



JavaScript

av

Carlos Aristondo

INNEHÅLLSFÖRTECKNING

LEKTION 1 VAD ÄR JAVASCRIPT #

LEKTION 2 VARIABLER #

LEKTION 3 RITA GEOMETRISKA FORMER #

LEKTION 4 KONTROLL STRUKTURER #

LEKTION 5 LOOPAR #

LÄNKAR #

VAD ÄR JAVASCRIPT?

JavaScript är ett nytt script-språk för Web-sidor. Script skrivna i JavaScript kan bäddas in i dina HTML-sidor. JavaScript ger dig många möjligheter att förbättra din HTML-sida med en del intressanta funktioner. Man kan säga att JavaScript är en utökning av HTML snarare än ett eget språk. Grundfunktioner som vi ska gå igenom stegvis snart.

JavaScript Display Möjligheter

JavaScript kan "DISPLAY" data på olika sätt:

- Skrivning i ett HTML-element, med hjälp av **innerHTML** .
- Skrivning i HTML-utgång med **document.write ()** .
- Skrivning i en varningsruta, med användning **window.alert ()**

Använda **innerHTML**

Att få tillgång till ett HTML-element, kan JavaScript använda **document.getElementById (id)** metoden.

Det **id** -attribut definierar HTML-element. Den **innerHTML** egendom definierar HTML-innehåll:

EXEMPEL

```
<!DOCTYPE html>
<html>
  <body>
    <h1>Output med innerHTML</h1>
    <p id="demo"></p>

    <script src="simple.js" type="text/javascript"> // svenska
tecken!!
      document.getElementById("demo").innerHTML = 5 + 6;
    </script>
  </body>
</html>
```

Output med innerHTML

11

Lektion1a.html

Använder document.write ()

För teständamål, är det lämpligt att använda `document.write ()` :

<pre><!DOCTYPE html> <html> <body> <h1>Output med document.write</h1> <script type="text/javascript"> document.write(5 + 6); </script> </body> </html></pre>	JavaScript out 11 Lektion1b.html Javascriptmall.html
--	--

Du kan använda den här filen som JavaScript-mallen. Kallas den som `JavaScriptMall.html`

Med användning window.alert ()

Du kan använda en varningsruta för att visa data:

<pre><!DOCTYPE html> <html> <body> <h1>JavaScript Out</h1> <script src="simple.js" type="text/javascript"> window.alert(5 + 6); </script> </body> </html></pre>	 <p>Lektion1c.html</p>
---	---

Mallen: javaScriptMall.html

Uppgift 1: Skriv ett program som visa ut ditt namn på skärmen.

Spara den först, men byt namn till **lektion1.html**. Öppna dokumentet i webbläsaren och se resultat.

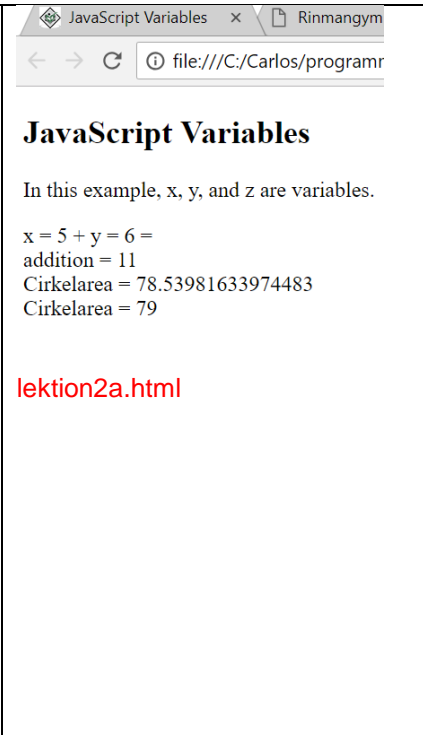
LEKTION 2 VARIABLER

VARIABLER

JavaScript variabler är behållare för lagring av datavärden.

I detta exempel, x, y och z, är variabler: Man deklarerar en variabel med hjälp av

```
var x = 5; // Kommentar Obs! glöm inte semikolon
```

<pre><!DOCTYPE html> <html> <body> <h2>JavaScript Variables</h2> <p>In this example, x, y, and z are variables.</p> <script type = "text/javascript"> var pi = Math.PI; // matematiksbibliotek var x = 5; var y = 6; var radie = 5; var addition = x + y; var cirkelArea = pi*Math.pow(radie,2); document.write(" x = " + x + " + y = " + y + " = " + "
"); document.write(" addition = " + addition + "
"); document.write(" Cirkelarea = " + cirkelArea + "
"); document.write(" Cirkelarea = " + Math.round(cirkelArea) + "
"); </script> </body> </html></pre>	
---	---

```
var x, y, z; // Statement 1
x = 5; // Statement 2
y = 6; // Statement 3
addition = x + y; // Statement
4
cirkelArea =
```

Från exemplet ovan kan du förvänta dig:

- x lagrar värdet 5
- y lagrar värdet 6
- addition lagrar värdet

```
pi*Math.pow(radie,2);
```

11

- cirkelarea = $\pi \cdot \text{radie}^2$

Math.pow(base,
exponenten) = radie^2
basen = radie och exp = 2

Math.round(tal) returnerar
ett avrundat tal.

JAVASCRIPT IDENTIFIERARE

Alla JavaScript **variabler** måste **identifieras** med **unika namn**.

Dessa unika namn är kallade **identifierare**.

Identifierare kan vara korta namn (som x och y) eller mer beskrivande namn (ålder, summa, totalVolume).

De allmänna reglerna för att konstruera namn för variabler (unika identifierare) är:

- Namn kan innehålla bokstäver, siffror, understreck och dollartecken.
- Namn måste börja med en bokstav
- Namn kan också börja med \$ och _ (men vi kommer inte att använda den i denna handledning)
- Namn är skiftlägeskänsliga (y och Y är olika variabler)
- Reserverade ord (som JavaScript sökord) kan inte användas som namn

EN RAD KOMMENTARER

Enda rad kommentaras Mha //.

All text mellan // och slutet av linjen kommer att ignoreras av JavaScript (inte kommer att verkställas).

Detta exempel använder en enda linje kommentar före varje kodrad:

JAVASCRIPT DATATYPER

JavaScript variabler kan hålla tal som 100 och textvärden som "John Doe".

I programmering är textvärden kallas **textsträngar**.

JavaScript kan hantera många typer av data, men nu, tänk bara på siffror och strängar.

Strängar skrivs inuti dubbel- eller enkla citationstecken. Siffror är skrivna utan citattecken.

Om du sätter ett antal inom citationstecken, kommer den att behandlas som en textsträng.

<pre><!DOCTYPE html> <html> <body> <h2>JavaScript Variabler</h2> <p>Strings are written with quotes.</p> <p>Numbers are written without quotes.</p> <p id="demo"></p> <script> var pi = Math.PI; // Math. bibliotek var person = "John Doe"; // strängar var answer = "Yes I am!"; //strängar document.write(pi + "
" + person + "
" + answer); </script> </body> </html></pre>	<p>JavaScript Variables</p> <p>Strings are written with quotes.</p> <p>Numbers are written without quotes.</p> <p>3.14 John Doe Yes I am!</p> <p>lektion2b.html</p>
--	---

Här kan du hitta alla matematiska funktioner: t ex Math.PI, se länken för mer information och exempel. Om du vill ha flera matematiska funktioner kolla länkar längst när.

Uppgift 2: fortsätt med programmet ovan (lektion2.html). Vi har redan gjort addition. Komplera programmet med subtraktion, division och multiplikation. Spara den först, men byt namn till **lektion2.html**. Öppna dokumentet i webbläsaren.

Operatörer	Operatörer enkel	Beskrivning	Exempel
Namn			
uppdagsoperatör	=	Spara värdet till	A=B

		höger om lika tecken i variabeln till vänster om likartat tecken.	
Addition (plus)	+	Lägger till två operander	A+B det blir 15
Subtraktion (minus)	-	Subtraherar andra operand från den första operand	A-B det blir -5
Multiplikation (gång)	*	Multiplifiera båda operanderna	A.B det blir 50
Division (delat)	/	Dela täljare med nämnare	B/A det blir 2
Rester efter division	%	Modulus Operator och återstoden av efter en heltal division	B%A det blir 0

Du kan skriva kortare aritmetiska expressioner

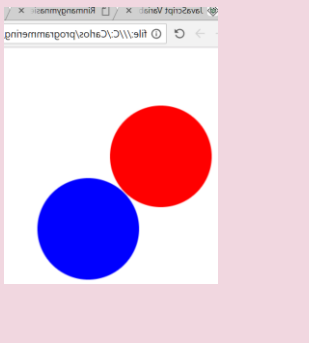
expression	equivalent to...
<code>y += x;</code>	<code>y = y + x;</code>
<code>x -= 5;</code>	<code>x = x - 5;</code>
<code>x /= y;</code>	<code>x = x / y;</code>

Uppgift 2: fortsätt med programmet ovan (lektion2a.html). Vi har redan gjort addition. Komplera programmet med subtraktion, division och multiplikation. Spara den först, men byt namn till [lektion2c.html](#). Öppna dokumentet i webbläsaren och se resultat..

RITA GEOMETRISKA FORMER

När man ska rita något använder man x- och y-koordinater för beskriva var det ska hamna. Hörnet längs upp till vänster har koordinaterna (0,0) = (x,y). det vill säga x-koordinat 0 och y-koordinat 0. När man går höger ökar x-koordinaten och när man går nedåt ökar y-koordinaten. Du behöver bara ha en fil i din mapp, den som heter "simple.js" som hjälper dig att rita figurer på skärmen.

Vi börja med ett enkelt exempel

<pre><!DOCTYPE html> <html> <body> <center><h2>Variables och grafik</h2></center> <script src="simple.js"> var x =100; var y = 150; var radie = 70; circle(x,y,radie,"red"); circle(x+100,y+100,radie,"blue"); </script> </body> </html></pre>	
<pre><script src="simple.js"> var x =100; var y = 150; var radie = 70; circle(x,y,radie,"red"); circle(x+100,y+100,radie,"blue"); </script></pre>	<p>"simple.js" filen som hjälper oss att rita figurer. Filen måste finnas i din mapp som alla dina JavaScript-uppgifter är.</p> <p>// En cirkel behöver fyra argument:</p> <p>// 1. X-koordinat för cirkelns mittpunkt // 2. Y-koordinat för cirkelns mittpunkt // 3. Radie // 4. Färg</p> <p>X= 100 pixlar y= 150 pixlar radie 70 pixlar.</p> <p>X & y ger positionen på skärmen (0,0) ligger i upp till vänster Lektion3a.html</p>

KONTROLL STRUKTURER

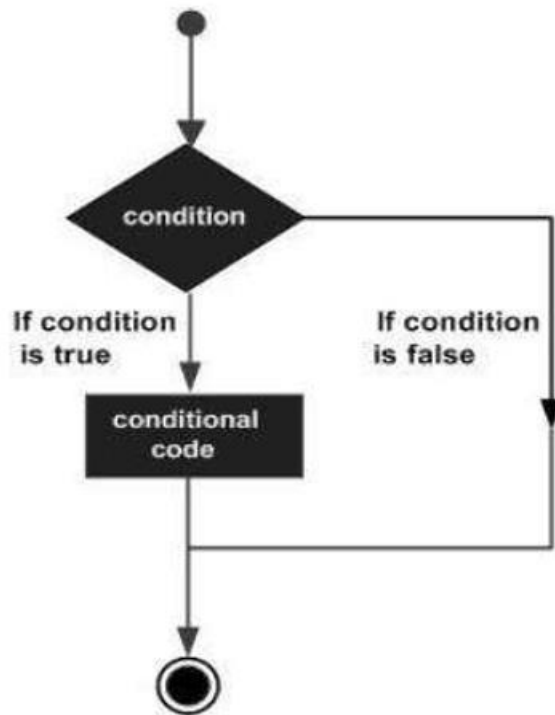
För att ett program ska vara interaktivt och kunna göra olika saker varje gång man kör det så behöver det kunna göra så kallade tester. Det är till exempel vanligt att kod testas om en variabel har ett visst värde, eller om variabeln är större än eller mindre än något värde. Om användaren av ett program till exempel får skriva in sitt personnummer så kan det sparas i en variabel. Programmet kan sedan testa om personnummer är korrekt och i så fall berätta när användaren fyller år etc.

I JavaScript använder man de engelska orden `if` och `else` för att skriva kod som utför olika saker beroende på om ett villkor är uppfyllt eller inte

Kontroll Strukturer

Beslutsfattande strukturer kräver att programmeraren anger ett eller flera villkor som ska utvärderas eller testas av programmet. Det borde vara tillsammans med ett uttalande eller uttalanden som ska utföras om villkoret är fast beslutet att vara sant och eventuellt andra uttalanden som ska utföras om villkoret är bestämt att vara falskt.

Följande bilder är den allmänna formen av en typisk beslutsstruktur som finns i de flesta programmeringsspråk.



Kontrollstruktur är element i källkoden som styr flödet av programkörning. Dom är

- If statement
- If ...else statement
- If...else if ...else statement
- switch case statement
- Conditional Operator

If-satsen

If-satsen är en konstruktion som du kan använda för att styra vad som ska göras i ditt program. Nästan alla program reagerar olika beroende på vad du matar in i dem.

Det tar ett uttryck inom parentes och ett uttalande eller ett block av uttalanden. Om uttrycket är sant utförs uttalandet eller blocket av uttalanden annars ignoreras dessa uttalanden.

Exempel

```
if (expression)
```

```
statement;
```

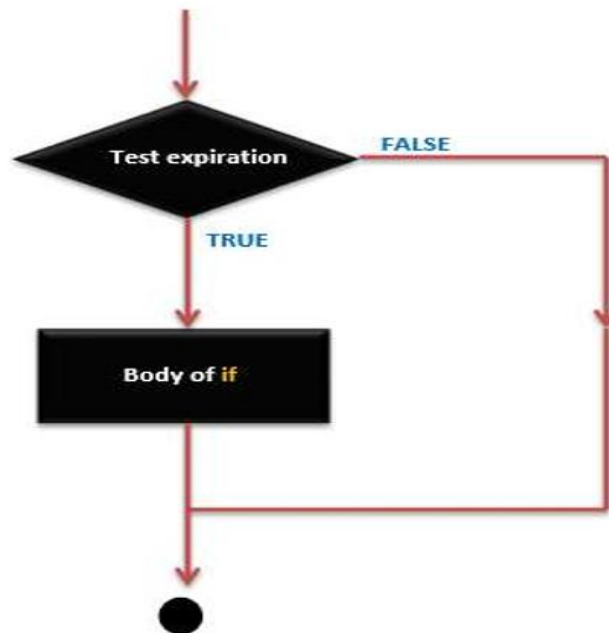
Exempel

```
if (expression)
```

```
{
```

```
Block of statements;
```

```
}
```



Jämförelseoperatorer

operator	description
==	Equal to
!=	Not equal to
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to

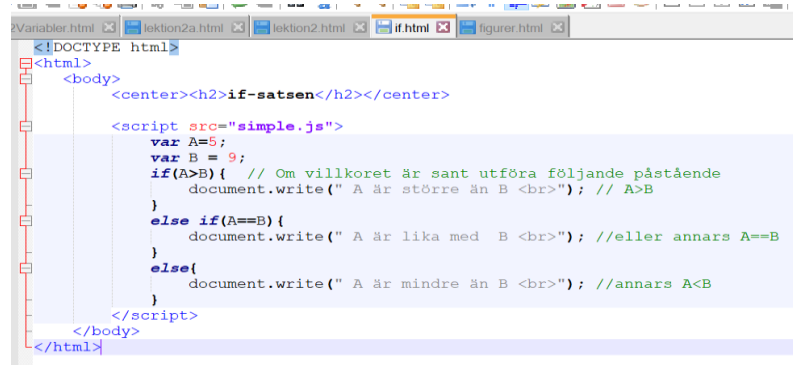
Några exempel:

```
1 (7 == 5)    // evaluates to false
2
3 (5 > 4)     // evaluates to true
4
5 (3 != 2)    // evaluates to true
(6 >= 6)     // evaluates to true
(5 < 5)      // evaluates to false
```

```
<!DOCTYPE html>
<html>
  <body>
    <center><h2>if-satsen</h2></center>
    <script src="simple.js">
      var A=5;
      var B = 9;
      if(A>B){ // Om villkoret är sant utföra följande påstående
        document.write(" A är större än B <br>"); // A>B
      }
      else if(A==B){
        document.write(" A är lika med B <br>"); //eller annars A==B
      }
      else{
        document.write(" A är mindre än B <br>"); //annars A<B
      }
    </script>
```

```
</body>
</html>
Lektion4a.html
```

JavaScript använder man måsvingar, {och }
, när man skriver block av kod som hänger ihop.



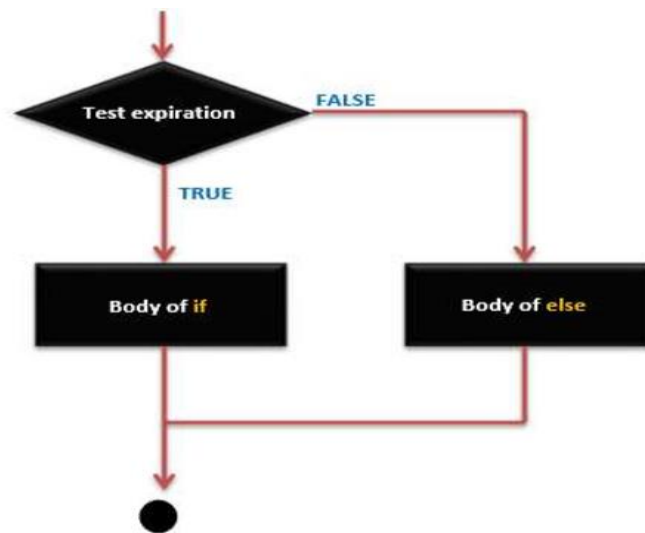
```
<!DOCTYPE html>
<html>
  <body>
    <center><h2>if-satsen</h2></center>
    <script src="simple.js">
      var A=5;
      var B = 9;
      if(A>B){ // Om villkoret är sant utföra följande påstående
        document.write(" A är större än B <br>"); // A>B
      }
      else if(A==B){
        document.write(" A är lika med B <br>"); //eller annars A==B
      }
      else{
        document.write(" A är mindre än B <br>"); //annars A<B
      }
    </script>
  </body>
</html>
```

If-else-satsen

Tvåvägsväl är ett val mellan två alternativ. Precis som i if-satsen görs valet pga ett enda villkor.

Är villkoret sant, utförs en eller flera instruktioner. Låt oss kalla dessa alternativ 1. Är villkoret falskt, utförs en annan uppsättning

```
if (expression)
  1.
  {
    Block of statements;
  }
  2.
```

if...else if ...else satsen

If - satsen kan följas av ett valfritt else – if - else satsen, vilket är mycket användbart för att testa olika villkor med singel if - else if satsen.

När du använder if – else - if - else satsen, kom ihåg

- If satsen kan ha noll eller ett else satsen och det måste komma efter if satsen.
- If satsen kan ha noll eller flera else satsen och de måste komma efter if satsen.

- När en else-if satsen sant, kommer ingen av de återstående att bli test.

If-else-if-else syntax

```
if (expression_1)
{
    Block of statements;
}
else if(expression_2)
{
```

```

<!DOCTYPE html>
<html>
  <body>
    <center><h2>if-satsen</h2></center>
    <script src="simple.js" type="text/javascript">
      var name = prompt("Hej, Vad heter du?");
      var age = prompt("Hur gammal är du?");
      text(50,150,20,"Hej " + name + "!", "black");
      if(age < 20)
      { // Om villkoret är sant utföra följande påstående
        text(50,250,20,"Glöm det", "black");
      }
      else if(age <25)
      {
        text(50,250,20,"Du behöver visa legitimation", "black");
      }
      Else
      {
        text(50,250,20,"Oj vad gammal du är!", "black");
      }
    </script>
  </body>
</html>

```

Lektion4b.html

```

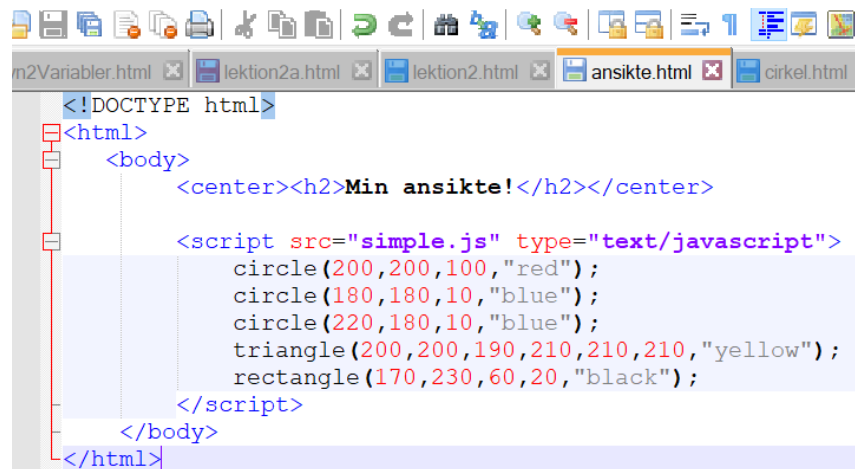
<!DOCTYPE html>
<html>
  <body>
    <center><h2>if-satsen</h2></center>
    <script src="simple.js" type="text/javascript">
      var name = prompt("Hej, Vad heter du?");
      var age = prompt("Hur gammal är du?");
      text(50,150,20,"Hej " + name + "!", "black");
      if(age < 20){ // Om villkoret är sant utföra följande påståen
        text(50,250,20,"Glöm det", "black");
      }
      else if(age <25){
        text(50,250,20,"Du behöver visa legitimation", "black");
      }
      else{
        text(50,250,20,"Oj vad gammal du är!", "black");
      }
    </script>
  </body>
</html>

```

Min ansikte !! prova och tänk på x-koordinaten och y-kordinaten

```
<!DOCTYPE html>
<html>
  <body>
    <center><h2>Min ansikte!</h2></center>
    <script src="simple.js" type="text/javascript">
      circle(200,200,100,"red");
      circle(180,180,10,"blue");
      circle(220,180,10,"blue");
      triangle(200,200,190,210,210,210,"yellow");
      rectangle(170,230,60,20,"black");
    </script>
  </body>
</html>
```

Lektion4c.html



```
<!DOCTYPE html>
<html>
  <body>
    <center><h2>Min ansikte!</h2></center>
    <script src="simple.js" type="text/javascript">
      circle(200,200,100,"red");
      circle(180,180,10,"blue");
      circle(220,180,10,"blue");
      triangle(200,200,190,210,210,210,"yellow");
      rectangle(170,230,60,20,"black");
    </script>
  </body>
</html>
```



Uppgift 4: Rita din snögubbe. Kallar filen [lektion4d.html](#). Öppna dokumentet i webbläsaren och se resultatet.

Uppgift 5: skapa ett program som fråga användaren vilken operation vill utföras med din miniräknare. Addition, subtraktion, multiplikation eller division (Tillåt inte division med noll **tips if (b!=0)** **b** är inte lika med noll. Kallar filen [lektion4e.html](#)

Tips!

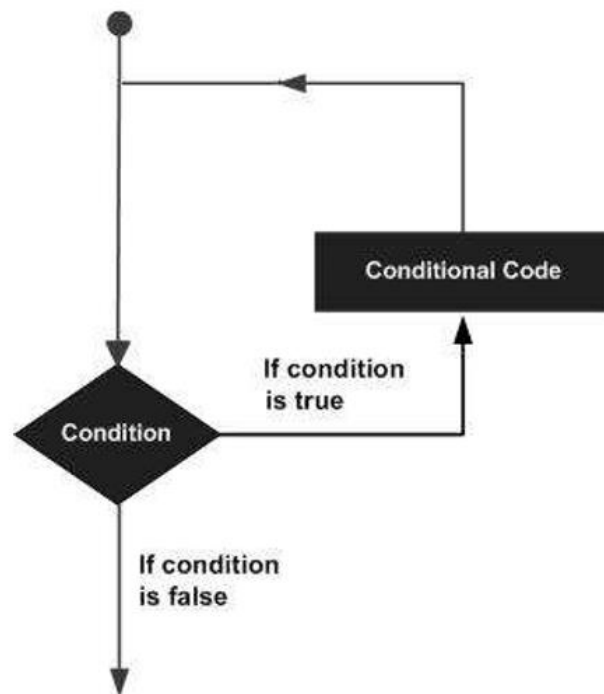
```
var operation =prompt("Var vänlig och skriv in addition, subtraktion, multiplikation eller division", "addition");
```

```
if ((operation == "addition") )  
{  
    //Din kok  
}
```

LOOPAR!

Programmeringsspråk ger olika kontrollstrukturer som möjliggör mer komplicerade körbanor.

Ett loop-uttalande tillåter oss att utföra ett uttalande eller en grupp uttalanden flera gånger och följande är den allmänna formen av ett loop-uttalande på de flesta programmeringsspråket.



C programmeringsspråk ger följande typer av loops för att hantera slingningskrav.

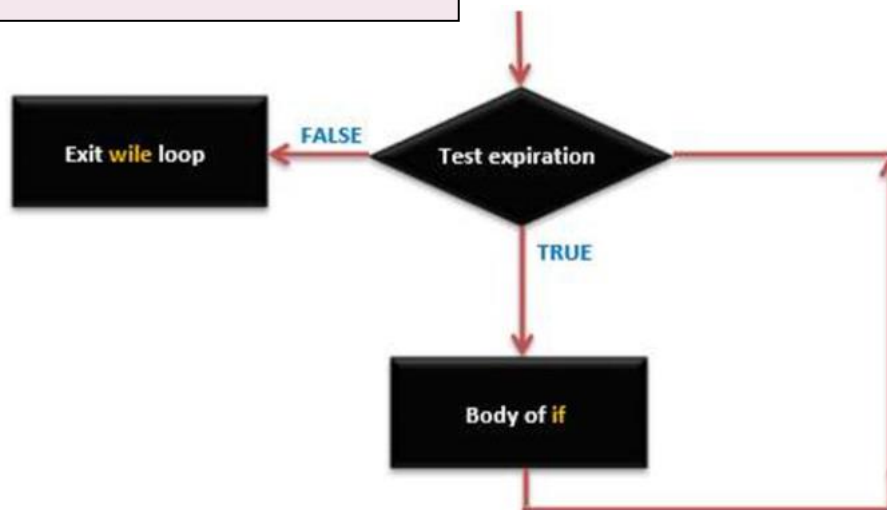
- while loop
- do - while loop
- for loop
- nested loop
- infinite loop

While loop

While loop ska loop kontinuerligt och oändligt tills uttrycket inuti parentesen, () blir falskt. Något måste ändra den testade variabeln, eller medan slingan aldrig kommer att gå ut.

```
while loop Syntax

while(expression)
{
    Block of statements;
}
```



```
<!DOCTYPE html>
<html>
  <body>
    <center><h2>While-
loop!</h2></center>
    <script src="simple.js"
type="text/javascript">
      var x = 100;
      while ( x <= 700) // true or false
      {
        circle(x,200,40,"orange");
        x = x+100; // x += 100;
      }
    </script>
  </body>
</html>
```

file:///C:/Carlos/programming/JavaScript/JavaScript/while_loop.html

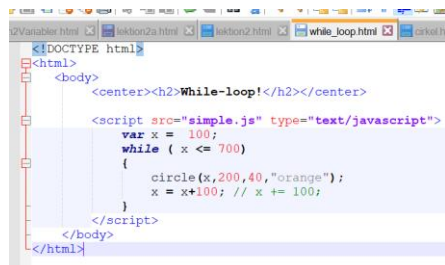
While-loop!



Ordet **while** kan översättas till så länge som. Det innebär att koden mellan mäsvingarna {} körs om och om igen så länge villkoret är uppfyllt. Vi sätter variabeln **x** till 100 och kör loopen så länge som **x** är mindre än

eller lika med 700. Varje gång ritas en cirkel ut och variabeln ökas med 100. Se resultatet. Vi får sju cirklar, men vi behöver bara skriva `circle` en gång.

Lektion5a.html



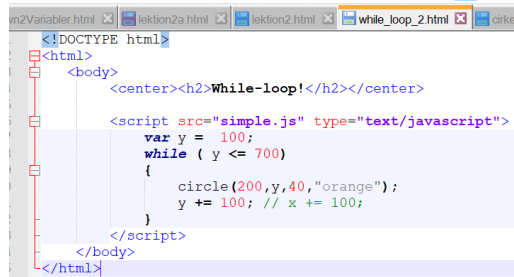
```
<!DOCTYPE html>
<html>
  <body>
    <center><h2>While-loop!</h2></center>
    <script src="simple.js" type="text/javascript">
      var x = 100;
      while ( x <= 700)
      {
        circle(x,200,40,"orange");
        x = x+100; // x += 100;
      }
    </script>
  </body>
</html>
```

```
<!DOCTYPE html>
<html>
  <body>
    <center><h2>While-
loop!</h2></center>
    <script src="simple.js"
type="text/javascript">
      var y = 100;
      while ( y <= 700)
      {
        circle(200,y,40,"orange");
        y += 100; // x += 100;
      }
    </script>
  </body>
</html>
```



Här sätter vi variabeln `y` till 100 och kör loopen så länge som `y` är mindre än eller lika med 700. Varje gång ritas en cirkel ut och variabeln ökas med 100.

Lektion5b.html



```
<!DOCTYPE html>
<html>
  <body>
    <center><h2>While-loop!</h2></center>
    <script src="simple.js" type="text/javascript">
      var y = 100;
      while ( y <= 700)
      {
        circle(200,y,40,"orange");
        y += 100; // x += 100;
      }
    </script>
  </body>
</html>
```

For loop

A for loop utförs uttalanden ett förutbestämt antal gånger. Kontrolluttrycket för slingan initieras, testas och manipuleras helt inom slingans parentes.

Varje slinga har upp till tre uttryck, som bestämmer dess funktion. Följande exempel visar generellt för loop-syntax. Observera att de

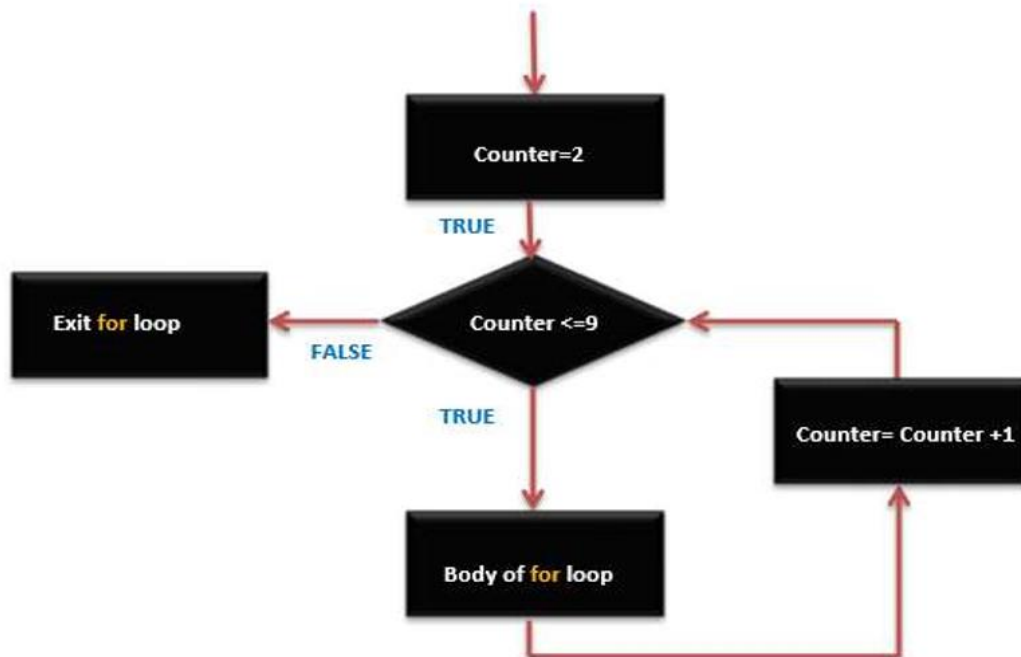
tre uttrycken i parentesen för loop-argument är separerade med semikolon.

for loop Syntax

```
for ( initialize; control; increment or decrement)
{
    // statement block
}
```

Example

```
var counter;
for(counter=2;counter <=9;counter++)
{
    //statements block will executed 10 times
}
```



```

<!DOCTYPE html>
<html>
  <body>
    <center><h2>for-loop!</h2></center>
    <script src="simple.js"
type="text/javascript">
      var x;
      for(x=100; x <= 700; x += 100) { // x = x
+ 100
        circle(x,200,40,"orange");
      }
    </script>
  </body>
</html>

```

När man kör koden kommer **circle** att anropas sju gånger, och resultatet ser likadant ut som **while** ovan.

- **x** initieras med 100
- **x<=700** kollas för varje varv (**true or false**). Detta kan översättas till: Är innehållet i variabeln **x** mindre än eller lika med 700? Inför första varvet kommer testet att svara ja, 100 är mindre än 700. Det betyder att loopens kodblock kommer att köras en gång.
- **x +=100** kod som utförs varje varv. **X** ökas med 100

[Lektion5c.html](#)

file:///C:/Carlou/programming/JavaScript/JavaScript/while_loop.html

While-loop!



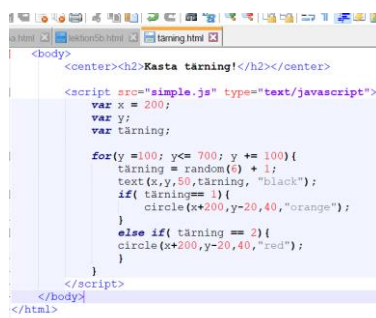
Kasta en tärning:

Med hjälp av funktionen `random(X)` kan man slumpa ett slumptal mellan 0 och $X-1$.

Om vi slumopar ett tal mellan 1 och 6 ska vi bara summera en etta då blir slumptalet mellan 1 och 6. Se exempel.

```
<!DOCTYPE html>
<html>
  <body>
    <center><h2>Kasta
tärning!</h2></center>
    <script src="simple.js"
type="text/javascript">
      var x = 200;
      var y;
      var tärning;
      for(y=100; y<= 700; y += 100){
        tärning = random(6) + 1;
        text(x,y,50,tärning, "black");
        if( tärning== 1){
          circle(x+200,y-20,40,"orange");
        }
        else if( tärning == 2){
          circle(x+200,y-20,40,"red");
        }
      }
    </script>
  </body>
</html>
```

Programmet kastar en tärning, om tärningen visar en etta ritas en cirkel med orange färg. Om tärningen visar en två då ritas en annan cirkel med röd färg.



```
<body>
  <center><h2>Kasta tärning!</h2></center>
  <script src="simple.js" type="text/javascript">
    var x = 200;
    var y;
    var tärning;
    for(y=100; y<= 700; y += 100){
      tärning = random(6) + 1;
      text(x,y,50,tärning, "black");
      if( tärning== 1){
        circle(x+200,y-20,40,"orange");
      }
      else if( tärning == 2){
        circle(x+200,y-20,40,"red");
      }
    }
  </script>
</body>
</html>
```

lektion5d.html

Uppgift 4: Rita 7 cirklar diagonal med både `while`-sats och `for`-sats.

Kallar filen [lektion5e.html](#). Öppna dokumentet i webbläsaren och se resultatet.

Uppgift 5: Skapa ett program som kasta en tärning.

- Om du får en etta kan du rita en rektangel.
- Om du får en två kan du rita en cirkel.
- Om du får en trea kan du rita en rektangel.
- Om du får en femma kan du rita en ring.
- Om du får en sex kan du rita en linje.

Kallar filen [lektion5f.html](#). Öppna dokumentet i webbläsaren och se resultatet.

Uppmaning:

Uppgift 6: Skapa ett program som gissa ett tal mellan 1-10. Om du får rätt får du en ballong. Slumpar ett tal och låt användaren gissa det.

Uppgift 7: Skapa ett program som simulera sten(cirkel), sax (triangel) och påsa(ring). Spelar mot dator som slumpar ett tal 1 motsvara sten, 2 motsvarar en sax och 3 motsvarar påse.

LÄNKAR

JAVASCRIPT

https://www.tutorialspoint.com/javascript/javascript_math_object.htm

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Math

SPELPROGRAMMERING.NU

<http://www.spelprogrammering.nu/>

LITTERATURFÖRTECKNING