Javascript

3 Layers of the Web

- HTML - for Content
- CSS - for Presentation
- JavaScript - for Behavior

```
<script type="text/javascript">
        // javascript code
        /*
                javascript code s
        */
</script>
```

Javascript

- An object-oriented scripting language.
- JavaScript programs can run both on the client and the server sides.
- Decision Making, complex calculations, Validate Data, Animate and Add Effects
- React to events
- Read, Write and Modify HTML elements
- Open and cross-platform
- allows interaction with properties of object
  - Internal built-in objects (e.g. window object).
  - Browser objects (e.g. document object).

Programming with Javascript

There are 2 ways:

- Write javascript code direct

```
<script type="text/javascript">
    // code goes here

</script>
```

- Include javascript source file

```
<script type="text/javascript" src="myscript.js">
</script>
```

Writing Javascript Program

```
<HTML>
  <TITLE> Displaying Text </TITLE>
  <BODY>
    <SCRIPT>
       document.write("<h1> Hello Good Day </H1>");
       document.write("<H3> Best of Luck. </H3>");
        alert("Hello");
    </SCRIPT>
  </BODY>
</HTML>
```

Statements, Comments, Variables

- Statements - Each line of code separated by new line or semicolon
- Comments
  - Single Line - // Single line of Code
  - Multiline - /* Some lines of Code */
- Variables - used to hold data and declared with the keyword var
- Variables can be Local or Global

```
var name;     // declare
var a = 10;   // declare and initialization
```

Variable Data Type

- Javascript is loosely typed language.
- Different Data Types
  - Numbers
    - Integer - 3
    - Float – 4.5
  - Strings
  - Boolean – True or False

Operator

- Arithmetic :
  - +, -, *, /, %, ++, --
- Comparison :
  - >, <, >=, <=, ==, ===, !=
- Logical :
  - && (AND), || (OR), ! (NOT)
- Assignment :
  - =, +=, -=, *=, /=, %=
- Conditional or Ternary - **? :**

Arithmetic Operator

Operator	Description	Example	Result
+	Addition	x=2 y=2 x+y	4
-	Subtraction	x=5 y=2 x-y	3
*	Multiplication	x=5 y=4 x*y	20
/	Division	15/5 5/2	3 2,5
%	Modulus (division remainder)	5%2 10%8 10%2	1 2 0
++	Increment	x=5 x++	x=6
--	Decrement	x=5 x--	x=4

Assignment Operator

Operator	Example	Is The Same As
=	x=y	x=y
+=	x+=y	x=x+y
-=	x-=y	x=x-y
*=	x*=y	x=x*y
/=	x/=y	x=x/y
%=	x%=y	x=x%y

Comparison Operator

Operator	Description	Example
==	is equal to	5==8 returns false
===	is equal to (checks for both value and type)	x=5 y="5" x==y returns true x===y returns false
!=	is not equal	5!=8 returns true
>	is greater than	5>8 returns false
<	is less than	5<8 returns true
>=	is greater than or equal to	5>=8 returns false
<=	is less than or equal to	5<=8 returns true

Logical Operator

Operator	Description	Example
&&	and	x=6 y=3 (x < 10 && y > 1) returns true
||	or	x=6 y=3 (x==5 || y==5) returns false
!	not	x=6 y=3 !(x==y) returns true

Program – Variables, Statements, Operators - 1

```
<HTML>
  <TITLE> Displaying Text </TITLE>
  <BODY>
    <SCRIPT>
          var n1 = 10;
          var n2 = 20;
          sum = n1 + n2;
          document.write("Sum is: " + sum);
    </SCRIPT>
  </BODY>
</HTML>
```

Program – Variables, Statements, Operators - 2

```
<script type="text/javascript">
    s1=12
    s2=28
    sum=s1+s2
    diff=s1-s2
    mult=s1*s2
    div=s1/s2
    document.write("<br>Sum: "+sum)
    document.write("<br>Difference: "+diff)
    document.write("<br>Multiply: "+mult)
    document.write("<br>Division: "+div)
</script >
```

Javascript Dialog Boxes - 1

- Alert Box
  - An alert box is often used if you want to make sure information comes through to the user.
  - When an alert box pops up, the user will have to click "OK" to proceed.

```
<script type="text/javascript">
alert("Hello World!")
</script>
```

Javascript Dialog Boxes - 2

- Prompt Box
  - A prompt box is often used if you want the user to input a value before entering a page.
  - When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.
  - If the user clicks "OK“, the box returns the input value. If the user clicks "Cancel“, the box returns null.

```
<script type="text/javascript">
x=prompt (“Enter your name”, “”)
document.write(“Name: <br>”+x)
</script>
```

Javascript Dialog Boxes - 3

- Confirm Box
  - A confirm box is often used if you want the user to verify or accept something.
  - When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.
  - If the user clicks "OK", the box returns true. If the user clicks "Cancel", the box returns false.

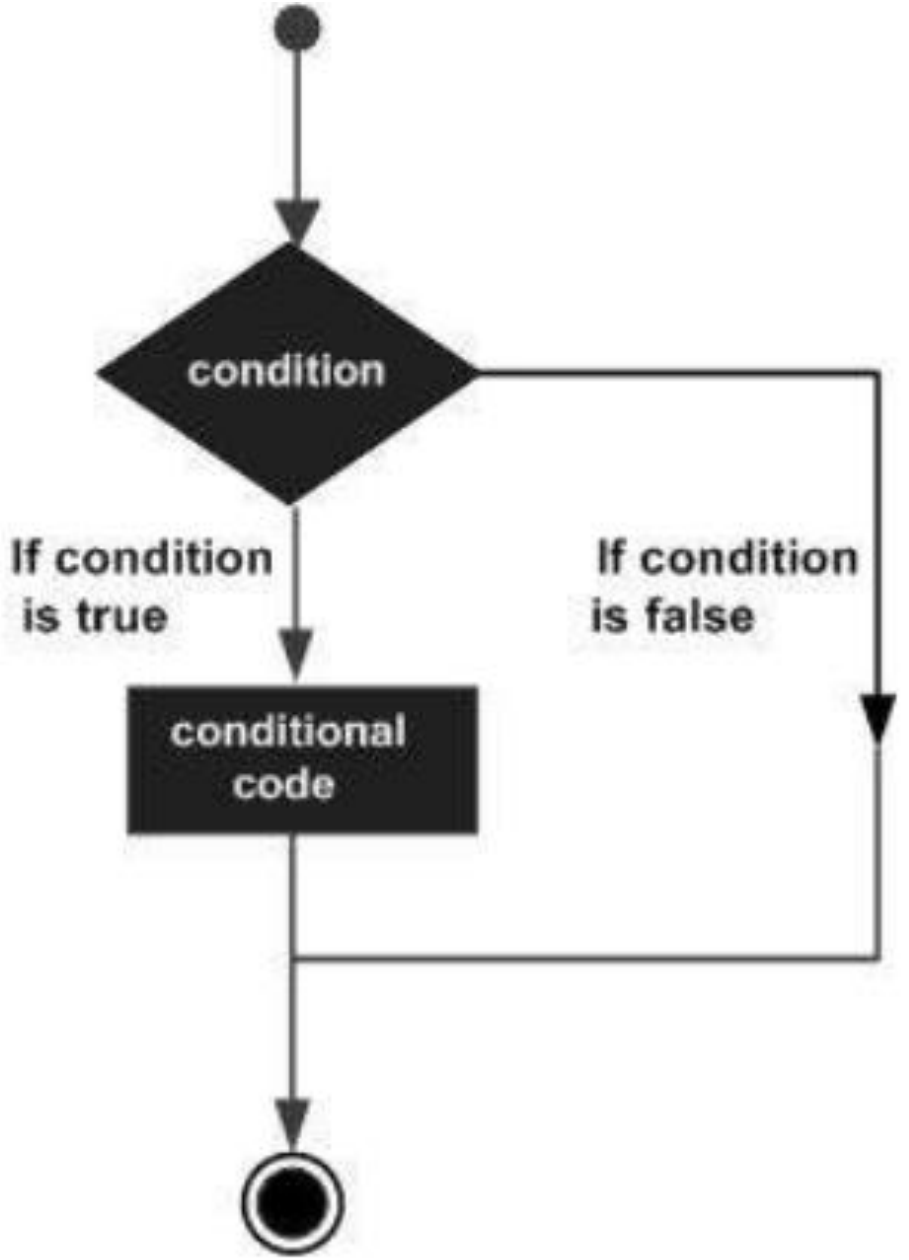
Javascript Dialog Boxes - 3

- Confirm Box – EX

```
<script>
    var txt;
    var r = confirm("Press a button!");
    if (r == true) {
      txt = "You pressed OK!";
    } else {
      txt = "You pressed Cancel!";
    }
document.write(txt);
</script>
```

Controlling Program Flow

- Conditions: Making Decisions – 2 Ways
  - if … else statement
  - Switch ....case



Controlling Program Flow

- JavaScript supports the following forms of **if..else** statement
  - if statement
  - if...else statement
  - if...else if... statement.

- Syntax
  - if(expression) {

    Statement(s) to be executed if true

    }

Example 1

```
<SCRIPT>
  var n1 = 10;
  var n2 = 20;
  var n3 = n1 + n2;
  document.write("Sum : " + n3);
  document.write("<br>Sum of " + n1 + " and " + n2 + " is " + n3);
   document.write("<br>")
  if(n3 >= 30)
     document.write("Greater or equals 30");
  else
      document.write("Condition Not Satisfied");
</SCRIPT>
```

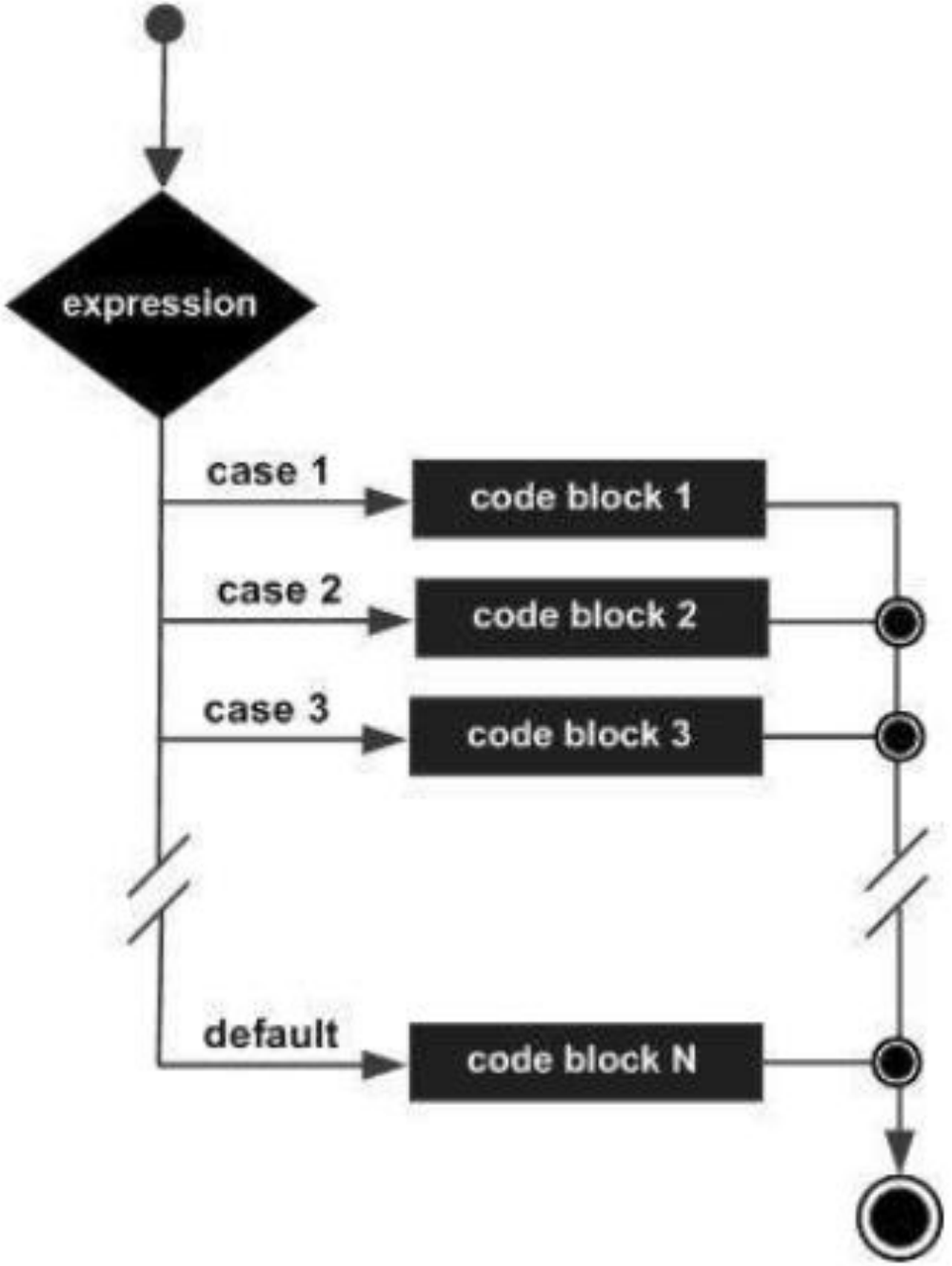
Example 2

```
<SCRIPT>
  var n = parseInt(prompt("Enter a number"));
  document.write("You Entered: " + n + "<br>");
  console.log("You Entered: " + n);
  if(n > 0)
     document.write("Greater than 0");
  else if(n < 0)
      document.write("Less than 0");
  else
      document.write("Equals 0");
</SCRIPT>
```

Controlling Program Flow

- switch ....case

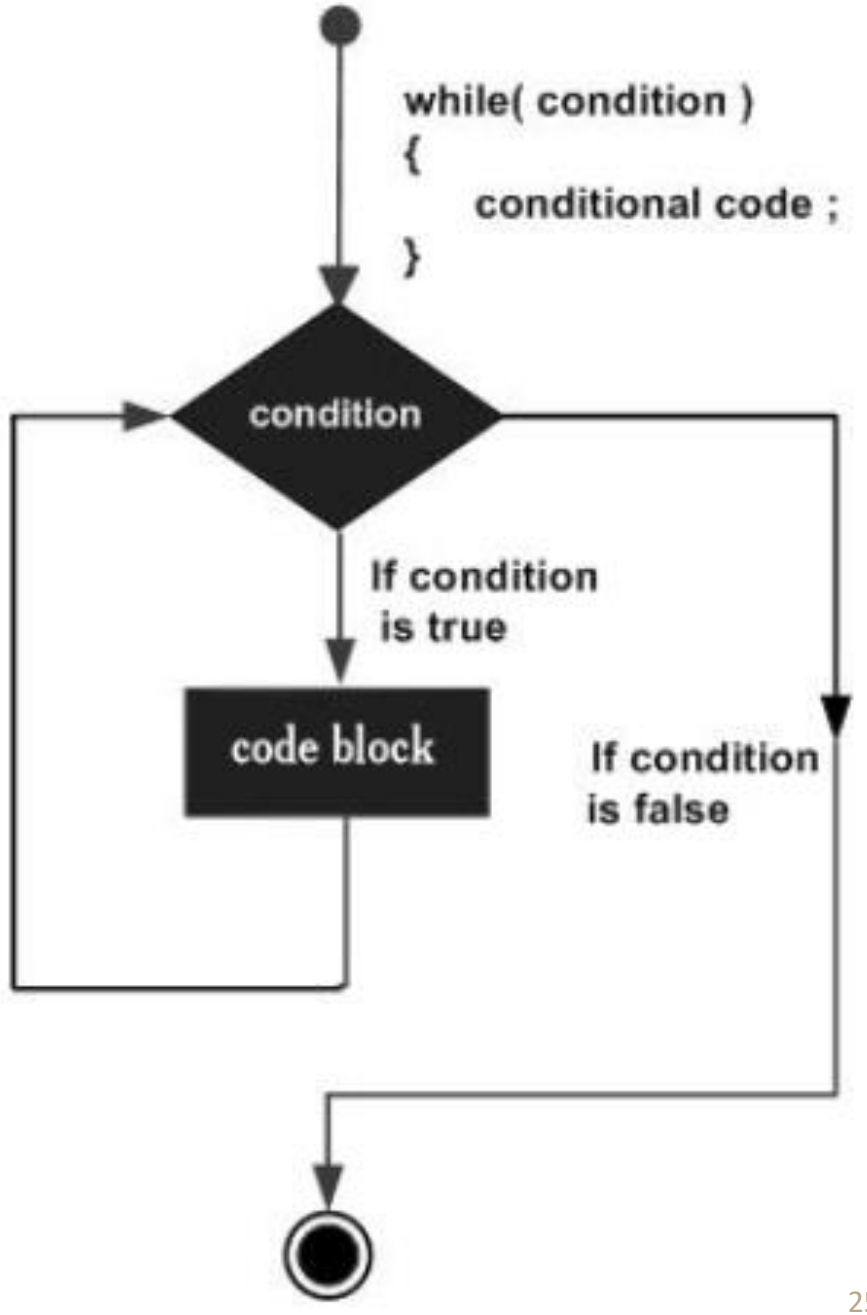
```
switch (expression) {
    case condition 1:
     statement(s);
     break;
    case condition 2:
     statement(s);
     break;
    ...
    case condition n:
     statement(s);
     break;
    default: statement(s)
}
```



```
<SCRIPT>
   var grade = 'A';
   switch (grade) {
     case 'A':
      document.write("Got A");
      break;
     case 'B':
      document.write("Got B");
      break;
     case 'C':
      document.write("Got C");
      break;
     case 'D':
      document.write("Got D");
      break;
     case 'F':
      document.write("Failed");
      break;
     default:
      document.write("Unknown grade")
   }
</SCRIPT>
```

Loop

- Used to perform an action repeatedly till satisfied condition meets.
- 3 Types of Loops
  - While loop
  - Do While loop
  - For loop
- These loops have
  - Initialization statement
  - Condition statement
  - Update (increment or decrement) statement



While Loop

```
<script>
var count = 1;
document.write("Starting Loop");
while (count <= 10) {
    document.write("<br>Count is : " + count);
    count++;
}
document.write("<br>Loop stopped!");
</script>
```

Starting Loop
Count is : 1
Count is : 2
Count is : 3
Count is : 4
Count is : 5
Count is : 6
Count is : 7
Count is : 8
Count is : 9
Count is : 10
Loop stopped!

Check - do while loop

```
//do - while
var count = 1;
do{
    document.write("<br>Count is : " + count);
    count++;
} while(count <=10);
```

Loop – For loop

- Syntax

```
for (initialize; condition; iteration) {
        Statement(s) to be executed if test condition is true
}
```

- Ex

```
for (i=1; i<=10; i++) {
        document.write("<br>" + i);
}
```

Array

- The **Array** object lets you store multiple values in a single variable.
- If you want to store values, either create 3 variables like shown

```
sport1 = “Football";
sport2 = “Tennis";
sport3 = “Cycling";
```

- Or use single array to store these 3 values
  - var sports = new Array("Football", "Tennis", "Cycling");
  - var sports = ["Football", "Tennis", "Cycling"];

```
<script>
   var sports = new Array( "Football", "Tennis", "Cycling”);
   document.write(sports[0]);
   document.write(sports[1]);
   document.write(sports[2]);
   var count = sports.length;
   // loop through array elements
   for(i=0; i< count; i++)
   {
      document.write("<br>Index " + i + " is " +      sports[i]);
   }
</script>
```

Javascript Strings

- Used for storing and manipulating text
- Zero or more characters within quotes.

```
/* String : J  a  v  a  s  c  r  i  p  t
   Index  : 0  1  2  3  4  5  6  7  8  9  */
```

```
var myText = "Javascript";
document.write("<br>" + myText.length);
document.write("<br>" + myText.charAt(4));
document.write("<br>" + myText.indexOf("va"));
document.write("<br>" + myText.substr(0,4));
document.write("<br>" + myText.toUpperCase());
document.write("<br>" + myText.toLowerCase());
```

Functions

- group of reusable code which can be called anywhere in program.
- This eliminates the need of writing the same code again and again
- Divide a big program code into a number of small and manageable functions.
- Syntax

```
function functionname(parameter-list) {
        //statements
}
```

Functions

- We can Categorize function on the basis of parameters and return type

Type	Parameters	Return Type
1	N	N
2	Y	N
3	N	Y
4	Y	Y

Function Type 1 – No Parameters & No Return Type

- How many times you call this function, you will get the same result.

Ex:

```
<script>
        // function defination
        function callme() {
                alert("Hello there");
        }
        callme();     // calling a function
        callme();
</script>
```

Function Type 2 – With Parameter & No Return Type

```
<head><script>
        function callme(name)
        {
                document.write("Hello, " +
name);
        }
        function f2(n1, n2) {
                var sum = n1 + n2;
                document.write(sum);
        }
</script> <head>
```

```
<body><script>
        callme("Sam");
        callme("Hare")
        f2(2,5);
        f2(10, 20);
</script></body>
```

Function Type 3
– No Parameter & Return Type

```
<head><script>
        function f3() {
                n1 = 2; n2 = 12;
                var sum = n1 + n2;
                return sum;
        }
</script></head>
<body><script>
        var returned_sum = f3();
        document.write(returned_sum);
        document.write(f3());
</script></body>
```

Function Type 4
– With Parameter & Return Type

```
<body><script>
        function f3(n1, n2) {
                var sum = n1 + n2;
                return sum;
        }
        var returned_sum = f3(10, 20);
        document.write(returned_sum);
        document.write(f3(20, 30));
</script></body>
```