



# Javascript 101



# Agenda

- Vad gör en dator?
- Vad gör ett programmeringsspråk?
- Vad är javascript?
- Användning av javascript
  - Statements
  - Variabler
  - Jämförelser
  - If else (Gör ett val)



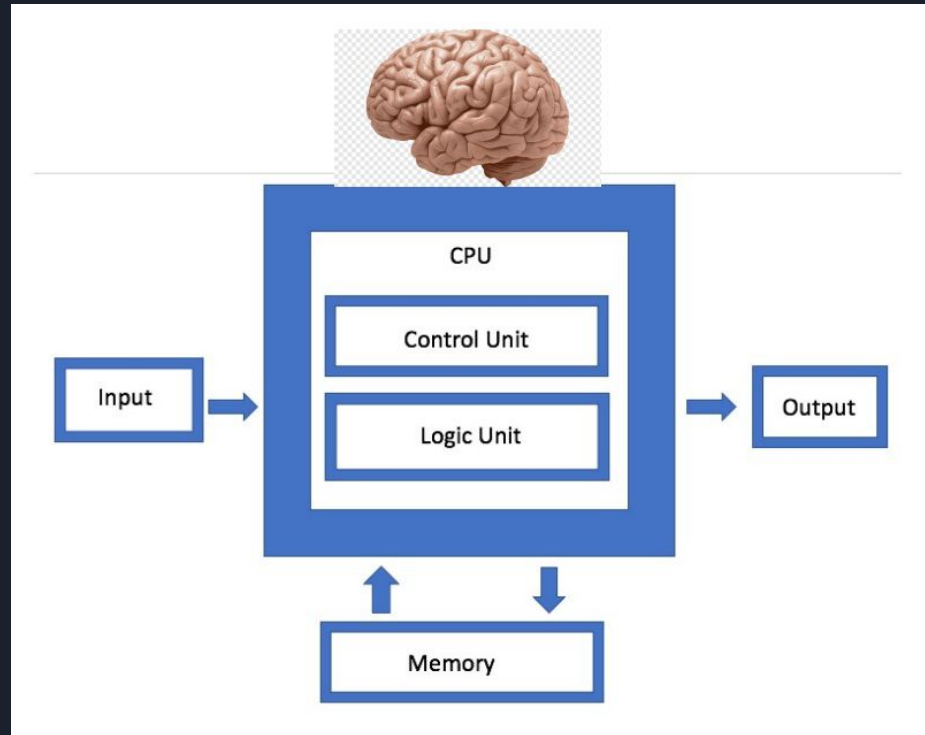
# Vad är javascript?

- För att förstå javascript behöver vi förstå vad ett programmeringsspråk är
- För att förstå ett programmeringsspråk behöver vi förstå vad en dator är...

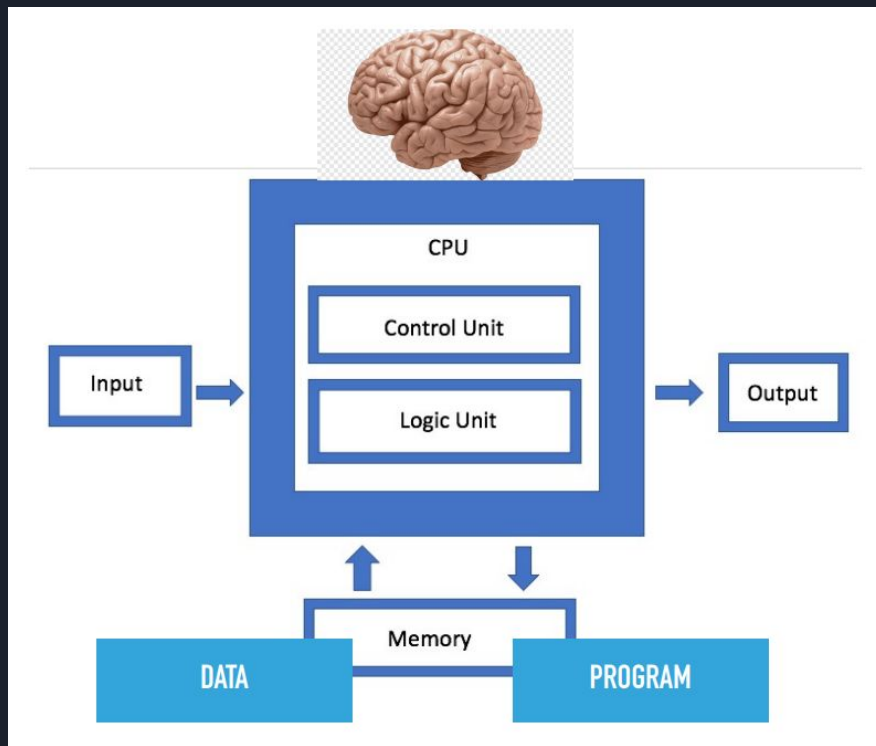
Vad gör en dator?



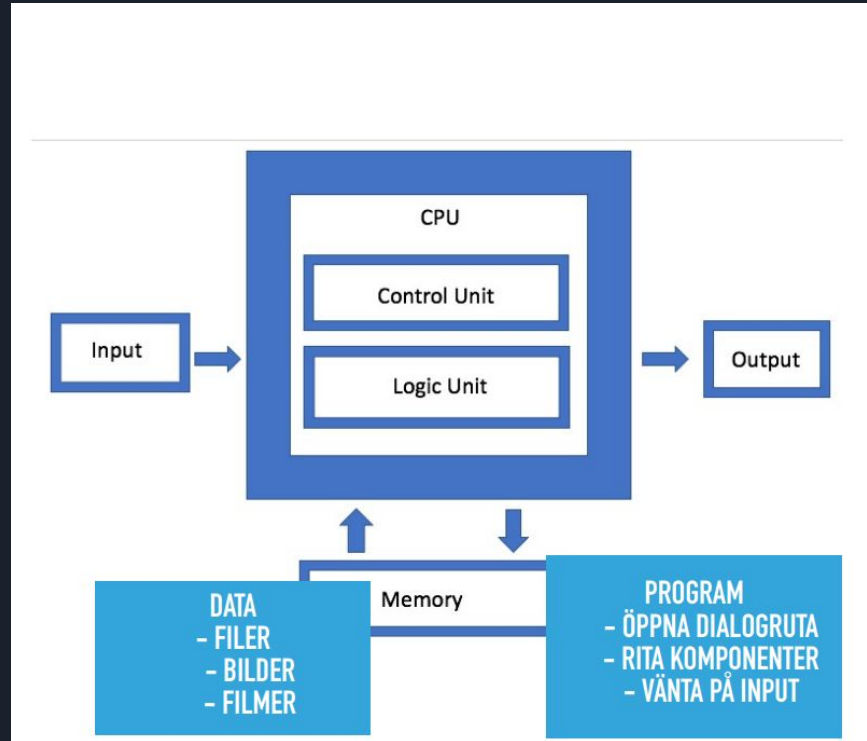
# Vad gör en dator?



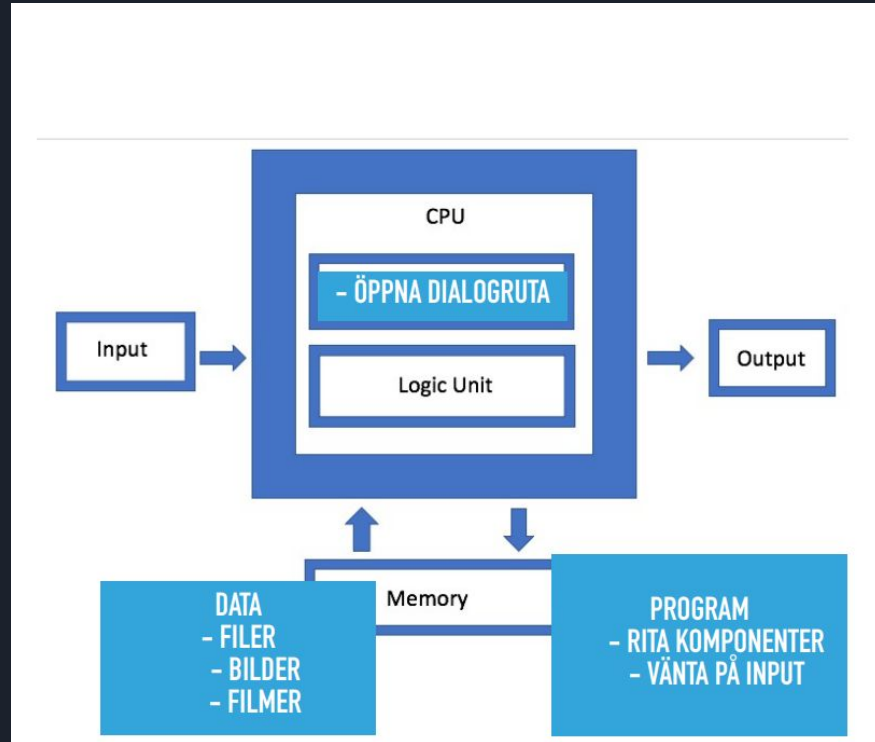
# Vad gör en dator?



# Vad gör en dator?

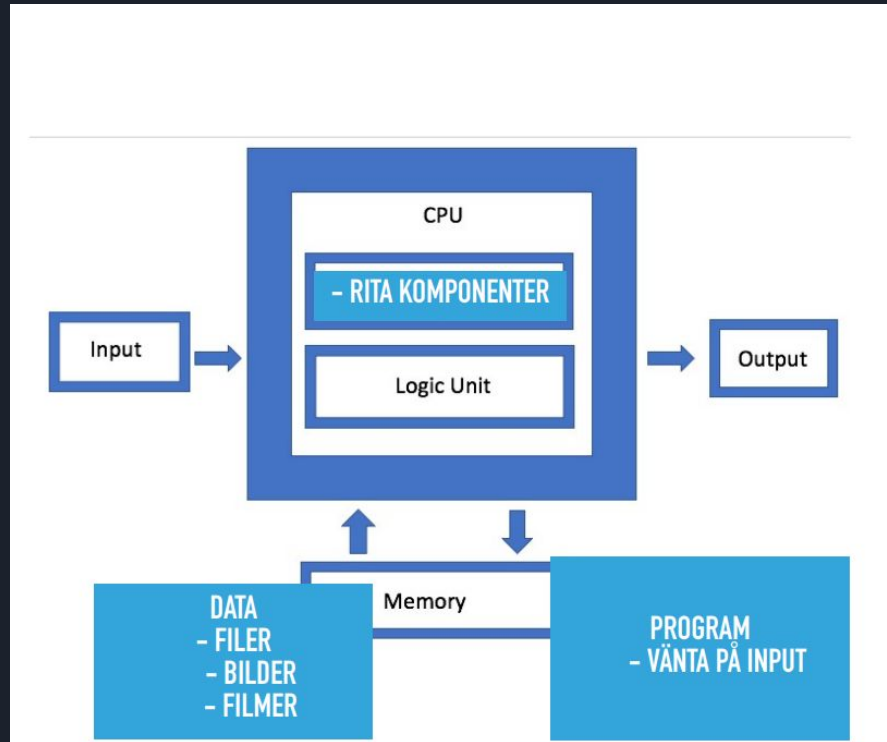


# Vad gör en dator?

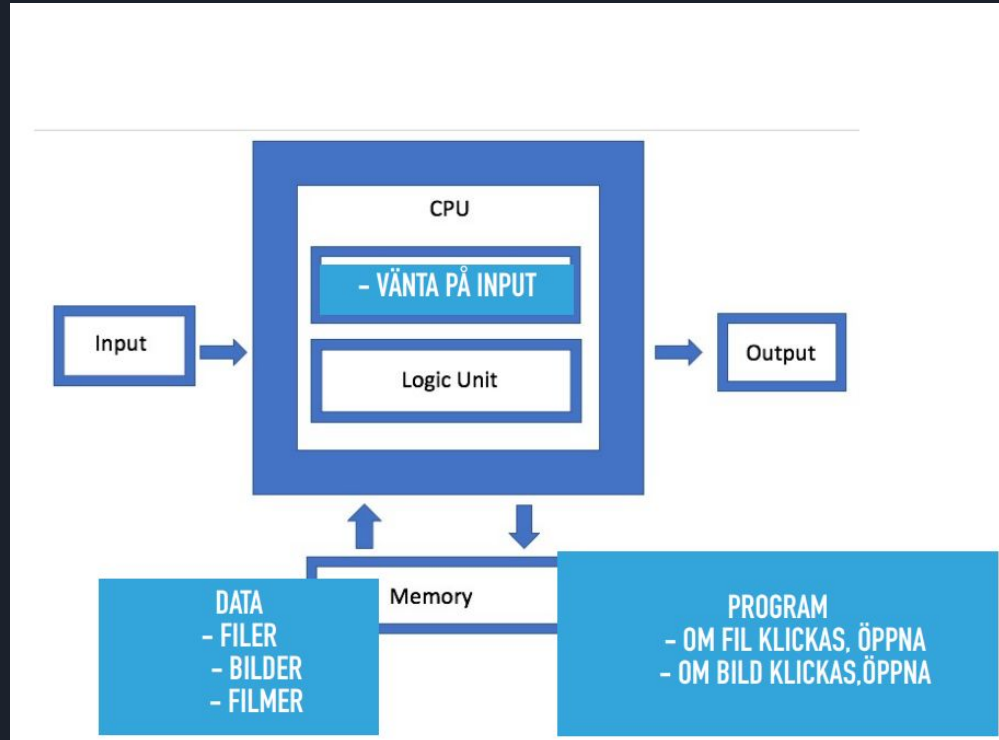




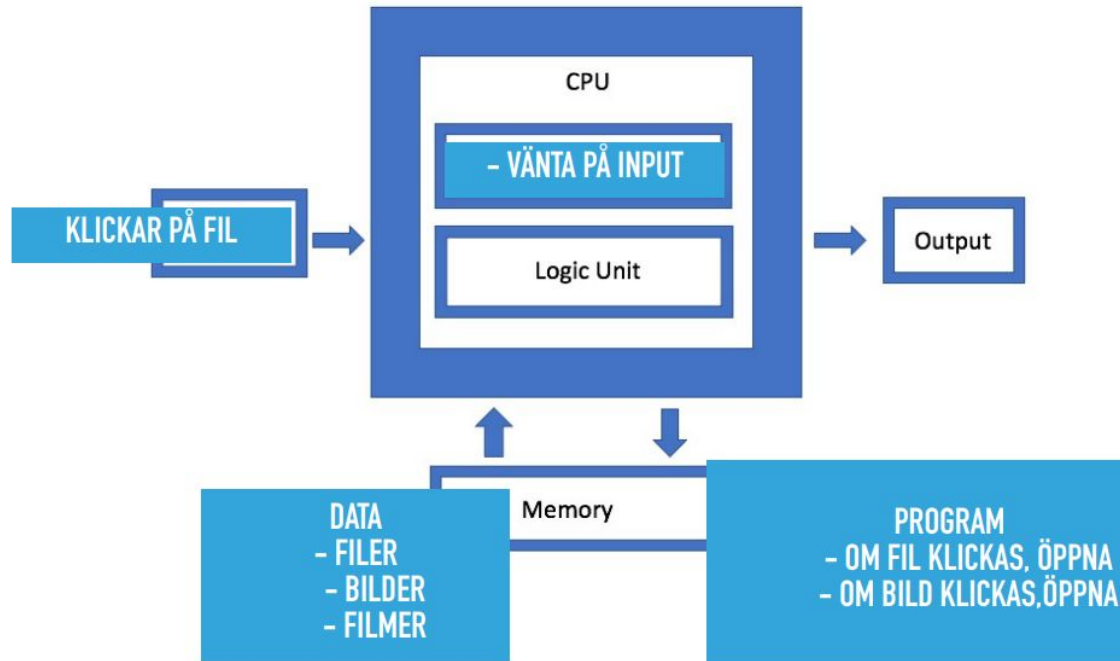
# Vad gör en dator?



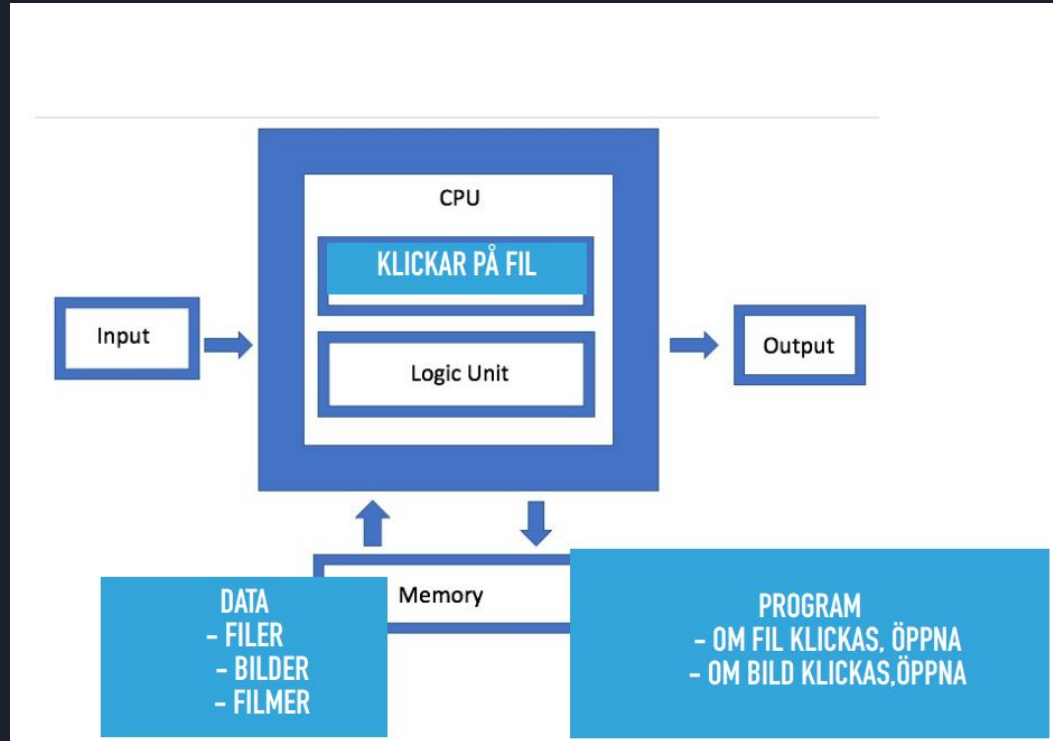
# Vad gör en dator?



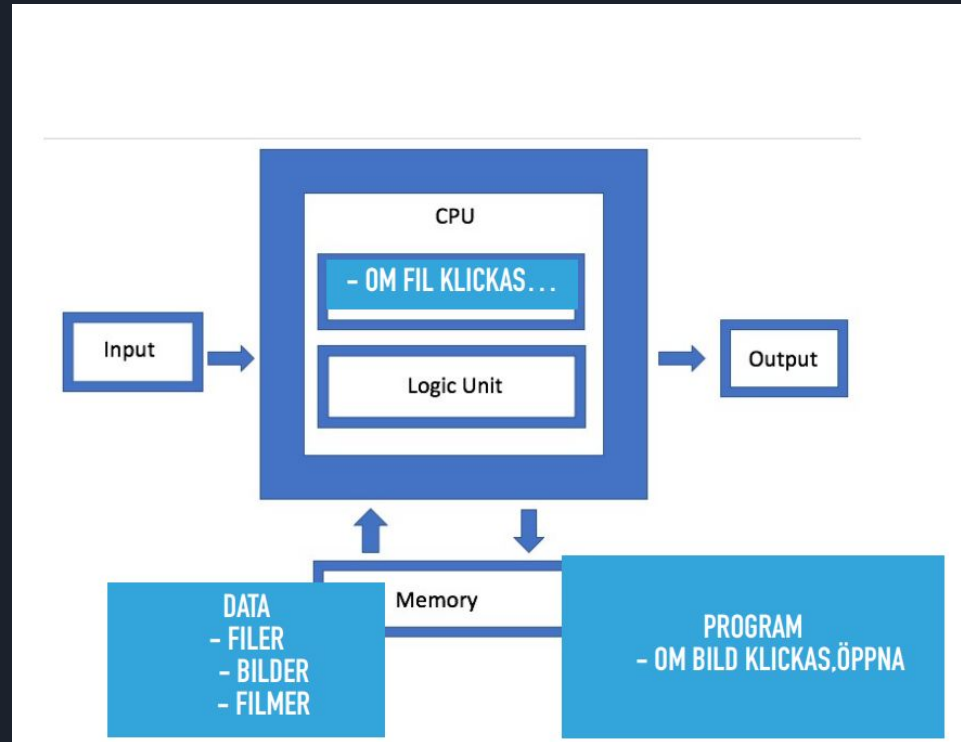
# Vad gör en dator?



# Vad gör en dator?



# Vad gör en dator?



Vad gör ett  
programmeringsspråk?





# Vad gör ett programmeringsspråk?

Demo: Skriv ett program med en bild på åsna och en bild på dokument. När någon trycker på åsnan ska ljud spelas upp och när någon trycker på dokumentet ska dokumentet snurra....

# Vad gör ett programmeringsspråk?

Datorer förstår inte programmeringsspråk, så dessa måste översättas

```
document.querySelector(".again").addEventListener("click", () => {  
  document.querySelector(".message").textContent = "Start guessing...";  
  document.querySelector(".number").textContent = "?";  
  document.querySelector(".guess").value = "";  
  score = 20;  
  document.querySelector(".score").textContent = score;  
  number = Math.floor(Math.random() * 20) + 1;  
});
```

Abstraction over  
0s and 1s

↓  
CONVERT TO MACHINE CODE = COMPILING

```
1101011010111010101110101100101110101010111101010  
0111101010111010100100111010111010101100010101100010  
101001001110111011111001110000001110101011110111010  
11010010000101001011101010110101011101010101010010  
00001110100100100111010101110101011100101011111010  
10010101001001111010011101010001010101001011010100  
11100100010001111010000101011100010100010101110101101  
0101001010100010101000111010010010111010100100010101  
111010100101110101000101011101010010111010101001001  
111001011101110101011101001010101010101010100101010  
011101011011010101001010111010111010101011100111010  
111010100111010100111010110101010101010101010101010
```

Happens inside the  
JavaScript engine

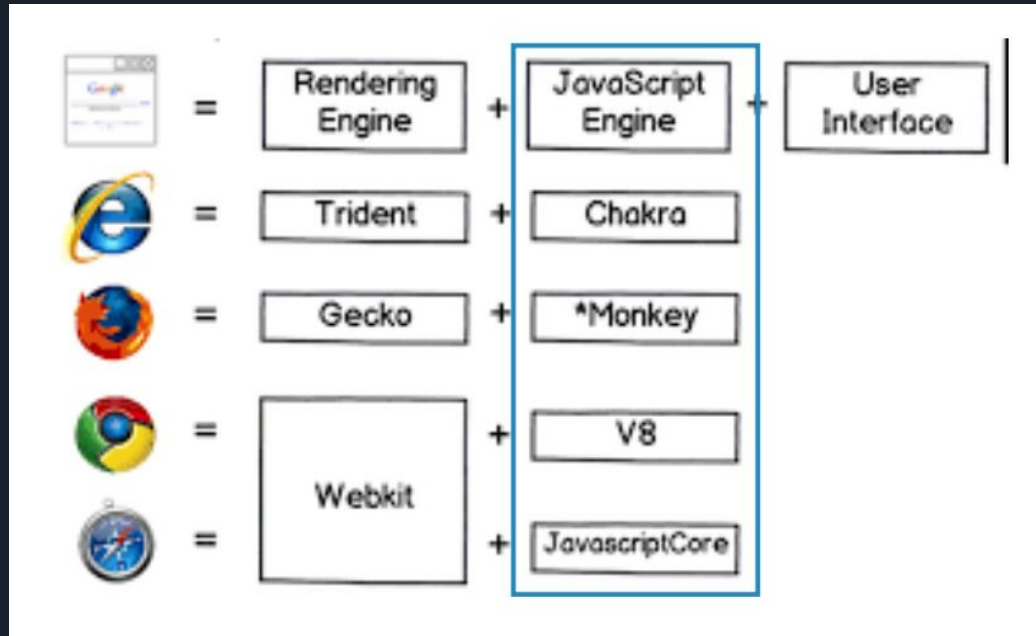


Vad är javascript?



# Vad är javascript?

Varje webbläsare har sin egna renderingengine och interpreter

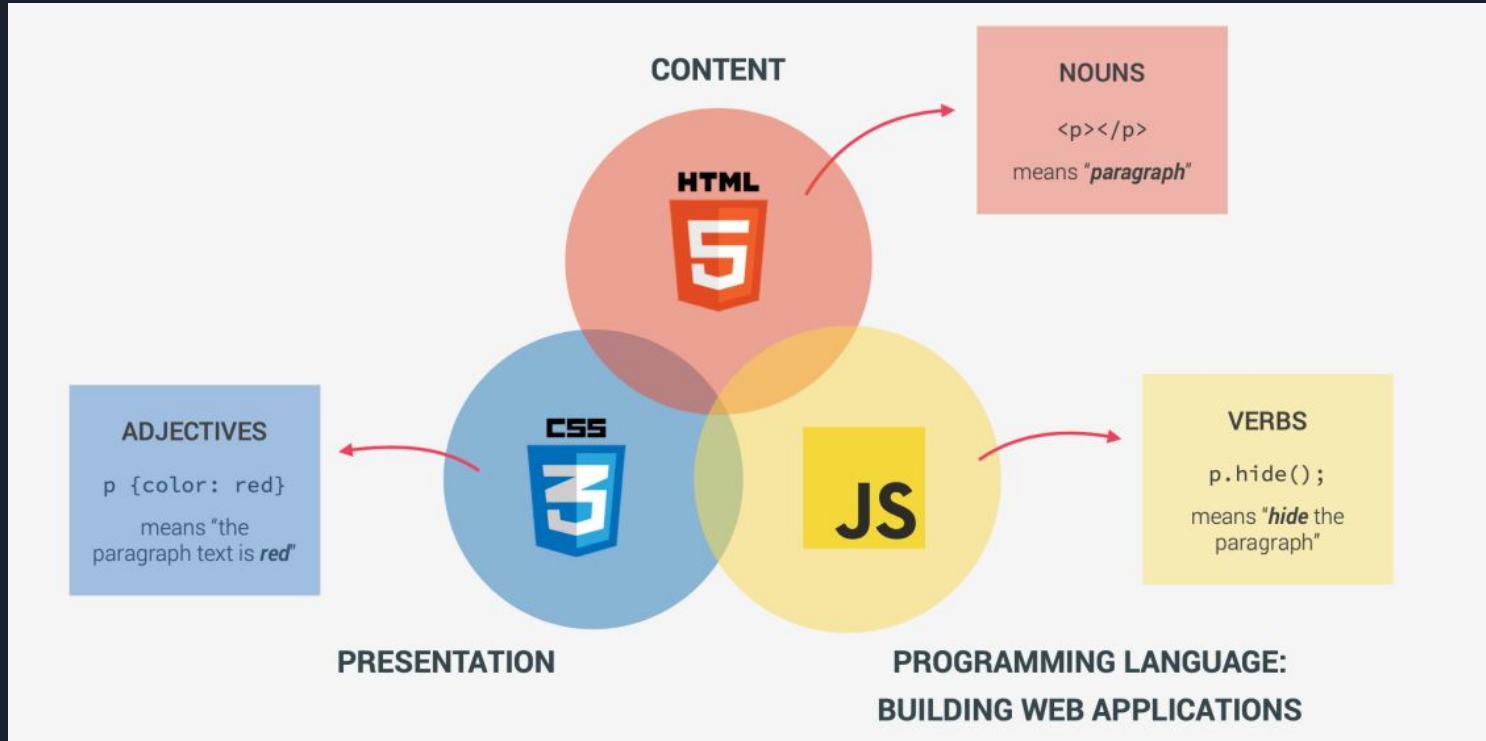




# Vad är javascript?

- Webbläsaren är som en dator inuti datorn
  - Har sitt egna interface (webbläsfönstret)
    - Flera program => Flera flikar
  - Har sitt egna programmeringsspråk (javascript)

# Vad är javascript?





# Vad är javascript?

- Animationer:

<https://developer.mozilla.org/en-US/docs/Web/API/Element/animate>

- Användarinput:

<https://developer.mozilla.org/en-US/docs/Web/API/Element/scrollHeight>

- Spel:

[https://developer.mozilla.org/en-US/docs/Games/Tutorials/2D\\_Breakout\\_game\\_pure\\_JavaScript](https://developer.mozilla.org/en-US/docs/Games/Tutorials/2D_Breakout_game_pure_JavaScript)

- Avancerade spel

<https://www.ma-no.org/en/programming/javascript/15-awesome-games-made-with-html5-and-javascript>



# Vad är javascript?

Med javascript kan vi konkret ändra

- Elements attribut
  - [https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_intro\\_lightbulb](https://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_lightbulb)
- Elements styling
  - [https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_intro\\_style](https://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_style)
- Visa/Gömma element
  - [https://www.w3schools.com/js/tryit.asp?filename=tryjs\\_intro\\_hide](https://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_hide)

# Användning av javascript





# Användning av javascript

Vi kan liksom för CSS lägga till Javascript på olika sätt i ett dokument

- Inline
- I head (intern/extern)
- I body (intern/extern)

Intern: `<script>....</script>`

Extern: `<script src="main.js"></script>`





# Användning av javascript

En användbar nyttig funktion som används flitigt är:

```
console.log();
```

Denna skriver ut saker i consolen...

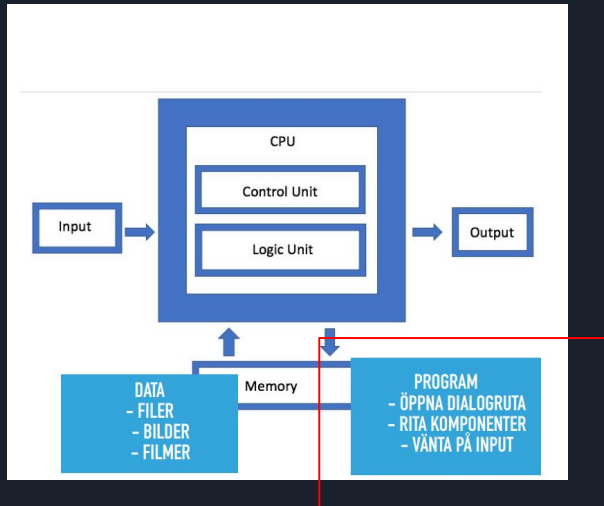
Statements



# Statements

Ett program består av *instruktioner* som vi såg innan

Ett annat ord för instruktioner är “*statements*”





# Statements

```
<!DOCTYPE html>
<html>
<body>

  <p id="demo1"></p>

  <script>

    let a, b, c;
    a = 5;
    b = 6;
    c = a + b;
    document.getElementById("demo1").innerHTML = c;

  </script>

</body>
</html>
```



# Statements

Statements separeras med semikolon ;

Oftast har man en statement per rad men är inte nödvändigt

```
a = 5; b = 6; c = a + b;
```



# Statements

Statements (eller instruktioner) är av tre slag

- **Deklarationer av variabler**
  - `let x;`
- **Tilldelningar av variabler**
  - `x = 5;`
  - `let c = 10;`
  - `document.getElementById("demo1").innerHTML = c;`
- **Actions**
  - `console.log(x);`



# Statements

## Övning

Skapa ett html dokument statements.html. Längst ned i body lägg en `<script></script>` i vilken du ska skriva din kod

```
<!DOCTYPE html>
<html>
<body>

<script>

// Här ska din kod finnas

</script>

</body>
</html>
```



# Statements

## Övning

- Deklarera 1 variabel "x" med värdet 5
- Deklarera 1 variabel "y" med värdet 2
- Deklarera 1 variabel "z" med innehållet av summan av x och y
- Printa ut i konsolen värdet z (console.log)
- Öppna dokumentet i webbläsaren, och öppna konsolen



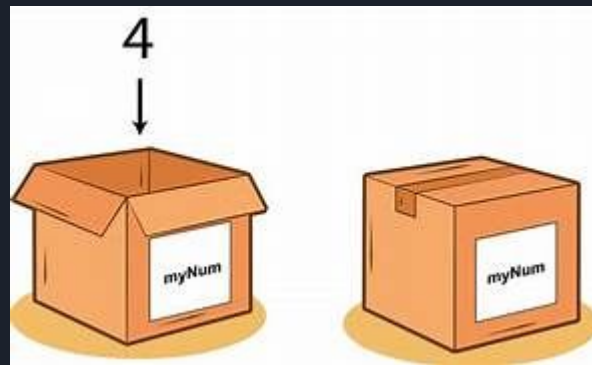
Variabler



# Variabler

Variabler är som lådor med en etikett som vi kan stoppa saker i

```
let myNum = 4;
```





# Variabler

När variablerna är deklarerade så kan vi hänvisa till dem via deras etiketter

(Här ser vi 3 statements där vi deklarerar 3 variabler)

```
let x = 5;  
let y = 6;  
let z = x + y;
```



# Variabler

Demo: Det finns regler kring hur vi namnger variablerna (lådorna)

- Namn bör vara unika
- Vi kan namnge dom som vadsomhelst men helst något beskrivande
  - “maxNum”, “nrOfApples” eller “nr\_of\_apples”
- Namn kan inte börja på siffror och får inte innehålla
  - `- % = + - / * } ? “ ‘ . , !`
- Vissa ord är reserverade och får inte heller användas
  - `let const for if else switch, etc...`



# Variabler

Demo: Vi kan deklarera variabler på 3 olika sätt

- **let** (vi kan ändra innehållet)
- **const** (vi kan inte ändra innehållet)
- **var**(använd inte denna)

```
const price1 = 5;  
const price2 = 6;  
let total = price1 + price2;
```



# Variabler

## Övning

Skapa ett dokument variables.html och länka denna gång till extern script

```
<!DOCTYPE html>
<html>
<body>

<script src="variables.js"></script>

</body>
</html>
```



# Variabler

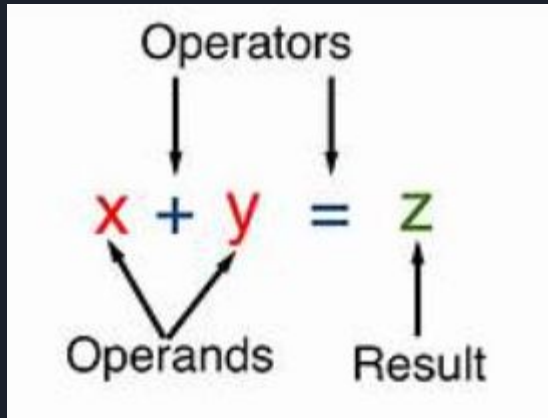
## Övning

- Skapa en let variabel med innehåll 1
- Ändra innehållet i samma variabel till 2
- Skapa en const variabel med innehåll 11
- Försök ändra innehållet i samma variabel till 12 och se felmeddelande

# Variabler

= kallas assignment operator

I matematik är vi vana vid omvänd förhållande



```
let z = x + y;
```





# Variabler

Assignment operator =

- Till vänster har vi en variabel alltid
- Till höger har vi ett uttryck (expression)

```
let z = x + y;
```



# Variabler

Demo: Uttryck (Expressions) kan ha följande värden

- **Matematiskt uttryck:**  $1$ ,  $100.5$ ,  $(1 + 3) * 7 - 8$ ,  $8 / 4$
- **Sträng:** "Hej hur mår du?"
- **Boolean:** `true`, `false`
- **Ingenting:** `undefined`



# Variables

## Övning

I samma dokument (variables.html och variable.js)

- Skapa en konstant variabel "x" med värdet 2
- Skapa en let variabel "secret" med det matematiska uttrycket:  $(3 * x + 4)$
- Försök att räkna ut vad secret är i huvudet och logga ut i konsolen och bekräfta
- Återanvänd "secret" och dubblera innehållet, hur gör du det?
- Logga ut i konsolen och se att du har lyckats
- Skapa en konstant variabel "message" med sträng värdet "Hello world" och logga ut
- Skapa en let variabel "allowedToEnter" med booleanska värdet true och logga ut
- Ändra "allowedToEnter" till false och logga ut



# Variabler

Dom typer av värden som en variabel får ha kallas "Datatyper"

- **Number:** 1, 100.5,  $(1 + 3) * 7 - 8$ ,  $8 / 4$
- **String:** "Hej hur mår du?"
- **Boolean:** true, false
- **Undefined:** undefined



# Variabler

Lite frågor om uttryck... Vad tror ni händer om vi...

- “Hej ” + “mår ” + “du?”
- “Hej” - “j”
- true + false
- true + true



# Variabler

Lite frågor om uttryck.... Vad tror ni händer om vi...

- “Is that ” + true
- “Hej” \* 3
- 1.99 + 0.01
- “5” + 4
- let x = 1; x = x + x;

# Jämförelser





# Jämförelser

Ofta vill vi undersöka ett värde och jämföra det med något annat

- Om vi kan checka ut `car` ifall den är tom (`car > 0`)
- Om åldern är tillåten för att få komma in på sidan (`age > 18`)
- Om användaren har validerat sin email (`validated == true`)
- Om användaren har skrivit bekräftat sitt lösenord (`password == confirmedPassword`)





# Jämförelser

Demo: Vi gör detta med operatorer (tidigare har vi sett = assignment operator)

- == (lika med)
- != (inte lika med)
- > (större än)
- >= (större eller lika med)
- < (mindre än)
- <= (mindre eller lika med)



# Jämförelser

## Övning

Skapa ett dokument `comparisons.html` länkat med externt script längst ned i bode `comparisons.js` som i föregående uppgift

```
<!DOCTYPE html>
<html>
<body>

<script src="variables.js"></script>

</body>
</html>
```



# Jämförelser

## Övning

- Skapa en variabel “name” med ditt namn
- Skapa en variabel “other” med någon annans namn
  - Logga ut likhetsjämförelse (==) mellan “name” och “other” (borde vara false)
  - Ändra “other” så att det är ditt namn
  - Logga ut likhetsjämförelse (==) mellan dem (borde vara true)
- Deklarera 2 variabler “x” och “y” med 2 valfria tal
  - Logga ut större än-jämförelse (>) så att true loggas ut (du kanske behöver ändra värde)
  - Logga ut mindre än-jämförelse (<) så att true loggas ut (du kanske behöver ändra värde)
  - Logga ut inte lika med-jämförelse (!=) så att false loggas ut (du kanske behöver ändra värde)



# Jämförelser

Logiska operatorer: Vi kan också jämföra olika flera olika jämförelser

- `3 > 5 && "hej" === "hej"`
- `3 < 5 && "hej" === "hej"`
- `!(3 > 5)`
- `!(3 < 5 && "hej" === "hej")`
- `!!(3 > 5)`
- `!!!(3 > 5)`

Prova själva att använda `&&` och `!`

If else (Gör ett val)





# If else (Gör ett val)

Conditional statements: Vi har ett speciellt statement som vi använder ofta för att låta våra program göra val

```
let age = 17;

if (age > 18) {
  console.log("Du får komma in");
}
```

```
const hour = new Date().getHours();
let greeting;

if (hour < 18) {
  greeting = "Good day";
} else {
  greeting = "Good evening";
}

document.getElementById("demo").innerHTML = greeting;
```



# If else (Gör ett val)

If statement

## Syntax

```
if (condition) {  
    // block of code to be executed if the condition is true  
}
```



# If else (Gör ett val)

Innanför { } grupperar vi statements!

## Syntax

```
if (condition) {  
    // block of code to be executed if the condition is true  
}
```





# If else (Gör ett val)

## If else statement

```
if (condition) {  
    // block of code to be executed if the condition is true  
} else {  
    // block of code to be executed if the condition is false  
}
```



# If else (Gör ett val)

## Övning

Skapa ett dokument ifelse.html länkat med externt script längst ned i body ifelse.js som i föregående uppgift

```
<!DOCTYPE html>
<html>
<body>

<script src="variables.js"></script>

</body>
</html>
```



# If else (Gör ett val)

```
if (condition) {  
    // block of code to be executed if the condition is true  
} else {  
    // block of code to be executed if the condition is false  
}
```

## Övning

- Gör en variabel "hour" med ett värde mellan 0 - 24
- Skriv en if else statement så att
  - Om hour är större (>) än 18 och (&&) mindre än (<) 6
    - Printar ut "Det är mörk"
  - Annars
    - Printar ut "Det är ljus"



# If else (Gör ett val)

else if statements

```
if (condition1) {  
    // block of code to be executed if condition1 is true  
} else if (condition2) {  
    // block of code to be executed if the condition1 is false and condition2 is true  
} else {  
    // block of code to be executed if the condition1 is false and condition2 is false  
}
```



# If else (Gör ett val)

## Demo: else if statements

```
const time = new Date().getHours();
let greeting;
if (time < 10) {
  greeting = "Good morning";
} else if (time < 20) {
  greeting = "Good day";
} else {
  greeting = "Good evening";
}
document.getElementById("demo").innerHTML = greeting;
```



# If else (Gör ett val)

```
if (condition1) {  
    // block of code to be executed if condition1 is true  
} else if (condition2) {  
    // block of code to be executed if the condition1 is false and condition2 is true  
} else {  
    // block of code to be executed if the condition1 is false and condition2 is false  
}
```

## Övning (använd ifelse.js)

- Gör en variabel "age" med en godtycklig ålder
- Om "age" är mindre än (<) 15 logga ut "saft"
- Om "age" är större än (>) 15 och (&&) mindre än (<) 18 loggar ut "kan smaka"
- Om "age" är större eller lika med (>=) 18 "öl"
- Om "age" är större eller lika med (>=) 21 "Tequila, arriba!!!"



# If else (Gör ett val)

```
if (condition1) {  
    // block of code to be executed if condition1 is true  
} else if (condition2) {  
    // block of code to be executed if the condition1 is false and condition2 is true  
} else {  
    // block of code to be executed if the condition1 is false and condition2 is false  
}
```

Skriv ett program som (ifelse.js)

- Deklarerar en konstant variabel “myAge” med en ålder
- Deklarerar en konstant variabel “myName” med ditt namn
- Deklarerar en let variabel “friendsAge” med en godtycklig ålder
- Deklarerar en let variabel “friendsName” med ett namn
- Jämför “myAge” med “friendsAge” i en “if else” jämförelse.
  - Om myAge är störst logga ut “myName”
  - Om myAge är mindre logga ut “friendsName”
  - Om dom är lika stora logga ut “????”



# If else (Gör ett val)

Demo: Ta input

Det finns en funktion “prompt” som frågar användare efter input



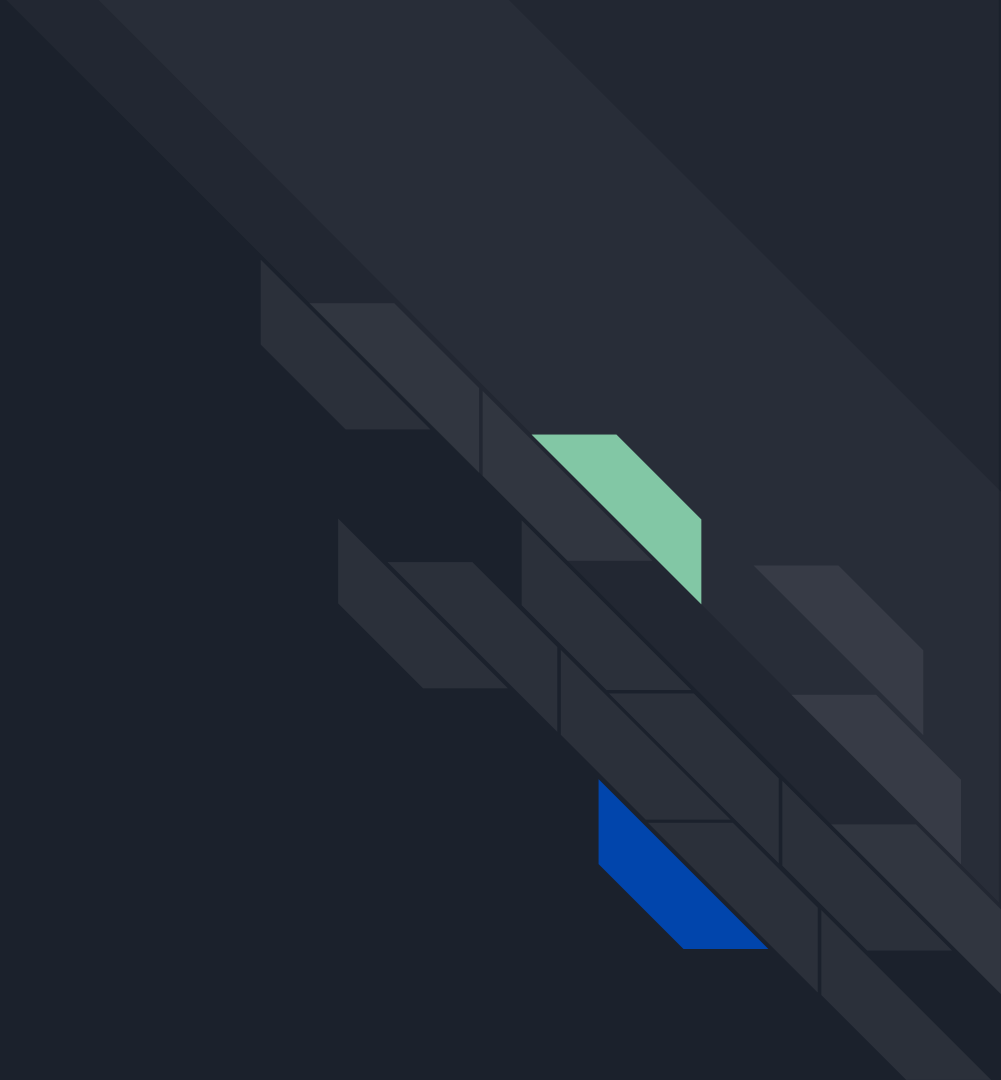


# If else (Gör ett val)

Skriv ett program (miniräknare)

- Skapa ett nytt dokument calculator.html och calculator.js
- Programmet
  - använder prompt för att deklarerera en variabel x
  - använder prompt för att deklarerera en variabel y
  - använder prompt för att deklarerera en variabel (operation): + - \* /
  - Med hjälp av dessa variabler gör uträkning
    - Ex: användare har gett x = 5, y = 7, operation = +
    - Resultat: Loggar ut 12

Loopar





# Loopar

Ibland vill vi göra samma sak flera gånger

```
let text = "";

for (let i = 0; i < 5; i++) {
  text += "The number is " + i + "<br>";
}

document.getElementById("demo").innerHTML = text;
```



# Loopar

```
for (let i = 0; i < 5; i++) {  
  text += "The number is " + i + "<br>";  
}
```

## Syntax

```
for (expression 1; expression 2; expression 3) {  
  // code block to be executed  
}
```

Expression 1: deklarerar en let variabel

Expression 2: jämför variabel med någonting

Expression 3: ökar variabelns värde



# Loopar

## Övning

Skapa ett dokument loop.html länkat med externt script längst ned i body loop.js som i föregående uppgift

```
<!DOCTYPE html>
<html>
<body>

<script src="variables.js"></script>

</body>
</html>
```



# Loopar

```
for (let i = 0; i < 5; i++) {  
  text += "The number is " + i + "<br>";  
}
```

## Övning

- Skriv en loop som printar ut ditt namn 10 gånger
- Skriv en loop som promptar användaren efter ett tal varje gång
  - Efter loop loggar ut summan av alla tal



# Läxa

Gör om övningar från föreläsning, repetera, repetera

Läs om de kapitel som kändes svåra... Du vet bäst vad du behöver göra...