

Programmierung - Lektion 2

...

PC 230901

Objekt

```
function start() {  
    ball = {}  
    ball.x = 100;  
    ball.y = 200;  
    ball.size = 20;  
    ball.color = "BlanchedAlmond";  
  
    circle(ball.x, ball.y, ball.size, ball.color);  
}
```

Ett smidigare sätt att definiera flera variabler hos ett **objekt**.

Om vi tänker oss en boll som vi vill definiera så har den flera egenskaper, t ex en position i x och y led samt en radie. Bollen kan ha flera egenskaper t ex en färg.

Då är det smidigt att skapa ett objekt som innehåller alla de här egenskaperna.

Egenskaperna behöver inte enbart vara variabler utan kan vara metoder som är knutna till objektet.

Objekt

```
function start() {  
    ball = {x: 100, y: 200, size: 20, color: "red"};  
    circle(ball.x, ball.y, ball.size, ball.color);  
}
```

Det finns ett kortare sätt att definiera ett objekt. Vi anger egenskaperna direkt när vi skapar objektet.

Notera att vi använder kolon när vi skapar objektet på detta sätt.

Skrivsättet ++

```
var x
```

```
var y
```

```
//Långa sättet
```

```
x = x + 1;
```

```
y = x - 1
```

```
//Korta sättet
```

```
x++;
```

```
y--;
```

Det är väldigt vanligt att vi vill räkna upp en variabel med ett steg. I de flesta moderna språk kan man använda en ++ operator.

JavaScript har även en operator för att minska med 1, den är naturligtvis --

Skrivsättet +=

```
var x  
//Långa sättet  
x = x + 100;  
//Korta sättet  
x += 100;
```

Det går även att öka med andra steg än 1, då används operatoren +=

Animering

```
function start() {  
    ball = {x: 0, y:0};  
}  
  
function update() {  
    circle(ball.x, ball.y , 35, "blue");  
    ball.x++;  
    ball.y++;  
}
```

Rörelse på en dataskärm fås genom att rita stillbilder efter varandra. Om jag vill att en boll ska röra sig över skärmen kan koden se ut så här

Animering

```
function start() {  
    ball = {x: 0, y:0};  
}  
  
function update() {  
    clearScreen();  
    circle(ball.x, ball.y , 35, "blue");  
    ball.x++;  
    ball.y++;  
}
```

Om vi vill ge illusionen av att bollen rör sig så rensar vi skärmen vid varje uppdatering

Villkor

```
function start() {  
    var name = prompt("Hej, vad heter du?");  
    var age = prompt("Hur gammal är du?");  
    text(50, 150, 20, "Hej " + name + "!", "black");  
    if (age < 18) {  
        text(50, 250, 20, "Du är inte myndig" , "black");  
    }  
    else {  
        text(50, 250, 20, "Du är myndig" , "black");  
    }  
}
```

Ett vanligt villkor i
många
programmeringsspråk
är if - satsen.

Den kan även
kombineras med else.

Jämförelseoperator i JavaScript

Beskriving	Operator
Lika med	==
Mindre än	<
Mindre än eller lika med	<=
Större än	>
Större än eller lika med	>=
Inte lika med	!=

Pröva följande program

```
// Gissa ett tal

let tal = prompt('Gissa ett tal');

if (tal == 42) {
    alert('Du gissade rätt');
} else if (tal > 42) {
    alert('Du gissade för högt');
}
else {    alert('Du gissade för lågt!');}
```

Switch och case

Om jag behöver skriva många if-satser för att kolla flera villkor så finns ett smidigare sätt som kallas case i många språk. I JavaScript börjar kommandot med switch.

Se koden på nästa sida och pröva den. Beskriv med egna ord hur den fungerar innan du provar att köra den.

```
switch(expression) {  
  case n:  
    code block  
    break;  
  case n:  
    code block  
    break;  
  default:  
    default code block  
}
```

```
<script src="https://koda.nu/simple.js">

  let fruit = prompt("Vad är din favoritfrukt?");

  let svar = "";

  switch (fruit) {
    case "Banan" :
      svar = "Bananerna är finfina idag";
      break; //Vi behöver inte kolla fler villkor
    case "Äpple":
      svar = "Gröna äpplen är goda";
      break; //Vi behöver inte kolla fler villkor
    case "Mango":
      svar = "Min favorit";
      break; //Vi behöver inte kolla fler villkor
    default:
      svar = "Jag känner inte till den frukten";
  }
  text(50,50,25,svar,"black");

</script>
```

Reagera på tangentbordet

I simple.js ingår funktioner för att “lyssna” på tangentbordet. I standard JavaScript måste detta hanteras lite annorlunda.

Notera i exemplet hur if-satserna skrivs väldigt kompakt.

```
function start(){
    ball = {x:0, y:0};
}

function update() {
    clearScreen();

    // Kontrollera om en tangent trycks ned
    if (keyboard.left) { ball.x--;}
    if (keyboard.right) {ball.x++;}
    if (keyboard.up) {ball.y--;}
    if (keyboard.down) {ball.y++;}

    circle(ball.x, ball.y , 50, "green");
}
```

Reagera på musen

I simple.js ingår funktioner för att “lyssna” efter hur musen rör sig och om någon knapp är nedtryckt. Här är ett simpelt ritprogram. Vi behöver endast en update funktion.

```
function update(){  
  circle(mouse.x, mouse.y, 10, "red");  
}
```

Ett något (men bara något) bättre ritprogram

Vi vill att vi ska rita om vi håller in vänster musknapp och sudda om man håller in höger musknapp. För att lyssna efter om vänster knapp är nedtryckt kan vi använda `mouse.left`. Vi ritar med röd färg och suddar genom att måla över med vit färg.

```
function update(){  
  if (mouse.left) {circle(mouse.x, mouse.y, 10, "red"); }  
  if (mouse.right) {circle(mouse.x, mouse.y, 10, "white"); }  
}
```