

## Textbaserad programmering

I kursen har vi provat på att skapa program med blockprogrammering. Tanken bakom många blockbaserade programspråk är att det ska vara enkelt att komma igång, gå att använda för en mängd olika ändamål och situationer samt inte vara för begränsat. Det finns dock situationer och lägen där de textbaserade språken lämpar sig bättre för den typ av program man ska skapa – då är det läge att gå över till ett textbaserat språk.

Största delen av all programutveckling i samhället sker med textbaserade språk och det finns en mängd olika språk att välja bland. För nybörjare rekommenderas ofta Python, Javascript, Kojo eller Ruby. Nybörjarspråken har till exempel en mindre ordrik *syntax*, det vill säga grammatik. För att få utskriften "Hej" att dyka upp på skärmen i Python räcker det att skriva enbart

```
print("Hej")
```

medan man i Java behöver skriva följande kod:

```
class Hej {  
    public static void main (String args []){  
        System.out.println("Hej");  
    }  
}
```

Språken har också andra skillnader som gör att de passar i olika sammanhang. En del språk är snabba att skriva program i, medan en del har egenskaper som gör att de lämpar sig extra bra i en specifik situation.

### **Likheter och skillnader**

De flesta programmeringsspråk grundar sig på samma byggstenar och du hittar de begrepp och konstruktioner vi talat om i samband med blockbaserad programmering, såsom repetition, alternativ, variabler och funktioner, även i textbaserade programmeringsspråk. I stället för att använda blocket "repetera 50" kan vi skriva motsvarande i text – hur detta ser ut beror på vilket språk vi använder. Alla textbaserade programmeringsspråk har en egen syntax, det vill säga regler som säger hur man ska skriva program i just det språket. Då vi går över från ett blockbaserat språk till ett textbaserat språk är mycket bekant sedan tidigare och vi kan dra nytta av sådant vi redan kan. Forskning har visat att erfarenhet av blockbaserad programmering gör det enklare att komma igång med textprogrammering (Grover, 2015).

Naturligtvis finns det också skillnader mellan att programmera med block och att skriva kod i textform. Att pussla ihop block gör att vi slipper fundera på syntaxfel (att skriva fel, glömma bort viktiga tecken osv.) eftersom språket är byggt så att man endast kan pussla ihop block som inte bryter mot språkets grammatiska regler. Då vi skriver program i text är det viktigt att alla tecken finns på exakt rätt plats och att allt stavats rätt. Dessutom är det i de flesta språk också viktigt att skilja mellan stor och liten bokstav. Textbaserade språk är generella och kan användas till det mesta, vilket gör att man kan hitta en mängd färdig kod inom olika områden som man kan återanvända i sina egna program. Om du till exempel vill skriva kod som styr poängräknaren i ett spel kanske du kan hitta någon annan som redan skrivit den koden. Då kan du kopiera, återanvända och vid behov modifiera den färdiga koden i ditt spel och slipper börja från början.

En annan stor skillnad ligger i det resultat vi kan förvänta oss av de program vi bygger. Medan vi från blockbaserade språk är vana vid att direkt få ett snyggt visuellt resultat i de program vi skapar, är det inte lika lätt att åstadkomma i textbaserade språk. I blockbaserade språk är vi dessutom vana vid att skriva program som körs baserat på händelser (en knapp trycks ned, ljudnivån överstiger en viss nivå, ett villkor uppfylls), medan det inte är lika lätt att åstadkomma i textbaserade språk. Det är så klart helt möjligt att få snygg grafik och bygga ett program utgående från händelser även då man skriver program i textform (det ser vi till exempel i alla datorspel), men detta är inte något man kan börja med som nybörjare.

### **När är det dags att gå över till ett textbaserat språk?**

Diskussionen kring programspråk var länge ett av de hetaste debatt- och forskningsområdet inom datavetenskap på universitetsnivå. I dagens skoldiskussion är en vanlig frågeställning när man "borde" eller "ska" gå över från ett blockbaserat språk till ett textbaserat språk. Tyvärr går det inte att ge några klara riktlinjer eller rekommendationer, utan det är ett beslut som man måste fatta baserat på vad man vill åstadkomma.

Som vi redan konstaterat har blockbaserade och textbaserade språk mycket gemensamt men också en hel del skillnader. Blockbaserade språk gör det möjligt att snabbt åstadkomma snygga visuella program. För att redovisa och göra presentationer, skapa spel, bygga berättelser, animera dialoger eller simulera fenomen passar därför blockbaserade programspråk. Dessa språk är dock begränsade (det finns ett ändligt antal block), medan textbaserade språk är generella. Då vi vill göra något som inte går att åstadkomma med ett blockbaserat språk är det alltså dags att gå över till ett textbaserat språk. Likaså är det inte alltid värt att hålla fast vid det blockbaserade språket även om det man vill åstadkomma går att göra i språket – om det är mycket enklare och effektivare att åstadkomma i ett textbaserat språk är det klart värt att fundera på att byta. Exempelvis går det att hantera och analysera större mängder data i Scratch, men det går mycket snabbare och enklare att bygga ett motsvarande program i Python.

Studier visar att en del upplever blockbaserad programmering som "lek" medan textbaserad programmering är "riktig" programmering (Weintrop och Wilensky, 2015). Detta hänger troligen ihop med kod-begreppet, där textbaserade program ses som autentiska. Då är det igen viktigt att påminna oss om att programmering är en process som handlar om mer än att bara skriva kod – att bygga en lösning i ett blockbaserat programspråk är precis lika mycket programmering som att göra det i ett textbaserat sådant.

### **Referenser**

Grover, Shuchi (2015) *"Systems of Assessments" for Deeper Learning of Computational Thinking in K-12*. Annual Meeting of the American Educational Research Association, Chicago, 15-20 april 2015.

Google (2015) Images of Computer Science: Perceptions Among Students, Parents and Educators in the U.S. Gallup research. Tillgänglig på webben: <https://services.google.com/fh/files/misc/images-of-computer-science-report.pdf>.

Skolverket (2017) Få syn på digitaliseringen på grundskolnivå. Ett kommentarmaterial till läroplanerna för förskoleklass, fritidshem och grundskoleutbildning.

Weintrop, David & Wilensky, Uri (2015) To block or not to block, that is the question: students' perceptions of blocks-based programming. In *Proceedings of the 14th International Conference on Interaction Design and Children (IDC '15)*. ACM, New York, NY, USA, 199-208.