

Lösningar teknik HT17 (uppdaterad v. 40)

Kom ihåg att prata med andra och fråga under lektionerna om du inte förstår lösningarna!

1. Se guiden för att spara och öppna filer på teknikbloggen under kategorin ”guider”. Fråga under lektionen om det fortfarande är oklart!
2. Kod innehåller exakta instruktioner om vad datorn ska göra. När man kör kod läser datorn instruktionerna och utför det den ska göra. I det interaktiva läget i Python skriver man en rad i taget, och kör raden man har skrivit genom att trycka på Enter-tangenten. I standardläget kan man skriva flera rader kod och kör alla rader man har skrivit genom att trycka på Run->Run Module (F5). Python läser då rad efter rad tills det inte finns några rader kvar att läsa. Då har programmet kört klart.
3. Variabeln abc har namnet abc, värdet 10 och är av typen int (heltal). Variabeln number har namnet number, värdet 10 och är av typen float (decimaltal). Variabeln student_name har namnet student_name, värdet ”Anna”, och är av typen string (tecken).
4. Rad ett är ok.
Rad två ger fel på grund av att det är ett mellanrum i variabelnamnet.
Rad tre är ok.
Rad fyra ger fel eftersom det är en siffra i början av variabelnamnet.

5. Det som skrivs ut:

```
Datorn läser det här!
```

6. `print( )` används för att skriva ut text till användaren, vilket är en slags output.
 - a. Man kan skriva:

```
print("Hello World!")
```

- b. Koden skriver ut:

```
Oscar  
namn
```

Förklaring: `namn` är namnet på en variabel med värdet ”Oscar”, så `print(namn)` skriver ut ”Oscar”. ”`namn`” (med citationstecken) är ett data av typen string utan ett namn. Man kan se det som att värdet

"namn" ligger i en låda i minnet som ingen har satt en etikett på. Alltså skriver `print("namn")` ut "namn".

7. Koden skriver ut:

```
Oscar
```

```
Oscar
```

8. Input använd för att lagra text som användaren (personen som kör programmet) skriver in. Man kan t.ex. fråga efter användarens namn och lagra detta i en variabel.

9. När man kör programmet skriver den först ut:

```
Skriv in ditt namn här--->
```

och väntar på att användaren ska skriva in något efter pilen. När användaren har skrivit in något och tryckt på Enter-tangenten, så lagras de tecken som användaren har skrivit in i variabeln `elev_namn_1`. Rad två och tre gör exakt samma sak igen.

10. I rad två saknas det en parentes.

I rad tre saknas det citationstecken kring "Anna" och så innehåller variabelnamnet ett mellanslag, vilket är förbjudet i Python.

I rad fem saknas det citationstecken kring "hej, jag heter Anna"

11. If, elif och else används i programmering för att välja att en viss kod ska köras beroende på om ett visst villkor är uppfyllt.

- 12.

När man sätter `x = 2` skriver programmet ut:

```
Inne i den första elif, x = 2
```

När man sätter `x = 1` skriver programmet ut:

```
Nu körs if, alltså är x = 1
```

När man sätter `x = 3` skriver programmet ut:

```
Inne i den andra elif, x = 3
```

När man sätter `x = 100` skriver programmet ut:

`x` är varken 1, 2 eller 3

13. För facit, skriv av koden (eller kopiera in den i IDLE) och testa. Kom ihåg att du måste byta ut citationstecken när du klistrar in kod i IDLE, eftersom de är olika tecken i Word och IDLE.
14. För facit, skriv av koden (eller kopiera in den i IDLE) och testa.
15. Programmet uppmanar användaren att skriva in Sveriges eller Norges huvudstad, och skriver sedan ut "Rätt" om användaren svarade Stockholm eller Oslo, eller "Fel" om användaren svarade något annat. Notera att **or** används när man vill att kod ska köras om något av flera villkor är uppfyllt. På samma sätt används **and** när man vill att samtliga av flera villkor ska vara uppfyllda.
16. Prova i standardläget av Python, läs kurslitteraturen och fråga under lektionen om det är oklart vad som händer.
- 17.

```
x = 2
while x < 11:
    print(x)
    x = x + 2
```

18. Prova i standardläget av Python, läs kurslitteraturen och fråga under lektionen om det är oklart vad som händer.
19. Prova i standardläget av Python, läs kurslitteraturen och fråga under lektionen om det är oklart vad som händer.
20. Prova i standardläget av Python, läs kurslitteraturen och fråga under lektionen om det är oklart vad som händer.
21. Programmet skriver ut:

SPAM!

Funktionsdefinitionen är:

```
def spam():
    print("SPAM!")
```

Funktionen anropas med:

```
spam()
```

22. Programmet skriver ut "SPAM!" 100 gånger. Funktionen anropas med argumentet 100.
23. Programmet lagrar värdet 9 i variabeln x. Funktionen returnerar summan av argumenten den anropas med. Den definieras med två parametrar: a och b. Den anropas med argumenten 3 och 6. Värdet 9 lagras i x.
24. Testa själv! Fråga under lektionen om oklart.
25. Man kan t.ex. skriva såhär:

```
def print_name(given_name, sir_name):  
    print("Förnamn: ", given_name)  
    print("Efternamn: ", sir_name)  
  
print_name("Molly", "Svensson")
```