

T02.01 Textformater og HexEditor

Spørsmål 1

Start med å unzippe filen og legg den i en katalog som du kaller TK0v02.

Filen heter ANSI.txt.

Windows-brukere:

Åpne filen ANSI.txt i Notisblokk ("Notepad"), og velg å lagre den i de tre andre mulige formatene (Dropdownmeny i Save As vinduet **Encoding**):

- * Unicode
- * Unicode big endian
- * UTF-8

Sørg for at du gir disse navn som gjør at du husker lagringsformatet.

Når du har gjort dette høyreklikker du og sjekker egenskaper ("Properties") for hver av filene.

Sett sammen Format og filstørrelse!

OSX-brukere:

Hold nede Ctrl-tasten og klikk på filene etter tur i Finder.

Velg "Get Info" fra menyen som kommer opp. Parr filkoding med filstørrelse.

TENK IGENNOM: Hvorfor leder de ulike kodingstypene til filer med ulike størrelse.

Spørsmål 2

Nå skal du åpne filen som er lagret i det Windows Notepad kaller ANSI-format i Hexeditoren din.

Hva er den **hexadesimale** og **desimale** ("titalls") koden for stor Å i Windows ANSI-format (Windows-1252)?

(Hint: Posisjon 0x**4F** i filen ANSI.txt...)

Spørsmål 3

Åpne nå filen med **Unicode Big Endian** koding av teksten, og klikk på det området i bildet under som inneholder koder som IKKE befant seg i den opprinnelige teksten (, og vanligvis heller ikke vil fremvises som text).

Framgangsmåte:

Sammenlign med innholdet i den opprinnelige filen. Ta hensyn til at dette formatet (Unicode kodet som UTF-16) benytter 16 bit pr tegn, mens den opprinnelige ("ANSI") bruker en byte. Finn hvilke byte som har blitt føyd til av Notepad når den lagret filen som "Unicode Big Endian".

- Du bør også sammenligne med filen som er lagret som "bare Unicode". Hva er forskjellen?

- Du bør også prøve å *fjerne* disse ekstra tegnene i Hexeditoren, lagre filen, og se hvordan den da blir seende ut i Notepad.)

Spørsmål 4

Åpne så filen som er kodet i UTF-8 i Hexeditoren din.

Hva er den *binære* koden for "Å" (stor å) i UTF-8?

Oppgi svaret **hexadesimalt**.

Spørsmål 5

Stor A er i ASCII 65 desimalt..

For å bytte ut Å'en i forrige oppgave med AA, så må du skrive inn de hexadesimale sifrene på posisjonen der koden/byten for Å'en var passert

Spørsmål 6

Åpne så en (kompilert) Java klasse-fil i Hexeditoren.

I likhet med mange andre filformater starter den med noen byte som sier hvilket format det er. Dette omtales vanligvis som et Magic Number.

Java klassefiler starter med den hexadesimale koden

Spørsmål 7

Hva er magic number for en ZIP-fil?

Spørsmål 8

Inspiser det binære innholdet i en moderne Powerpoint-fil (type **.pptx**).

Hva slags magic number har denne?

T02.03 Komprimering (ordboks--)

Spørsmål 1

En tekstfil inneholder 1200 forskjellige ord kodet i Unicode med UTF-16. Hvor mange bits trengs for å identifisere hvert enkelt ord dersom vi skal bruke ordbok-koding (dictionary encoding) til å komprimere filen?

(Hint: Hvor mange ganger må du gange/multiplisere 2 med seg selv for å få et tall som er større enn 1200?)

Spørsmål 2

Dersom meldingen:

xyx yyx xxy xxy

ble komprimert med LZW og bruker en start-ordbok med x, y og mellomrom som første, andre og tredje ord, hvordan ser den endelige ordboken ut?

Spørsmål 3

Dersom vi bruker ordbok-komprimering, og starter med en ordbok som inneholder x (=1), y (=2) og mellomrom (=3) (som bekrrevet i dagens forelesning)så vil den komprimerte versjonen av meldingen:

"xyx yyx xxy xxy yyx " bli " ???????? "

Spørsmål 4

Gitt at X er kodet som 0, Y er kodet som 1 og _ er kodet som 2.

Den komprimerte meldingen "110120023010245343" svarer da til " ????? "