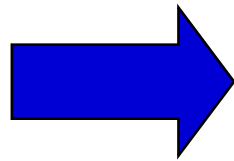


INF1040 – Digital representasjon

Oppsummering



100101010000111

Ragnhild Kobro Runde, Fritz Albrechtsen

Eksamen – I

- Fredag 7. desember 2007.**
- 09.00 – 12.00**
 - **Møt senest 08.45!**
- Ta med legitimasjon!**

- Ingen hjelpemidler tillatt, heller ikke kalkulator.**

Eksamen – II

- ❑ Les gjennom hele oppgaven før du begynner å løse den.
 - Få en oversikt over hva som skal gjøres.
- ❑ Kontroller at oppgavesettet er komplett før du begynner å besvare det.
 - Antall sider står på forsiden.
- ❑ Dersom du savner opplysninger i oppgaven, kan du legge dine egne forutsetninger til grunn og gjøre rimelige antagelser, så lenge de ikke bryter med oppgavens ”ånd”. Gjør i så fall rede for forutsetningene og antagelsene du gjør. Dette gjelder også flervalgsoppgavene.
 - Vi går runde i eksamenslokalet ca 9.30 – spør oss!

Eksamen – III

- ❑ Dine svar skal skrives på disse oppgavearkene, ikke på separate ark. Dette gjelder både spørsmål med avkryssnings svar og spørsmål hvor du bes om å regne ut noe. I de oppgavene hvor det skal regnes ut noe, anbefaler vi at du først skriver en kladd på et eget ark før du fører svaret inn på rett plass i oppgavearkene.
 - Tilleggs-ark kan selvfølgelig brukes hvis det er nødvendig (vis tydelig sammenhengen!)

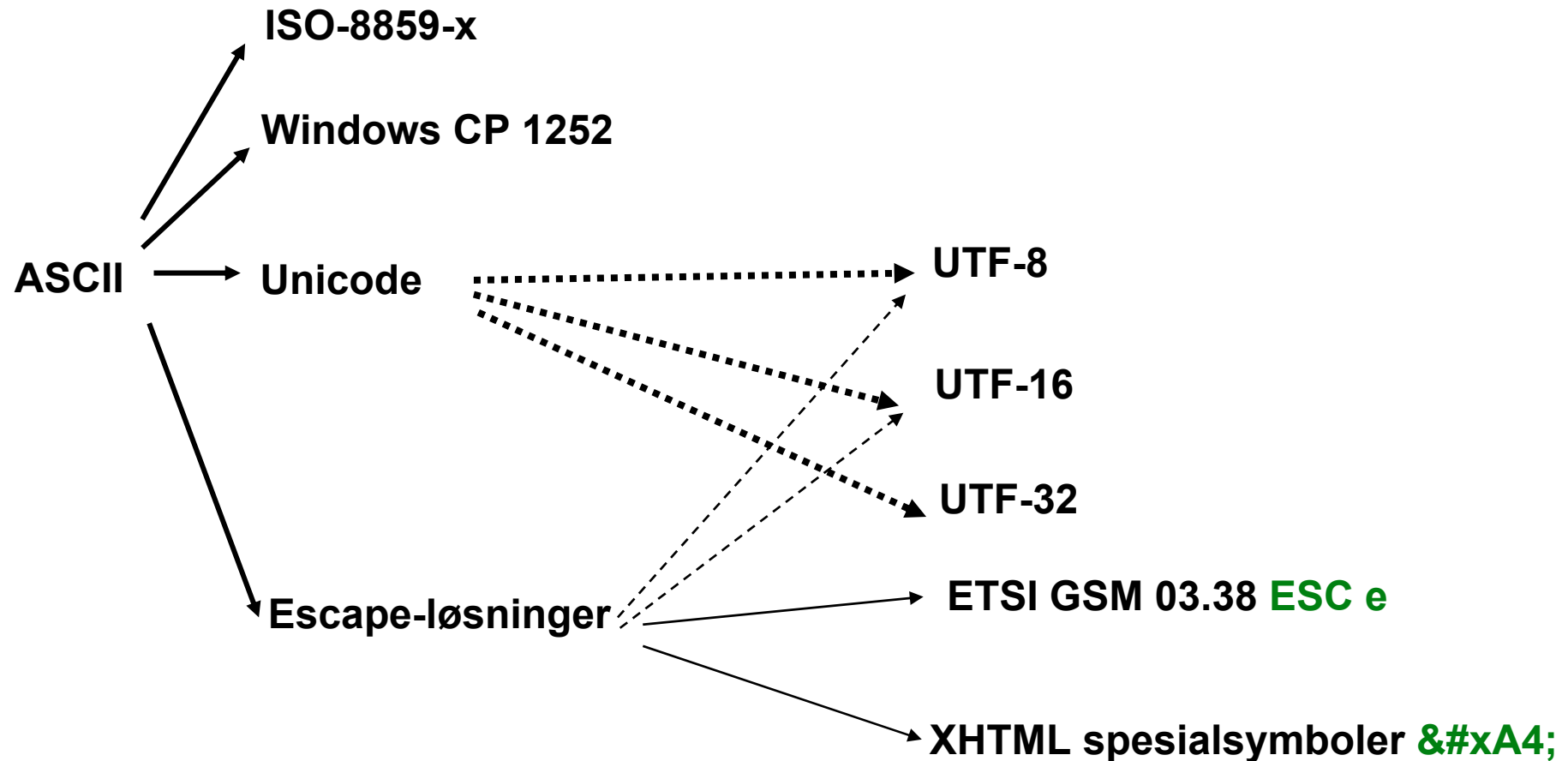
Eksamen – IV

- ❑ 30 av spørsmålene er flervalgsoppgaver med fem alternativer der bare ett svar er riktig.
- ❑ På disse oppgavene får du 4 poeng for riktig svar, -1 for feil, og 0 dersom du ikke svarer. Den som svarer i hytt og vær vil komme ut med 0 poeng her!
- ❑ Finner du det rette svaret, men ”garderer” med ett eller flere ekstra kryss, så trekkes du ett poeng for hvert feil kryss.
- ❑ Hvis du har satt et kryss i en avkrysningsboks og etterpå finner ut at du *ikke* ønsket et kryss der, kan du skrive ”FJERN” like til venstre for avkrysningsboksen.

Eksamen – V

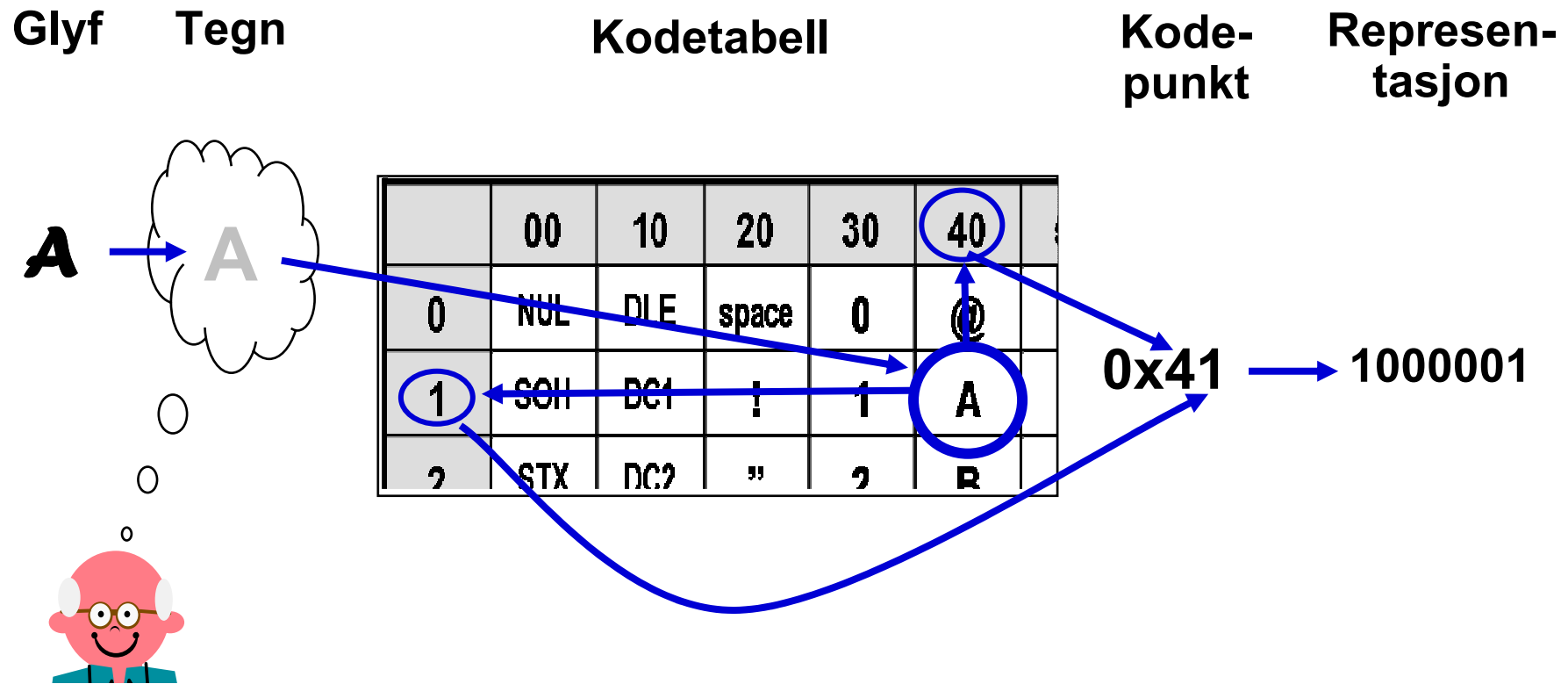
- ❑ I de n siste oppgavene skal du finne svarene selv. Du skal beskrive hvordan du tenker – ikke bare skrive ned et svar. Oppgavene kan ha flere delspørsmål. Pass på at du svarer på alle disse! Maksimalt antall poeng for hver oppgave er henholdsvis...
- ❑ Disponer tiden slik at du rekker å svare på mest mulig! Du kan for eksempel bruke ca 2 minutter per oppgave på de 30 flervalsoppgavene (totalt en time), og deretter...

Tegnsett: ASCII og etterfølgerne



Fra glyf til representasjon

Må ha standarder for: tegn ↔ kodepunkt ↔ representasjon



Fra kodepunkt til representasjon

- Vanligvis "rett fram etter nesa"

0x41 → 1000001

- men av og til unntak basert på "escape"-prinsippet ("det bitmønsteret som kommer nå, er ikke en vanlig representasjon – det skal spesialbehandles")

- Eksempel: Spesialtegn i XML

"Rett fram etter nesa"-bitmønsteret for &, dvs 00100110, representerer ikke &, men begynnelsen på et spesialsymbol: **&#UNICODE-kodepunkt;**

Da må også & være et spesialsymbol:

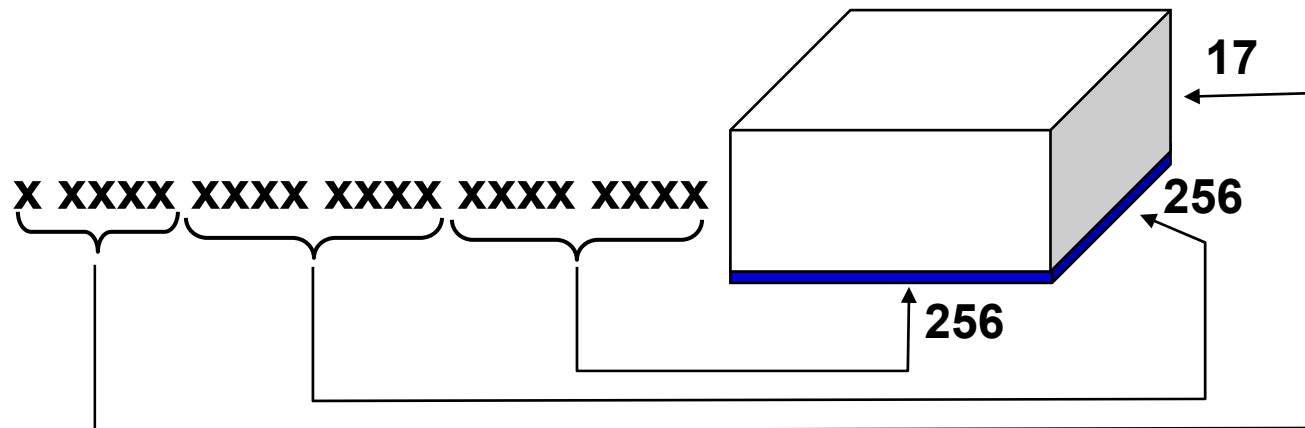
&

→ 00100110 00100011 01111000 00110010 00111010 00111011

Unicode og ISO 10646

- ❑ 21 bit, med mulighet for 1 114 112 tegn
- ❑ Tegnsettet er delt opp i 17 plan med max. $2^{16} = 65536$ tegn i hvert plan
- ❑ Plan 0: BMP – Basic Multilingual Plane U+0000 to U+FFFF
- ❑ Plan 1: SMP – Supplementary Multilingual Plane
– historiske språk (f. eks. egyptiske hieroglyfer), musikk
- ❑ Plan 2: SIP – Supplementary Ideographic Plane
– sjeldne kinesiske tegn
- ❑ Plan 14: SPP – Supplementary Special Purpose Plane – tag characters

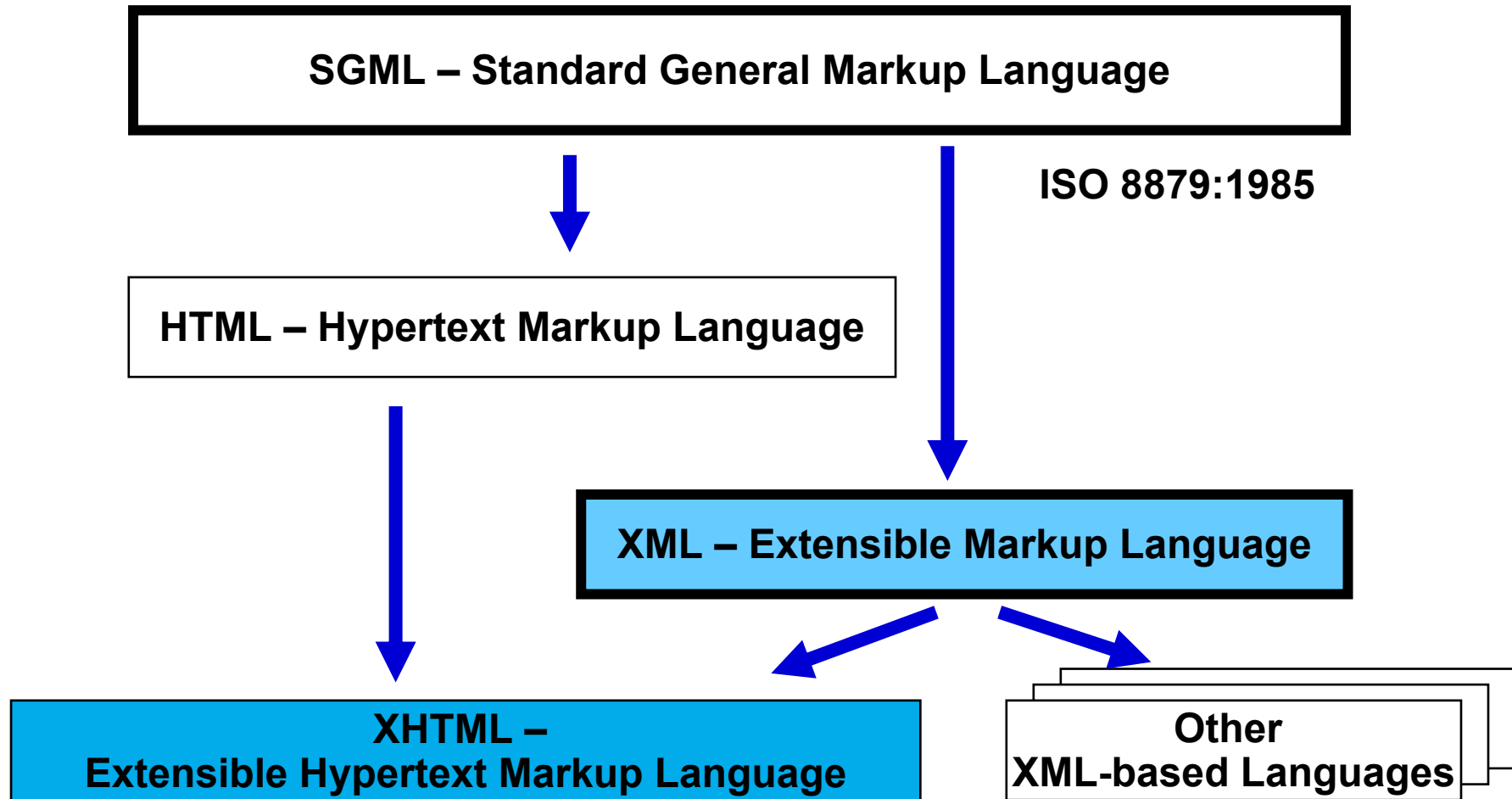
I Unicode skriver vi U+ istedenfor 0x



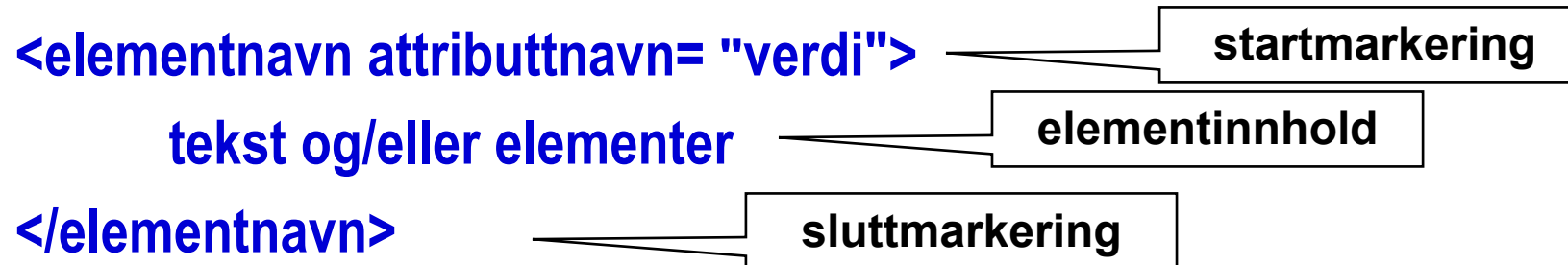
og så har vi alle Unicode-transformasjonene

- ❑ **Character encoding forms (fra kodepunkt til "kodingsenheter")**
 - **UTF-8 – variabel lengde fra 1 til 4 bytes**
0xxx xxxx er kompatibel med ASCII
 - **UTF-16 – variabel lengde fra 1 til 2 bytes**
1 byte for tegn i BMP, ellers surrogatpar
 - **UTF-32 – fast lengde 4 bytes**
- ❑ **Character encoding schemes (fra "kodingsenhet" til bytesekvens)**
 - UTF-8
 - UTF-16 med byte-order mark eller lignende
 - UTF-16BE
 - UTF-16LE
 - UTF-32 med byte-order mark eller lignende
 - UTF-32BE
 - UTF-32LE

SGML-familien



Oppbyggingen av et XML-element



- ❑ Et element må ha både en *startmarkering* og en *sluttmarkering*
`<markering>elementinnhold</markering>`
- ❑ Tomme elementer kan ha en kombinert kombinert start- og sluttmarkering `<markering />`

Krav til et velformet XML-dokument:

- 1. Én rot***
- 2. Perfekt nøstede elementer***

Stiler og stilark

- ❑ En **stil** er en regel ("rule") som gir nettleseren instruksjoner og hint om hvordan nettsiden skal presenteres
- ❑ Et **stilark** ("style sheet") inneholder et antall slike regler
- ❑ Stiler kan angis
 - "Inline": Som verdi til attributtet style i en markering
`<h1 style="font-weight:bold; color:blue; ">`
 - "Embedded": I et internt stilark i nettsidehodet
`<head>`
`<title>Beskrivelse INF1040</title>`
`<style type = "text/css">`
`h1 { font-weight:bold; color:blue; }`
`</style>`
`</head>`
 - "Linked": I et eksternt stilark (se neste lysark)

Linket stilark – med henvisning

på filen `example.css`:

```
/* Eksempel på linket stilark */  
h1 {  
  font-weight:bold;  
  color:blue;  
}
```

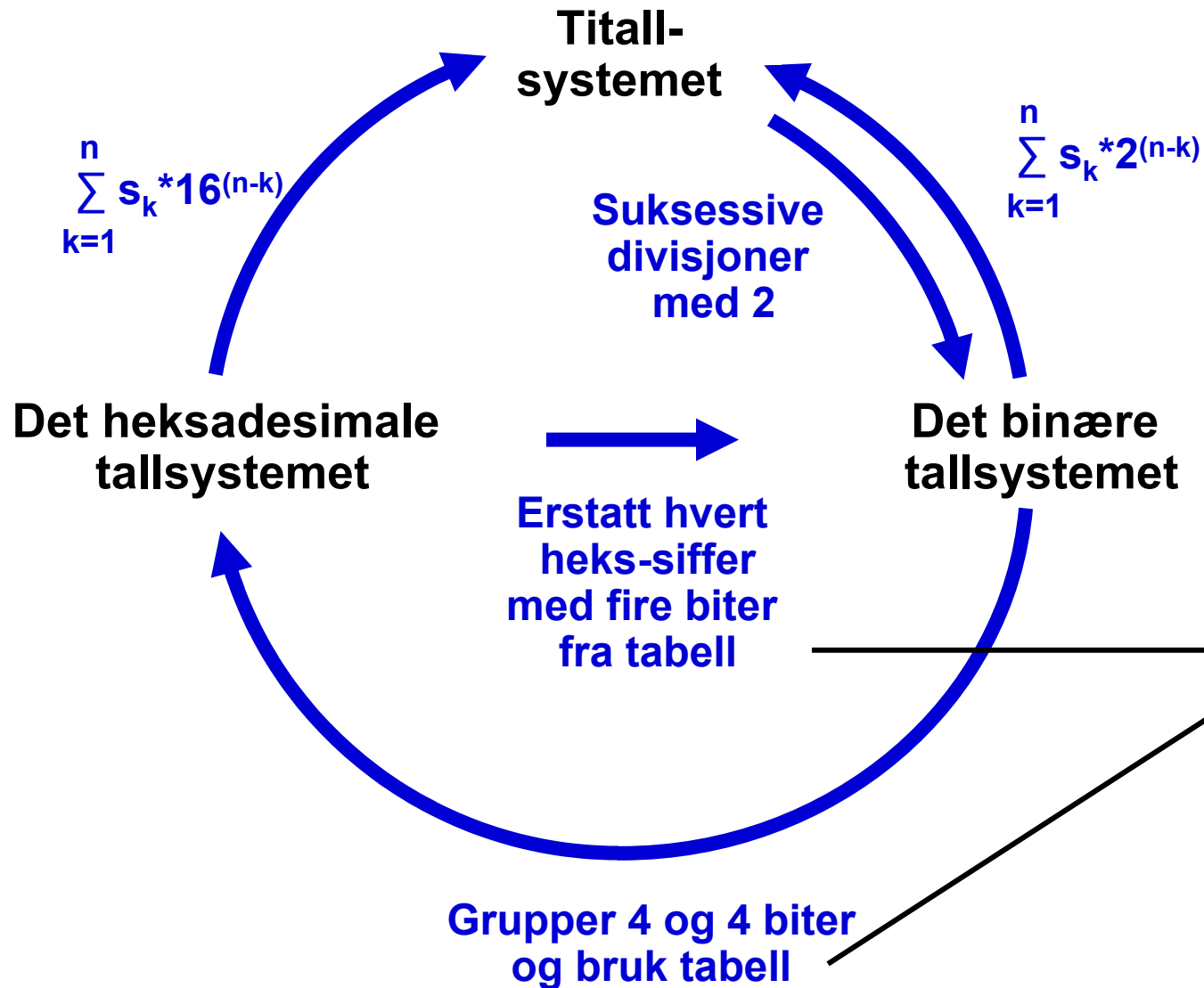
```
<html>  
<head>  
<title>Beskrivelse INF1040</title>  
<link rel="stylesheet" type="text/css" href="example.css"/>  
</head>  
<body>  
<h1> Innhold INF1040</h1>  
<p>  
...  
</p>  
</body>  
</html>
```

*En fordel med det linkede stilarket er at det kan henvises til fra flere nettsider.
Dette kan brukes til å gi et helhetlig preg på nettstedet!*

Tall kan representeres på mange måter

- ❑ Tall som tekst (ASCII-tegn representasjon)
- ❑ Heltall
 - Positive
 - Negative
(flere mulige representasjoner, toerkomplement er vanlig)
- ❑ Tall avbildet på heltall
(for eksempel bruk av bias)
- ❑ Flyttall

Konvertering mellom tallsystemer



binært	heks
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

Toerkomplementet for negative heltall

- Det binære toerkomplementet er lett å beregne:

Ta et binært tall

Erstatt alle 0 med 1, alle 1 med 0

(legg merke til at vi må vite antall biter for tallrepresentasjonen)

Legg til 1

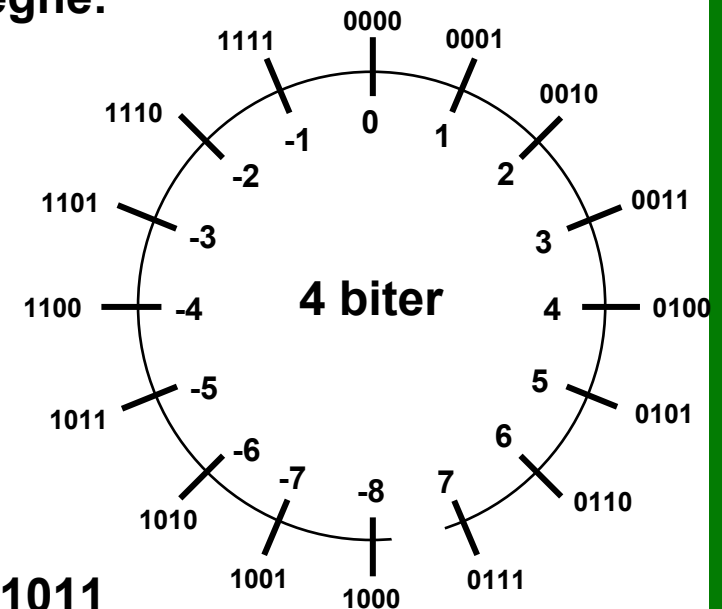
- Eksempel:

$21_{10} = 0001\ 0101$ (forutsetter 8 biter)

toerkomplementet er $1110\ 1010 + 1 = 1110\ 1011$

Regne ut $53 + (-21)$:

$$\begin{array}{r} 0011\ 0101 \\ +\ 1110\ 1011 \\ \hline = \cancel{00}10\ 0000 = 32_{10} \end{array}$$



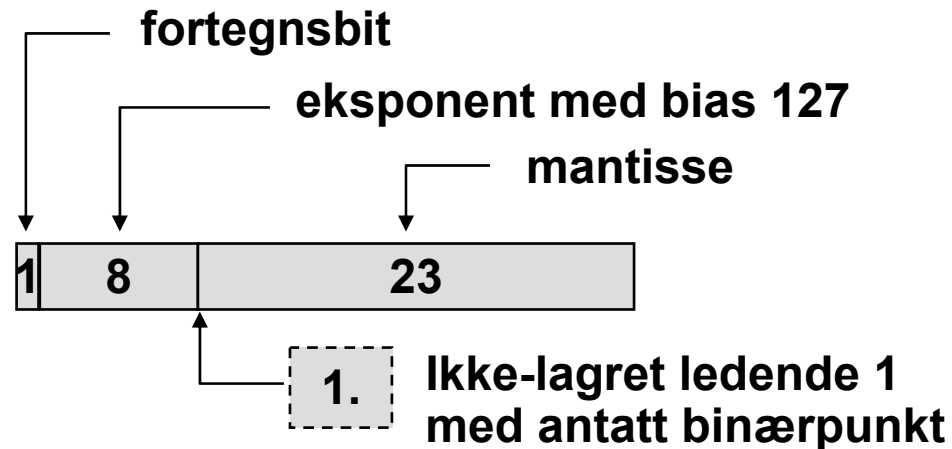
*Populært kalt
"klokke-
aritmetikk"*

Tall avbildet på heltall – tall med "bias"

- ❑ Vi skal representere tallene $[-128, \dots, 127]$ (8 biters tallrepresentasjon)
- ❑ Vi legger en bias 128 til alle tallene, slik at vi istedenfor kan representere tallene $[0, \dots, 255]$ – og det er jo helt kurant
- ❑ Ved addisjon kommer bias med to ganger, så vi må trekke den fra igjen
- ❑ **Eksempel (forutsetter 8 biters tallrepresentasjon og derfor bias 128):**
Vi skal addere 53 og -21 .
 $53 \text{ bias } 128 = 181 = 1011\ 0101_2$
 $-21 \text{ bias } 128 = 107 = 0110\ 1011_2$
 $181 + 107 - 128 = 10110101_2 + 0110\ 1011_2 - 1000\ 0000_2$
 $= 1010\ 0000_2 = 10 * 16 = 160 = 32 \text{ bias } 128$
- ❑ Dette prinsippet brukes for eksponenten i flytende tall

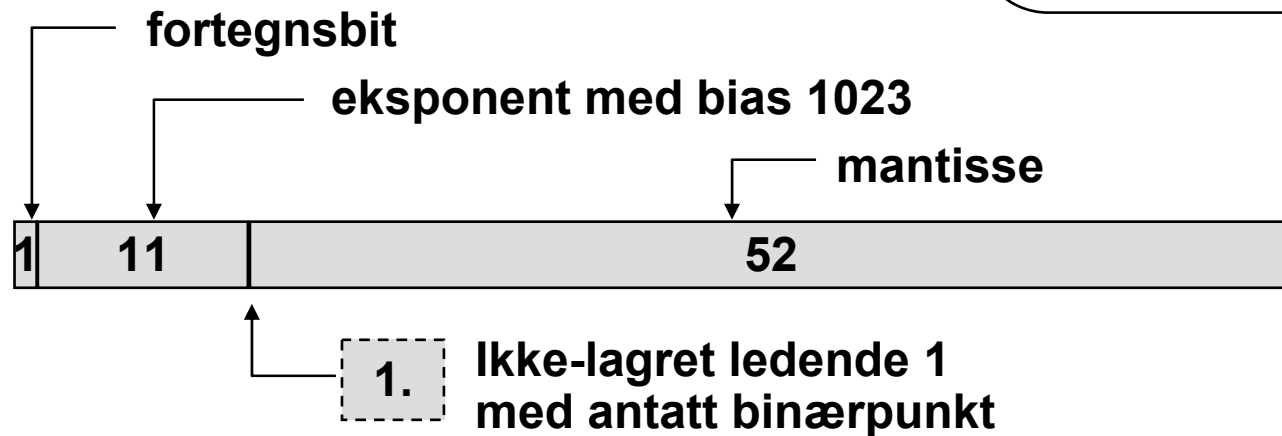
Binære flytende tall (IEEE 754)

Single precision



Vi legger en bias til eksponenten slik at representasjonen aldri er negativ!

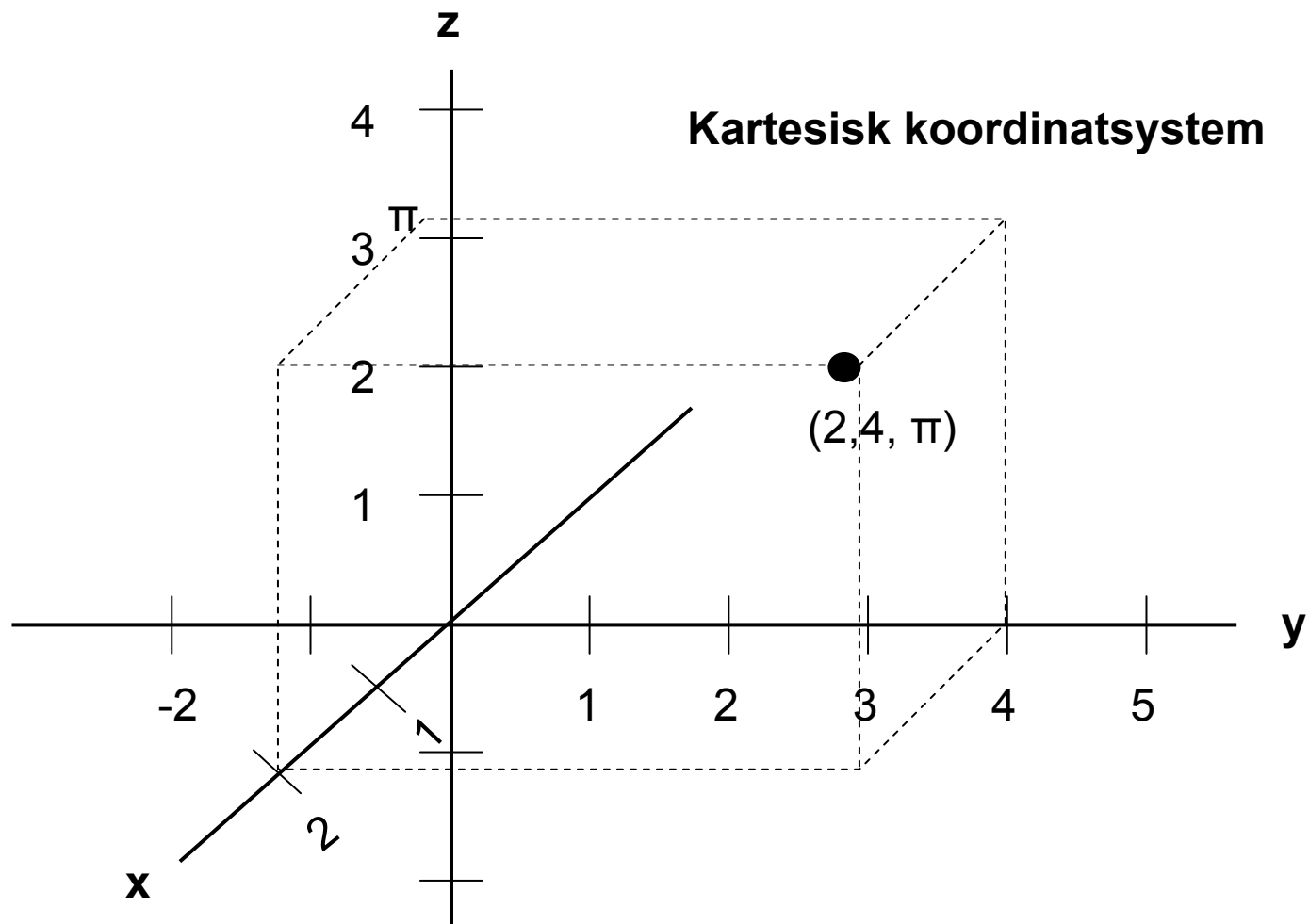
Double precision



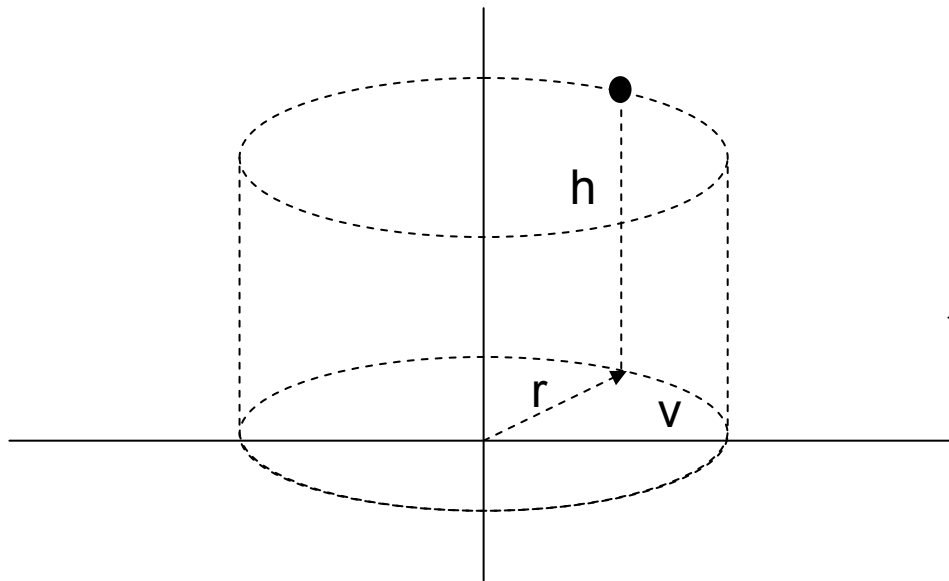
Mantissen er normalisert, slik at første bit alltid er 1. Derfor sparer vi plass ved ikke å lagre denne biten!

Punkter i tredimensjonalt rom

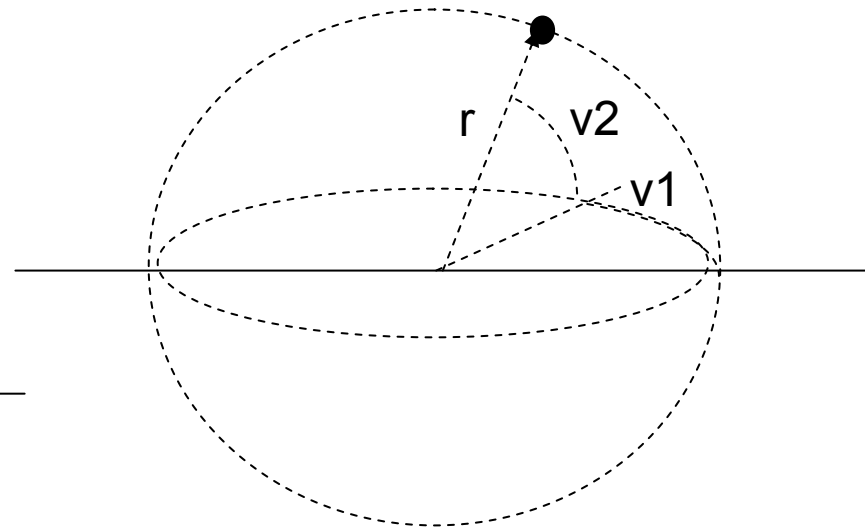
- I et tredimensjonalt rom trenger vi et *koordinattrippel*.
- **Eksempel: Punktet $(2, 4, \pi)$**



Alternativer til kartesiske koordinater – tre dimensjoner



Sylinderkoordinater



Sfæriske koordinater

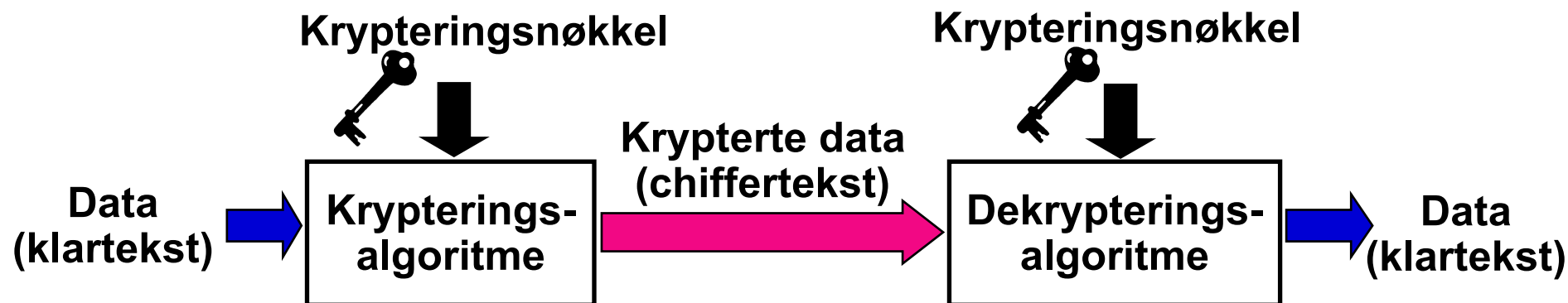
Representasjoner av geometri

- ❑ En uendelig mengde punkter med uendelig presis beliggenhet kan ikke representeres i en datamaskin
- ❑ To ulike løsninger;
 - "Vektorrepresentasjon":
Representerer noen viktige punkter, og avlede de øvrige punktene matematisk ved behov.
Egnet for "regulære" geometrier.
 - "Rasterrepresentasjon"
Bygge opp representasjonen av et endelig antall "punkter med utstrekning".
Gir vanligvis bare en tilnærmet korrekt geometri.

Sikring av data

- ❑ **Kritiske data må sikres mot lesing og endring av uvedkommende (kryptering) eller skjules (steganografi)**
- ❑ **Ved kryptering brukes oftest en kombinasjon av symmetriske og asymmetriske krypteringsteknikker**
- ❑ **Vurderinger av sikkerheten mot "knekking" av krypteringer er kun basert på antagelser og empiri, intet er bevist**
- ❑ **Steganografi brukes for å skjule en melding i et dekke**
- ❑ **Vannmerker brukes for å gi tilleggsopplysninger som ikke kan fjernes uten å ødelegge dekket**

Kryptering av data



□ Formål:

- Gjøre data som sendes eller lagres uleselige for uvedkommende.

□ utfordringer:

- Finne tilstrekkelig gode krypterings- og dekrypterings-algoritmer.
- Gjøre det praktisk umulig å finne krypteringsnøkkelen (“knekke koden”).
- Administrere krypteringsnøkler.

Krypteringsprinsipper

❑ Krypteringsnøkkel

- **Symmetrisk kryptering**
Samme nøkkel brukes for både kryptering og dekryptering
- **Asymmetrisk kryptering**
To sammenhengende nøkler,
den ene brukes for kryptering, den andre for dekryptering

❑ Håndtering av data

- **Stream-kryptering**
Krypterer bitene eller bytene etter hvert som de kommer
- **Blokk-kryptering**
Opererer på en blokk av biter – typisk fra 64 til 384 – ad gangen

❑ Algoritmer

- **Substitusjonsprinsippet**
Bytte ut biter eller bytes med andre biter og bytes
- **Transformasjonsprinsippet**
Endre rekkefølgen på biter eller bytes

Praktisk bruk av asymmetrisk kryptering

- ❑ Alice vil sende en hemmelig melding til Bob
 - Alice krypterer meldingen med Bobs offentlige nøkkel
 - Bob er den eneste som kan dekryptere meldingen, fordi bare han er i besittelse av den tilhørende private nøkkelen

- ❑ Alice vil sende en melding til Bob på en slik måte at Bob kan forsikre seg om at den virkelig er fra henne
 - Alice krypterer meldingen med sin egen private nøkkel
 - Bob ser at meldingen trolig kommer fra Alice, og dekrypterer med hennes offentlige nøkkel.
Går det bra, kan han gå ut fra at meldingen virkelig kommer fra Alice.