iterate

Hva alle utviklere må vite om tegnsettenkoding

Stein Magnus Jodal

JavaZone X, 8. september 2011

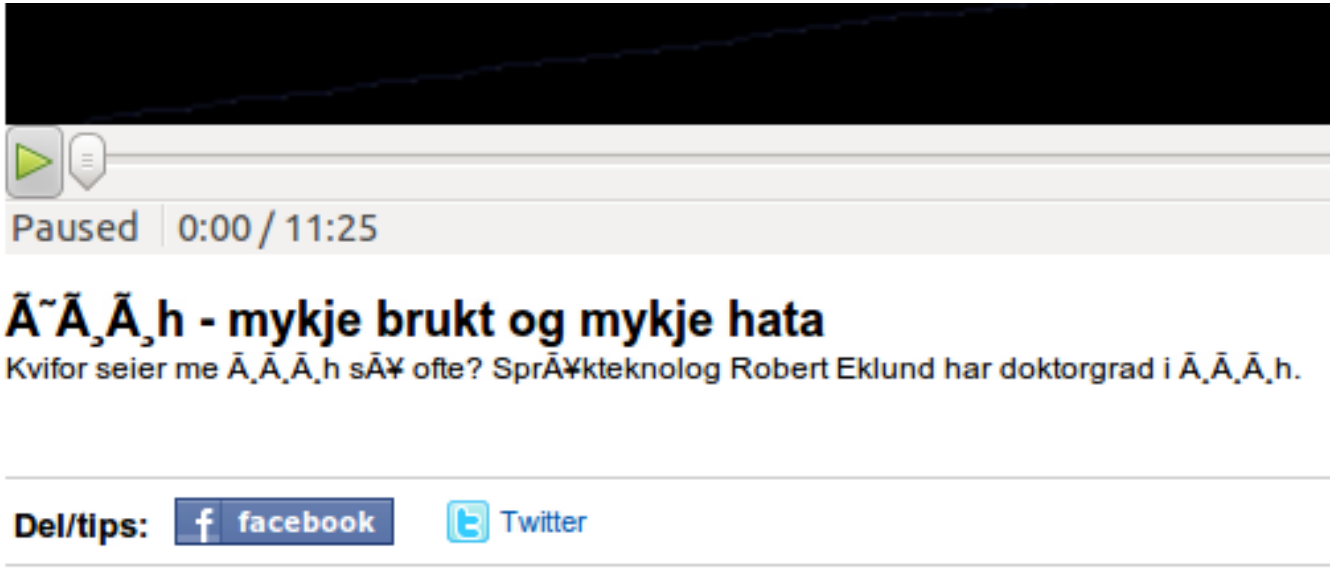
Hvorfor må jeg vite noe om tegnsettenkoding?

Fordi

- du bruker strenger i koden din
- programmet ditt snakker med omverdenen

Hvorfor må jeg vite noe om tegnsettenkoding?

NRK klarer det ikke...



Skjermdump fra nrk.no

Hvorfor må jeg vite noe om tegnsettenkoding?

Spreke UiO-studenter får det ikke til...

Skjermdump fra osi.uio.no

Hvorfor må jeg vite noe om tegnsettenkoding?

Selv Nordea klarer det ikke...

Annualisert avkastning (%)
1 Ã¥r
2 Ã¥r
3 Ã¥r
5 Ã¥r
10 Ã¥r

Skjermdump fra nordea.no

Tegnsettenkoding

er en mapping mellom tegn og noe en maskin forstår

Bilder av drinksmachine og G-Kenny på Flickr

1960-tallet: ASCII

- 33 kontrolltegn og 97 grafiske tegn, inkludert det engelske alfabetet og tall
- 7 bits: 0-127 desimalt

NULL	\n	0	1	A	B	C	a	b	c	~
0	10	48	49	65	66	67	97	98	99	126

1984: MacRoman

- ASCII pluss 128 tegn, bl.a. fra vest-europeiske språk
- 8 bits: 0-255 desimalt
- Standard på Mac frem til OS X

Æ	Ø	Å	æ	ø	å	Ö	ß
174	175	129	190	191	140	133	167

1985-1992: Latin-1 / ISO-8859-1

- ASCII pluss 128 tegn, bl.a. fra vest-europeiske språk
- 8 bits: 0-255 desimalt

Æ	Ø	Å	æ	ø	å	Ö	ß
198	216	197	230	248	229	214	223

1990-tallet: Windows-1252

- ASCII pluss mesteparten av Latin-1, men med noen ikkesynlige tegn byttet ut med en dæsj Latin-15
- 8 bits: 0-255 desimalt
- Ofte feilaktig merket som Latin-1

Æ	Ø	Å	æ	ø	å	Ö	ß	€
198	216	197	230	248	229	214	223	128

1991: Unicode

- Ikke en enkoding, men et tegnsett
- Også kjent som «Universal Character Set» (UCS) og ISO-10646
- Under kontinuerlig utvidelse siden 1991
- Definerer nå oppunder 100.000 tegn
- Definerer i tillegg til tegnsettet også noen enkodinger slik som UCS-2, UCS-4 og UTF-16

1993: UTF-8

- ASCII-tegnene er på plass 0-127
- Inkluderer alle tegn i Unicode-settet
- Hvert tegn er enkodet som 1-4 bytes
- Vest-europeiske tegn tar som regel 1-2 bytes

A	B	C	Æ	Ø	Å
65	66	67	(195, 134)	(195, 152)	(195, 133)

Hva har Nordea gjort galt?

Annualisert avkastning (%)
1 Ã¥r
2 Ã¥r
3 Ã¥r
5 Ã¥r
10 Ã¥r

- Vi forventet ett tegn «å», men fikk to tegn «Ã¥»
- Med andre ord: ett tegn på to bytes ble vist som to adskilte tegn

Hva har Nordea gjort galt?

Annualisert avkastning (%)
1 Ã¥r
2 Ã¥r
3 Ã¥r
5 Ã¥r
10 Ã¥r

Tese

Teksten er UTF-8-enkodet, men ble tolket som Latin-1

Hva har Nordea gjort galt?

Annualisert avkastning (%)
1 Ã¥r
2 Ã¥r
3 Ã¥r
5 Ã¥r
10 Ã¥r

Bevis

- «å» UTF-8-enkodet blir to bytes: (195, 165) desimalt
- I Latin-1 er 195 «Ã» og 165 er «¥» □

Hvor ble feilen gjort?

Ikke lett å si...
det du ser på en nettside kan påvirkes av

- dataene i databasen
- tilkoblingen til databasen
- enkoding av filer på serveren
- header i HTTP-responsen
- header i HTML-dokumentet
- standardenkoding i nettleseren

Hva skal jeg gjøre?

Bruk én standard tekstrepresentasjon internt i programmet ditt

- Enten, velg en enkoding som spiser det meste av tegn
  - Les: UTF-8

Hva skal jeg gjøre?

Bruk én standard tekstrepresentasjon internt i programmet ditt

- Enten, velg en enkoding som spiser det meste av tegn
  - Les: UTF-8
- Eller, bruk programmeringsspråkets Unicode-kapable strengtype:
  - `java.lang.String` i Java, Scala, etc.
  - `unicode` i Python 2.x, `string` i Python 3.x
  - `string` i Ruby 1.9 vet om sin egen enkoding

Hva skal jeg gjøre?

Vokt grensene til programmet ditt

- På vei inn
  - Konverter alt til internenkodingen
  - Leser du en fil i en annen enkoding? Konverter den!

Hva skal jeg gjøre?

Vokt grensene til programmet ditt

- På vei inn
  - Konverter alt til internenkodingen
  - Leser du en fil i en annen enkoding? Konverter den!
- På vei ut
  - Eventuelt konverter til forespurt enkoding
  - Vær alltid tydelig på hva du gir fra deg

Hva skal jeg gjøre?

Ved tvil

- Bruk UTF-8

Hva skal jeg gjøre?

Oppsummert

- Kun én intern enkoding
- Vokt grensene
- Ved tvil: Bruk UTF-8

iterate

(Hva)

alle utviklere må vite om tegnsettenkoding

@jodal // www.jodal.no