



**Översikt**

- Översikt JavaScript.
- Kontrollera flödet.
- Arbeta med loopar.

.eerc

**Lektion 1: Översikt JavaScript**

- Historik.
- Samverkan med DOM, AJAX.
- Syntax.
- Variabler.
- Funktioner.
- Operatorer.
- Arrays.

.eerc

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

## Historik

- Togs fram som ett programmeringspråk i mitten av 1990-talet, lanserat 1997.
- Bygger på ECMA-262, uppdaterat frekvent sedan dess.
- De motorer som finns i webbläsare är betydligt snabbare än sina föregångare.
- Motor tillhandahåller:
  - Variabler.
  - Operatörer.
  - Funktioner.
  - Conditional statements och loopar.
  - Möjlighet att skapa objekt, med miljövariabler, metoder och händelser.

.eerc

---

---

---

---

---

---

---

---

## Samverkan med DOM, AJAX

- Document Object Model.
- Tillsammans med DOM kan JavaScript göra mer:
  - lägga till eller ta bort poster i lista.
  - lägga till, förändra eller ta bort text.
  - förändra CSS style för element.
  - reagera på händelser, exempelvis klick på mus.
  - validera formulär.
  - visa meddelande.
  - inhämta information om webbläsare.
- JavaScript tillsammans med XMLHttpRequest API (AJAX) kan asynkrona anrop göras.

.eerc

---

---

---

---

---

---

---

---

## Syntax

- Enkel syntax för att skriva statement, deklarerar variabler och lägga till kommentarer.
- Statments är rad som representerar operation som skall utföras.
- Skulle kunna vara:
  - Deklaration av variabel.
  - Tilldelning av värde.
  - Anrop till funktion.
- Enkel syntax för att skriva statement, deklarerar variabler och lägga till kommentarer.
- Kommentarer kan skrivas på en eller flera rader.

```
var dennaVariabel = 3;
counter = counter + 1;
GaFillDetta(); // Gå till funktion
document.write("Ett väldigt långt\
Meddelande");
/* Kommentar kan vara på flera
rader. */
```

.eerc

---

---

---

---

---

---

---

---

---

---

## Variabler

- Variabler används för att lagra information.
- Tre sätt att deklarerar variabel:
  - Ge ett namn och ett värde: `fras = "Hallå";`
  - Deklaration utan ett värde: `var bil;`
  - Kombination: `var fordon = "Personbil";`
- Datatyp definieras inte, försöker att räkna ut typ.
- Känner igen tre enkla datatyper:
  - String.
  - Number.
  - Boolean.
- Deklarerad men ej tilldelat värde = `undefined`.

.esrec

---

---

---

---

---

---

---

---

## Variabler (forts.)

- Operator `typeof` kan användas för att räkna ut datatyp.

```
var data = 99;
...
if (typeof data == "number") {
    // data är numeriskt
}
```

.esrec

---

---

---

---

---

---

---

---

## Funktioner

- Är en namngiven sekvens av statements, som utför en bestämd arbetsfunktion.
- Efter deklaration, kan funktionen anropas från din kod.
- Kod i funktionen kommer att köras, när detta är klart kommer att återvända till nästa steg efter funktionsanropet.
- Nyckelordet `function` används för att deklarerar start för funktion.
- Ett namn definieras, är case-sensitive, därefter definieras eventuella argument inom parantens.
- Kodkropp inom `{ }`.
- Argument används som variabler.

.esrec

---

---

---

---

---

---

---

---

---

---

## Funktioner (forts.)

- Om variabel deklarerar inom funktion, kan den bara användas av funktionen och kommer att plockas bort från minnet när funktion avslutas.
- Globala variabler deklarerar utanför funktionen, är även närbar inuti funktion.
- Anonyma funktioner kan definieras, dvs utan ett namn.

```
function CalculateBill(numberOfNightsStay, nightlyRate, extras) {  
    return (numberOfNightsStay * nightlyRate + extras);  
}  
...  
var TotalAmountOwned = CalculateBill(10, 100, 50);
```

.eeec

---

---

---

---

---

---

---

---

## Operatörer

- Operatörer används för att utföra operationer på variabler och värden.
- Några huvudtyper i JavaScript:

Typ av operatör	Utför vad
Arithmetic	Beräkningar
Assignment	Tilldelar
Comparison	Jämför
Logiska	Logiska operationer
String	Hanterar strängar
Ternary	Tilldelar värde till variabel baserat på villkor

.eeec

---

---

---

---

---

---

---

---

## Arithmetic Operator

- Utförs på numeriska värden för matematiska funktioner.

Operator	Namn	Exempel	Förklaring
+	Addition	x = y	Summan av x och y
-	Subtraktion	x - y	Skillnad mellan x och y
*	Multiplikation	x * y	Produkten av x och y
/	Division	x / y	Kvoten av x och y
%	Modulus	x % y	Det som blir kvar efter division x med y
++	Öka	x ++	Ökar x med ett.
--	Minskar	x --	Minskar x med ett.

.eeec

---

---

---

---

---

---

---

---

---

---

## Assignment Operator

- Används för att tilldela numeriskt värde till variabel.

Tilldelning	Exempel	Beskrivning
$x = y$	$x = y$	Värdet till vänster sätts för det högra värdet
$x += y$	$x = (x + y)$	Addition
$x -= y$	$x = (x - y)$	Subtraktion
$x *= y$	$x = (x * y)$	Multiplikation
$x /= y$	$x = (x / y)$	Division
$x \% = y$	$x = (x \% y)$	Modulus

.eeec

---

---

---

---

---

---

---

---

## Comparison Operator

- Används för att jämföra två numeriska värden.

Operator	Namn	Exempel	Resultat
<code>==</code>	Lika med	$x == y$	Returnerar true om x är lika med y
<code>===</code>	Identisk	$x === y$	Returnerar true om x är lika med y och är av samma typ
<code>!=</code>	Inte lika	$x != y$	Returnerar true om x inte är lika med y
<code>&lt;&gt;</code>	Inte lika	$x <> y$	Returnerar true om x inte är lika med y
<code>!==</code>	Inte identisk	$x !== y$	Returnerar true om x inte är lika med y och inte är av samma typ

.eeec

---

---

---

---

---

---

---

---

## Comparison Operator (forts.)

Operator	Namn	Exempel	Resultat
<code>&gt;</code>	Större än	$x > y$	Returnerar true om x är större än y
<code>&gt;=</code>	Större än eller lika med	$x >= y$	Returnerar true om x är större än eller lika med y
<code>&lt;=</code>	Mindre än eller lika med	$x <= y$	Returnerar true om x är mindre än eller lika med y

.eeec

---

---

---

---

---

---

---

---

---

---

## Logical Operator

Operator	Beskrivning	Exempel	Resultat
&&	And	(x < 10 && y > 1)	Returnerar true om både uttrycken är lika, annars false
	Or	(x == 5    y == 5)	Returnerar true om både uttrycken inte är lika, annars false
!	Not	!(x == y)	Returnerar true om både uttrycken inte är lika, annars false

.eeec

---

---

---

---

---

---

---

---

## String Operator

Operator	Namn	Exempel	Resultat
+	Sammanlänkar	x + y	Sammanlänkar x och y
+=	Tilldelning	x += y	Lägger till y till x

.eeec

---

---

---

---

---

---

---

---

## Ternary Operator

▪ variabelnamn = (villkor) ? värde1:värde2

▪ Exempelvis:

```
var nu = new Date();
var meddelande = "God";
if (nu.getHours() > 17) {
    meddelande += " kväll.";
    document.write(meddelande);
}
else {
    meddelande += " dag.";
    document.write(meddelande);
}
```

.eeec

---

---

---

---

---

---

---

---

---

---

## Ternary Operator -exempel

```
<!DOCTYPE html>
<html lang="sv">
  <head>
    <meta charset="utf-8" />
    <title>modul3demoternary</title>
  </head>
  <body>
    <script>
      var nu = new Date();
      var meddelande = "God";
      if (nu.getHours() > 17) {
        meddelande += " kväll.";
        document.write(meddelande);
      }
      else {
        meddelande += " dag.";
        document.write(meddelande);
      }
    </script>
  </body>
</html>
```

modul3demoternary.htm

.eerec

---

---

---

---

---

---

---

---

## Arrays

- Area där information kan lagras.
- Exempel:  
var frukter = ['Apelsiner', 'Bananer'];
- Skillnad mellan array och objekt:
  - I JavaScript använder array numrerat index.
  - I JavaScript använder objekt namngivna index.
- Array är en speciell typ av objekt, operator `typeof` kan användas för att inhämta värde i array.

.eerec

---

---

---

---

---

---

---

---

## Arrays -exempel

```
<!DOCTYPE html>
<html lang="sv">
  <head>
    <meta charset="utf-8" />
    <title>modul3demoarray</title>
  </head>
  <body>
    <p id="frukt"></p>
    <script>
      var frukter = ["Apelsiner", "Bananer"];
      document.getElementById("frukt").innerHTML = "Frukt i array: " +
      frukter;
    </script>
  </body>
</html>
```

modul3demoarray.htm

.eerec

---

---

---

---

---

---

---

---

---

---

## Lektion 2: Kontrollera flödet

- Kontrollera flödet.
- `if` statement.
- `if Else` statement.
- `else if` statement.
- `switch` statement.

.eerec

---

---

---

---

---

---

---

---

## Kontrollera flöde

- I varje programmeringsspråk är det viktigt att kunna kontrollera flödet.
- `if` statement.
- `switch` statement.

.eerec

---

---

---

---

---

---

---

---

## `if` statement

- Vanligaste sättet att kontrollera flödet.
- Ger möjlighet att arbeta med olika vägar för logiken, beroende på givet villkor.
- När villkoret undersökts till sant (`true`), kommer ett block av kod att exekveras.
- Kan användas ensamt eller tillsammans med `else` och `else if`.

.eerec

---

---

---

---

---

---

---

---

---

---



### if statement exempel

```
if (hour < 18) {  
    greeting = "Goddag";  
}
```

.eerec

---

---

---

---

---

---

---

---

### if else statement exempel (forts.)

```
if (hour < 18) {  
    greeting = "Goddag";  
} else {  
    greeting = "Godkväll";  
}
```

.eerec

---

---

---

---

---

---

---

---

### else if statement exempel (forts.)

```
if (time < 10) {  
    greeting = "Godmorgon";  
} else if (time < 20) {  
    greeting = "Goddag";  
} else {  
    greeting = "Godkväll";  
}
```

.eerec

---

---

---

---

---

---

---

---

---

---

## switch statement

- Ett annat sätt att kontrollera flödet, är med `switch statement`.
- `switch` exekverar ett set av logik, beroende på värde för given parameter.
- Nyckelordet `break` används för att förhindra att kod kör nästa case automatiskt.
- Nyckelordet `default` används när ingen match hittas.
- Flera case kan användas.

.eeec

---

---

---

---

---

---

---

---

## switch statement exempel

```
<html lang="sv">
  <head>
    <meta charset="utf-8" />
    <title>modul3demoswitch</title>
  </head>
  <body>
    <script>
      mittTest = "Hejsan3";
      switch (mittTest) {
        case 'Hejsan1':
          document.write("Hejsan1");
          break;
        case 'Hejsan2':
          document.write("Hejsan2");
          break;
        case 'Hejsan3':
          document.write("Hejsan3");
          break;
        default: document.write('Ett annat värde');
      }
    </script>
  </body>
</html>
```

modul3demoswitch.html

.eeec

---

---

---

---

---

---

---

---

## Övning Arbeta med if och switch



.eeec

---

---

---

---

---

---

---

---

---

---

### Lektion 3: Arbeta med loopar

- Loopar.
- while loop.
- do loop.
- for loop.
- foreach loop.

.eerec

---

---

---

---

---

---

---

---

### Loopar

- I JavaScript kan följande typer av loopar användas:
  - while loop.
  - do loop.
  - for loop.

.eerec

---

---

---

---

---

---

---

---

### while loop

- while loop kommer att undersöka ett villkor, så länge detta villkor uppfylls (true) kommer block av kod att exekveras.

```
<!DOCTYPE html>
<html lang="sv">
  <head>
    <meta charset="utf-8" />
    <title>modul3demowhile</title>
  </head>
  <body>
    <script>
      myInt = 0;
      while(myInt <= 10) {
        document.write("Inuti loop:" + myInt + "<br>");
        myInt++;
      }
      document.write("Utanför loop:" + myInt);
    </script>
  </body>
</html>
```

modul3demowhile.html

.eerec

---

---

---

---

---

---

---

---

---

---

## do loop

- do loop liknar while loop med undantaget att kontroll av villkoret sker i slutet av loop.
- Garanterar att do loop exekveras minst en gång.

```
<!DOCTYPE html>
<html lang="sv">
  <head>
    <meta charset="utf-8" />
    <title>modul3demodowhile</title>
  </head>
  <body>
    <script>
      myInt = 0;
      do {
        document.write("Inuti loop:" + myInt + "<br>");
        myInt++;
      }
      while (myInt <= 10);
      document.write("Utanför loop:" + myInt);
    </script>
  </body>
</html>
modul3demodowhile.html
```

.eerec

---

---

---

---

---

---

---

---

## for loop

- for loop fungerar som while loop med undantaget att syntax inkluderar initalisering och modifikation.

Exempel: `for (x = 0; x < 10; x++)`

- for loop används när du vet exakt hur många gånger du vill utföra ditt uttryck inom loop.

.eerec

---

---

---

---

---

---

---

---

## for loop exempel

```
<!DOCTYPE html>
<html lang="sv">
  <head>
    <meta charset="utf-8" />
    <title>modul3demofor</title>
  </head>
  <body>
    <script>
      for (myInt = 0; myInt <= 10; myInt++)
      {
        document.write("Inuti loop:" + myInt + "<br>");
      }
      document.write("Utanför loop:" + myInt);
    </script>
  </body>
</html>
modul3demofor.html
```

.eerec

---

---

---

---

---

---

---

---

---

---

## foreach loop

- `foreach` loop används för att läsa igenom lista över poster, finns i de flesta programmeringsspråk.
- Fungerar på array och collections.
- Finns inte stöd för detta i många webbläsare, därför bör man inte använda denna sats.

.eeec

---

---

---

---

---

---

---

---

## Övning Arbeta med loopar



.eeec

---

---

---

---

---

---

---

---

## Repetitionsfrågor

.eeec

---

---

---

---

---

---

---

---

---

---