



# CUI Abbottabad

Department of Computer Science

## Web Technologies

Lecture -11-[CLO-2]  
Javascript-Part4

# Agenda

---

- Introduction to control statements
- Control/Selection statements
  - If-statement
  - If-else statement
  - If-else if-else statement
  - Conditional operator
  - Switch
- Repetition statements
  - Loops
  - For
  - While
  - Do-while
  - Counter-controlled loops
  - Sentinel controlled loops

# INTRODUCTION TO CONTROL STATEMENTS

---

- ▶ All scripts can be written in terms of only three control statements
  - ▶ sequence
  - ▶ selection
  - ▶ repetition

# SELECTION STATEMENTS AND THEIR TYPES

---

- ▶ JavaScript supports selection statements which are used to perform different actions based on different conditions.
- ▶ A selection statement refers to a piece of code that does one ACTION based on one condition, and another ACTION based on another condition.
- ❖ **if statement**  
use this statement to execute some code only if a specified condition is true
- ❖ **if...else statement**  
use this statement to execute some code if the condition is true and another code if the condition is false
- ❖ **if...else if....else statement**  
use this statement to select one of many blocks of code to be executed
- ❖ **switch statement**  
use this statement to select one of many blocks of code to be executed

# IF STATEMENTS

---

## Syntax

**if** (*condition*)

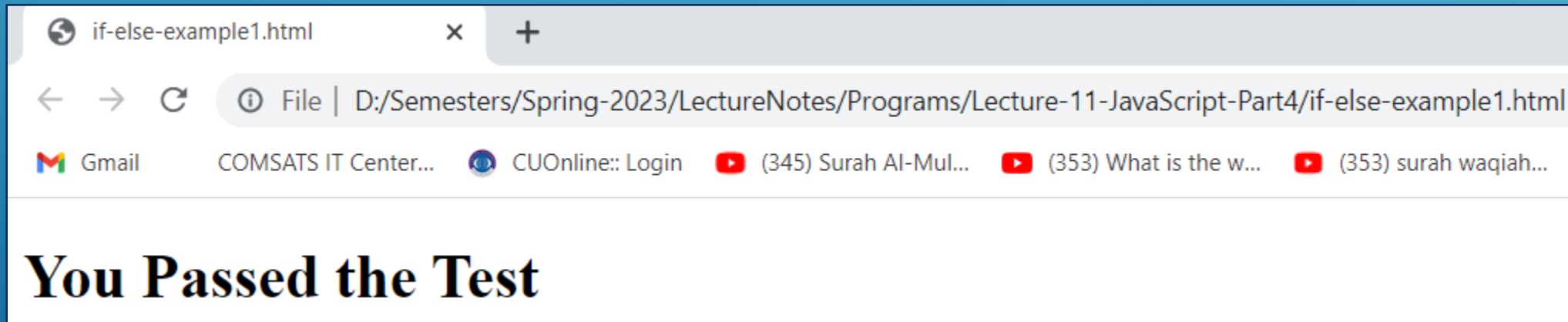
```
{  
  //code to be executed if condition is true  
}
```

- “if” is a reserved word and always written in lowercase letters otherwise syntax error occurred
- “condition” will be any user defined actual condition that is being tested.
  - Condition can be a simple variable name
  - Or it can be any test expression the result of which can be true or false
  - Usually, comparison and logical operators are used in condition
- Inside {} **curly** brackets (or braces), we specify one or more statements which will be executed only if the condition is satisfied/true.
- If inside the curly brackets you want to write only single statement, then in this case use of {} is optional.
- If inside the curly brackets you want to write more than one statement, then use of {} is compulsory.

# IF STATEMENT-EXAMPLE-1

```
1 <html>
2 <body>
3 <script type="text/javascript">
4     var marks = 50;
5     if( marks > 49 )
6     {
7         document.write("<b>You Passed the Test</b>");
8     }
9 </script>
10 </body>
11 </html>
```

Listing-11.1-if-else-example1.html

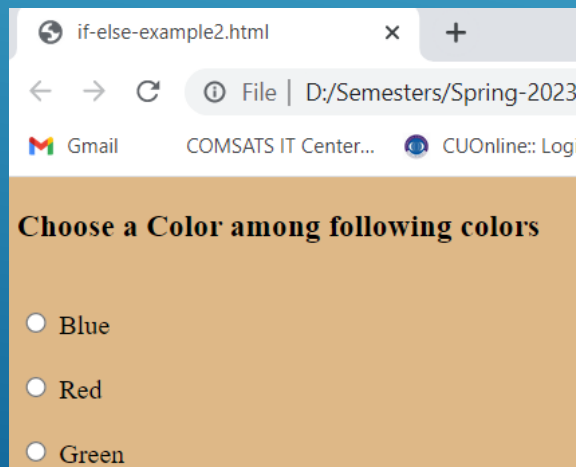


Output-11.1-if-else-example1.html

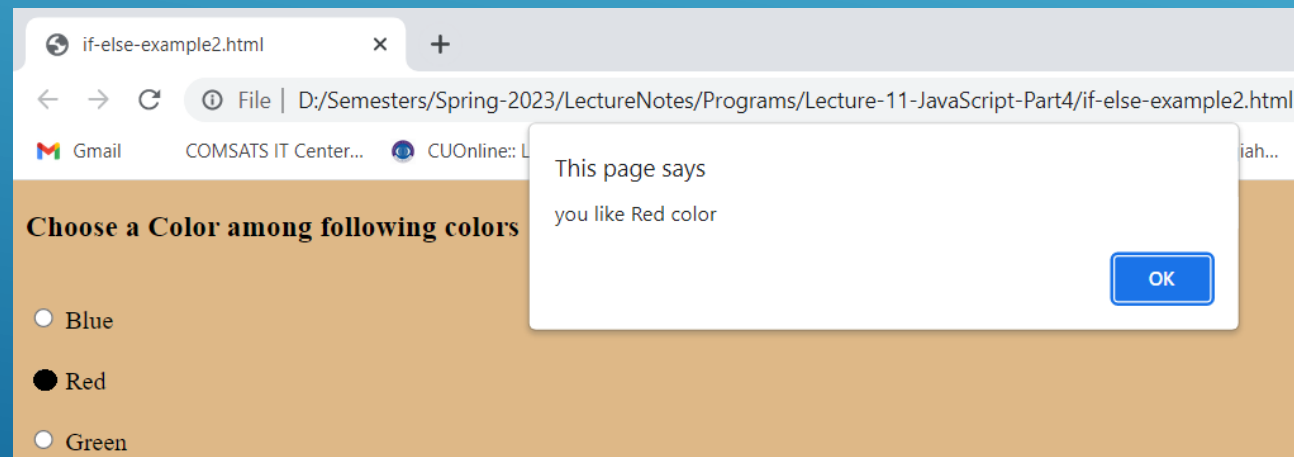
# IF STATEMENT-EXAMPLE-2

Listing-11.2-if-else-example2.html

```
1  <html>
2  <body bgcolor="burlywood">
3
4      <script type="text/javascript">
5
6          function favcolor(mycolor)
7          { alert("you like "+mycolor+" color"); }
8
9      </script>
10 <h3>Choose a Color among following colors</h3><br/>
11 <input type="radio" name="color" value="Blue" onclick=favcolor(this.value); /> Blue <br /><br />
12
13 <input type="radio" name="color" value="Red" onclick=favcolor(this.value); /> Red <br /><br />
14
15 <input type="radio" name="color" value="Green" onclick=favcolor(this.value); /> Green
16
17 </body>
18 </html>
```



output-11.2 (a)



output-11.2 (b)

# IF STATEMENT-EXAMPLE-2-EXPLANATION

---

- ▶ When the program executed in browser then it ask us to select a color among given radio buttons Line#11,Line#13, Line#15
- ▶ Note there is **onclick=myfavcolor(this.value);** in each of tag of input type radio.
  - ▶ Onclick is an event which works only when user clicks this radio button: as the user clicks this button then browser will call the function “myfavcolor() and send the relevant color value of this button using **(this.value)**
  - ▶ After click of radio button the control come in function at Line#6-7.
  - ▶ In JavaScript the “function” keyword is used before function name.
  - ▶ Function favcolor() will receive the value of radio button that has been clicked and then display the result using alert() dialog box
- ▶ Listing-11.2 is showing an example of an event(onclick) occurred on html radio buttons:
- ▶ Send the event to user-defined function myfavcolo() JavaScript
- ▶ In response JavaScript handle the event and display the result



# IF-ELSE STATEMENTS-AND-EXAMPLE-3

if...else statement is used to execute some code if a condition is true and another code if the condition is not true.

## Syntax

if (condition)

```
{ // code to be executed if condition is true }
```

else

```
{ // code to be executed if condition is not true }
```

- if the condition is true the code in the if block will be executed.
- If the condition is false the code in the else block will be executed.
- The else part tells the browser what to do if the condition is not true.

```
1 <html>
2 <head></head>
3 <body>
4   <script type="text/javascript">
5
6     var age;
7     age=window.prompt("Please enter you age: ");
8     age=parseInt(age);
9     if( age > 17 )
10    {
11      document.write("<b>Eligible for ID card</b>");
12    }
13    else
14    {
15      document.write("<b>Not eligible for ID card</b>");
16    }
17
18  </script>
19 </body>
20 </html>
```

Listing-11.3-if-else-example3.html

# IF-ELSE STATEMENTS-AND-EXAMPLE-3

The screenshot shows a web browser window with the address bar displaying the file path: `D:/Semesters/Spring-2023/LectureNotes/Programs/Lecture-11-JavaScript-Part4/if-else-example3.html`. The browser's address bar also shows the file name `if-else-example3.html`. The browser's tabs include `Gmail`, `COMSATS IT Center...`, `CUOnline:: L`, `surah kahf - Google...`, and `surah waqiah...`. The browser's toolbar shows the `File` menu, `Back`, `Forward`, `Refresh`, `Home`, `Search`, `Star`, `Print`, `Fullscreen`, and `More` buttons. The browser's content area displays a form with the text `Please enter you age:` and a text input field containing the value `19`. Below the input field are two buttons: `OK` and `Cancel`. A blue callout box with the text `As user clicks ok button, the alert box show the result according to if-else conditions` points to the `OK` button. An arrow points from the `OK` button to the text `Eligible for ID card` in the browser's status bar.

This page says  
Please enter you age:  
19

OK Cancel

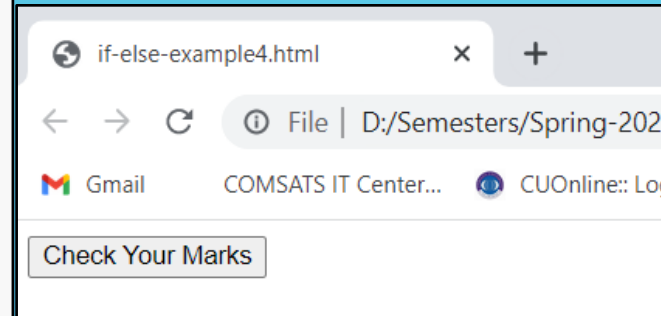
As user clicks ok button, the alert box show the result according to if-else conditions

Eligible for ID card

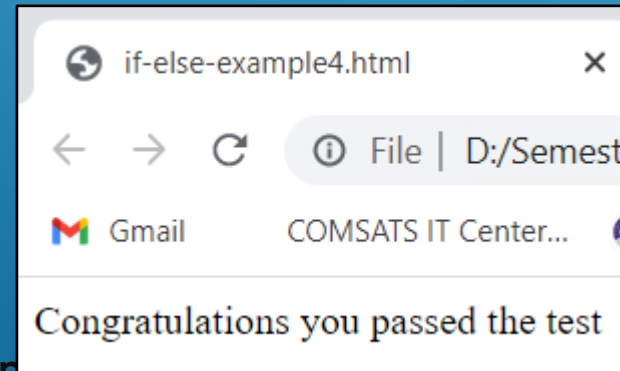
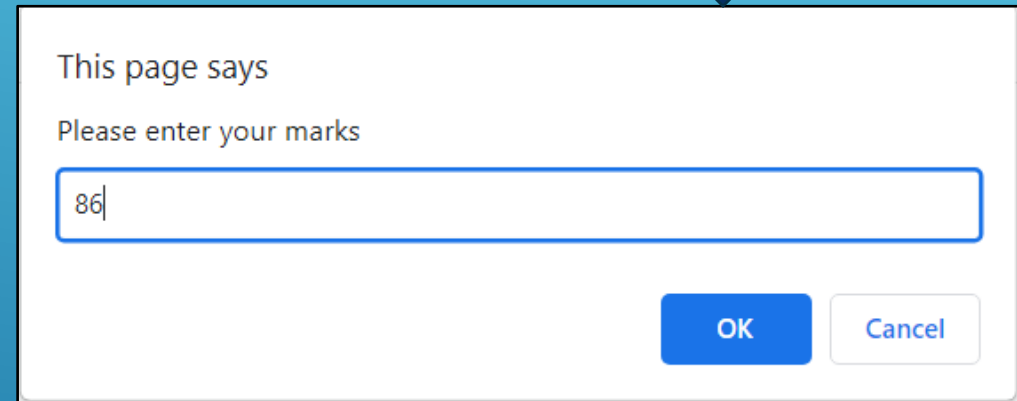
# IF-ELSE STATEMENTS-AND-EXAMPLE-4

```
1 <html>
2 <body>
3 <button onclick="result()">Check You Marks</button>
4
5 <script>
6   function result()
7   {
8     var x,z;
9     var y=prompt("Please enter your marks","");
10    y=parseInt(y);
11    if (y>=50)
12    {
13      x="Congratulations you passed the test";
14      document.write(x)
15    }
16    else
17    {
18      x="Sorry you are failed in the test";
19      document.write(x)
20    }
21  }
22 </script>
23 </body>
24 </html>
```

Listing-11.4-if-else-example4.html



Button Clicks  
displayed input box



Result displayed  
according to if-else  
condition

# IF-ELSE-IF-ELSE STATEMENTS

---

if....else if...else statement is used to select one of several blocks of code to be executed. It is basically a nested if.

Syntax

**if (*condition1*)**

```
{  
  code to be executed if condition1 is true  
}
```

**else if (*condition2*)**

```
{  
  code to be executed if condition2 is true  
}
```

**else**

```
{  
  code to be executed if neither condition1 nor condition2 is true  
}
```

# IF-ELSE-IF-ELSE STATEMENT-EXAMPLE-5

```
1 <!DOCTYPE html>
2 <html><head><title>if-else-example-5</title></head>
3 <body>
4 <script type="text/javascript">
5   var marks = 53;
6   if( marks >= 50 && marks <= 59)
7   {
8     document.write("<b>Just Pass</b>");
9   }
10  else if( marks >= 60 && marks <= 69 )
11  {
12    document.write("<b>Pass with Grade B</b>");
13  }
14  else if( marks >= 70 && marks <= 80)
15  {
16    document.write("<b>Pass with Grade A</b>");
17  }
18  else if( marks > 80 && marks <= 100)
19  {
20    document.write("<b>Outstanding</b>");
21  }
22  else if(marks < 50)
23  {
24    document.write("<b>Fail</b>");
25  }
26 </script>
27 </body>
28 </html>
```

Listing-11.5-if-else-example5.html

# IF-ELSE-IF-ELSE STATEMENT-EXAMPLE-6

```
1 ▼ <!DOCTYPE html><html>
2 ▼ <body>
3   <input type="button" onclick="favbook()" value="Your Fav Book">
4 ▼ <script type="text/javascript">
5   function favbook()
6 ▼ {   var x;
7     var book=prompt("your favorite book","");
8     if( book == "history" )
9 ▼   {   x="You Like History"
10      document.write(x);
11    }
12    else if( book == "maths" )
13 ▼    {
14      x="You Like Maths"
15      document.write(x);
16    }
17    else if( book == "economics" )
18 ▼    {
19      x="You Like Economics"
20      document.write(x);
21    }
22    else
23 ▼    {
24      x="Unknown Book"
25      document.write(x);
26    }
27  }
28 </script></body></html>
```

Listing-11.6-if-else-example6.html

# NESTED IF-ELSE STATEMENT-EXAMPLE-7

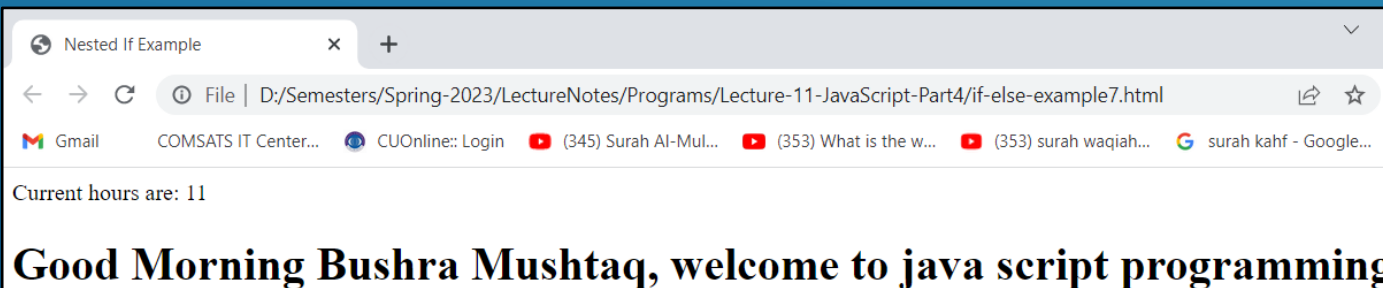
```
1 <html>
2 <head>
3 <title>Nested If Example</title>
4 <script type="text/javascript">
5     var name;
6     var now = new Date();
7     var hour = now.getHours();
8
9     name=window.prompt("please Enter your name");
10    if(hour<12)
11        document.write("<h1>Good Morning");
12        if(hour>=12)
13        {
14            hour = hour -12;
15            if(hour<6)
16                document.write("<h1>Good Afternoon");
17            if(hour>=6)
18                document.write("<h1>Good Evening");
19        }
20    document.writeln(name+", welcome to java script programming</h1>");
21 </script>
22 </head>
23 <body>
24 </body>
25 </html>
```

Listing-11.7-if-else-example7.html

This page says

please Enter your name

OK Cancel



# CONDITIONAL (TERNARY) OPERATOR (?:)

- ▶ Returns one of two expressions depending on a condition

## Syntax

`test ? expression1 : expression2`

## Parameters

- ▶ Test Any Boolean expression.
- ▶ Expression1 An expression returned if *test* is **true**. More than one expressions may be linked by a comma expression.
- ▶ Expression2 An expression returned if *test* is **false**. More than one expression may be a linked by a comma expression
- ▶ The `?:` operator can be used as a shortcut for an **if...else** statement. It is typically used as part of a larger expression where an **if...else** statement would be awkward.
- ▶ For example:

```
var now = new Date();
```

```
var greeting = "Good" + ((now.getHours() > 17) ? " evening." : " day.");
```

Example: Listing 11.8-conditional\_op\_example.html



# Switch Statement

---

- The switch statement is used to perform different action based on different conditions.
- The if else if is not always the best solution, especially when all of the branches depend on the value of a single variable.
- a switch statement handles exactly this situation, and it does so more efficiently than repeated if...else if statements.
- **Switch statement is suitable when options are limited and different**
- The basic syntax of the switch statement is to give an expression to evaluate and several different statements to execute based on the value of the expression.
- The interpreter checks each case against the value of the expression until a match is found. If nothing matches, a default condition will be used.
- The **break** statements indicate to the interpreter the end of that particular case.
- If they were omitted, the interpreter would continue executing each statement in each of the following cases till it catch break statement.

# Switch Statement Syntax-Example-1

```
switch(n)
{
case 1:
    execute code block 1
    break;
case 2:
    execute code block 2
    break;
// so on all cases
default:
    // code to be executed if n
    //is different from case 1 and 2
}
```

```
1  <!DOCTYPE html>
2  <html>
3  <head><title>Switch-Example-1</title></head>
4  <body>
5  <script type="text/javascript">
6  var grade=window.prompt("enter letter grade"); //character
7  switch (grade)
8  {
9      case 'A': document.write("Good job");
10         break;
11     case 'B': document.write("Pretty good");
12         break;
13     case 'C': document.write("Passed");
14         break;
15     case 'D': document.write("Not so good");
16         break;
17     case 'F': document.write("Failed");
18         break;
19     default: document.write("Unknown grade")
20 }
21 </script>
22 </body>
23 </html>
```

Listing-11.9-switch-example1.html

# Switch-Example-2

```
1  <html><head></head>
2  <body>
3      <button onclick="checkday()">Click Here</button>
4
5      <script type="text/javascript">
6          function checkday(y)
7          {   var x;
8              var y=prompt("Please enter a day name","");
9
10             switch (y)
11             {   case "monday":
12                 case "tuesday":
13                 case "wednesday":
14                     alert('More Busy Days');
15                     break;
16                 case "thursday":
17                 case "friday":
18                     alert('Less Busy Days');
19                     break;
20                 default:
21                     alert('Holiday');
22                     break;
23             }
24         }
25     </script></body></html>
```

Listing-11.10-Switch-example2.html

# Switch Statement Syntax-Example-3

```
1 <html>
2 <head>
3   <script type="text/javascript">
4     var choice;
5     var startTag;
6     var endTag;
7     var validInput=true;
8     var listType;
9     choice=window.prompt("Select a list style:\n"+"1(numbered), 2(lettered, 3(roman numbered)","1");
10    switch(choice)
11    {   case "1":
12          startTag="<ol>";
13          endTag="</ol>";
14          listType="<h1>Numbered List</h1>";
15          break;
16        case "2":
17          startTag="<ol style='list-style-type:upper-alpha'>";
18          endTag="</ol>";
19          listType="<h1>Lettered List</h1>";
20          break;
21        case "3":
22          startTag="<ol style='list-style-type:upper-roman'>";
23          endTag="</ol>";
24          listType="<h1>Roman Numbered List</h1>";
25          break;
26        default:
27          validInput = false;
28          break;
29    }
```

Listing-11.11-Switch-example3.html

# Switch Statement Syntax-Example-3

```
30     if(validInput===true)
31     {
32         document.writeln(listType+startTag);
33         for(var i=1;i<=3;++i)
34             {document.writeln("<li>List item"+i+"</li>");}
35         document.writeln(endTag);
36     }
37     else
38         document.writeln("Invalid choice"+choice);
39 </script>
40
41
42 </head>
43
44 <body>
45 </body>
46 </html>
```

This page says

Select a list style:

1(numbered), 2(lettered, 3(roman numbered)

OK

Cancel

Listing-11.11-switch-example3.html

Switch-example3.html

File | D:/Semesters/Spring-2023/LectureNotes/Programs/Lecture-11-JavaScript-Part4/Switch-example3.html

Gmail COMSATS IT Center... CUOnline:: Login (345) Surah Al-Mul... (353) What is the w... (353) surah waqiah...

## Roman Numbered List

- i. List item1
- ii. List item2
- iii. List item3
- iv. List item4
- v. List item5
- vi. List item6
- vii. List item7
- viii. List item8
- ix. List item9

# OPERATOR(===) AND(!==)

---

## The Strict Equals (===) and Strict Does Not Equal (!==) Operators

- ▶ As we mentioned JavaScript can convert between types for you. This includes cases in which you're comparing values. For example, the comparison `"75" == 75` yields the value `true` because JavaScript converts the string `"75"` to the number `75` before performing the equality (`==`) comparison.
- ▶ To prevent implicit conversions in comparisons, which can lead to unexpected results, JavaScript provides the **strict equals** (`===`) and **strict does not equal** (`!==`) operators. The comparison `"75" === 75` yields the value `false` because one operand is a string and the other is a number.
- ▶ Similarly, `75 !== 75` yields `true` because the operand's types are not equal, therefore the values are not equal.
- ▶ If you do not use these operators when comparing values to `null`, `0`, `true`, `false` or the empty string (`""`), [javascriptlint.com](http://javascriptlint.com)'s **JavaScript validator** displays warnings of potential implicit conversions.

# JAVASCRIPT LOOPS

---

- ▶ JavaScript performs several types of repetitive operations, called "looping".
- ▶ Loops are set of instructions used to repeat the same block of code till a specified condition returns false or true depending on how you write its test condition.

JavaScript supports different kinds of loops:

1. **for** - loops through a block of code a number of times
  2. **while** - loops through a block of code while a specified condition is true
  3. **do/while** - also loops through a block of code while a specified condition is true
- ▶ To control the loops you can use counter variable that increments or decrements with each repetition of the loop.
  - ▶ The For statements are best used when you want to perform a loop a specific number of times.
  - ▶ The While and do while statements are best used to perform a loop an undetermined number of times.
  - ▶ In addition, you can use the break and continue statements within loop statements.

# THE FOR LOOP

---

- ▶ The For loop is executed till a specified condition returns false. It has basically the same syntax then in other languages.

## Syntax

**for (initialization; condition; iteration)**

**{**

**Statement(s) to be executed if test condition is true**

**}**

It has three important parts

- ▶ **The loop initialization** where we initialize our counter to a starting value. The initialization statement is executed before the loop begins.
- ▶ **The condition statement** which will test if the given condition is true or not. If condition is true then code given inside the loop will be executed otherwise the for loop terminates.
- ▶ **The iteration statement** where you can increase or decrease your counter.
- ▶ You can put all the three parts in a single line separated by a semicolon.



# FOR LOOP EXAMPLE PROGRAMS

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4  <script type="text/javascript">
5
6  for(i = 0; i < 5; i++)
7  {
8  document.write("i = " + i);
9  document.write("<br />");
10 }
11
12 </script>
13 </body>
14 </html>
```

Listing-11.12-for-loop-example1.html

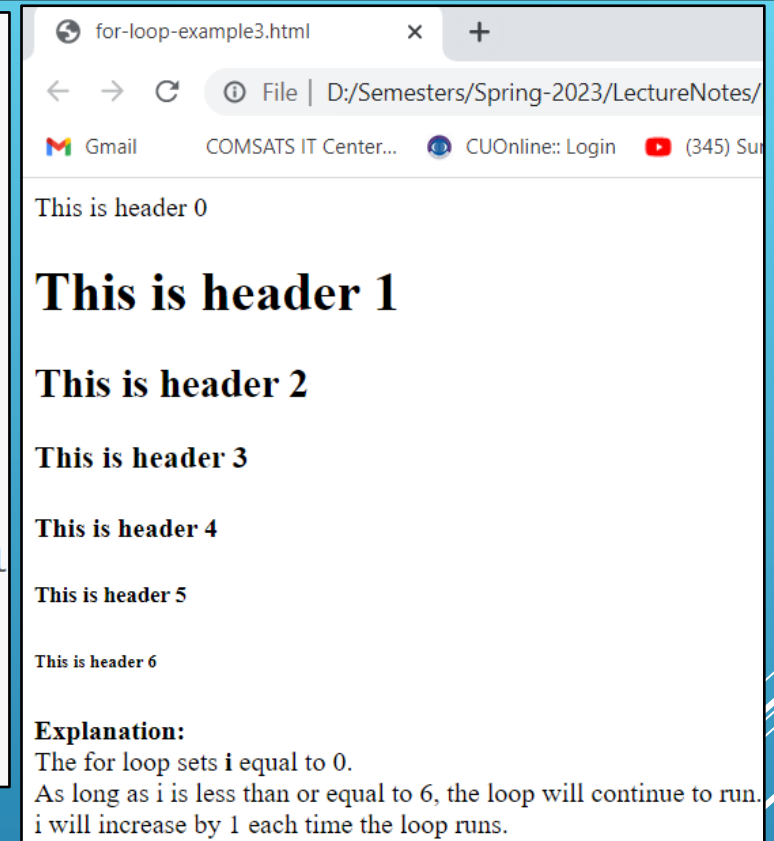
```
1  <!DOCTYPE html>
2  <html>
3      <head><title>For-Loop-Example2</title></head>
4  <body>
5  <script type="text/javascript">
6
7  var i;
8  document.write("Starting Loop" + "<br />");
9  for(i = 0; i < 10; i++)
10 {
11     document.write("The value of i is = " + i );
12     document.write("<br />");
13 }
14 document.write("Loop stopped!");
15
16 </script>
17 </body>
18 </html>
```

Listing-11.13-for-loop-example2.html

# FOR LOOP EXAMPLE PROGRAMS....

```
1 <!DOCTYPE html>
2 <html>
3 <body>
4   <script type="text/javascript">
5     for (i=0; i<=6; i++)
6     {
7       document.write("<h" + i + ">This is header " + i)
8       document.write("</h" + i + ">")
9     }
10  </script>
11
12  <p><b>Explanation:</b><br>
13    The for loop sets <b>i</b> equal to 0.<br>
14    As long as i is less than or equal to 6, the loop will
15    continue to run.<br>
16    i will increase by 1 each time the loop runs.</p>
17 </body>
18 </html>
19
```

Listing-11.14-for-loop-example3.html



Output-11.14-for-loop-example3.html

# FOR LOOP EXAMPLE PROGRAMS....

```
1 <html><head>
2   <style type="text/css">
3     table      {   width:300px;border-collapse;
4                 background-color:lightblue}
5     table, td, th {   border:1px solid black;padding:4px;}
6     th        {   text-align:left;color:white;background-color:darkblue;}
7     tr.oddrow  {   background-color:white;}
8   </style>
9
10  <script type="text/javascript">
11    var amount;
12    var principal=1000.00;
13    var rate=0.05;
14    document.writeln("<table>");
15    document.writeln("<caption>Calculating Compound Interest</caption>");
16    document.writeln("<thead><tr><th>year</th><th>Amount</th></tr></thead><tbody>");
17    for(var year=1;year<=10;++year)
18    {
19      amount=principal*Math.pow(1.0+rate,year);
20      if(year%2!=0)
21      { document.writeln("<tr class='oddrow'><td>"+year+"</td><td>"+amount.toFixed(2)+"</td></tr>");
22      }
23      else
24      {
25        document.writeln("<tr><td>"+year+"</td><td>"+amount.toFixed(2)+"</td></tr>");
26      }
27    }
28    document.writeln("<tbody>");
29    document.writeln("</table>");
30  </script>   </head>
31  <body></body>
32 </html>
```

Listing-11.15-for-loop-example4.html

# FOR LOOP EXAMPLE PROGRAMS....

for-loop-example4.html

File | D:/Semesters/Spring-2023/

Gmail COMSATS IT Center... CUOnline:: Login

Calculating Compound Interest

year	Amount
1	1050.00
2	1102.50
3	1157.63
4	1215.51
5	1276.28
6	1340.10
7	1407.10
8	1477.46
9	1551.33
10	1628.89

Output-11.15-for-loop-example4.html

# WHILE LOOP EXAMPLE PROGRAMS

- ▶ The while statement will execute a block of code while a condition is true.
- ▶ If the condition becomes false, the statements within the loop stop executing and control passes to the statement following the loop.

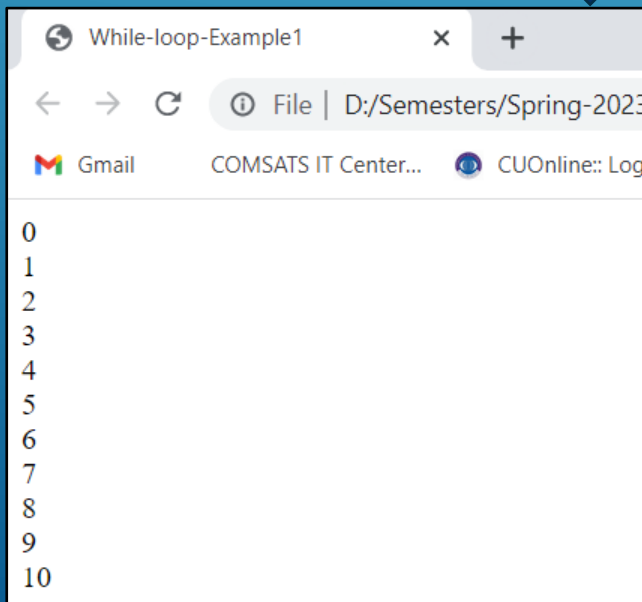
## Syntax:

**while (condition)**

**{**

**code to be executed**

**}**



```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>While-loop-Example1</title>
5  </head>
6  <body>
7  <script type="text/javascript">
8      var i=0;
9      while (i<=10)
10     {
11         document.write(i + "<br>")
12         i++;
13     }
14 </script>
15 </body>
16 </html>
```

# WHILE LOOP EXAMPLE PROGRAMS....

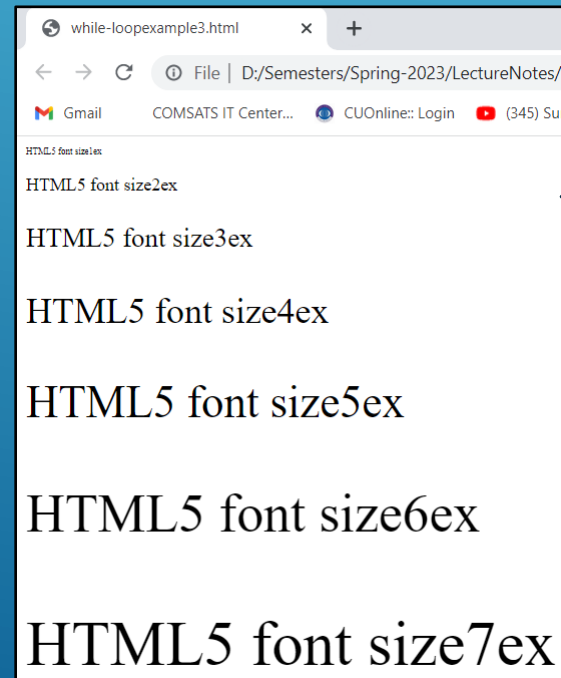
```
1  <!DOCTYPE html>
2  <html><head><title>while-loop-example2</title></head>
3  <body>
4
5  <button onclick="myfunction()">Try it</button>
6
7  <script>
8  function myfunction()
9  {
10  var i=0;
11  while (i<5)
12  {
13  document.write("The number is " + i + "<br>");
14  i++;
15  }
16  }
17  </script>
18  </body>
19  </html>
```

Listing-11.17-while-loop-example2.html

# WHILE LOOP EXAMPLE PROGRAMS...

```
1 <html>
2 <body>
3 <script type="text/javascript">
4
5     var counter=1;
6     while(counter<=7)
7     {
8         document.writeln("<p style='font-size:"+counter+"ex>HTML5 font size"+counter+"ex</p>");
9         ++counter;
10    }
11 </script>
12 </body>
13 </html>
```

Listing-11.18-while-loop-example3.html



# DO WHILE LOOP

The **do...while** loop is similar to the **while** loop except that it will execute a block of code once, and then it will repeat the loop while a condition is true.

## Syntax

```
do
{
    code to be executed
}
```

**While (condition);**

Use a Do While loop to run the same block of code while or until a condition is true.

**This loop will always be executed atleast once, even if the condition is false,** because the statements are executed before the condition is tested.

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4  <script type="text/javascript">
5    i=0;
6    do
7  {
8    document.write("The number is " + i + "<br>");
9    i++;
10 }
11 while (i<=5);
12 </script>
13 </body>
14 </html>
```

Listing-11.19-dowhile-example1.html



# DO WHILE LOOP....

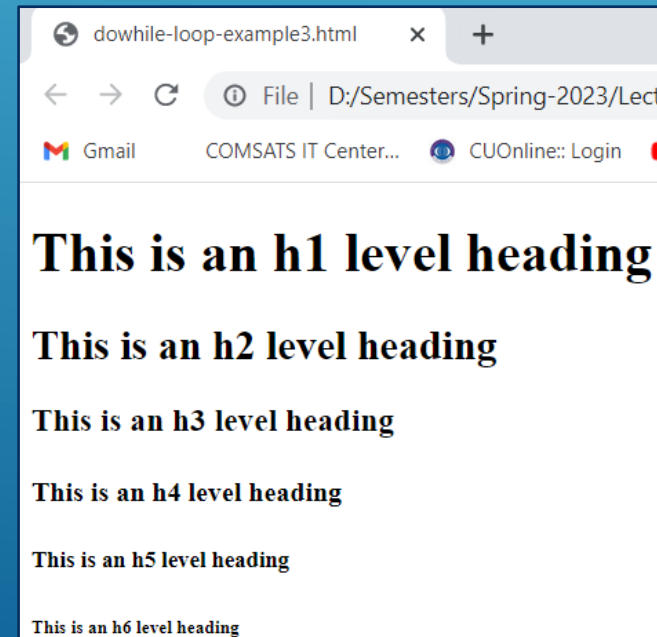
```
1  <!DOCTYPE html>
2  <html>
3  <body>
4  <script type="text/javascript">
5
6  var count = 0;
7  document.write("Starting Loop" + "<br />");
8  do
9  {
10     document.write("Current Count : " + count + "<br />");
11     count++;
12 }
13 while (count < 10);
14 document.write("Loop stopped!");
15
16 </script>
17
18 </body>
19 </html>
```

Listing-11.20-dowhile-example1.html

# DO WHILE LOOP....

```
1 <html><head></head>
2 <body>
3 <script type="text/javascript">
4
5     var counter=1;
6     do
7     {
8         document.writeln("<h"+counter +">This is an h"+counter+"level heading</h"+counter+">");
9         ++counter;
10    }while(counter<=6)
11 </script>
12 </body>
13 </html>
```

Listing-11.21-dowhile-loop-example3.html



# BREAK AND CONTINUE STATEMENTS

---

- ▶ There may be a situation when you need to come out of a loop without reaching at its bottom.
- ▶ There may also be a situation when you want to skip a part of your code block and want to start next iteration of the loop.
- ▶ To handle all such situations, JavaScript provides **break** and **continue** statements.
- ▶ These statements are used to immediately come out of any loop or to start the next iteration of any loop respectively.

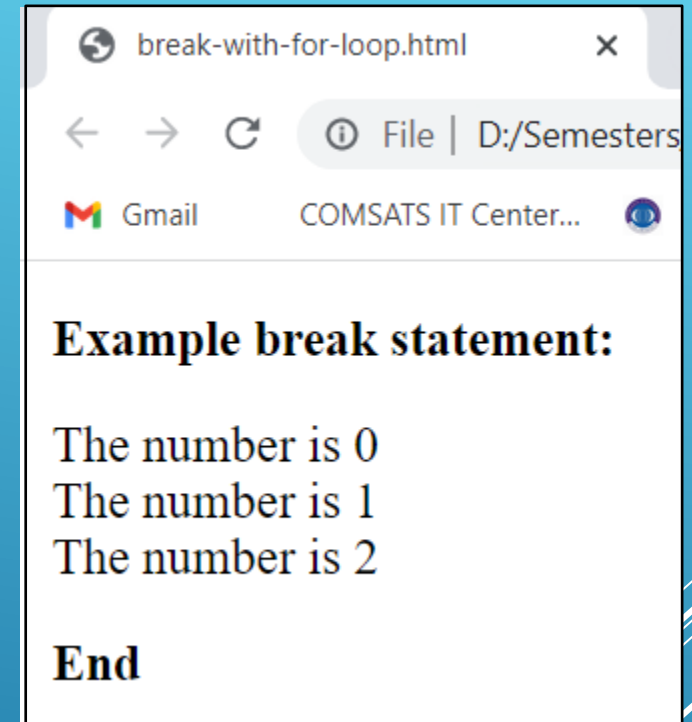
The break statement terminates the current while or for loop and continues executing the code that follows after the loop.

A continue statement terminates the remaining execution of the block of statements in a while or for loop and continues execution of the loop with the next iteration number.

# BREAK WITH FOR LOOP...

```
1 <!DOCTYPE html>
2 <html>
3 <head></head>
4 <body>
5 <script type="text/javascript">
6     document.write("<p><b>Example break statement:</b></p>");
7
8     var i = 0;
9     for (i=0; i<=10; i++)
10    {
11        if (i==3)
12        {
13            break;
14        }
15        document.write("The number is " + i + "<br />");
16    }
17
18    document.write("<p><b>End</b><p>");
19 </script>
20 </body>
21 </html>
```

Listing-11.22-break-with-for-loop.html

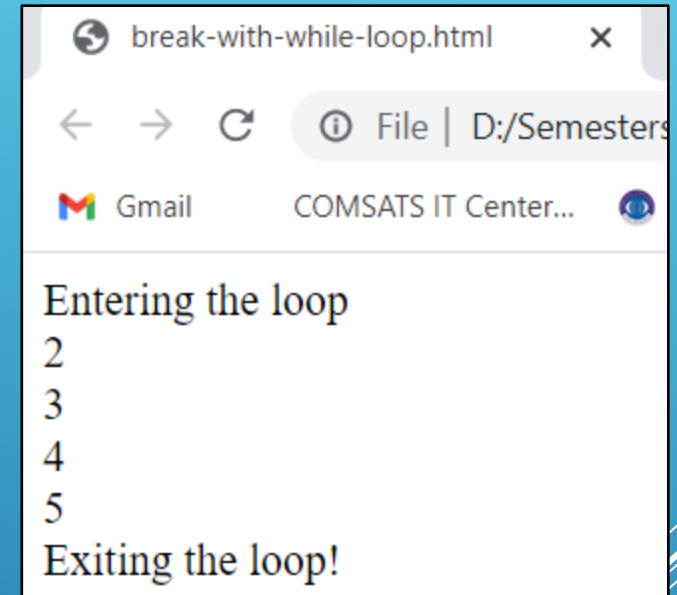


Output-11.22

# BREAK WITH WHILE LOOP...

```
1 <html>
2 <body>
3   <script type="text/javascript">
4
5     var x = 1;
6     document.write("Entering the loop<br /> ");
7     while (x < 20)
8     {
9       if (x == 5)
10      {
11        break; // breaks out of loop completely
12      }
13      x = x + 1;
14      document.write( x + "<br />");
15    }
16    document.write("Exiting the loop!<br /> ");
17
18  </script>
19 </body>
20 </html>
```

Listing-11.23-break-with-while-loop.html

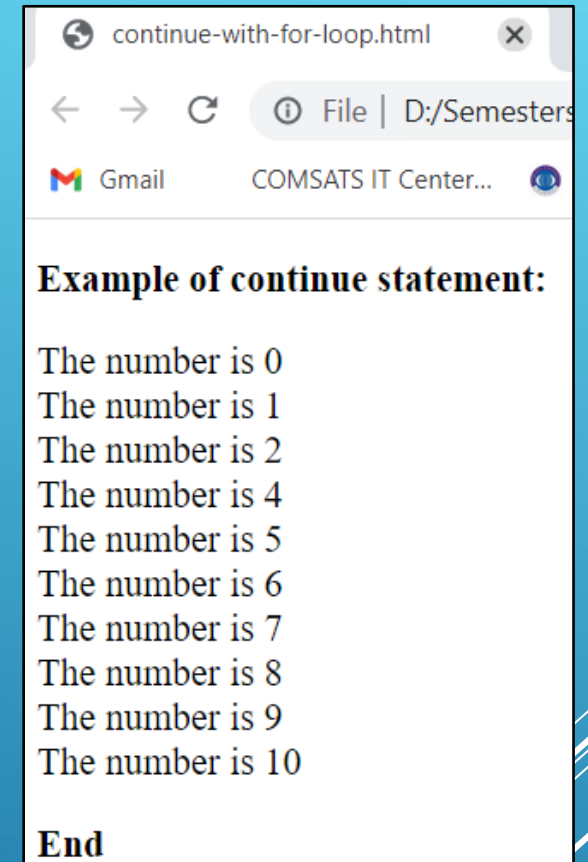


Output-11.23

# CONTINUE WITH FOR LOOP EXAMPLE

```
1 <html>
2 <body>
3 <script type="text/javascript">
4     document.write("<p><b>Example of continue statement:</b><p>");
5
6     var i = 0;
7     for (i=0; i<=10; i++)
8     {
9         if (i==3)
10        {
11            continue;
12        }
13        document.write("The number is " + i + "<br />");
14    }
15
16    document.write("<p><b>End</b><p>");
17 </script>
18 </body>
19 </html>
```

Listing-11.24-continue-with-for-loop.html

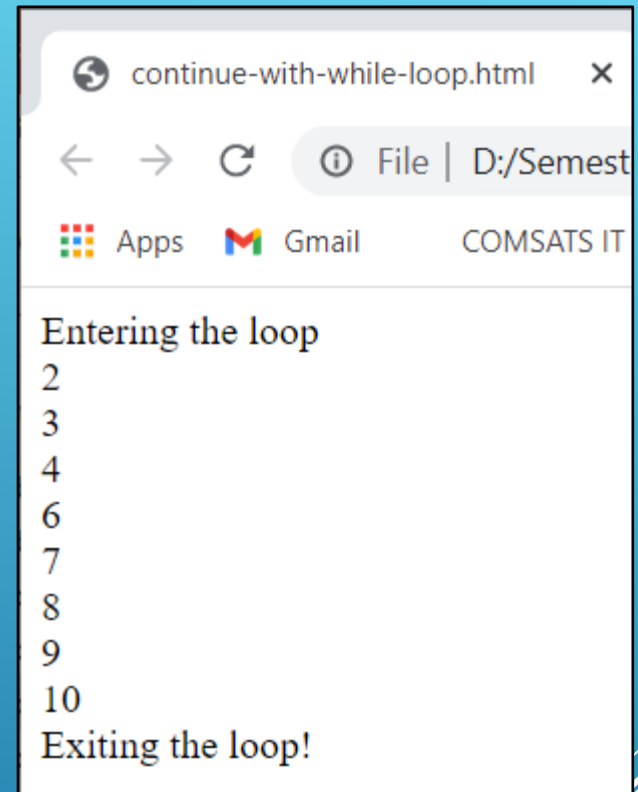


Output-11.24

# CONTINUE WITH WHILE LOOP EXAMPLE

```
1 <html>
2 <body>
3   <script type="text/javascript">
4     var x = 1;
5     document.write("Entering the loop<br /> ");
6     while (x < 10)
7     {
8       x = x + 1;
9       if (x == 5)
10      {
11        continue;
12        // skip rest of the loop body
13      }
14      document.write( x + "<br />");
15    }
16    document.write("Exiting the loop!<br /> ");
17  </script>
18 </body>
19 </html>
```

Listing-11.25-continue-with-while-loop.html



Output-11.25

# COUNTER-CONTROL LOOP

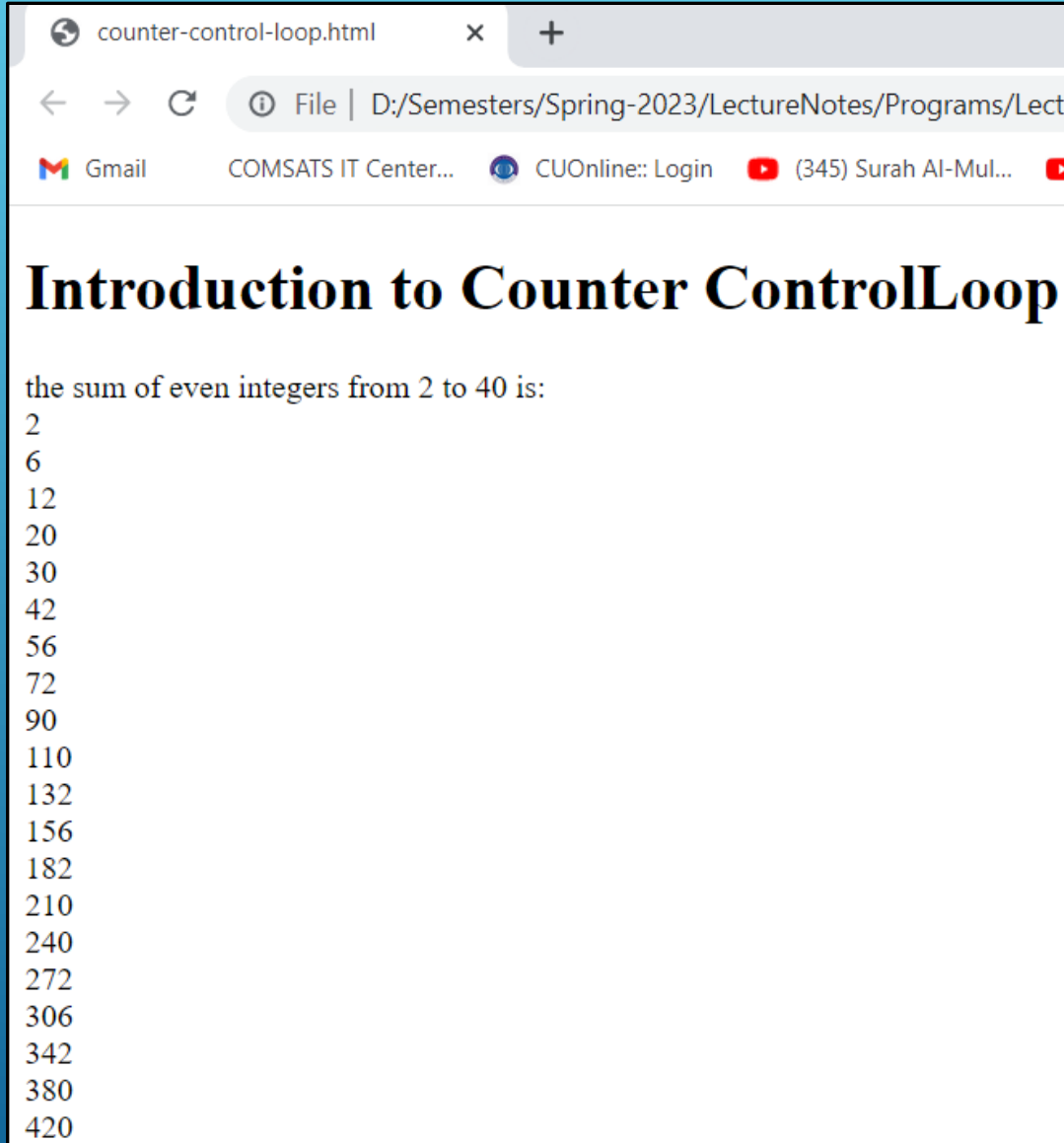
- Often called definite repetition, because the number of repetitions is known before the loop begins executing

```
1  <html>
2  <head>
3      <h1>Introduction to Counter ControlLoop</h1>
4
5      <script type="text/javascript">
6          var sum=0;
7          document.writeln("the sum of even integers from 2 to 40 is: <br/>");
8          for(var number=2; number<=40;number+=2)
9              {
10                 sum+=number;
11                 document.writeln(sum+"<br/>");
12             }
13      </script>
14
15  </head>
16  <body>
17  </body>
18  </html>
```

Listing-11.26-counter-control-loop.html



# COUNTER – CONTROLLED REPETITION

A screenshot of a web browser window. The address bar shows a local file path: D:/Semesters/Spring-2023/LectureNotes/Programs/Lect... The page title is "Introduction to Counter ControlLoop". The main content area displays the text "the sum of even integers from 2 to 40 is:" followed by a list of even numbers from 2 to 420, with each number on a new line. The browser's taskbar at the bottom shows icons for Gmail, COMSATS IT Center..., CUOnline:: Login, and a YouTube video titled "(345) Surah Al-Mul...".

```
counter-control-loop.html x +  
← → ↻ ⓘ File | D:/Semesters/Spring-2023/LectureNotes/Programs/Lect...  
Gmail COMSATS IT Center... CUOnline:: Login (345) Surah Al-Mul...  
  
Introduction to Counter ControlLoop  
  
the sum of even integers from 2 to 40 is:  
2  
6  
12  
20  
30  
42  
56  
72  
90  
110  
132  
156  
182  
210  
240  
272  
306  
342  
380  
420
```

Output-11.26

# SENTINEL CONTROLLED REPETITION

---

- ▶ Sentinel-controlled repetition
  - ▶ Special value called a sentinel value (also called a signal value, a dummy value or a flag value) indicates the end of data entry
  - ▶ Often is called indefinite repetition, because the number of repetitions is not known in advance
- ▶ Choose a sentinel value that cannot be confused with an acceptable input value

# SENTINEL CONTROLLED REPETITION

```
1  <html>
2  <head>
3      <h1>repetition</h1>
4      <script type="text/javascript">
5          var i=0;
6          var total=0;
7          var marks;
8          var grade=0;
9          var counter;
10         var subjnos=0;
11         while(i==0)
12         {
13             marks = window.prompt("Enter another subject's marks");
14             grade=parseInt(marks);
15             total=total+grade;
16             document.writeln("the total marks so far are"+total+"<br>");
17             counter=window.prompt("please enter 0 if you want to continue");
18             i=parseInt(counter);
19             subjnos++;
20         }
21
22         document.writeln("the average is equal to"+total/subjnos);
23
24     </script>
25 </head>
26 <body>
27 </body>
28 </html>
```

Listing-11.27-Sentinel-control-loop.html