

Dojo Toolkit

A Training course for new developer

6 November 2022

Prepared by GIS Co., Ltd.

Nattaphon Churangsarit (Bank)

Enterprise GIS Application Architect Lead

An aerial photograph of a dense urban skyline, likely a major city, with numerous high-rise buildings and skyscrapers. The image is overlaid with a semi-transparent teal color. The text 'Course Outlines' is centered in a large, white, sans-serif font.

Course Outlines

To learn the **Dojo Toolkit** training course, you should be familiar with the following:

- [JavaScript](#)
- [HTML](#)
- [CSS](#)

Please bring your own Laptop with minimum spec.:

- CPU 2.0 GHz
- 64-bit architecture
- 4 or more CPUs/cores
- At least 8 GB of memory/RAM
- At least 4 GB of disk space
- Windows 10

Install programs to your laptop with the following:

- [Visual Studio Code](#)

- **Dojo Toolkit (1 MD)**
 - Concept
 - Widget
 - Declarative vs Programmatic
 - Create Custom Widget
 - Use Custom Widget
 - Useful modules
 - Get Started Coding with Dojo
 - Dojo with ArcGIS API for JavaScript

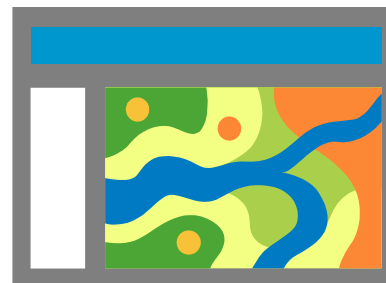
Learning objective

- Implement GIS web application using ArcGIS API for JavaScript
- Get to know basic dojo concept.
- Get to know the widget concept.
- Design and create custom widget for use in web application.
- Get to know about AMOS.

An aerial view of a dense city skyline, likely Manila, Philippines, featuring numerous high-rise buildings and a prominent skyscraper on the right. The entire image is overlaid with a semi-transparent teal color. The word "Concept" is centered in white text.

Concept

ArcGIS API for JavaScript



Website

Developer resources

- SDK Home page
- Esri on GitHub
- GeoNet

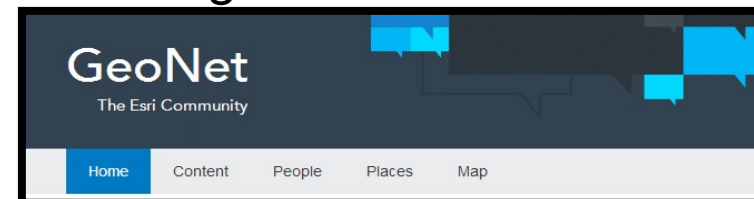
esri.github.io



js.arcgis.com

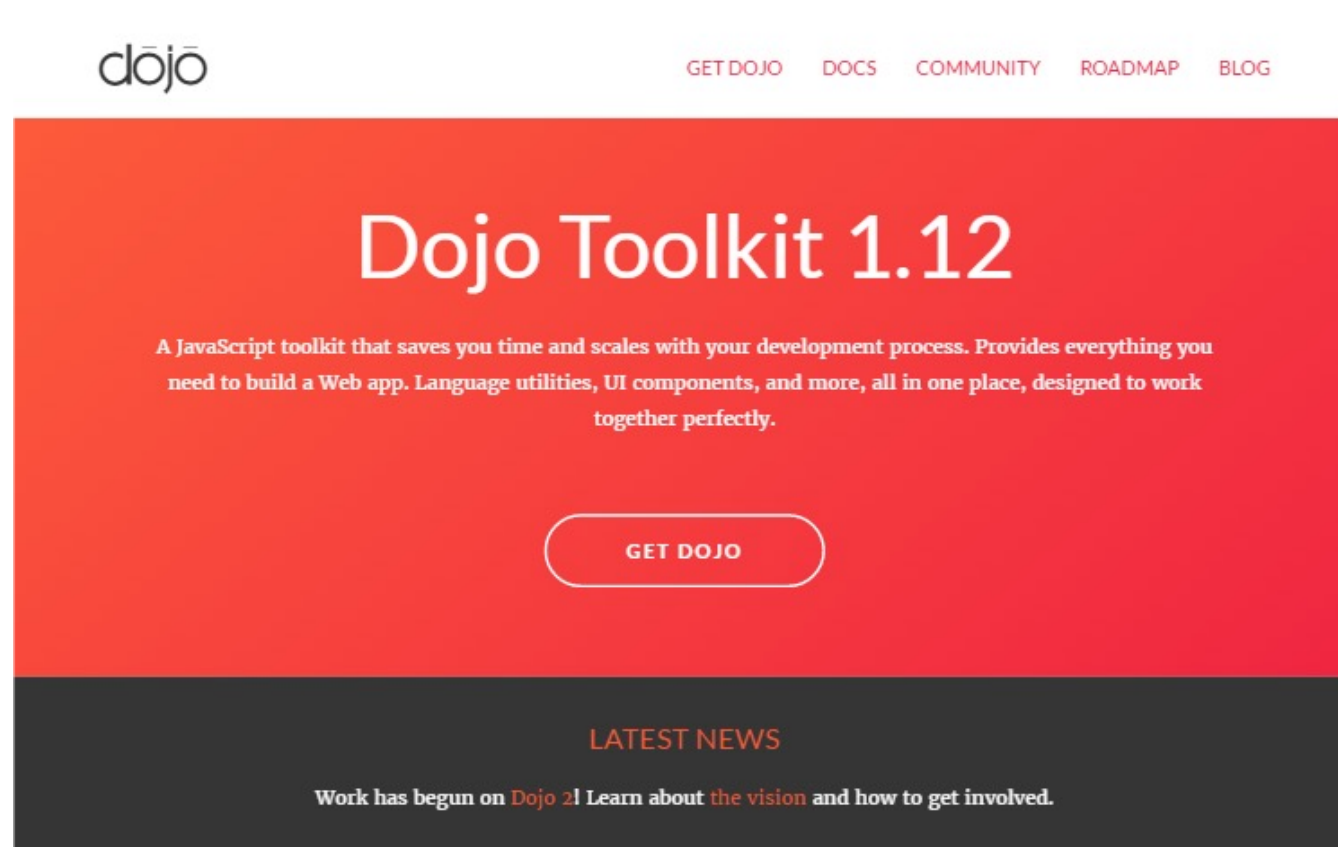


geonet.esri.com



Developer resources

- dojotoolkit.org



How to use

- Request API from server (online or local)
- Configure Dojo
- Require Widget or Class

How to use

- Request API from server (online or local)

Online (CDN)

```
<link rel="stylesheet" href="https://js.arcgis.com/3.20/esri/css/main.css">  
<script src="https://js.arcgis.com/3.20/"></script>
```

Local

```
<link rel="stylesheet" href="https://<your domain>/3.20/esri/css/main.css">  
<script src="https://<your domain>/3.20/"></script>
```

How to use

- Configure Dojo

```
<script>  
    dojoConfig = {  
        async: true,  
        parseOnLoad: true  
    }  
</script>
```

How to use

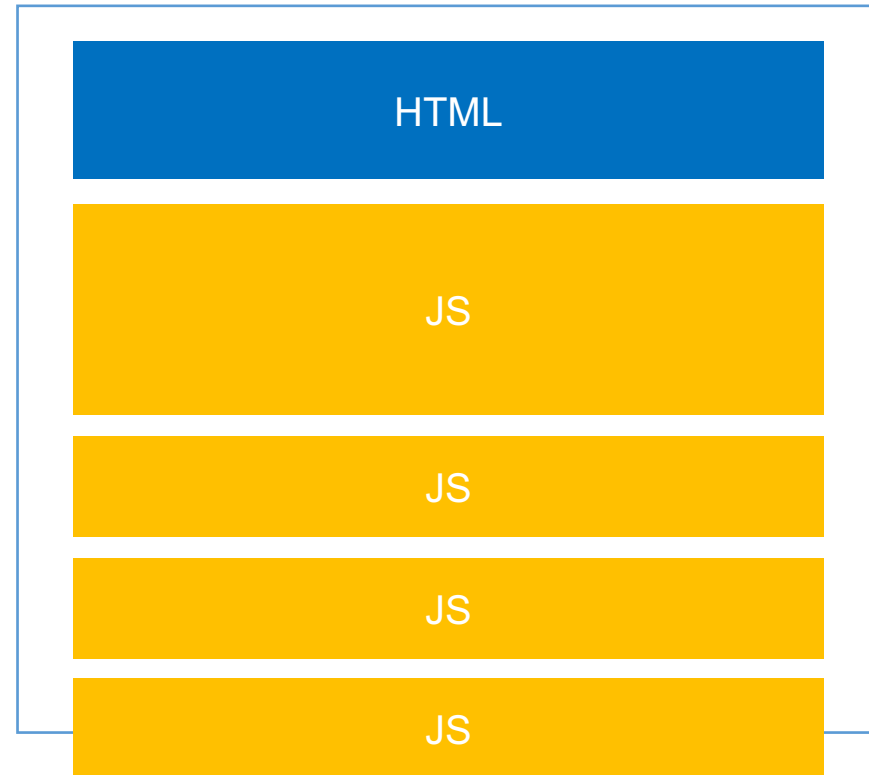
- Require Widget or Class

```
require(["dojo", "yourWidget"],  
        function (dojo, yourWidget){  
            // Put your code here.  
        }  
);
```

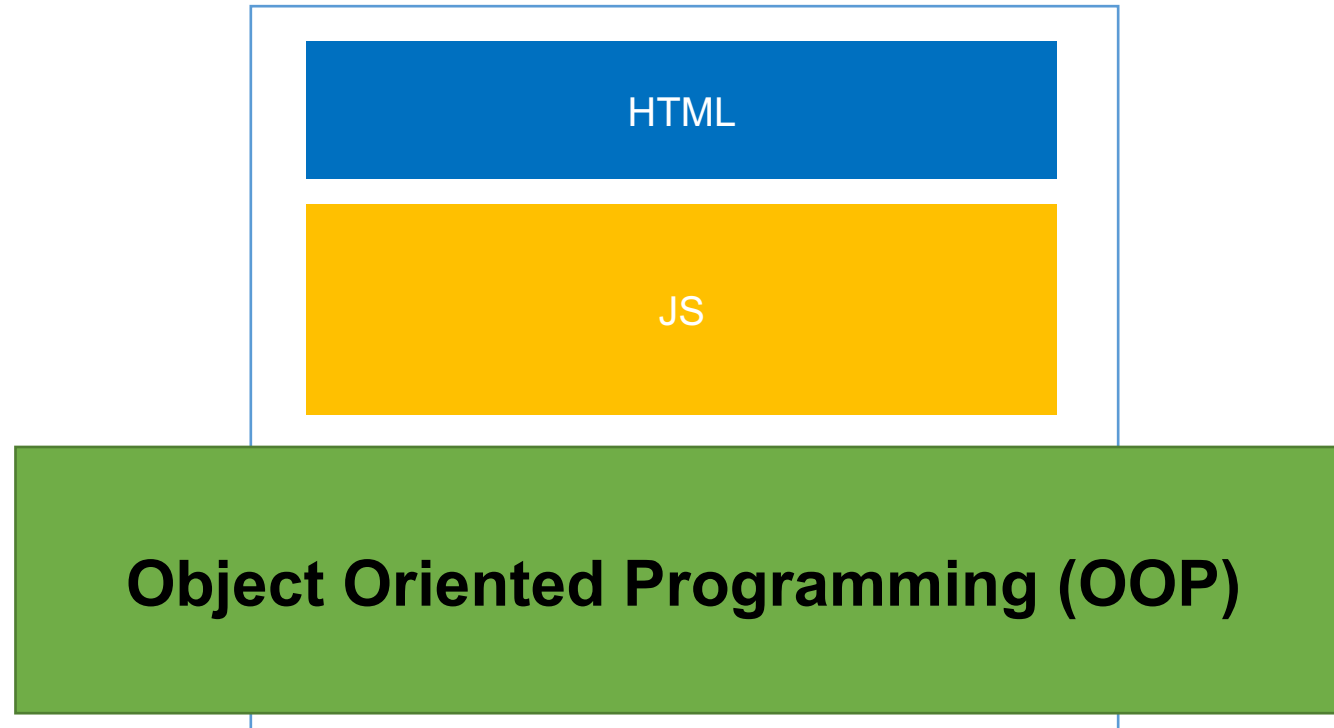


Widget

What is widget?

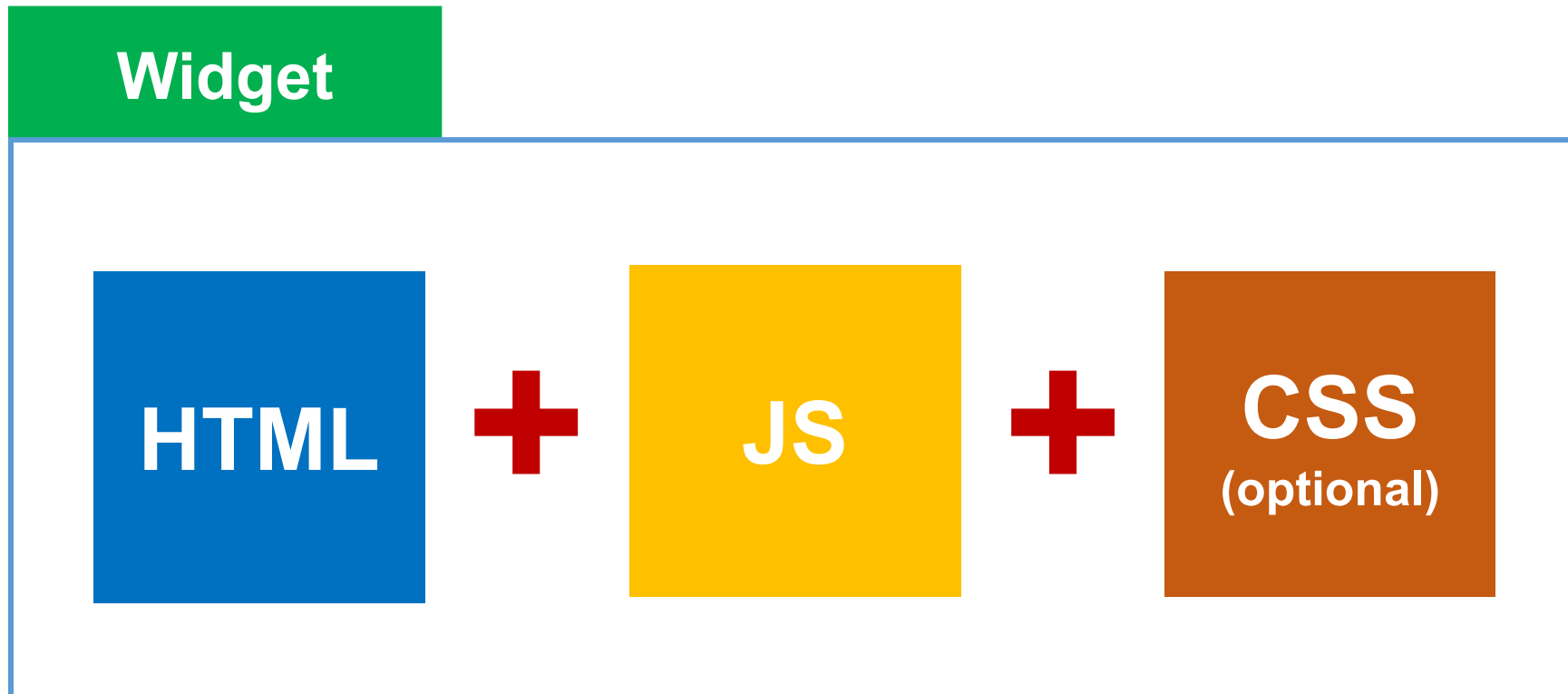


What is widget?











What is widget?

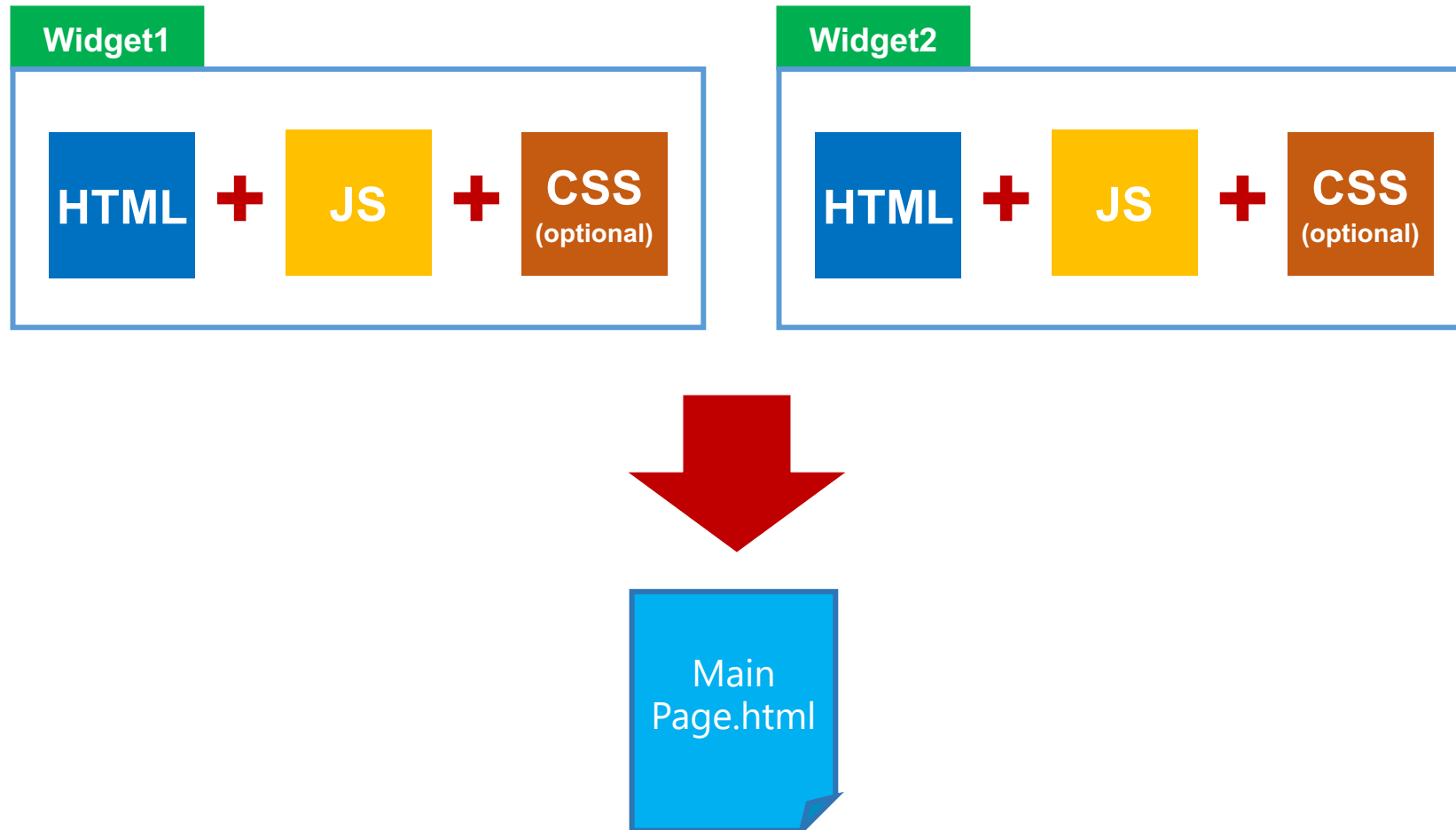
- widget is a class defined in a module that creates a UI component.



What is widget?

| | |
|--|---------------------------|
| ▼  myApp | 2 items folder |
| ▼  widget | 4 items folder |
| ▶  css | 0 items folder |
| ▼  images | 1 item folder |
|  defaultAvatar.jpg | 24.9 kB JPEG image |
| ▼  templates | 1 item folder |
|  AuthorWidget.html | 199 bytes HTML document |
|  AuthorWidget.js | 1.2 kB JavaScript program |

What is widget?



Why custom widget?

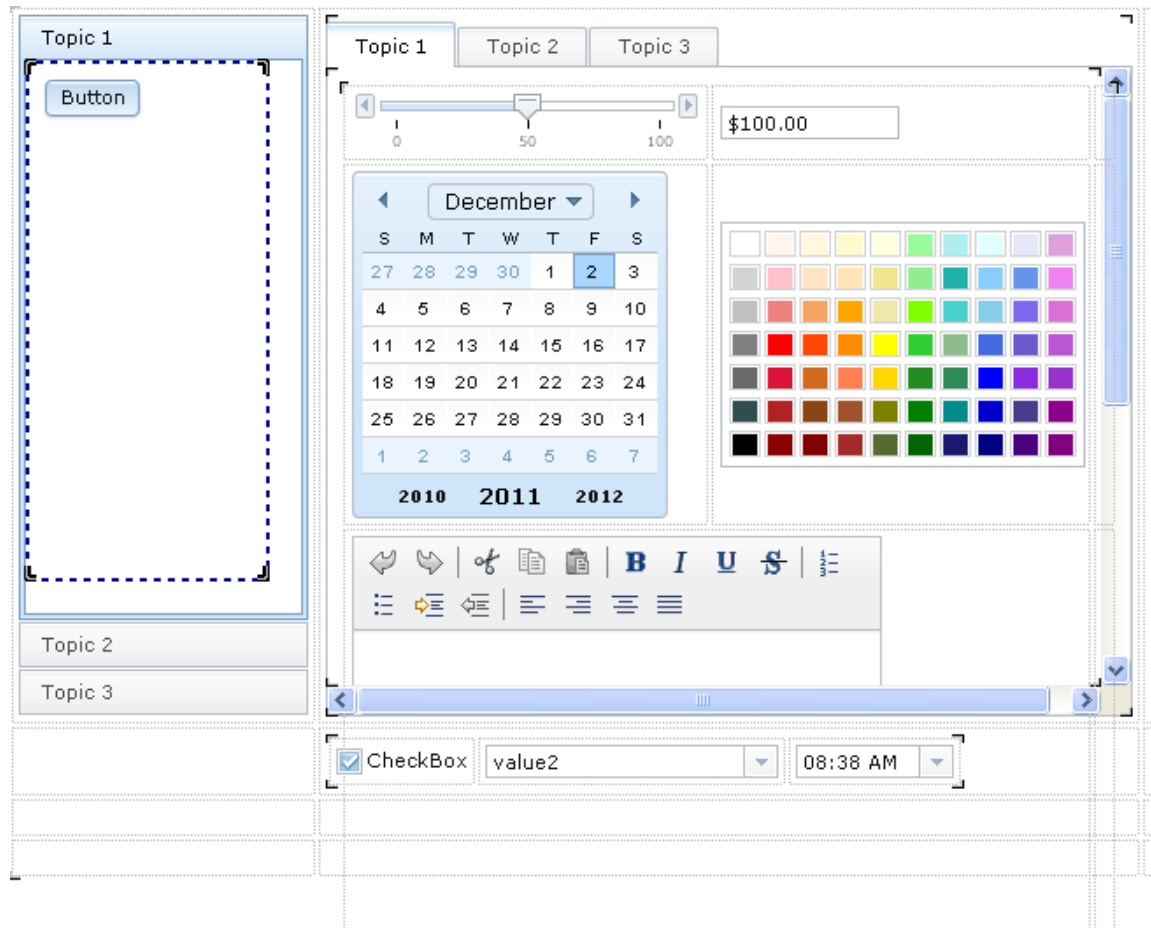
- Reusable / Sharable
- Build widgets with your custom business logic.
- Create functional forms, layouts, and much more.

Dojo Toolkit main packages

- **Dojo** - Core class
- **Dijit** - Widgets, UI, Control
- **Dojox** - Addition Content



Ready to use widget of dojo (example)



- Button
- TextBox
- Slider
- ColorPicker
- Calendar
- Etc.

An aerial view of a dense city skyline, likely Manila, Philippines, featuring numerous high-rise buildings and a prominent skyscraper on the right. The entire image is overlaid with a semi-transparent teal color.

Declarative vs Programmatic

Declarative

```
require(["dojo/parser", "dijit/form/Button"]);
```

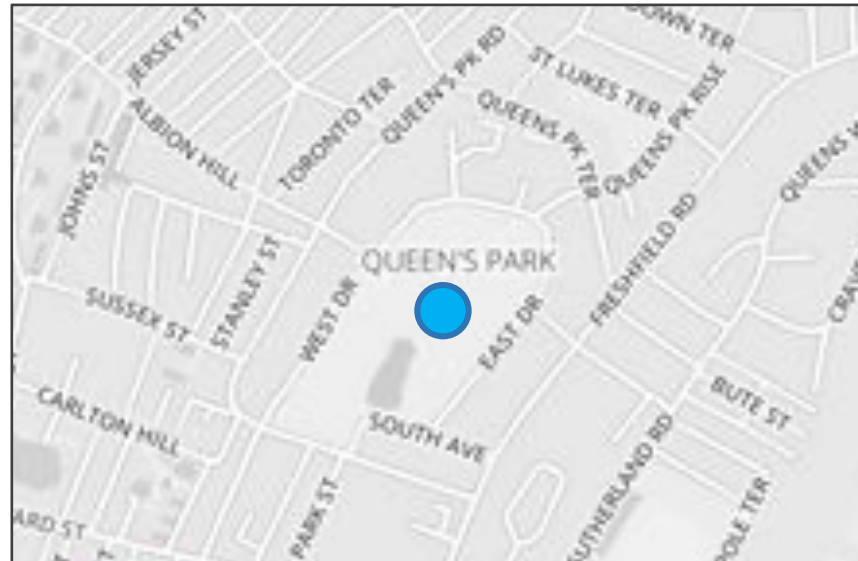
```
<button data-dojo-type="dijit/form/Button" type="button">Click me too!  
  <script type="dojo/on" data-dojo-event="click" data-dojo-args="evt">  
    require(["dojo/dom"], function(dom){  
      dom.byId("result2").innerHTML += "Thank you! ";  
    });  
  </script>  
</button>  
<div id="result2"></div>
```

Declarative

- When we choose

X:

Y:



Programmatic

JavaScript Code

```
require(["dijit/form/Button", "dojo/dom", "dojo/domReady!"], function(Button, dom){  
  // Create a button programmatically:  
  var myButton = new Button({  
    label: "Click me!",  
    onClick: function(){  
      // Do something:  
      dom.byId("result1").innerHTML += "Thank you! ";  
    }  
  }, "progButtonNode").startup();  
});
```

HTML Code

```
<button id="progButtonNode" type="button"></button>  
<div id="result1"></div>
```

Programmatic

- When we choose

X1:

Y1:

X2:

Y2:

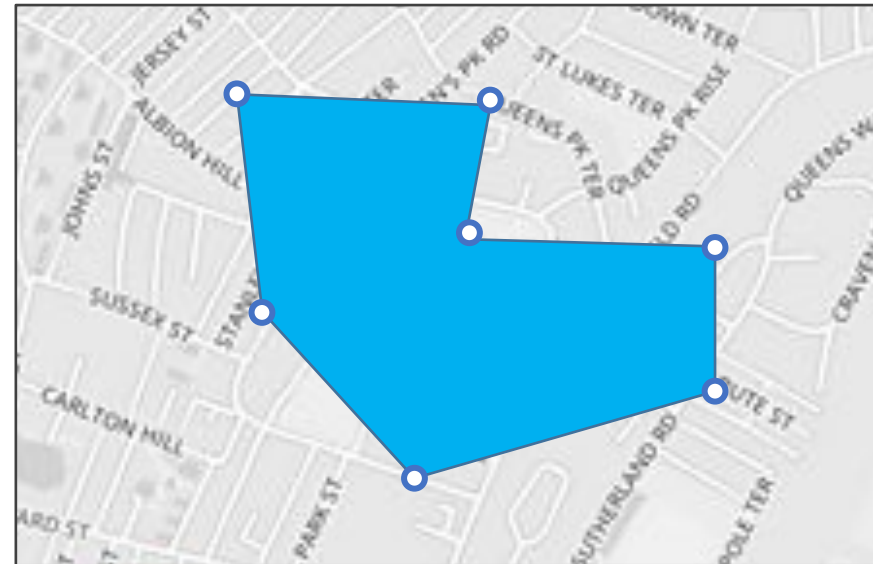
X3:

Y3:

·

·

·



An aerial view of a city skyline, likely Manila, Philippines, featuring numerous high-rise buildings and a dense urban landscape. The entire image is overlaid with a semi-transparent teal color. The text "Create a widget" is centered in white, sans-serif font.

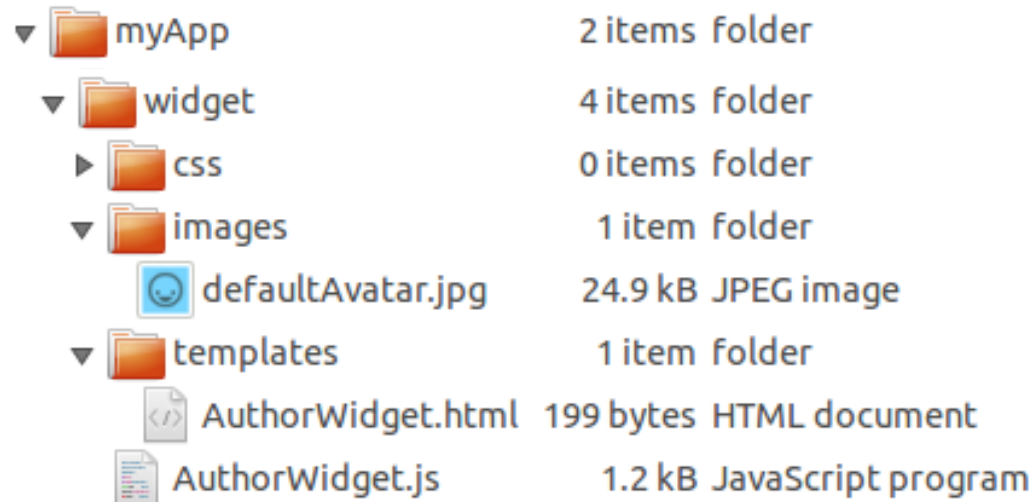
Create a widget

Step to create widget

- Create files structure for custom widget
- Define widget template (user interface)
- Write your code

Step to create widget

- Create files structure for custom widget



Module ID: `“myApp/widget/AuthorWidget”`

Step to create widget

- Define widget template (user interface)

```
<div>
```

```
  <h3 data-dojo-attach-point="nameNode">${name}</h3>
```

```
  <img class="${baseClass}Avatar" src="" data-dojo-attach-point="avatarNode
```

```
  <p data-dojo-attach-point="bioNode">${!bio}</p>
```

```
</div>
```

HTML Code

JavaScript Code

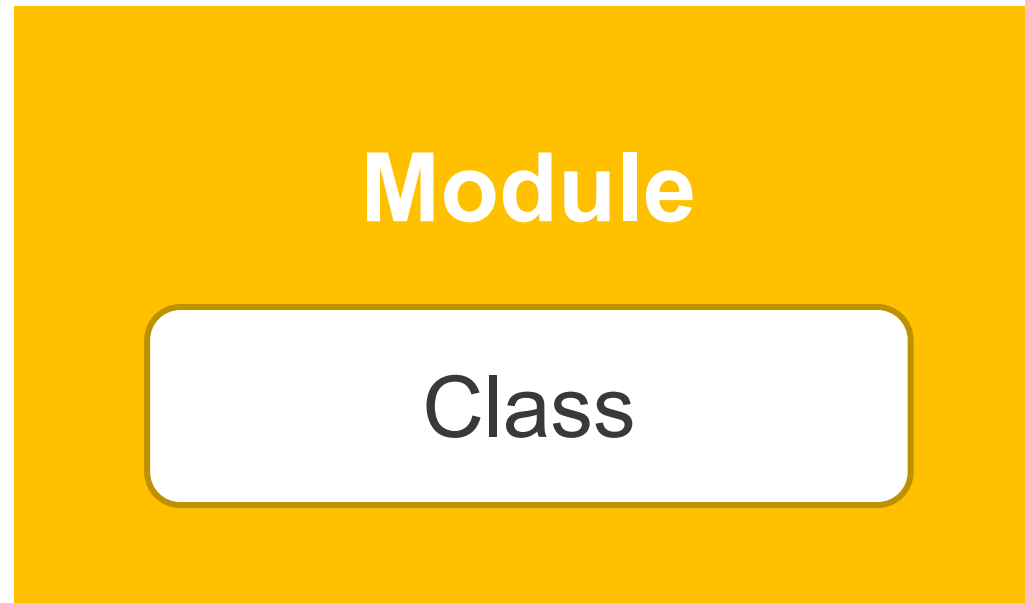
```
postCreate: function () {  
    var favoriteNode = this.bioNode;  
}
```

Step to create widget

- Write your code
 - Create module
 - Declare class
 - Implement functionality

Step to create widget

- Write your code



Class must live in module

Step to create widget

- Write your code
 - Create module

```
define([  
    "dojo/_base/declare",  
    "dojo/dom",  
    "app/dateFormatter"  
], function (declare, dom, dateFormatter) {  
    return declare (null, {  
        showDate: function (id, date) {  
            // Do something  
        }  
    });  
});  
});
```

Module load other module

Step to create widget

- Write your code
 - Declare class

Class name

Superclass

```
var MyClass = declare("mynamespace.MyClass", MyParentClass, {
    myProperty: 12,
    myMethod: function () {
        // Perform any functionality here
    }
});
```

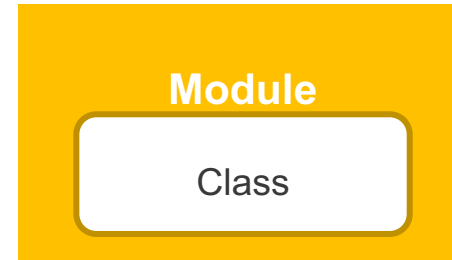
Step to create widget

- Write your code
 - Declare class

```
1 define([
2   "dojo/_base/declare",
3   "dijit/_WidgetBase",
4   "dijit/_TemplatedMixin",
5   "dijit/_WidgetsInTemplateMixin",
6 ], function (declare, _WidgetBase,
7   _TemplatedMixin, _WidgetsInTemplateMixin) {
8
9   return declare([
10    _WidgetBase,
11    _TemplatedMixin,
12    _WidgetsInTemplateMixin], {
13
14    constructor: function (param) {
15    },
16
17    postCreate: function () {
18      this.inherited(arguments);
19    },
20
21    startup: function () {
22      this.inherited(arguments);
23    }
24  });
25
26 });
```

Class

Module



Step to create widget

- Write your code
 - Declare class

```
1  define([
2    "dojo/_base/declare",
3    "dijit/WidgetBase",
4    "dijit/_TemplatedMixin",
5    "dijit/_WidgetsInTemplateMixin",
6  ], function (declare, _WidgetBase,
7    _TemplatedMixin, _WidgetsInTemplateMixin) {
8  }
```

Widget lifecycle (minimum req's)

```
var wgDeclare = declare([], {  
  
    /* Start: Basic Properties */  
    widgetsInTemplate: true,  
    templateString: template,  
    baseClass: "myTemplate",  
    /* End: Basic Properties */  
  
    constructor: function () {  
  
    },  
    postCreate: function () {  
        this.inherited(arguments);  
    },  
    startup: function () {  
        this.inherited(arguments);  
    }  
});  
return wgDeclare;
```

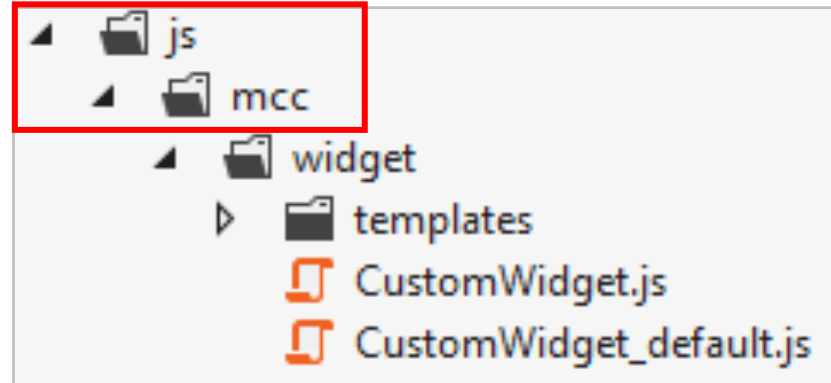
An aerial view of a city skyline, likely San Francisco, with a prominent skyscraper on the right. The image is overlaid with a semi-transparent green filter. The text "Use widget in your application" is centered in white.

Use widget in your
application

Step to use your widget

- Configure Dojo
- Require and instant widget

Configure Dojo

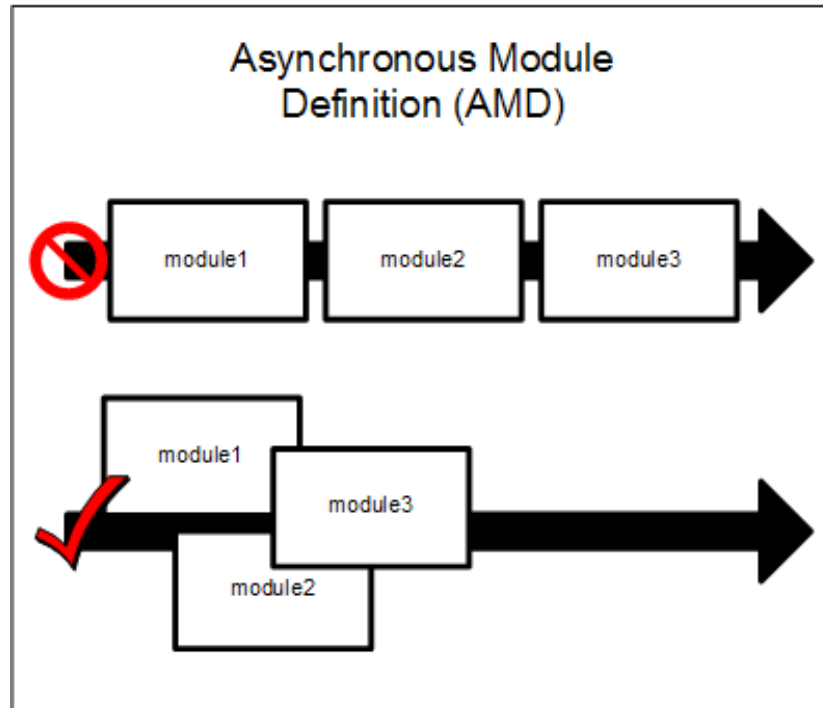


```
dojoConfig = {  
  parseOnLoad: true,  
  isDebug: false,  
  packages: [{  
    name: 'mcc',  
    location: baseLocation + '/js/mcc'  
  }],  
  async: true,  
};
```

“mcc/widget/CustomWidget”

Configure Dojo

```
dojoConfig = {  
  parseOnLoad: true,  
  isDebug: false,  
  packages: [{  
    name: 'mcc',  
    location: baseLocation + '/js/mcc'  
  }],  
  async: true,  
};
```



Require and instant widget

```
<script type="text/javascript">  
    require(["dojo/ready", "mcc/widget/CustomWidget_default"],  
        function (ready, CustomWidget_default) {  
            ready(function () {  
                var customWidget = new CustomWidget_default({}, "myWid  
                customWidget.startup();  
            });  
        });  
</script>
```

An aerial view of a city skyline, likely Manila, Philippines, featuring numerous high-rise buildings and a dense urban landscape. The entire image is overlaid with a semi-transparent teal color. The text 'Useful modules' is centered in a white, sans-serif font.

Useful modules

“dojo/on”

- Use for event handler

```
postCreate: function () {
    this.inherited(arguments);
    on(map, "click", lang.hitch(this, "onMapClick"));
},

onMapClick: function (evt) {
    // Do something.
}
```

"dojo/_base/lang"

- **lang.hitch()** Use to provide proper context for various methods

```
postCreate: function () {
    this.inherited(arguments);
    on(map, "click", lang.hitch(this, "onMapClick"));
},

onMapClick: function (evt) {
    // Do something.
}
```

“dojo/Evented”

- Inherit from “**dojo/Evented** widget emit events to other widgets.

```
onSpatialSuccess: function (response) {  
    this.emit("spatial-success", response.records);  
}
```

Widget
t1

```
this.emit("spatial-success", response.records);
```



Widget
t2

```
on(Widget1, "spatial-success", lang.hitch(this, "doSomething"));
```

An aerial view of a city skyline, likely Manila, Philippines, featuring numerous high-rise buildings and a dense urban layout. The entire image is overlaid with a semi-transparent teal color. The text "Get Started" is centered in a large, white, sans-serif font.

Get Started

Hello Dojo

Getting started with Dojo is as simple as including the dojo.js script in a web page, just like any other JavaScript file. Dojo is available on popular [CDNs](#), so to get started enter the following in a file named hellodojo.html and open it in your web browser.

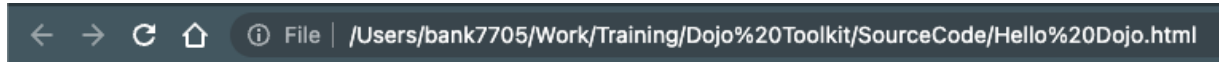
```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Tutorial: Hello Dojo!</title>
</head>
<body>
<h1 id="greeting">Hello</h1>
<!-- load Dojo -->
<script src="//ajax.googleapis.com/ajax/libs/dojo/1.10.4/dojo/dojo.js"
data-dojo-config="async: true"></script>
</body>
</html>
```

Get started



Hello Dojo

Result



Hello

Your first dojo button

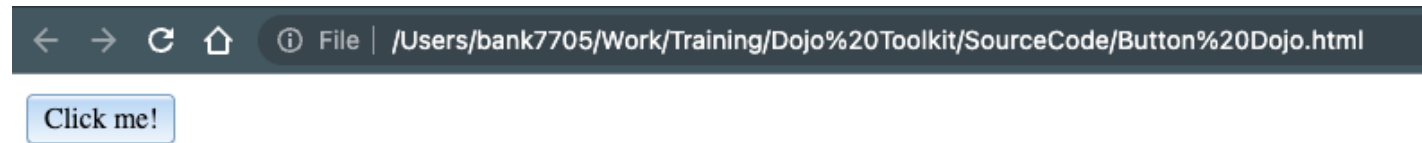
```
<body class="claro">
<script>
  require(["dojo/ready", "dijit/form/Button", "dojo/dom"], function (ready, Button, dom) {
    ready(function () {
      // Create a button programmatically:
      var myButton = new Button({
        label: "Click me!",
        onClick: function () {
          // Do something:
          dom.byId("result1").innerHTML += "Thank you! ";
        }
      }, "progButtonNode");
    });
  });
</script>

<button id="progButtonNode" type="button"></button>
<div id="result1"></div>
</body>
```

Your first dojo button

Result

Button with claro style



The background of the slide is an aerial photograph of a city skyline, likely San Francisco, with a semi-transparent teal overlay. The text is centered in white.

Dojo with ArcGIS API for JS

Dojo + ArcGIS API for JavaScript

```
<meta name="viewport" content="initial-scale=1, maximum-scale=1,user-scalable=no" />
<title>Simple Map</title>
<link rel="stylesheet" href="https://js.arcgis.com/3.41/esri/css/esri.css">
<link rel="stylesheet" href="https://js.arcgis.com/3.41/dijit/themes/claro/claro.css" media="screen">
<style>
  html,
  body,
  #map {
    height: 100%;
    margin: 0;
    padding: 0;
  }
</style>
<script src="https://js.arcgis.com/3.41/"></script>
```

```
<script>
var map;

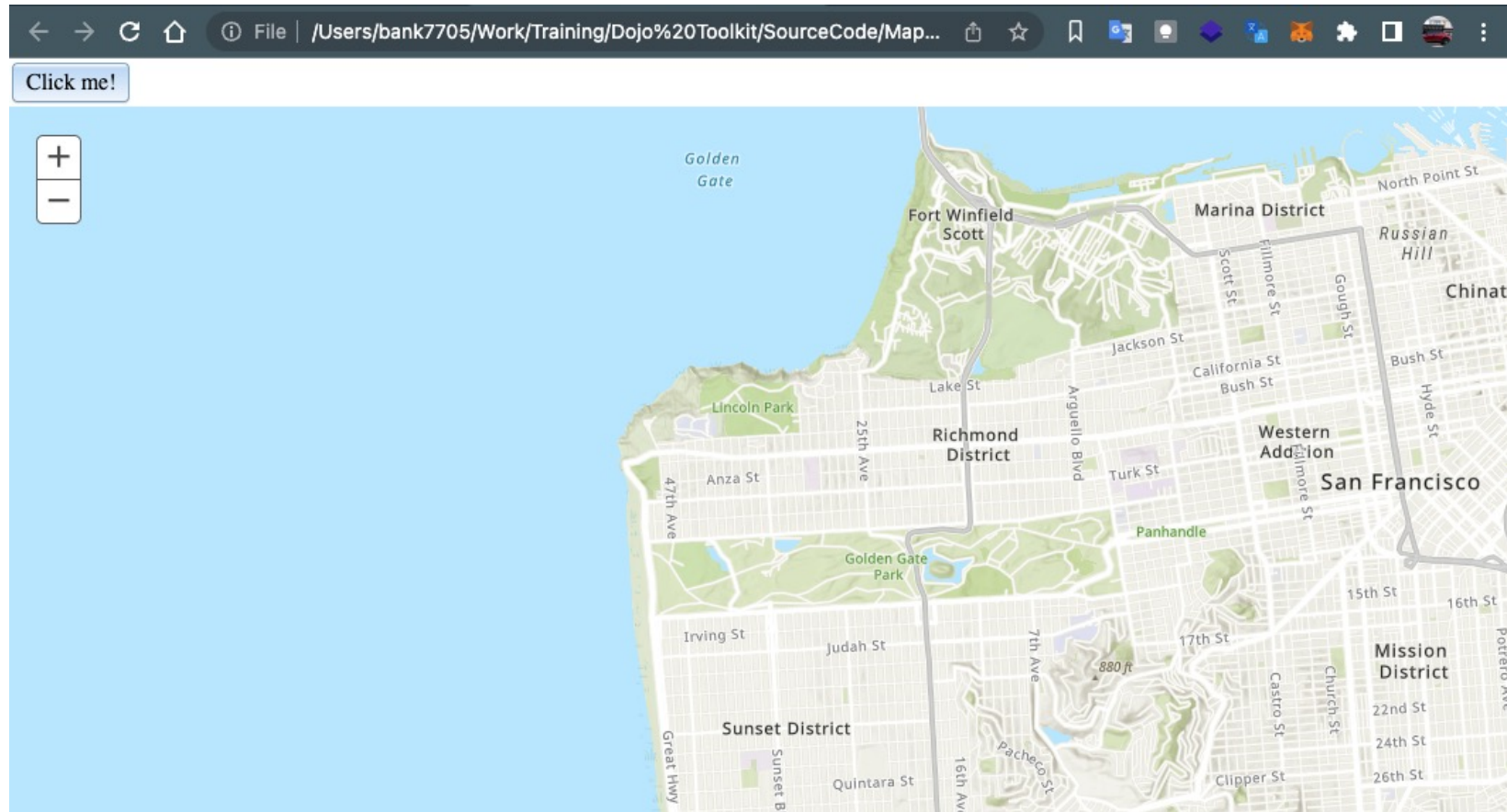
require(["esri/map", "dijit/form/Button", "dojo/domReady!"], function (Map, Button) {
  map = new Map("map", {
    basemap: "topo-vector", //For full list of pre-defined basemaps, navigate to http://arcgis.com/arcgis/1JVo6Wd
    center: [-122.45, 37.75], // longitude, latitude
    zoom: 13
  });

  var myButton = new Button({
    label: "Click me!",
    onClick: function () {
      // Do something:
      dom.byId("result1").innerHTML += "Thank you! ";
    }
  }, "progButtonNode");
});
</script>
```

```
<body class="claro">
  <button id="progButtonNode" type="button"></button>
  <div id="result1"></div>
  <div id="map"></div>
</body>
```

Dojo + ArcGIS API for JavaScript

Result



A teal-tinted photograph of a dense city skyline, likely Manila, Philippines, featuring numerous high-rise buildings and a prominent skyscraper on the right. The word "END" is overlaid in large, white, sans-serif capital letters in the center of the image.

END