

java script basics

- java script is a **client side scripting language**
- scripting languages are almost equivalent to **light weight** programming language
- java script is **case sensitive**

java script supported entity

- data types
- variables
- functions
- objects
- arrays
- constructor
- conditional statements
 - if
 - else if
 - else
 - switch
- loop
 - for loop
 - while loop
 - do while loop
 - for/in

operators supported in JS

< <= > >= == ===
& | && || + - * /
+= -= ++ -- *= != %

JavaScript doesn't support

- classes
- interface
- enum
- structures

where to write JavaScript??

javascript code can be written inside html file or in javascript file



javascript code written in html file is called as **inline javascript**

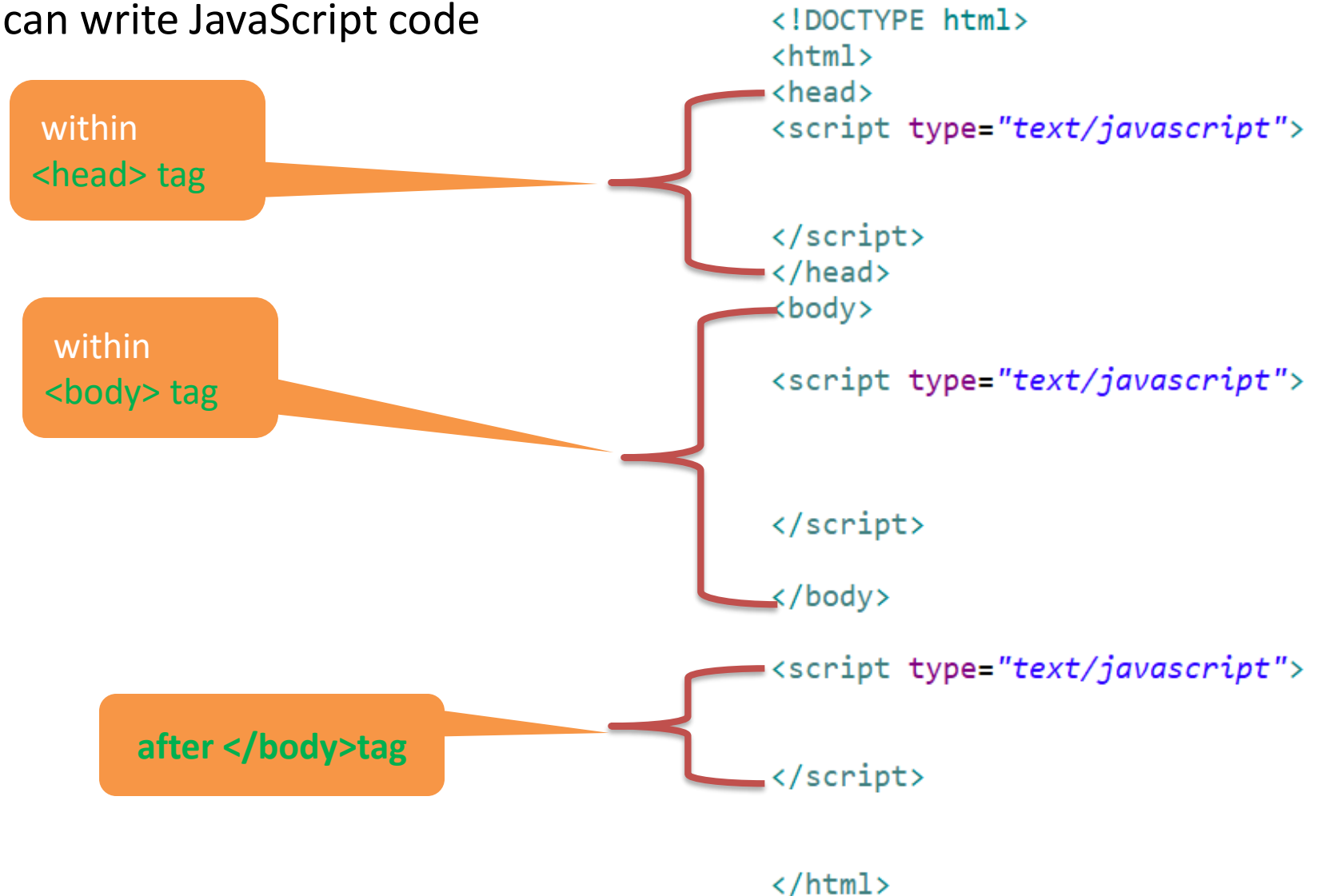


javascript code written in external javascript file is called as **external javascript**

writing JavaScript in html file

- to write JavaScript code in html file you have to use **<script>** tag.

we can write JavaScript code



variables in JavaScript

- using variables we can store data

declaring variables:

we can declare variables **with** or **without** using **var** keyword

```
var v=10;  
    a=9.8;  
    b=1000;  
    c='ABC';
```

```
var d="abc";  
var e=objects;  
f=false;  
g=functions;
```

- JavaScript is a **loosely typed** language

(data type is not required while declaring variable)

ex : **var v1=10; or v1=10 ;**

strictly typed language/strongly typed language

(data type is required while declaring variable)

ex : **in c# | java : int i=10 ;**

functions in JavaScript

using functions we can perform some actions/work

syntax:

```
function function_name(v1,v2,v3....)
{
    ---JavaScript code---
}
```

C# | java sample method
return type

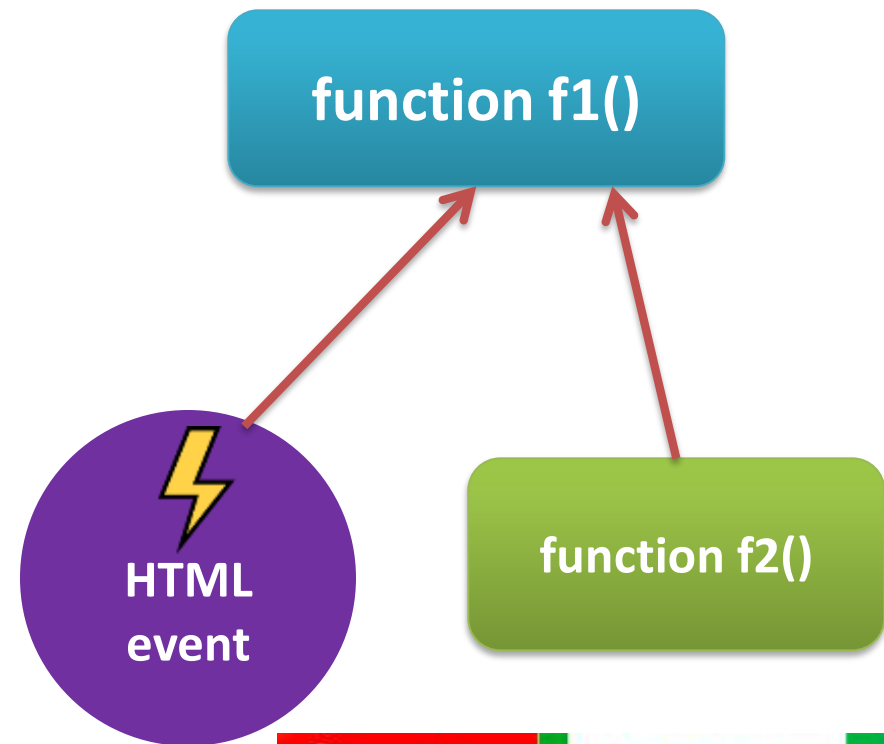
```
public int function_name(v1,v2,v3....)
{
    ---Java | C# code---
}
```

JavaScript function declaration will **not support return type** but we can return values using return keyword.

JavaScript will **not support access modifiers**

calling functions

- we can call a function from another function or from an html event.
- html event is a code executed for user actions.



js sample1



how will you display output in JavaScript ?

`alert(message)`

`document.write(text)`

req:1.when the user **clicks on submit1** display 'hello'

2.when the user **moves mouse on top of submit2**: display 'hi'

```
<html>
<head>
  <title>My Page</title>
  <script type="text/javascript">
    function f1()
    {
      alert('HELLO');
    }
    function f2()
    {
      f3();
    }
    function f3() {
      document.write('HI')
    }
  </script>
</head>
<body>
  <input type="submit" id="btn1" value="submit1" onclick="f1()" />
  <input type="submit" id="btn2" value="submit2" onmouseover="f2()" />
</body>
</html>
```

sample.html

SUBMIT1 SUBMIT2

onclick
ondblclick
onkeydown
onkeypress
onkeyup
onmousedown
onmousemove
onmouseout
onmouseover

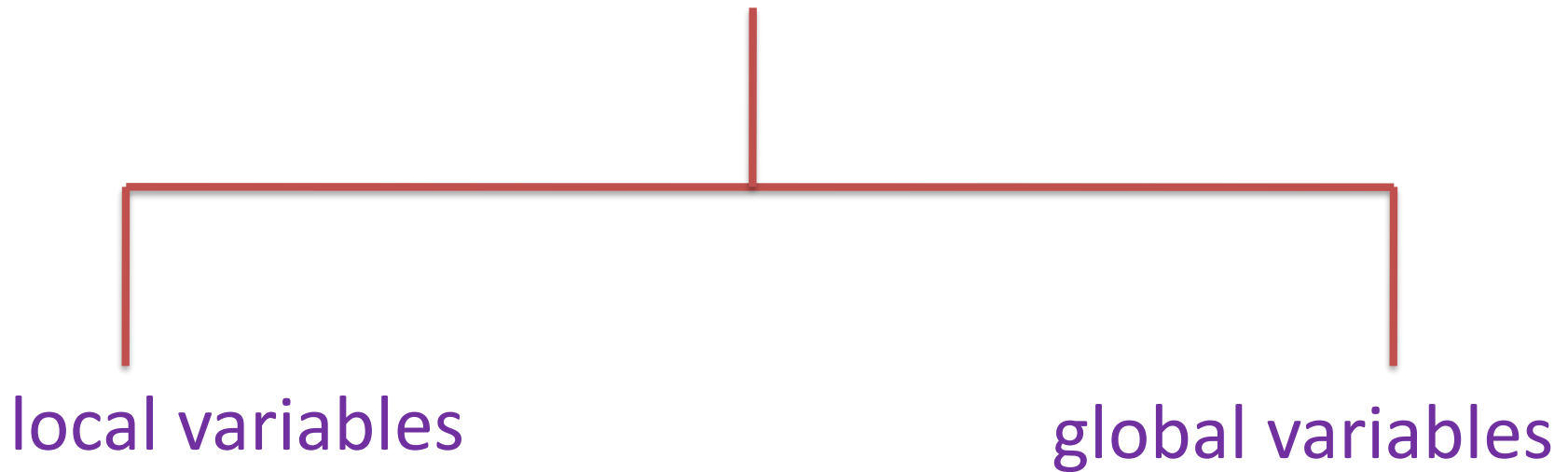
calling one function

er function

which event will be fired when user clicks
On a button?

calling function with an event

types of variables



local variables : variables declared inside the function

global variables : **variables** declared **outside** the **function** or variables declared **in function body** without **var** keyword.

local & global variables

```
<script type="text/javascript">
  var v1 = 10;
  v2 = "palle";
  function fn1(v3, v4)
  {
    var v5 = "palle university";
    v1 = v1 + 6;
    v6 = "hello";
  }
  function fn2()
  {
    alert(v6);
    alert(v5);
  }
</script>
```

Annotations:

- `v1, v2` are global variables (pointing to `var v1 = 10;` and `v2 = "palle";`)
- `v3, v4` are parameters/local variables (pointing to `fn1(v3, v4)`)
- `v5` is local variable (pointing to `var v5 = "palle university";`)
- we can call global variable inside a function (pointing to `v1 = v1 + 6;`)
- `v6` is global variable as we are not using var keyword (pointing to `v6 = "hello";`)
- `v6` is global variable we can use anywhere (pointing to `alert(v6);`)
- `v5` is local variable we cannot use in another function (pointing to `alert(v5);` with a red X)

conditional statements in javaScript

- **java script** is a **light weight** programming language
- java script supports conditional statements
if ,else if and else statements .
switch statements(faster compared to multiple
else if statements)

note:

all conditional statements (if ,else if ,switch) are
written in function body

syntax for if else in javascript

```
if(exp1)
```



If exp1 is true enters into body
or goto else if

```
{  
/*----js code----*/  
}
```

```
else if(exp2)
```



If exp2 is true enters into body
or goto next else if

```
{  
/*----js code----*/  
}
```

```
else if(exp3)
```



If exp3 is true enters into body
or goto else

```
{  
/*----js code----*/  
}
```

```
else
```



Only one else statement is
possible for if

```
{  
/*----js code----*/  
}
```

Note: we can write 0 or more else if statements
We can write 0 or 1 else statement.

using multiple expressions in conditional statements

if(exp1 && exp2) **If exp1 and exp2 both are true, execute the body**

```
{  
----js code----  
}
```



Use conditional and (&&) operator

if(exp1 || exp2) **If any one of the expression is true ,execute the body**

```
{  
----js code----  
}
```



Use conditional or(||) operator

syntax for switch statement in JavaScript

```
switch(variable/expression)
{
  case ☐ :
    /*jsgcode*/
    break;

  case ☐ :
    /*jsgcode*/
    break;

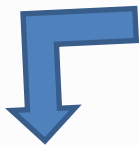
  case ☐ :
    /*jsgcode*/
    break;

  default:
    /*jsgcode*/
    break;
}
```

☐ → predicted value for variable

Assignment on conditional Statements:-

DisplayDayname.html



On click: "fndisplaydayname(3)"

function fndisplaydayname(dno)
{
1 → Monday Invalid day
2 → Tuesday
3 → Wednesday
4 → Thursday
5 → Friday
6 → Saturday
7 → Sunday
}

if else statement assignment solution

```
<html>
<head>
  <title></title>
  <script type="text/javascript" >
    function fndisplaydayname(dno) {
      if (dno == 1) {
        alert("monday");
      }
      else if (dno == 2) {
        alert("tuesday");
      }
      else if (dno == 3) {
        alert("wednesday");
      }
      else if (dno == 4) {
        alert("thursday");
      }
      else if (dno == 5) {
        alert("friday");
      }
      else if (dno == 6) {
        alert("saturday");
      }
      else if (dno == 7) {
        alert("sunday");
      }
      else {
        alert("invalid day");
      }
    }
  </script>
</head>
<body>
<input type="submit" id="btnsubmit" value="displaydayname" onclick="fndisplaydayname(3)" />
</body>
</html>
```


assignment solution using switch

```
<html >
<head>
  <title></title>
  <script type="text/javascript">
    function displayday(dno) {
      switch (dno) {
        case 1:
          alert("monday")
          break;
        case 2:
          alert("tuesday")
          break;
        case 3:
          alert("wednesday")
          break;
        case 4:
          alert("thursday")
          break;
        case 5:
          alert("friday")
          break;
        case 6:
          alert("saturday")
          break;
        case 7:
          alert("sunday")
          break;
        default:
          alert("invalid day")
          break;
      }
    }
  </script>
</head>
<body>
<input type="submit" id="tbsubmit" value="displayday" onclick="displayday(3)" />
</body>
</html>
```

arrays

- using arrays we can store multiple values under a single variable name.
- usually, to store 3 persons name ,we will declare three variables

```
var name1 = "paul";  
var name2 = "kevin";  
var name3 = "brain";
```

incorrect approach

using arrays

```
var names=['paul','kevin','brain'];
```

array syntax & internals

array syntax: `var array_name=[item1,item2,item3...]`

array sample: `var names=['paul','kevin','brain'];`



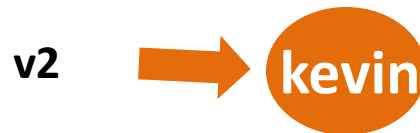
to retrieve 1st cell data

```
var v1 = names[0]
```



to retrieve 2nd cell data

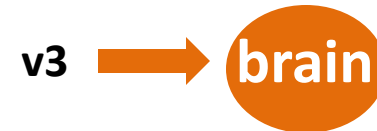
```
var v2 = names[1];
```



to retrieve 3rd cell data

```
var v3 = names[2];
```

v3 = "brain"





to modify 2nd cell data:

`names[1] = "kavitha";` if this code is executed kevin will be replaced with kavitha

to count total number of cells in array: `names.length`

loops

JavaScript supports

-  **for()**
- **for/in**  similar to for each loop in **c#** and **java**
- **while()**
- **do while()**

- **loops** are useful for **eliminating** the duplication of code (or) for executing a set of code for a specific number of times
- in this tutorial we will learn only for loop

loop sample

var nos =

101	21	67	278	45	334
-----	----	----	-----	----	-----

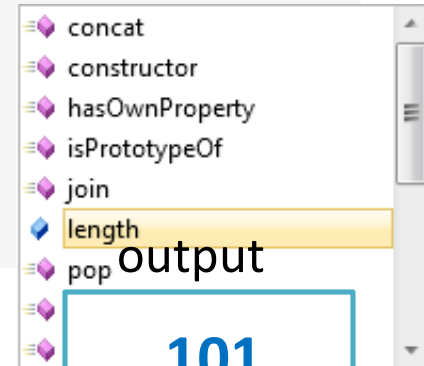
req: display all items present in the array nos

index numbers 0 1 2 3 4 5

```
alert (nos[ 0 ]);  
alert (nos[ 1 ]);  
alert (nos[ 2 ]);  
alert (nos[ 3 ]);  
alert (nos[ 4 ]);  
alert (nos[ 5 ]);
```

using loops , we can reduce the logical code duplication

```
for( i=0 ; i<nos.length ; i++ )  
{  
    alert (nos[ i ])  
}
```



use length property to find the length of array



if i add another element to the array, will it work..?

current i value

i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6

condition

0<=5
1<=5
2<=5
3<=5
4<=5
5<=5
6<=5

output

101
21
67
278
45
334

end of loop

debugging

- using **debugging** we can see **code execution seq.**
- we can **see run time variables data.**
- we can get **missing intelligence.**

where the JavaScript code are debugged ?

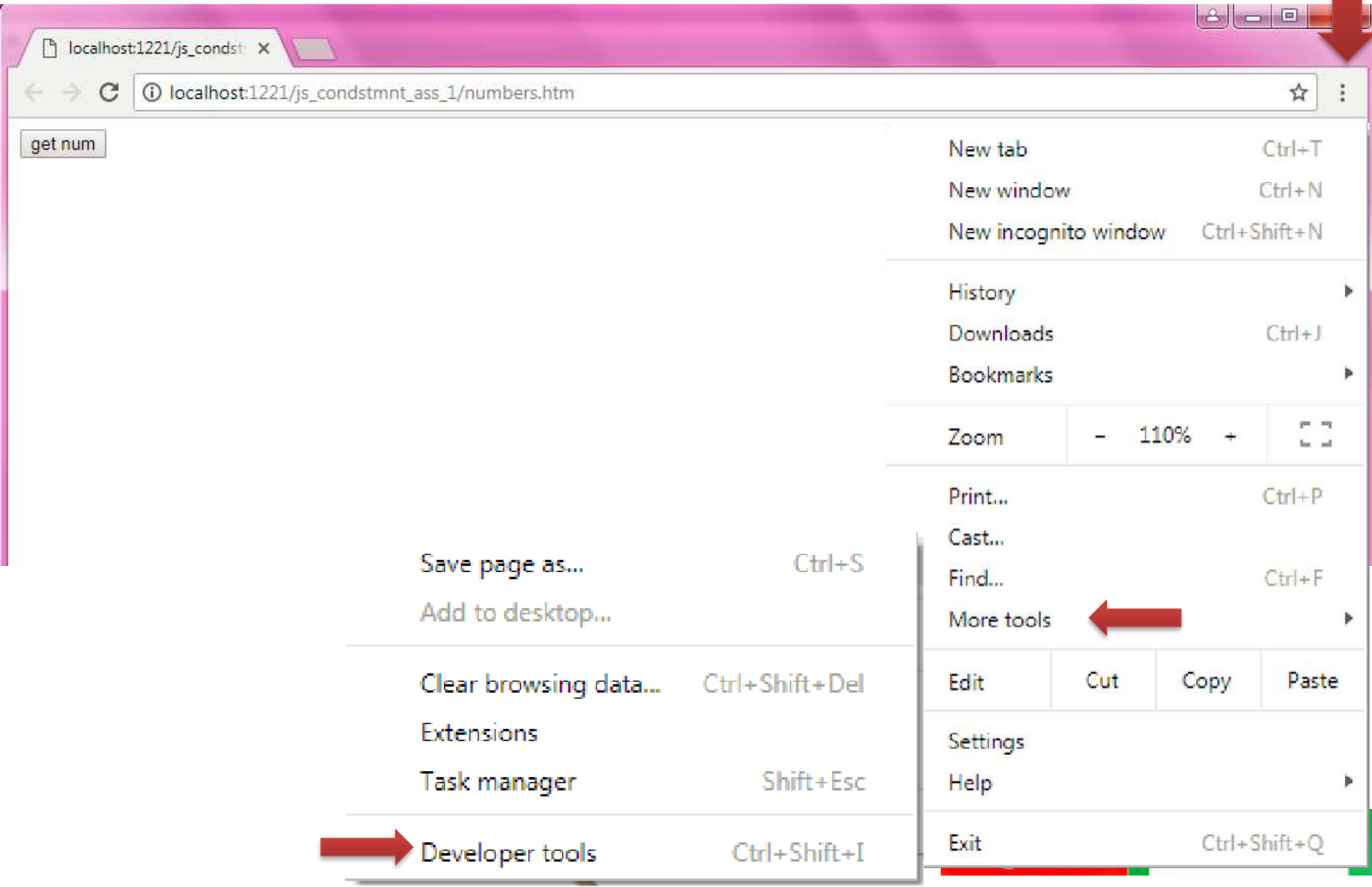
- visual studio
- local server
- web server
- **browser** ✓

debugger keyword

- for debugging JavaScript code , we have to include **debugger;** command in required function.

```
<script type="text/javascript">  
    function f1() {  
        debugger;  
        -----js code-----  
    }  
</script>
```

debugging in chrome – part1



debugging in chrome – part2

F11

localhost:1472/jsass_21st

localhost:1472/jsass_21st_feb/try.htm

getnum

click on button to start debugging. use F10 | F11 key to navigate

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1
2
3 <html>
4 <head>
5   <title></title>
6   <script type="text/javascript">
7     fun 10 on f1() {
8       debugger;
9       v1 = input#btn1
10      v2 =
11    }
12
13  </script>
14 </head>
15 <body>
16 <input type="tex
17 <input type="but
18 </body>
19 </html>
20
21
```

See the runtime variables data
we will get the missing intelligence support.

v1 = input#btn1

- accept: ""
- accessKey: ""
- align: ""
- alt: ""
- assignedSlot: null
- attributes: NamedNodeMap {0: type, 1: id}
- autocapitalize: "none"
- autocomplete: ""
- autofocus: false
- baseURI: "http://localhost:1472/jsass_21st_feb/try.htm"
- checked: false
- childElementCount: 0
- childNodes: NodeList [1]

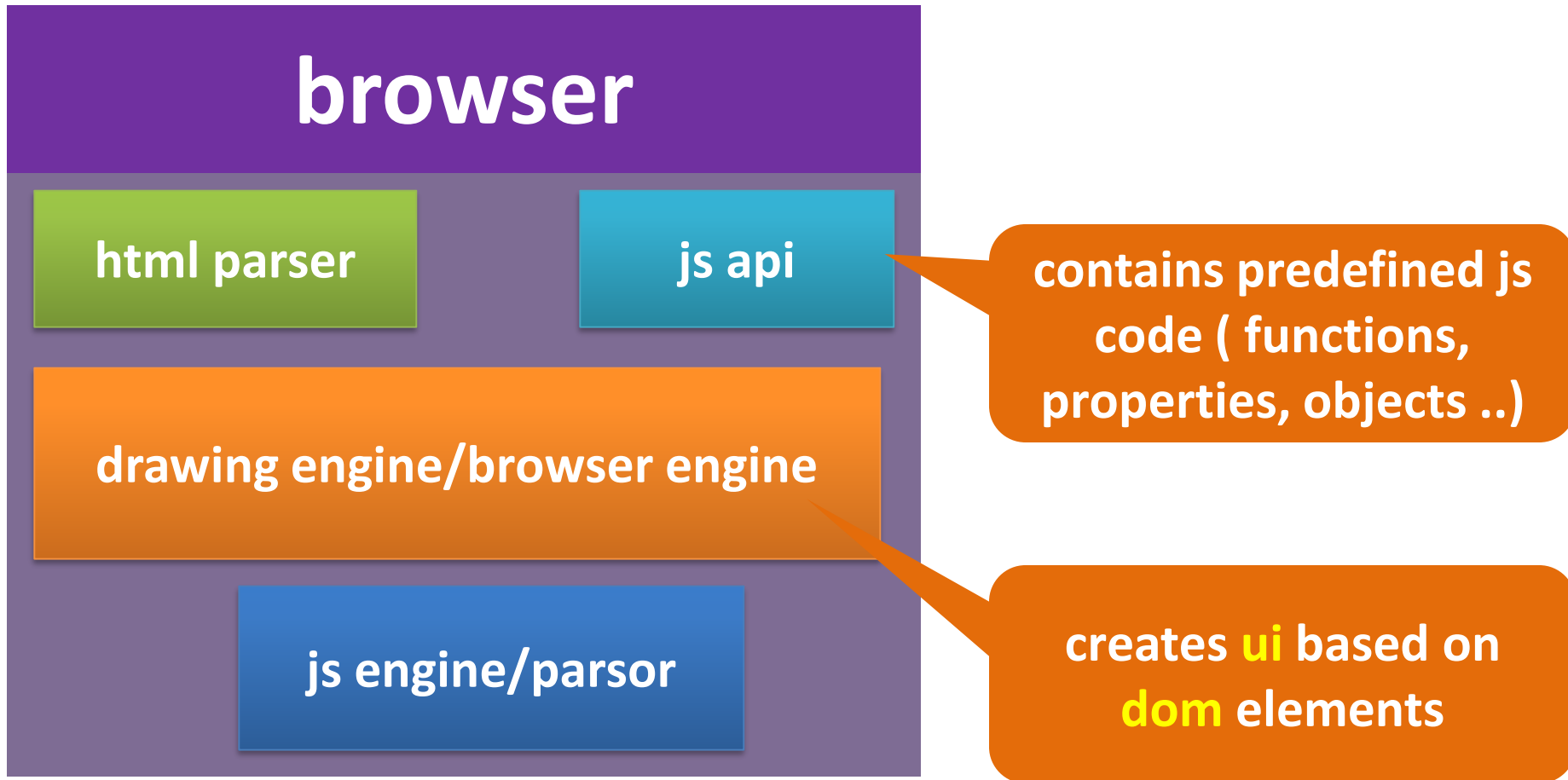
num" />

dom

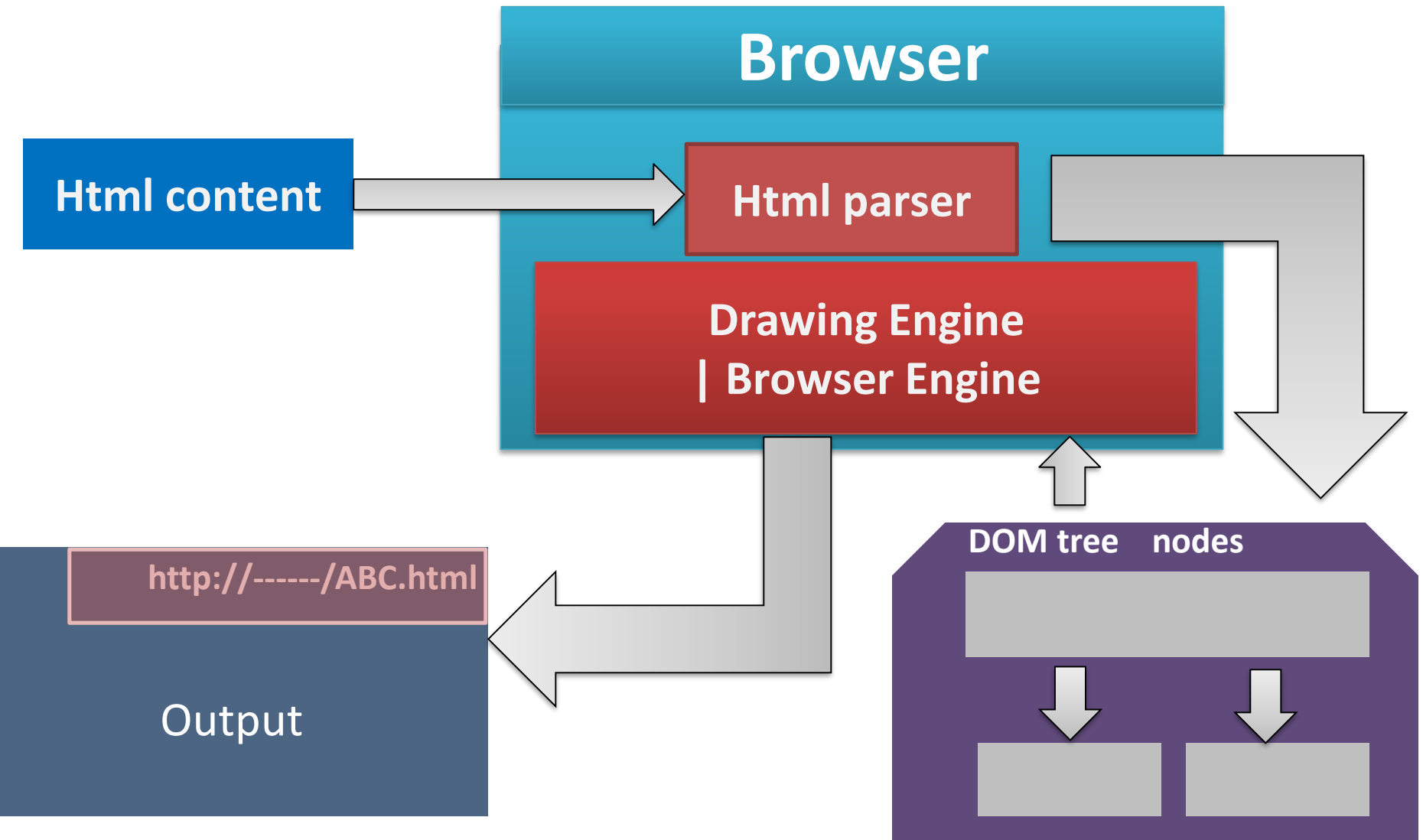
- dom refers to document object model.
- after sending the html to browser each html element will be converted to a node.

browser architecture

browser is a software which contains lot of sub software

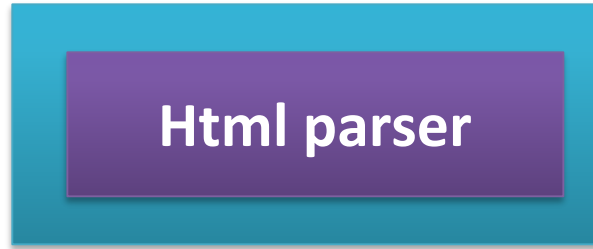


html to ui conversion



html to dom conversion

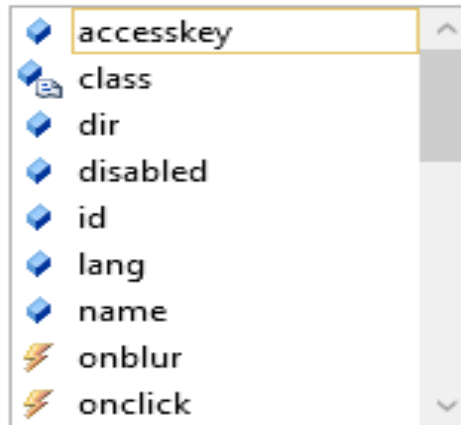
```
<html>
<head>
  <title></title>
</head>
<body>
  Name:<input type="text" id="tb1" />
</body>
</html>
```



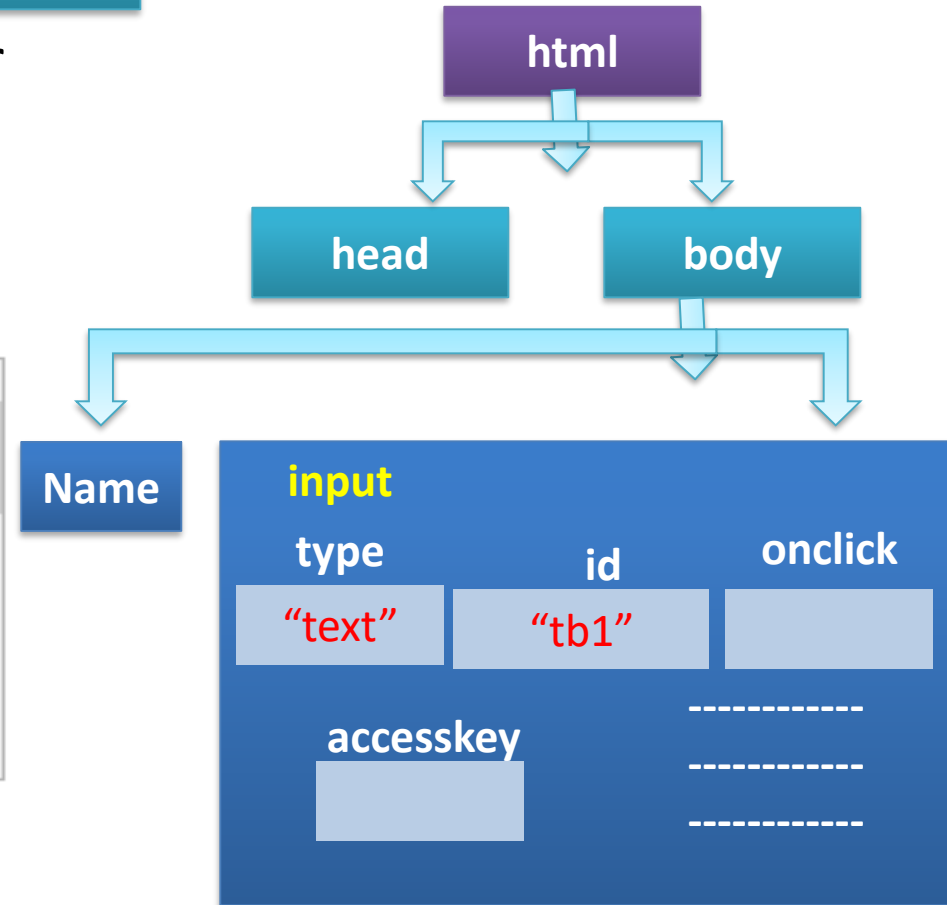
html parser will identify the first/root element in .html file



browser



While typing give space you will get intellisense window with all properties and html events



Note: unused properties /events in input tag will also be present in dom tree node

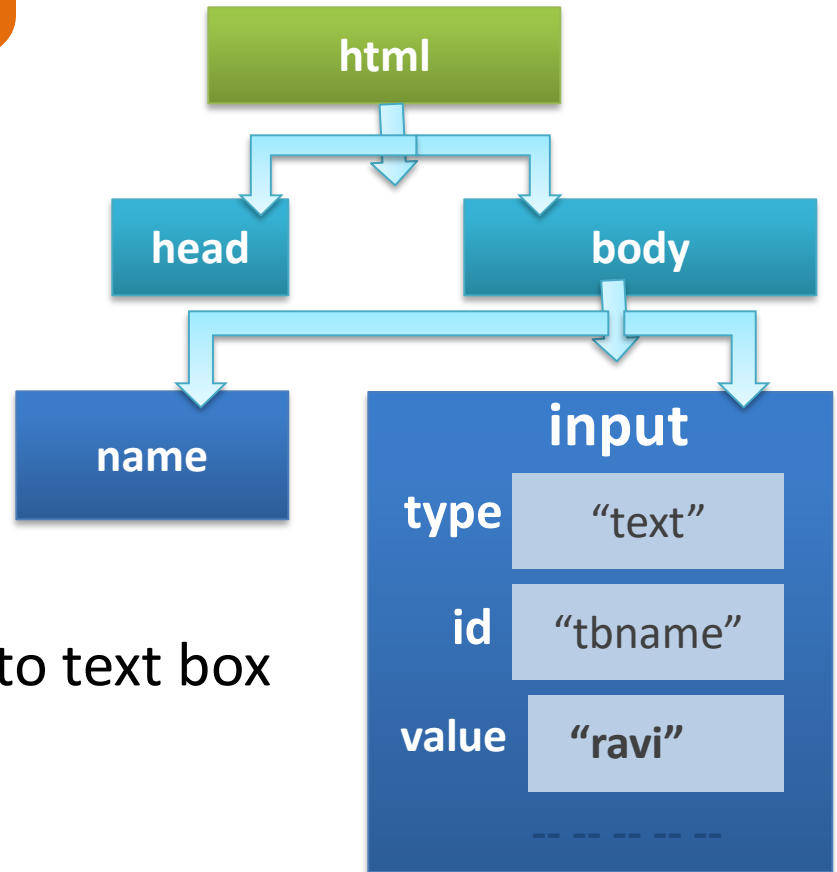
user actions in dom

remember all user actions
will be recorded in dom

http://-----/palle.html

name:

assume user typed some name into text box



req:read this value properties data and change the value by using
JavaScript

getting node reference

```
function f1()
{
  v1 = document.getElementById("tbname");
  v2 = v1.value;
  alert(v2);
}
```

name:



where exactly the name is stored??

v1

v1

v2

input	
id	"tbname"
type	"text"
value	"ravi"
checked	false

input	
id	"btnsbmt"
type	"submit"
value	"submit"
onclick	f1()

req:
read the textbox data using javascript

now , we understood how to access a property present in a node.

create a node reference

simply write **v1.value**

how will you see all the properties present in a node

there is a way to see all properties in a node
we will understand that in next slide

observing nodes data

----abc.html

palle

finish

v1

input

type: "text"

autofocus: false

checked: false

id: "txt1"

onclick: ""

ondblclick: ""

onmouseover: ""

placeholder: ""

value: "palle"

classname: ""

```
<script type="text/javascript">
```

```
function f1() {
```

```
  debugger;
```

```
  v1 = document.getElementById("txt1");
```

```
}
```

req: to access these properties while debugging

assume user has enabled debugging ;

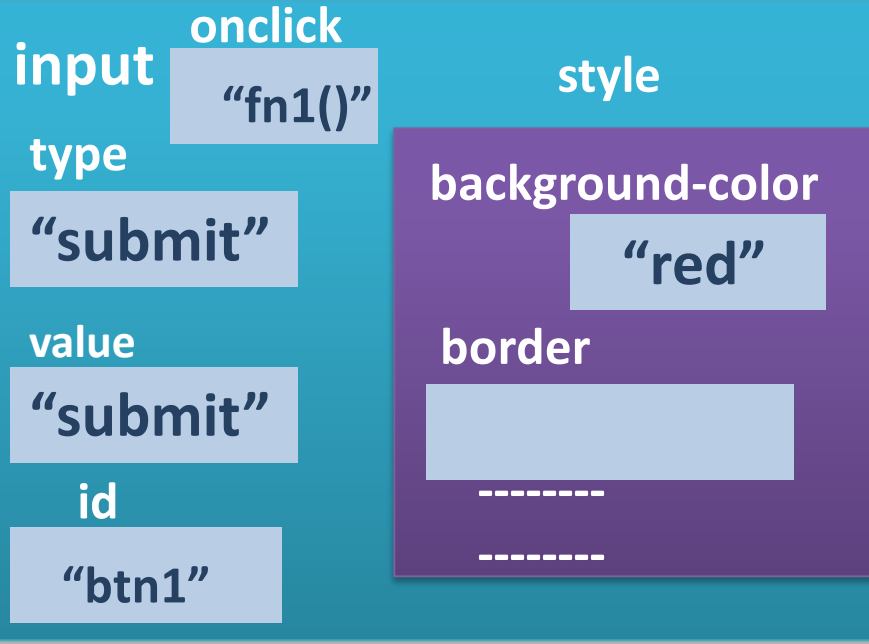
user can use the
required properties

F11

finding control properties lab-1

```
<input type="submit" id="btn1" value="submit" onclick="fn1()" style="background-color:red"/>
```

<http://--/sample.html>



```
function fn1()  
{  
  debugger;  
  var v2 = v1.style.backgroundColor;  
  var v1 = document.getElementById("btn1");  
}
```

how will you retrieve data



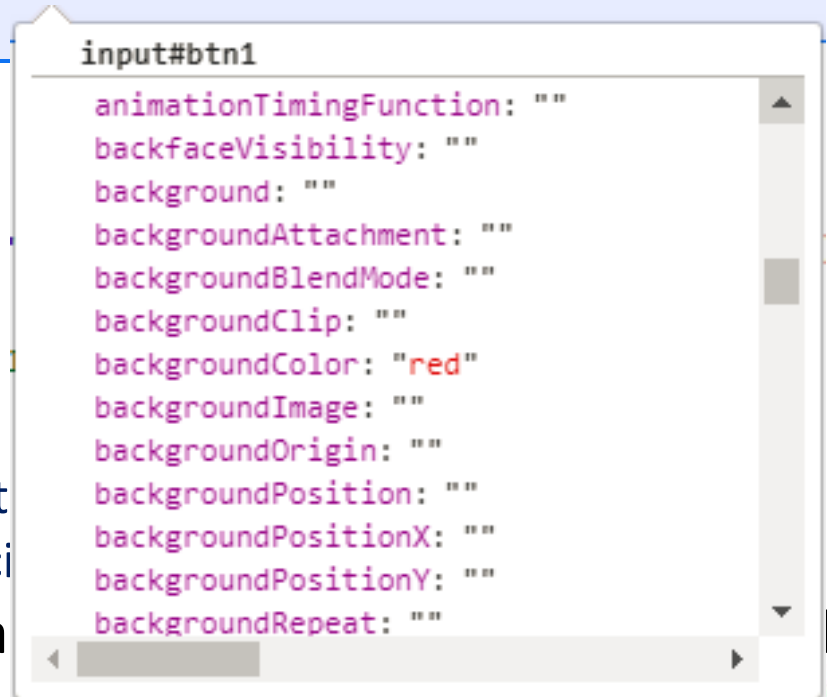
assume user has enabled debugging

how will you access the properties of button

create a node reference

how to associate

req: write a function to read node data background color using javascript



finding control properties –lab2

```
cc var v2 = v1.style.borderBottomColor;  
   var v3 = v1.style.borderRightColor;  
   var v4 = v1.style.borderLeftColor;  
   var v5 = v1.style.borderTopColor;
```

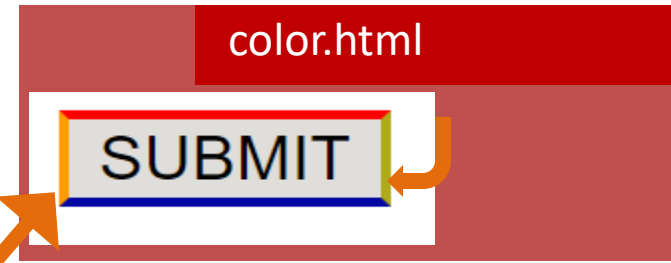
browser

```
function f1()  
{  
  debugger;  
  var v1 = document.getElementById("sub1");  
}
```

scroll the intellisense
window.



create a node reference



assume user has enabled
debugging

v1

input

type	id	value	onclick
submt	sub1	submit	f1()

style

borderleftcolor	orange	-----
borderrightcolor	yellow	-----
borderbottomcolor	blue	-----
bordertopcolor	red	-----

validations

we can verify whether user given data is in valid format or not

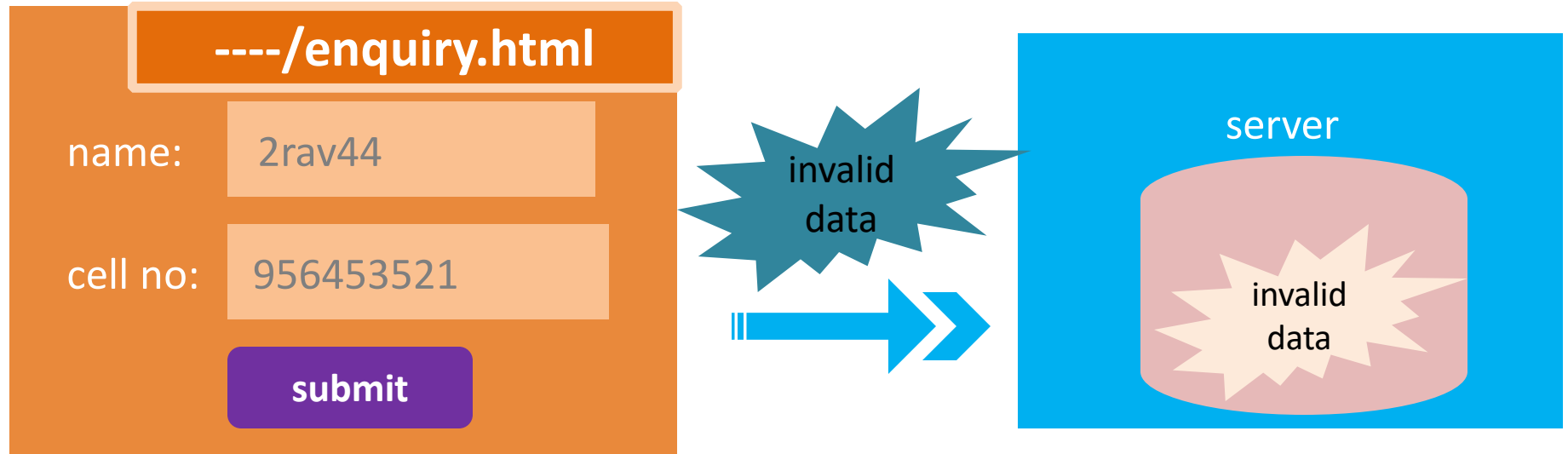


Why should we validate the user given data?

We will understand in the next slide

why to validate a data

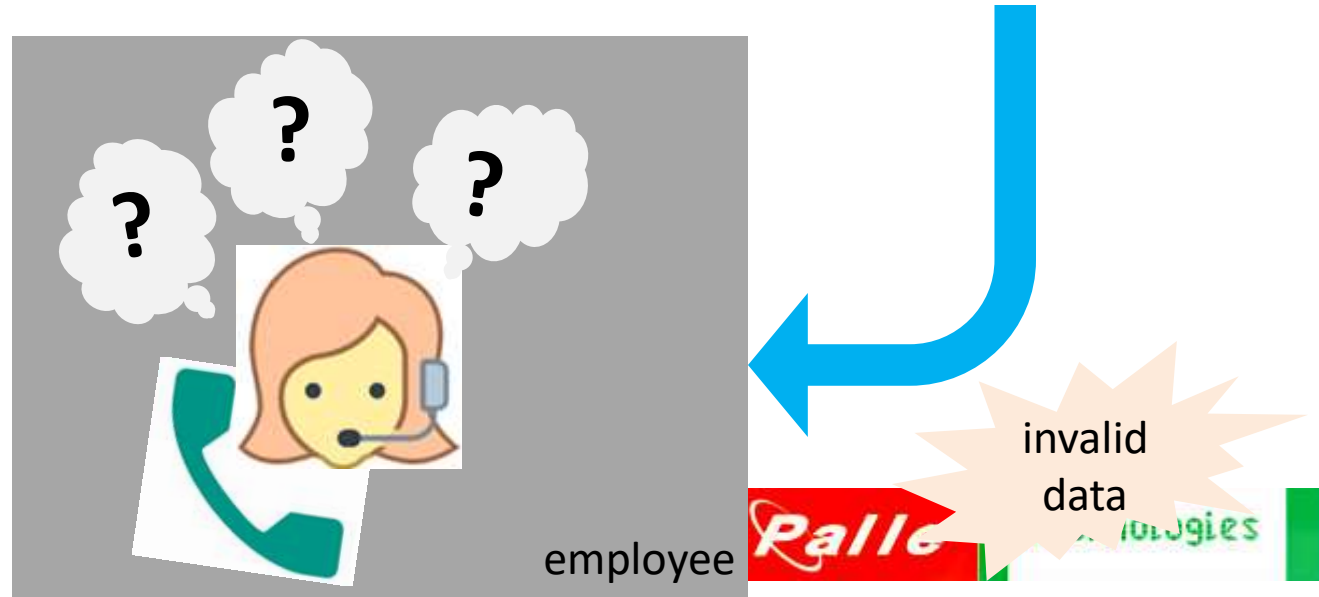
assume there is company having a webpage enquiry.html



as a programmer we must not allow the bad data inside the application



how to find which is good data and bad data?



validation using regular expressions

regular expression:

we can find whether a variable data is present in valid format or not

regular expression:

meta char + **quantifiers** + **brackets**

meta characters

Meta chars	descriptions/meaning
.	Any single character
\w	Used for finding word characters(a-z,A-Z,0-9,_)
\W	Used for finding any non word characters
\d	For finding digits
\D	For finding non digits
\s	White space
\S	Non white space

types of quantifiers

Quantifiers	meanings
(Expression) +	Expression for 1 or more times
(Expression) *	Expression for 0 or more times
(Expression) ?	Expression for 0 or 1 time
(Expression) {no}	Expression for exactly specified number of times
(Expression) {minno,maxno}	Expression specified in that specific range

brackets

Brackets[]	examples
[set of characters]	[a b h l]
[range of characters]	[d-k] means any character between d to k
[Checking special character]	Any special character used in regular expression must be placed in square brackets

sample validations

```
var v1 = '909089790';
```

req:to check whether v1 has 10 digits or not?

regular expression : `\d{10}`

*regular expression is written in between

`/^_____$/`

```
var rex1 = /^\\d{10}$/;
```

match function

- **match** function is used for validations.
- `variable_which_holds_data.match(variable_holding_regular_expression);`

returns null if variable data not matching to regular expression

```
if(v.match(rex1)==null)
{
    alert("error message");
}
```

returns null if variable data not matching to regular expression

```
if(!v.match(rex1))
{
    alert("error message");
}
```

validation lab-1

- write validation expression for checking a variable named “v” is having + or – symbol exactly one time between one or more english alphabets.
- write validation expression for indian cell phone number i.e (+91-xxxxxxxxxx) or without +91-

sol:var v2=/^[+]*[91]*[-]*\d{10}\$/;

validation sample2

- req:name must contain minimum 3 alphabets ,maximum can be 10



A.Html

name:

submit

Regular expression: `/^[a-z A-Z]{3,10}$/`

validation sample3

- show validation expression for +91-10digit mob num
- validation for email id
- `var v3 = /^\\w+([\\.-]?\\w+)*@\\w+([\\.-]?\\w+)*([\\.]?\\w)+$/`
- validation for full name with a space in between