

# **706.088 INFORMATIK 1**

## **FUNKTIONEN, TEILGEBIETE DER INFORMATIK & GESCHICHTE**

# **WIEDERHOLUNG PROGRAMMIERUNG**

# HELLO WORLD



```
print("Hello World!")
```

```
Hello World!
```

# DEKLARATION VON VARIABLEN

- › Boolesche Werte (True, False)
- › Natürliche Zahlen (integer: 1, 11, 100000)
- › Reelle Zahlen (float: 0.1, 3.7)
- › Text (string: "Kette von Zeichen")
- › Kein Wert (None)
- › Tupel
- › Listen
- › Dictionary

# Operatoren

Operatoren definieren die Interaktion von Variablen und (deren) Werten

Operator	Beispiel	Ausgabe
+x, -x	-2	-2
+, -	$3 + 4$	7
*, /	$3 * 4$	12

# OperatorOPERATORENAusgabe

---

Operator	Beispiel	Ausgabe
//, %	13//3, 13 % 3	4,1
**	3**2	9

# OPERATOREN

Operator	Beispiel	Ausgabe
Operator	Beispiel	Ausgabe
not	not True	False
and, or	True or False, True and False	True, False
<, <=, >=, >, ==	4 > 5	False

# Operator **O**PERATOREN Ausgabe

---

## Zuweisung

Operator	Beispiel	Ausgabe
=	a = 4, b = a	a = 4, b = 4
+=	a += b	a = 8, b = 4
-=	a -= b	a = 4, b = 4
*=	a *= 3	a = 12, b = 4
/=	a /= b	a = 3.0, b = 4
//=	b //= a	a = 3.0, b = 1.0

# KONTROLLSTRUKTUREN

```
if True:  
    print("always here")  
else:  
    print("never here")
```

```
always here
```

# SCHLEIFEN

```
print("forever")
while True:
    print("and ever")
```

```
forever
and ever
and ever
and ever
...
...
```

# SCHLEIFEN

```
list = ["test1", "test2", "test3"]
for element in list:
    print(element)
```

```
test1
test2
test3
```

# FUNKTIONEN

# FUNKTIONEN

dienen der Kapselung von Code nach logischen Einheiten

```
pets = ["Hamster", "Cat", "Dog", "Canary"]

def not_a_dog_person(animal):
    return animal.replace("Dog", "Fish")

for pet in pets:
    print("Tom: {:10} Tim: {}".format(pet, not_a_dog_person(pet)))
```

```
Tom: Hamster      Tim: Hamster
Tom: Cat          Tim: Cat
Tom: Dog          Tim: Fish
Tom: Canary       Tim: Canary
```

# FUNKTIONEN

```
def add(x, y):  
    return x + y  
  
def print_result(result):  
    print("Result: {}".format(result))  
  
def main():  
    a = 10  
    b = 20  
    c = add(a, b)  
    print_result(c)  
  
if __name__ == "__main__":  
    main()
```

```
Result: 30
```

# DEFINITION EINES GUTEN PROGRAMMS

- › Funktionalität
- › Lesbarkeit
- › Wartbarkeit
- › Robustheit
- › Korrektheit
- › Effizienz
- › Portierbarkeit

# EINGABE-/AUSGABE PARAMETER

```
import sys  
  
print(sys.argv)
```

```
$ python3 sysargv.py test1 123 --flag  
['sysargv.py', 'test1', '123', '--flag']
```

```
import sys
sum = 0
for element in sys.argv[1:]:
    sum += int(element) # same as: sum = sum + int(element)
print(sum)
```

```
$ python3 sysargv2.py 1 2 3 4 5 6
21
```

# GESCHICHTE DER INFORMATIK

# **WAS IST INFORMATIK**

**Wissenschaft der " systematischen " oder " automatischen  
Verarbeitung " von " Information "**

**Information + Automatik**

# GESCHICHTE DER INFORMATIK

- › Aus 3 Teilgebieten zusammengesetzt
  - › Mathematik
  - › Mechanik
  - › Elektronik
- › ... oder vom Abakus zum Quantencomputer

# MATHEMATIK UND INFORMATIK

- › Verschiedene Meilensteine zwischen 30.000 v. Chr. und 200 n. Chr.
- › Abakus (200 n. Chr.)
- › Begründung des Dezimalsystem (500 n. Chr.)
- › Arabische Ziffern in Europa (1200 n. Chr.)
- › Algorithmen zur Überführung von Multiplikationen in Additionen (1600 n. Chr.)

# GESCHICHTE DER INFORMATIK

'Logische Maschinen' bereits im 13. Jahrhundert

Wunsch der Automation von mathematischen  
Berechnungen

Erstes **mechanisches** Rechengerät wurde im Jahr 1623 von  
Wilhelm Schickard gebaut

# MATHEMATIK, MECHANIK UND INFORMATIK

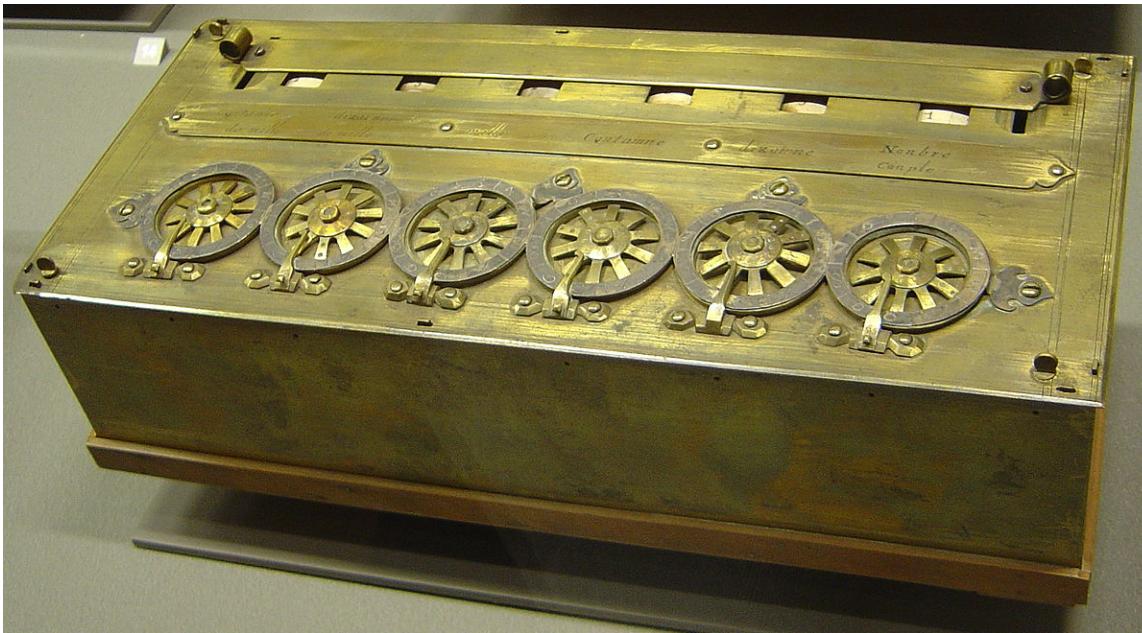


Schickard'sche Rechenmaschine (1623 n. Chr)  
By [Herbert Klaeren](#) - Transferred from [1], CC BY-SA 3.0, Link

# SCHICKARD'SCHE RECHENMASCHINE

- › 1623 n. Chr
- › Basierend auf Zahnräderen
- › Addition und Subtraktion von bis zu **sechsstelligen** Zahlen
- › Speicherüberlauf
  - » Akustisches Signal
- › Pläne bis 1960 verloren

# BLAISE PASCAL - PASCALINE

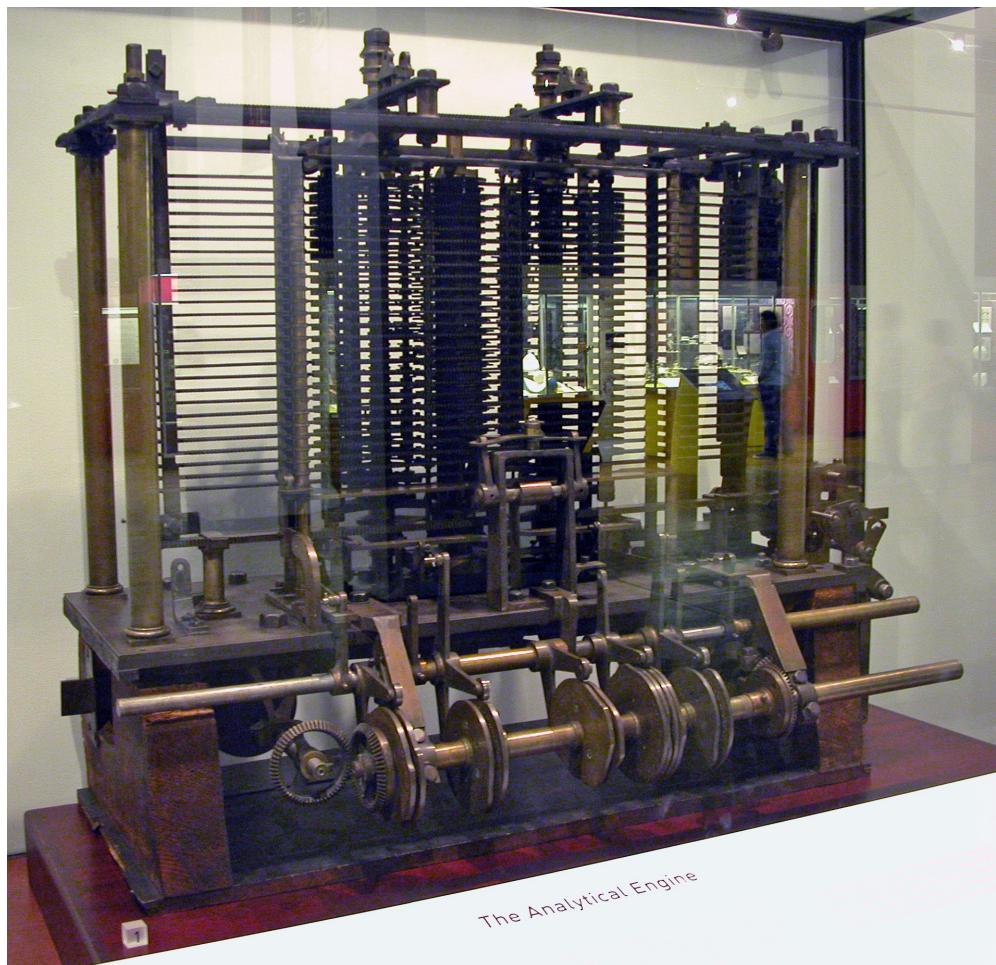


Pascaline  
By © 2005 David Monniaux / , CC BY-SA 3.0, Link

- › Erster (mechanischer) Taschenrechner
- › 50 Prototypen
- › Öffentliche Präsentation 1645
- › Funktionsumfang
  - » Addition
  - » Subtraktion
  - » von 2 Zahlen

# MATHEMATIK, MECHANIK UND INFORMATIK

- › Charles Babbage (1791 - 1871)
  - » Mathematiker, Philosoph, Erfinder und Entwickler.
  - » “Einer der Väter des (mechanischen) Computers”
  - » Bekannt für die Babbage- oder Difference-Engine zur Lösung polinomialer Gleichungen.



Analytical Machine von Babbage

Von Bruno Barral (ByB), CC BY-SA 2.5, <https://commons.wikimedia.org/w/index.php?curid=6839854>

# MATHEMATIK, MECHANIK UND INFORMATIK

- › Herman Hollerith (1860 - 1929)
  - » Erfand das Lochkartensystem zur Datenspeicherung (1884)
  - » Automatische Auswertung der gespeicherten Daten durch 'Lochkarten-Leser'
  - » Ebnete den Weg zur Speicherung von ersten Computer-Instruktionen.
  - » Verwendung der Hollerith Maschine (Tabulating machine ) zur Volkszählung 1890 in den USA

1	1	3	0	2	4	10	On	S	A	C	E	a	c	e	g	EB	SB	Ch	Sy	U	Sh	Hk	Br	Rm
2	2	4	1	3	E	15	Off	IS	B	D	F	b	d	f	h	SY	X	Fp	Cn	R	X	Al	Cg	Kg
3	0	0	0	0	W	20		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
A	1	1	1	1	0	25	A	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
B	2	2	2	2	5	30	B	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
C	3	3	3	3	0	3	C	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
D	4	4	4	4	1	4	D	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
E	5	5	5	5	2	C	E	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
F	6	6	6	6	A	D	F	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
G	7	7	7	7	B	E	G	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
H	8	8	8	8	a	F	H	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
I	9	9	9	9	b	c	I	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

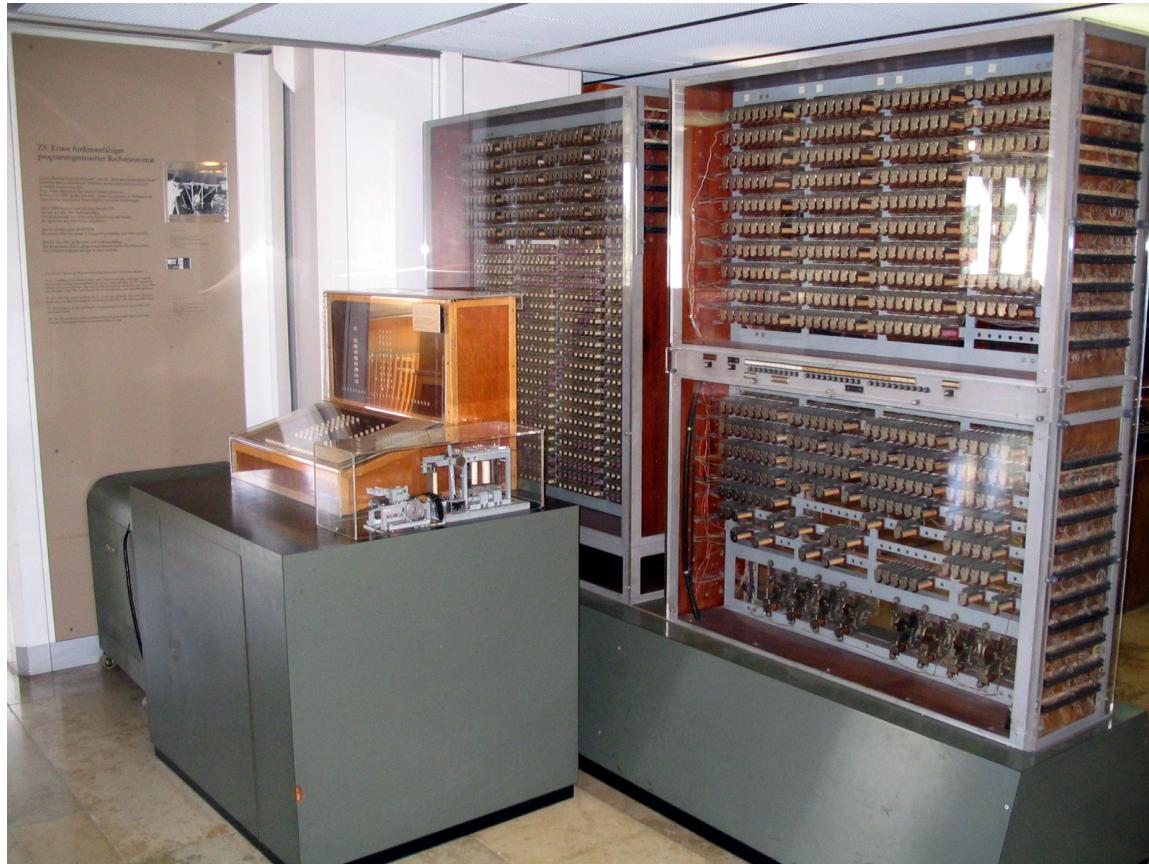
Lochkarte

By Unknown - Library of Congress <http://memory.loc.gov/mss/mcc/023/0008.jpg>, Public Domain,  
<https://commons.wikimedia.org/w/index.php?curid=30538485>

# KONRAD ZUSE (1910 - 1995)

Entwickelte den **ersten Digital-rechner** weltweit,  
die **Z3** (1941)

- › 600 Relais für den Prozessor
- › 1600 Relais für den Speicher
- › Basierte auf **binärer Gleitkommaarithmetik**
- › So groß wie ein (sehr großer) Kleiderschrank
- › 15-20 Arithmetische Operationen / Sekunde Weitere Informationen zu Konrad Zuse: <http://zuse.zib.de>



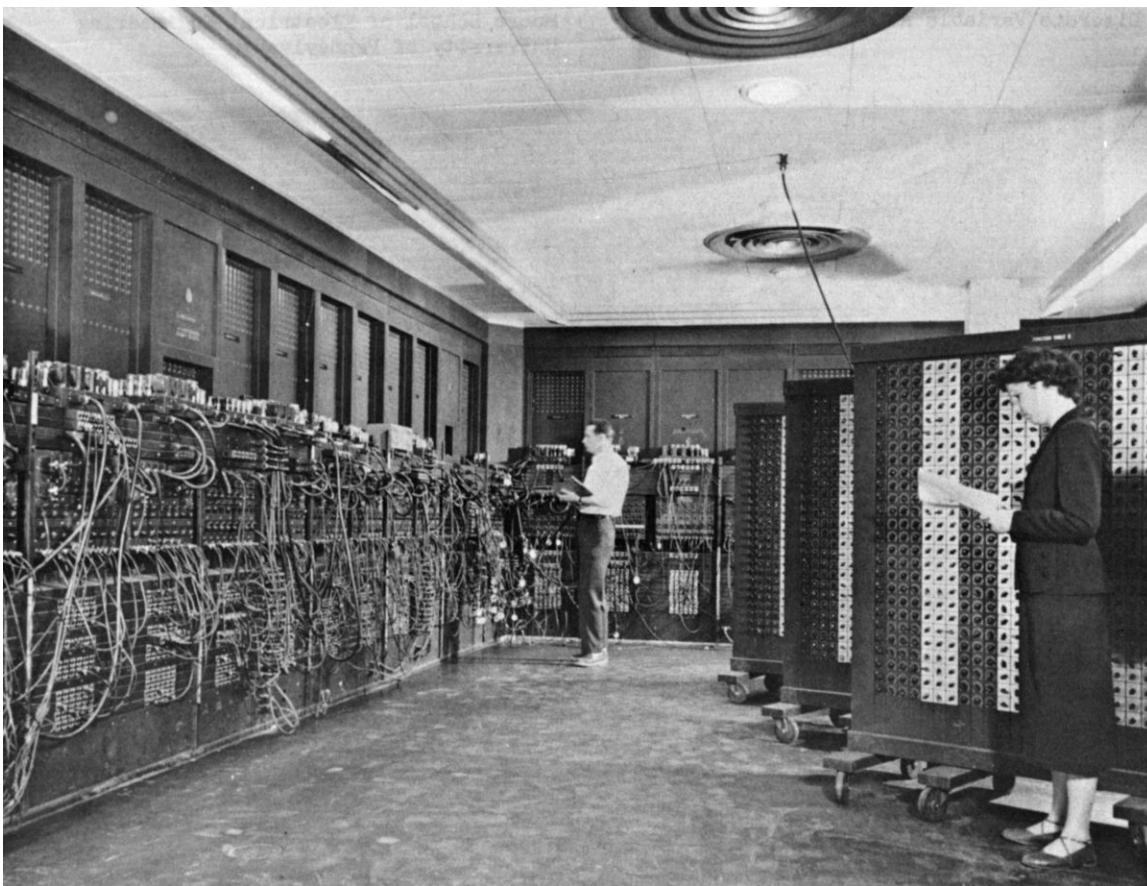
Z3 im Deutschen Museum

Von [Venusianer](#) aus der [deutschsprachigen Wikipedia](#), CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=3632073>

# ENIAC

Erster 'rein elektronischer' Rechner

- › 1946 vorgestellt
- › Elektronenröhren (Vacuum Tubes) zur elektrischen Speicherung von Zahlen
- › 17.468 Elektronenröhren
- › 167 m<sup>2</sup>
- › 27 Tonnen
- › 170 kW
- › USD 468.000 ~ heute USD 6.8 Mio



Eniac

Von Unbekannt - U.S. Army Photo, Gemeinfrei, <https://commons.wikimedia.org/w/index.php?curid=55124>

# ANGLEBIGKEIT VON ELEKTRONENRÖHREN



By Stefan Riepl ([Quark48](#)) - Self-photographed, [CC BY-SA 2.0 de](#), [Link](#)

Ca. alle zwei Tage kam es zu einem Ausfall einer  
Elektronenröhre

15 Minuten bis fehlerhafte Elektronenröhre gefunden wurde

# FIRST BUG



- › Bezeichnung 'Bug' für Fehler in Programmabläufen geht zurück auf Computer Pionierin **Grace Hopper**
- › Am 9. September 1947 dokumentierte sie eine Motte in einem Relais des Computers **Mark II Aiken Relay Calculator** im Log-Buch mit den Worten: "**First actual case of bug being found**"

17/2

9/9

0800

Autam started

1000

" stopped - autam ✓

13"uc (032) MP-MC

(033) PRO 2

convdt

{ 1. 2700 9.037 847 025

9.037 846 995 convdt

~~1.98267000~~ 2. 130476415 (3) 4.615925059 (-2)

2. 130476415

2. 130676415

Relays 6-2 in 033 failed special sped test  
in relay " 10.000 test .

Relay  
2145  
Relay 3370

1100

Started Cosine Tape (Sine check)

1525

Started Multi Adder Test.

1545



Relay #70 Panel F  
(moth) in relay.

1600

First actual case of bug being found.

autamt started.

1700 closed down .

By Courtesy of the Naval Surface Warfare Center, Dahlgren, VA., 1988. - U.S. Naval Historical Center Online Library  
Photograph NH 96566-KN, Public Domain, <https://commons.wikimedia.org/w/index.php?curid=165211>



Grace Hopper, By James S. Davis - Public Domain, <https://commons.wikimedia.org/w/index.php?curid=12421475>  
Grace Hopper "Queen of Software" on Letterman 

# MAILÜFTTERL

- › 1955 an der TU Wien von Heinz Zemanek mit Studierenden gebaut.
- › erster Volltransistor Rechner am Europäischen Festland
- › 3.000 Transistoren, 5.000 Dioden, 20 km Draht, 132 kHz
- › Transistorspende von Philips
- › erste Berechnung 1958

# MAILÜFTERL

 go there

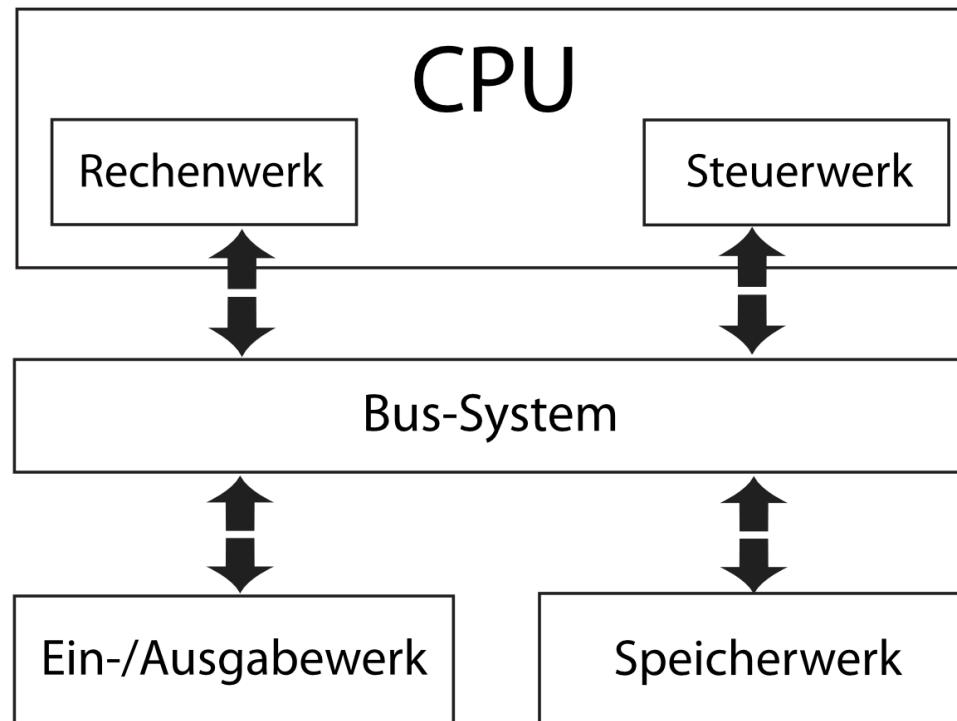
# INFORMATIK

Johann von Neumann erfindet 1946 die  
**Von Neumann Architektur:**

- › Programm und Daten werden gleichzeitig im  
Hauptspeicher gehalten

# AUFBAU EINES COMPUTERS

Von Neumann Architektur als Referenzmodell für einen Computers



# INFORMATIK

Shockley, Bardeen und Brattain (Bell Labs) entdecken 1947 den ersten Transistoreffekt und stellen den ersten **Transistor** vor.

› 1956 Nobelpreis für Physik

## Rechenleistung bis ca. 1960

- › 1-2 KByte Speicher
- › 0.02 MIPS

## Rechenleistung 1986 bis heute

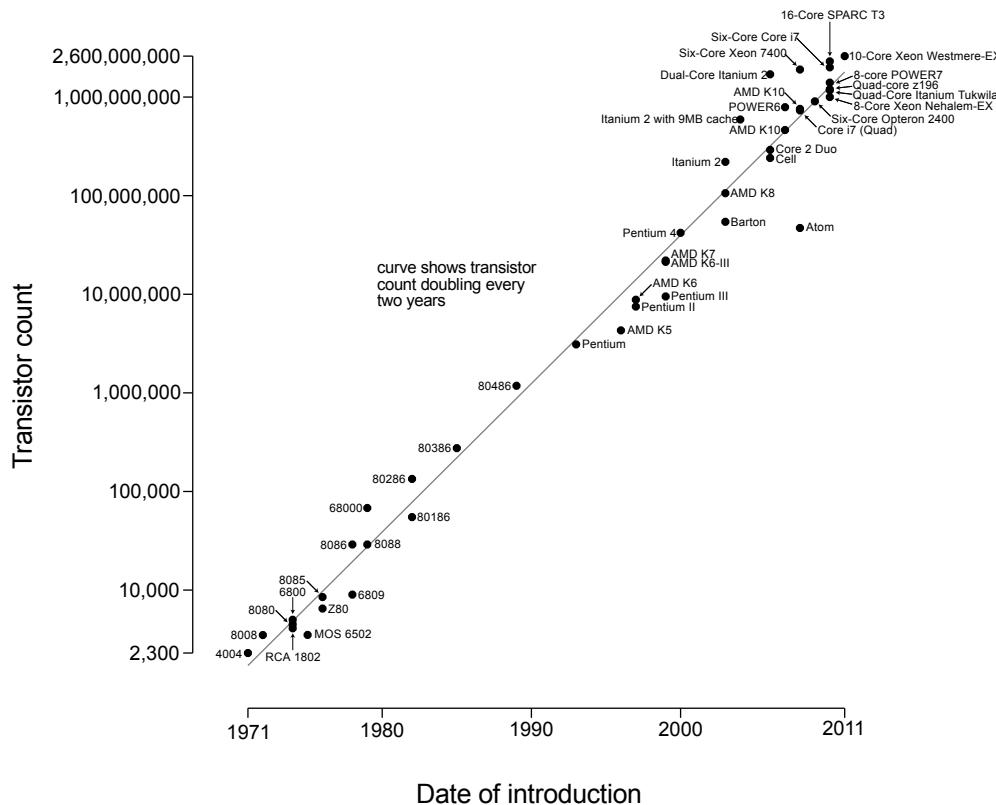
- › 8 MByte - ~ 32GByte
- › > 100 MIPS

# MOORE'S LAW

- › Anzahl der Transistoren verdoppelt sich in regelmäßigen Abständen (alle 2 Jahre). → Leistung in **MIPS** verdoppelt sich auch.

# MOORE'S LAW

Microprocessor Transistor Counts 1971-2011 & Moore's Law



By Wgsimon - Own work, CC BY-SA 3.0, <https://commons.wikimedia.org/w/index.php?curid=15193542>

# **FRAGEN?**

# NÄCHSTES MAL

2016-11-09 16:00