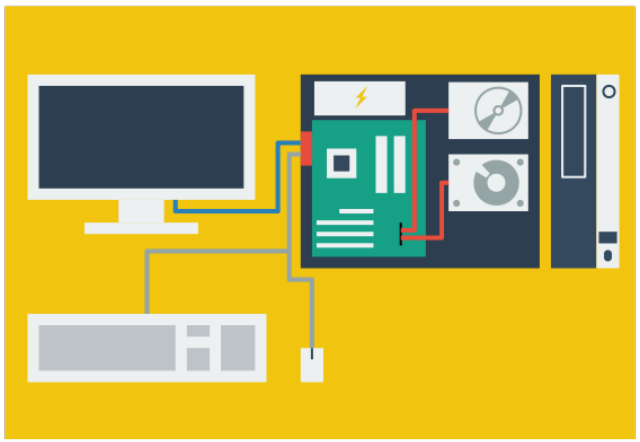
How does a Computer Work?

How does a Computer Work?

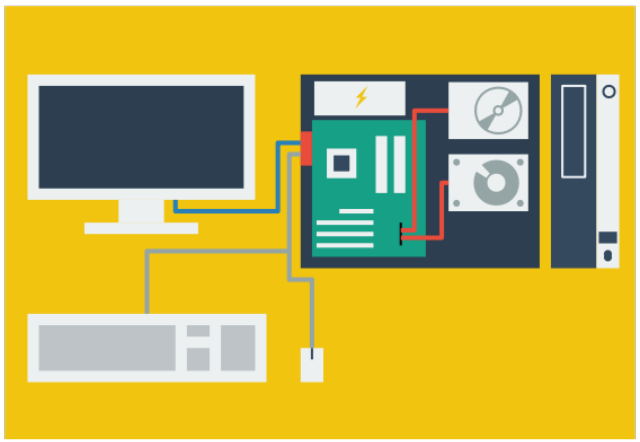
Everything you need to know about hardware, compilers, and formal languages to get started in Python

Hardware vs Software

Hardware vs Software

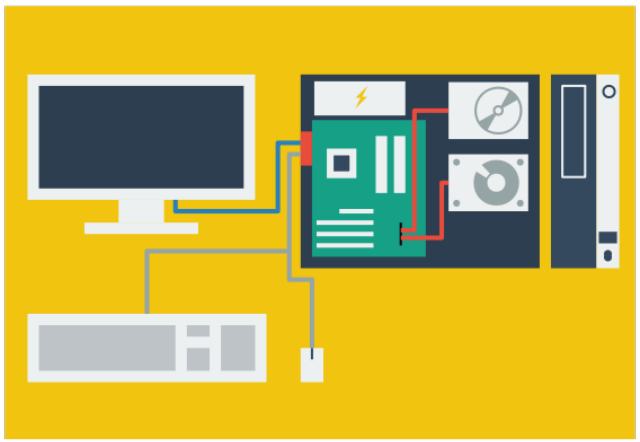


Hardware vs Software

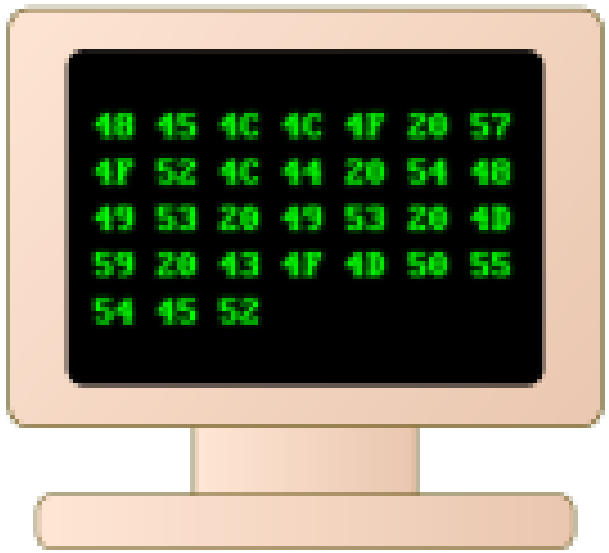


vs.

Hardware vs Software



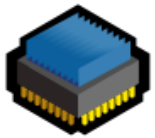
vs.



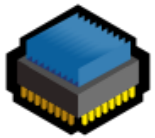
Hardware Examples

Hardware Examples

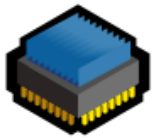
Hardware Examples

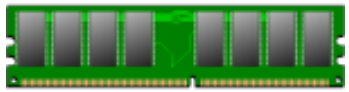


Hardware Examples

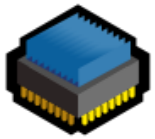


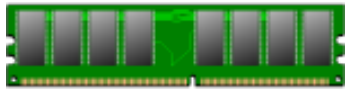
Hardware Examples



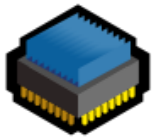


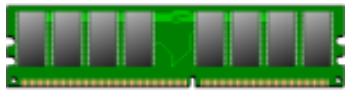
Hardware Examples





Hardware Examples





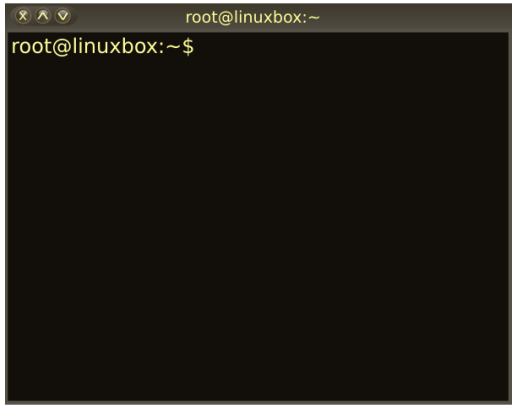
Software Examples

Software Examples

System Software:

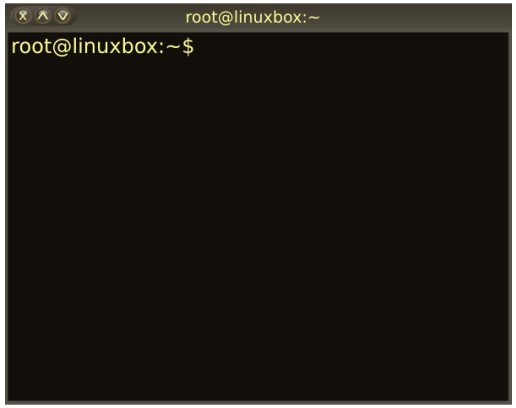
Software Examples

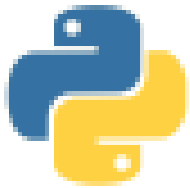
System Software:



Software Examples

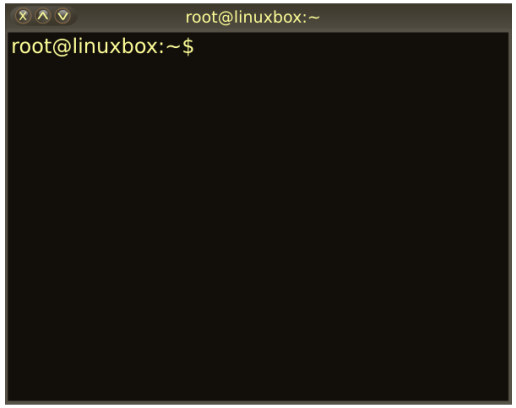
System Software:

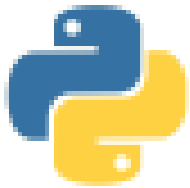




Software Examples

System Software:

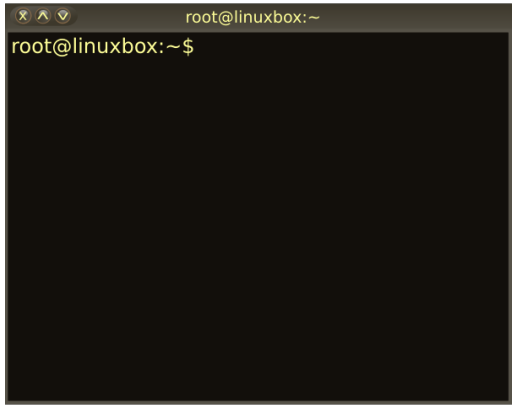


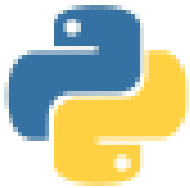


Application Software:

Software Examples

System Software:

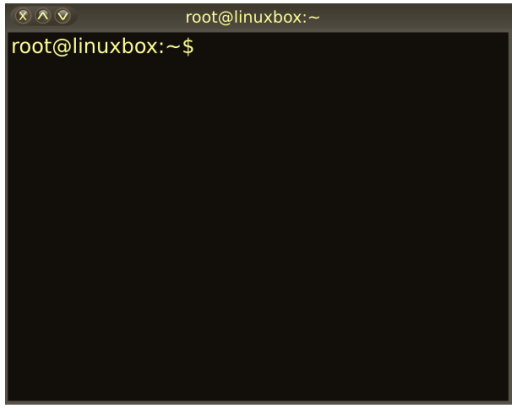




Application Software:

Software Examples

System Software:



Application Software:

Concept Check!

For each of the following, which are examples of hardware and which are examples of software?

- Your computer's CPU
- Your favorite videogame
- The google chrome browser
- Your computer webcam
- Your graphics card (GPU)
- Google sheets
- Your gaming headset
- Microsoft Windows 10
- Your first Python program
- Thumb drive containing your 1st Python program

Solutions!

For each of the following, which are examples of hardware and which are examples of software?

Solutions!

For each of the following, which are examples of hardware and which are examples of software?

- Your computer's CPU **hardware (processor)**

Solutions!

For each of the following, which are examples of hardware and which are examples of software?

- Your computer's CPU — **hardware (processor)**
- Your favorite videogame — **software (application)**

Solutions!

For each of the following, which are examples of hardware and which are examples of software?

- Your computer's CPU **hardware (processor)**
- Your favorite videogame **software (application)**
- The google chrome browser **software (application)**

Solutions!

For each of the following, which are examples of hardware and which are examples of software?

- Your computer's CPU **hardware (processor)**
- Your favorite videogame **software (application)**
- The google chrome browser **software (application)**
- Your computer webcam **hardware (input device)**

Solutions!

For each of the following, which are examples of hardware and which are examples of software?

- Your computer's CPU — **hardware (processor)**
- Your favorite videogame — **software (application)**
- The google chrome browser — **software (application)**
- Your computer webcam — **hardware (input device)**
- Your graphics card (GPU) — **hardware (processor)**

Solutions!

For each of the following, which are examples of hardware and which are examples of software?

- Your computer's CPU **hardware (processor)**
- Your favorite videogame **software (application)**
- The google chrome browser **software (application)**
- Your computer webcam **hardware (input device)**
- Your graphics card (GPU) **hardware (processor)**
- Google sheets **software (application)**

Solutions!

For each of the following, which are examples of hardware and which are examples of software?

- Your computer's CPU — **hardware (processor)**
- Your favorite videogame — **software (application)**
- The google chrome browser — **software (application)**
- Your computer webcam — **hardware (input device)**
- Your graphics card (GPU) — **hardware (processor)**
- Google sheets — **software (application)**
- Your gaming headset — **hardware (input/output)**

Solutions!

For each of the following, which are examples of hardware and which are examples of software?

- Your computer's CPU — **hardware (processor)**
- Your favorite videogame — **software (application)**
- The google chrome browser — **software (application)**
- Your computer webcam — **hardware (input device)**
- Your graphics card (GPU) — **hardware (processor)**
- Google sheets — **software (application)**
- Your gaming headset — **hardware (input/output)**
- Microsoft Windows 10 — **software (system sw)**

Solutions!

For each of the following, which are examples of hardware and which are examples of software?

- Your computer's CPU — **hardware (processor)**
- Your favorite videogame — **software (application)**
- The google chrome browser — **software (application)**
- Your computer webcam — **hardware (input device)**
- Your graphics card (GPU) — **hardware (processor)**
- Google sheets — **software (application)**
- Your gaming headset — **hardware (input/output)**
- Microsoft Windows 10 — **software (system sw)**
- Your first Python program — **software (application)**

Solutions!

For each of the following, which are examples of hardware and which are examples of software?

- Your computer's CPU **hardware (processor)**
- Your favorite videogame **software (application)**
- The google chrome browser **software (application)**
- Your computer webcam **hardware (input device)**
- Your graphics card (GPU) **hardware (processor)**
- Google sheets **software (application)**
- Your gaming headset **hardware (input/output)**
- Microsoft Windows 10 **software (system sw)**
- Your first Python program **software (application)**
- Thumb drive w/ Python prog **hardware (memory/storage)**

Diving Deeper: How to Store Information?

Diving Deeper: How to Store Information?

Computer is just a bunch of on/off switches:

Diving Deeper: How to Store Information?

Computer is just a bunch of on/off switches:

- **off** – 0
- **on** – 1

Diving Deeper: How to Store Information?

Computer is just a bunch of on/off switches:

- **off** – 0
- **on** – 1

How to store things?

Diving Deeper: How to Store Information?

Computer is just a bunch of on/off switches:

- **off** – 0
- **on** – 1

How to store things?

Diving Deeper: How to Store Information?

Computer is just a bunch of on/off switches:

- **off** – 0
- **on** – 1

How to store things?

- Integer numbers – binary code

Diving Deeper: How to Store Information?

Computer is just a bunch of on/off switches:

- **off** – 0
- **on** – 1

How to store things?

- Integer numbers – binary code
- Decimal numbers – store the integer and decimal place in binary

Diving Deeper: How to Store Information?

Computer is just a bunch of on/off switches:

- **off** – 0
- **on** – 1

How to store things?

- Integer numbers – binary code
- Decimal numbers – store the integer and decimal place in binary
- Letters/Words – store binary codes that can be looked-up in ASCII table

Binary Codes

Binary Codes

Binary:

Binary Codes

Binary: Count up from 0, but there are only 2 digits:

- 0
- 1

Binary Codes

Binary: Count up from 0, but there are only 2 digits:

- 0
- 1

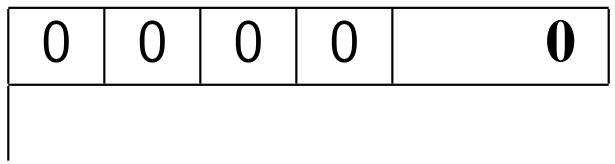
So instead of "reaching 10" after 9, 10 is the next number after 1:

Binary Codes

Binary: Count up from 0, but there are only 2 digits:

- 0
- 1

So instead of "reaching 10" after 9, 10 is the next number after 1:



Binary Codes

Binary: Count up from 0, but there are only 2 digits:

- 0
- 1

So instead of "reaching 10" after 9, 10 is the next number after 1:

0	0	0	0	**0**
0	0	0	1	**1**

Binary Codes

Binary: Count up from 0, but there are only 2 digits:

- 0
- 1

So instead of "reaching 10" after 9, 10 is the next number after 1:

0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**2**

Binary Codes

Binary: Count up from 0, but there are only 2 digits:

- 0
- 1

So instead of "reaching 10" after 9, 10 is the next number after 1:

0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**2**
0	0	1	1	**3**

Binary Codes

Binary: Count up from 0, but there are only 2 digits:

- 0
- 1

So instead of "reaching 10" after 9, 10 is the next number after 1:

0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**2**
0	0	1	1	**3**
0	1	0	0	**4**

Binary Codes

Binary: Count up from 0, but there are only 2 digits:

- 0
- 1

So instead of "reaching 10" after 9, 10 is the next number after 1:

0	0	0	0	**0**
0	0	0	1	**1**
0	0	1	0	**2**
0	0	1	1	**3**
0	1	0	0	**4**

$\vdots$

String Encodings

String Encodings

- To get letters, we map binary numbers to letters using a lookup table

String Encodings

- To get letters, we map binary numbers to letters using a lookup table
- A sequence of letters (such as a word) is called a *string*

String Encodings

- To get letters, we map binary numbers to letters using a lookup table
- A sequence of letters (such as a word) is called a *string*
- Common encodings are *ASCII* and *Unicode*

String Encodings

- To get letters, we map binary numbers to letters using a lookup table
- A sequence of letters (such as a word) is called a *string*
- Common encodings are *ASCII* and *Unicode*

You can find ASCII tables on *Wikipedia*, or other websites:

100 0001	101	65	41	A
100 0010	102	66	42	B
100 0011	103	67	43	C
100 0100	104	68	44	D
100 0101	105	69	45	E
100 0110	106	70	46	F

image cred: Wikipedia

How a Computer Works

Instruction Set:

Code	Instruction
00000001 A B	Add $A + B \rightarrow A$
00000010 A B	Divide $A/B \rightarrow A$
⋮	⋮
01011011 A	Print $A \rightarrow$ Terminal
⋮	⋮
10100100 A C	Save $A \rightarrow C$ (on hd)

Memory:

Address	Contents
00000000	01101011
00000001	11001100
00000010	10001000
00000011	10101110
⋮	⋮
11111111	00100000

Assembly Language

Assembly Language

Instruction:

0000 1010	0000 0001	0000 0010

Assembly Language

Instruction:

0000 1010	0000 0001	0000 0010
↑	↑	↑
instruction	Mem A	Mem B

Assembly Language

Instruction:

0000 1010	0000 0001	0000 0010
↑	↑	↑
instruction	Mem A	Mem B

Assembly Code:

0000 1010	0000 0001	0000 0010
ADD	A	B

Assembly Language

Instruction:

0000 1010	0000 0001	0000 0010
↑	↑	↑
instruction	Mem A	Mem B

Assembly Code:

0000 1010	0000 0001	0000 0010
ADD	A	B

Program:

```
INPUT   A
ADD     A B
PRINT   C
PRINT   A
```

Assembly Language

Instruction:

0000 1010	0000 0001	0000 0010
↑	↑	↑
instruction	Mem A	Mem B

Assembly Code:

0000 1010	0000 0001	0000 0010
ADD	A	B

Program:

```
INPUT   A
ADD     A B
PRINT   C
PRINT   A
```

https://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-instruction-set-reference-manual-325383.pdf

Concept Check!

Assume that

- `INPUT_I A`: reads an integer input from the keyboard, and stores it in memory address `A`
- `ADD_I A B` adds 2 integers stored in `A` and `B` and stores the result by overwriting `A`
- `PRINT_I A` decodes the contents of the memory address `A` as an integer and prints it to the screen

What does the following snippet of assembly code do?
(Assume `B` contains the binary encoding for the integer 2)

```
INPUT_I   A
ADD_I     A B
PRINT_I   A
```

Solution!

Solution!

```
INPUT_I   A
```
← reads a number from keyboard

Solution!

```
INPUT_I   A      ← reads a number from keyboard
ADD_I     A B    ← adds 2 to the number
```

Solution!

```
INPUT_I   A      ← reads a number from keyboard
ADD_I     A B    ← adds 2 to the number
PRINT_I   A      ← prints the result
```

Solution!

```
INPUT_I   A      ← reads a number from keyboard
ADD_I     A B    ← adds 2 to the number
PRINT_I   A      ← prints the result
```

The code reads an integer from the keyboard, adds 2, and prints the answer

High-Level Languages

High-Level Languages

computer languages:

High-Level Languages

computer languages:	*compiler / interpreter* $\rightarrow$

High-Level Languages

computer languages:	*compiler / interpreter* $\rightarrow$	*assembly code*

High-Level Languages

computer languages: → *compiler / interpreter* → *assembly code* → *assembler* →

High-Level Languages

computer languages: → *compiler / interpreter* → *assembly code* → *assembler* → *binary code*

High-Level Languages

computer languages: *compiler / interpreter* $\rightarrow$ *assembly code* *assembler* $\rightarrow$ *binary code*

High-level computer languages:

High-Level Languages

computer languages: *compiler / interpreter* $\rightarrow$ *assembly code* *assembler* $\rightarrow$ *binary code*

High-level computer languages:

High-Level Languages

computer languages: *compiler / interpreter* $\rightarrow$ *assembly code* *assembler* $\rightarrow$ *binary code*

High-level computer languages:

Formal Languages

Formal Languages

Natural languages:

Formal Languages

Natural languages:

- Capable of quickly expressing complex ideas
- Expressions can carry nuance
- Ambiguous – we infer meaning based on context

Formal Languages

Natural languages:

- Capable of quickly expressing complex ideas
- Expressions can carry nuance
- Ambiguous – we infer meaning based on context

Examples:

- English language
- Sign language
- Latin

Formal Languages

Natural languages:

- Capable of quickly expressing complex ideas
- Expressions can carry nuance
- Ambiguous – we infer meaning based on context

Examples:

- English language
- Sign language
- Latin

Formal languages:

Formal Languages

Natural languages:

- Capable of quickly expressing complex ideas
- Expressions can carry nuance
- Ambiguous – we infer meaning based on context

Examples:

- English language
- Sign language
- Latin

Formal languages:

- Each expression has specific meaning
- Impossible to express nuance
- Absolutely no ambiguity in an expression

Formal Languages

Natural languages:

- Capable of quickly expressing complex ideas
- Expressions can carry nuance
- Ambiguous – we infer meaning based on context

Examples:

- English language
- Sign language
- Latin

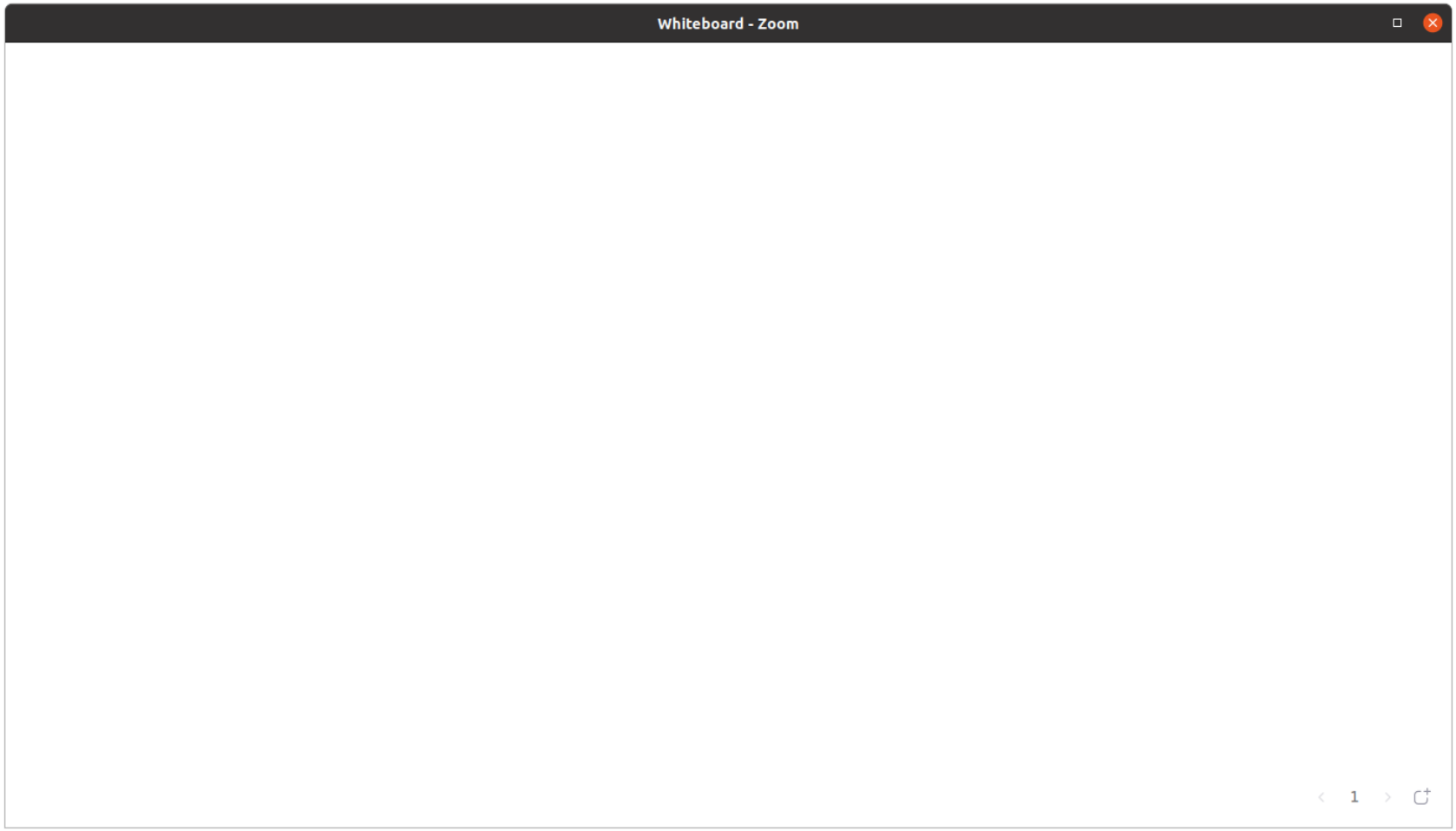
Formal languages:

- Each expression has specific meaning
- Impossible to express nuance
- Absolutely no ambiguity in an expression

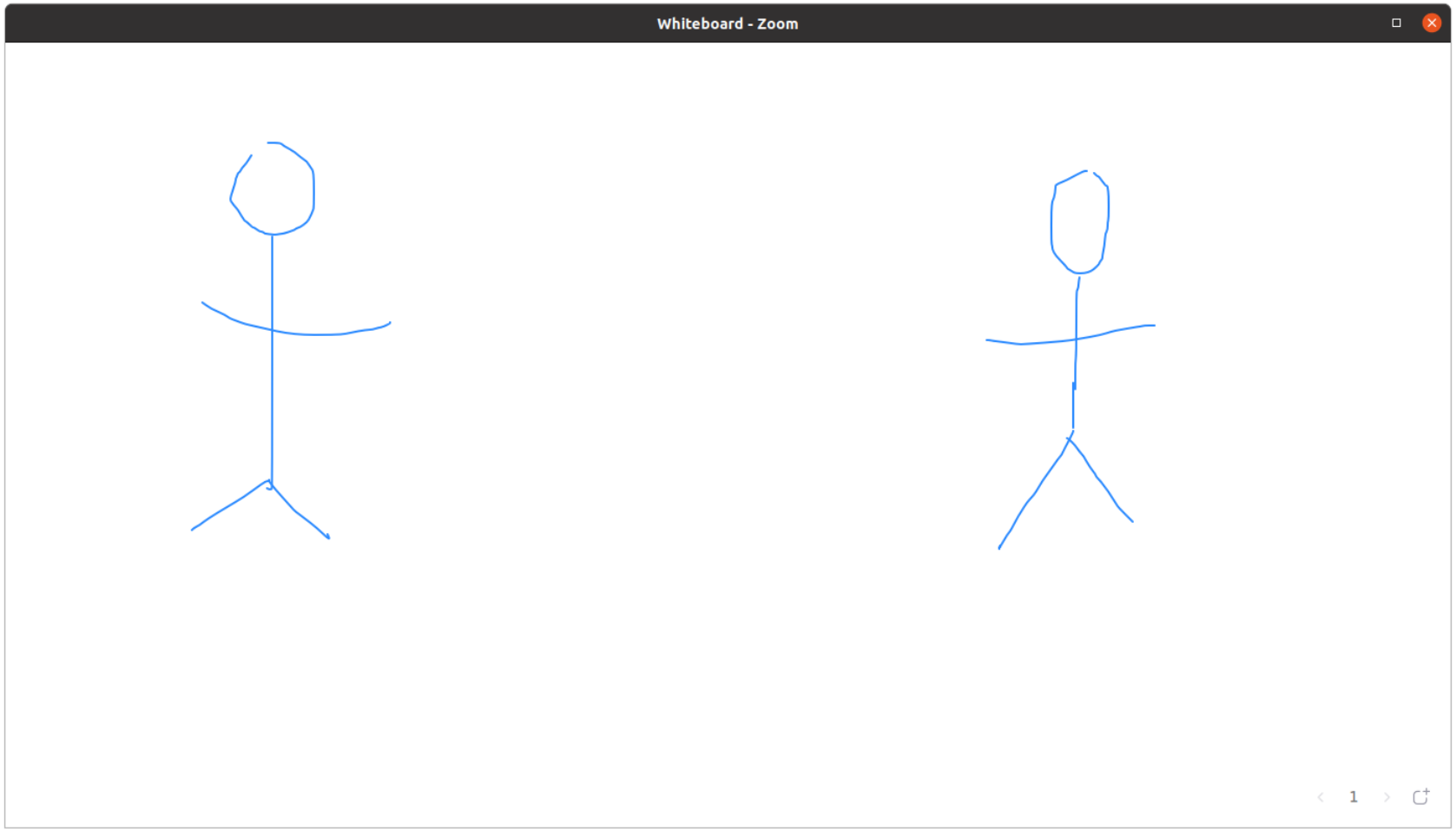
Examples:

- Assembly instructions
- Python code
- Mathematical logic

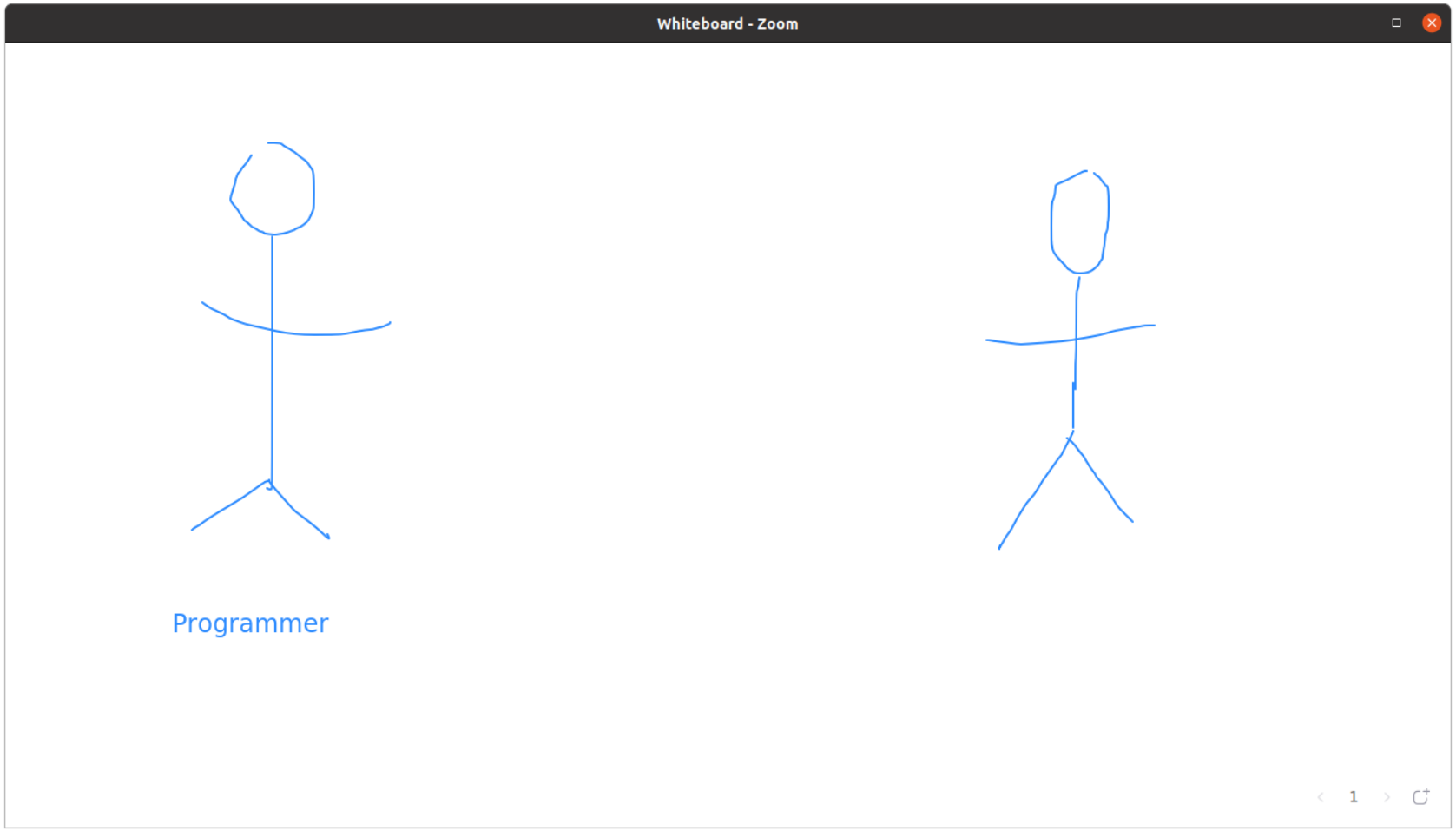
A Programmer's Joke



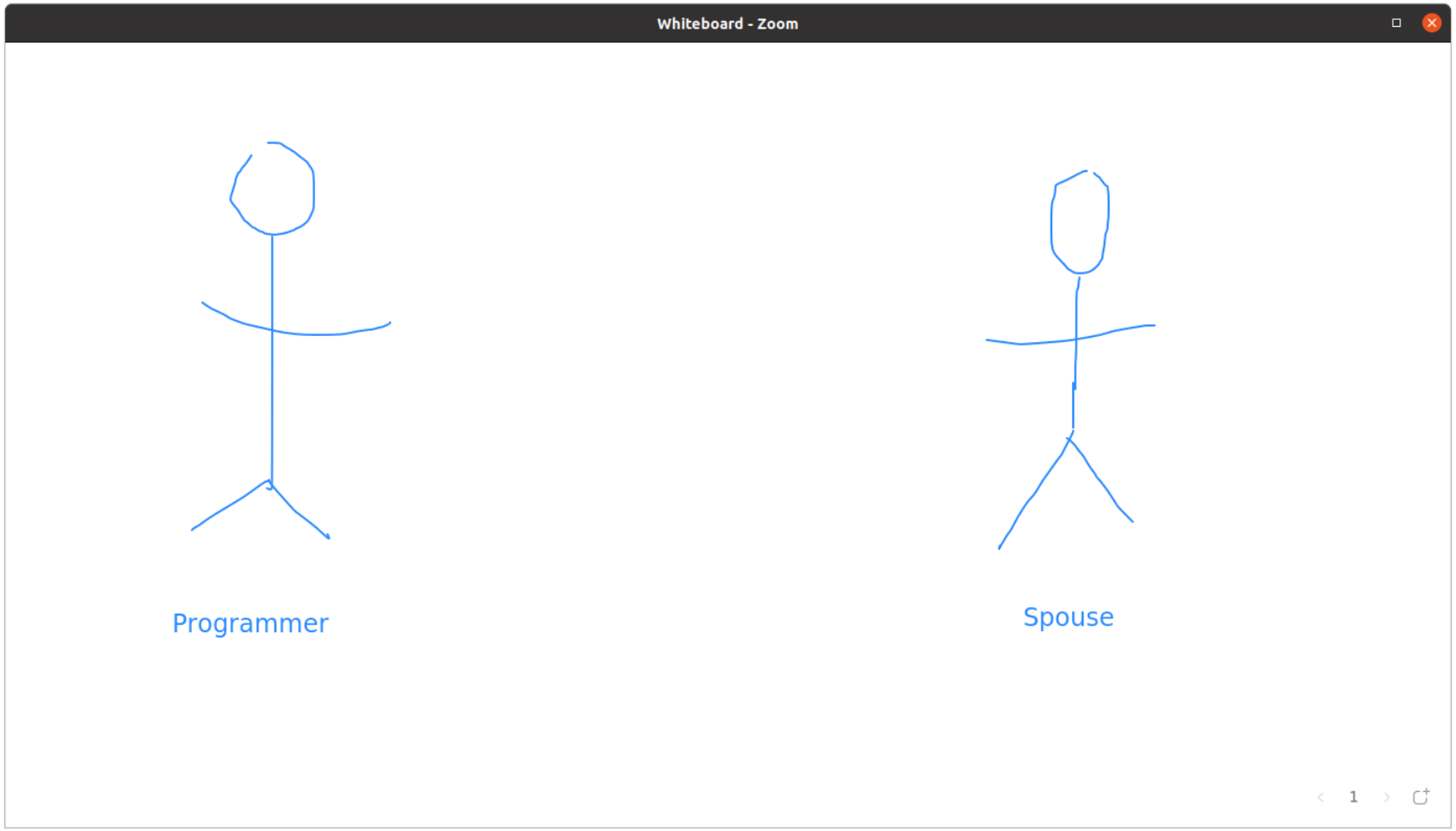
A Programmer's Joke



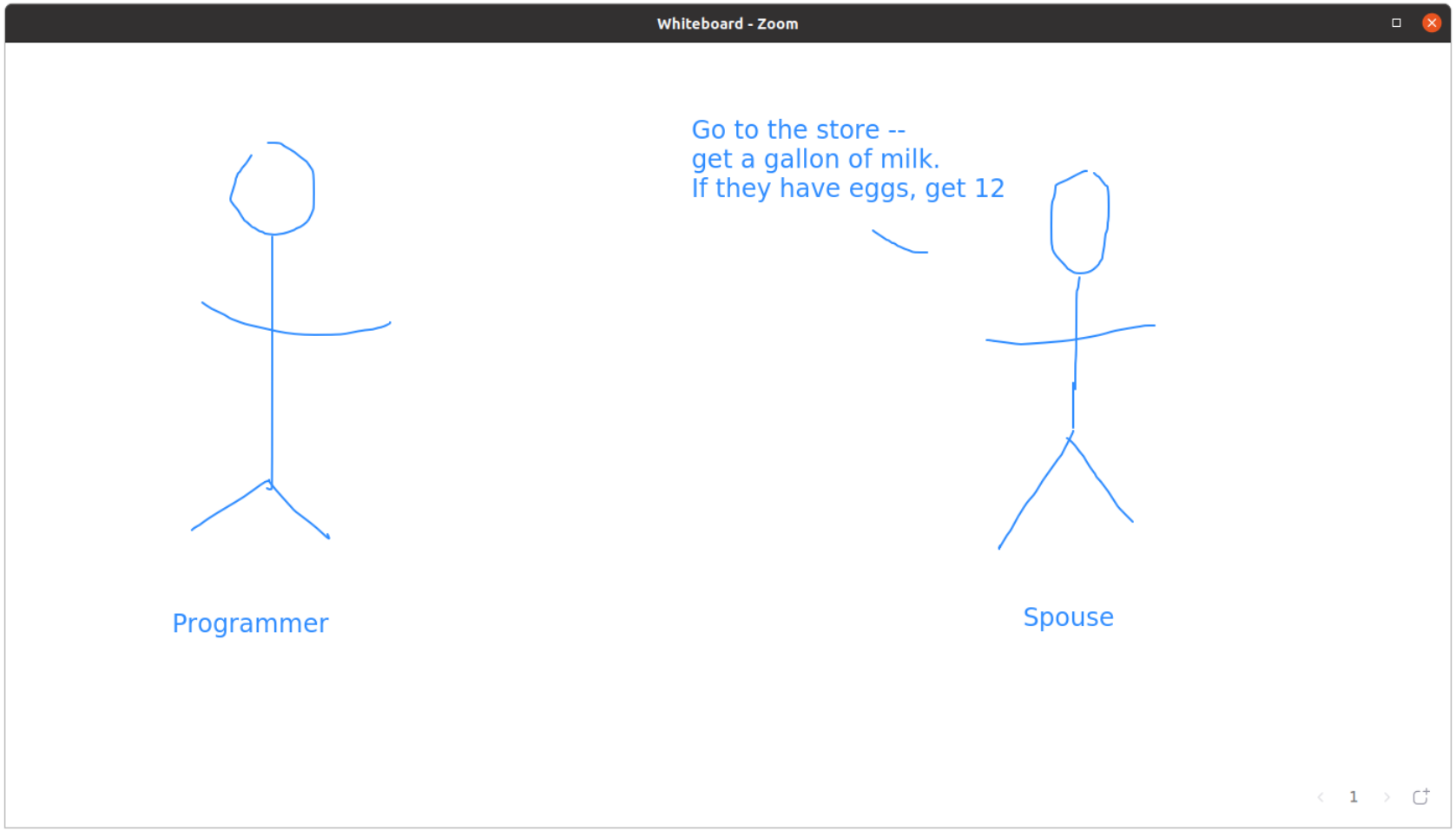
A Programmer's Joke



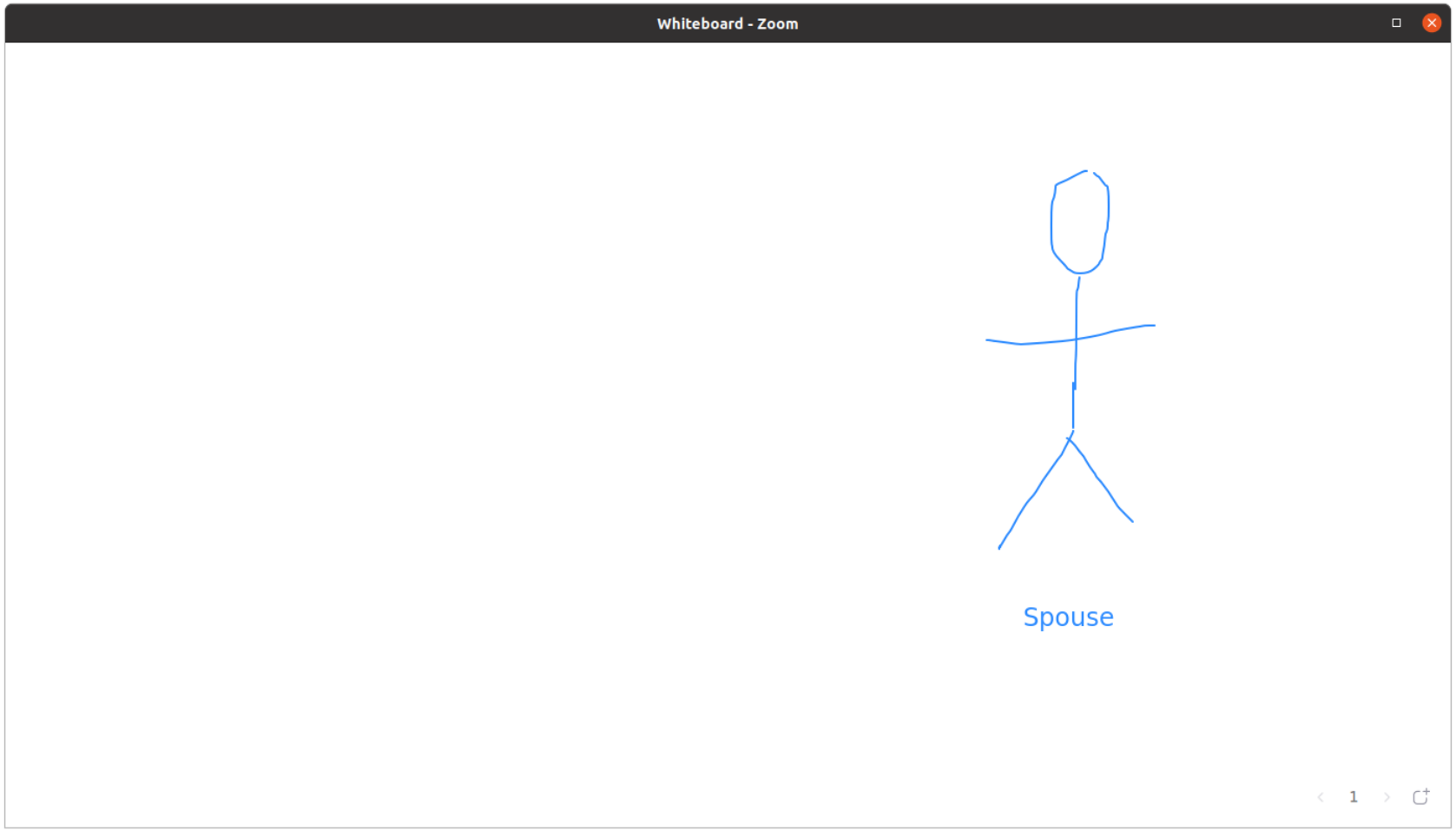
A Programmer's Joke



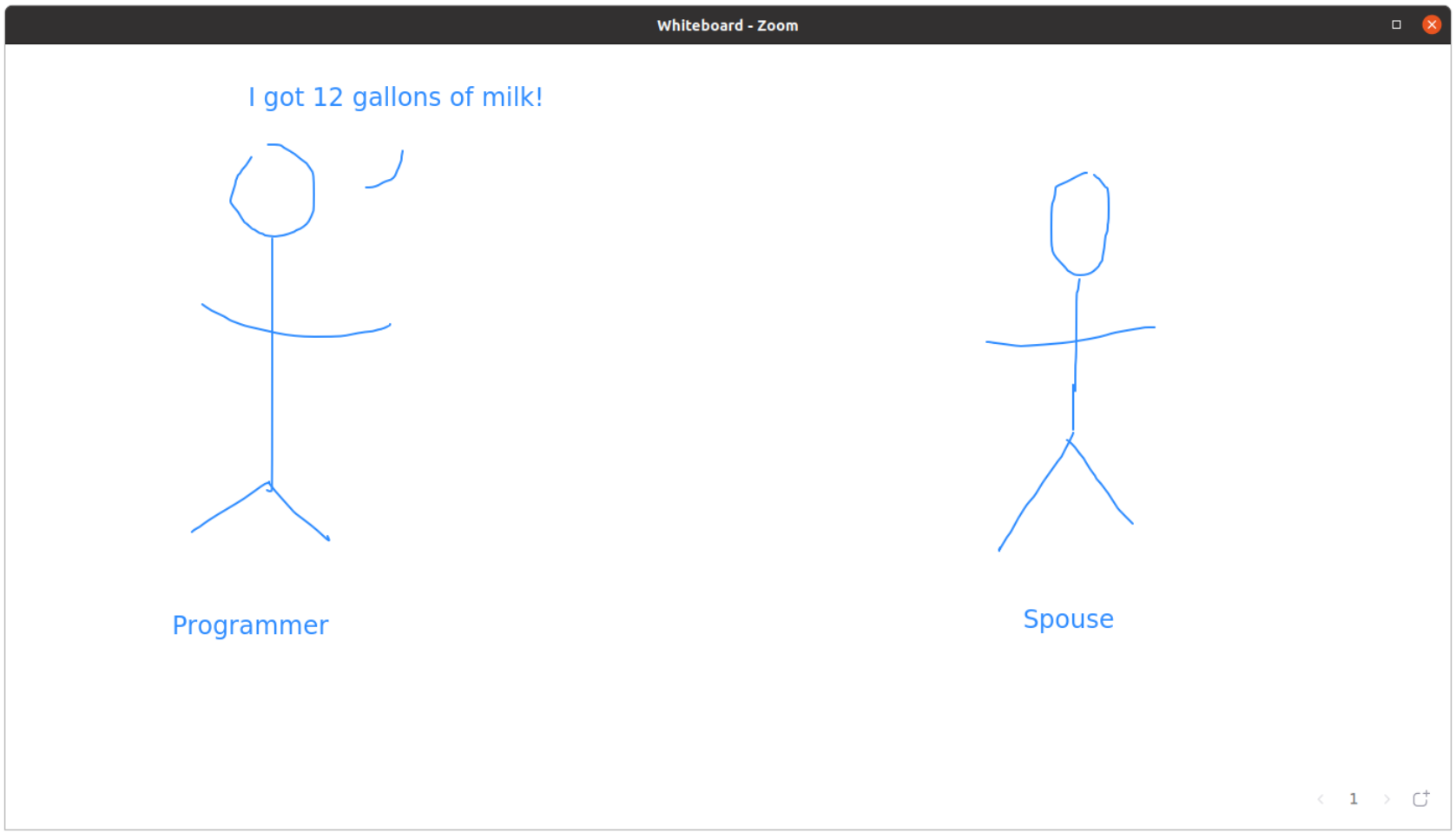
A Programmer's Joke



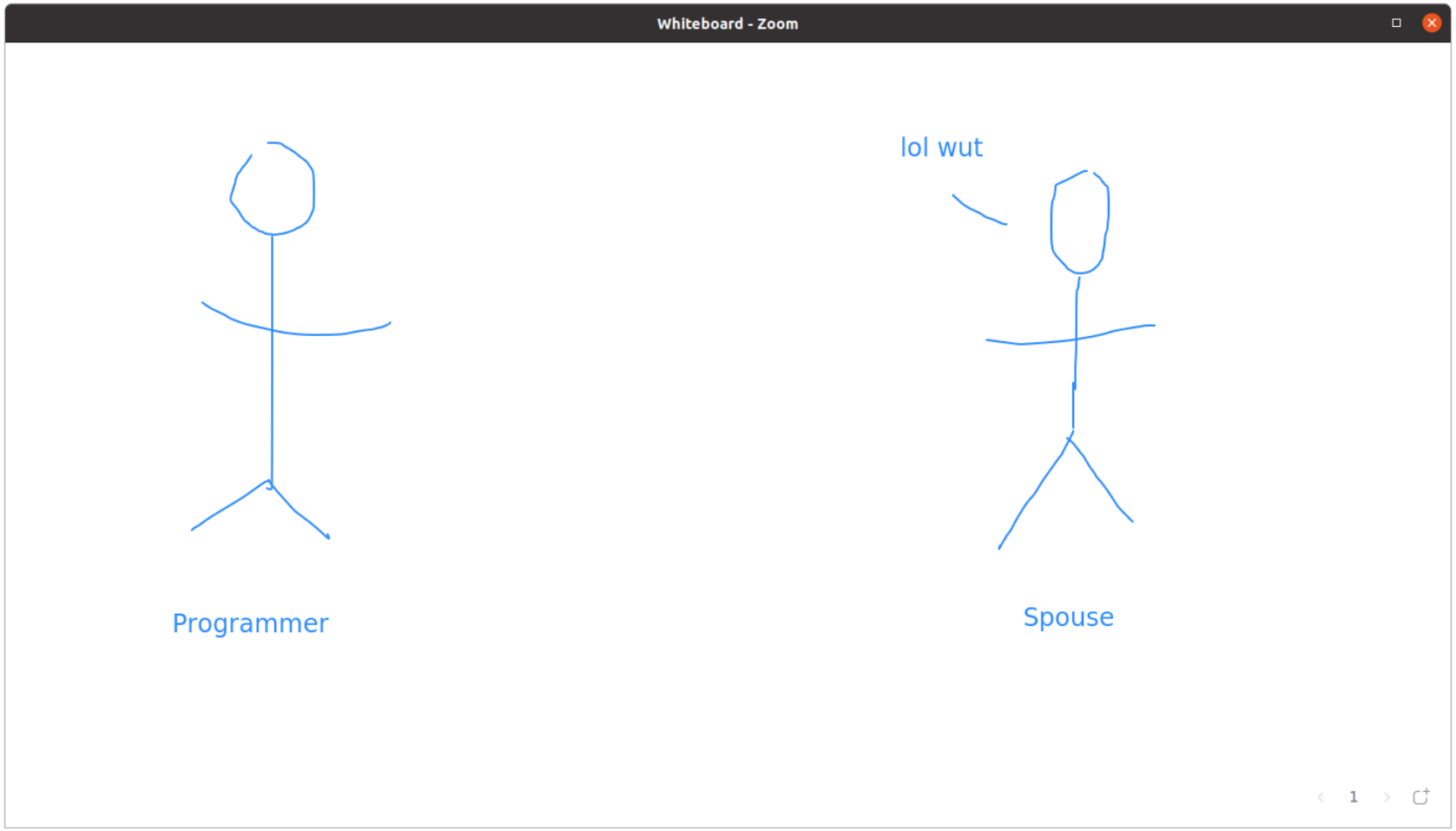
A Programmer's Joke



A Programmer's Joke



A Programmer's Joke



A Programmer's Joke

"Get a gallon of milk. If they have eggs, get 12."

Concept Check!

Which of the following would you write in a formal language?

- An email
- A computer program
- A research paper
- A mathematical proof
- A computer science textbook

Solutions!

Which of the following would you write in a formal language?

Solutions!

Which of the following would you write in a formal language?

- An email **No**

Solutions!

Which of the following would you write in a formal language?

- An email **No**
- A computer program **Yes**

Solutions!

Which of the following would you write in a formal language?

- An email **No**
- A computer program **Yes**
- A research paper **No**

Solutions!

Which of the following would you write in a formal language?

- An email **No**
- A computer program **Yes**
- A research paper **No**
- A mathematical proof **Yes**

Solutions!

Which of the following would you write in a formal language?

- An email **No**
- A computer program **Yes**
- A research paper **No**
- A mathematical proof **Yes**
- A computer science textbook **No**