

Javascript

- Javascript
- High level, dynamic ,untyped ,interpret
- Platform independence ဖြစ်ပြီး Interactive websites တစ်ခုတည်ဆောက်ရာတွင် အသုံးပြုသည်။
 - Validation
 - Drop-down menus
 - Date time display
 - Dialog box ,pop-up window ,alert box အစရိဘာလှည်တို့ ဖော်ပြရာတွင် အဓိက အသုံးပြုသည်။

- Data types
 - Numbers
 - String
 - Boolean
- ```
> typeof("hello")
< "string"

```
- ```
> typeof(4);
< "number"
-----
```
- ```
> typeof(4.0f);
✖ Uncaught SyntaxError: Invalid or unexpected token

```
- ```
> typeof(4.0);
< "number"
-----
```
- ```
> typeof(true)
< "boolean"

```
- ```
>
```

- Variable

- Var Variable ကြညာရပ် တွင် var ကိုသာ အသုံးပြုသည်။

The screenshot shows the Chrome DevTools interface with the Sources tab selected. In the left sidebar, 'index.js' is listed under Snippets. The main pane displays the following code:

```
3
4 var name=prompt("Please type your name !");
5
```

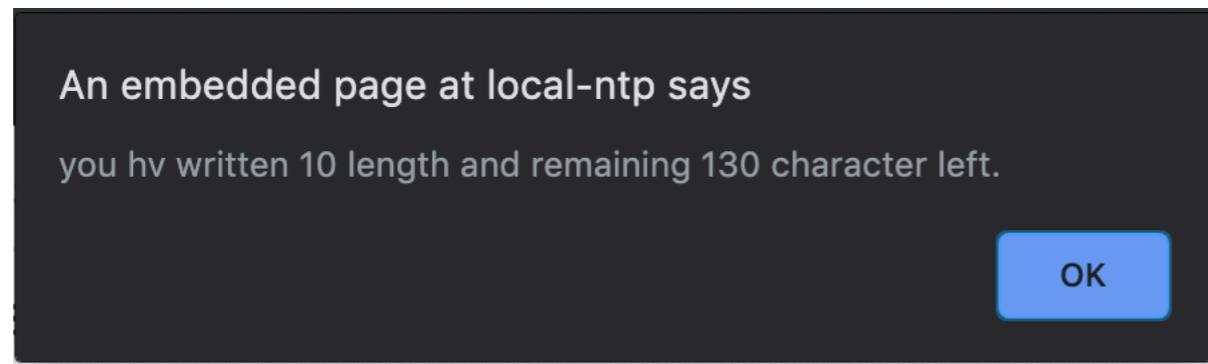
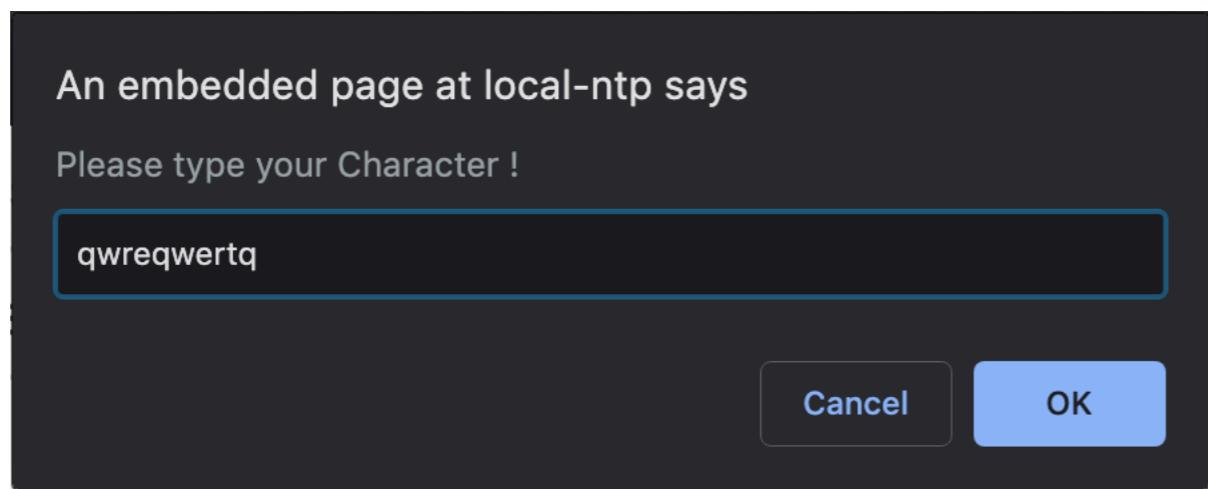
The cursor is positioned at the end of line 4. A callout arrow points from the text 'Variable ကြညာရပ် တွင် var ကိုသာ အသုံးပြုသည်။' to the word 'var' in the code.

Below the code, the status bar shows 'Line 1, Column 1'. The bottom pane shows the JavaScript console output:

```
> var name=prompt("type of your name");
< undefined
< undefined
> name
< "mg mg"
> |
```

The word 'index.' is visible on the right side of the console output.

- String Concatenation ,Length and retrieving character
- ```
var tweet=prompt("Please type your Character !");
var tweet_length=tweet.length;
alert("you hv written "+tweet_length+" length and remaining "
+(140-tweet_length)+" character left.");
```



- Arithmetic Operator and Modulo

- + - \* / %

- ```
var dtp=prompt("Please type your myanmar date of birth !");
var humanAge=((dogAge-2)*4)+21;

alert("Your human age is"+humanAge);
```

humanAge

29

- Function ,parameter and return

- ```
function sayHello(){
 console.log("Hello,");
}
function sayName(name){
 console.log("Hello,"+name);
}
function getName(name){
 return name;
}

sayHello();
sayName("ko ko");
alert(getName("maung maung"));|
```

Hello,  
Hello,ko ko  
undefined

An embedded page at local-ntp says  
maung maung

OK

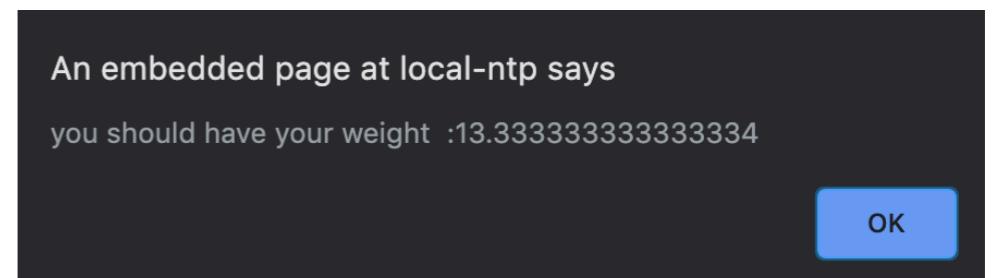
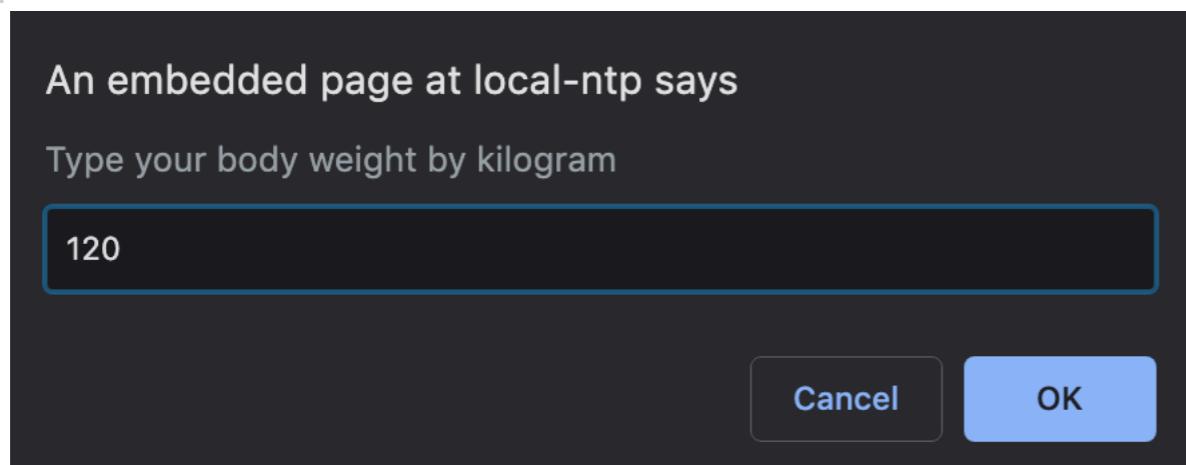
- Eg. Body Mass Index program.

- ```
var weight=prompt("Type your body weight by kilogram");
var height=prompt("Type your height by meter");
var bmi=bmiCal(weight,height);

function bmiCal(weight,height){

    return weight/Math.pow(height,2);
}

alert("you should have your weight :"+bmi);
```



- random() and floor()
- Random ၏ range သည် 0 မှ 1 ကြားတွင်သာ ရှိသည်။
- Floor သည် အသမ ကိန်းများ အား ကိန်းပြည့် အဖြစ် ပြောင်းပေးနိုင်သည်။

-

```
var rand=Math.random();

// 1 to 6 but never reach on 6.
var range=(rand*6)+1;

// to get without decimal number.
var deci=Math.floor(range);

console.log(deci);
```

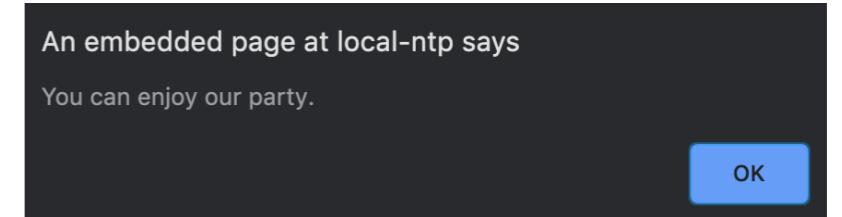
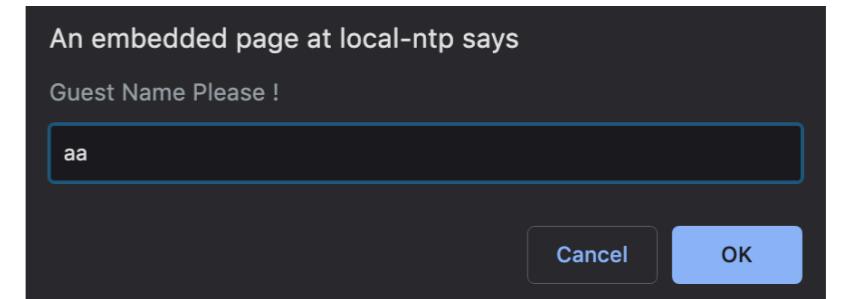
- Comparator
- === is equal to (data types තුළතු එහි තෝරාවයි)
- == is equal to (data types තුළතු වෙත්තෙක් එහි)
- != is not equal to
- > < greater and less than
- >= , <= greater than and equal to ,less than and equal to

- Array
- .push() သည် array ၏ data နောက်ဆုံးတွင် ထပ် ထည့်ရန် ။
- .pop() သည် array ၏ နောက်ဆုံးမှ data အား ဖယ်ဝှက်ယူရန်။
- ```
var array=["a","b","c",2];
console.log(array);
console.log(array.length);
console.log(array[2]);
array.includes(2);
|
```

```
var guest_list=["aa","bb","cc","dd"];
var guest=prompt("Guest Name Please !");

if(guest_list.includes(guest)){
 alert("You can enjoy our party.");
}else{
 alert("You are not allowed to enter.");
}
```

▶ (4) ["a", "b", "c", 2]  
4  
c  
true



## If ( ) { } else { }

```
var your_name=prompt("Your Name please !");
var partner_name=prompt("Partner Name please !");

var love_score=Math.random()*100;
love_score=Math.floor(love_score)+1; // 1-100;

function calculate(love_score){

 if(love_score==100){
 alert("Conguration ,you and your partener is like romeo and juliet and with the full 100 percent.");
 }else if(love_score>70 && love_score<1000){
 alert("Conguration,both love each other with the "+love_score+"%");
 }else{
 alert("You may still need to know about your partner with the "+love_score+"%");
 }
}

calculate(love_score);
```

## While () {}

```
var count=prompt("How many time ,do you want to say goodbye !");

var i=1;

while(i<=count){
 console.log("goodbye "+i);
 i++;
}
```

4

goodbye 1

goodbye 2

goodbye 3

goodbye 4

**For ( var i=0; i < 5 ;i++){ }**

```
var count=prompt("How many time ,do you want to say goodbye !");

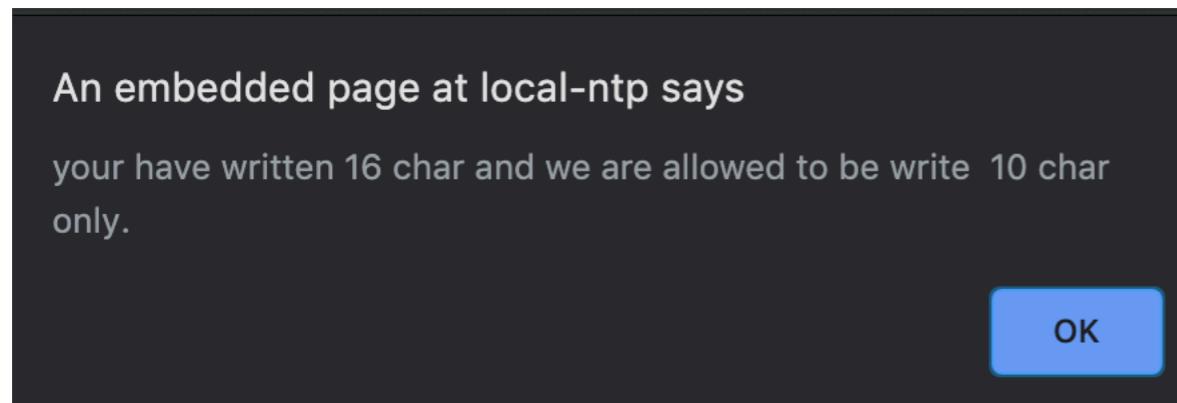
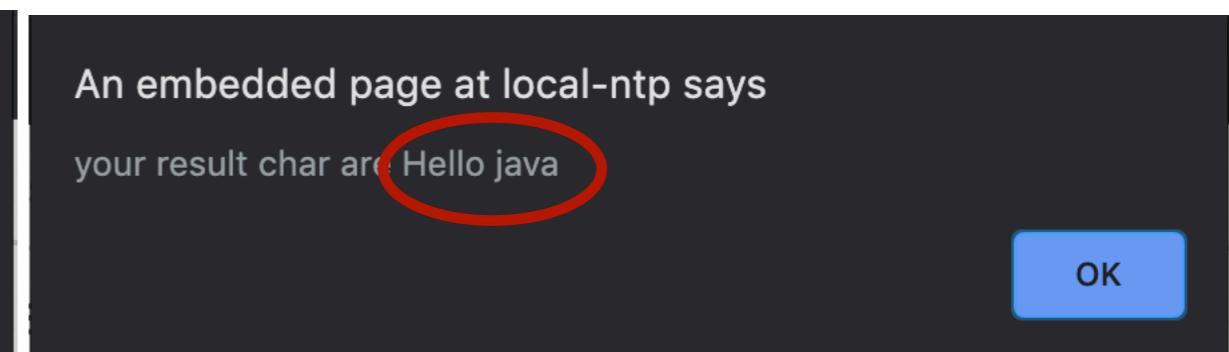
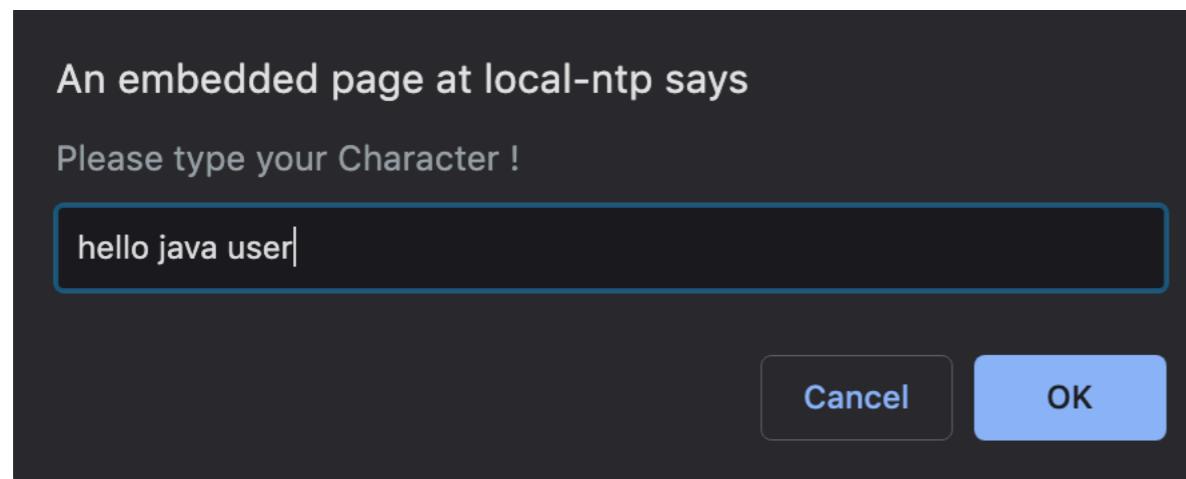
for(var i=1;i<=count;i++){
 console.log("goodbye "+i);
}
```

5  
goodbye 1  
goodbye 2  
goodbye 3  
goodbye 4  
goodbye 5  
undefined

- **Slice( ) and toUpperCase()**

- ```
var tweet=prompt("Please type your Character !");
var tweet_length=tweet.length;
var first_char=tweet.slice(0,1).toUpperCase();

var slice_Char=tweet.slice(1,10);
alert("your have written "+tweet_length+" char and we are allowed to
alert("your result char are "+first_char+slice_Char);
```



- Variable (let(modern), var (old))

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>variable</title>
5  </head>
6  <body>
7
8  <script >
9      //variables
10     //let name="ko ko",age=10,dob=20/05/1999;
11     var name1='su su',age=2,dob=22/03/2014 ;//using var
12     alert(name1)
13     //alert(name1)
14 </script>
15 </body>
16 </html>
```

- Constant

```
<script >
    const name='koko'
    alert(name)
</script>
```

- **Inline** →

```
<head>
  <meta charset="utf-8">
  <title>My Website</title>
</head>
<body onload="alert('hello');">
  <h1>Hello</h1>
</body>
```

- **Internal** →

```
<head>
  <meta charset="utf-8">
  <title>My Website</title>
</head>
<body>
  <h1>Hello</h1>

  <script type="text/javascript">
    alert("hello");
  </script>
</body>
```

- **External** →

```
<html lang="en" dir="ltr">
  <head>
    <meta charset="utf-8">
    <title>My Website</title>
  </head>
  <body>
    <h1>Hello</h1>
    <script src="scriptFile/index.js"></script>
  </body>
</html>
|
```

• DOM (Document Object Model)

← → ⌂ ⌂ File | /Volumes/MinKhant/Web/web%20Lesson/web%20lesson

Enter Cancel

- [Google Search](#)
- Second
- Third

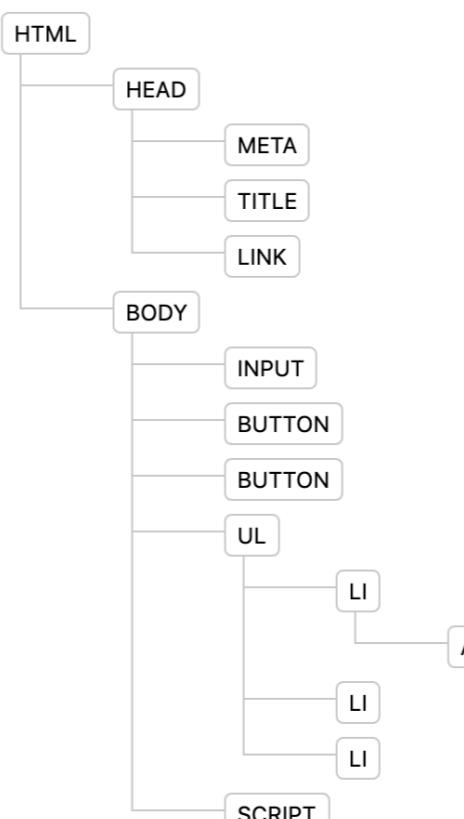
Elements Console S

top

```
> document.firstChild;
<-  <html lang="en" dir="ltr">
    ><head>...</head>
    ><body>...</body>
  </html>

> document.firstChild.firs
<- ▼<head>
    <meta charset="utf-8">
    <title>My Website</title>
    <link rel="stylesheet" href="style.css">
  </head>

> document.firstChild.last
<- ▼<body>
    <input type="checkbox" name="checkbox">
    <button class="btn active">Active</button>
    <button class="btn">Cancel</button>
  <ul>...</ul>
  <script src="scriptFile/index.js"></script>
</body>
```



```

graph TD
    HTML[HTML] --> HEAD[HEAD]
    HEAD --> META[META]
    HEAD --> TITLE[TITLE]
    HEAD --> LINK[LINK]
    HTML --> BODY[BODY]
    BODY --> INPUT[INPUT]
    BODY --> BUTTON1[BUTTON]
    BODY --> BUTTON2[BUTTON]
    BODY --> UL[UL]
    UL --> LI1[LI]
    UL --> LI2[LI]
    UL --> LI3[LI]
    BODY --> SCRIPT[SCRIPT]
    LI1 --> A[A]
  
```

HTML Tree Generator

Node Details

Tag Name:

Node ID:

of Children:

Brought to you by [Joel Saupe](#)

• QuerySelector

• Bye Bye

Enter

- [Google Search](#)
- Second
- Third

The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The console output is as follows:

```
> var head=document.firstChild.lastElementChild.firstElementChild;
< undefined
> head.innerHTML="Bye Bye";
< "Bye Bye"
> head.style.color="blue";
< "blue"
> document.querySelector("h1").style="purple";
< "purple"
>
```

Good Bye

Enter

- [Google Search](#)
- Second
- Third

The screenshot shows the Chrome DevTools interface with the 'Console' tab selected. The console output is as follows:

```
> document.querySelector("input").style.color="green";
< "green"
> document.querySelector("button").style.color="green";
< "green"
> document.querySelector(".list");
< ▼<li class="list">
  <a href="https://www.google.com">Google Search</a>
</li>
> document.querySelectorAll(".list")[2].style.color="red";
< "red"
> document.querySelector(".list a").style.backgroundColor="yellow";
< "yellow"
>
```

- Add , remove and toggle

- Hello

Java

-
- [Google Search](#)
- Second
- Third

The screenshot shows the Chrome DevTools interface. The top navigation bar has tabs for Elements, Console, Sources (which is selected), and Network. Below the tabs is a sidebar with a 'Java' section containing a list of items. The main content area shows a file named 'index.js' with the following code:

```
1 document.querySelector("button").classList.add("invisible");
2 document.querySelector("button").classList.remove("invisible");
3 document.querySelector("button").classList.toggle("invisible");
4 document.querySelector("h2").classList.toggle("huge");
5
```

The status bar at the bottom indicates 'Line 4, Column 55'. Below the code editor is a toolbar with various icons. The 'Scope' tab is selected in the toolbar.

The 'Console' tab is selected in the bottom navigation bar. The console output shows the execution of the code:

```
< undefined
> document.querySelector("h2").classList.remove("huge");
< undefined
> document.querySelector("h2").classList.toggle("huge");
< true
> document.querySelector("h2").classList.toggle("huge");
< false
```

The 'Default levels' dropdown and a settings gear icon are also visible at the bottom right of the console area.

- Text context & Attribute

- **Hello**

Java

- Enter
- [Google Search](#)
- Second
- Third

The screenshot shows the Chrome DevTools interface with the 'Sources' tab selected. The left sidebar displays a file tree with 'index.js*' expanded, showing lines 1 and 2. The main content area shows the code in 'index.js' at Line 1, Column 1:

```
> document.querySelector("h1").innerHTML;
< "strong>Hello</strong>"
> document.querySelector("h1").textContent;
< "Hello"
> document.querySelector("h1").innerHTML=<em>Hello</em>;
< "<em>Hello</em>"
> document.querySelector("a").attributes;
< ▶ NamedNodeMap {0: href, href: href, length: 1}
> document.querySelector("a").getAttribute("href");
< "https://www.google.com"
> document.querySelector("a").setAttribute("href","https://www.bing.com");
< undefined
```



Draws



Player 1

Player 2

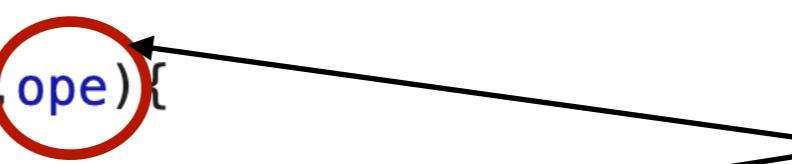


- Function

- Function အား argument အနေဖြင့် ပြန်ယူနိုင်သည်။

- ```
function plus(num1,num2){
 return num1+num2;
}
function minus(num1,num2){
 return num1+num2;
}
function operator(num1,num2,ope){
 return ope(num1,num2);
}
```

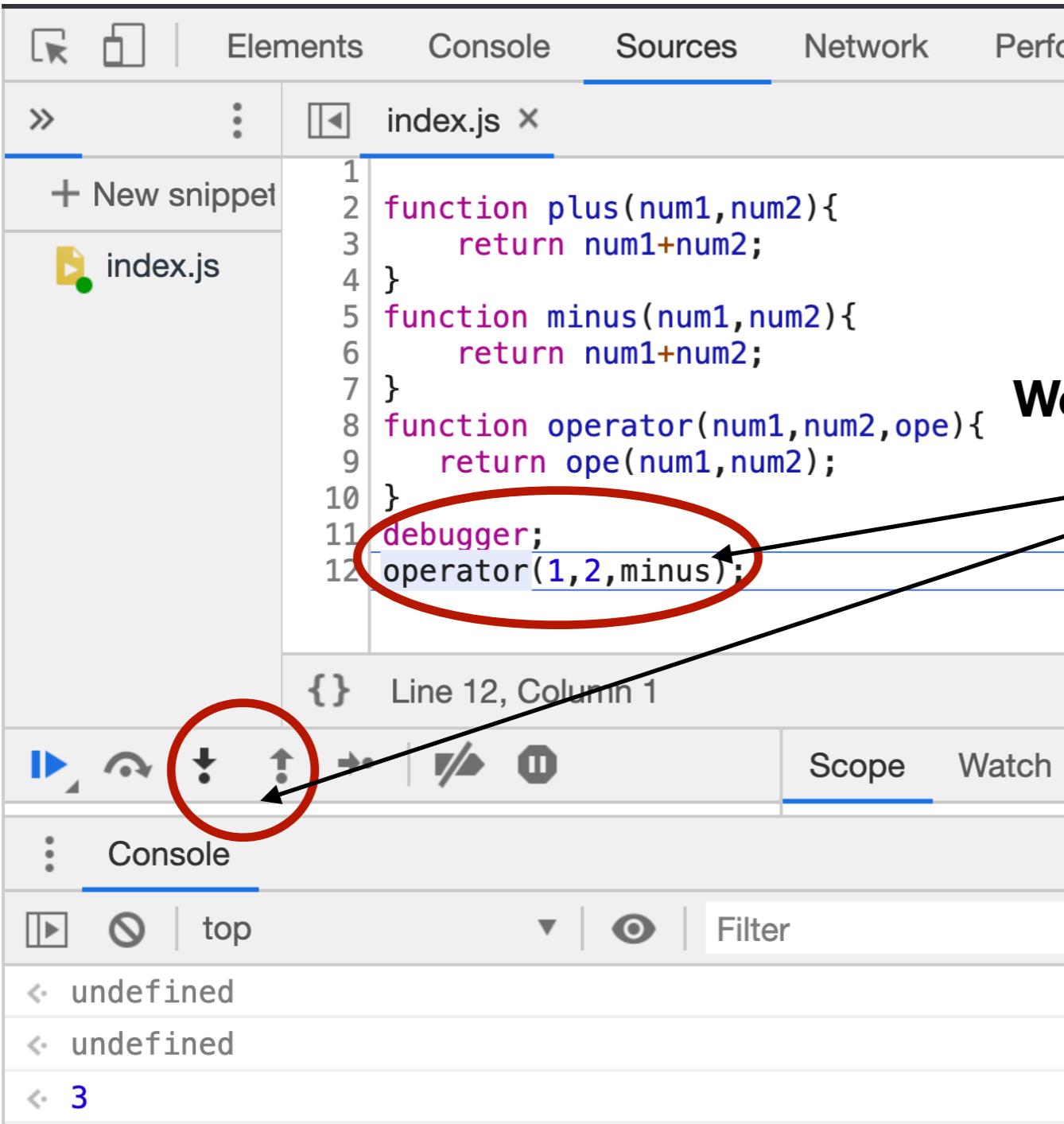
`operator(1,2,minus);`



Function Argument



- Debugger

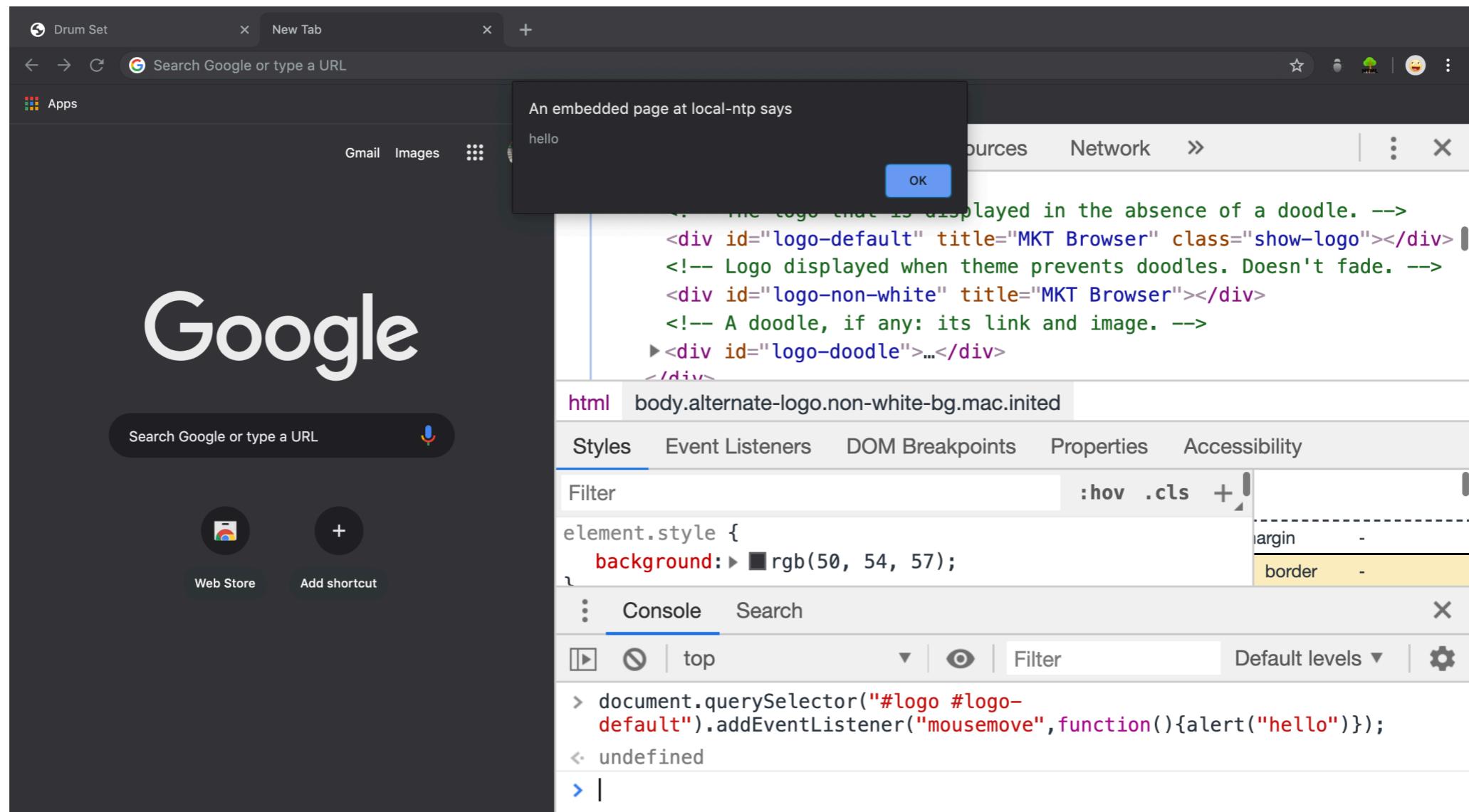
- A screenshot of the Chrome DevTools Sources tab for a file named "index.js". The code contains several functions: plus, minus, and operator. A "debugger;" statement is located at line 11. The code is as follows:

```
1 function plus(num1,num2){
2 return num1+num2;
3 }
4 function minus(num1,num2){
5 return num1+num2;
6 }
7 function operator(num1,num2,ope){
8 return ope(num1,num2);
9 }
10
11 debugger;
12 operator(1,2,minus);
```

A red oval highlights the "debugger;" statement at line 11. A black arrow points from this oval to the step-over icon (represented by a small circle with a dot) in the toolbar below the code editor. Another black arrow points from the step-over icon to the text "We can go to check step by step" on the right. The status bar at the bottom of the code editor shows "Line 12, Column 1".

We can go to check step by step

- `addEventListener("name",function);`
- `name = click,mousemove,mouseover,...`
- ```
> document.querySelector("#logo #logo-
        default").addEventListener("mousemove",function(){alert("hello")});
< undefined
```



Not logged in Talk Contributions Create account Log in

Article [Talk](#)

MKT Script

From Wikipedia, the free encyclopedia

Not to be confused with [Java \(programming language\)](#), [Java \(software platform\)](#), or [Javanese script](#).
For the uses of [JavaScript](#) on Wikipedia, see [Wikipedia:Javascript](#).

JavaScript

Paradigm	Multi-paradigm: event-driven, ^[1] functional, imperative, object-oriented (prototype-based)
Designed by	Brendan Eich
Developer	Netscape Communications

JavaScript ([/dʒɑːvə skrɪpt/](#)),^[8] often abbreviated as JS, is a high-level, interpreted programming language that conforms to the ECMAScript specification. JavaScript has curly-bracket syntax, dynamic typing, prototype-based object-orientation, and first-class functions.

Alongside [HTML](#) and [CSS](#), JavaScript is one of the core technologies of the World Wide Web.^[9] JavaScript enables interactive web pages and is an essential part of web applications. The vast majority of websites use it,^[10] and major web browsers have a dedicated [JavaScript engine](#) to execute it.

As a multi-paradigm language, JavaScript supports [event-driven](#), [functional](#), and [imperative](#) (including [object-oriented](#) and [prototype-based](#)) [programming styles](#). It has [APIs](#) for working with text, [arrays](#), dates, [regular expressions](#), and the [DOM](#), but the language itself does not include any [I/O](#) such as [networking](#), [storage](#), or [graphics](#) facilities. It relies upon the host environment in which it is embedded to provide those features.

Elements Console Sources Network Performance Memory Application Security Audits × 1 ⋮ X

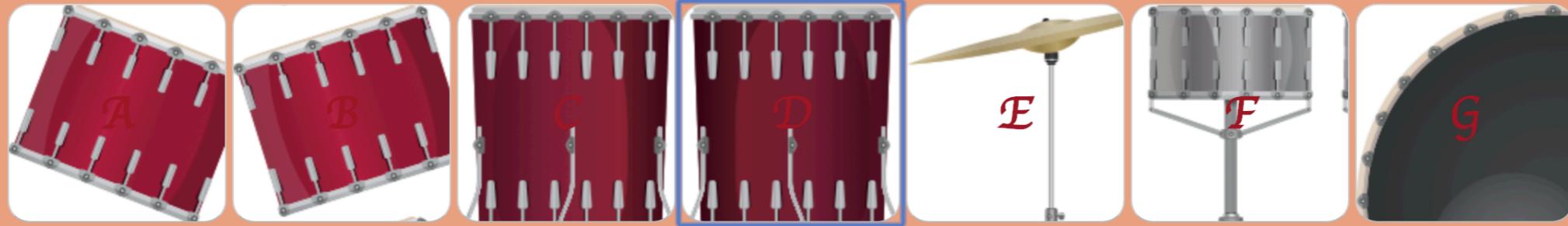
top Filter Default levels ▾

JavaScript

```
< "JavaScript"
> document.querySelector("#firstHeading").innerHTML="MKT Script";
< "MKT Script"
> document.querySelector("#firstHeading").addEventListener("click",function(ev){console.log(ev)});
✖ Uncaught SyntaxError: missing ) after argument list VM738:1
> document.querySelector("#firstHeading").addEventListener("click",function(ev){console.log(ev)});
< undefined
▶ MouseEvent {isTrusted: true, screenX: 289, screenY: 197, clientX: 289, clientY: 118, ...} VM748:1
▶ MouseEvent {isTrusted: true, screenX: 289, screenY: 197, clientX: 289, clientY: 118, ...} VM748:1
```



Play Drum



Created by MKT (JDC)