
Instuderingsfrågor inför Prov 2 i Python

Provet definieras inte av dessa frågor utan av **kursboken**, följande sidor:

Teori:	<i>Koda matte med Python</i> , avsn. 1.5 – 1.10, sid 54-102.
Övningar:	1.5, sid 62. 1.6, sid 69. 1.7, sid 76. 1.8, sid 85. 1.9, sid 97. 1.10, sid 102.
Lösningar:	1.5-1.10, sid 130-154.

Svar till alla instuderingsfrågor kan hittas i kursboken *Koda matte med Python*.

1.5 Gissa tal – ett spel, kursboken sid 54-62 & 130-133

1. Med vilken sats skriver man i Python ett *enkelt* val (utan alternativ)?
2. Med vilken sats skriver man i Python ett *tvåvägsval* (mellan två alternativ)?
3. Med vilken sats skriver man i Python ett *tre-* eller *flervägsval* (mellan tre eller flera alternativ)?
4. Skriv ett program som läser in tre heltal, hittar och skriver ut det *största* av dem med hjälp av enkla *if*-satser.
5. Modifiera programmet i fråga 4 så att det blir det *minsta* av de tre talen.

Se programmet **GissaTal_1** på sid 54. Testa koden om du ev. inte kan besvara frågorna teoretiskt:

6. Vad händer om man, när **GissaTal_1** körs, matar in tal > 20 eller < 1 ?
7. Vad händer om man, när **GissaTal_1** körs, av misstag matar in en bokstav istället för tal?
8. Vad är skillnaden mellan koderna `=` (t.ex. i rad 5) och `==` (i rad 8) i Python?
9. Val mellan hur många alternativ gör *if-elif-else*-satsen i programmet **GissaTal_1**?
10. Hur många villkor har *if-elif-else*-satsen? Varför står efter *else* inget villkor?
11. Kan man i programmet **GissaTal_1** ersätta *if-elif-else*-satsen i raderna 8-13 med tre separata, enkla *if*-satser? I så fall gör det och testa!
12. Vad är nackdelarna i Gissa tal-spelet, version 1?

Testa och experimentera gärna med koden **GissaTal_2** i Visual Studio, kursboken sid 56:

13. Varför sätts i rad 8 `wrongGuess` till `True`? Skulle programmet även fungera med `False`? Motivera!
14. Varför kallas `wrongGuess` för en *logisk variabel*?
15. Kan logiska variabler i Python tilldelas även 0 och 1?
16. På vilket sätt styr den logiska variabeln `wrongGuess` *while*-loopen i **GissaTal_2**?
17. Betyder namnet `wrongGuess` att denna variabel alltid lagrar en fel gissning? Motivera namnvalet.
18. Vilka rader tillhör *while*-loopen?

19. Kan **while**-loopens villkor ersättas av `wrongGuess = True` ?
20. Kan **while**-loopens villkor ersättas av `wrongGuess == True` ?
21. Kan man i koden se hur många varv **while**-loopen kommer att ha?
22. När avslutas **while**-loopen i **GissaTal_2**? Kan den bli en evighetsloop? Motivera ditt svar.
23. Vilken variabel räknar antalet varv?
24. Vad är nackdelarna i Gissa tal-spelet, version 2?

Testa och experimentera gärna med koden **GissaTal_3** i Visual Studio, kursboken sid 58:

25. Vad precis gör **import**-satsen i rad 6? Varför behövs den? På vilken rad används det som importeras?
26. Är koden **random.random()** ett *anrop* eller *definitionen* av funktionen **random()**?
27. Vad är skillnaden mellan **random** och **random()**?
28. Vilken typ av tal slumpar funktionen **random()** och i vilket intervall hamnar slumpalen? Ange det störst och det minst möjliga. Vet du inte, testa!
29. Var någonstans är funktionen **random()** definierad?
30. Varför har vissa satser i **GissaTal_3** inga indragningar, andra en indragning (på 3 blanktecken) och vissa andra två indragningar (på 3 blanktecken var)?
31. Görs indragningarna i fråga 23 av estetiska skäl (God programmeringsstil, sid 18) eller tillhör de Pythons syntax och producerar felmeddelande om man inte gör dem?
32. Hur många villkor har **if-elif-else**-satsen? Varför står efter **else** inget villkor?
33. Val mellan hur många alternativ görs i **if-elif-else**-satsen i programmet **GissaTal_3**?
34. På vilket sätt realiserar programmet användarens ev. önskemål om att avbryta spelet och avslöja programmets hemliga tal?
35. På vilket sätt skulle man kunna vidareutveckla Gissa tal-spelet, version 3?
36. Vad åstadkommer algoritmen Intervallalvering på sid 59? Testa den. Är den verkligen optimal?
37. Varför är algoritmen Intervallalvering *rekursiv*?
38. Skriv en **while**-sats som skriver ut de 50 första positiva heltalen.
39. Skriv en **while**-sats som beräknar och skriver ut summan $1 + 2 + 3 + \dots + 100$.

Testa och experimentera gärna även i fortsättningen med programmen i Visual Studio:

1.6 Hantering av slumpal, kursboken sid 63-69 & 134-137

40. Redogör för begreppet *pseudoslumptal*.
41. Varför genererar koden `1 + int(random.random() * 20)` heltalsslumptal mellan 1 och 20?
42. Kan man i koden se hur många varv en **for**-loop kommer att ha?
43. Hur många varv har **for**-satsen i rad 8-9 av programmet **Slumptal** på sid 63?
44. Det finns ett skrivfel i kommentaren på rad 14. Kan du upptäcka det?
45. Vad betyder **range** i **for**-satsens huvud?

46. Vad innebär parenteserna i `range( , )` i for-satsens huvud?
47. Mellan vilka två reserverade ord står for-satsens räknare?
48. Beskriv med egna ord for-satsens flödesplan på sid 64.
49. Till vilket värde initieras for-satsens räknare?
50. Ser man i koden att for-satsens räknare uppdateras?
51. Kan for-satsen bli en evighetsloop? Motivera ditt svar.
52. Varför måste man initiera `slumplist` till en tom lista på rad 13 i programmet `Slumpta1` på sid 64?
53. Gör de båda plustecknen på rad 15 samma sak? Om inte, vad är deras skillnad? Hur kan Pythons interpretator skilja deras funktionalitet?
54. Koden `sorted(slumplist)` på rad 16 sorterar slumplistan. Är koden ett anrop eller definitionen av funktionen `sorted()`?
55. Skriv en for-sats som skriver ut de 50 första positiva heltalen.
56. Skriv en for-sats som beräknar och skriver ut summan $1 + 2 + 3 + \dots + 100$.

1.7 Funktioner i programmering, kursboken sid 70-76 & 138-141

57. Vad menas med *modularisering* i programmering?
58. Vad är målet med modularisering?
59. Redogör för begreppet *funktion* i programmering.
60. Varför måste en funktion som en modul som separeras från övrig kod, ha ett *namn*?
61. Om man jämför en funktion med en svart låda, vad kallas de värden som kommer in i den och det värdet som kommer ut ur den?
62. Om man jämför en funktion med en legobit, vad kan legobitens taggar jämföras med?
63. Kan en funktion ha flera *parametrar*?
64. Kan en funktion ha flera *returvärden*?
65. Vilket reserverat ord *måste* stå i kroppen till en funktion med returvärde?
66. Vad händer med kod i en funktion som man ev. skriver *efter* return-satsen?
67. Vad menas med *anropet* av en funktion?
68. Vad inleder det reserverade ordet `def` i Python, en funktions *definition* eller dess *anrop*?
69. Vad kallas parametrarna i *definitionen* av en funktion?
70. Vad kallas parametrarna i *anropet* av en funktion?
71. Skriv i Python *definitionen* av en funktion som kodar den matematiska funktionen $y = 2x + 5$.
72. Skriv i Python *anropet* av funktionen i fråga 71 för $x = 3$.
73. Vad blir anropet av följande funktion om man vill addera 5 och 6:

```
def sum(a, b) :  
    return a + b
```
74. Till vilket namn returneras en funktions returvärde inuti funktionens definition?
75. Utförs koden som man skriver i en funktions definition, vid exekvering, om funktionen inte anropas?

1.8 Kryptering, kursboken sid 77-85 & 142-146

76. Vad är ASCII-koder?
77. Varför måste varje tecken ha en kod?
78. Vilken funktion i Python ger ASCII-koden till ett tecken?
79. Vilken funktion i Python ger tecknet till en ASCII-kod?
80. Hur kan man i Python ta ut ett tecken ur en sträng?
81. Vilket tecken tar ut `text[0]` ur strängen `text = 'Gustav'`?
82. Vad returnerar `len(text)` om `text` är definierad som i fråga 81?
83. Returnerar `ord('a')` ett tal eller ett tecken?
84. Returnerar `chr(97)` ett tal eller ett tecken?
85. Vilket tecken blir `chr(ord('a'))`?
86. Vilket tecken blir `chr(ord('a')+1)`?
87. Beskriv krypteringsalgoritmen som används i funktionen `EncryptFkt` på sid 79?
88. Beskriv återställningsalgoritmen som används i programmet `EncryptText` på sid 78?
89. Vad innebär *punktnotation* i programmering?
90. Varför måste i funktionen `EncryptFkt` (sid 79) variabeln `newText` tilldelas den tomma strängen?
91. Vad returnerar `len('')`?
92. Vad gör pythonkoden `ch = chr(ord(ch) + k)`?

1.9 Primtal, kursboken sid 86-97 & 147-150

93. Definiera primtal. Vilket ord i definitionen av primtal är viktigast (avgörande)?
94. Vad säger Aritmetikens fundamentalsats?
95. Är 1 ett primtal? Motivera!
96. Varför sätts i programmet `PrimalsTest` på sid 87, rad 6 den logiska variabeln `prim` till `True`? Skulle programmet även fungera med `False`?
97. Varför kallas `and` i programmet `PrimalsTest` (rad 9) för *logisk operator*? Vad är en *logisk operator*?
98. Hur fungerar `and` i while-satsens villkor av programmet `PrimalsTest` (rad 9)?
99. Varför försvinner `if-else`-satsen i raderna 16-19 av programmet `PrimalsTest` när det görs om till funktionen `primtest()` på sid 90?
100. Innehåller filen `PrimFkt.py` (sid 90) definitionen eller anropet av funktionen `primtest()`?
101. Innehåller filen `PrimTest.py` (sid 90) definitionen eller anropet av funktionen `primtest()`?
102. Varför kan anropet av funktionen `primtest()` skrivas som villkor i `if-else`-satsen på rad 9 av programmet `PrimTest` på sid 90?
103. Om man exekverar filen `PrimFkt.py` (sid 90) händer ingenting. Varför?
104. Var på din dator måste filen `PrimFkt.py` finnas för att koden i filen `PrimTest.py` (sid 90) ska kunna exekveras?

105. Innehåller filen `PrimFaktorer.py` (sid 95) definitionen eller anropet av funktionen `faktorisera()`?
106. Beskriv algoritmen *Primtalsfaktorisering* (sid 95) med egna ord. Varför är den rekursiv?
107. Varför är funktionen `faktorisera()` rekursiv?

1.10 Rekursion, sid 98-102 & 151-154

108. Beskriv *Fibonaccis rekursionsformel* (sid 99) med egna ord. Hur läses $F(n)$?
109. Beräkna med Fibonaccis rekursionsformel $F(4)$.
110. Hur många kaninpar kommer det enligt rekursionsformeln att finnas efter 4 månader? Överensstämmer det med tabellen på sid 98?
111. Varför är formeln rekursiv?
112. Vad är nackdelen med Fibonaccis rekursionsformel?
113. Innehåller filen `Fibonacci.py` (sid 100) definitionen eller anropet av funktionen `fib()`?
114. Varför är funktionen `fib()` rekursiv?
115. Vilket variabelnamn har den formella parametern i funktionen `fib()`?
116. Vilket variabelnamn har den aktuella parametern i `fib()`?
117. Vad är den praktiska nackdelen med det rekursiva programmet `Fibonacci` (sid 100) ?
118. Kan man koda Fibonacci även med en iterativ algoritm?