Informatik 1
Data and Algorithms

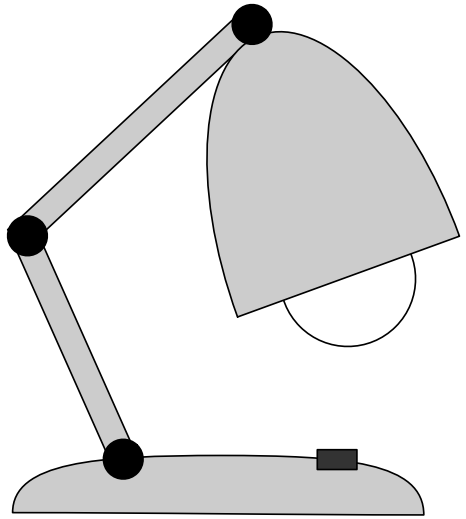
Prof. Harald Gall

University of Zurich, Department of Informatics

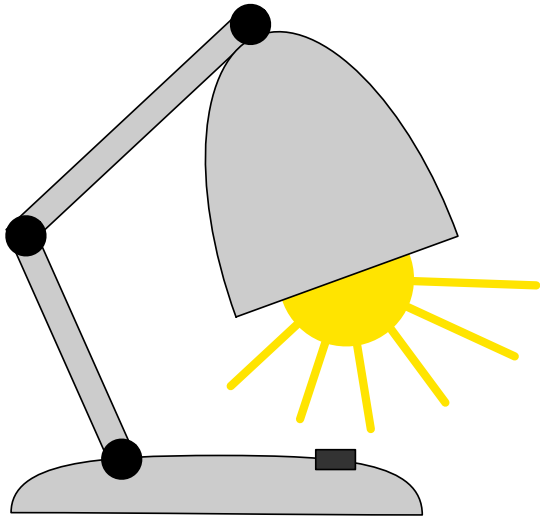
How does a computer represent data?


https://commons.wikimedia.org/wiki/File:Binary_Code.jpg

Computers differentiate between ON and OFF



0



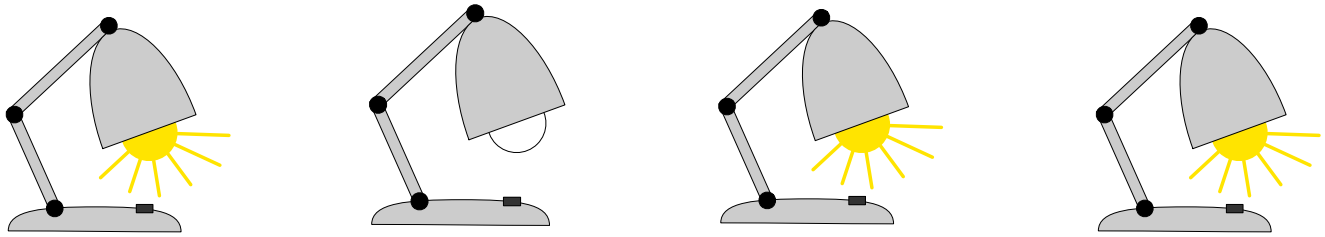
1

Conventional Counting

1	4	6	9
1000s	100s	10s	1s
10^3	10^2	10^1	10^0

Binary Counting

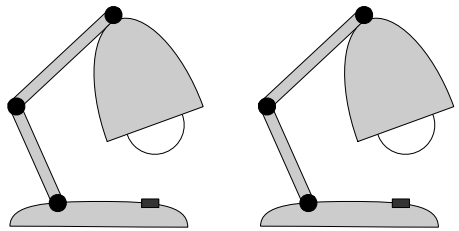
...



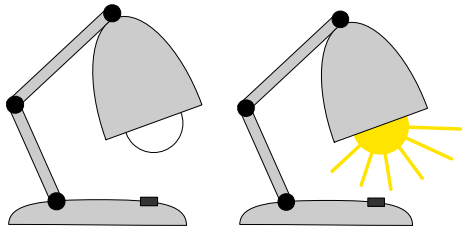
$2^3 = 8$ $2^2 = 4$ $2^1 = 2$ $2^0 = 1$

1x8 + 0x4 + 1x2+ 1x1 = 11

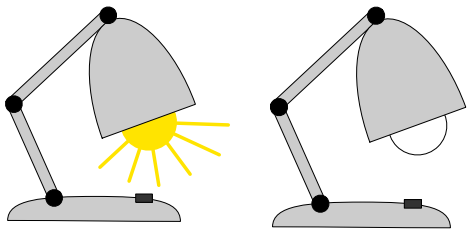
Examples



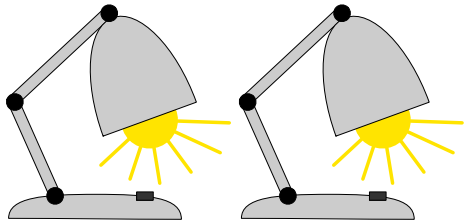
= 0



= 1

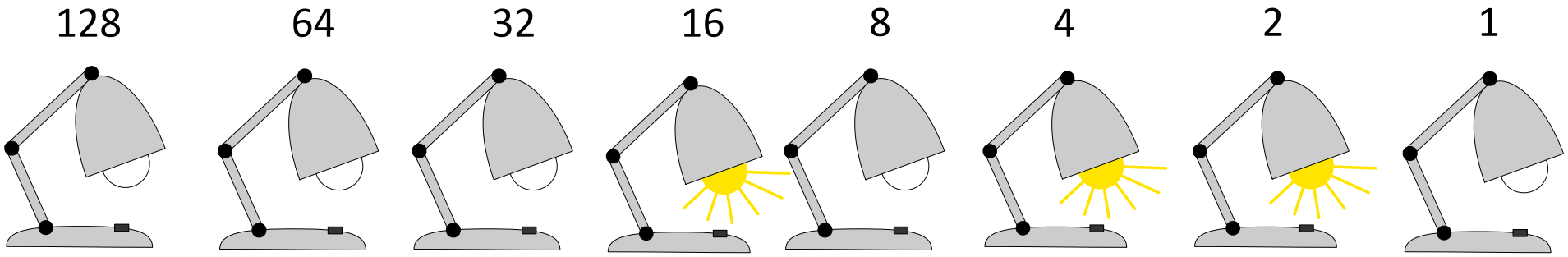


= 2



= 3

1 Byte (= 8 Bit)



Minimal Value? → 0

Maximal Value? → 255

This Value? → 22

USASCII code chart

b_7 b_6 b_5 → / Bits	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
b_4 b_3 b_2 b_1 ↓ / Column → / Row ↓	0	1	2	3	4	5	6	7
0 0 0 0	0	NUL	DLE	SP	0	@	P	`	p
0 0 0 1	1	SOH	DC1	!	1	A	Q	a	q
0 0 1 0	2	STX	DC2	"	2	B	R	b	r
0 0 1 1	3	ETX	DC3	#	3	C	S	c	s
0 1 0 0	4	EOT	DC4	$	4	D	T	d	t
0 1 0 1	5	ENQ	NAK	%	5	E	U	e	u
0 1 1 0	6	ACK	SYN	&	6	F	V	f	v
0 1 1 1	7	BEL	ETB	'	7	G	W	g	w
1 0 0 0	8	BS	CAN	(	8	H	X	h	x
1 0 0 1	9	HT	EM	)	9	I	Y	i	y
1 0 1 0	10	LF	SUB	*	:	J	Z	j	z
1 0 1 1	11	VT	ESC	+	;	K	[	k	{
1 1 0 0	12	FF	FS	,	<	L	\	l	|
1 1 0 1	13	CR	GS	-	=	M	]	m	}
1 1 1 0	14	SO	RS	.	>	N	^	n	~
1 1 1 1	15	SI	US	/	?	O	_	o	DEL

1000001 = 65

1000001 = A

1000001 = ..?

Binary code does not have a meaning *per se*, the meaning depends on the interpretation!

https://en.wikipedia.org/wiki/ASCII#/media/File:USASCII_code_chart.png

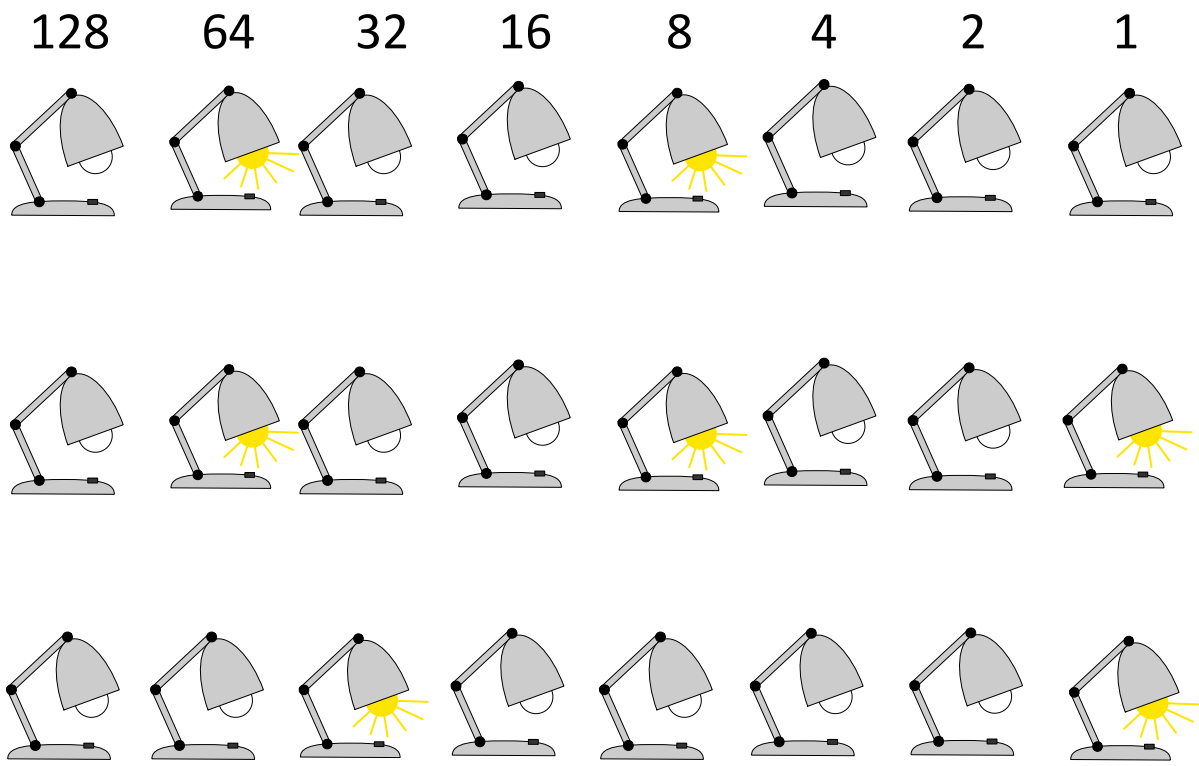
Text is a string of characters

A = 65

ABC = 65, 66, 67

Aa! = 65, 97, 33

Example



= 72 = H

= 73 = I

= 33 = !

How to actually instruct a computer to do something?

Programming Language

- A programming language is a formal language that specifies a set of instructions (or commands) that can be given to a computer to produce various kinds of output. (Wikipedia)

- Programming Languages define syntax, valid ways to express intent through declarations, statements and expressions.

- The semantics of a program are properties that describe the meaning of the executed instructions.

Computer Program

- A computer program is a collection of instructions that perform a specific task when executed by a computer.
- Programs can run indefinitively (“endless loop”).

Algorithm, an informal definition

An algorithm unambiguously defines the steps to perform a particular type of task.

An algorithm takes input parameters, performs a series of computations, and returns an output.

An algorithm must halt eventually.

Cooking recipes resemble the definition of an algorithm. Due to the lack of clear syntax and semantics, their execution is typically not well-defined.

Examples of Algorithms

- Sorting values
- Finding the shortest route from home to university
- Calculating the cost of an assembled good (e.g., car engine)
- Predicting the stock market price

Algorithm: Find name in phone book

1. Open the phone book
2. As long as you do not see the name, turn the page
3. Remember the number next to the name

This is pseudocode, a list of informal statements in an imaginary programming language that are to be executed.

What happens if the name does not exist?

Algorithm: Find name in phone book, Take 2

1. Open the phone book
2. As long as you do not see the name, turn the page
3. Did you find the name?
   → Remember the number next to the name
4. Did you come to the end of the phone book?
   → The name does not seem to be there.

Algorithm: Count the number of people

1. num = 0
2. for every person in the room:
    1. num = num + 1

Programs can remember intermediate values by assigning them to variables.

The execution time of this algorithm grows linearly with the number of people to count. The more people in the room, the longer it takes to count them.

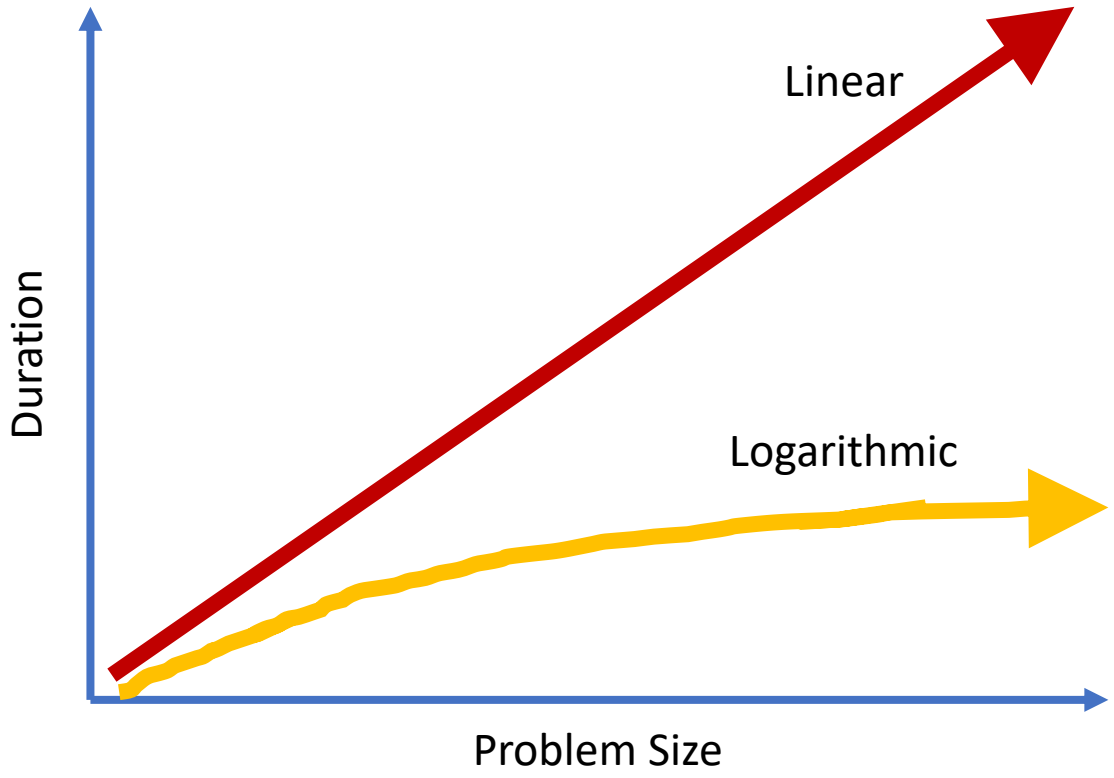
Social Algorithm: Counting

1. All of you stand up and remember the number ”1”
2. Find somebody else in the audience that is standing. Get in contact with each other and add up your numbers.
3. One of you sits down, the other one goes back to step 2

What is different to the first approach?

Division of the problem into smaller pieces, better scaling.

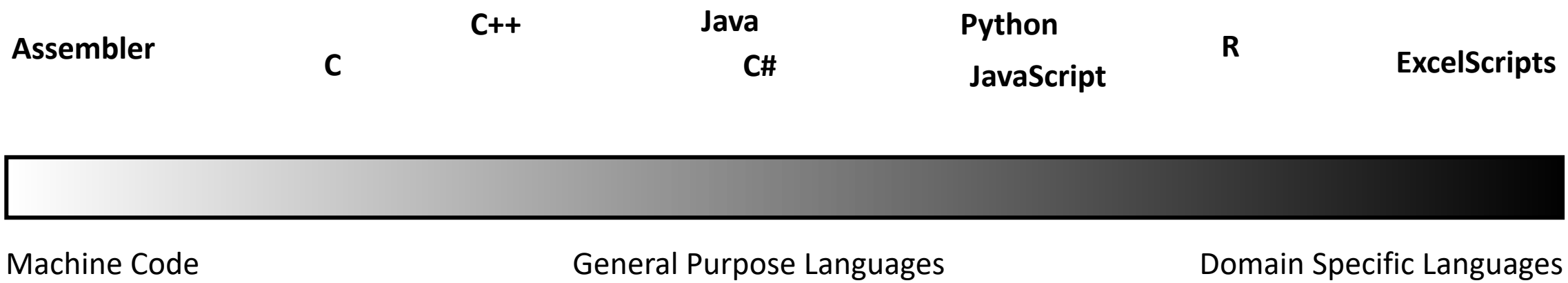
Some algorithms are faster than others



In our "social algorithm", counting a room 2 times the size would only have taken insignificantly longer!

First steps in Python

We are going to learn Python. Why?



Python is a high-level programming language. It is a general language that features many advanced concepts, but has little syntactical overhead, which makes it easy to learn.

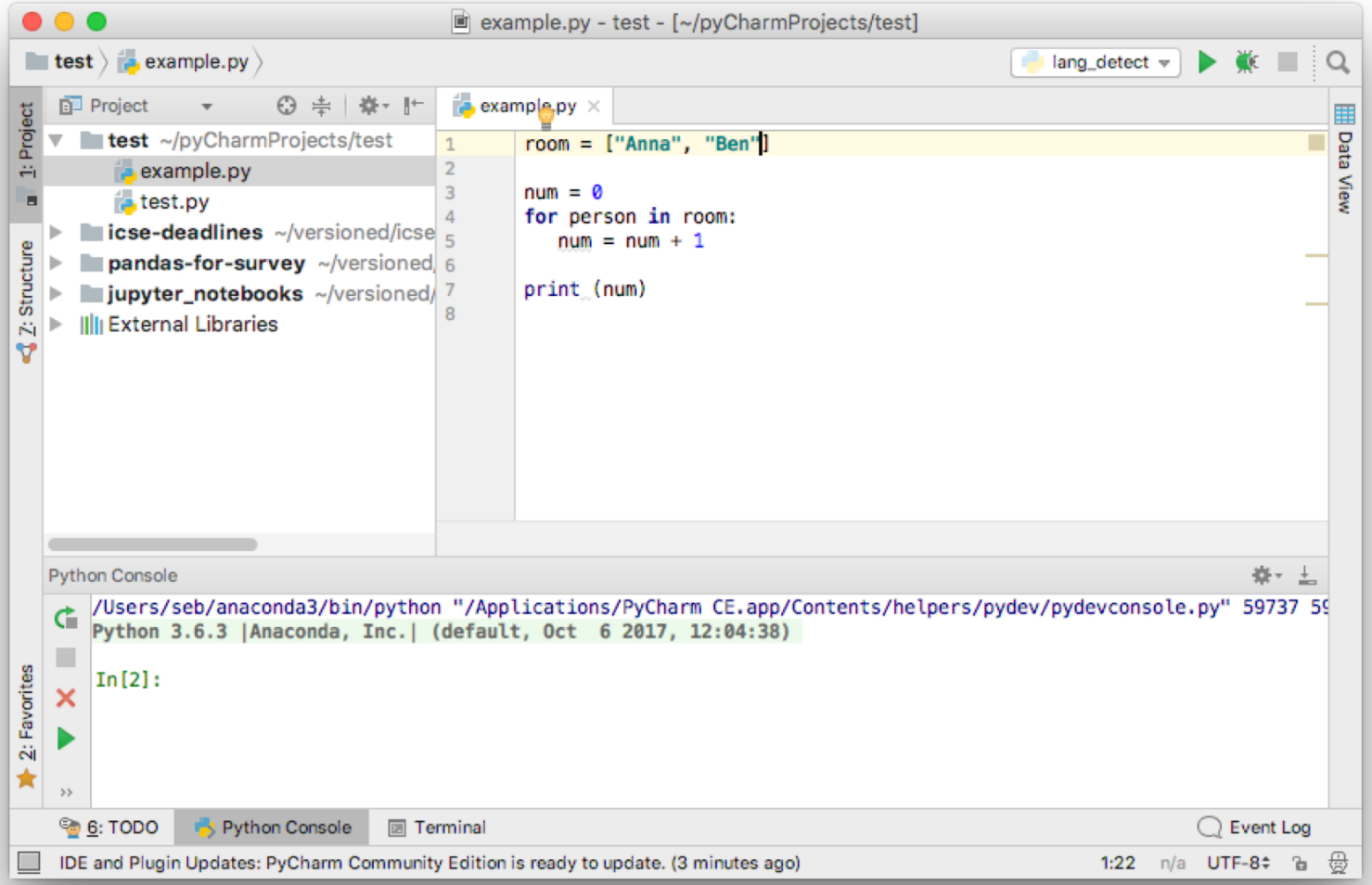
Python Programs are Plain-Text

```
room = [“Anna”, “Ben”]
num = 0
for person in room:
    num = num + 1
print (num)
```

```
$ python3 counting.py
2
```

We will learn how to represent data that is more complex than numbers and strings.

Python programs can be executed by running an interpreter.



An Integrated Development Environment makes it easier to read, write and execute programs.

Several alternative IDEs exist, such as PyCharm or Anaconda

Find an IDE that you like and get comfortable using the integrated tools (syntax highlighting, debugger, etc.)

You can request input and print on the screen

```
print ("hello world")
msg = "hello world"
print (msg)

name = input()
print ("hello " + name)
```

Use the `print()` command to print information on the screen.

You can print literal values (such as strings) and variables.

Use the `input()` function to request information from the user.

Use `+` to concatenate different strings.

Course Contents

Todays lecture in a nutshell

- Computers represent data in a binary form, combining multiple bits allows to encode numbers and even strings of characters
- Computers can be instructed through statements
- Computers can remember intermediate computation results
- Algorithms are repeatable recipes to solve a particular problem type
- Different algorithms might exist for the same task
- Programming languages like Python provide means to input/output
- Python programs are plain-text and need to be interpreted

Contents of the Remaining Course

- Data Types, Operators, Expressions
- Conditional and Iterative Statements
- Functions
- Built-In Data Structures: Tuples, Lists, Dictionaries
- Testing and Debugging
- Object Oriented Programming:
  Information Hiding, Inheritance, Design Principles
- Code Organization: Modules and Version Control