Structure of an HTML website = html + <tab>

```
<!DOCTYPE html>

<html>

<head>

          <meta charset="utf-8">

          <title></title>

</head>

<body>


</body>

</html>
```

Head
What you don’t see on the browser

Body

What you see in the browser

Website Elements

HTML - The structure of an application, where you define heading, subheading, paragraph, bullet and numbering, images etc..

CSS - It takes care of the styling in a website, for example the colour, background colour, font, paragraphing and layout

Javascript - What makes our website dynamic.. Answer to the question when xx happens.. Do xxx -> eg: When the button is clicked.. Open a new page.., when the page is loaded.. Show something

Html + tab - structure

<head></head> - things that you dont’ see -> configuration

<title></title> - title in the app bar

<body></body> whatever in the white area

All these elements will enter directly

<h1></h1> - Heading / title

<h2></h2> until <h6></h6> - subheading / subtitle

<p></p> - paragraph

Hr - horizontal ruler

Lorem + tab -> A long text in case you don’t long paragraph

All these elements will not create new line directly

<strong></strong> - bold

<em></em> - italic

Img + tab -> to show image (get the link or locally, if locally, it has to be in the same folder)

A + tab -> to link to other website

Is you want to enter add <br> - break

Keyboard Shortcut

ctrl + x / cmd + x cut

ctrl + c / cmd c - copy
Ctrl + v - cmd v - paste

Ctrl +z / cmd z- undo

Ctrl + y / cmd + shift + z - redo (undo my undo)

Ctrl + s -> / cmd + z - save my code

Alt + tab -> / option + tab to change my window

In browser i use ctrl + r - cmd + r -> refresh my browser

List and numbering

Comments (ignoring the code)

<!-- -->

In sublime you highlight and ctrl + / or command + /

Base 2 (Binary)

	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Base 10	128	64	32	16	8	4	2	1
65		1	0	0	0	0	0	1
74		1	0	0	1	0	1	0
123		1	1	1	1	0	1	1
11					1	0	1	1

1) You look for the biggest power of 2 from the number, put one there
2) You remove/minu the biggest power of 2 from the number , eg 74-64 = 10
3) Repeat step 1 until the number become 0

Base 16

1,2,3,4,5,6,7,8,9,a,b,c,d,e,f

Base 10	16^1	16^0
123	7	b
193	c	1

1) You divide the number with 16 (123/16) = eg: 7.xxxxxx . The first number will be 7
2) If the number is bigger than 9 refer to the list
3) Then you will look for the difference between the number and answer in a * 16 = 123 - (16*7) = 11
4) If the number is bigger than 9 refer to the list, look for 11 in the list , 11 = b

Exercise

Base 10 to Base 2

23 - 00010111

200 - 11001000

89 - 01011001

Base 2 to Base 10

11010010 = 128 + 64 +16 +2 = 210

01100101 = 64 + 32 + 4 + 1 = 101

00110111 = 32 + 16 + 7 = 55

Base 10 to Base 16

233 = e9

101 = 65

43 = 2b

255 - ff

0 - 00

Base 16 to Base 10

a1 = (16 * 10) + 1 = 161

35 = (16 * 3) + 5 = 53

7e = 126

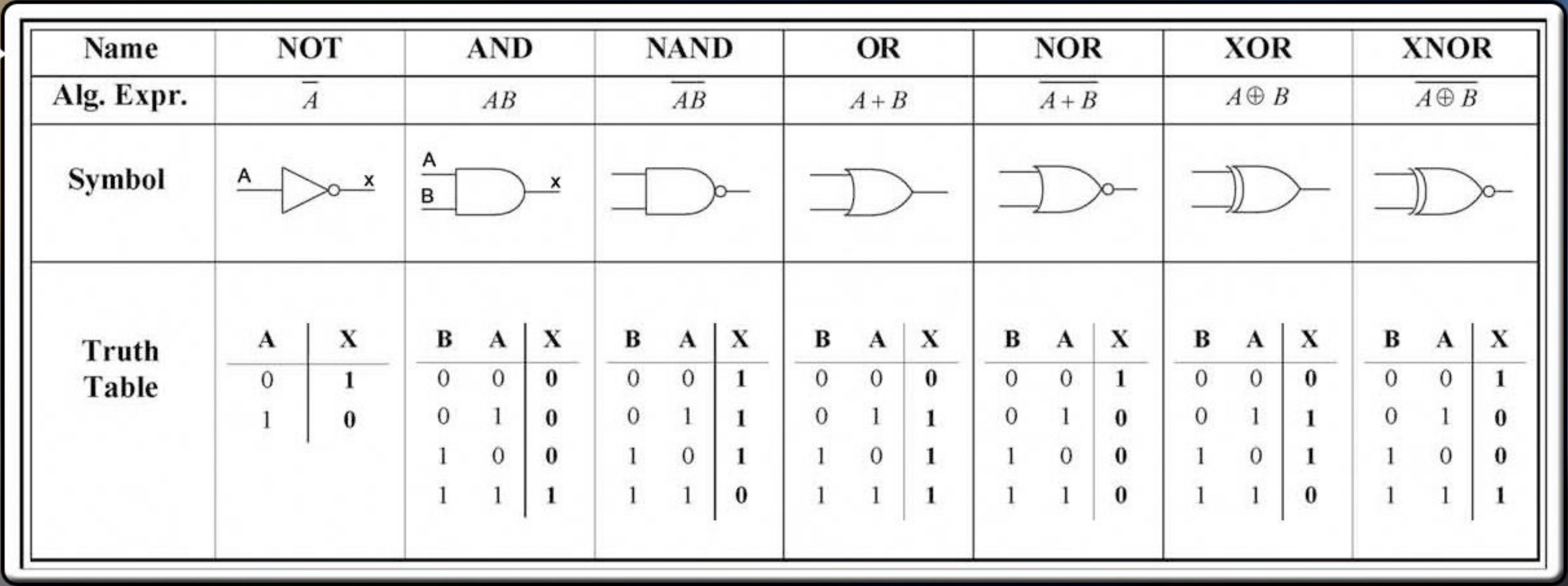
AND Logic

tired	hungry	tired && hungry
false	false	false
false	true	false
true	false	false
true	true	true

OR Logic

tired	hungry	tired || hungry
false	false	false
false	true	true
true	false	true
true	true	true

Logic Gates



Name	NOT	AND	NAND	OR	NOR	XOR	XNOR
Alg. Expr.	$\overline{A}$	AB	$\overline{AB}$	$A+B$	$\overline{A+B}$	$A\oplus B$	$\overline{A\oplus B}$
Symbol	A X	A B X					
Truth Table	A X 0 1 1 0	B A X 0 0 0 0 1 0 1 0 0 1 1 1	B A X 0 0 1 0 1 1 1 0 1 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 1	B A X 0 0 1 0 1 0 1 0 0 1 1 0	B A X 0 0 0 0 1 1 1 0 1 1 1 0	B A X 0 0 1 0 1 0 1 0 0 1 1 1

i	j	expectation	
0	0	#	No need to repeat
1	0 1	##	Repeat 2
2	0 1 2	###	Repeat 3 times
3	0 1 2 3	####	Repeat 4 times

Variable = i , constant = lines - 6

i	j	expectation	lines-i-1
0	0 1 2 3 4	#####	5 6-1-0
1	0 1 2 3	####	4 6-1-1
2	0 1 2	###	3

	0	1	2	3	4
0	#	0	#	0	#
1	0	#	0	#	0
2	#	0	#	0	#
3	0	#	0	#	0
4	#	0	#	0	#

When i is even and j is even ……
When i is even and j is odd ……
When i is odd and j is even ….
When i is odd and j is odd …..

RANDOM NUMBER BETWEEN 0 1 AND 2

	0	1	2
Math.floor(Math.random()*3)	0-0.9999999999	1-1.99999999	2-2.999999999
Math.round(Math.floor(Math.random()*2))	0- 0.499999999	0.5-1.49999999	1.5-1.999999999
	0.5 / 2 = 1/4	1/2	0.5 / 2 = 1/4

We cannot use Math.round to generate a random number because the first and the last number will have lesser chance the appear than the number in the middle..

[1,2,3,4,5,6,7,8,9,10] - first round -> bigger.. Look at the top half (5 + 10) / 2 = 8

[1,2,3,4,5,6,7,8,9,10] smaller - > look for at the bottom half - 5 + 8 / 2 = 7

[1,2,3,4,5,6,7,8,9,10] - smaller - > look at the bottom half - 5 + 7 / 2 = 6

[1,2,3,4,5,6,7,8,9,10] -> Found the number

If i do using binary search i will find the number in 4 rounds, if not it will be in 6 round (Linear Search)

Number is 6…

1 - 100

63 - With linear search i will find it within 63 tries

50 -> 1) lower than the searched number (50 + 100 / 2) = 75

75 -> 2) higher than the searched number (50 + 75) /2 = 63

With binary search I get it within 3 tries..

32 -> with linear search I will get it within 32 tries , within 3 tries

[0 ...100]

50 -> 1)

The number is higher than the searched number (50 + 0 / 2) =25

[25 ...50]

25 -> 2) The number is lower than the search number (25 +50 / 2) =

[25 ...38]

38 -3 The number is higher than the searched number (25 + 38 / 2) = 32 ,

Every time, the area of search is divided into 2..

We say that binary search is a **O(log(n))** algorithm - > Let say that you need to find a number between 1 and 100, your max try will be: .. (confirm with you)