JavaScript

Objective

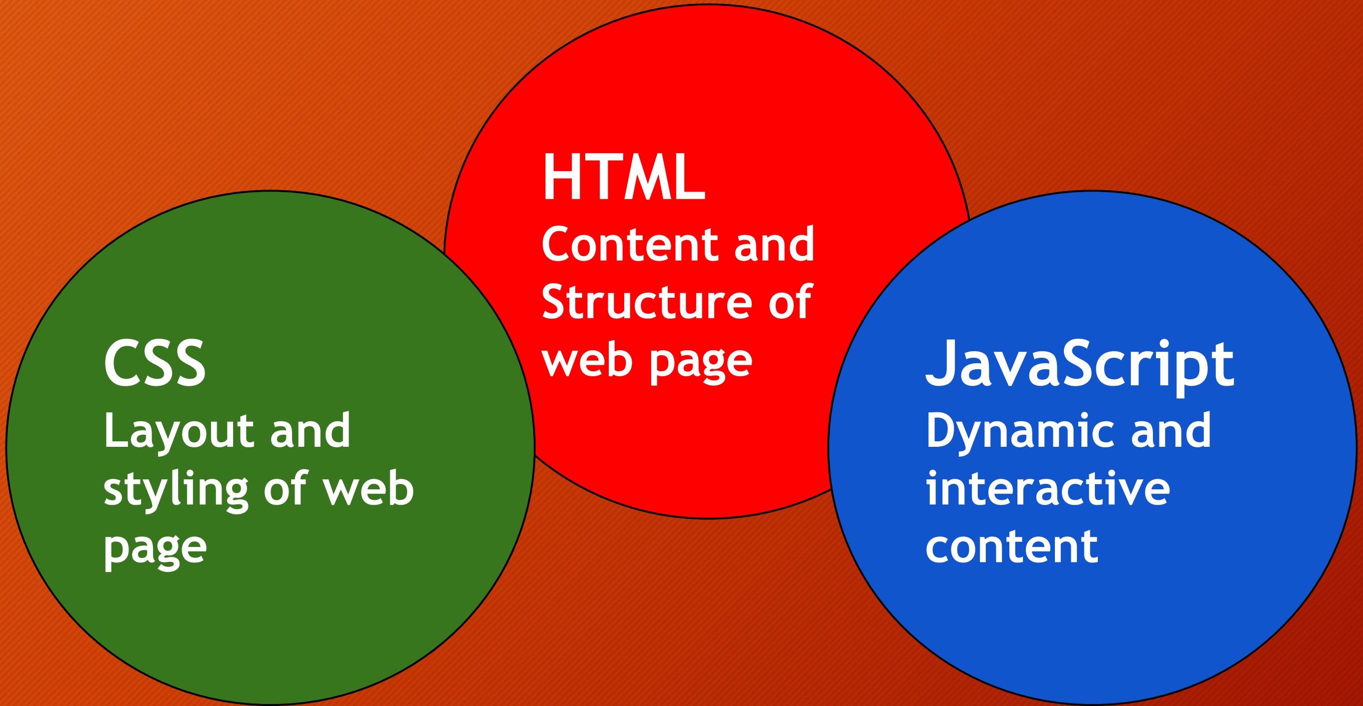
- To equip students with the understanding of basics of JavaScript and its features.
- Students will learn to program the behavior of web pages using JavaScript.
- To use JavaScript to provide dynamic and interactive content on webpages.

Outcomes

- At the end of this presentation, students are expected to learn
  - to create dynamic and interactive content using JavaScript.
  - how to use JavaScript with other languages, specifically html and css3.

Why JavaScript?

- HTML is used to define the content and structure of web pages.
- CSS specifies the layout and styling of the web page.
- JavaScript provides the dynamic and interactive content on web pages.



HTML
Content and Structure of web page

CSS
Layout and styling of web page

JavaScript
Dynamic and interactive content

Introduction

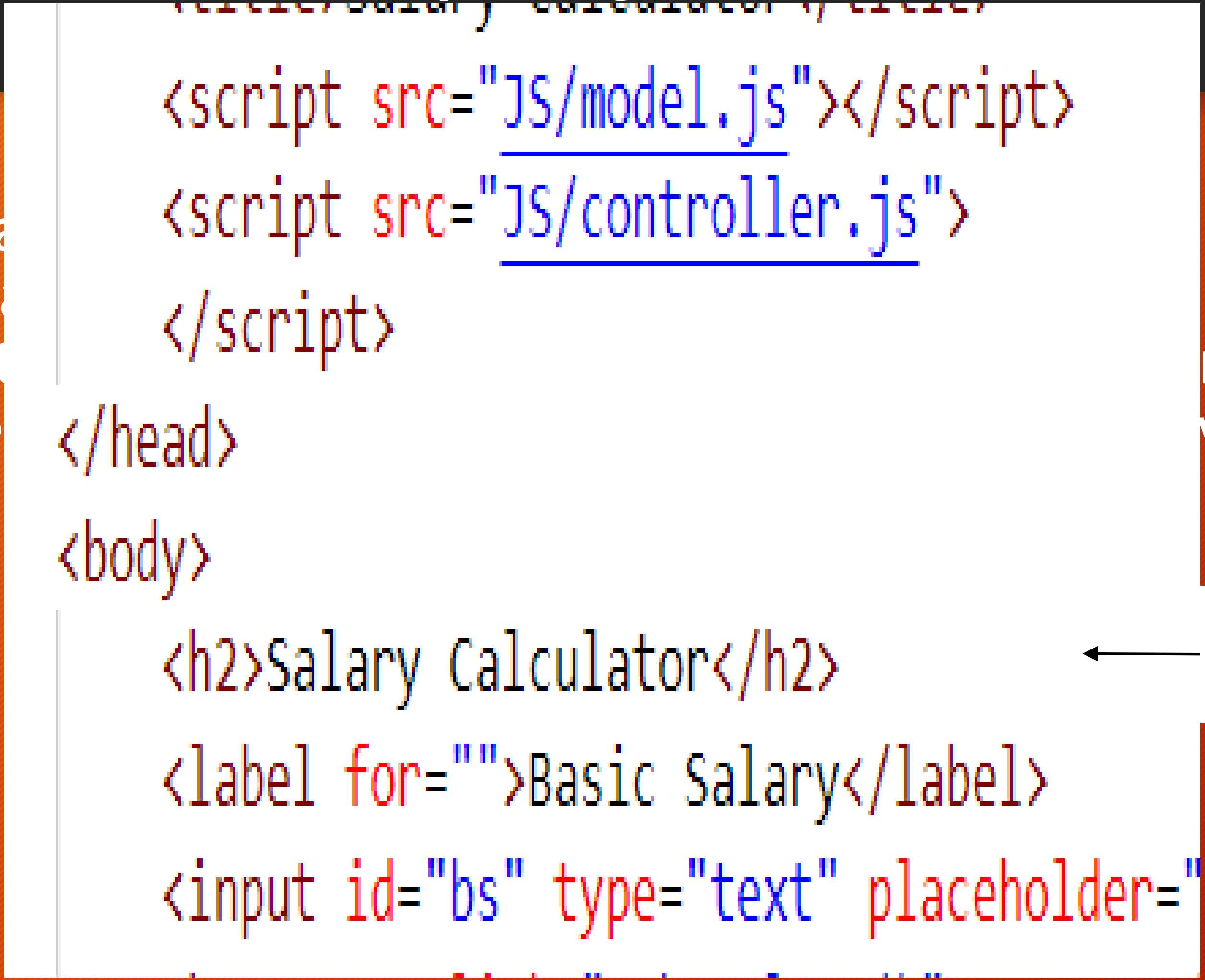
- ❖ JavaScript is a scripting language designed specifically for use in web browsers to make websites more dynamic.
- ❖ JavaScript is a client side language which means all the action occurs on the client's (reader's) side of things.
- ❖ JavaScript can change HTML content.
- ❖ It can change HTML attribute values.
- ❖ It can change HTML Styles.
- ❖ JavaScript can show/hide HTML elements.

Features of JavaScript

- JavaScript is a client side scripting language.
- It is a loosely typed language.
- It is a function oriented language.
- JavaScript supports DOM manipulation.

Client Side Scripting

- Java
- If Ja then the script is sent nt.
- The with HTML code.

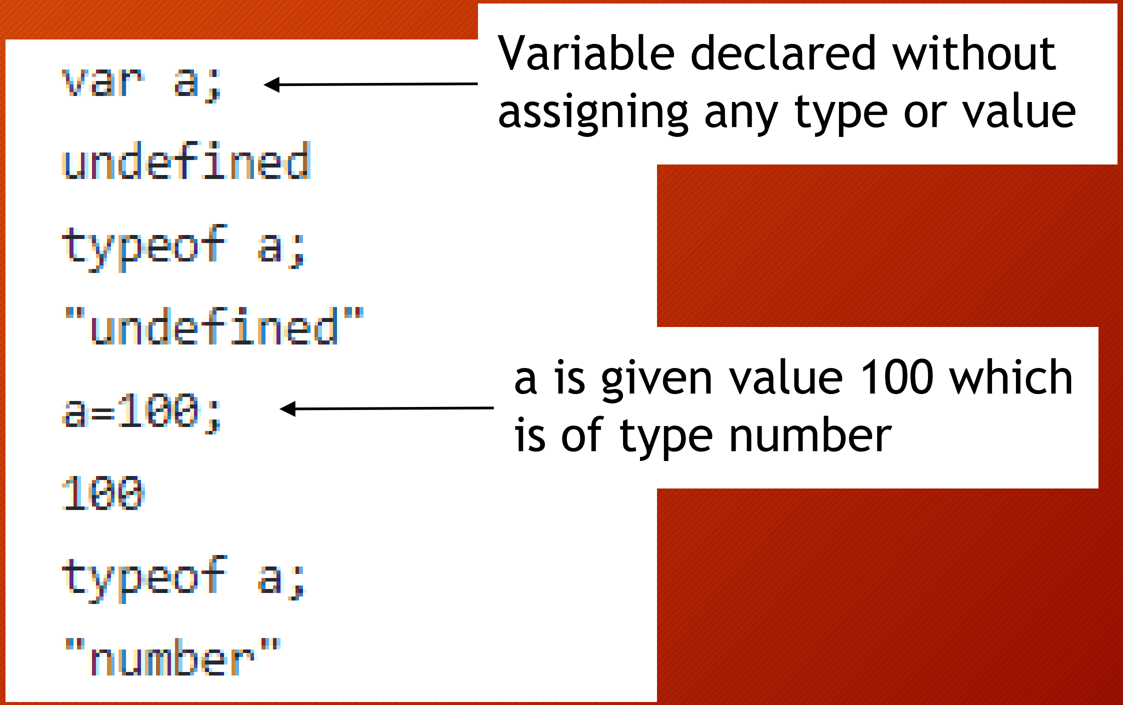


```
    <title>Salary Calculator</title>
    <script src="JS/model.js"></script>
    <script src="JS/controller.js">
    </script>
</head>
<body>
    <h2>Salary Calculator</h2>
    <label for="">Basic Salary</label>
    <input id="bs" type="text" placeholder="
```

Embedding Javascript files within HTML content.

Loosely Typed Language

- If no type or value is assigned to a variable, then it is considered undefined.



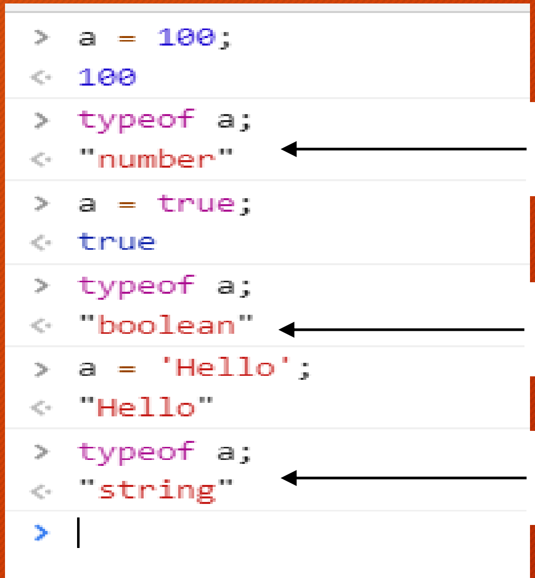
```
var a;
undefined
typeof a;
"undefined"
a=100;
100
typeof a;
"number"
```

Variable declared without assigning any type or value

a is given value 100 which is of type number

Loosely Typed Language

- One variable can have multiple types depending on the value assigned to it.



```
> a = 100;
<· 100
> typeof a;
<· "number"
> a = true;
<· true
> typeof a;
<· "boolean"
> a = 'Hello';
<· "Hello"
> typeof a;
<· "string"
> |
```

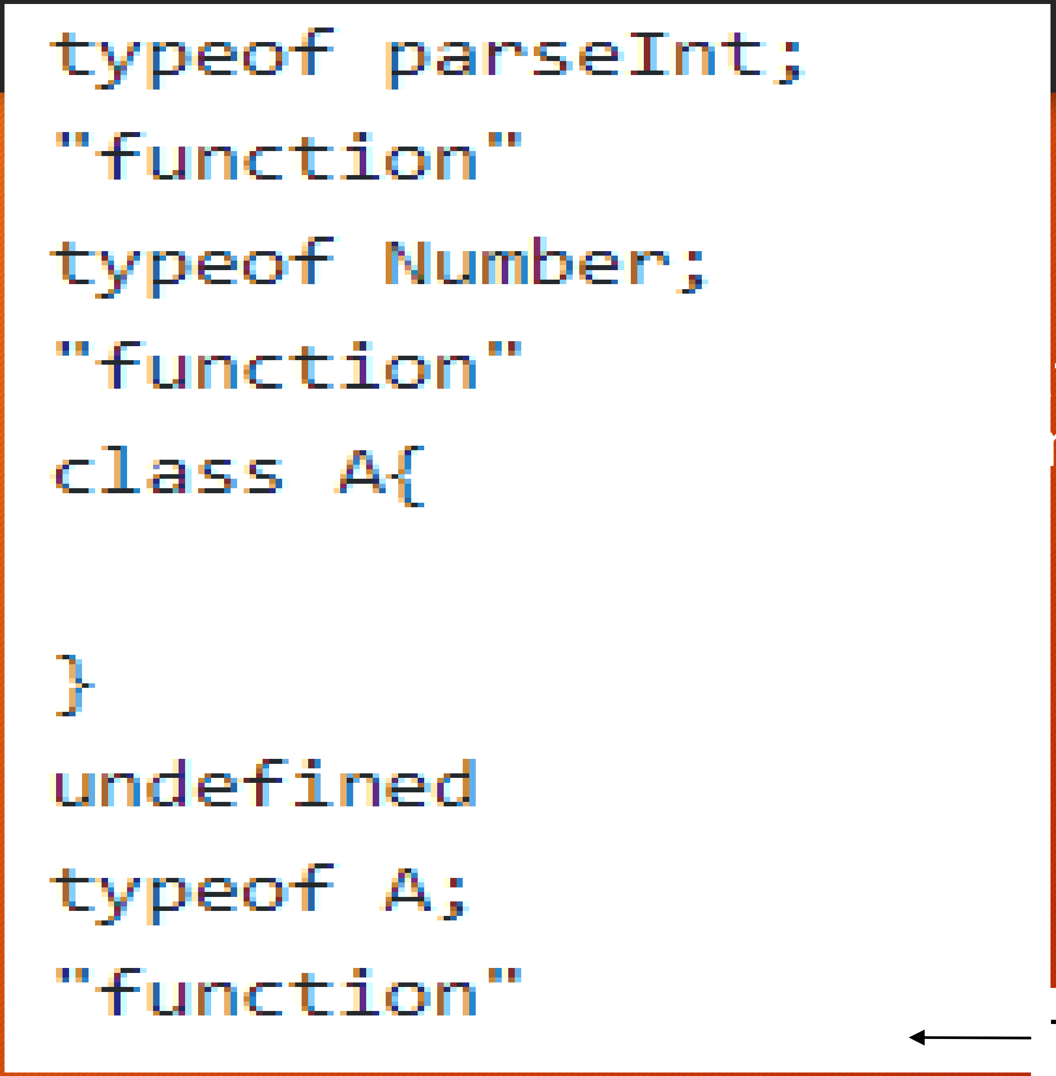
a is of type ‘number’.

a is of type ‘boolean’.

a is of type ‘string’.

Function Oriented Language

- ns in JavaScript.
- ment to another function.
- ion.
- manner as well.



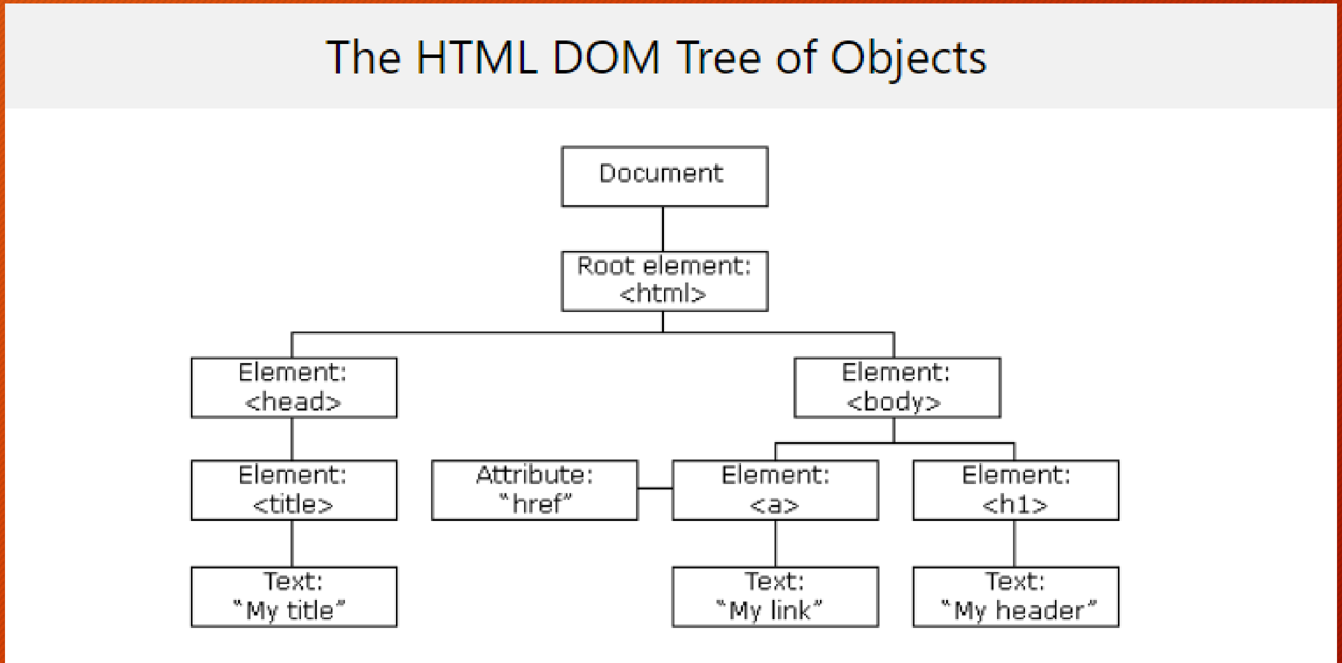
```
typeof parseInt;
"function"
typeof Number;
"function"
class A{


}
undefined
typeof A;
"function"
```

Type of class is function

JavaScript Supports DOM manipulation

- When a web page is loaded, the browser creates a Document Object Model of the page.
- The HTML DOM model is constructed as a tree of Objects:



JavaScript Supports DOM manipulation

- With the object model, JavaScript gets all the power it needs to create dynamic HTML:
  - JavaScript can change all the HTML elements in the page
  - JavaScript can change all the HTML attributes in the page
  - JavaScript can change all the CSS styles in the page
  - JavaScript can remove existing HTML elements and attributes
  - JavaScript can add new HTML elements and attributes
  - JavaScript can react to all existing HTML events in the page
  - JavaScript can create new HTML events in the page

Document Object Model

- The DOM is a W3C (World Wide Web Consortium) standard.
- The DOM defines a standard for accessing documents:
  - "The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document."
- The W3C DOM standard is separated into 3 different parts:
  - Core DOM - standard model for all document types
  - XML DOM - standard model for XML documents
  - HTML DOM - standard model for HTML documents

HTML Document Object Model

- The HTML DOM is a standard object model and programming interface for HTML. It defines:
  - The HTML elements as objects
  - The properties of all HTML elements
  - The methods to access all HTML elements
  - The events for all HTML elements
- In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

JavaScript History

- JavaScript was invented by Brendan Eich in 1995.
- It was inspired from Java, Scheme and Self.
- It was developed under the name Mocha which was changed to LiveScript for it's first beta release.
- Later it was renamed JavaScript.
- In December 1995, soon after releasing JavaScript for browsers, Netscape introduced an implementation of the language for server-side scripting with Netscape Enterprise Server.

ECMA Script

- ECMA Script (or ES) is a scripting language standardized by ECMA International in ECMA-262 and ISO/IEC 16262.
- It was created to standardize JavaScript, so as to foster multiple independent implementations.
- First edition of ECMA Script was published in June 1997.
- Multiple versions have been developed subsequently with the latest version in June 2018 including features for asynchronous iterations, new regular expression and rest/spread parameters.

JavaScript Engine

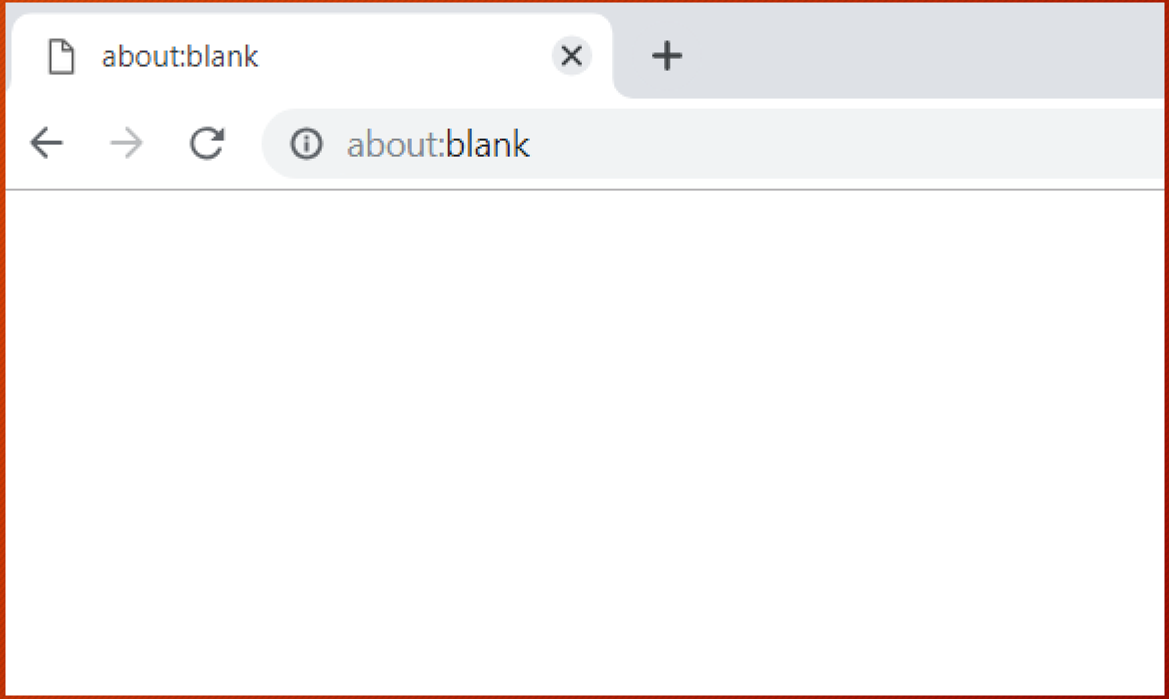
- JavaScript engine is a computer program to execute Javascript code.
- JS engines are developed by web browser vendors.
- In a browser, the JS engine runs in concert with the rendering engine via the Document Object Model (DOM).

JavaScript Engine

S No.	JS Engine	Used By
1.	Chrome V8	Google Chrome, Chromium browsers
2.	Spider Monkey	Firefox
3.	JavaScriptCore	Apple
4.	Chakra	Microsoft Edge

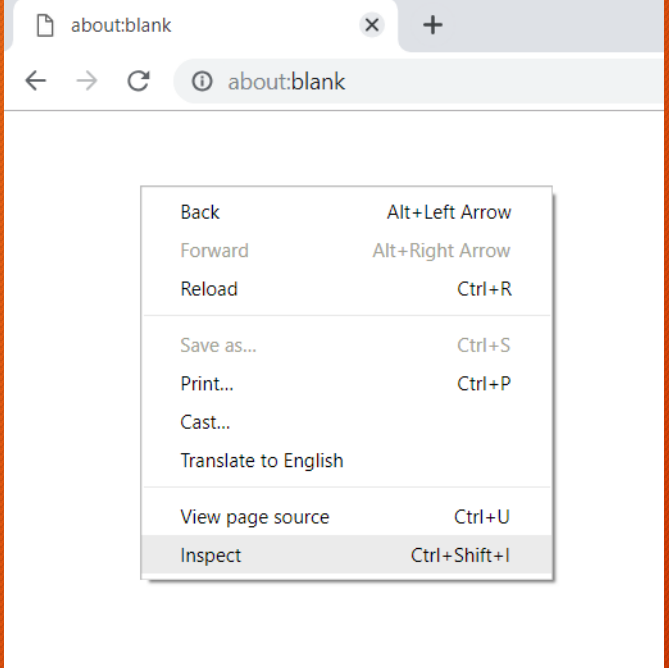
Developer Console

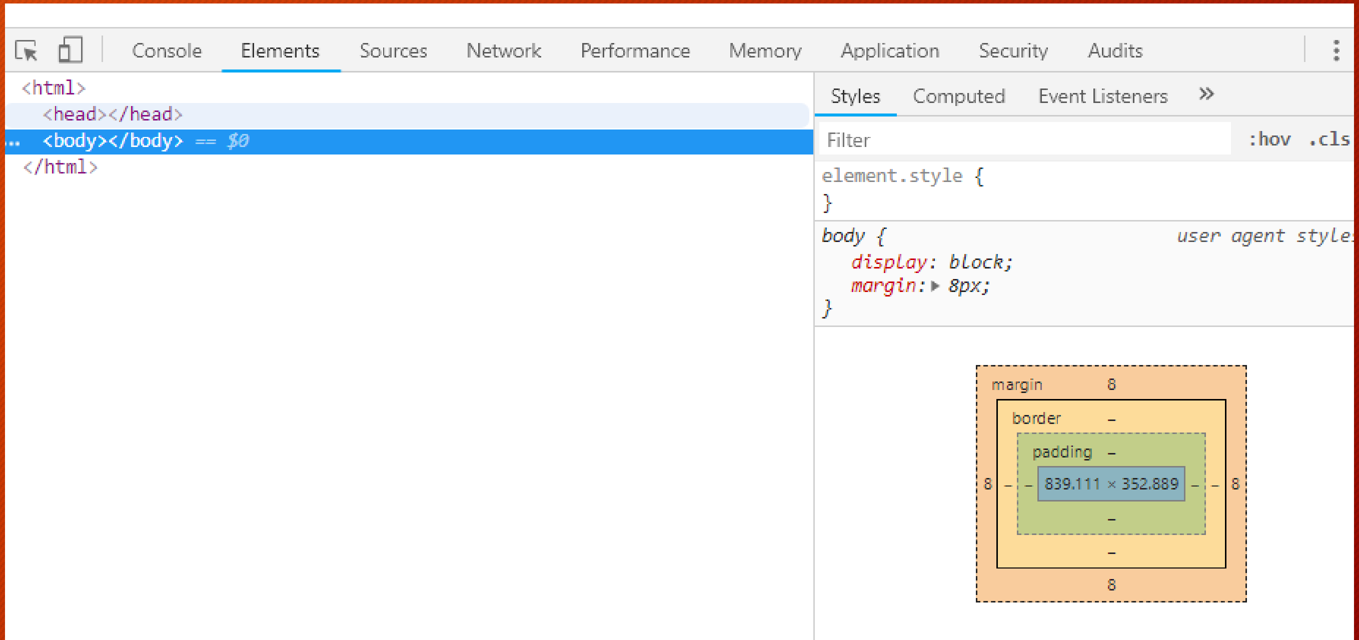
- Console in web browser can be used to execute JavaScript code.
- Steps to open console:
- Open a url “about:blank”



Developer Console

- Right click and select Inspect.



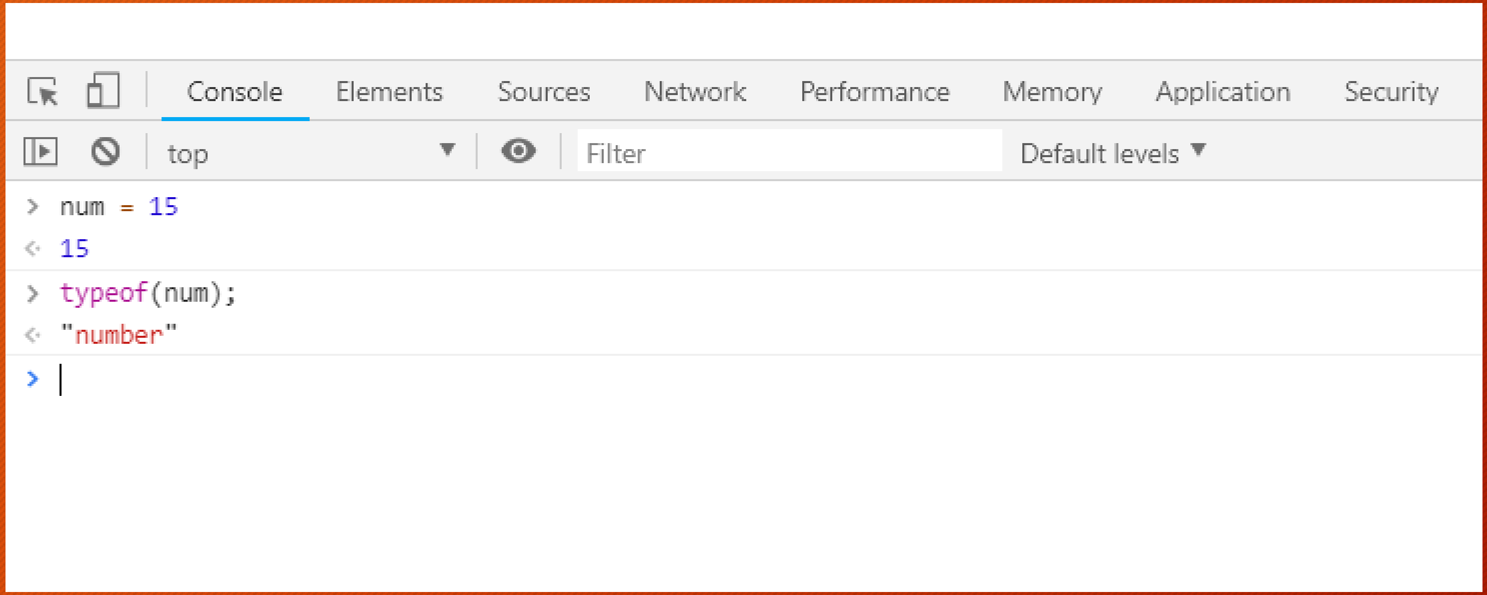


```
<html>
  <head></head>
  <body></body> == $0
</html>
```

```
element.style {
}
body {                                   user agent style
  display: block;
  margin: ▸ 8px;
}
```

Developer Console

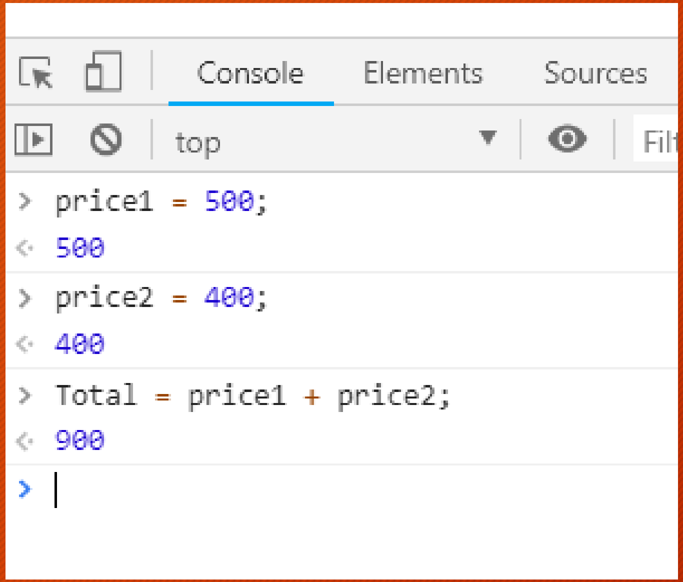
- Click on Console and write JavaScript code in console.



```
> num = 15
<· 15
> typeof(num);
<· "number"
>
```

JavaScript Variables

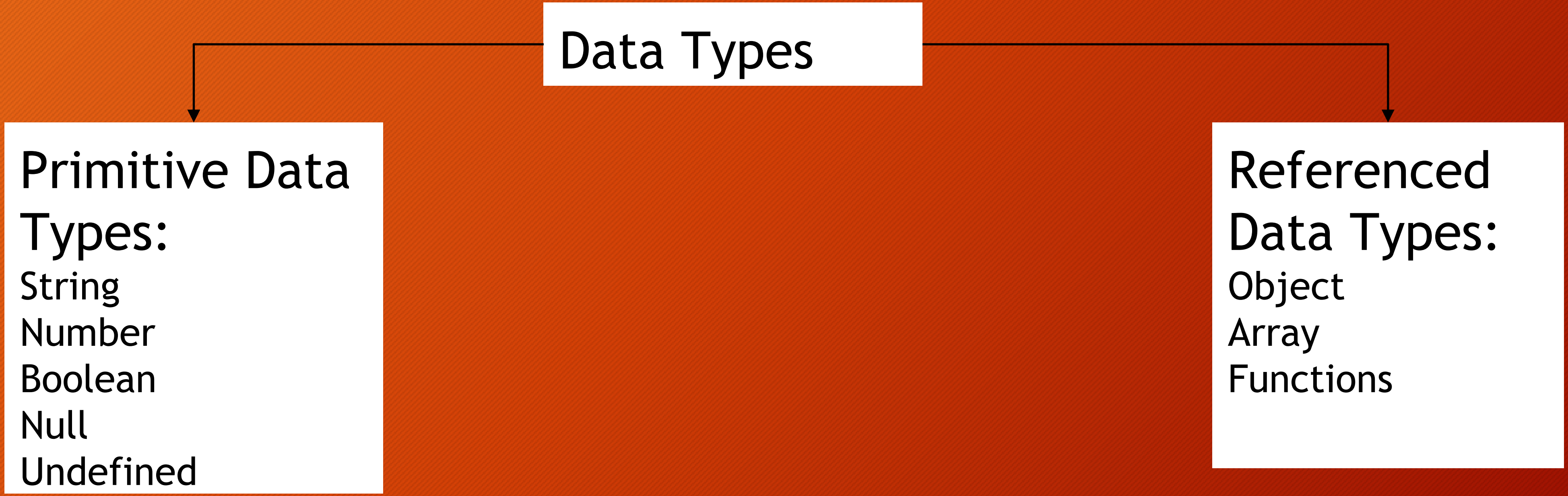
- JavaScript variables are containers for storing data values.



```
> price1 = 500;
<· 500
> price2 = 400;
<· 400
> Total = price1 + price2;
<· 900
>
```

Data Types

- There are two different data types in JavaScript:



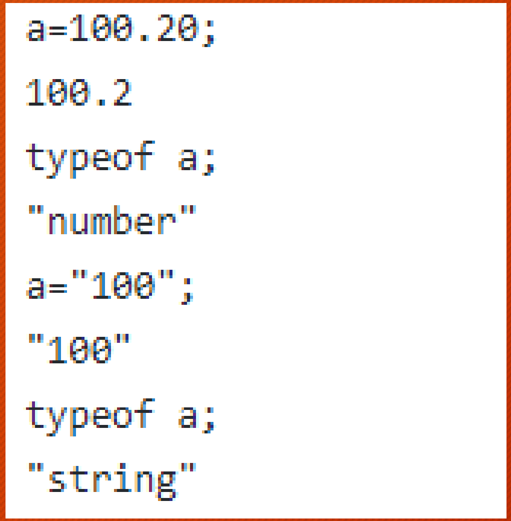
Data Types

Primitive Data Types:
String
Number
Boolean
Null
Undefined

Referenced Data Types:
Object
Array
Functions

Data Types

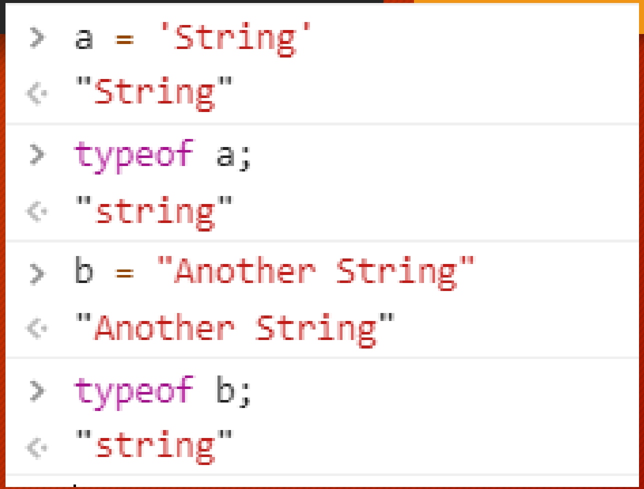
- JavaScript has dynamic types. That means the same variable can be used to hold different data types.



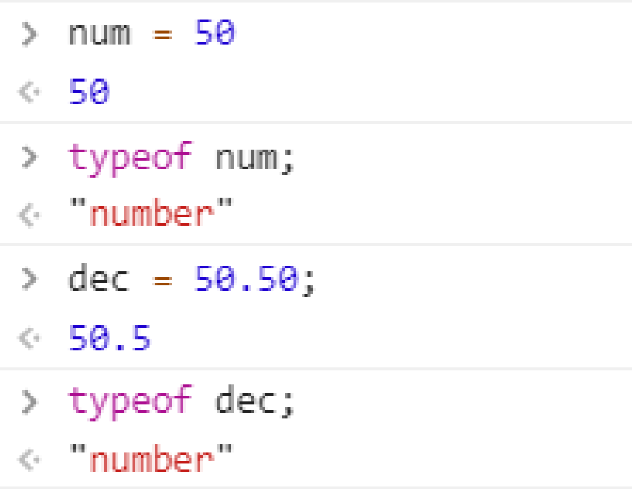
```
a=100.20;
100.2
typeof a;
"number"
a="100";
"100"
typeof a;
"string"
```

Primitive Data Types

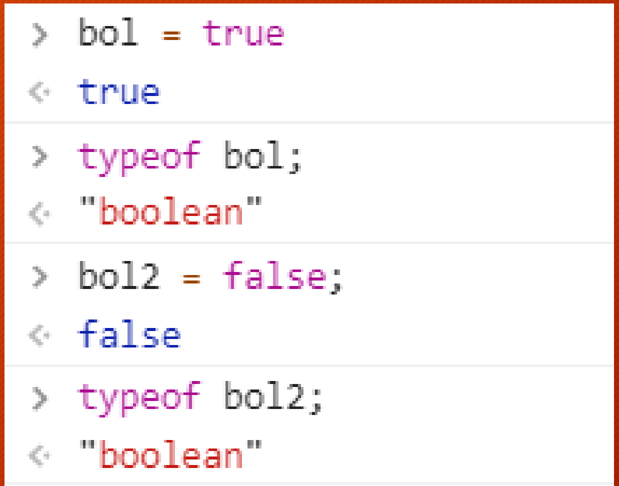
- ❖ Strings: can be written using single or double quotes.
- ❖ Numbers: can be written with or without decimals.
- ❖ Booleans: can have only two values.
- ❖ NULL: something that does not exist.
- ❖ Undefined: a variable without a value.



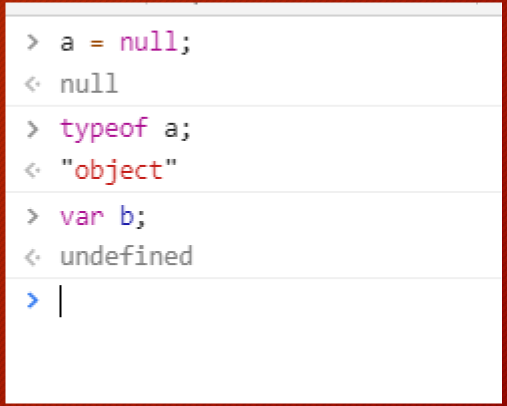
```
> a = 'String'
< "String"
> typeof a;
< "string"
> b = "Another String"
< "Another String"
> typeof b;
< "string"
```



```
> num = 50
< 50
> typeof num;
< "number"
> dec = 50.50;
< 50.5
> typeof dec;
< "number"
```



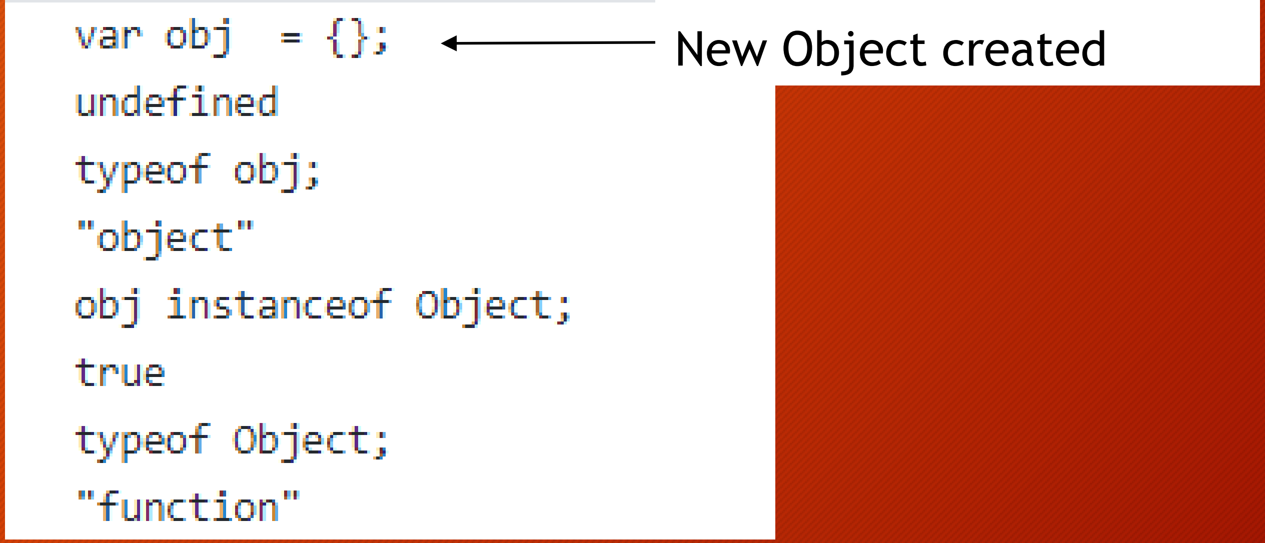
```
> bol = true
< true
> typeof bol;
< "boolean"
> bol2 = false;
< false
> typeof bol2;
< "boolean"
```



```
> a = null;
< null
> typeof a;
< "object"
> var b;
< undefined
> |
```

Reference Data Types: Object

- Objects: can be created without declaring a class.
- Objects in JavaScript are dynamic, we can define any number of attributes of the object.

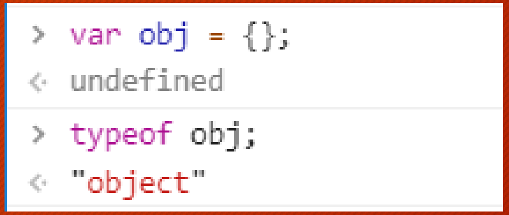


```
var obj  = {};
undefined
typeof obj;
"object"
obj instanceof Object;
true
typeof Object;
"function"
```

New Object created

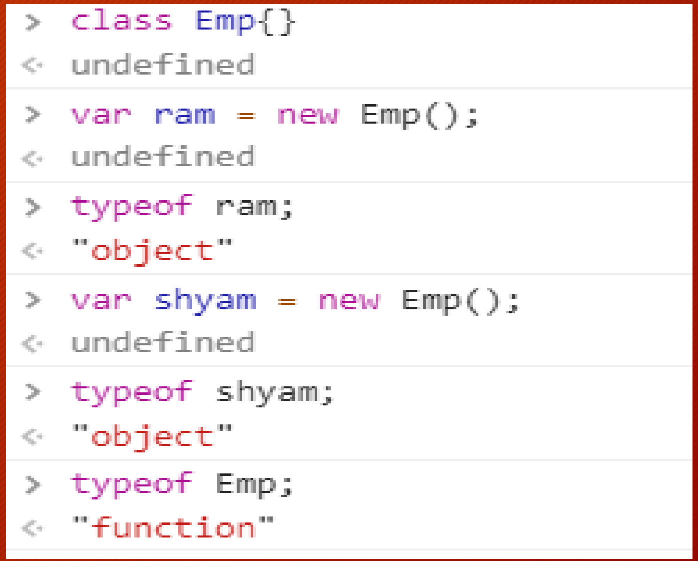
Object

- ❖ There are two approaches to build any object:
  - ➢ Object Literal Style: used to create singleton objects.



```
> var obj = {};
< undefined
> typeof obj;
< "object"
```

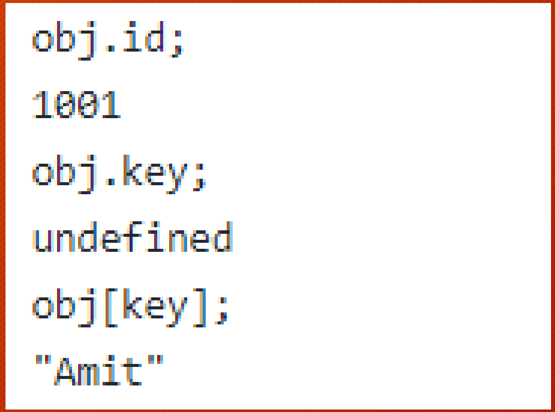
  - ➢ Classical Object Style: It can be used to grow and shrink objects dynamically.



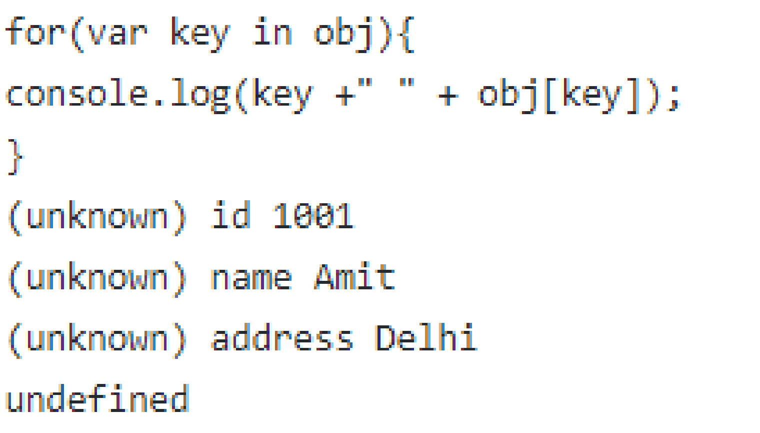
```
> class Emp{}
< undefined
> var ram = new Emp();
< undefined
> typeof ram;
< "object"
> var shyam = new Emp();
< undefined
> typeof shyam;
< "object"
> typeof Emp;
< "function"
```

Accessing an Object

- Objects can be accessed in two ways:
  - ➢ Using . (dot) operator
  - ➢ Using [] (square bracket) operator



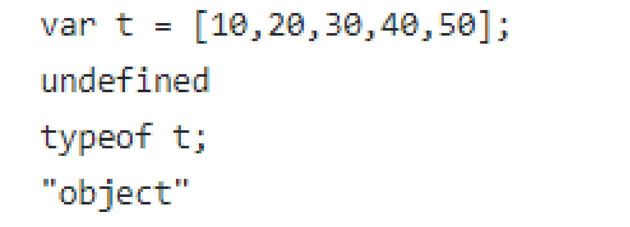
```
obj.id;
1001
obj.key;
undefined
obj[key];
"Amit"
```



```
for(var key in obj){
console.log(key +" " + obj[key]);
}
(unknown) id 1001
(unknown) name Amit
(unknown) address Delhi
undefined
```

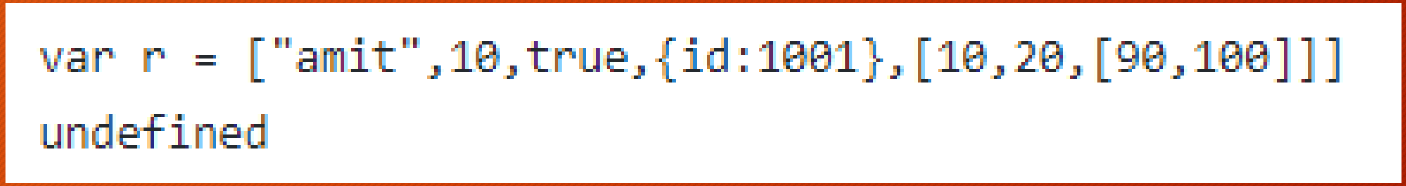
Reference Data Types: Array

- Array: array are special type of objects. Array uses numbers to access its “elements”.



```
var t = [10,20,30,40,50];
undefined
typeof t;
"object"
```

- There can be variables of different types in the same array.

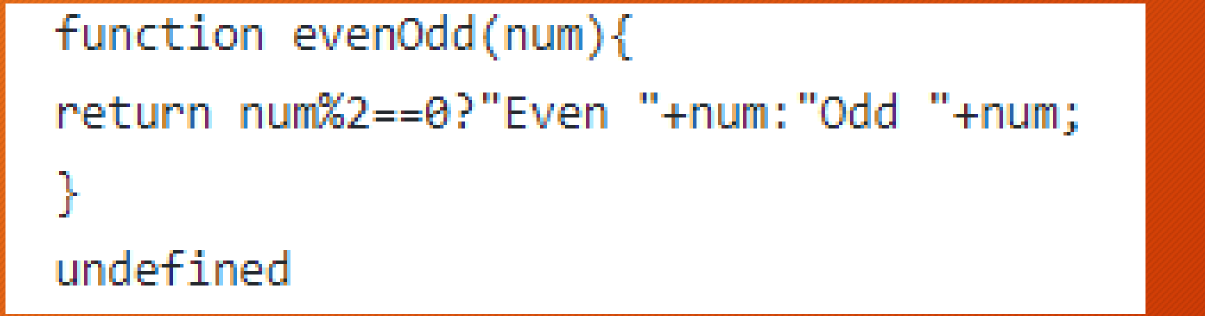


```
var r = ["amit",10,true,{id:1001},[10,20,[90,100]]]
undefined
```

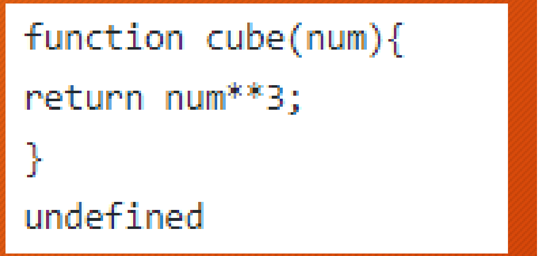
Reference Data Types: Functions

- In JavaScript, functions can be used the same way as reference data type.
- Functions behave as objects and can be passed as a parameter into another function.
- A function can also return another function.

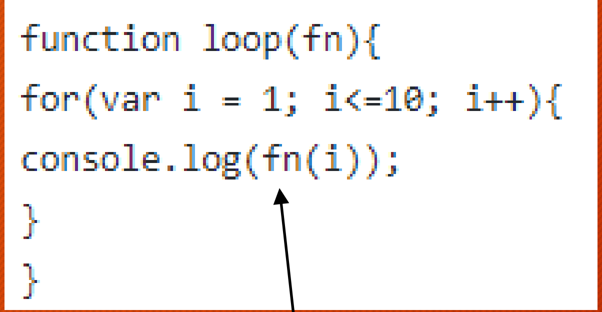
Functions



```
function evenOdd(num){
return num%2==0?"Even "+num:"Odd "+num;
}
undefined
```



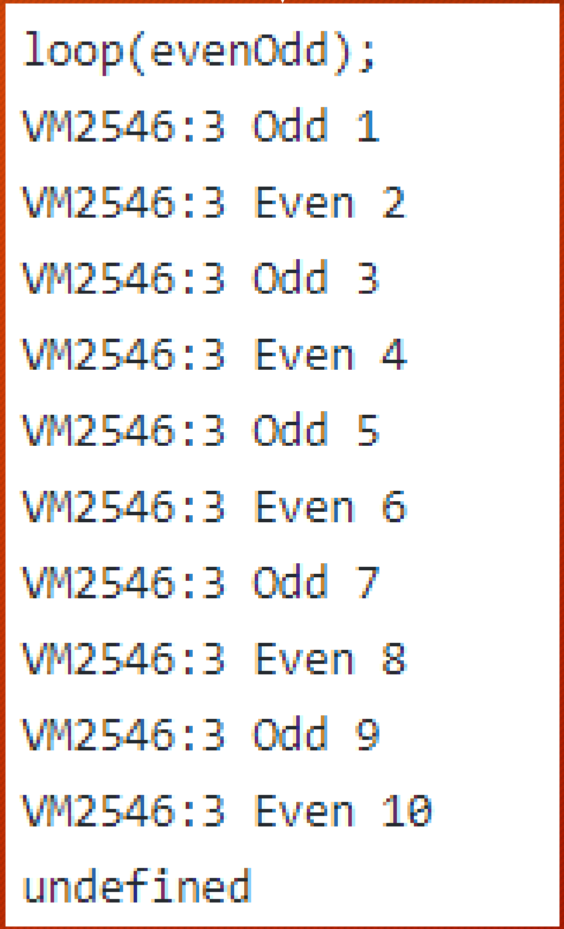
```
function cube(num){
return num**3;
}
undefined
```



```
function loop(fn){
for(var i = 1; i<=10; i++){
console.log(fn(i));
}
}
```

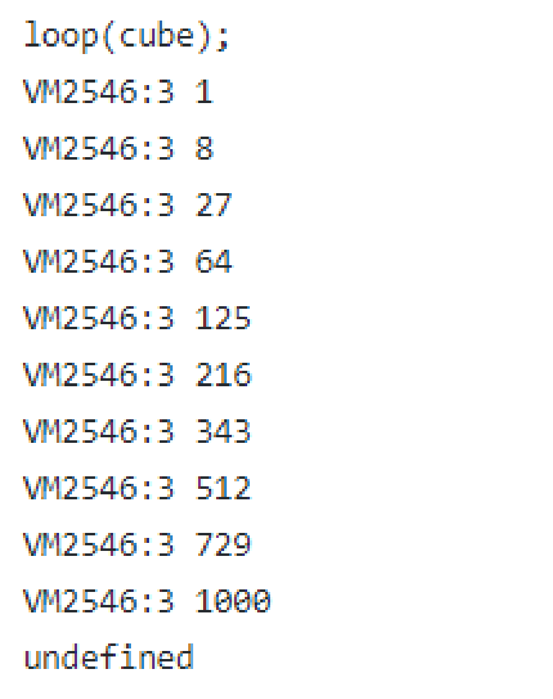
Function that takes another function as argument

Passing evenOdd function to loop function



```
loop(evenOdd);
VM2546:3 Odd 1
VM2546:3 Even 2
VM2546:3 Odd 3
VM2546:3 Even 4
VM2546:3 Odd 5
VM2546:3 Even 6
VM2546:3 Odd 7
VM2546:3 Even 8
VM2546:3 Odd 9
VM2546:3 Even 10
undefined
```

Passing cube function to loop function



```
loop(cube);
VM2546:3 1
VM2546:3 8
VM2546:3 27
VM2546:3 64
VM2546:3 125
VM2546:3 216
VM2546:3 343
VM2546:3 512
VM2546:3 729
VM2546:3 1000
undefined
```

References

- w3schools. Link: https://www.w3schools.com/js/
- Tutorials Point. Link: https://www.tutorialspoint.com/javascript/
- Geeks for Geeks. Link: https://www.geeksforgeeks.org/javascript-tutorial/
- The Modern JavaScript Tutorial. Link: https://javascript.info/

Video References

- JavaScript tutorial for beginners. Link: https://www.youtube.com/watch?v=W6NZfCO5SIk
- Learn JavaScript. Link: https://www.youtube.com/watch?v=PkZNo7MFNFg
- JavaScript Tutorial. Link: https://www.youtube.com/watch?v=PwsigsH4oXw

Thank You