Lecture 0 - Scratch

Saturday, July 19, 2025 7:24 AM

© 2025 Soham Gurav. These notes are part of my CS50x course journey. Shared for educational purposes only. Do not reproduce or redistribute without permission.

Computer science -> Basically the study of information(How do we represent it & process it) & it is all about problem solving at the end of the day.

Application of ideas from CS to real world problems of interest

Problem solving by computational thinking

What is problem solving ?

Input (The problem to solve) → computer program [ALGORITHM] → Output (The solution)

But how do we represent our input & output in a global & standardized way such that our computer program understands it as well ?

Unary Notation (Base 1) : Numbers are represented by using a single symbol repeated a number of times equal to the value being represented & Zero represented by the absence of any symbol

example : 4 can be written as | | | |
So this is like the basic raw counting on fingers

Binary Notation (Base 2) : Numbers are represented using two symbols : 0 & 1. It is a positional notation system which means instead of having different symbols for each number , we can use the same symbols in varying positions to resemble different numbers.

For base two each digit gets progressively multiplies by said base '**2**'

0 -> Don't use the value
1 -> Use the value

	x8	x4	x2	x1
zero				0
one				1
two			1	0
three			1	1
four		1	0	0

A Binary Digi**t** = A Bit

Like light bulb off = 0
light bulb on = 1

8 bits = 1 byte

Because in electronic devices like our phones , there are many transistors that can be on or off (1 or 0) and are used for information to be stored.

So to write bigger numbers we need more digits that is more bits

So this is counting on fingers but we introduced 'states' that the fingers can be in = the fingers can be in two states : up / down

In a positional notation system , any symbol has base(radix) times the value of the symbol before it

Decimal Notation (Base 10) : This is the standard numeral system we learn
where we have 10 symbols (as base 10) and we use them(0123456789) with a base(radix) pf 10 in positional notation.
So computers are representing information in exactly the same way (binary) as we have been (decimal) since our childhood but only difference is that they have fewer symbols at their disposal.

	x1000	x100	x10	x1	
7				7	7x1
45			4	5	4x10 + 5x1
237		2	3	7	2x100 + 3x10 + 7x1
4567	4	5	6	7	4x1000 + 5x100 + 6x10 + 7x1

An Algorithm is a process / set of rules (basically just code.) followed by the computer to solve certain problems.

Our ideal is to develop an algorithm which is correct (obviously) & efficient

Correct Algorithm : An algorithm which solves the problem is a correct algorithm

Efficient Algorithm : An algorithm which is faster & shorter is a more efficient algorithm

Example : An Algorithm for finding a particular name in the phone book :
Pseudocode =

1. Pick up phone book
2. Open to middle of phone book
3. Look at page
4. If person is on page
5. Call Person
6. Else if person is earlier in book
7. Open to middle of left half of book.
8. Go back to line 3
9. Else if person is later in book
10. Open to middle of right half of book
11. Go back to line 3
12. Else
13. Quit

This is called a loop as it enforces the running of code over & over again until the problem is solved. Here the loop is used to divide each halved section of the book into halves again.

These are called conditionals which provide conditions as boolean expressions which if are true then a function is carried out otherwise another function is carried out/next line of code is read.
5 is done if 4 is true , if not then 5 is skipped and 6 is read.

These are verbs/actions which we tell the computer to do (often represented as functions)

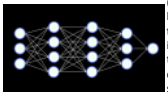
How to make an Algorithm

1. Write how to solve the problem in English
2. Write How to solve the problem in Pseudocode (Basically in english but very precisely & succinctly)
3. Write the steps in code (i.e programming language)

Hence what we do is train it by giving it lots & lots of data input , letting it figure out what it should say
and it boils down to probability & statistics otherwise known as Large Language Models.

and so on

Which are implemented by Neural Networks
(Inspired by biology as we have neurons that transmit electrical signals in the same way transmitting the input data and formulizing statistically the most probabilistic answer for the final neuron to output



Artificial Intelligence

We might think something like a chatbot can be made by using conditionals like
if user says hello then
say hello
if else user says goodbye then
say goodbye
but this could escalate very quickly as there are impossible number of questions a user could ask hence explicitly writing every question as a programmer is not at all convenient.

If computers intake binary numerals as numbers then how do we process alphabets ?
We assign each alphabet & symbol(',',/,;) its own unique number - that is the only solution as of now & it is called ASCII(American Standard Code for Information Interchange) & it uses one byte for each letter.
A = 65 , B = 66 & so on a = 97 , b = 98 & so on.
So if someone sends us a message = Hi ! then our phone receives it as binary = 01001000 01101001 00 100001
which it converts to numbers = 72 105 33 and then into alphabets & symbols through ASCII = Hi ! , This message is of 3 bytes .

But 1 Byte means we can go up to 255 Numbers and if we assign letters & symbols to each of these 255 numbers it will be enoug h for the English language however for some other languages more characters are required hence more numbers are required to assign to them henc e computers even use 16 bit ie 2 bytes for one character when writing in some languages or even 32 bits (Can represent 4 billion numbers so that many characters so some numbers are assigned to emojis !).
Hence ASCII was left behind and most computers now use UNICODE which is like a superset of ASCII and it has more symbols (fro m various languages) all assigned a number as one character can be 8 to 32 bits.

Then how do we computers represent colors ? Computers use a system called RGB (Red Green Blue) in which each pixel on the screen is assigned an RGB code because each pixel has 3 bars (Red Green Blue) and each of these bars have 256 values (1 byte) ranging from 0 - 255 and the combination of these values gives us a color by mixing the proportions of red green & blue . for example , 0 0 0 will give the pixel(3 bytes) a black color. This was there are 16.7 million color combinations possible.

So a picture of size 1 mb means it has 1 million bytes stored in it so around 1million/3 pixels make up that picture

So if computers take binary input and give numbers , letters and colors as output then
how does the computer know if a particular binary input is supposed to be output as a letter (by ASCII) or as a number ?

A computer determines how to treat the binary values based on its context
the context being what program is being used.
When programming a software, developers specify data types (integers, characters, strings) which tell the computer how to interpret the binary data.

Problem Set 0

Going to make an interactive game in scratch
Requirements as put forth by cs50x =

- atleast two sprites
- atleast three scripts in total
- atleast one conditional
- atleast one loop
- atleast one variable
- atleast one custom block (which takes atleast one input)
- 12+ puzzle pieces total

© 2025 Soham Gurav. These notes are part of my CS50x course journey. Shared for educational purposes only. Do not reproduce or redistribute without permission.

It will be a quiz game

First backtrack : F1 (hans zimmer F1 soundtrack) → Backtrack 2 : Choose one : Driver recruit Mechanic Recruit → Backtrack 3 : You have chosen to become a "__" Pass the test & get to choose your team !

Backtrack 4 :

Q1.Driver = How long is an F1 circuit in kms?
ans = 305
Q1.Mechanic = which force is implemented for the better grip of the car ?
ans = downforce

Q.2.Driver = What is the DRS System ? (Write full form)
ans = drag reduction system
Q>3.Mechanic = What type of tyre wears out the quickest ?
ans = soft

Q.3.Driver = On minimum how many different types of tyres is an F1 driver required to use during a race ?
ans = 2
Q3.Mechanic = According to FIA rules what is the maximum fuel consumption limit during a race?(in kg)
ans = 70

You have Passed the "__" test → Choose your team = Redbull Oracle Mercedes Petronas Mclaren Ferrari → Congratulations! You are now a F1 Driver (Not Literally!)

Conversion from Decimal to Any Radix :

1. Let a = given decimal number
2. write 1 under the largest number 'b' in radix table which is less than or equal to a.
3. subtract b from a to get a - b.
4. write 1 under the largest number 'c' less than or equal to a - b
5. subtract c from (a - b) to get (a - b - c)
6. Keep doing this until zero is achieved.

eg. Decimal to Binary for the number 75 :
Binary radix table should only be drawn till 128 as 128>75>64

128	64	32	16	8	4	2	1
	1	0	0	1	0	1	1

1. Largest no. less than or = to 75 is 68 so we write 1 under 64 then 75 - 64 = 11
2. Largest no. less than or = to 11 is 8 so we write 1 under 8 and 11 - 8 = 3
3. Largest no. less than or = to 3 is 2 so we write 1 under 2 and 3 - 2 = 1
4. Largest no. less than or = to 1 is 1 so we write 1 under 1 and 1 - 1 = 0
5. Fill the gaps between with 0