribute1 = component.get('v.parentAttribute1');
var attribute2 = component.get('v.parentAttribute2');

//在组件中找到这个ID
var childComponent = component.find('childCmp');
childComponent.myMethod(attribute1, attribute2);
}
})
```

子Aura组件

```
<aura:component >
<aura:method name="myMethod"
action="{!c.executeMyMethod}">
<aura:attribute name="param1"
type="String"/>
<aura:attribute name="param2"
type="String"/>
   </aura:method>
</aura:component>
```

```
({
   executeMyMethod : function (component,
event, helper) {
        var params =
event.getParam('arguments');
        console.log('Param 1: '+ params.param1);
        console.log('Param 2: '+ params.param2);
    }
})
```



Attributes

NAME	TYPE	ACCESS
arguments	list	global
name	string	global

第55节 使用系统事件（System Error)

知识点(What)：
①使用aura:systemError来处理服务器出错

第5天

```
<aura:handler event="aura:systemError" action="{!c.handleError}"/>
```

调用Apex方法

```
var action = cmp.get("c.throwErrorApex");
action.setCallback(cmp, function(response){
    cmp.set("v.response", response);
});
$A.enqueueAction(action);
```

产生Exception的Apex代码

```
@AuraEnabled
public static void throwErrorApex() {
            Study__c m = new Study__c();
            insert m; // 必须输入字段没有指定，会出Exception
}
```

Demo51

第56节 使用系统事件（aura:valueDestroy）

第5天

知识点(What)：
①使用aura:valueDestroy来做组件消失前的处理

```
<aura:handler name="destroy" value="{!this}" action="{!c.handleDestroy}"/>
```

第57节 使用Workspace API（openTab）

第5天

知识点(What)：

①什么是WorkSpaceAPI，有什么用?

※ Lightning **console** app

②使用openTab来打开一个新的Console Tab

先在Component里面实例化一个WorkSpaceAPI

```
<lightning:workspaceAPI aura:id="workspace"/>
```

使用openTab来打开Console的一个Tab

```
var workspaceAPI = component.find("workspace");
workspaceAPI.openTab({
    url: '#/sObject/a005h00000mD2tuAAC/view',
    focus: true
});
```

Demo53

第58节 使用Workspace API（getFocusedTabInfo）

第5天

知识点(What)：
①使用getFocusedTabInfo来取得当前的TabId

先在Component里面实例化一个WorkSpaceAPI

```
<lightning:workspaceAPI aura:id="workspace"/>
```

【How】 使用getFocusedTabInfo来取得当前的TabId

```
var workspaceAPI = component.find("workspace");
workspaceAPI.getFocusedTabInfo().then(function(response) {
    var focusedTabId = response.tabId;
    //workspaceAPI.closeTab({tabId: focusedTabId});

    //设置当前Tab
    component.set("v.FocusedTab", focusedTabId);
})
.catch(function(error) {
    console.log(error);
});
```

Demo53

第59节 使用Workspace API（setTabLabel）

第5天

知识点(What)：
①使用setTabLabel来设置Tab的内容

【How】 使用setTabLabel来设置Tab的内容

```
var workspaceAPI = component.find("workspace");
workspaceAPI.getFocusedTabInfo().then(function(response) {
    var focusedTabId = response.tabId;

    //设置当前Tab内容
    workspaceAPI.setTabLabel({
        tabId: focusedTabId,
        label: "自己想设置的内容"
    });
})
.catch(function(error) {
    console.log(error);
});
```

Demo50

第60节 使用Workspace API（setTabIcon）

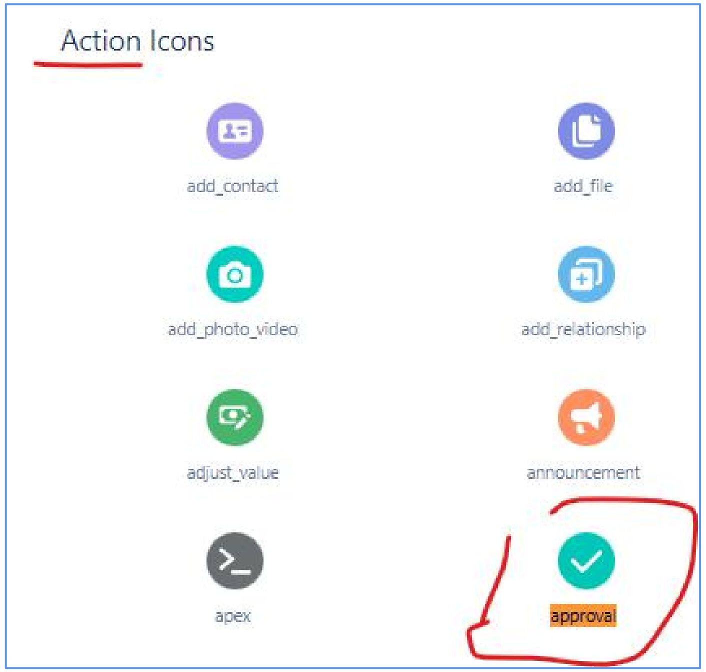
第5天

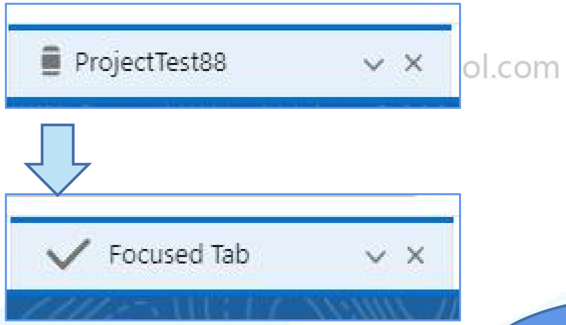
知识点(What)：
①使用setTabIcon来设置Tab的Icon

【How】 使用setTabIcon来设置Tab的Icon

```
var workspaceAPI = component.find("workspace");
workspaceAPI.getFocusedTabInfo().then(function(response) {
    var focusedTabId = response.tabId;

       //设置当前Tab的Icon
       workspaceAPI.setTabIcon({
          tabId: focusedTabId,
          icon: "action:approval",
          iconAlt: "Approval"
       });
})
.catch(function(error) {
    console.log(error);
});
```





Demo50

第61节 使用Workspace API（setTabHighlighted）

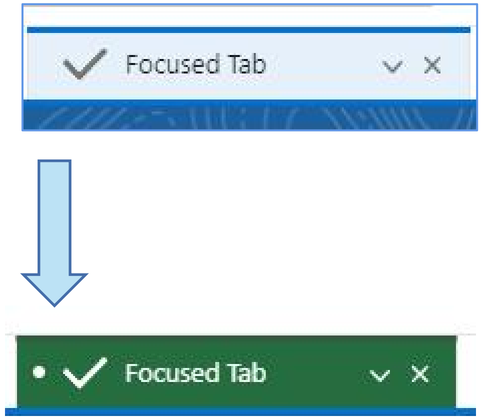
第5天

知识点(What)：
①使用setTabHighlighted来设置Tab为高亮

【How】 使用setTabLabel来设置Tab的内容

```
var workspaceAPI = component.find("workspace");
workspaceAPI.getFocusedTabInfo().then(function(response) {
    var focusedTabId = response.tabId;

    //设置当前Tab内容
       workspaceAPI.setTabHighlighted({
              tabId: focusedTabId,
              highlighted: true,
              options: {
              pulse: true,
              state: "success"
              }
       });
})
.catch(function(error) {
    console.log(error);
});
```



Demo53

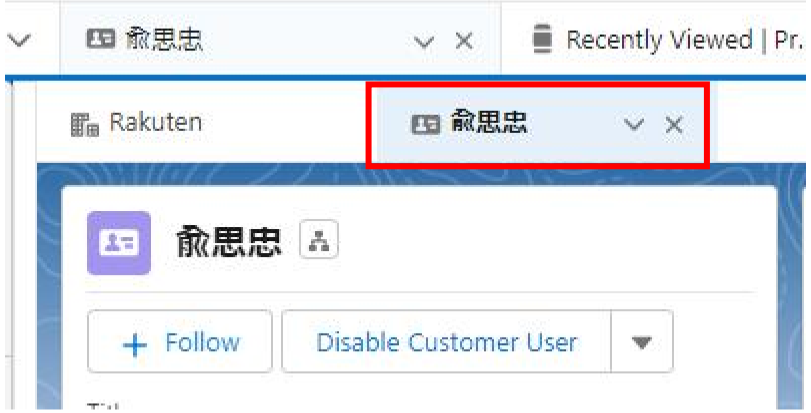
第62节 使用Workspace API（isSubtab）

第5天

知识点(What)：
①使用isSubtab来判断当前Tab是不是子Tab
(仅适用于Lightning console)

【How】 使用isSubtab来判断当前Tab是不是子Tab

```
var workspaceAPI = component.find("workspace");
workspaceAPI.getFocusedTabInfo().then(function(response) {
    workspaceAPI.isSubtab({
        tabId: response.tabId
    }).then(function(response) {
        if (response) {
            confirm("This tab is a subtab.");
        }else {
            confirm("This tab is not a subtab.");
        }
    });
})
.catch(function(error) {
    console.log(error);
});
```



Demo53

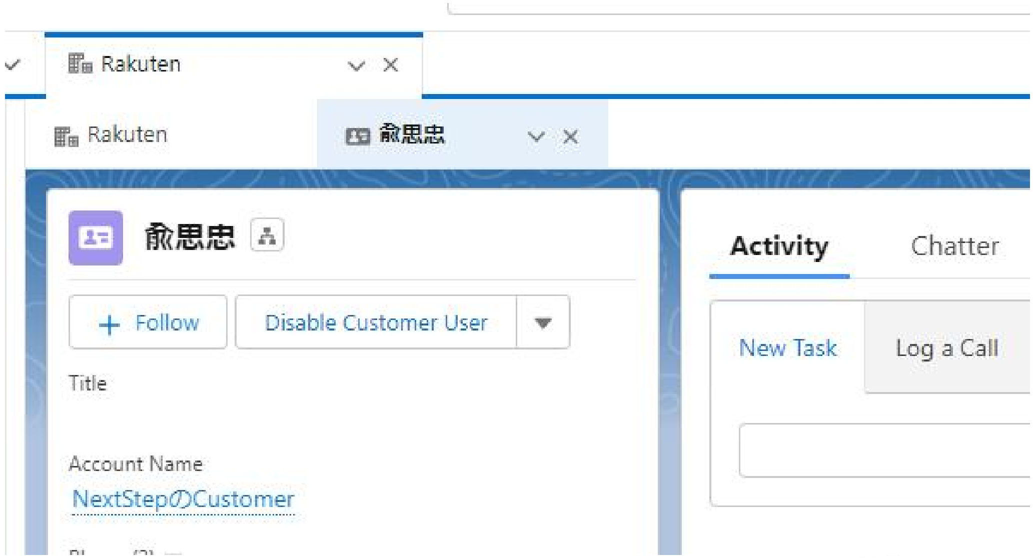
第63节 使用Workspace API（openSubtab）

第5天

知识点(What)：
①使用openSubtab来打开子Tab
(仅适用于Lightning console)

【How】 使用openSubtab来打开子Tab

```
var workspaceAPI = component.find("workspace");
workspaceAPI.openTab({
    url: '/lightning/r/Account/0015h000012S2KfAAK/view',
    focus: true
}).then(function(response) {
    workspaceAPI.openSubtab({
        parentTabId: response,
        url: '/lightning/r/Contact/0035h00000afZ0nAAE/view',
        focus: true
    });
})
.catch(function(error) {
    console.log(error);
});
```



Demo53

第64节 使用Utility Bar API（setUtilityIcon）

知识点(What)：
①什么是Utility Bar API，有什么用?
※ Lightning app
②使用setUtilityIcon来设置Utility Bar的图标

第5天

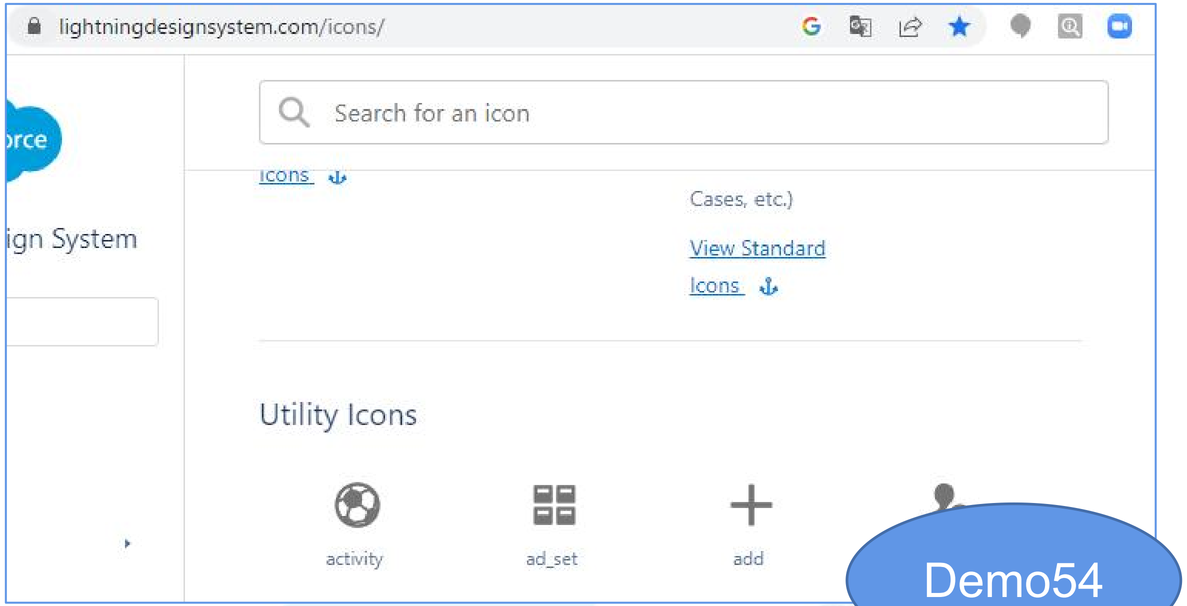
先在Component里面实例化一个utilityBarAPI

```
<lightning:utilityBarAPI aura:id="utilitybar"/>
```

使用setUtilityIcon来设置Utility Bar的图标

```
var utilityAPI = component.find("UtilityBar2023");
utilityAPI.setUtilityIcon({icon: 'insert_tag_field'});
```

注：
Icon的名称，也是在下面的网址内查找到
https://www.lightningdesignsystem.com/icons/


lightningdesignsystem.com/icons/

Demo54

第65节 使用Utility Bar API（setUtilityLabel）

梦想IT学校

知识点(What)：
①使用setUtilityLabel来设置Utility Bar的名称

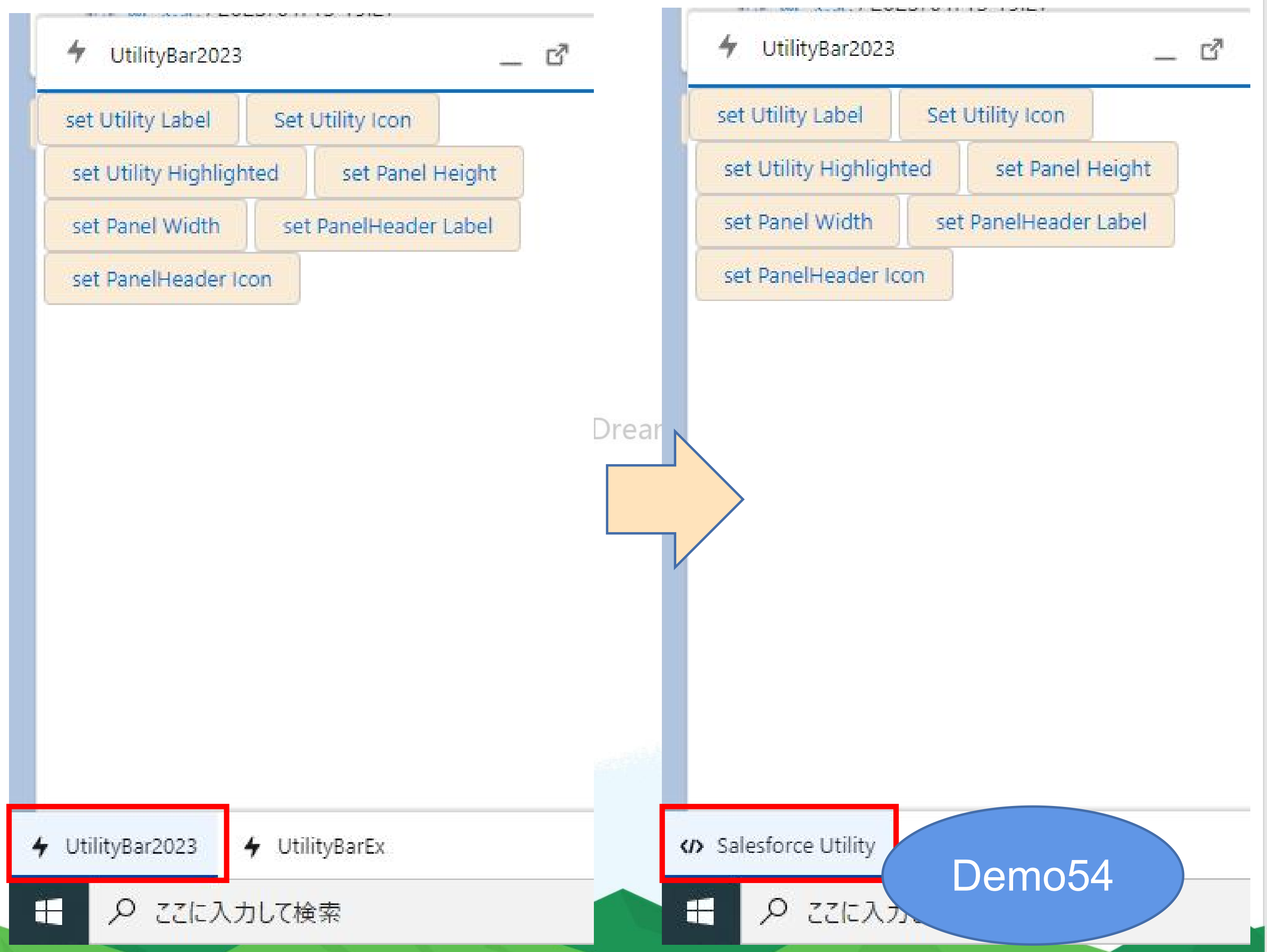
第5天

先在Component里面实例化一个utilityBarAPI

```
<lightning:utilityBarAPI aura:id="utilitybar"/>
```

使用setUtilityIcon来设置Utility Bar的图标

```
var utilityAPI = component.find("UtilityBar2023");
utilityAPI.setUtilityLabel({icon: 'Salesforce Utility'});
```



Demo54

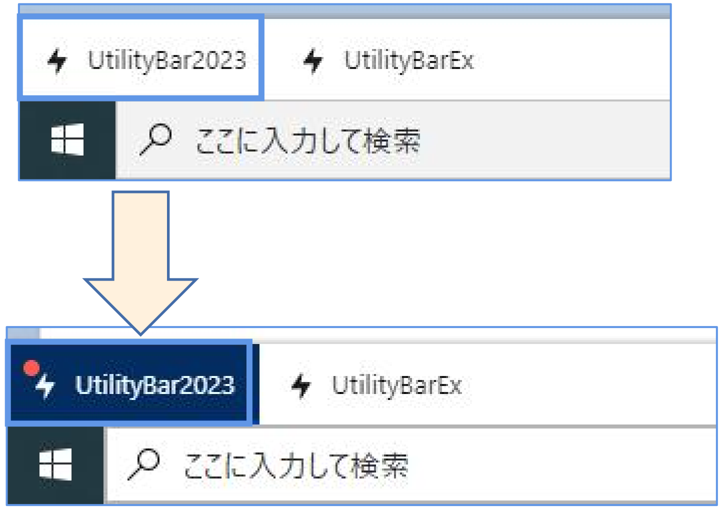
第66节 使用Utility Bar API（setUtilityHighlighted）

知识点(What)：
①使用setUtilityHighlighted来设置Utility Bar的高亮属性

第5天

先在Component里面实例化一个utilityBarAPI

```
<lightning:utilityBarAPI aura:id="utilitybar"/>
```



使用setUtilityHighlighted来设置Utility Bar的高亮属性

```
var utilityAPI = component.find("UtilityBar2023");
utilityAPI.setUtilityHighlighted({highlighted:true});
```

Demo54

第67节 使用Utility Bar API（setPanelWidth和setPanelHeight）

第5天

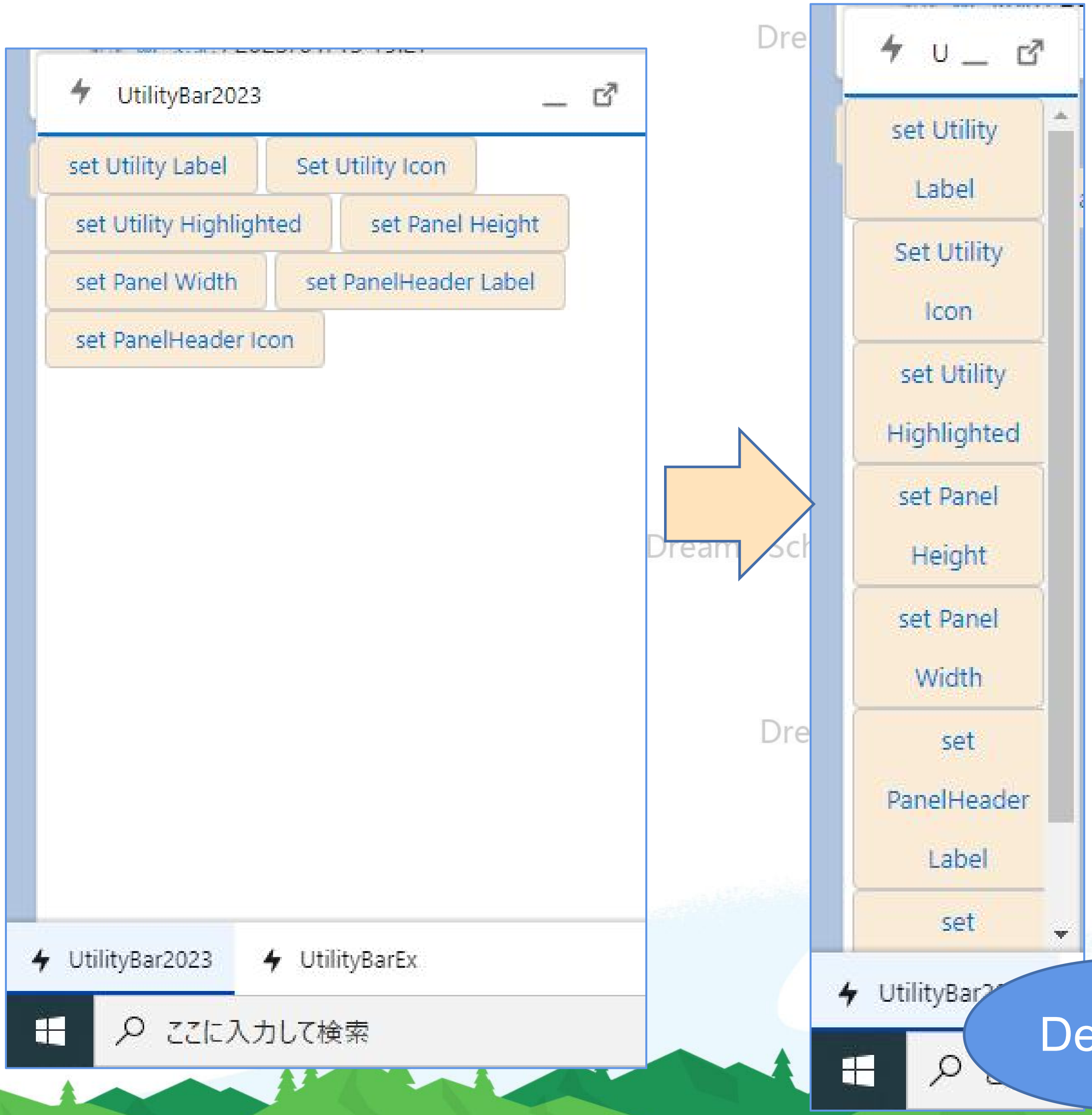
知识点(What)：
①使用setPanelWidth来设置Utility Bar的宽度
②使用setPanelHeight来设置Utility Bar的高度

先在Component里面实例化一个utilityBarAPI

```
<lightning:utilityBarAPI aura:id="utilitybar"/>
```

使用setPanelWidth来设置Utility Bar的宽度

```
var utilityAPI = component.find("UtilityBar2023");
utilityAPI.setPanelWidth({widthPX:120});
utilityAPI.setPanelHeight({heightPX :120});
```



Demo54

第68节 使用Utility Bar API（setPanelHeaderLabel）

知识点(What)：
①使用setPanelHeaderLabel来设置Utility Bar的头部名称

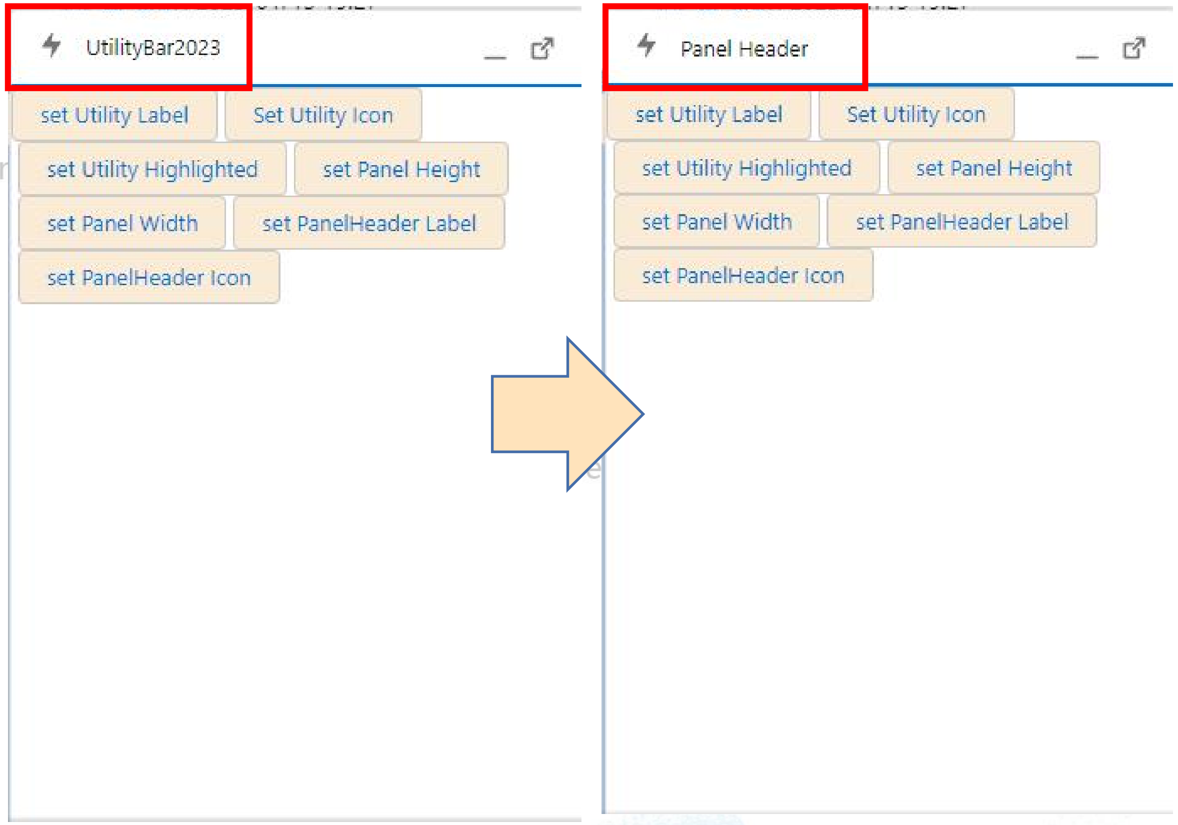
第5天

先在Component里面实例化一个utilityBarAPI

```
<lightning:utilityBarAPI aura:id="utilitybar"/>
```

使用setUtilityIcon来设置Utility Bar的图标

```
var utilityAPI = component.find("UtilityBar2023");
utilityAPI.setPanelHeaderLabel({label  :'Panel Header'});
```



Demo54

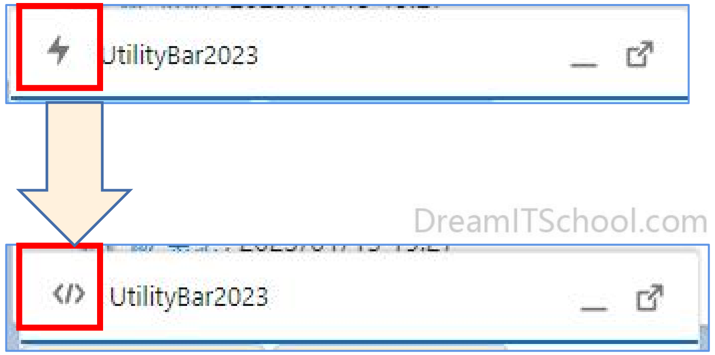
第70节 使用Utility Bar API（setPanelHeaderIcon）

知识点(What)：
①使用setPanelHeaderIcon来设置Utility Bar的头部图标

第5天

先在Component里面实例化一个utilityBarAPI

```
<lightning:utilityBarAPI aura:id="utilitybar"/>
```



使用setPanelHeaderIcon来设置Utility Bar的头部图标

```
var utilityAPI = component.find("UtilityBar2023");
utilityAPI.setPanelHeaderIcon({icon: 'insert_tag_field'});
```

Demo54

第71节 使用Utility Bar API（openUtility）

第5天

知识点(What)：
①使用openUtility来打开Utility Bar

先在Component里面实例化一个utilityBarAPI

```
<lightning:utilityBarAPI aura:id="utilitybar"/>
```

使用setPanelHeaderIcon来设置Utility Bar的头部图标

```
var utilityAPI = component.find("UtilityBar2023");
utilityAPI.getAllUtilityInfo().then(function(response) {
    var myUtilityInfo = response[0]; //
    utilityAPI.openUtility({
        utilityId: myUtilityInfo.id
    });
    //※如果不指定参数，就是打开自己当前的Utility Bar

})
.catch(function(error) {
    console.log(error);
});
```

Demo54