



Introduction to JS Object

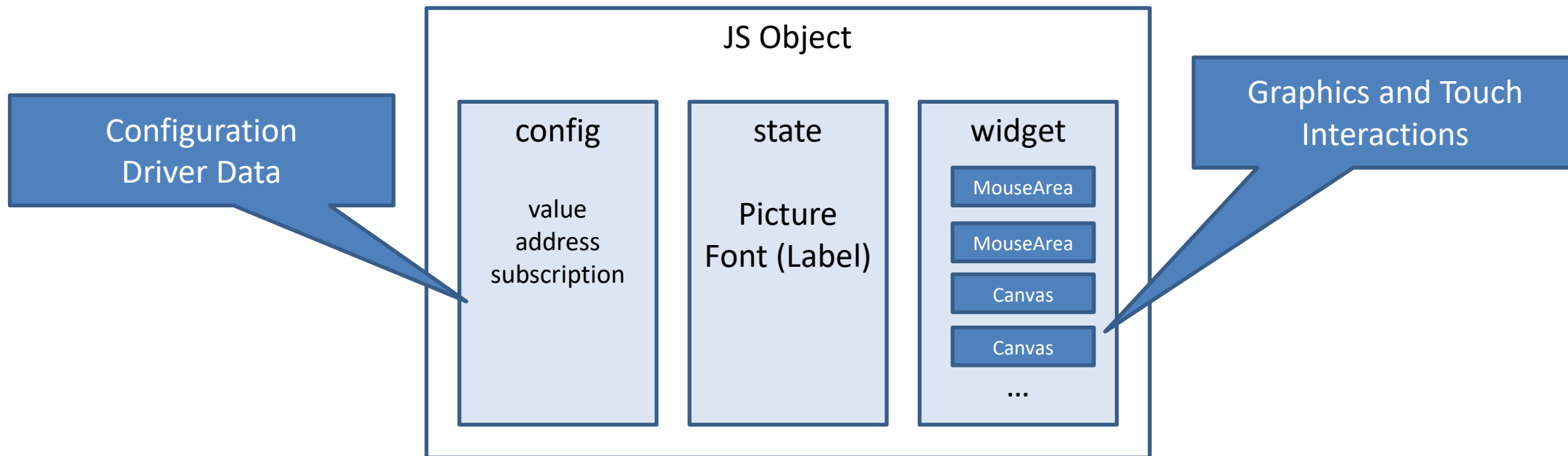
Mao Hsu
Dec 9, 2020



[https://commons.wikimedia.org/wiki/File:A_tailor_fitting_a_customer_\(Belgrade,_1937\).jpg](https://commons.wikimedia.org/wiki/File:A_tailor_fitting_a_customer_(Belgrade,_1937).jpg)

JS Object

A New Customizable Object



Outline

- JavaScript Crash Course
- JS Object Demos
- JS Object Basics

- Online JS Object SDK Manual:
https://dl.weintek.com/public/Document/JS_Object_SDK/Current/index.html



JS

JavaScript Crash Course

JavaScript Crash Course

- JavaScript is not Java
- [What is JavaScript?](#)
- Who uses JavaScript? (or ECMAScript dialects)
 - Web browsers
 - And browser extensions you installed
 - Server applications (Node.js)
 - OpenOffice scripts/macros

JavaScript Crash Course

- Data types

- Boolean: `true` or `false`

```
var realWorld = false;  
var matrix = true;
```

- Number

```
var i = 0;  
var x = 2**10; // 1024; 2 to the power of 10
```

- String

```
var a = ""; // The empty string: it has zero characters  
var b = 'There is no spoon.';  
var c = "Only human.";
```

- Arrays ([MDN](#))

```
var e = ["the", 1, true]; // 3 elements of different types
```

JavaScript Crash Course

Data and Structure Types

JavaScript		EBPro Macro
Boolean	bool
Number	(unsigned) short/int
	float/double
String	char (or char array)
undefined		
null		
Symbol		
Object		
function		

JavaScript Crash Course

- Declare a variable and print it

```
var greeting = "Please stand by. The webinar will begin shortly.";
console.log(greeting);
// Print the greeting to JavaScript console (JS in cMT Diagnoser)
```

- Declare a function

```
function factorial(n) {
    if (n === 0)
        return 1; // 0! = 1

    return n * factorial(n - 1);
}

console.log(factorial(3)); // returns 6
```

JavaScript Crash Course

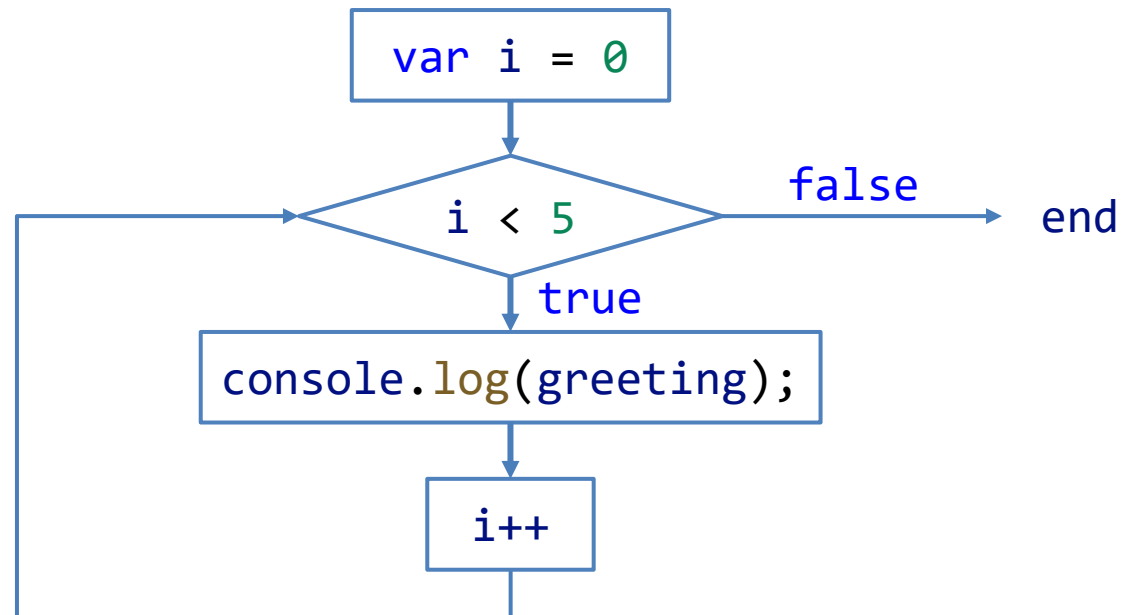
- Print to console

```
var greeting = "Please stand by. The webinar will begin shortly.";  
console.log(greeting);  
// Print the greeting to JavaScript console (JS in cMT Diagnoser)
```

JavaScript Crash Course

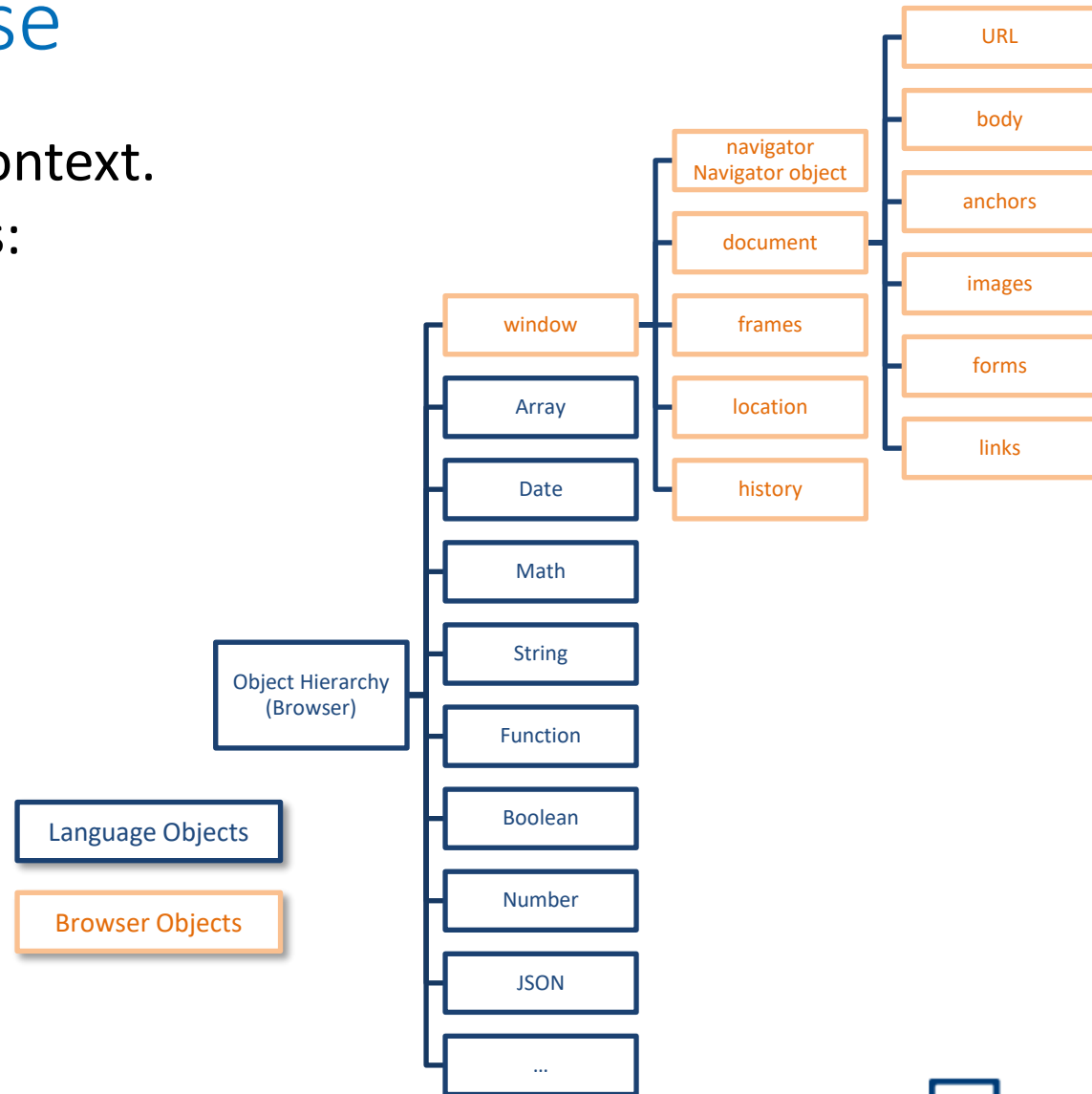
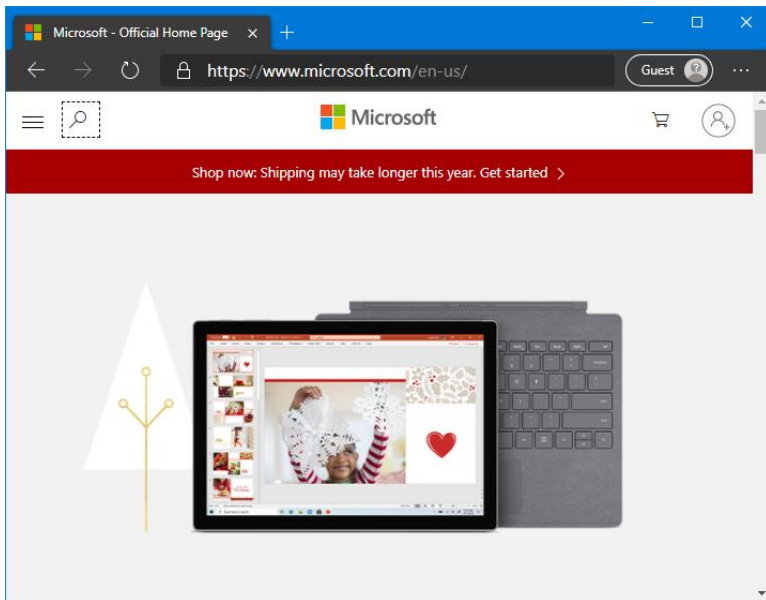
- For-loops

```
var greeting = "Please stand by. The webinar will begin shortly.\n";  
for (var i = 0; i < 5; i++) {  
  console.log(greeting);  
}  
// Print 5 greetings (i=0,1,2,3,4)
```



JavaScript Crash Course

- JavaScript has an execution context. For example, in web browsers:



JavaScript Crash Course

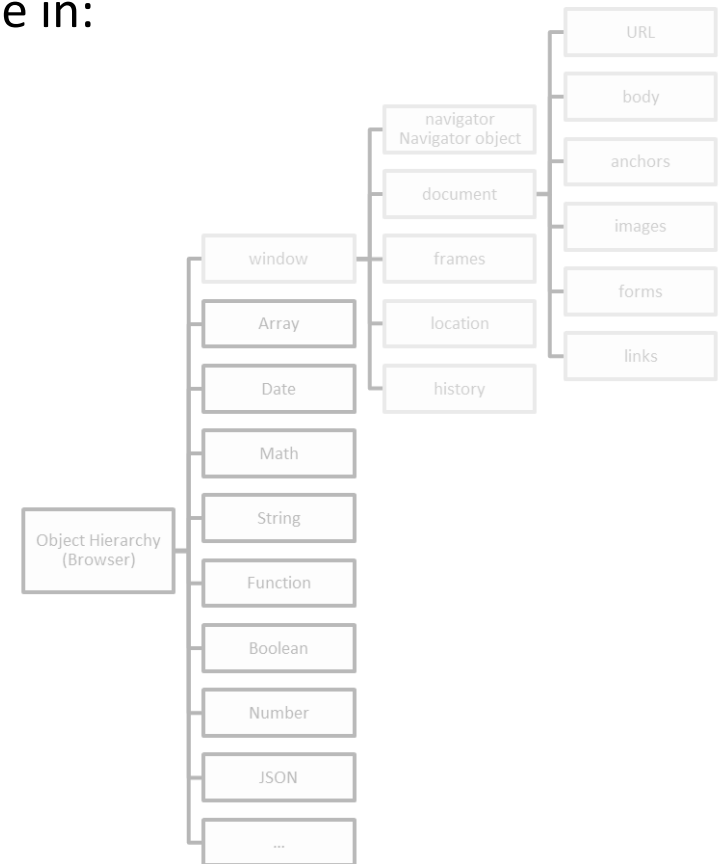
- Browser scripts sometimes do not work on server, or JS object
 - For example, in browser, click F12 to open Web Console and type in:

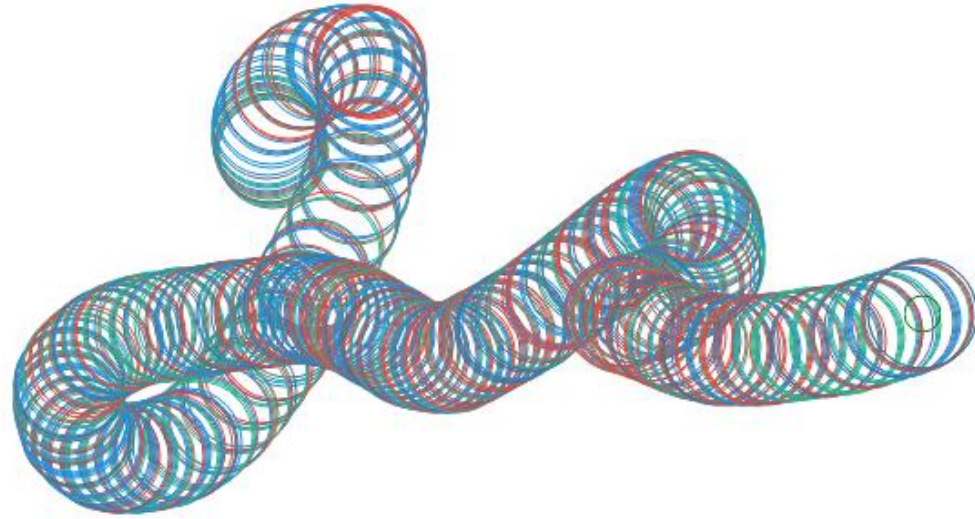
```
// Go to example.com  
window.location = "https://example.com";
```

- In Node.js (if you have it), it will show:

```
ubuntu@MAO-LAPTOP:~$ node  
Welcome to Node.js v14.6.0.  
Type ".help" for more information.  
> window.location = "https://exmaple.com"  
Uncaught ReferenceError: window is not defined  
>
```

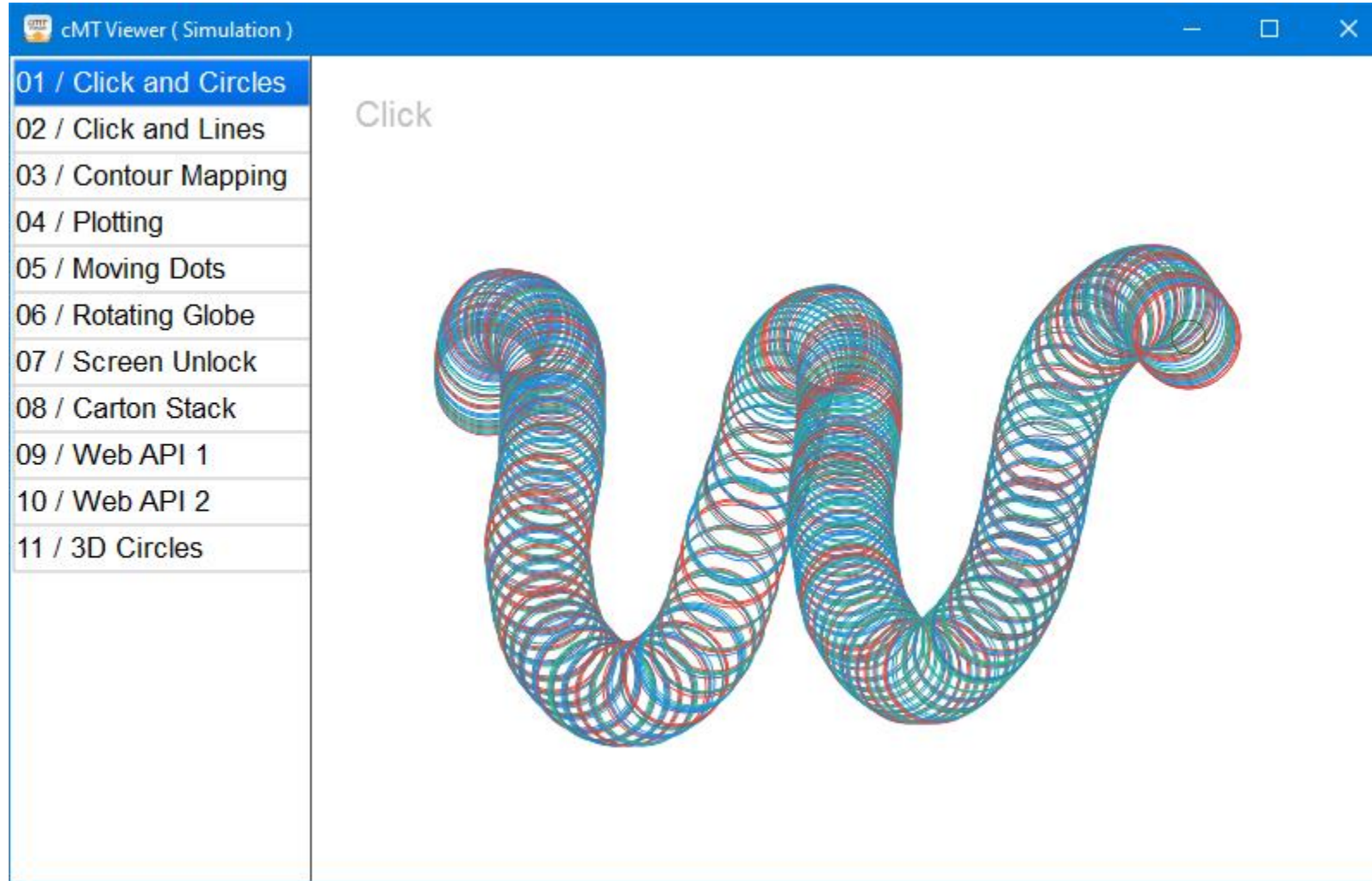
- For JS Object:
We have a window object but not the same one.
Nothing will happen (it's not a browser).



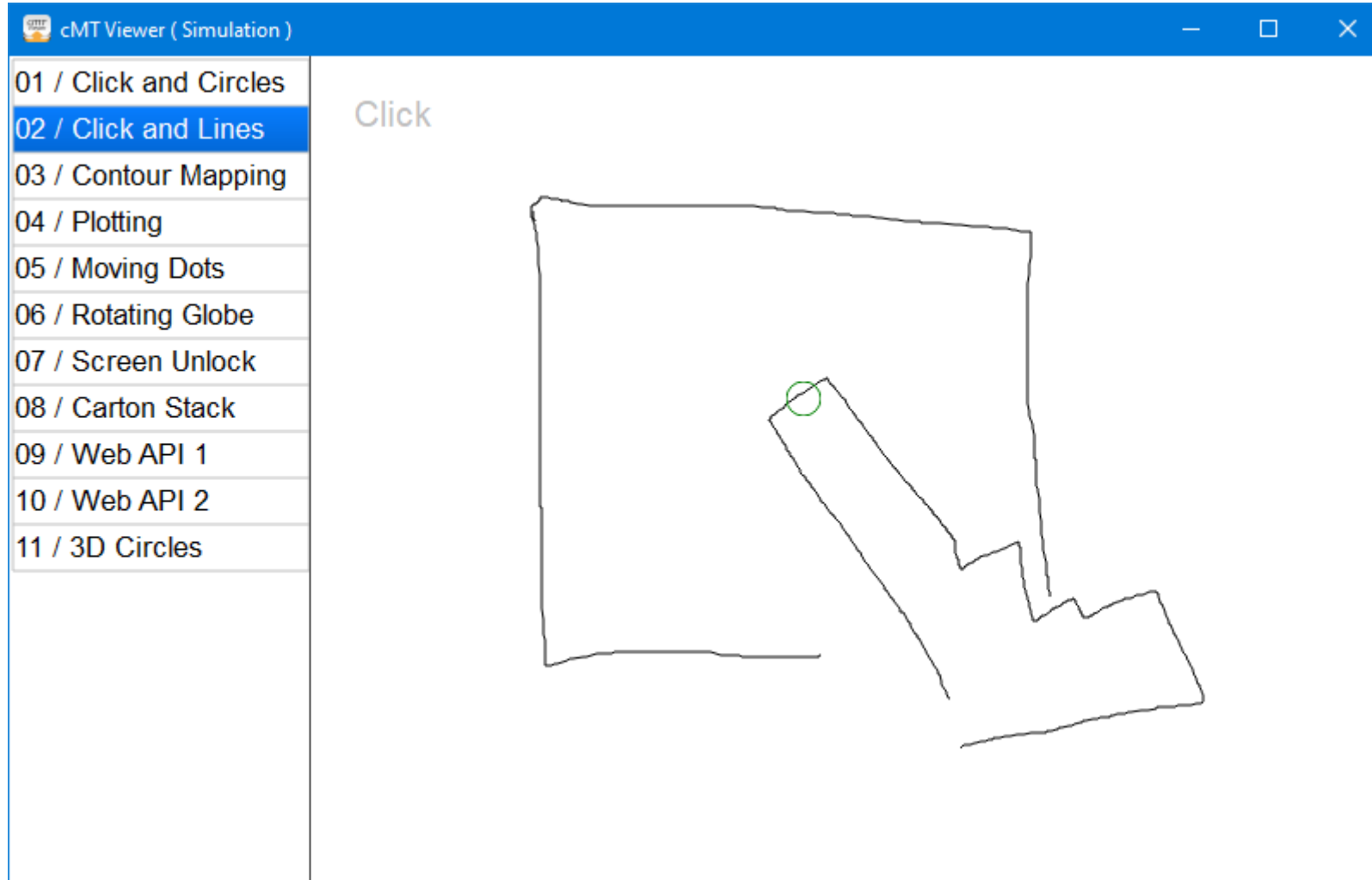


JS Object Demos

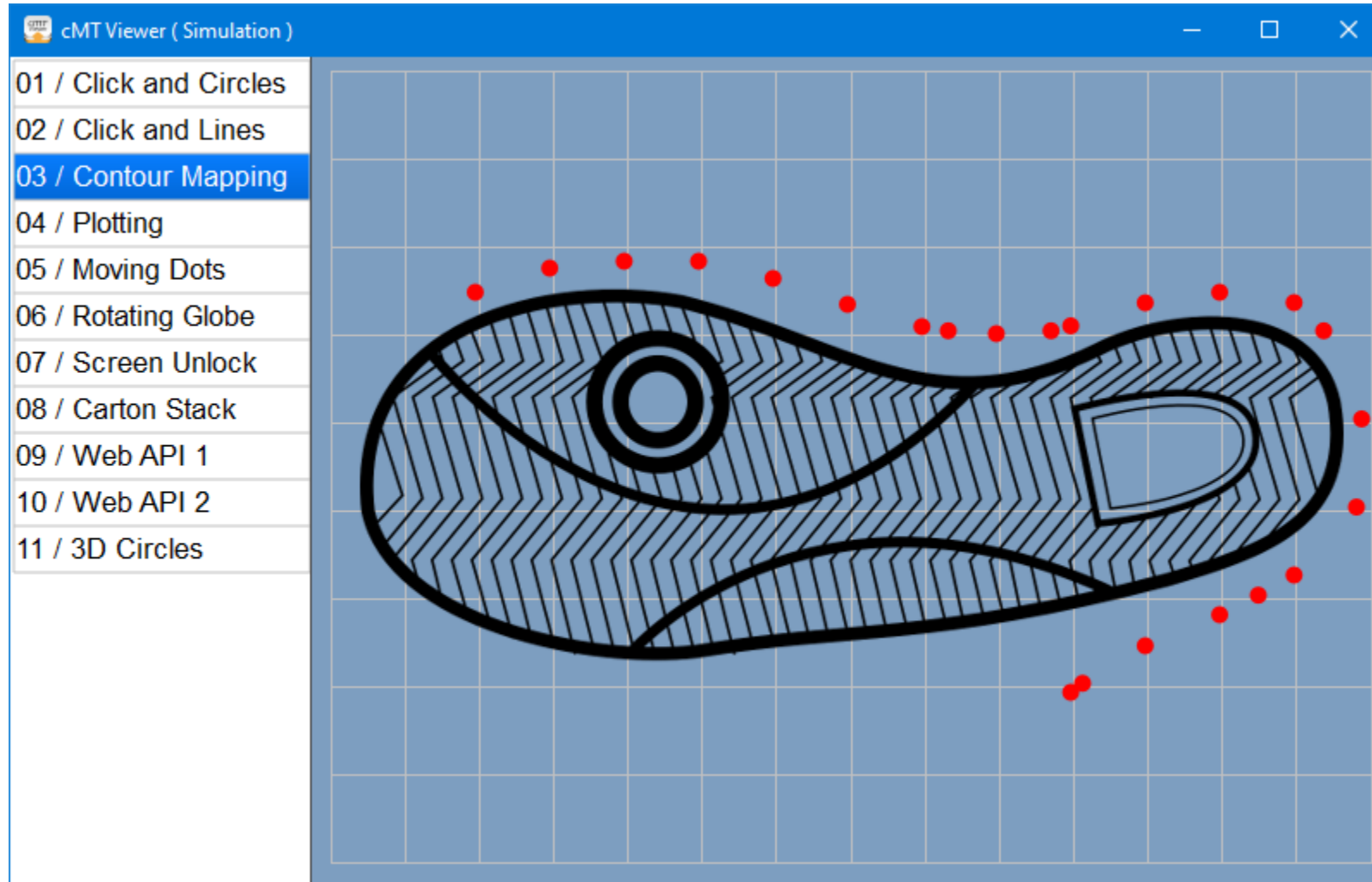
01 / Click and Circles



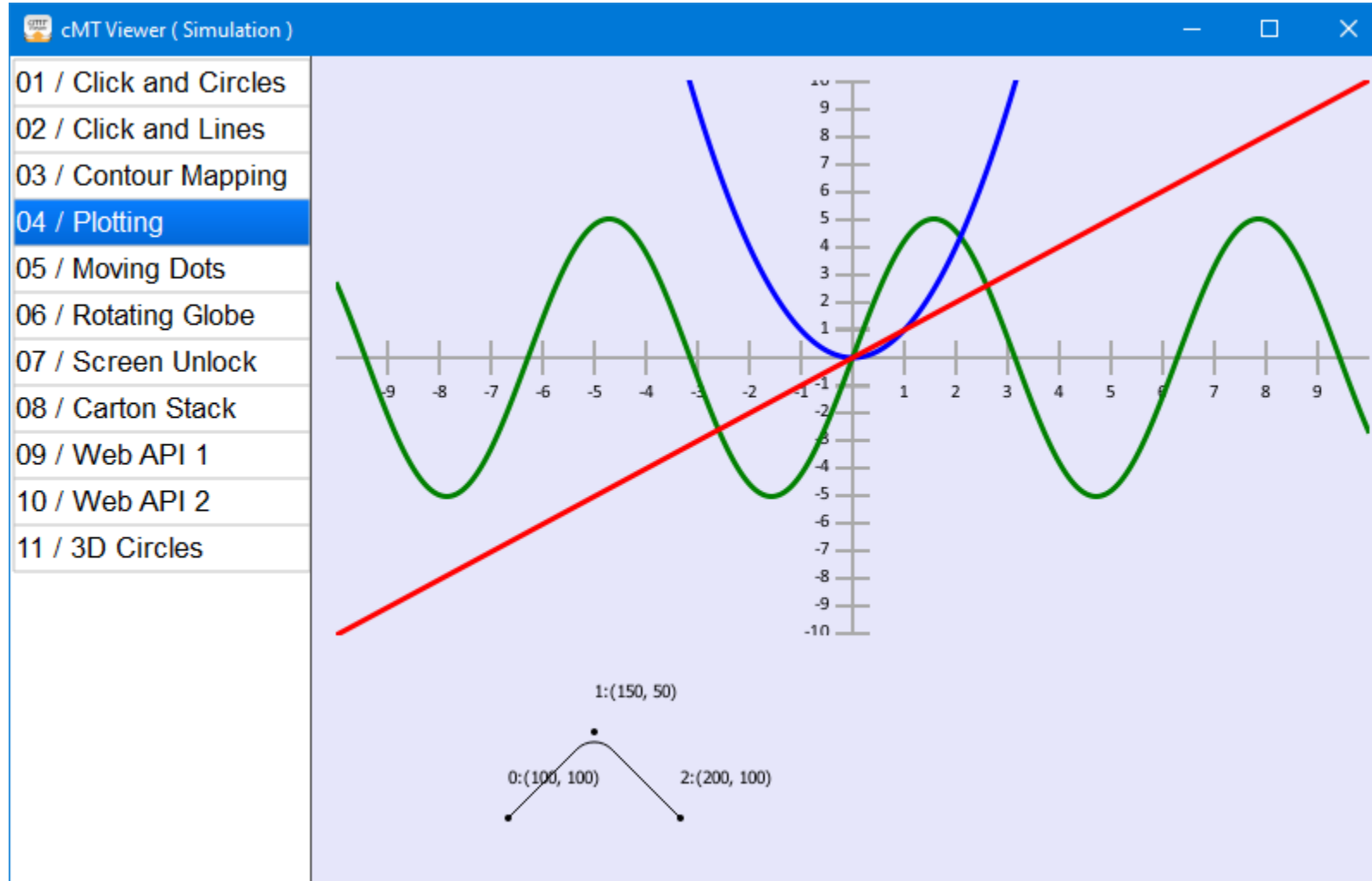
02 / Click and Lines



03 / Contour Mapping



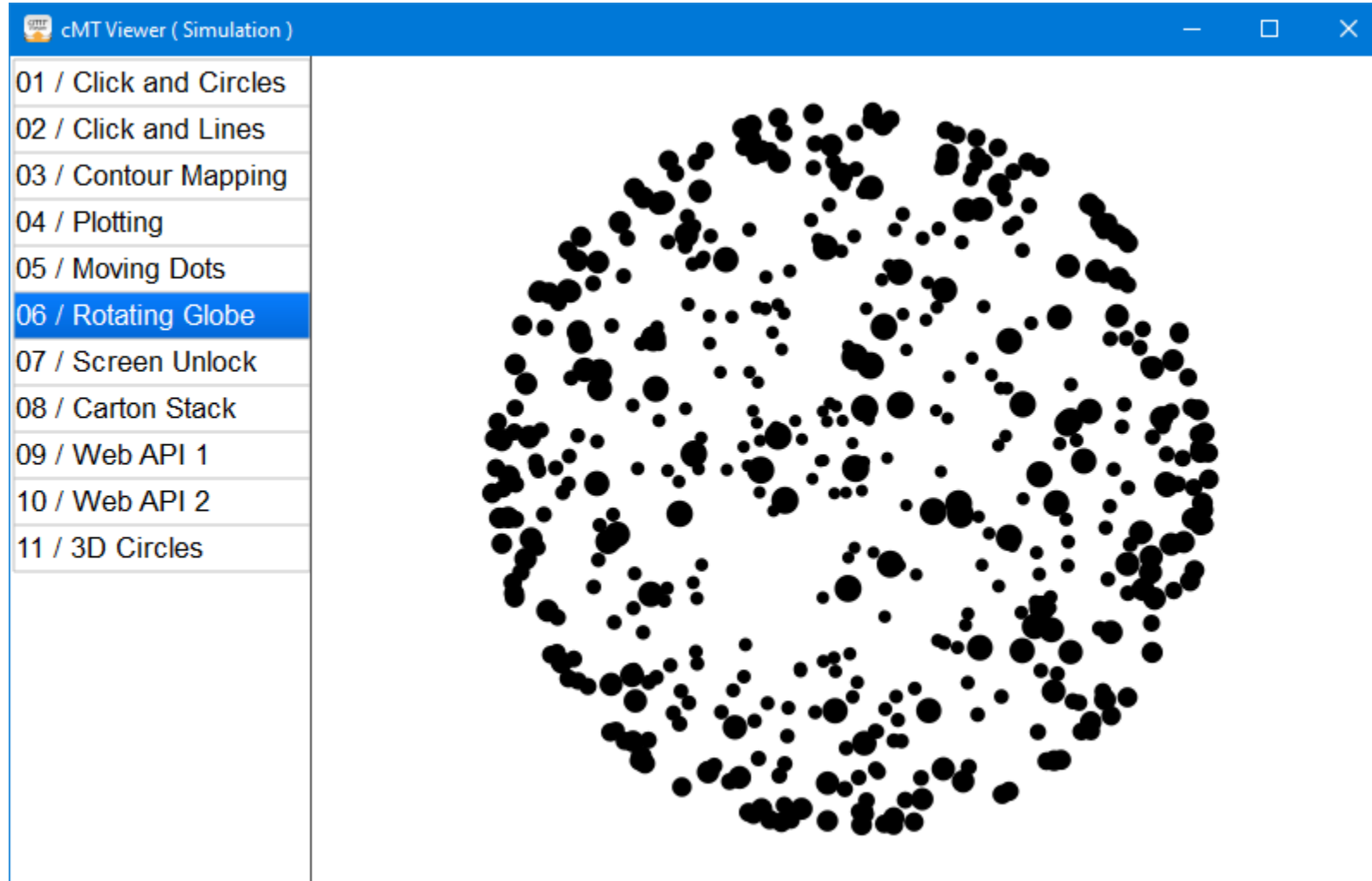
04 / Plotting



05 / Moving Dots



06 / Rotating Globe





07 / Screen Unlock

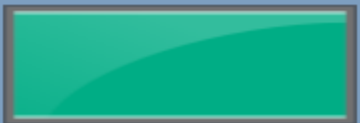
cMT Viewer (Simulation)

01 / Click and Circles
02 / Click and Lines
03 / Contour Mapping
04 / Plotting
05 / Moving Dots
06 / Rotating Globe
07 / Screen Unlock
08 / Carton Stack
09 / Web API 1
10 / Web API 2
11 / 3D Circles

Unlock or trigger by drawing a specific pattern:



Pattern



>>

Saved Pattern

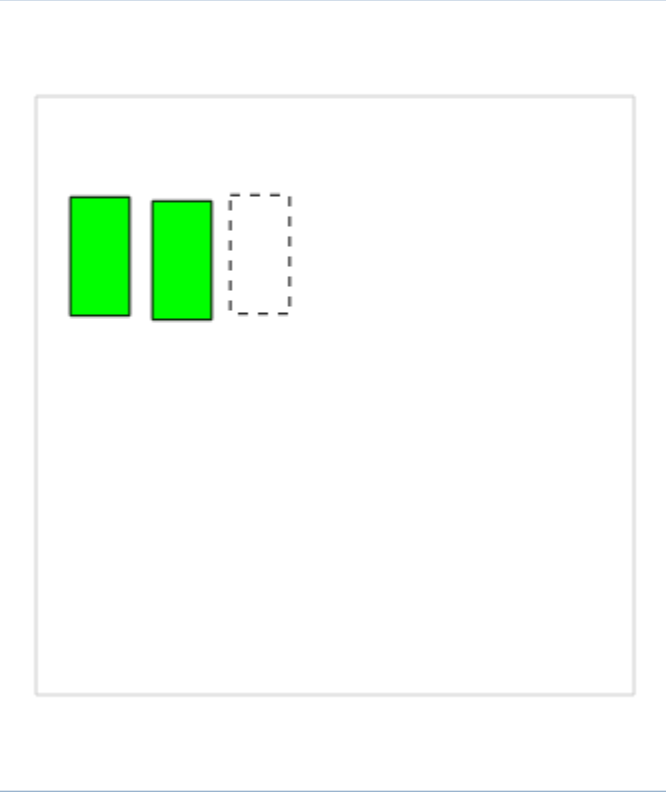
14789

08 / Carton Stack

cMT Viewer (Simulation)

01 / Click and Circles
02 / Click and Lines
03 / Contour Mapping
04 / Plotting
05 / Moving Dots
06 / Rotating Globe
07 / Screen Unlock
08 / Carton Stack
09 / Web API 1
10 / Web API 2
11 / 3D Circles

A simple example for carton stacking



Pallet width: 1000 mm
Pallet height: 1000 mm
Carton width: 100 mm
Carton height: 200 mm


Reset Deg. = 0
Undo

09 / Web API 1

cMT Viewer (Simulation)

- 01 / Click and Circles
- 02 / Click and Lines
- 03 / Contour Mapping
- 04 / Plotting
- 05 / Moving Dots
- 06 / Rotating Globe
- 07 / Screen Unlock
- 08 / Carton Stack
- 09 / Web API 1**
- 10 / Web API 2
- 11 / 3D Circles

Get weather for a random city from openweathermap.org



City: Tainan City

Weather: Mist

Temp.: 18.2

Temp. Feels Like: 18.9

Temp. Range: 17.0 ~ 20.0

10 / Web API 2

cMT Viewer (Simulation)

Request time (UTC): 2020-12-09T18:45:08.595Z
Weather report: Taipei, Taiwan

Light rain
21 °C
11 km/h
8 km
2.5 mm

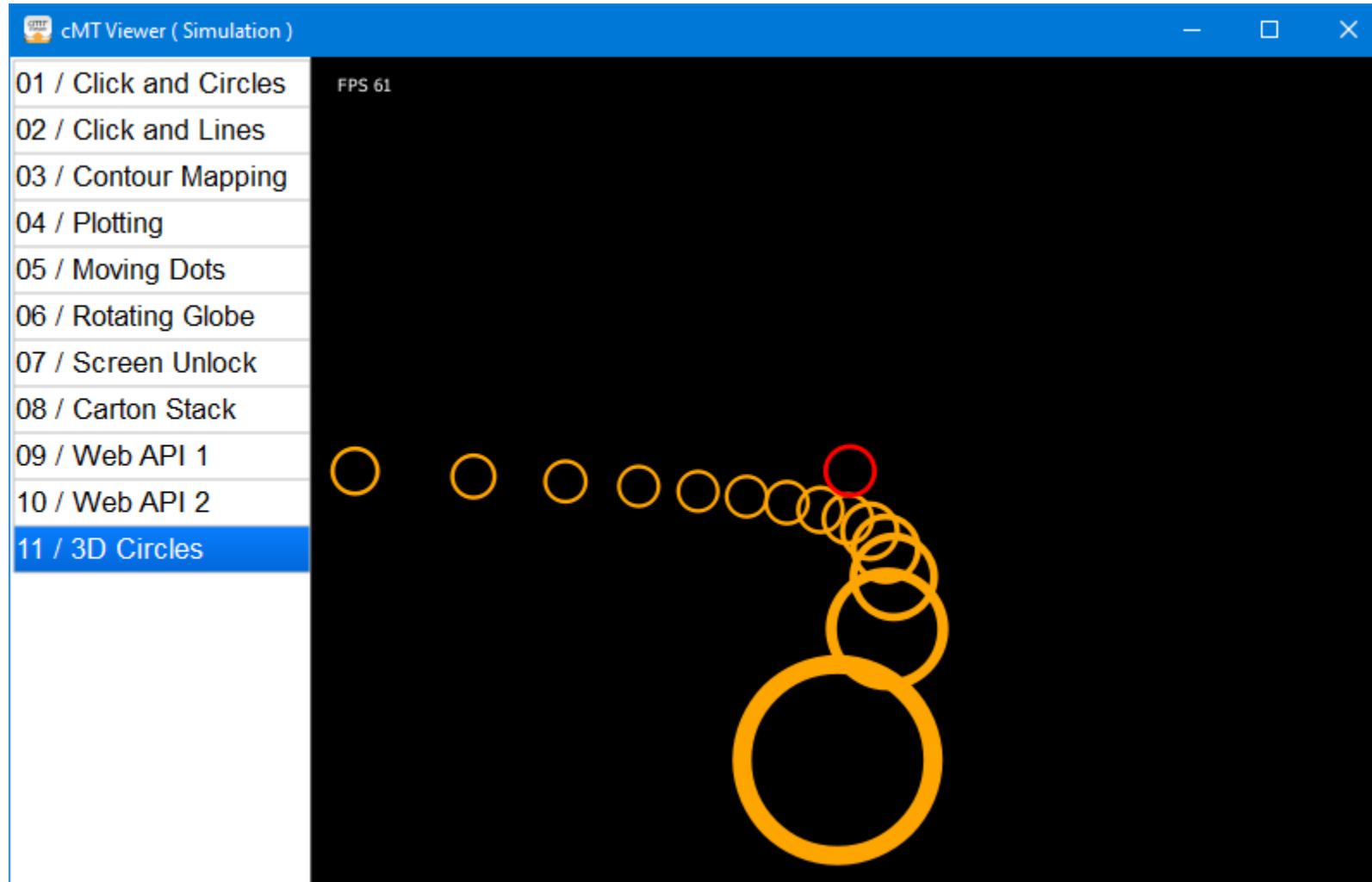
Thu 10 Dec		
Morning	Noon	Evening
Light rain 20 °C 16-21 km/h 8 km 3.2 mm 94%	Light rain 20 °C 17-21 km/h 9 km 1.7 mm 92%	Light rain 17 °C 17-26 km/h 9 km 1.4 mm 91%

Fri 11 Dec		
Morning	Noon	Evening
Patchy rain po... 20 °C 12-14 km/h 9 km 0.6 mm 63%	Light rain sho... 21 °C 12-14 km/h 10 km 0.1 mm 84%	Light rain 19 °C 12-18 km/h 10 km 0.5 mm 81%

Sat 12 Dec		
Morning	Noon	Evening
Light rain sho... 19 °C 13-15 km/h 9 km 1.0 mm 77%	Light rain sho... 19 °C 16-19 km/h 10 km 0.4 mm 74%	Light rain 16 °C 18-27 km/h 10 km 0.3 mm 71%

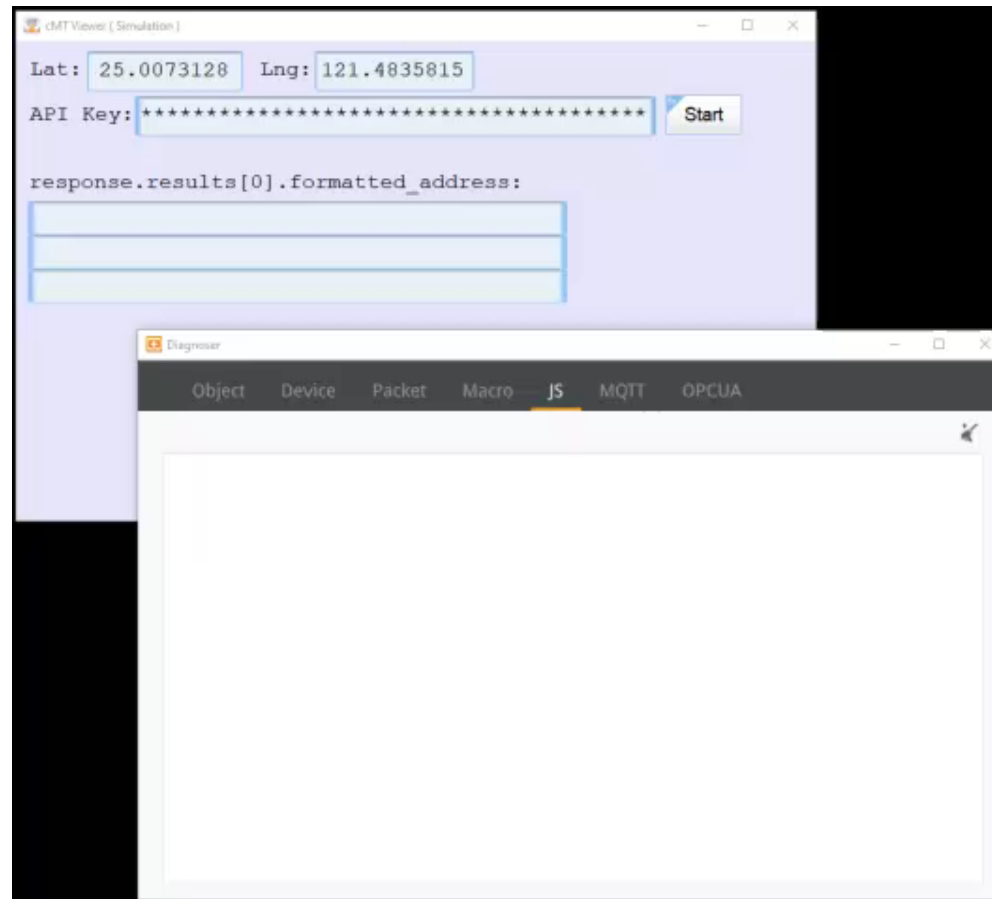
01 / Click and Circles
02 / Click and Lines
03 / Contour Mapping
04 / Plotting
05 / Moving Dots
06 / Rotating Globe
07 / Screen Unlock
08 / Carton Stack
09 / Web API 1
10 / Web API 2
11 / 3D Circles

11 / 3D Circles



Google Reverse Geocoding

- [Overview | Geocoding API | Google Developers](#)
 - Latitude & Longitude => Street Address





JS Object Basics

Overview



- JS Object is a viewer-side object
 - Runs on cMT Viewer
 - On cMT-SVR, if there is no viewer, it does not run
- In EBPro, the object name is called “co_xxx”
 - “co” originates from custom object, the initial name for this object



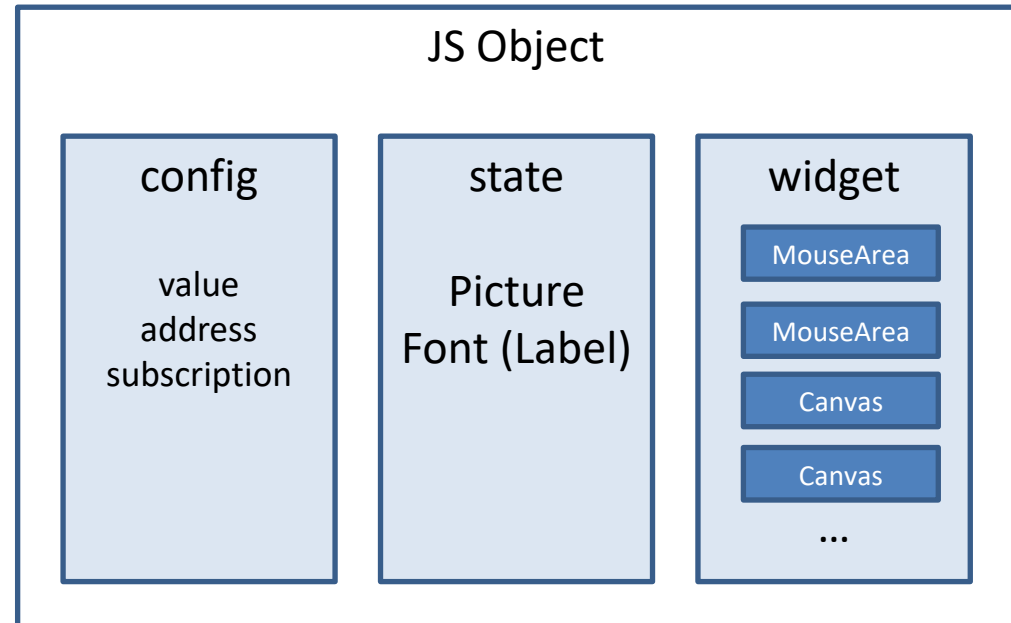
- JS Resource is a management tool for JavaScript libraries
 - Not an EBPro object
- Extends the functionality of JS Object

JS Object

- Support most modern JavaScript syntax (ECMAScript 2020)
- Resources:
 - EBPro Manual
 - Online JS Object SDK

JS Object Introduction

- JS Object (`this`)
 - config (`this.config`)
 - state (`this.state`)
 - Shape
 - Font (Label)
 - widget (`this.widget`)
A Container to accommodate
 - Multiples of `MouseArea` instances
 - Multiples of `Canvas` instances



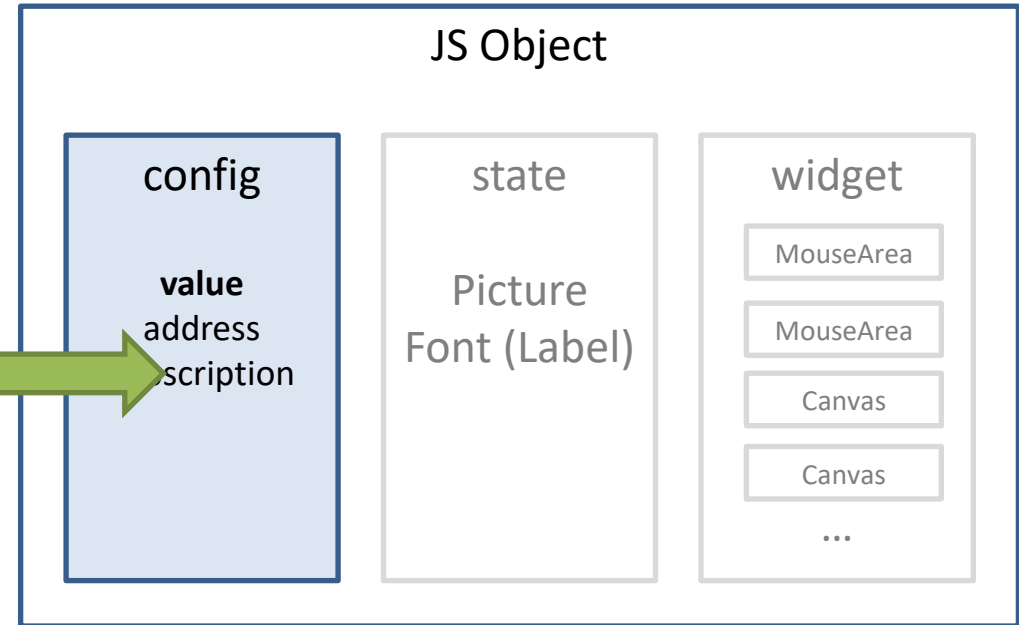
- MouseArea: Detect mouse actions
- Canvas: Draw shapes by using HTML5 canvas-like functions
 - https://www.w3schools.com/html/html5_canvas.asp

JS Object – config

JS Object Object's Properties

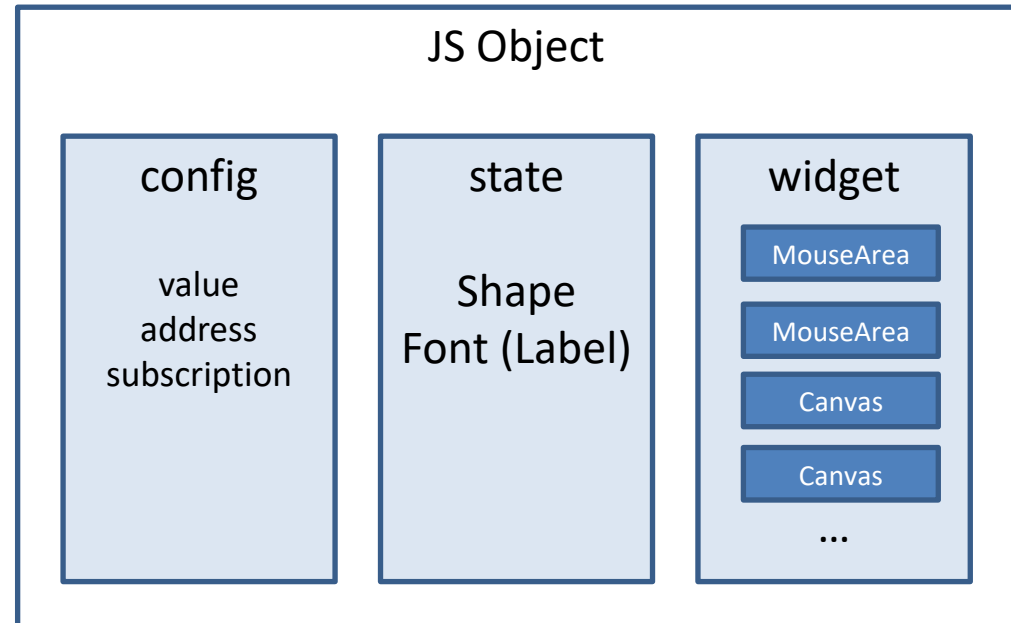
Config Source code Security Shape Font Profile

Name	Type	Value
config	Object (4)	
value	Number	123
value_2	Number	123
value_3	Number	123
array	Array (5)	
[0]	Number	123
[1]	Number	123
[2]	Number	123
[3]	Number	123
[4]	Number	123



JS Object Introduction

- What you can do with JS object
 - Transfer between PLW/LW (between different addresses)
 - Change state by different criteria (Other than what we have in Bit/Word Lamp)



JS Object – Web Request

- Use libcurl instead of requests or http module like in Node JS
- Setup basic curl

```
// ... ignored
const Curl = net.Curl;
const decoder = new TextDecoder('utf-8');
const url = 'http://wttr.in/';

var curl = new Curl.Easy();
curl.setOpt(Curl.Easy.option.URL, url);
curl.setOpt(Curl.Easy.option.FOLLOWLOCATION, true);
curl.setOpt(Curl.Easy.option.SSL_VERIFYPEER, false);
curl.setOpt(Curl.Easy.option.HTTPGET, true);
curl.setOpt(Curl.Easy.option.USERAGENT, "curl/7.00.0");
```

JS Object – Web Request

- Setup WRITEFUNCTION to receive data

```
// This is used to receive the response data
curl.setOpt(Curl.Easy.option.WRITEFUNCTION, function (buf) {
    var responseData = decoder.decode(buf);
    canvas.clearRect(0, 0, self.widget.width, self.widget.height);
    canvas.fillStyle = "black";
    canvas.font = "12px Consolas";

    canvas.fillText("Request time (UTC): " + lastRequestTime.toISOString(), 12, 24);
    var yOffset = 36;
    responseData.split("\n").forEach((line) => {
        line = line.replace(/[\u001b\u009b][[()#;?]*(?:[0-9]{1,4}(?:;[0-9]{0,4})*)?[0-9A-ORZcf-nqry=><]/g, '');
        canvas.fillText(line, 12, yOffset);
        yOffset += 12;
    })
});
```

JS Object – Web Request

- Setup trigger

```
function sendRequest(e) {  
    if (e) {  
        canvas.fillStyle = "green";  
        canvas.fillRect(e.x - 10, e.y - 10, 20, 20);  
    }  
    lastRequestTime = new Date();  
    var multi = new Curl.Multi();  
    multi.onMessage((easyHandle, result) => {  
        multi.removeHandle(easyHandle);  
    });  
  
    multi.addHandle(curl);  
}  
  
mouseArea.on("click", sendRequest);  
  
// request once when js object is created  
sendRequest(null);
```

End of Introduction to JS Object

Thank you for joining us today!

Leave us a note so we know what to cover next for JS Object.

CONTACT US
AND FIND US

