

INF105 - 03
Binærkoding av tall / tekst

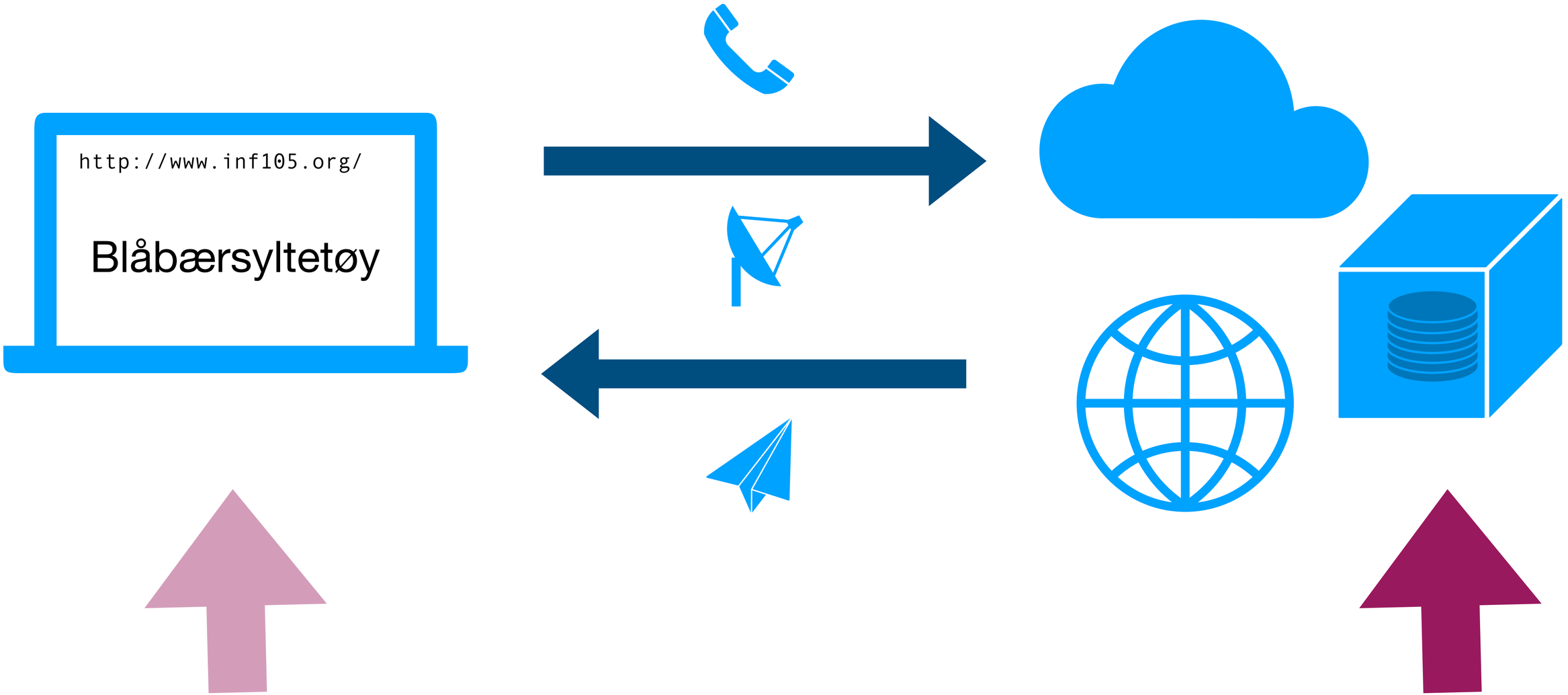
David Grellscheid

Q&A

- Må vi ha grunnleggende programmerings ferdigheter?
- Har du/dere en oversikt på timeplanen i en tabell?
- Finner man presentasjonene fra forelesning noe sted?



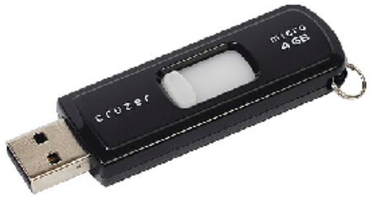
Rød tråd



Gruppetime 1



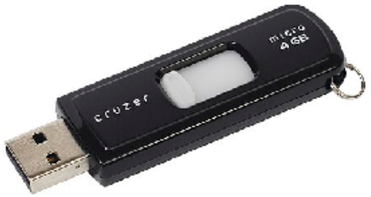






Gruppetime 1





alpine Linux



setup-alpine

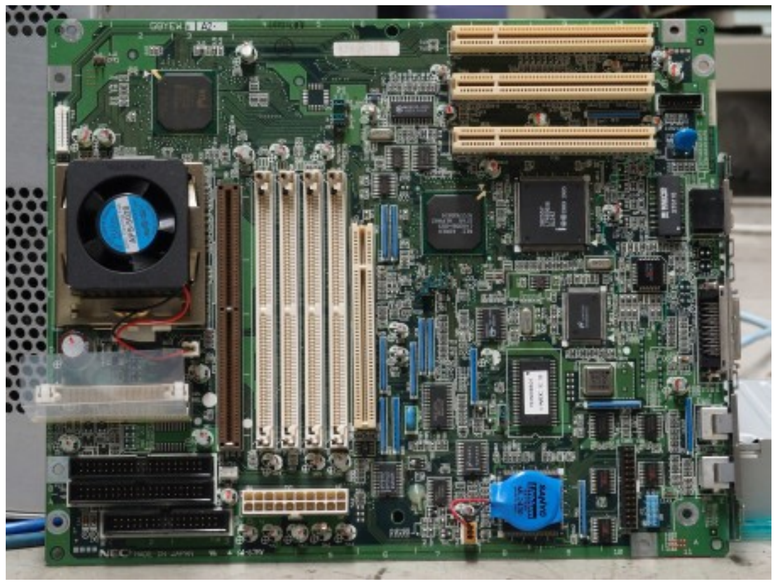
Gruppetime 1

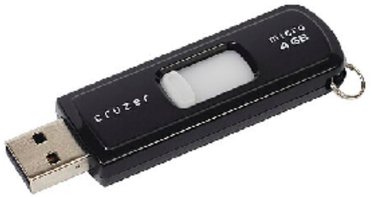






Gruppetime 1





alpine Linux

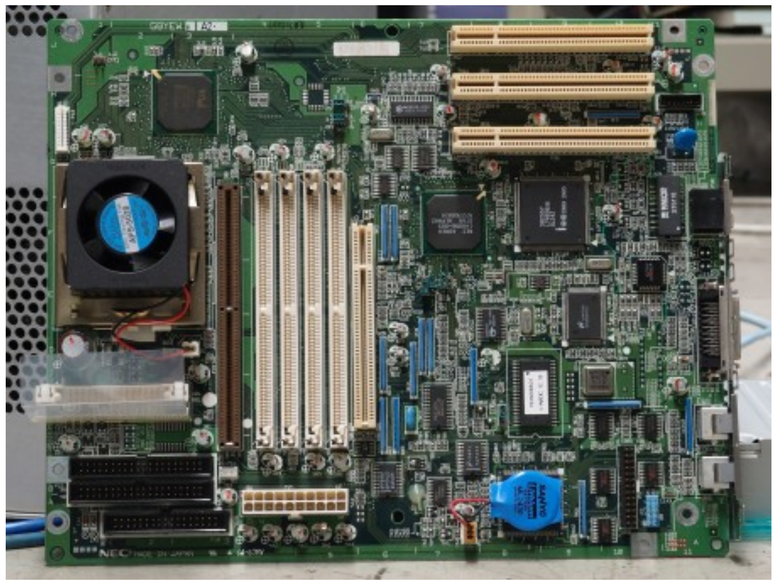


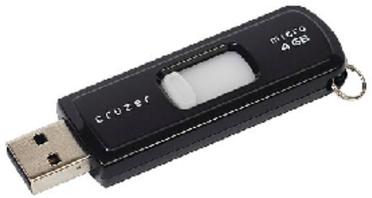
setup-alpine

Virtual machine



Gruppetime 1







setup-alpine

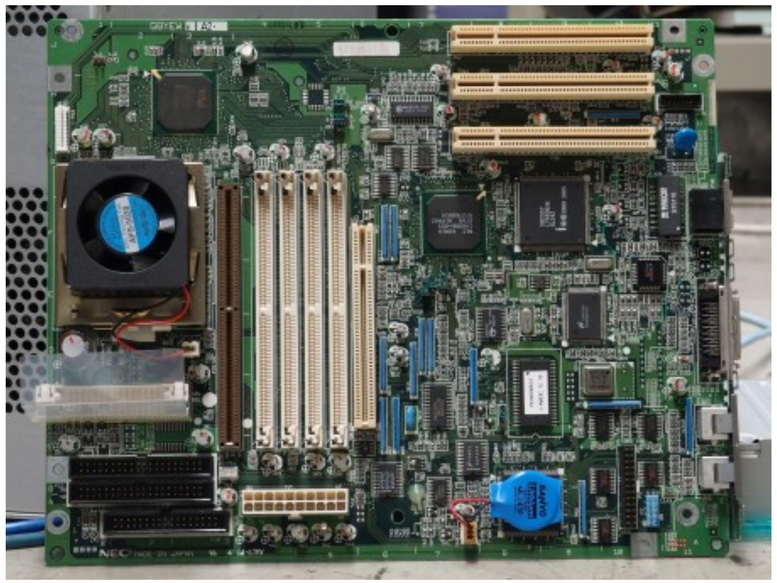
Virtual machine

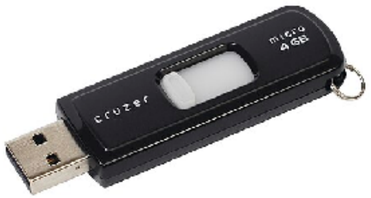


alpine-virt-3.24.1-aarch64.iso



Gruppetime 1







setup-alpine

Virtual machine

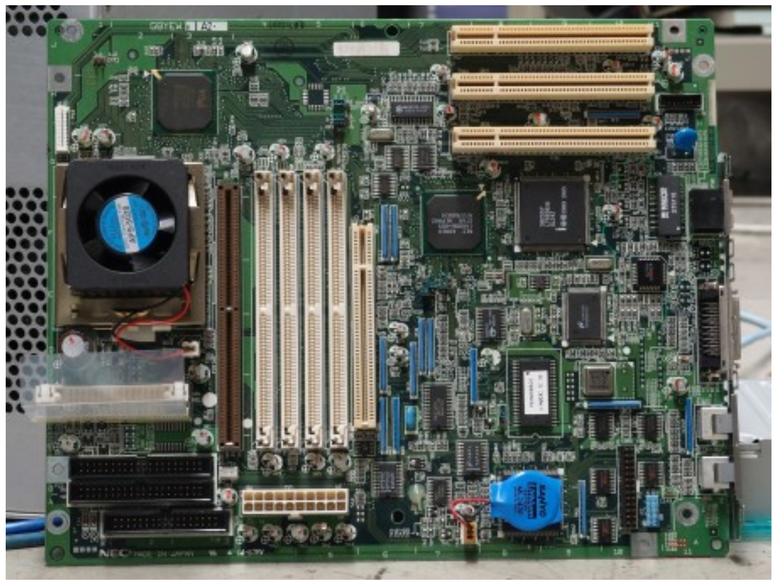
alpine-virt-3.24.1-aarch64.iso





setup-alpine

Gruppetime 1

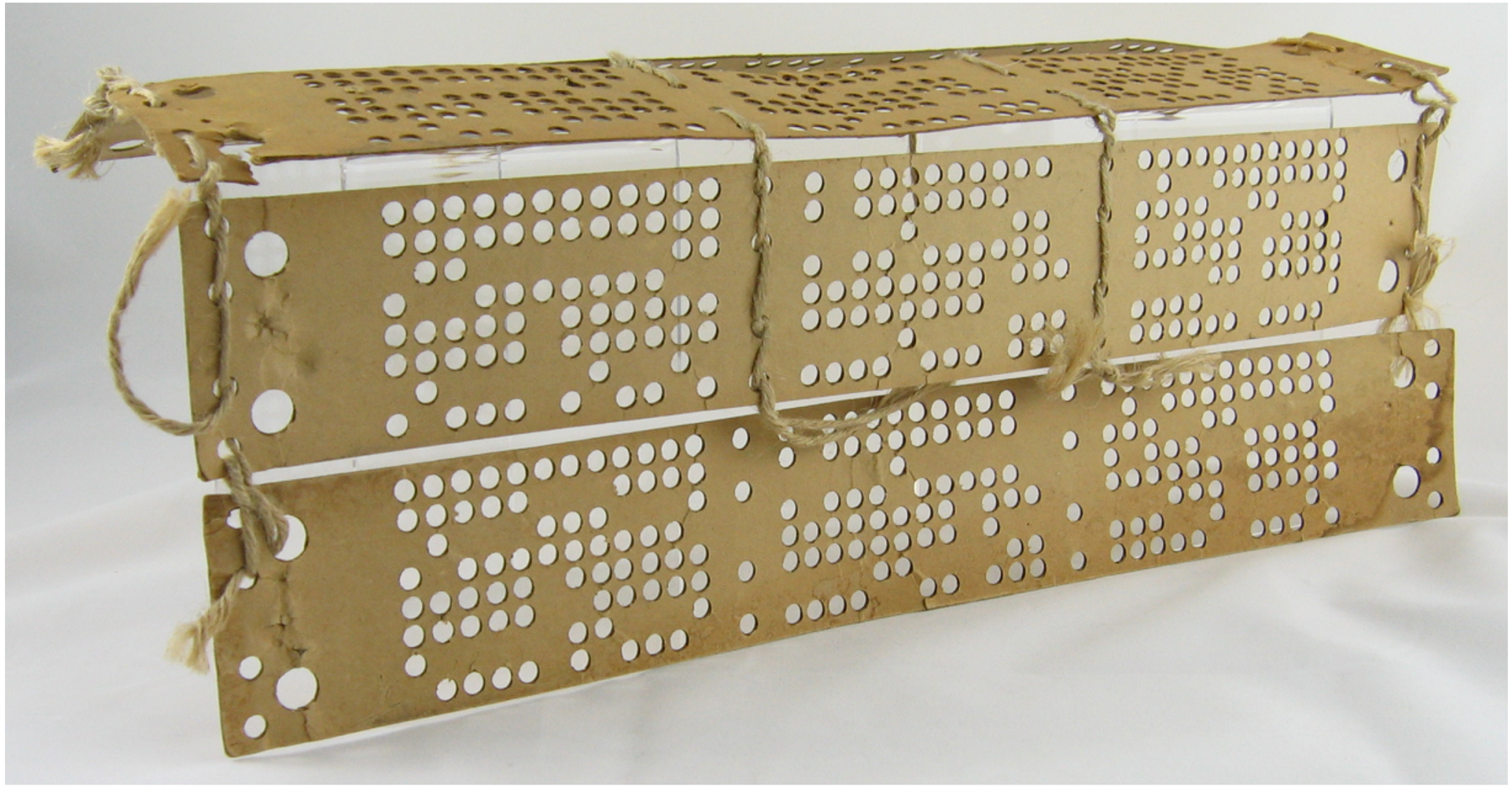




Virtual machine



Lagre data



https://obsoletemedia.org/

Desimaltall

- Desimal 749

  $7 \times 100 + 4 \times 10 + 9 \times 1$

7	4	9
100	10	1
10^2	10^1	10^0

Desimaltall

- Desimal 749

  $7 \times 100 + 4 \times 10 + 9 \times 1$

- Trenger 10 ulike symboler:
- 0 1 2 3 4 5 6 7 8 9
- ০ ১ ২ ৩ ৪ ৫ ৬ ৭ ৮ ৯
- ٠ ١ ٢ ٣ ٤ ٥ ٦ ٧ ٨ ٩

7	4	9
100	10	1
10^2	10^1	10^0

Binærtall

- Trenger bare 2 ulike symboler:
- 1 0
- hull / ingen hull
- strøm på / av

Binærtall

- Trenger bare 2 ulike symboler:
- 1 0
- hull / ingen hull
- strøm på / av

2^2	2^1	2^0

Binærtall

- Trenger bare 2 ulike symboler:
- 1 0
- hull / ingen hull
- strøm på / av

4	2	1
2^2	2^1	2^0

Binærtall

- Trenger bare 2 ulike symboler:
- 1 0
- hull / ingen hull
- strøm på / av

1	0	1
4	2	1
2^2	2^1	2^0

Binærtall

- Trenger bare 2 ulike symboler:
- 1 0
- hull / ingen hull
- strøm på / av
- **"én bit"**

1	0	1
4	2	1
2^2	2^1	2^0

0b101 = 1 × 4 + 0 × 2 + 1 × 1

Binærtall

- Trenger bare 2 ulike symboler:
- 1 0
- hull / ingen hull
- strøm på / av

1	0	1
4	2	1
2^2	2^1	2^0

0b101 = 1 × 4 + 0 × 2 + 1 × 1

Noen tall å huske

- potenser av 2 fra 0 til 12:
  1 2 4 8 16 32 64 128 256 512 1024 2048 4096
- 2^16 = 65536
- 2^32 = 4 milliard noe

Hvor mange tall?

- 3 desimalsifrer: 10^3 = 1000 tall: 000 – 999
- 3 binærsifrer: 2^3 = 8 tall: 0b000 – 0b111

Hvor mange tall?

- 3 desimalsifrer: 10^3 = 1000 tall: 000 – 999
- 3 binærsifrer: 2^3 = 8 tall: 0b000 – 0b111
- 8 binærsifrer: 2^8 = 256 tall: 0b0000 0000 – 0b1111 1111

én byte = 8 bit

Hex - Husk denne

- 8 bit i en byte kan deles opp i 2 "**nibbles**" av 4:
- 256 = 16 x 16
- Trenger 16 symboler: 0-9, A-F

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

Hex - Husk denne

- 8 bit i en byte kan deles opp i 2 "**nibbles**" av 4:
- 256 = 16 x 16
- Trenger 16 symboler: 0-9, A-F

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

0xA2 = 0b 1010 0010 = 128 + 32 + 2 = 162

Koding av positive heltall

Koding av positive heltall

- 1 byte / 8 bit: 0 – 255
- 2 byte / 16 bit: 0 – 65 535
- 4 byte / 32 bit: 0 – 4 294 967 296

"unsigned integer"

Overflow

- f.eks 8 bit unsigned
- 250 + 10 = 4 ? True
- 1111 1010
  + 0000 1010
  ==========
  (1) 0000 0100

Negative tall?

8 bit signed integer

- Bruk første bit til fortegn: 0 -> +   1 -> -
- 0111 1110 = 126
- 0111 1111 = 127
- 1000 0000 = -128 (har flere valg her, dette er Two's Complement, flest brukt nå)
- 1000 0001 = -127
- ...
- 1111 1110 = -2
- 1111 1111 = -1

desimaltall - i morgen

Tekst - ASCII

- trenger 2x 26 bokstaver, 10 sifrer, noen tegn (. , : ; ! ? + -)
- allerede mer enn 64 (2^6)
- ASCII bruker 7 bit 0000 0000 - 0111 1111 (bit 8 er alltid "0")
- 0x00 - 0x7F

Tekst - ASCII

https://en.wikipedia.org/wiki/ASCII

Tekst - iso8859

- Bruk de andre 128 posisjoner som er ikke med i ascii:
- 1000 0000 - 1111 1111
- 0x80 - 0xFF
- Bit 8 er "1"

Tekst - ISO8859-x

https://en.wikipedia.org/wiki/ISO/IEC_8859

Binary representation

1 byte = 8 bits = 2 hex digits.
Can have 256 possible values

binary	hex			
"0000 0000"	"00"			
"0000 0001"	"01"			
"0000 0010"	"02"			
...	...			
"0100 0001"	"41"			
...	...			
"1110 0110"	"e6"			
...	...			
"1111 1101"	"fd"			
"1111 1110"	"fe"			
"1111 1111"	"ff"			

Binary representation

Many possible interpretations of those bit patterns.
Here: unsigned or signed integers

binary	hex	uint8	int8	
"0000 0000"	"00"	0	0	
"0000 0001"	"01"	1	1	
"0000 0010"	"02"	2	2	
...	...	...	...	
"0100 0001"	"41"	65	65	
...	...	...	...	
"1110 0110"	"e6"	230	-26	
...	...	...	...	
"1111 1101"	"fd"	253	-3	
"1111 1110"	"fe"	254	-2	
"1111 1111"	"ff"	255	-1	

Binary representation

Many possible interpretations of those bit patterns.
Here: text

binary	hex	ascii	iso8859-1	iso8859-2	iso8859-5
"0000 0000"	"00"	*NUL*	*NUL*	*NUL*	*NUL*
"0000 0001"	"01"	*SOH*	*SOH*	*SOH*	*SOH*
"0000 0010"	"02"	*STX*	*STX*	*STX*	*STX*
...	...	...	...	...	...
"0100 0001"	"41"	A	A	A	A
...	...	...	...	...	...
"1110 0110"	"e6"	✘	æ	ć	ц
...	...	...	...	...	...
"1111 1101"	"fd"	✘	ý	ý	§
"1111 1110"	"fe"	✘	þ	ţ	ў
"1111 1111"	"ff"	✘	ÿ	˙	џ

Unicode representation

Each glyph has one number from 0x0 to 0x10ffff

glyph	unicode point	name	utf-8
A	U+0041	*Latin Capital letter A*	"41"
æ	U+00e6	*Latin Small letter Æ*	"c3 a6"
ć	U+0107	*Latin Small letter C with acute*	"c4 87"
ц	U+0446	*Cyrillic Small Letter Tse*	"d1 86"
ن	U+0646	*Arabic Letter Noon*	"d9 86"
و	U+0648	*Arabic Letter Waw*	"d9 88"
ر	U+0631	*Arabic Letter Reh*	"d8 b1"
स	U+0938	*Devanagari Letter Sa*	"e0 a4 b8
ღ	U+10E6	*Georgian Letter Ghan*	"e1 83 a6"
人	U+4EBA	*CJK Unified Ideograph-4EBA*	"e4 ba ba"
𐏈	U+103C8	*Old Persian Sign Auramazdaa*	"f0 90 8f 88"
🤓	U+1F913	*Nerd Face*	"f0 9f a4 93"
🇿🇦	-	-	-
🇿	U+1F1FF	*Regional Indicator Symbol Letter Z*	"f0 9f 87 bf"
🇦	U+1F1E6	*Regional Indicator Symbol Letter A*	"f0 9f 87 a6"

Tekst - Unicode UTF-8

Code point ↔ UTF-8 conversion

First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
U+0000	U+007F	0*yyyzzzz*			
U+0080	U+07FF	110*xxxyy*	10*yyzzzz*		
U+0800	U+FFFF	1110*wwww*	10*xxxxyy*	10*yyzzzz*	
U+010000	U+10FFFF	11110*uvv*	10*vvwwww*	10*xxxxyy*	10*yyzzzz*