

Ett program är en uppsättning instruktioner som programmeraren talar om för datorn hur den ska utföras, eftersom datorn inte kan tänka upp såna applikationer själv .

1 Javascript är ett av de vanligaste programmeringsspråken och är inbyggt i nästan alla webbläsare.

2 Var redo att bli frustrerad Javascript är ett helt nytt språk med några likheter till C# var beredd på att ta pauser och läsa om sidor.

3 Ett program är instruktioner på datorns minnen som skrivs av människor, dessa program kan lätt bli komplicerade och lärdom behövs.

Bits

1 Bits = Bits kan vara vilka som helst tvåvärda saker dem beskrivs oftast som 0 och 1 inuti datorn har dem formen av starka eller låga elektriska laddningar, en stark eller svag signal eller en skarp eller dum plats.

1 Bits = All information på datorn är uppbyggd av Bits som representerar sekvenser av 0 - 1.

1 En dator använder ett binärt system vilket är uppbyggd av Bits alltså 0 - 1

1 Bytes = en byte består av 8 Bit medans en Bit endast består av 0 -1 En byte kan representera 256 värden.

Values

2 En dators arbetande minne är uppbyggd av ett hav av Bits cirka 30 miljarder

för att kunna jobba med så många Bits behöver vi separera dem in till delar som representerar delar av information inom Javascript kallas dessa delar Values.

2 En Value = En samling Bitar som representerar information och genom att kombinera bitarna på olika sätt kan datorn representera och manipulera en enorm mängd olika data som text, siffror, ljud eller video

2 Skapa ett numeriskt Value i Javascript görs enkelt genom att skriva in numret direkt i programmet utan att behöva lägga till några extra termer

let number = 13; Nu har vi skapat en numerisk Value

Arithmetic

3 Arithmetic är operators = plus, minus, gånger eller delat

3 Dessa har en order till dem gånger * och delat / har högre prioritering än plus + och minus -.

Speciella nummer

4 = NaN, Infinity, -Infinity anges då operationer som inte leder någonstans sker som 0 / 0 eller 0 - 0.

Strings

5 = String är datatyper som representerar text och skrivs inom citattecken =
let hej = "hej"; Nu har jag skrivit en string i Javascript.

5 Backstrings ` Används ofta för att skriva en string som innehåller speciella tecknen
let hej = `hej \${100/2}` blir = hej 50

5 skapa en ny line = let hej = "Första delen /n Andra delen"

5 Du kan inte använda operationer på string som minus eller delat men du kan använda + för att sy ihop flera strings = "con" + "cat" + "e" + "nate"

Unary Operators

6 Unära Operatorer är operatorer som endast använder en enda operand detta skiljer sig mot binära operatorer som använder två operander exempel på Unär

```
let fem = 5;  
let fem = -minuFems;  
console.log(fem); = -5
```

Så skrivs Unära operatorer ett annat exempel skulle vara

```
let i = 5;  
i++;  
console.log(i); = 6
```

Boolean Values

7 Är Values med villkor som `console.log(3 > 2)` True

operatorerna som används är
== lika med
!= inte lika med
<= mindre eller lika med
>= Större eller lika med

Logical operator

8 dessa operatorer används tillsammans med boolean values dessa är And, or, not

|| = or, den kommer att ge True om något av dem två värdena är true
&& = And, båda värdena måste vara true för att det ska bli true

Empty values

9 Null och Undefined används för att markera saknaden av värde.

Automatic type conversion

10 Automatisk typkonvertering sker när operationer med olika datatyper omvandlas till samma datatyp. Javascript försöker gissa vilken datatyp som passar ihop med operationen.

Exempel

```
let result 5 + "10";  
console.log(result); // 510
```

Javascript omvandlade 5 till "5" och sedan satte ihop strings med varandra för att bilda 510

Short-circuiting of logical operators

11 Short-circuiting of logical operators = kontrollerar om den första delen i en logisk operation är false eller true och ger ett svar baserat på det.

om true || false kommer operationen att bli true eftersom den kollar endast på den första delen av operationen.

Detta hjälper med flytet på programmet och optimerar prestanda.

```
let result = true || false;  
console.log(result); // Output: true
```

// Eftersom den första delen (true) är sann, blir hela uttrycket sann och JavaScript kortsluter utvärderingen.

```
let username = ""; // eller undefined eller null
```

```
let displayUsername = username || "Guest";  
console.log(displayUsername); // Om username är falsk, visas "Guest" som standard
```

