

30.10.24 Agenda

17:15 - 19:00 | 19:00 - 19:15 | 19:15 - 21:15
Pause

- Datentypen + Operatoren + Raumzeitkomplexität
- Exkursion = Binäre Zahlensysteme
 - * Ganze Zahlen
 - * Negative ganze Zahlen
 - * Fließkommazahlen IEEE 754
- Datentypen + Speicherplatz in Python

Datentypen in Python (Grundlagen der Informatik)

- Neben der Laufzeitkomplexität, gibt es die Raum-/Zeitkomplexität \Rightarrow Gesamtmenge an Speicher, die der Algorithmus verwendet

Primitiven Daten (Grunddatentypen)

Art von Daten	Datentyp	$x \in \mathbb{Z}$
Ganze Zahlen	Integer	Bsp: -1, 98, 0, -273
Logische Wahrheitswerte	Boolean	Bsp: True, False 1, 0

Gleitkommazahl / Fliekommazahl	Float	Bsp: 3, 141, -21.5
Zeichenkette	Strings	"Hello World"

Arithmetische

Operatoren fr numerische Datentypen (Float, Integer)

Addition +, Division /,
 Subtraktion -, Ganzzahlige Division //,
 Modulo (Rest bei ganzzahliger Division) %,
 Multiplikation *, Potenzieren ** ($2^{+4} = 2^4$)

$x = x + 1 \rightarrow x += 1$
 $x = x - 1 \rightarrow x -= 1$

} Mit jedem
 arithmetischen
 Operator
 erweiterbar

\Rightarrow Zuweisung
 kein Gleichheits-
 zeichen (math.)

$x = x + 1$

Logische Operatoren fr Wahrheitswerte

$==$	ist gleich (Werte)	$x == y$
$===$	Vergleicht Datentyp & Wert	$x === y$
$!=$	ungleich	$x != y$
$>$	grer	$x > y$

\geq	größer oder gleich	$x \geq y$
$<$	kleiner	$x < y$
\leq	kleiner oder gleich	$x \leq y$

Boolesche Operatoren für Wahrheitswerte

Bedeutung	Syntax
UND	AND
ODER	OR
Nicht	NOT

Wertetabelle

AND

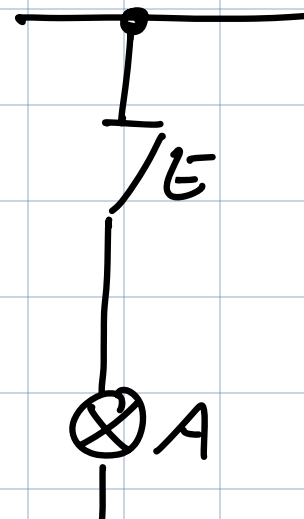
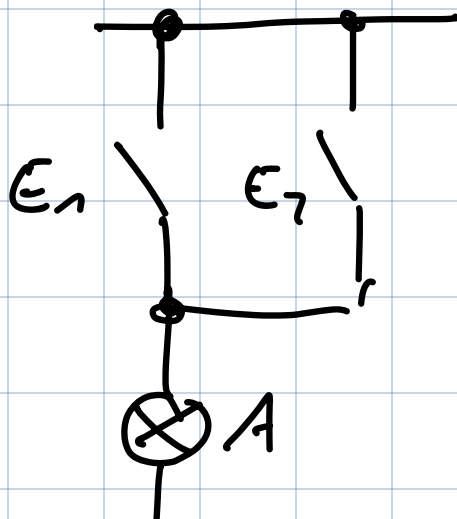
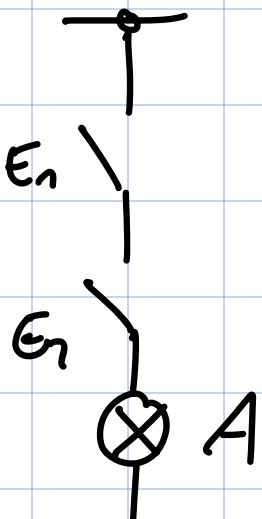
E_1	E_2	A
0	0	0
0	1	0
1	0	0
1	1	1

OR

E_1	E_2	A
0	0	0
0	1	1
1	0	1
1	1	1

Not

E	A
0	1
1	0



Theoretische Größen & Wertebereich (C, C++, Java)

→ Nicht Klassenarbeit relevant

Art von Daten	Datentyp	Wertebereich (Dezimalsystem)	Speicherplatz (Kapazität)
Ganze Zahl	Byte	0...255 -128...127	1 Byte $\hat{=}$ 8 Bit
	Integer	$-2^{31} \dots 2^{31} - 1$	4 Byte $\hat{=}$ 32 Bit
	Long	$-2^{63} \dots 2^{63} - 1$	8 Byte $\hat{=}$ 64 Bit
Fließkommazahl	Float	$\approx -3,4 \cdot 10^{38} \dots 3,4 \cdot 10^{38}$	4 Byte $\hat{=}$ 32 Bit
	Double	$\approx -1,7 \cdot 10^{309} \dots 1,7 \cdot 10^{309}$	8 Byte $\hat{=}$ 64 Bit
Wahrheitswert	Boolean	True, False 1, 0	1 Byte $\hat{=}$ 8 Bit

↳ Anmerkung Float: 7 Nachkommastellen
Double: 15 Nachkommastellen

Exkursion: Zahlensystemen

Hexadezimalsystem Basis = 16

Menge: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

ziffer

Dezimalsystem Basis = 10

Menge = $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$
Ziffer

Oktalesystem Basis = 8

Menge = $\{0, 1, 2, 3, 4, 5, 6, 7\}$

Binäre Zahlensystem Basis 2

Menge = $\{0, 1\}$

Aufgabe Bis 20:15 Uhr

- Umwandeln von positiv ganze Zahlen (Dezimal \rightarrow Binär & Binär \rightarrow Dezimal)
- Umwandeln von negativ ganzen Zahlen (Dezimal \rightarrow Binär)
- Zahlenbeispiele: 17 dez. \rightarrow Xbinär?
1011 Binär \rightarrow Xdez?
- 11 dez \rightarrow Xbinär?
- Festhalten der Ergebnisse in einer Powerpoint

- AB 20:15 Uhr Präsentation + Diskussion

Fließkommazahlen Nach IEEE 754

Bsp: -11,625

Informationen:

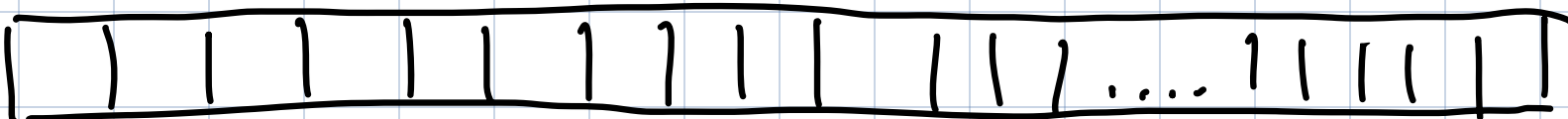
- Vorzeichen
- Zahlenwert (Vor & Nach-
den Komma)
- Ort vom Komma

Information:

- Vorzeichen
- Mantisse (Zahl)
- Charakteristik (Exponenten)

Float : 32 Bit Single-Precision-Float

- 1 Bit : Vorzeichen
- 8 Bit : Charakteristik (Exponent)
- 23 Bit : Mantisse



1 Bit	8 Bit	23 Bit
Vorzeichen	Charakteristik	Manisse

Double: 64 Bit Double-Precision-Float

1 Bit: Vorzeichen

11 Bit: Charakteristik

52 Bit: Manisse

316 \rightarrow 3,16 \cdot 10² Charakteristik

Manisse Basis

Umwandlung von Dezi in Binär

1. Vorkommazahl umwandeln

2. Nachkommazahl umwandeln

3. Normalisieren \rightarrow Manisse (Komma bis

4. Charakteristik ^{8 Bit Exponent} [Exponent + Bias] ! letzte 1 verschieben

float 32 Bit $2^8 - 1 = 127$

double 64 Bit $2^{11} - 1 = 1023$

\Rightarrow Bias ist ein fester Zahlenwert
(127 oder 1023)

5. Vorzeichen $0 \rightarrow$ Positiv

1 → Negativ

6. Fließkommazahl aufschreiben



Bsp: -11,625

1. $11 : 2 = 5 \text{ R } 1$
 $5 : 2 = 2 \text{ R } 1$
 $2 : 2 = 1 \text{ R } 0$
 $1 : 2 = 0 \text{ R } 1$

⇒ Schriftliche Division

Diagram: A vertical red line with a red arrow pointing up labeled 'Ziel' (Goal) and a red arrow pointing down labeled 'Start'. A purple arrow points down from the '0' in the last row to the word 'Fertig' (Finished) written in purple.

11_{Dez} → 1011_{Binär}

2. $0,625 \cdot 2 = 1,25$
 $0,25 \cdot 2 = 0,5$
 $0,5 \cdot 2 = 1,0$

⇒ Schriftliche Multiplikation

Diagram: A vertical red line with a red arrow pointing up labeled 'Start' and a red arrow pointing down labeled 'Ziel' (Goal). A green box highlights the results 1,25, 0,5, and 1,0. A purple arrow points down from the '1,0' to the word 'Fertig' (Finished) written in purple.

0,625_{Dez} → 0,101_{Binär}

3. Normalisieren → Nachkomma

Exponent
1

$$\underline{1011}, 101 \rightarrow 1, \underline{011101} \cdot 2^3$$

Na-hisse

Exponentwert \rightarrow Wie oft wandert mein Koma bis zur letzten 1.

4. Characteristic Exponent + Bias

$$3 + 127 = 130_{\text{decimal}}$$

130	:	2	=	65	R0
65	:	2	=	32	R1
32	:	2	=	16	R0
16	:	2	=	8	R0
8	:	2	=	4	R0
4	:	2	=	2	R0
2	:	2	=	1	R0
1	:	2	=	<u>0</u>	R1

Fertig

Ziel ↑
Start ↓

$$130_{\text{dez}} \rightarrow 10000010_{\text{bin}}$$

Charakteristisch

5. Vorzeichen Negativ $\rightarrow 1$

6. Zusammenfasse

1,011101 \rightarrow Mantisse

1 0000010 \rightarrow Charakteristisch

1 \rightarrow Vorzeichen

1 | 10000010 | 011101000...000

- 11,625 Dez \rightarrow

1 1000010 01110100...000 Binär

Objekt-Header