

# debugging



by Park Jinyung

[jh.park@theuber.co.kr](mailto:jh.park@theuber.co.kr)

# INDEX

- What is debugging?
- How to debugging?
- 🌎 Chrome Dev Tools
  - alert();
  - console
  - Network
  - Breakpoints
  - Blackboxing

# What is debugging?

# bug?

프로그램 내의 결함이나 문제점을 이야기하는 것으로  
프로그래밍 언어 COBOL의 개발을 주도한  
그레이스 호퍼가 1945년 Mark II의 오작동 원인을  
찾다가 컴퓨터에 나방이 껴있는 걸 발견한 것

# bug?

0800 Antran started  
1000 " stopped - antran ✓  
13" 06 (032) MP - MC  
(033) PRO 2  
coset

$\left\{ \begin{array}{l} 1.2700 \quad 9.037847025 \\ 9.037846795 \end{array} \right.$  coset  
~~1.982147000~~  
~~2.130476715(-3)~~ 4.615925059(-2)  
2.130476415  
2.130676415

Relays 6-2 in 033 failed special speed test  
in relay

Relay  
2145  
Relay 3370

1100 Started Cosine Tape (Sine check)  
1525 Started Multi Adder Test.

1545



Relay #70 Panel F  
(moth) in relay.

1630 Antran started.  
1700 closed down.

First actual case of bug being found.

# debugging === removeBug

컴퓨터 프로그램의 정확성이나  
논리적인 오류(버그)를 검출하여 제거하는 과정

# How to debugging?



# Chrome Dev Tools



# Chrome Dev Tools

- `alert();`
- `console`
- Network
- Breakpoints
- Blackboxing

```
alert();
```

PASS 😊

console



Chrome Dev Tools > `console`

```
console.log();
```



Chrome Dev Tools > `console`

`console.log();`

`console.dir();`

`console.time();`

`console.group();`

`console.table();`



`console.log();` // 로그

`console.dir();` // 폴더화(객체)

`console.time();` // 경과시간

`console.group();` // 그룹지정

`console.table();` // 데이터 시각화

```
console.log();
```

 **Chrome Dev Tools > console > console.log()**

```
console.log(arr[2]);
```

```
console.log('이름: ' + obj.name);
```

```
if (goal === true) {  
    console.log('성공');  
} else {  
    console.log('실패');  
}
```

```
console.dir();
```



Chrome Dev Tools > `console` > `console.dir();`

```
var el = document.getElementById('name');

console.log(el);
console.dir(el);
```



Chrome Dev Tools > `console` > `console.dir();`

```
> console.log(el);
  ▼<div id="header">
    ►<div class="header_inner">...</div>
    ►<div id="gnb" style="height: 1259px;">...</div>
  </div>
```



Chrome Dev Tools > `console` > `console.dir();`

```
> console.dir(el);
  ▼div#header ⓘ
    accessKey: ""
    align: ""
    assignedSlot: null
  ►attributeStyleMap: StylePropertyMap {size: 0}
  ►attributes: NamedNodeMap {0: id, id: id, length: 1}
    autocapitalize: ""
    baseURI: "http://210.16.195.78:8040/converter/simulation-tool.do"
    childElementCount: 2
  ►childNodes: NodeList(5) [text, div.header_inner, text, div#gnb, text]
  ►children: HTMLCollection(2) [div.header_inner, div#gnb, gnb: div#gnb]
```

```
console.time();
```



Chrome Dev Tools > `console` > `console.time();`

```
console.time('반복문 경과 시간');

var arr = [];
for (var i = 1; i < 1000; i++) {
    for (var j = 1; j < 1000; j++) {
        arr.push(j);
    }
}

console.timeEnd('반복문 경과 시간');
```



Chrome Dev Tools > `console` > `console.time();`

```
> console.time('반복문 경과 시간');

var arr = [];
for (var i = 1; i < 1000; i++) {
    for (var j = 1; j < 1000; j++) {
        arr.push(j);
    }
}

console.timeEnd('반복문 경과 시간');
```

반복문 경과 시간: 20.490234375ms

VM167:10

```
console.group();
```

 **Chrome Dev Tools > console > console.group();**

```
var _name = ['박진형', '김혜인', '김웅진', '최지영'];
var _location = ['하남', '청주', '인천', '경기'];

console.group('이름');
console.log(_name);
console.groupEnd('이름');

console.group('지역');
console.log(_location);
console.groupEnd('지역');
```

 Chrome Dev Tools > `console` > `console.group();`

```
> var _name = ['박진형', '김혜인', '김웅진', '최지영'];
var _location = ['하남', '청주', '인천', '경기'];

console.group('이름');
console.log(_name);
console.groupEnd('이름');

console.group('지역');
console.log(_location);
console.groupEnd('지역');
```

## ▼ 이름

VM737:

```
▶ (4) ["박진형", "김혜인", "김웅진", "최지영"]
```

VM737:

## ▼ 지역

VM737:

```
▶ (4) ["하남", "청주", "인천", "경기"]
```

VM737:

```
console.table();
```



## Chrome Dev Tools > `console` > `console.table()`;

```
var data = [
  {
    "name": "박진형",
    "location": "하남",
    "tel": "4989"
  },
  {
    "name": "정혜윤",
    "location": "남양주",
    "tel": "4987"
  },
  {
    "name": "김웅진",
    "location": "일산",
    "tel": "4978"
  }
];
```



Chrome Dev Tools > `console` > `console.table();`

```
> console.table(data);
```

VM9116:1

(index)	name	location	tel
0	"박진형"	"하남"	"4989"
1	"정혜윤"	"남양주"	"4987"
2	"김웅진"	"일산"	"4978"

▶ Array(3)



# Network



# Chrome Dev Tools > Network

The screenshot shows the Network tab in the Chrome DevTools interface. At the top, there are tabs for Elements, Console, Sources, Network (which is selected), Memory, Performance, Application, Security, Audits, and Adblock Plus. Below the tabs is a toolbar with icons for play, stop, and search, followed by a 'View' section containing 'Group by frame', 'Preserve log', 'Disable cache', and 'Offline' checkboxes. A dropdown menu labeled 'No throttling' is open, showing options: Disabled (selected), No throttling (highlighted in blue), Presets, Fast 3G, Slow 3G, Offline, Custom, test\_loading, and Add... . The main area displays a timeline from 10 ms to 50 ms with various network requests listed below it. The columns in the list are Status, Type, and several others that are mostly empty or have placeholder text like 'Status'.



# Breakpoints





# Chrome Dev Tools > Breakpoints



The screenshot shows the Chrome DevTools interface with the Sources tab selected. The left sidebar displays the file structure, and the main pane shows the code for `common_chart.js`. A red arrow points to line 673, column 75, where a breakpoint is set. The right sidebar contains sections for Watch, Call Stack, Scope, and Breakpoints, all currently empty. The bottom console pane shows the message "Console was cleared".

DevTools - 210.16.195.78:8040/converter/simulation-tool.do

Elements Console Sources Network Memory Performance Application Security Audits Adblock Plus HTML validator

Page Filesystem Overrides >

top  
210.16.195.78:8040  
converter  
simulation-tool.do  
resources  
www.google-analytics.com  
www.googletagmanager.com

simulation-tool.do polyfill.js common.js common\_chart.js

650     /\*\*  
651     \* Frequency  
652     \* 允許값: min 0 이상, max 0 이하  
653     \*/  
654     freqMin = 0.01;  
655     freqMax = 10;  
656     if (freqMin > frequency || frequency > freqMax) { // 이상,  
657         \_htmlItem.push(''<span class="item">Freq.</span>'')  
658         \_listAddFlag = false;  
659     }  
660       
661     \_htmlItem = \_htmlItem.join('\n');  
662       
663     /\* 범위를 벗어난 항목이 한개라도 존재하면 팝업내 목록에 표시 \*/  
664     if (!\_listAddFlag) {  
665         \_html.push('<li>');  
666         \_html.push(''<div class="partname">' + \_partsName + '</div>'')  
667         \_html.push(''<div class="condition">' + \_partsCondition + '</div>'')  
668         \_html.push('</li>');  
669     }

{ } Line 673, Column 75

Watch  
No watch expressions

Call Stack  
Not paused

Scope  
Not paused

Breakpoints  
No breakpoints

XHR/fetch Breakpoints

DOM Breakpoints

Global Listeners

Event Listener Breakpoints

Console What's New Search

top Filter Default levels ▾ VM849:1

Console was cleared

< undefined

>



# Chrome Dev Tools > Breakpoints



The screenshot shows the Chrome DevTools interface with the following tabs open:

- Sources**: The active tab, showing the file structure and code for `simulation-tool.do`. A blue box highlights the "File Navigation" section in the left sidebar.
- Console**: Shows the message "Console was cleared".
- Network**: Shows network requests for `top`, `210.16.195.78:8040`, `converter`, `simulation-tool.do`, `resources`, `www.google-analytics.com`, and `www.googletagmanager.com`.

The main code editor area displays the following JavaScript code from `common_chart.js`:

```
650      /**
651       * Frequency
652       * 유도값: min 0 이상, max 0 이하
653       */
654   freqMin = 0.01;
655   freqMax = 10;
656   if (freqMin > frequency || frequency > freqMax) { // 이상,
657     _htmlItem.push('      <span class="item">Freq.</span>');
658     _listAddFlag = false;
659   }
660
661   _htmlItem = _htmlItem.join('\n');
662
663   /* 범위를 벗어난 항목이 한개라도 존재하면 팝업내 목록에 표시 */
664   if (!_listAddFlag) {
665     _html.push('<li>');
666     _html.push('      <div class="partname">' + _partsName);
667     _html.push('      <div class="condition">');
668     _html.push('        <input type="checkbox" /> ' + name);
669   }
670 }
```

The status bar at the bottom right indicates `VM849:1`.



# Chrome Dev Tools > Breakpoints



The screenshot shows the Chrome DevTools interface with the Sources tab selected. The left sidebar lists the project structure, including files like simulation-tool.do, polyfill.js, common.js, and common\_chart.js. The common\_chart.js file is currently open in the Source Editor, showing code related to frequency calculations and HTML generation. A blue callout box highlights the "Source Editor" label. The right panel contains the Breakpoints sidebar, which is currently empty, indicating no breakpoints have been set.

DevTools - 210.16.195.78:8040/converter/simulation-tool.do

Elements Console Sources Network Memory Performance Application Security Audits Adblock Plus HTML validator

Page Filesystem Overrides »

top  
210.16.195.78:8040  
converter  
resources  
www.google-analytics.com  
www.googletagmanager.com

simulation-tool.do polyfill.js common.js common\_chart.js »

650  
651  
652  
653  
654  
655  
656  
657  
658  
659  
660  
661  
662  
663  
664  
665  
666  
667  
668  
669

```
    /**
     * Frequency
     * min 이하, max 이하
     */
freqMin = 0.01;
freqMax = 10;
if (freqMin > frequency || frequency > freqMax) { // 이하,
    _htmlItem.push('      <span class="item">Freq.</span>');
    _listAddFlag = false;
}

_htmlItem = _htmlItem.join('');

```

Source Editor [업데이트 표시]  
Line 673, Column 75

Console What's New Search

Console was cleared

Default levels ▾

VM849:1

Breakpoints

- No watch expressions
- Not paused
- Not paused
- No breakpoints
- XHR/fetch Breakpoints
- DOM Breakpoints
- Global Listeners
- Event Listener Breakpoints



# Chrome Dev Tools > Breakpoints



The screenshot shows the Chrome DevTools interface with the Sources tab selected. The left sidebar displays the file structure, and the main pane shows the code for `common_chart.js`. A red arrow points to line 673, which contains a conditional statement. A blue box labeled "Debug Info" covers the right sidebar, which includes sections for Watch, Call Stack, Scope, and Breakpoints.

DevTools - 210.16.195.78:8040/converter/simulation-tool.do

Elements Console Sources Network Memory Performance Application Security Audits Adblock Plus HTML validator

Page Filesystem Overrides »

top  
210.16.195.78:8040  
converter  
resources  
www.google-analytics.com  
www.googletagmanager.com

simulation-tool.do polyfill.js common.js common\_chart.js

650       `/*`  
651       `* Frequency`  
652       `* 유도값: min 0 이상, max 0.8`  
653       `*/`  
654       `freqMin = 0.01;`  
655       `freqMax = 10;`  
656       `if (freqMin > frequency || frequency > freqMax) { // 이상,`  
657       `_htmlItem.push(''<span class="item">Freq.</span>'`  
658       `_listAddFlag = false;`  
659     }  
660       
661     `_htmlItem = _htmlItem.join('\n');`  
662       
663     `/* 범위를 벗어난 항목이 한개라도 존재하면 팝업내 목록에 표시 */`  
664     `if (!_listAddFlag) {`  
665       `_html.push('<li>');`  
666       `_html.push(''<div class="partname">' + _partsName + '`  
667       `_html.push(''<div class="condition">');`  
668       `_html.push('</li>');`  
669     }

{ } Line 673, Column 75

Console What's New Search

Default levels ▾

Console was cleared

< undefined

>

Watch No watch expressions

Call Stack Not paused

Scope Not paused

Breakpoints

XHR/fetch Breakpoints

DOM Breakpoints

Global Listeners

Event Listener Breakpoints

VM849:1

 **Chrome Dev Tools > Breakpoints** 

## Debug Info

- Watch
- Call Stack
- Scope
- Breakpoints
- Event Listener Breakpoints

# Chrome Dev Tools > Breakpoints ⏱

The screenshot shows the Chrome Dev Tools interface with the Sources tab selected. A breakpoint is set on line 2277 of the `common_chart.js` file. The code snippet is as follows:

```
* @param {string} typeObj 차트 유형
* @param {object} targetObj 정보 가져올 대상
* @return {object} 선택한 정보
*/
 getInfoData: function (typeObj, targetObj) {
  targetObj = targetObj || {
    selectedPartId: 'selectedList', // default id 2: Selected Parts (선택 목록의 높수 정보)
    modelModeName: 'modelTypeList', // default name 2: Model Mode (체크박스 value 정보)
    mountingName: 'mountingTypeList', // default name 2: Mounting (체크박스 value 정보)
    solderName: 'solderTypeList' // default name 2: Solder (체크박스 value 정보)
  };
  var semChart = SEM.chart; // semChart = {isRun: false, chartOriginData: {...}, chartTempObj: {...}}
  var info = {}; // ajax request data info = {}
  var simulationType = [
    '|Z|': 'TEMPDC',
    'R': 'TEMPDC',
    '|X|': 'TEMPDC',
    '|Z|_R': 'TEMPDC',
    'C': 'TEMPDC',
    'L': 'TEMPDC',
    'Q': 'TEMPDC',
    'DF': 'TEMPDC',
    'SDB': 'TEMPDC',
    'Smithchart': 'TEMPDC',
    'LTemp': 'TEMPDC',
    'ACRF': 'TEMPDC',
    'Ls': 'TEMPDC',
    'Rs': 'TEMPDC',
    '|L|_R': 'TEMPDC'
  ];
  ...
}

Line 2277, Column 30
```

The Watch sidebar on the right shows the following variables:

- Debugger paused
- Watch
  - info: Object
    - \_\_proto\_\_: Object
- Call Stack
  - getInfoData common\_chart.js:2279
  - (anonymous) common\_chart.js:824
  - dispatch jquery-1.11.3.min.js:4
  - r.handle jquery-1.11.3.min.js:4
- Scope
- Local
  - capacitorPart: undefined
  - getCheckedValue: f getCheckedValue(name)
  - inductorPart: undefined
  - info: {}
    - infoPart: undefined
    - modelModeChecked: undefined
    - mountingChecked: undefined
    - sType: undefined
  - semChart: {isRun: false, chartOriginData: {...}, chartTempObj: {...}}
  - simulationType: undefined
  - solderChecked: undefined
  - targetObj: {selectedPartId: "selectedList", modelModeName: "modelTypeList", mountingName: "mountingTypeList", solderName: "solderTypeList"}
  - this: Object

관찰하고 싶은 객체, 변수 등을 등록

# Chrome Dev Tools > Breakpoints ⏱

The screenshot shows the Chrome DevTools interface with the Sources tab selected. A breakpoint is set on line 2277 of the file `common_chart.js`. The code snippet is as follows:

```
* @param {string} typeObj 차트 유형
* @param {object} targetObj 정보 가져올 대상
* @return {object} 선택한 정보
*/
 getInfoData: function (typeObj, targetObj) {
  typeObj = {chartType: "InductorVoltage"}, target
  targetObj = targetObj || {
    selectedPartId: 'selectedList', // default id 2t : Selected Parts (대상 목록의 높수 정보)
    modelModeName: 'modelTypeList', // default name 2t : Model Mode (체크박스 value 정보)
    mountingName: 'mountingTypeList', // default
    solderName: 'solderTypeList' // default name
  };
  var semChart = SEM.chart; // semChart = {isRun: fa
  var info = {}; // ajax request data info = {}
  var simulationType = [
    '|Z|': 'TEMPDC',
    'R': 'TEMPDC',
    '|X|': 'TEMPDC',
    '|Z|_R': 'TEMPDC',
    'C': 'TEMPDC',
    'L': 'TEMPDC',
    'Q': 'TEMPDC',
    'DF': 'TEMPDC',
    'SDB': 'TEMPDC',
    'Smithchart': 'TEMPDC',
    'LTemp': 'TEMPDC',
    'ACRF': 'TEMPDC',
    'Ls': 'TEMPDC',
    'Rs': 'TEMPDC',
    '|V|_L': 'TEMPDC'
  ];
  ...
}

Line 2279, Column 30
```

The right panel displays the **Call Stack** and **Scope** panes. The **Call Stack** pane shows the stack trace from `getInfoData` down to `r.handle`. The **Scope** pane shows local variables and their current values.

**Call Stack**

- getInfoData common\_chart.js:2279
- (anonymous) common\_chart.js:824
- dispatch jquery-1.11.3.min.js:4
- r.handle jquery-1.11.3.min.js:4

**Scope**

- Local
  - capacitorPart: undefined
  - getCheckedValue: f getCheckedValue(name)
  - inductorPart: undefined
  - info: {}
    - infoPart: undefined
    - modelModeChecked: undefined
    - mountingChecked: undefined
    - sType: undefined
  - semChart: {isRun: false, chartOriginData: {...}, chartTempO...}
  - simulationType: undefined
  - solderChecked: undefined
  - targetObj: {selectedPartId: "selectedList", modelModeName...}
  - this: Object

실행중인 정보

# Chrome Dev Tools > Breakpoints

The screenshot shows the Chrome DevTools interface with the 'Breakpoints' tab selected. The left pane displays the source code for 'common\_chart.js' with several breakpoints set, indicated by red dots. The right pane shows the 'Scope' and 'Call Stack' sections. A large blue button labeled 'Scope' is overlaid on the right side of the interface.

Debugger paused

Watch

info: Object

Call Stack

getInfoData common\_chart.js:2279

(anonymous) common\_chart.js:824

dispatch jquery-1.11.3.min.js:4

r.handle jquery-1.11.3.min.js:4

Scope

Local

- capacitorPart: undefined
- getCheckedValue: f getCheckedValue(name)
- inductorPart: undefined
- info: {}
- infoPart: undefined
- modelModeChecked: undefined
- mountingChecked: undefined
- sType: undefined
- semChart: {isRun: false, chartOriginData: {...}, chartTempO...}
- simulationType: undefined
- solderChecked: undefined
- targetObj: {selectedPartId: "selectedList", modelModeName...}
- this: Object

Console What's New Search

Default levels ▾

Line 2279, Column 30

Scope

현재 시점의 유효 범위에서 등록된 변수 목록

# Chrome Dev Tools > Breakpoints

The screenshot shows the Chrome Dev Tools interface with the Sources tab selected. A blue callout box labeled "Breakpoints" points to the Breakpoints section in the right sidebar. In the code editor, line 2277 of `common_chart.js` is highlighted:

```
    var semChart = SEM.chart; semChart = {isRun: false, chartOriginData: {...}, chartTempObj: {...}}
    var info = {}; // ajax request data info = {}
```

The right sidebar displays the current scope variables and the Breakpoints section, which lists two breakpoints:

- `common_chart.js:2277`: `var semChart = SEM.chart;`
- `common_chart.js:2310`: `info.graphType = typeObj.chartType;`

At the bottom of the sidebar, there are sections for XHR/fetch Breakpoints, DOM Breakpoints, Global Listeners, and Event Listener Breakpoints.

지정한 포인트에서 중단

# Chrome Dev Tools > Breakpoints

The screenshot shows the Chrome DevTools interface with the 'Breakpoints' tab selected in the top navigation bar. The main pane displays a portion of the 'common\_chart.js' file, specifically the 'getInfoData' function. A blue callout box highlights the 'Event Listener Breakpoints' section in the right-hand sidebar. This sidebar lists various event types under the 'Control' category, with 'submit' checked, indicating it's a currently active breakpoint.

```
    * @param {string} typeObj 차트 유형
    * @param {object} targetObj 정보 가져올 대상
    * @return {object} 선택한 정보
    */
 getInfoData: function (typeObj, targetObj) {
    targetObj = targetObj || {
        selectedPartId: 'selectedList', // default id 값 : Selected Parts (내장 목록의 높수 정보)
        modelModeName: 'modelTypeList', // default name 값 : Model Mode (체크박스 value 정보)
        mountingName: 'mountingTypeList', // default name 값 : Mounting (체크박스 value 정보)
        solderName: 'solderTypeList' // default name 값 : Solder (체크박스 value 정보)
    };
    var semChart = SEM.chart;
    var info = {}; // ajax request data
    var simulationType = {
        'Z': 'TEMPDC',
        'R': 'TEMPDC',
        'X': 'TEMPDC',
        'Z_R': 'TEMPDC',
        'C': 'TEMPDC',
        'L': 'TEMPDC',
        'Q': 'TEMPDC',
        'DF': 'TEMPDC',
        'SDB': 'TEMPDC',
        'Smithchart': 'TEMPDC',
        'LTemp': 'TEMPDC',
        'ACRF': 'TEMPDC',
        'Ls': 'TEMPDC',
        ...
    }
}
```

Event Listener Breakpoints

- Animation
- Canvas
- Clipboard
- Control
  - resize
  - scroll
  - zoom
  - focus
  - blur
  - select
  - change
  - submit
  - reset
- DOM Mutation
- Device

체크한 event가 발생할 때 중단

# Chrome Dev Tools > Breakpoints

The screenshot shows the Chrome Dev Tools interface with the 'Sources' tab selected. The left sidebar displays the file structure and the current file is 'common\_chart.js'. The code editor shows a function definition with several lines highlighted in blue, indicating they are part of a selected block. Line 2308 is specifically highlighted with a red box and has a blue arrow pointing to it from the top right corner of the slide. The right sidebar contains the 'Breakpoints' panel, which lists a single breakpoint at line 2308 of 'common\_chart.js'.

```
2290     return infoDown;
2291 },
2292 /**
2293 * 차트 그릴 때 필요한 정보 가져오기 - ajax request data
2294 * @function SEM.chart.getInfoData
2295 * @param {string} typeObj 차트 유형
2296 * @param {object} targetObj 정보를 대상
2297 * @return {object} 선택한 정보
2298 */
2299 getInfoData: function (typeObj, targetObj) {
2300     targetObj = targetObj || {
2301         selectedPartId: 'selectedList', // default id 값 : Selected Parts (선택 목록의 목수)
2302         modelModeName: 'modelTypeList', // default name 값 : Model Mode (체크박스 value 정보)
2303         mountingName: 'mountingTypeList', // default name 값 : Mounting (체크박스 value 정보)
2304         solderName: 'solderTypeList' // default name 값 : Solder (체크박스 value 정보)
2305     };
2306 }
2307 var semChart = SEM.chart;
2308 var info = {}; // ajax request data
2309 var simulationType = {
2310     'Z': 'TEMPDC',
2311     'R': 'TEMPDC',
2312     '|X|': 'TEMPDC',
2313     'Z|R': 'TEMPDC',
2314     'C': 'TEMPDC',
2315     'L': 'TEMPDC',
2316     'Q': 'TEMPDC',
2317     'DF': 'TFMPDC'
2318 }
```

# Chrome Dev Tools > Breakpoints

The screenshot shows the Chrome Dev Tools interface with the 'Sources' tab selected. The left sidebar lists files: polyfill.js, common.js, simulation-tool.do, jquery-1.11.3.min.js, and common\_chart.js. The code editor displays common\_chart.js with several lines highlighted in blue, indicating they are part of a breakpoint. A red box highlights the 'Breakpoints' section in the right sidebar, which contains a single entry: common\_chart.js:2308 with the line of code: var info = {};

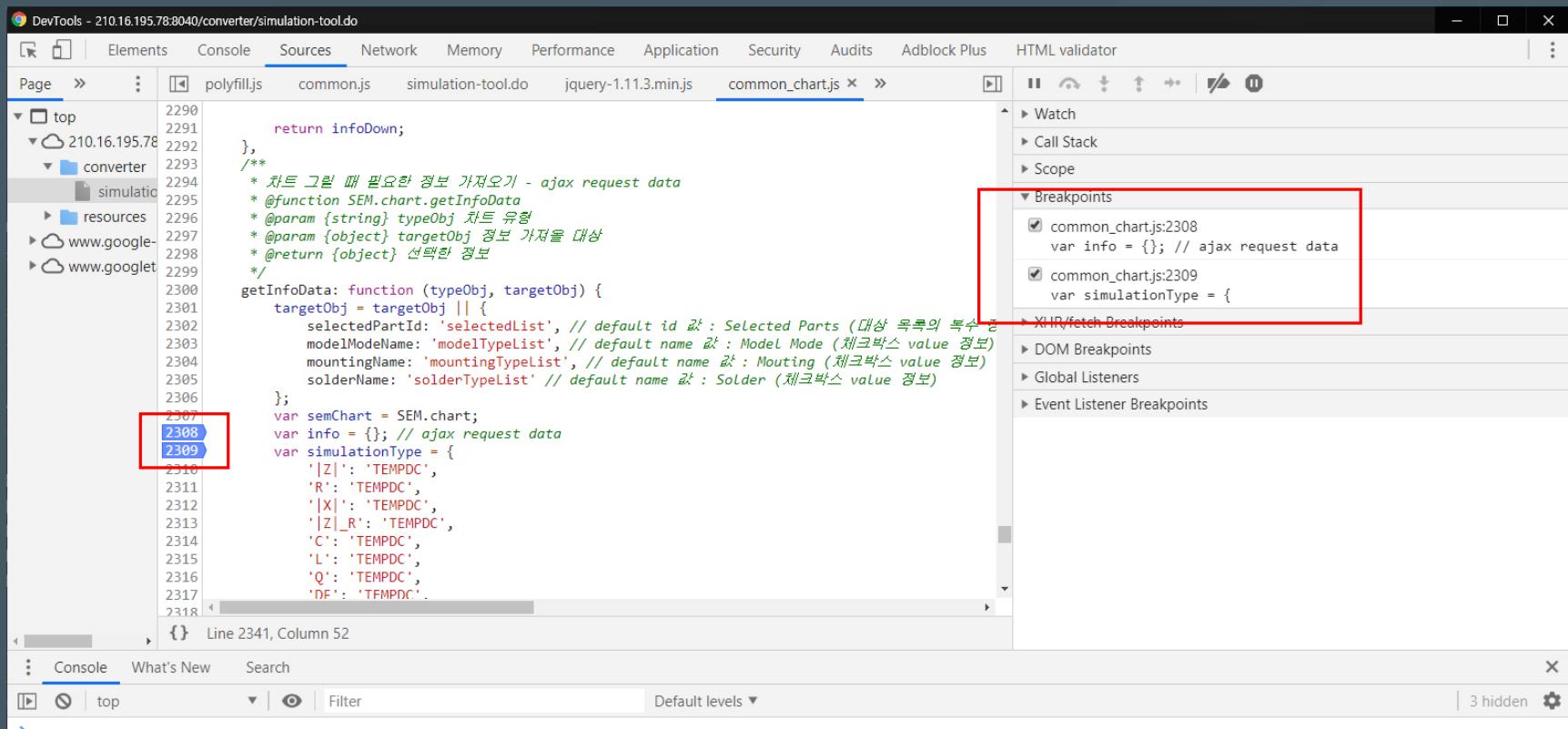
```
2290     return infoDown;
2291 },
2292 /**
2293 * 차트 그릴 때 필요한 정보 가져오기 - ajax request data
2294 * @function SEM.chart.getInfoData
2295 * @param {string} typeObj 차트 유형
2296 * @param {object} targetObj 정보를 대상
2297 * @return {object} 선택한 정보
2298 */
2299 getInfoData: function (typeObj, targetObj) {
2300     targetObj = targetObj || {
2301         selectedPartId: 'selectedList', // default id 값 : Selected Parts (선택 목록의 목수)
2302         modelModeName: 'modelTypeList', // default name 값 : Model Mode (체크박스 value 정보)
2303         mountingName: 'mountingTypeList', // default name 값 : Mounting (체크박스 value 정보)
2304         solderName: 'solderTypeList' // default name 값 : Solder (체크박스 value 정보)
2305     };
2306     var semChart = SEM.chart;
2307     var info = {};
```

Line 2341, Column 52

Console What's New Search

Default levels ▾ 3 hidden

# Chrome Dev Tools > Breakpoints



다중 선택, 중단점 사용 여부 선택 가능

# Chrome Dev Tools > Breakpoints

The screenshot shows the Chrome Dev Tools interface with the 'Breakpoints' tab selected. The left pane displays a portion of the JavaScript code for 'boltev\_prev\_contact.js'. A red box highlights the play button icon in the toolbar above the code editor. The right pane shows the 'Paused on breakpoint' status, the call stack, and the local variables. A specific breakpoint is highlighted in yellow, showing the line of code: `var carPrice = formData.totalPrice;` at line 583.

```
// 할부금 계산기 function
function priceCalculator() {
    /*
        carPrice = 차량금액;
        preValue = 선수금;
        totalPrice = 총 상환액;
        installPrice = 할부원금;
        carRate = 할부이율(%);
        monthSelect = 할부개월;
        incomeTotalPrice = 총 지급이자;
        monthPrice = 월 할부금;
        newInstallPrice = 대출금액;
        incomePrice = 대출이자;
    */
    var carPrice = formData.totalPrice;
    var preValue = parseInt($("#prePrice").val()) * 10000;
    var totalPrice = parseInt($("#totalPrice").text());
    var installPrice = parseInt($("#installPrice").text());
    var carRate = Number($("#carRate").val().replace(/[^~\.\d]/g, '')) / 100;
    var monthSelect = parseInt($("#monthSelect").val());
    var incomeTotalPrice = 0;
    var incomePrice = 0;
    var monthPrice, newInstallPrice;

    // 할부원금 (차량금액 - 선수금)
    installPrice = carPrice - preValue;

    // 월 할부금 (월리금 곱동방식 계산 로직)
    monthPrice = installPrice * carRate / 12 * Math.pow((1 + carRate / 12), monthSelect) / (Math.pow((1 + carRate / 12), monthSelect) - 1);

    // 총 지급이자
    newInstallPrice = installPrice;

    for (i = 0; i <= monthSelect - 1; i++) {
        // 대출이자 계산 (대출금액 * 할부이율 / 12)
        incomePrice = newInstallPrice * carRate / 12;

        // 대출금 계산 (대출금액 - (월 할부금 - 대출이자))
        newInstallPrice = newInstallPrice - (monthPrice - incomePrice);
    }
}
```

Paused on breakpoint

Call Stack

Scope

Local

Breakpoints

Global

DOM Breakpoints

중단점을 그만하고 코드 계속 실행(다음 중단점 이동)

# Chrome Dev Tools > Breakpoints

The screenshot shows the Chrome Dev Tools interface with the 'Breakpoints' tab selected. The left pane displays a portion of the JavaScript code from 'boltev\_prev\_contact.js'. A red box highlights the play button icon in the toolbar above the code editor. The right pane shows the 'Breakpoints' panel, which includes sections for 'Paused on breakpoint', 'Call Stack', 'Scope', 'Local', 'Global', 'Breakpoints' (which is currently selected and has a yellow background), 'XHR/fetch Breakpoints', and 'DOM Breakpoints'. In the 'Breakpoints' section, there is a checkbox for 'boltev\_prev\_contact.js:583' with the condition 'var carPrice = formData.totalPrice;'. The status bar at the bottom indicates 'Line 583, Column 17'.

```
// 할부금 계산기 function
function priceCalculator() {
    /*
        carPrice = 차량금액;
        preValue = 선수금;
        totalPrice = 총 상환액;
        installPrice = 할부원금;
        carRate = 할부이율(%);
        monthSelect = 할부개월;
        incomeTotalPrice = 총 지급이자;
        monthPrice = 월 할부금;
        newInstallPrice = 대출금액;
        incomePrice = 대출이자;
    */
    var carPrice = formData.totalPrice;
    var preValue = parseInt($("#prePrice").val()) * 10000;
    var totalPrice = parseInt($("#totalPrice").text());
    var installPrice = parseInt($("#installPrice").text());
    var carRate = Number($("#carRate").val().replace(/[^~\.\d]/g, '')) / 100;
    var monthSelect = parseInt($("#monthSelect").val());
    var incomeTotalPrice = 0;
    var incomePrice = 0;
    var monthPrice, newInstallPrice;

    // 할부원금 (차량금액 - 선수금)
    installPrice = carPrice - preValue;

    // 월 할부금 (월리금 곱동방식 계산 로직)
    monthPrice = installPrice * carRate / 12 * Math.pow((1 + carRate / 12), monthSelect) / (Math.pow((1 + carRate / 12), monthSelect) - 1);

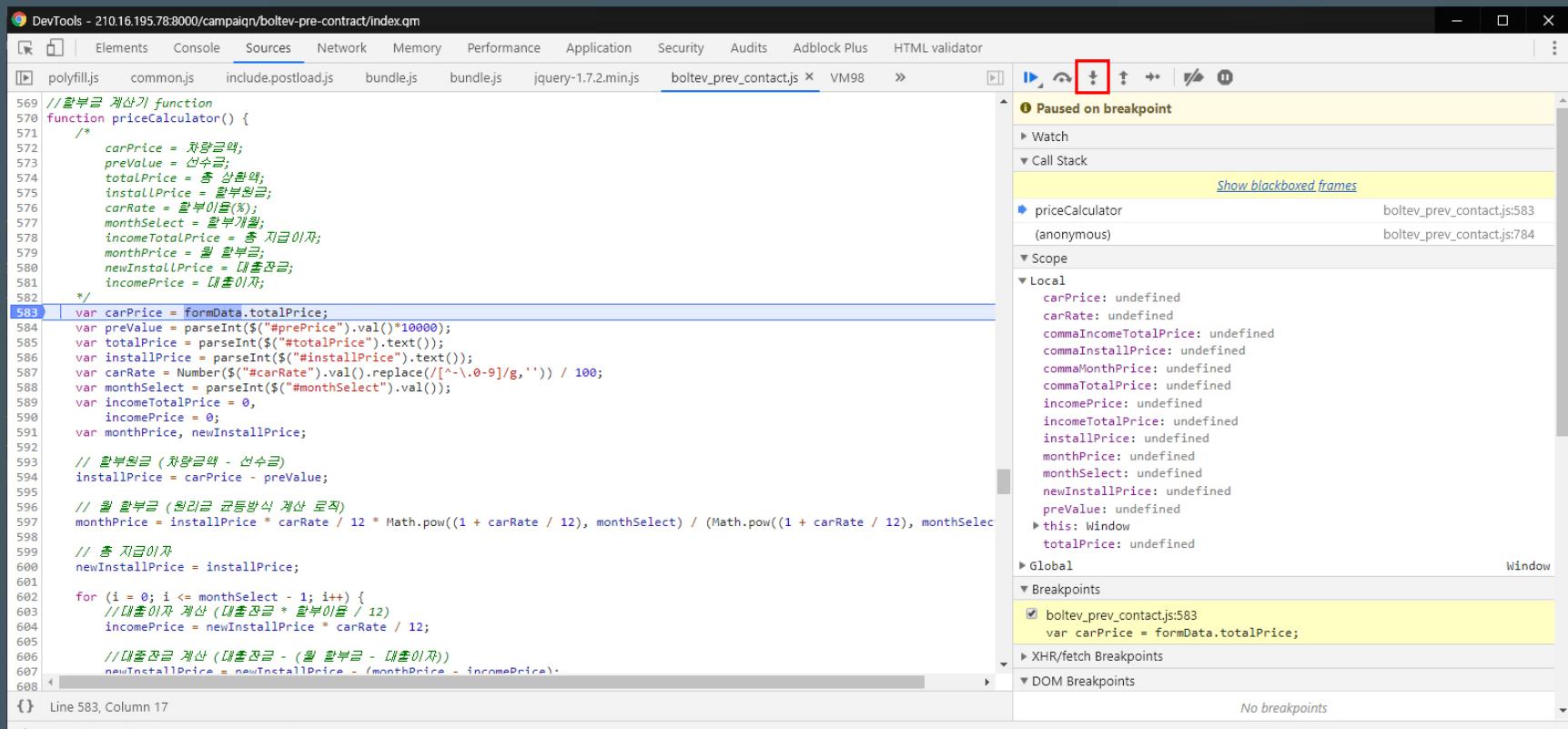
    // 총 지급이자
    newInstallPrice = installPrice;

    for (i = 0; i <= monthSelect - 1; i++) {
        // 대출이자 계산 (대출금액 * 할부이율 / 12)
        incomePrice = newInstallPrice * carRate / 12;

        // 대출금 계산 (대출금액 - (월 할부금 - 대출이자))
        newInstallPrice = newInstallPrice - (monthPrice - incomePrice);
    }
}
```

다음 함수 코드로 이동

# Chrome Dev Tools > Breakpoints



The screenshot shows the Chrome Dev Tools interface with the 'Breakpoints' tab selected in the top navigation bar. The main pane displays a portion of the JavaScript code for 'boltev\_prev\_contact.js'. A red box highlights the play button icon in the toolbar above the code editor.

```
569 // 할부금 계산기 function
570 function priceCalculator() {
571   /*
572     carPrice = 차량금액;
573     preValue = 선수금;
574     totalPrice = 총 상환액;
575     installPrice = 할부원금;
576     carRate = 할부이율(%);
577     monthSelect = 할부개월;
578     incomeTotalPrice = 총 지급이자;
579     monthPrice = 월 할부금;
580     newInstallPrice = 대출금액;
581     incomePrice = 대출이자;
582   */
583   var carPrice = formData.totalPrice;
584   var preValue = parseInt($("#prePrice").val()) * 10000;
585   var totalPrice = parseInt($("#totalPrice").text());
586   var installPrice = parseInt($("#installPrice").text());
587   var carRate = Number($("#carRate").val().replace(/[^~\.\d]/g, '')) / 100;
588   var monthSelect = parseInt($("#monthSelect").val());
589   var incomeTotalPrice = 0;
590   var incomePrice = 0;
591   var monthPrice, newInstallPrice;
592
593   // 할부원금 (차량금액 - 선수금)
594   installPrice = carPrice - preValue;
595
596   // 월 할부금 (월리금 곱동방식 계산 로직)
597   monthPrice = installPrice * carRate / 12 * Math.pow((1 + carRate / 12), monthSelect) / (Math.pow((1 + carRate / 12), monthSelect) - 1);
598
599   // 총 지급이자
600   newInstallPrice = installPrice;
601
602   for (i = 0; i <= monthSelect - 1; i++) {
603     // 대출이자 계산 (대출금액 * 할부이율 / 12)
604     incomePrice = newInstallPrice * carRate / 12;
605
606     // 대출금 계산 (대출금액 - (월 할부금 - 대출이자))
607     newInstallPrice = newInstallPrice - (monthPrice - incomePrice);
608   }
609 }
```

The right sidebar contains the Breakpoints panel. It shows a list of breakpoints, with one specific breakpoint highlighted in yellow. This highlighted breakpoint is located at line 583, column 17 of the code, where the variable 'carPrice' is assigned the value of 'formData.totalPrice'. The 'Breakpoints' section also includes sections for 'Global', 'XHR/fetch Breakpoints', and 'DOM Breakpoints', both of which are currently empty.

함수의 내부 코드로 이동

# Chrome Dev Tools > Breakpoints

The screenshot shows the Chrome Dev Tools interface with the 'Breakpoints' tab selected. The left pane displays a portion of the JavaScript code from 'boltev\_prev\_contact.js' (lines 569 to 608). The right pane shows the 'Breakpoints' panel, which is currently paused on a breakpoint at line 583. The 'Breakpoints' section lists a single breakpoint for 'boltev\_prev\_contact.js:583'. The 'Scope' section shows the current variable context, and the 'Global' section shows the global object 'Window'. The 'Call Stack' section shows the stack trace: 'priceCalculator' at line 583 and '(anonymous)' at line 784.

```
// 할부금 계산기 function
function priceCalculator() {
    /*
        carPrice = 차량금액;
        preValue = 선수금;
        totalPrice = 총 상환액;
        installPrice = 할부원금;
        carRate = 할부이율(%);
        monthSelect = 할부개월;
        incomeTotalPrice = 총 지급이자;
        monthPrice = 월 할부금;
        newInstallPrice = 대출금액;
        incomePrice = 대출이자;
    */
    var carPrice = formData.totalPrice;
    var preValue = parseInt($("#prePrice").val()) * 10000;
    var totalPrice = parseInt($("#totalPrice").text());
    var installPrice = parseInt($("#installPrice").text());
    var carRate = Number($("#carRate").val().replace(/[^~\.\d]/g, '')) / 100;
    var monthSelect = parseInt($("#monthSelect").val());
    var incomeTotalPrice = 0;
    var incomePrice = 0;
    var monthPrice, newInstallPrice;

    // 할부원금 (차량금액 - 선수금)
    installPrice = carPrice - preValue;

    // 월 할부금 (월리금 곱동방식 계산 로직)
    monthPrice = installPrice * carRate / 12 * Math.pow((1 + carRate / 12), monthSelect) / (Math.pow((1 + carRate / 12), monthSelect) - 1);

    // 총 지급이자
    newInstallPrice = installPrice;

    for (i = 0; i <= monthSelect - 1; i++) {
        // 대출이자 계산 (대출금액 * 할부이율 / 12)
        incomePrice = newInstallPrice * carRate / 12;

        // 대출금 계산 (대출금액 - (월 할부금 - 대출이자))
        newInstallPrice = newInstallPrice - (monthPrice - incomePrice);
    }
}
```

Paused on breakpoint

Watch

Call Stack

Scope

Local

Global

Breakpoints

XHR/fetch Breakpoints

DOM Breakpoints

현재 함수에서 빠져나와 다음 함수로 이동

# Chrome Dev Tools > Breakpoints

DevTools - 210.16.195.78:8000/campaign/boltev-pre-contact/index.js

Sources Network Memory Performance Application Security Audits Adblock Plus HTML validator

polyfill.js common.js include.postload.js bundle.js jquery-1.7.2.min.js boltev\_prev\_contact.js VM98

```
569 // 할부금 계산기 function
570 function priceCalculator() {
571   /*
572     carPrice = 차량금액;
573     preValue = 선수금;
574     totalPrice = 총 상환액;
575     installPrice = 할부원금;
576     carRate = 할부이율(%);
577     monthSelect = 할부개월;
578     incomeTotalPrice = 총 지급이자;
579     monthPrice = 월 할부금;
580     newInstallPrice = 대출금액;
581     incomePrice = 대출이자;
582   */
583   var carPrice = formData.totalPrice;
584   var preValue = parseInt($("#prePrice").val()) * 10000;
585   var totalPrice = parseInt($("#totalPrice").text());
586   var installPrice = parseInt($("#installPrice").text());
587   var carRate = Number($("#carRate").val().replace(/[^~\.\d]/g, '')) / 100;
588   var monthSelect = parseInt($("#monthSelect").val());
589   var incomeTotalPrice = 0;
590   var incomePrice = 0;
591   var monthPrice, newInstallPrice;
592
593   // 할부원금 (차량금액 - 선수금)
594   installPrice = carPrice - preValue;
595
596   // 월 할부금 (월리금 곱동방식 계산 로직)
597   monthPrice = installPrice * carRate / 12 * Math.pow((1 + carRate / 12), monthSelect) / (Math.pow((1 + carRate / 12), monthSelect) - 1);
598
599   // 총 지급이자
600   newInstallPrice = installPrice;
601
602   for (i = 0; i <= monthSelect - 1; i++) {
603     // 대출이자 계산 (대출금액 * 할부이율 / 12)
604     incomePrice = newInstallPrice * carRate / 12;
605
606     // 대출금액 계산 (대출금액 - (월 할부금 - 대출이자))
607     newInstallPrice = newInstallPrice - (monthPrice - incomePrice);
608   }
}
```

{ } Line 583, Column 17

Paused on breakpoint

Watch

Call Stack

Scope

Local

Global

Breakpoints

XHR/fetch Breakpoints

DOM Breakpoints

No breakpoints

# Chrome Dev Tools > Breakpoints

DevTools - 210.16.195.78:8000/campaign/boltev-pre-contact/index.js

Sources    Elements    Console    Network    Memory    Performance    Application    Security    Audits    Adblock Plus    HTML validator

polyfill.js    common.js    include.postload.js    bundle.js    bundle.js    jquery-1.7.2.min.js    boltev\_prev\_contact.js    VM98

```
569 // 할부금 계산기 function
570 function priceCalculator() {
571   /*
572     carPrice = 차량금액;
573     preValue = 선수금;
574     totalPrice = 총 상품액;
575     installPrice = 할부원금;
576     carRate = 할부이율(%);
577     monthSelect = 할부개월;
578     incomeTotalPrice = 총 지급여자;
579     monthPrice = 월 할부금;
580     newInstallPrice = 대출금액;
581     incomePrice = 대출여자;
582   */
583   var carPrice = formData.totalPrice; carPrice = 45930000
584   var preValue = parseInt($("#prePrice").val()) * 10000; preValue = 1000000
585   var totalPrice = parseInt($("#totalPrice").text()); totalPrice = 45
586   var installPrice = parseInt($("#installPrice").text()); installPrice = 44930000
587   var carRate = Number($("#carRate").val().replace(/[^~\.\-0-9]/g,'')) / 100; carRate = 0.019
588   var monthSelect = parseInt($("#monthSelect").val()); monthSelect = 12
589   var incomeTotalPrice = 0, incomeTotalPrice = 0
590   incomePrice = 0; incomePrice = 0
591   var monthPrice, newInstallPrice; monthPrice = undefined, newInstallPrice = undefined
592
593   // 할부원금 (차량금액 - 선수금)
594   installPrice = carPrice - preValue; installPrice = 44930000, carPrice = 45930000, preValue = 1000000
595
596   // 월 할부금 (월리금 곱동방식 계산 로직)
597   monthPrice = installPrice * carRate / 12 * Math.pow((1 + carRate / 12), monthSelect) / (Math.pow((1 + carRate / 12), monthSelect) - 1)
598
599   // 총 지급여자
600   newInstallPrice = installPrice;
601
602   for (i = 0; i <= monthSelect - 1; i++) {
603     // 대출여자 계산 (대출금액 * 할부이율 / 12)
604     incomePrice = newInstallPrice * carRate / 12;
605
606     // 대출금 계산 (대출금액 - (월 할부금 - 대출여자))
607     newInstallPrice = newInstallPrice - monthPrice;
608   }
609 }
```

{ } Line 597, Column 2

Debugger paused

Watch

Call Stack

Scope

Local

Global

Breakpoints

Window

No breakpoints

# Blackboxing



# Chrome Dev Tools > Blackboxing

The screenshot shows the Chrome DevTools interface with the Sources tab selected. The left sidebar displays the file structure of the current page, with 'simulation-tool.do' selected. The main pane shows a portion of the 'common\_chart.js' file, specifically lines 650 through 673. The code is annotated with green and red highlights. A context menu is open on the right side of the interface, with the 'Dock side' option highlighted.

DevTools - 210.16.195.78:8040/converter/simulation-tool.do

Sources

simulation-tool.do polyfill.js common.js common\_chart.js

650     /\*\*  
651     \* Frequency  
652     \* 允許範囲: min 0 이상, max 0.01  
653     \*/  
654     freqMin = 0.01;  
655     freqMax = 10;  
656     if (freqMin > frequency || frequency > freqMax) { // 이상,  
657         \_htmlItem.push(''<span class="item">Freq.</span>'')  
658         \_listAddFlag = false;  
659     }  
660       
661     \_htmlItem = \_htmlItem.join('\n');  
662       
663     /\* 범위를 벗어난 항목이 한개라도 존재하면 팝업내 목록에 표시 \*/  
664     if (!\_listAddFlag) {  
665         \_html.push('<li>');  
666         \_html.push(''<div class="partname">' + \_partsName + '</div>'')  
667         \_html.push(''<div class="condition">'');  
668         \_html.push(''</div>'')  
669     }

{ } Line 673, Column 75

Console What's New Search

Default levels ▾

Console was cleared

undefined

VM849:1

Dock side

Focus debugger

Hide console drawer Esc

Search Ctrl+Shift+F

Run command Ctrl+Shift+P

Open file Ctrl+P

More tools

Shortcuts

Settings F1

Help

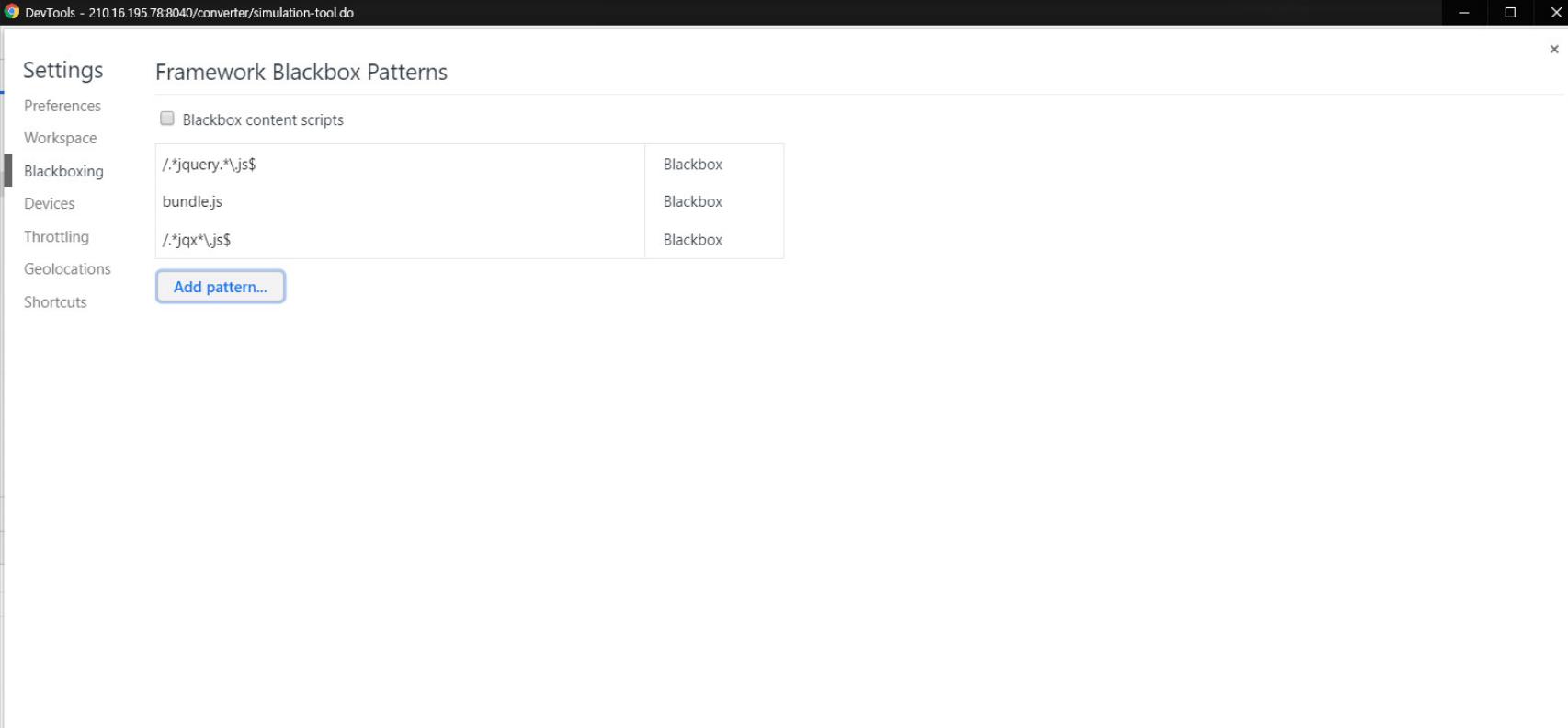
XHR/fetch Breakpoints

DOM Breakpoints

Global Listeners

Event Listener Breakpoints

# Chrome Dev Tools > Blackboxing



The screenshot shows the 'Blackboxing' section of the Chrome DevTools settings. The left sidebar has 'Blackboxing' selected. The main area is titled 'Framework Blackbox Patterns' and contains a table with three rows:

Pattern	Action
./jquery.*\.js\$	Blackbox
bundle.js	Blackbox
./jqx*.js\$	Blackbox

An 'Add pattern...' button is located at the bottom left of the table.



# Chrome Dev Tools > Testing



END

