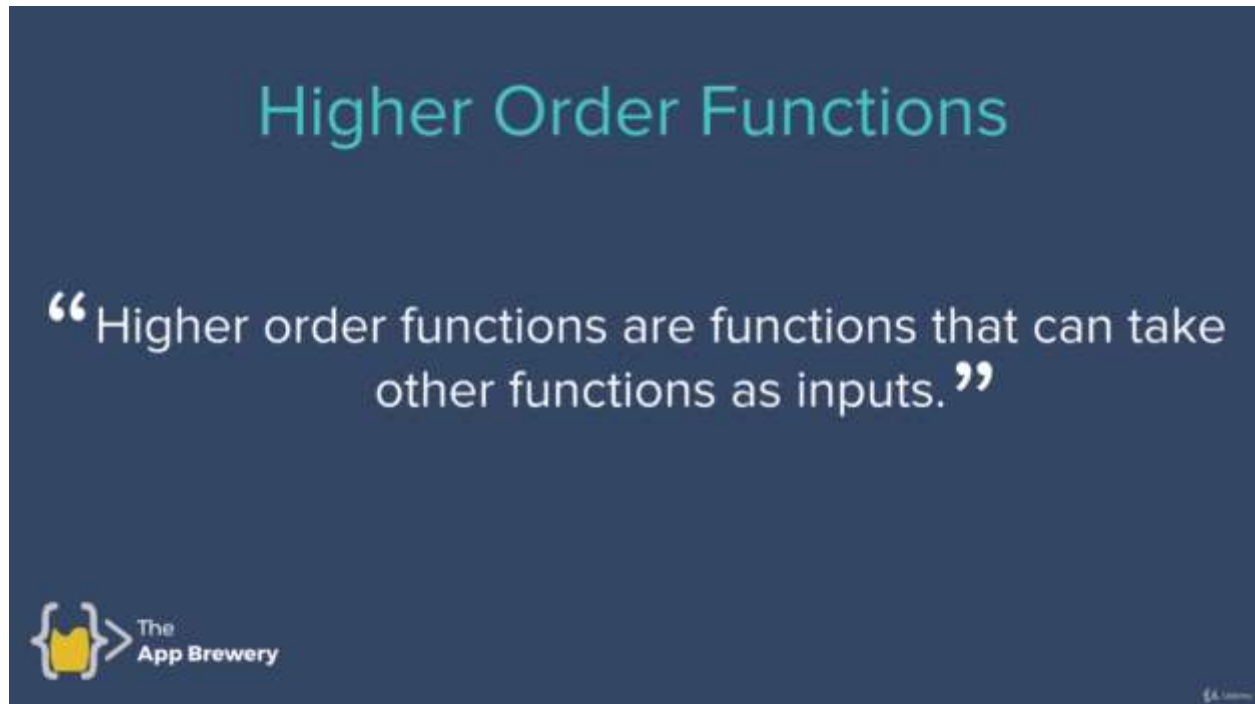


Higher Order Functions and Passing Functions as Arguments



And this feature is available in number of modern languages like **JavaScript, Java, Ruby, Python, Haskell, C++, PHP, C#, Go, Swift etc.** It's not universal means not available in all programming languages but in a number of programming languages on which **JavaScript** is rely on.

Example:

```
function add(num1, num2){  
  return num1+num2;  
}  
  
function subtract(num1, num2){  
  return num1-num2;  
}  
  
function multiply(num1, num2){  
  return num1*num2;  
}  
  
function division(num1, num2){
```

```
return num1/num2;
```

```
}
```

```
function calculator(num1, num2, operator){
```

```
    return operator(num1, num2);
```

```
}
```

//Let's add debugger and see every step of how code runs

```
debugger(3,4,multiply);
```

The screenshot shows a web browser displaying the Wikipedia article for "JavaScript". The article text states: "Not to be confused with Java (programming language), Javaneese script, or ECMAScript. *js* redirects here. For the Microsoft dialect used in Internet Explorer, see JScript. For the uses of JavaScript on Wikipedia, see Wikipedia:JavaScript. JavaScript (ⁱ/dʒəˈvæskript/)^[1] often abbreviated **JS**, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS^[1] As of 2022, 98% of websites use JavaScript on the client side for webpage behavior.^[12] All major web browsers have a dedicated JavaScript engine to execute the code on users' devices. JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard^[14] it has dynamic typing, prototype-based object-orientation, and first-class functions. It is multi-paradigm, supporting event-driven, functional, and imperative programming styles. It has application

The Chrome DevTools console shows the following code being executed:

```
> $0
< <h1 id="firstHeading" class="firstHeading mw-first-heading">
  ::before
    "JavaScript"
  </h1>
> $0.innerHTML = "Muhammad Ahmad";
< 'Muhammad Ahmad'
> $0.innerHTML = "AhmadScript";
< 'AhmadScript'
> |
```



WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute

Help
Learn to edit
Community portal
Recent changes
Upload file

Tools

What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Not logged in - Talk - Contributions - Create account - Log in

Article - Talk

Read - Edit - View history

Search Wikipedia

JavaScript



From Wikipedia, the free encyclopedia

Not to be confused with Java (programming language), Javanese script, or ECMAScript. ".js" redirects here. For the Microsoft dialect used in Internet Explorer, see JScript. For the uses of JavaScript on Wikipedia, see Wikipedia:JavaScript.

JavaScript (/ˈdʒɑːvəskript/^[10]), often abbreviated **JS**, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS.^[11] As of 2022, 98% of websites use JavaScript on the client side for webpage behavior,^[12] often incorporating third-party libraries.^[13] All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard.^[14] It has dynamic typing, prototype-based

JavaScript

```
<html>
<head>
  <title>
    <form method="post" action="#" id="formvalue" onkeyup="
      drawChart()" />
    </form>
  </title>
</head>

<script type="text/javascript" src="https://www.google.com/jsapi"></
script>
<script type="text/javascript">

var bid = 43;
var ask = 31;

google.load("visualization", "1", {packages:["corechart"]});
google.setOnLoadCallback(drawChart);
function drawChart() {
  var data = google.visualization.arrayToDataTable([
    ['Price', 'Quantity'],
    ['Value #1', bid],
    ['Value #2', ask],
  ]);
```

Screenshot of JavaScript source code with HTML

Paradigm

Multi-paradigm: event-driven,
functional, imperative, procedural,

```
> $0;
< ▶<h1 id="firstHeading" class="firstHeading mw-first-heading">...</h1>
```

```
> $0.addEventListener("click", function(){
  console.log("I got clicked!");
});
```

```
< undefined
```

```
I got clicked!
```

VM920:2

```
>
```

```
$0.addEventListener("click", function() {
  console.log("I got clicked");
});
```

```
$0.addEventListener("click", respondToClick);

function respondToClick() {
  console.log("I got clicked");
}
```

First input is input1 and second input is the function that we should be called once h1 detect to click event.

```
$0.addEventListener(input1, respondToClick);

function respondToClick() {
  console.log("I got clicked");
}
```

Input1 specifies what event it should specify and input2 specifies that what should it do that's it's once detected. And it's different from above functions.

```
$0.addEventListener(input1, input2);

function respondToClick() {
  console.log("I got clicked");
}
```



Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate
Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file
Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Article Talk

JavaScript

From Wikipedia, the free encyclopedia

Not to be confused with **Java** (programming language), *Japanese script*, or *ECMAScript*.
"js" redirects here. For the Microsoft dialect used in Internet Explorer, see *JScript*.
For the uses of JavaScript on Wikipedia, see *Wikipedia:JavaScript*.

JavaScript (/dʒəˈvæskript/^[a]) often abbreviated **JS**, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS.^[b] As of 2022, 98% of websites use JavaScript on the client side for webpage behavior.^[c] It often incorporates third-party libraries.^[b] All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.

JavaScript is a high-level, often just-in-time compiled language that conforms to the ECMAScript standard.^[d] It has dynamic typing, prototype-based

JavaScript



Screenshot of JavaScript source code with HTML

Paradigm Multi-paradigm: event-driven, functional, imperative, procedural

```
> function calculator(num1, num2){
    return num1+num2;
}
calculator(2,3);
< 5

> function calculator(num1, num2){
    return num1*num2;
}
calculator(2,3);
< 6

> |

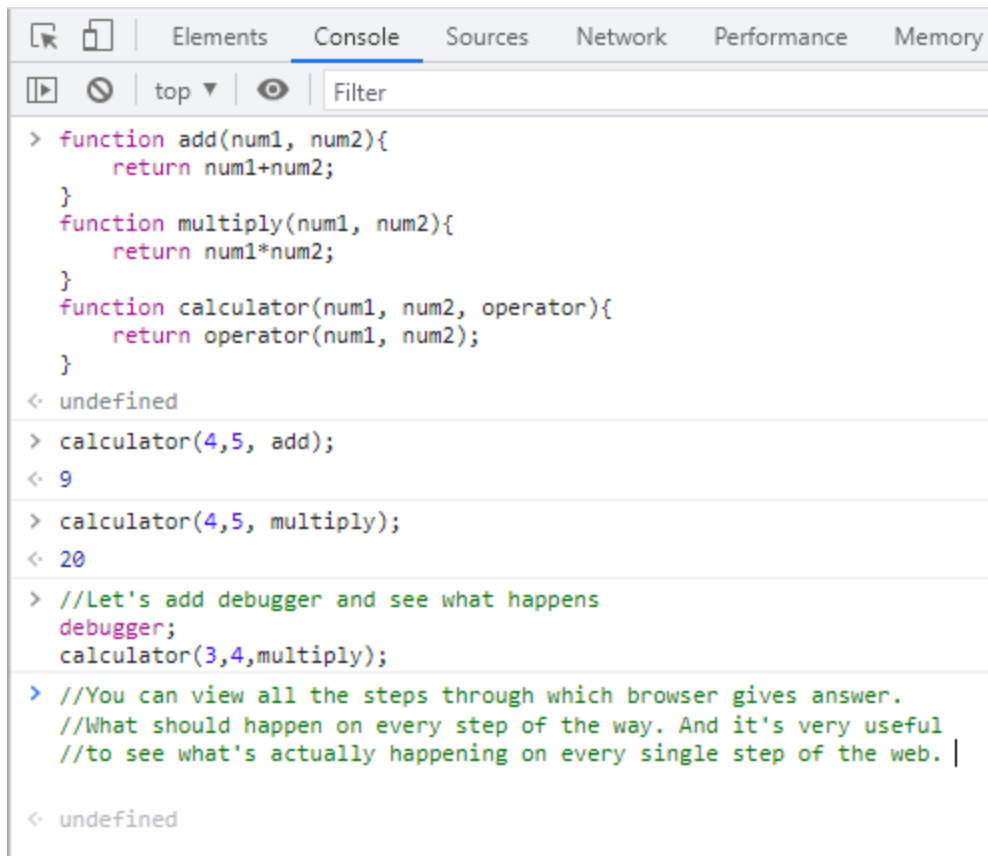
> function add(num1, num2){
    return num1+num2;
}
function multiply(num1, num2){
    return num1*num2;
}
function calculator(num1, num2, operator){
    return operator(num1, num2);
}
< undefined

> calculator(4,5, add);
< 9

> calculator(4,5, multiply);
< 20

>
```

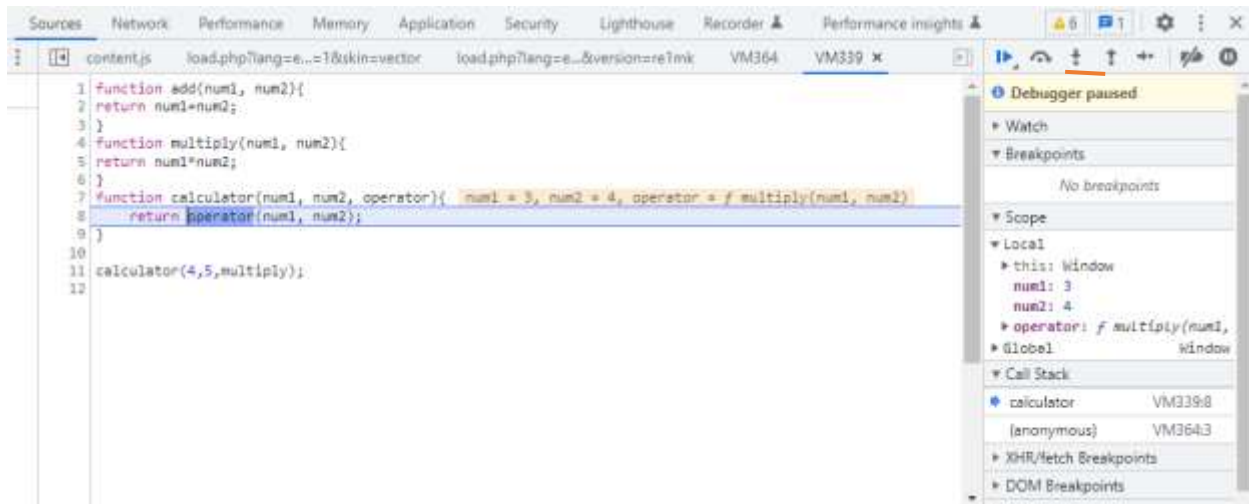
```
function calculator(num1, num2){
    return num1+num2;
}
calculator(1,1);
>
> function calculator(num1, num2){
    return num1*num2;
}
calculator(2,3);
>
>
```



The screenshot shows the Chrome DevTools Console with the following content:

- Navigation bar: Elements, Console (selected), Sources, Network, Performance, Memory.
- Filter: top, eye icon, Filter.
- Code input area:

```
> function add(num1, num2){
  return num1+num2;
}
function multiply(num1, num2){
  return num1*num2;
}
function calculator(num1, num2, operator){
  return operator(num1, num2);
}
< undefined
> calculator(4,5, add);
< 9
> calculator(4,5, multiply);
< 20
> //Let's add debugger and see what happens
  debugger;
  calculator(3,4,multiply);
> //You can view all the steps through which browser gives answer.
  //What should happen on every step of the way. And it's very useful
  //to see what's actually happening on every single step of the web. |
< undefined
```



The screenshot shows the Chrome DevTools Sources panel with the following content:

- Navigation bar: Sources (selected), Network, Performance, Memory, Application, Security, Lighthouse, Recorder, Performance insights.
- File list: content.js, load.php?lang=e...=1&skin=vector, load.php?lang=e...&version=re1mk, VM364, VM339 (selected).
- Code editor (content.js):

```
1 function add(num1, num2){
2   return num1+num2;
3 }
4 function multiply(num1, num2){
5   return num1*num2;
6 }
7 function calculator(num1, num2, operator){
8   return operator(num1, num2);
9 }
10
11 calculator(4,5,multiply);
12
```
- Debugger panel (right):
 - Debugger paused
 - Watch
 - Breakpoints: No breakpoints
 - Scope
 - Local
 - this: window
 - num1: 3
 - num2: 4
 - operator: f multiply(num1,
 - Global: window
 - Call Stack
 - calculator VM339:8
 - (anonymous) VM364:3
 - XHR/fetch Breakpoints
 - DOM Breakpoints

