Examinator:
Anders Haraldsson

TDP019: Projekt: Datorspråk
Användarhandledning version 1.2
2010-05-23

Benjamin Wallin
Mikael Andersson
Grupp: IP1-A-1

Användarhandledning
Version 1.2

Innehåll

Bakgrund

Xtat som språket heter kommer ursprungligen från att vi ville göra ett språk som enkelt kan hantera xml-filer och utifrån dem skapa statistik med diagram och tabeller. X står för xml och tat, som ger stat i uttal, är en förkortning av statistik.
Målet var även att hitta tecken och kombinationer som gör att man får ett flyt vid programmeringen och där av har vi använt oss av: §, eftersom den är lätt att nå och skriva.

Börja programmera i Xtat

Allmänna tips

Några allmänna tips som du bör tänka på när du programmerar är att hålla dig till ett språk, dels när du döper variabler och även i dina kommentarer.
Rekommendationerna är att du håller dig till engelska för att slippa problem som kan uppkomma med tecken som t.ex. å, ä och ö.
För att man enkelt ska kunna gå in och ändra i koden vid ett senare tillfälle bör man även skriva lagom med kommentarer som beskriver vad olika saker gör för något.
Det sista tipset handlar om något som vi använder oss av i exempelkoden, det kallas indentering, vilket ofta är ett tab-tecken, man kan även använda mellanslag och då används vanligtvis 4 mellanslag. Detta används för att visa vilken kod som hänger ihop med vilken.

Alla våra tips kommer tillämpas i hela denna handledning så att du blir väl medveten om dem.

Grunderna

För att visa att det är Xtat du programmerar så omsluter du din kod med
<?xt och xt?> -taggarna. Det kommer då att se ut så här:

```
<?xt
  /* Din kod i språket Xtat. */
xt?>
```

Filen som du sedan ska ladda upp på servern ska ha ändelsen **.xt** . Startsidan får då lämpligtvis namnet **index.xt**. För att det ska fungera måste även webbservern vara rätt konfigurerad och stödja språket Xtat, läs mer om det i kapitlet *Webbhotell*.

Du kan även använda Xtat tillsammans med vanlig html, se exempel nedan, för att programmera din hemsida.
Det kan då se ut så här:

```
<html>
  <body>
    <strong><?xt print 'hello world'  xt?></strong>
  </body>
</html>
```

Kommentarer i språket

När man skriver mycket kod så är det viktigt att skriva bra kommentarer i koden, dels för att man själv ska komma ihåg vad t.ex. en funktion gör och dels för att förenkla för att andra ska kunna gå in och göra ändringar.
I Xtat så skriver man kommentarer genom att skriva:

```
/*  Kommentar  */
```

Man kan även skriva kommentarer på flera rader, då ser det ut så här:

```
/*
      En längre kommentar
      på flera rader.
*/
```

Variabler

Xtat använder sig av ett *∫-tecknet* (paragraftecken) för att skapa variabler. I variablerna kan man lagra saker som exempelvis siffror, arrayer eller strängar.

Siffror

```
$int = 20
$float = 4.55
```

Strängar

```
$text = 'hej jag programmerar i Xtat'
$newtext = 'Kan skriva med en siffra 25'
```

För att lätt kunna veta vad som är konstanter eller inte så ska man använda stora bokstäver när man skapar konstanter, det ser då ut så här:

Konstanter

```
$PI = 3.14
```

Matematik

Man vill ofta göra olika beräkningar och för det så kan man använda sig av vanliga matematiken addition (+) , subtraktion (−), multiplikation (*) och division (/). Prioriteten för de olika räknesätten är att multiplikation och division har högre prioritet än addition och subtraktion och beräknas alltså före. Vill man att det ska beräknas på ett annat sätt så kan man använda parenteser runt det som man vill ska räknas ut först.

Kod före

```
$math = 3 * 2 + 9
$math2 = 3 * (2 + 9)
```

Beräknad

```
$math = 15
$math2 = 33
```

Har du lagrat en siffra i en variabel så kan du även använda variabeln i dina beräkningar.

```
$number = 15
```

Kod före

```
$newnumber = $number + 10
$newnumber2 = $number / 3 * 4
```

Beräknad

```
$newnumber = 25
$nwenumber2 = 20
```

Arrayer

Ibland vill man kunna lagra fler saker än ett tal eller sträng i en variabel, det är det som arrayer är till för. I Xtat kan man lagra både tal, strängar och andra variabler i en array.

```
$myarray = array(1,2,3,'sträng',$var)
```

Dessa värden är då lagrade i `$myarray`. Du kan dock inte skriva ut `$myarray` på samma sätt som du kan med andra variabler utan du måste använda dig av en loop för att iterera igenom varje element som du sedan kan skriva ut. Läs kapitlerna ”Utskrift på skärmen” och ”Loopar” för mer information om hur det fungerar.

Utskrift på skärmen

Ofta vill man kunna skriva ut beräkningarna man gör eller allmänna texter så att de syns för användaren . Detta gör man med `print` och sedan det man vill skriva ut, t.ex. en variabel, sträng eller nummer.

Kod	*Utskrift*
`print 25`	`25`
`print 'en sträng'`	`en sträng`
`print 'en sträng med 25'`	`en sträng med 25`
`$string = 'strängen'`	
`print $string`	`strängen`

If-satser

If-satser är en typ av en frågesats, där den gör en sak om villkoret är uppfyllt och en annan sak om den inte är det. För detta använder man sig av olika villkorsoperationer, de olika som finns förklaras nedanför.

Villkorsoperationer

`==` Kontrollerar om värdena på båda sidor är lika, är så fallet så returneras true eller false.

`!=` Kollar så att hägersidan är skild från vänstersidan, är den det så ger den true annars false.

`>=` Förväntar sig siffror på båda sidorna och kolla om högersidan är större eller lika med än vänster, då retunerar den true annars false.

`<=` Samma som ovan men retunerar true om högersidan är mindre eller lika med än vänstersidan.

Exempel 1

```
$var = 10
If($var == 10)
   print 'Hej!'
else
  print 'nej..'
end
```

Utskrift

```
Hej!
```

Exempel 2

```
$var=10
If($var <= 9)
     Print 'talet är litet'
elseif($var == 10)
    print 'talet är 10'
else
     print 'konstigt tal'
end
```

Utskrift

```
talet är 10
```

Har man enkla villkorsoperationer att skriva finns det ett enkelt sätt att göra gör detta på en rad med enrads-if-funktioner. Den kan vara lämplig när man vill göra små utskrifter, t.ex. kontroll av om ett tal är rätt eller fel.

Exempel

```
$mynumber = 10
($mynumber == 10) ? print 'rätt' : print 'fel'
```

Utskrift

```
rätt
```

Loopar

Har du läst tidigare kapitlet ”Arrayer” så förstår du kanske lite vad du kan använda loopar till. Du kan helt enkelt loopa igenom en array eller ett visst antal gånger och göra något för varje varv som den körs.
I Xtat så finns det två olika loopar som pressenteras här under.

For-loop

For-loopen är bra om man vill kunna iterera igenom ett givet område, t.ex. loppa från 1 till 10. Den fungerar så att man ger tre argument där första är startvärde, den andra är hur länge den ska loopa och den sista hur man stegar sig framåt från startvärdet.

Exempel 1

```
for(§i = 1, §i <= 5, §i++)
     /* Din kod */
end
```

Exempel 2

```
for(§i=1, §i =< 5, §i++)
      print §i
end
```

Utskrift

```
1 2 3 4 5
```

Förklaring

I detta exempel så börjar man med att tilldela variabeln §i värdet 1. För varje iteration så skriver den ut talet som §i innehåller. Den adderar även på §i med ett i slutet av varje iteration och följer samma mönster tills §i innehåller ett värde som är större än 5.

Så vill du kunna skriva ut långa talföljder kan en for-loop vara en enkel lösning.

While-loop

Vet du inte hur många gånger du ska loopa så är while-loopen en bättre lösning. Den loopar helt enkelt så länge som argumentet är true, blir det false så slutar den. Vi visar med exempel.

Exempel

```
$args = true
$i = 2
while($args)
  print 'skriver ut'
  $i = $i + 2
  if($i >= 10)
      $args = false
  end
end
```

Utskrift

```
skriver ut skriver ut skriver ut skriver ut skriver
ut
```

Som du ser så lägger vi även på tal på `$i` i loopen men det är inget som skrivs ut utan används endast för att kontrollera om if-villkoret uppfylls. Här ser ni även vikten i att använda indentering när ni programmerar.

Funktioner

Funktioner är något som är väldigt användbart eftersom man kan skapa funktioner för att sedan anropa de på flera ställen, på så sätt kan man bryta ut kod som förekommer på många ställen och skapa en funktion istället. På det sättet slipper man återupprepning samt om man vill ändra på något behöver man bara göra det på ett ställe.

Exempel

```
func max($x,$y,$z)
  $maxnum = $x
  if($maxnum < $y)
    $maxnum = $y
  end
  if ($maxnum < $z)
    $maxnum = $z
  end
   return $maxnum
end
```

Anrop till funktionen

```
$maxnum = max(1,2,3)
print $maxnum
```

Utskrift

```
3
```

Det går även att sätta ett standardvärde på t.ex. det sista argumentet i parameterlistan. Då används det värdet om man väljer att inte använda den parametern vid funktionsanropet.

Exempel

```
func max($x,$y,$z = 0)
  $maxnum = $x
  if($maxnum < $y)
    $maxnum = $y
  end
  if ($maxnum < $z)
    $maxnum = $z
  end
   return $maxnum
end
```

Anrop till funktionen

```
$maxnum = max(1,2)
print $maxnum

$maxnum2 = max(1,2,3)
print $maxnum2
```

Utskrift

```
2 3
```

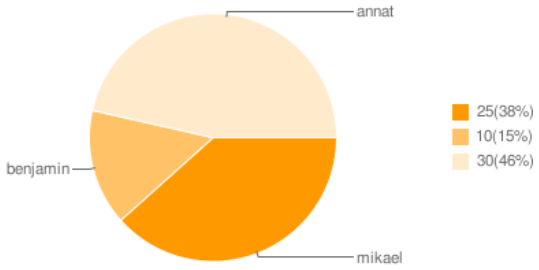
Xtat-funktioner

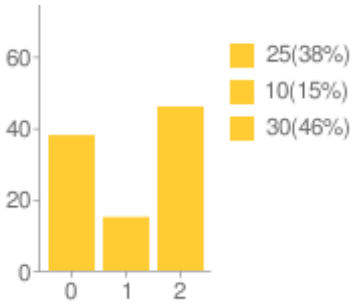
Då Xtat är till för de som arbetar mycket med statistik så finns det funktioner för att skapa diagram på liknande sätt som man använder en while-loop eller if-sats, som vi tagit upp i tidigare kapitel. Xtat innehåller idag två olika typer av diagram, ett cirkeldiagram och ett stapeldiagram och båda fungerar på liknande sätt som visas här under. Funktionerna vill ha tre argument, en array med siffror, en array med namn och slutligen en typ av ändra % eller st. Det är dock endast arrayen med siffror som är obligatorisk. Dessa arrayer lagras först i en variabel som man sedan använder i funktionsanropet. Se kodexempel nedan.

Exempel

```
$number = array(25,10,30)
$names = array('Mikael','Benjamin','Annat')
circle($number,$names)
bar($number,$names)
```

Utskrift





En enkel hemsida

Nu har du lärt dig språket Xtat. För att visa hur man kan tillämpa Xtat-språket så har vi gjort ett exempel på en enkel hemsida som använder sig av språket Xtat. En demoversion där språket är tillämpat tillsammans med html finns för visning på http://bnet.info/test/testfile.xt, om ni inte själva väljer att ladda upp den på en egenkonfigurerad server.

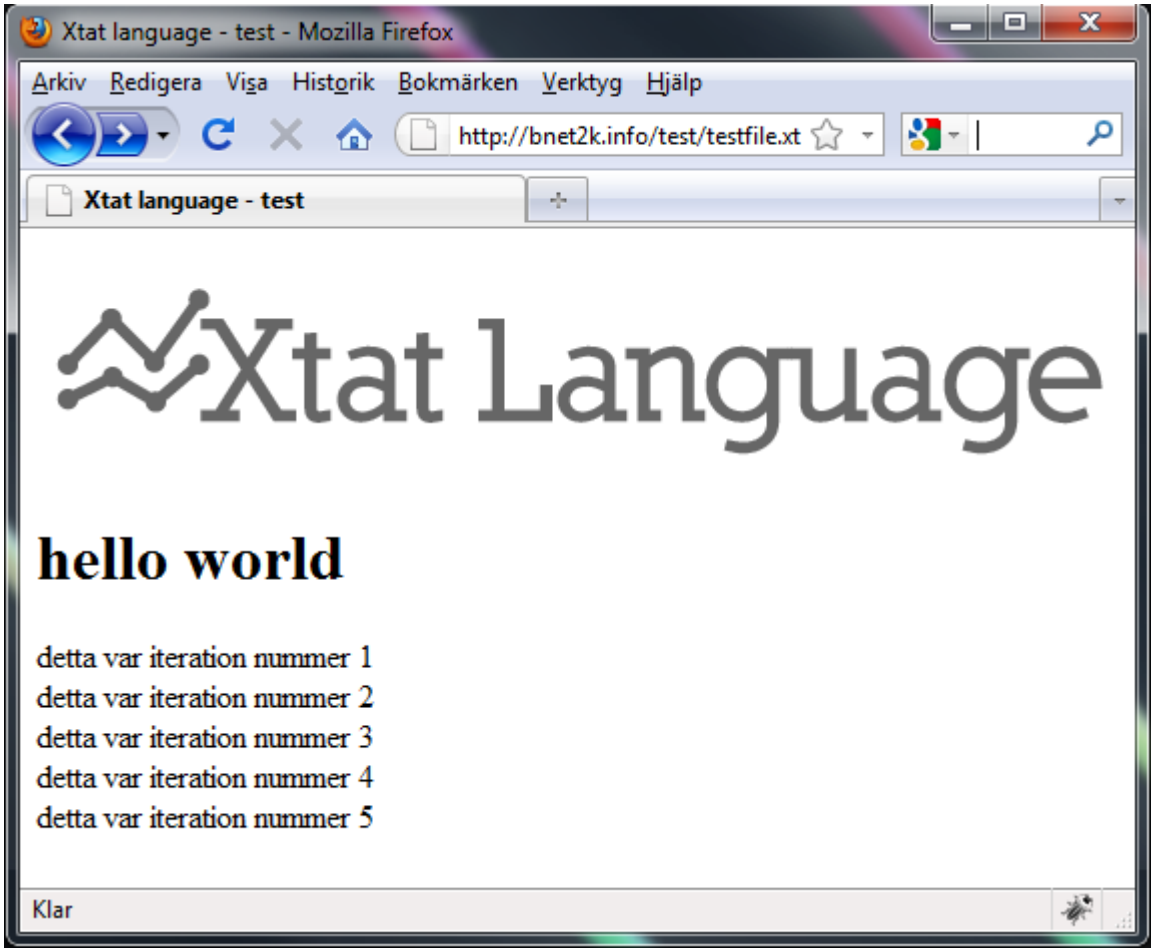


Bild 1: En exempelhemsida

Kod för bild 1

```
<html>
  <title>Xtat language - test</title>
  <body>
    <img src='/xtat_logo.png' alt='xtat-logo'/><br/>
    <h1><?xt print 'hello world' xt?></h1>
    <?xt
      for(§i=1,§i <= 5,§i++)
        print 'detta var iteration nummer '
        print §i
        print '<br />'
      end
    xt?>
  </body>
</html>
```

Webbhotell

Då det inte är så många webbhotell som stödjer språket Xtat än så tänkte vi tipsa om att ni kan kontakta Benjamin genom *benwa852@student.liu.se* för ett konto på en server som stödjer språket.

Har ni en egen server och vill installera ett stöd för Xtat så bör ni läsa dokumentationen *implementationshandledning.pdf* - som finns att ladda ner på Xtat:s projekthemsida.