

Page 2 / 49

Exit Full Screen

JavaScript Basics

functionality  
Behaviour

HTML → skeleton  
CSS → appearance

Instructor: Love Babbar



A diagram showing a rectangular box. To the right of the box, the text 'By Image' is written. Below the box, the text 'AED' is written. An arrow points from the box towards the 'AED' text.

Page 4 / 49

Press Esc to exit full screen

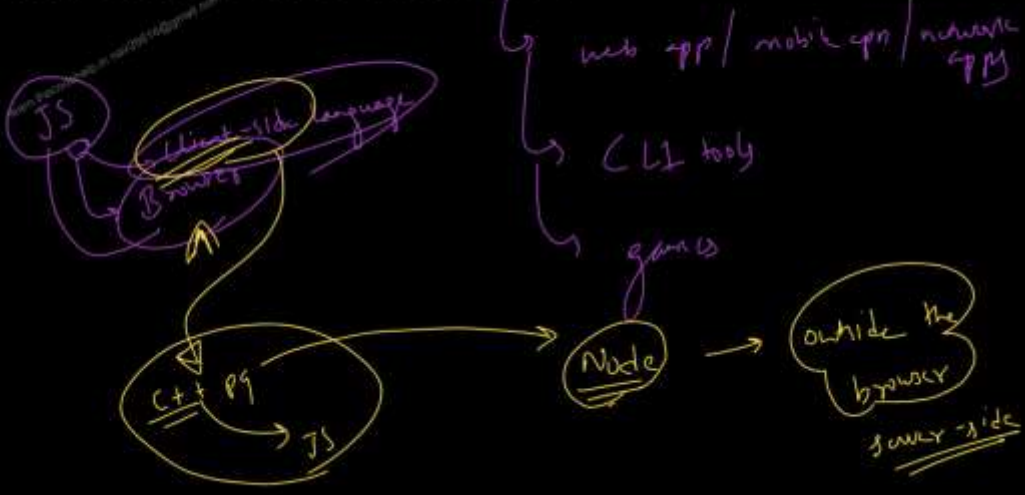
What can we do with JS ?

web app / mobile app / network app  
CLI tool  
games

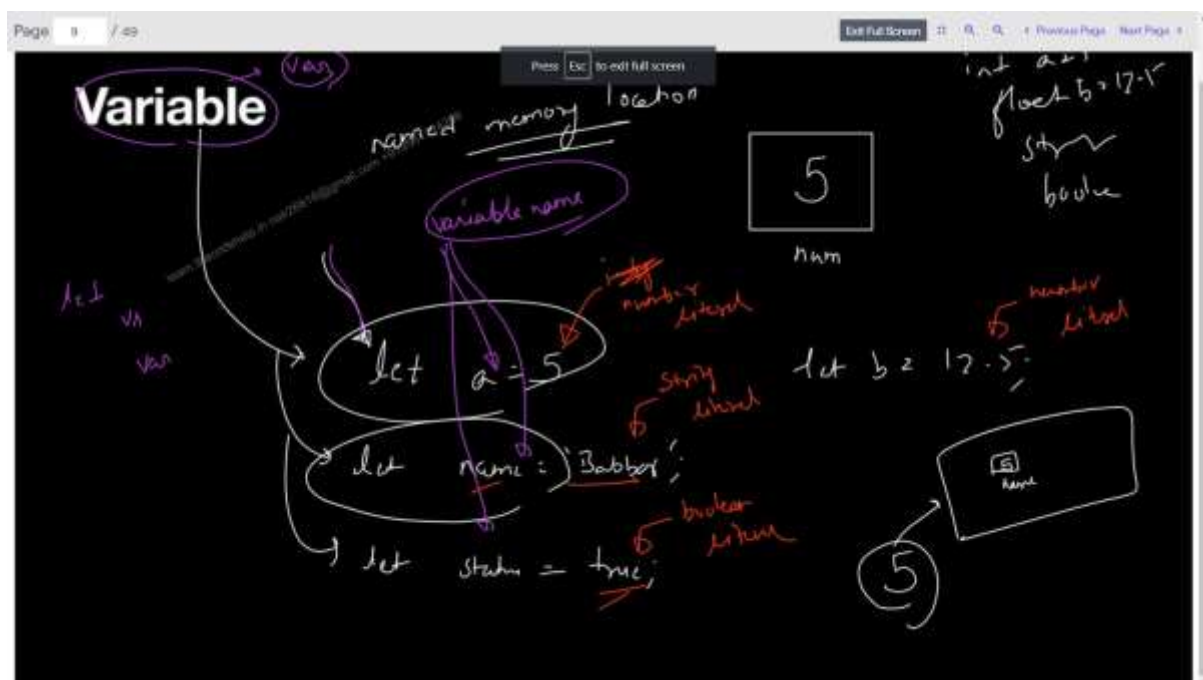
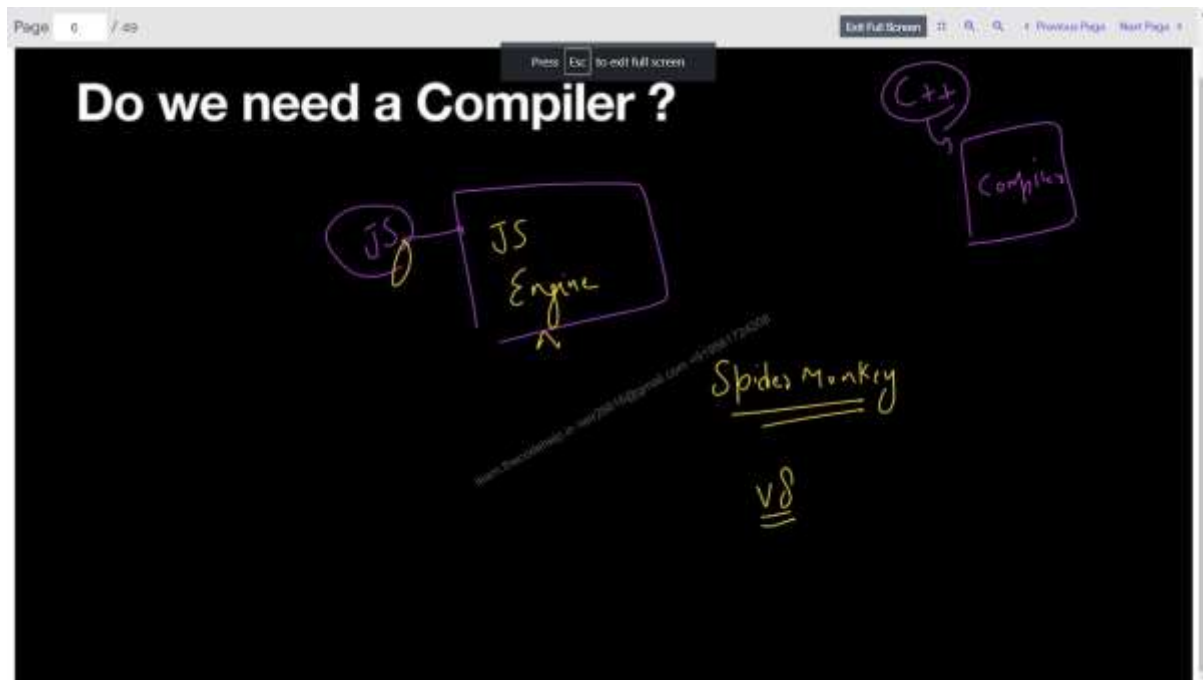
JS → client-side language  
Browser

C++ PG → JS

Node → outside the browser  
server-side



A diagram illustrating the scope of JavaScript. On the left, 'JS' is circled and labeled 'client-side language' and 'Browser'. An arrow points from this to a circle containing 'C++ PG' and 'JS', with an arrow from 'C++ PG' to 'JS'. From this circle, an arrow points to a circle labeled 'Node', which is labeled 'games'. From 'Node', an arrow points to a cloud-like shape labeled 'outside the browser' and 'server-side'.



Page 10 / 49

scope

let

let a = 5

let a = 1

re-declaration

let a = 5

let

if (true) {

let a = 5

console.log(a)

}

console.log(a)

Page 11 / 49

variable naming :-

Rules

- cannot be a reserved keyword (if, let, var etc)
- meaningful
- cannot start with a no
- cannot contain space or '-'
- camel case

first Name

age of Student

love Bhabha

rahul Sharma

anur Kumar Singh

let let = 5

let name = 'love'

if ( )

let if = 5

let a = 5

Page 13 / 59

# Constants

const a = 5;

a = 6;

7

Page 14 / 59

# Primitive Types

- String → 'Love Babbar' → sequence of characters
- Number → 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 441, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 472, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 490, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516, 517, 518, 519, 520, 521, 522, 523, 524, 525, 526, 527, 528, 529, 530, 531, 532, 533, 534, 535, 536, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553, 554, 555, 556, 557, 558, 559, 560, 561, 562, 563, 564, 565, 566, 567, 568, 569, 570, 571, 572, 573, 574, 575, 576, 577, 578, 579, 580, 581, 582, 583, 584, 585, 586, 587, 588, 589, 590, 591, 592, 593, 594, 595, 596, 597, 598, 599, 600, 601, 602, 603, 604, 605, 606, 607, 608, 609, 610, 611, 612, 613, 614, 615, 616, 617, 618, 619, 620, 621, 622, 623, 624, 625, 626, 627, 628, 629, 630, 631, 632, 633, 634, 635, 636, 637, 638, 639, 640, 641, 642, 643, 644, 645, 646, 647, 648, 649, 650, 651, 652, 653, 654, 655, 656, 657, 658, 659, 660, 661, 662, 663, 664, 665, 666, 667, 668, 669, 670, 671, 672, 673, 674, 675, 676, 677, 678, 679, 680, 681, 682, 683, 684, 685, 686, 687, 688, 689, 690, 691, 692, 693, 694, 695, 696, 697, 698, 699, 700, 701, 702, 703, 704, 705, 706, 707, 708, 709, 710, 711, 712, 713, 714, 715, 716, 717, 718, 719, 720, 721, 722, 723, 724, 725, 726, 727, 728, 729, 730, 731, 732, 733, 734, 735, 736, 737, 738, 739, 740, 741, 742, 743, 744, 745, 746, 747, 748, 749, 750, 751, 752, 753, 754, 755, 756, 757, 758, 759, 760, 761, 762, 763, 764, 765, 766, 767, 768, 769, 770, 771, 772, 773, 774, 775, 776, 777, 778, 779, 780, 781, 782, 783, 784, 785, 786, 787, 788, 789, 790, 791, 792, 793, 794, 795, 796, 797, 798, 799, 800, 801, 802, 803, 804, 805, 806, 807, 808, 809, 810, 811, 812, 813, 814, 815, 816, 817, 818, 819, 820, 821, 822, 823, 824, 825, 826, 827, 828, 829, 830, 831, 832, 833, 834, 835, 836, 837, 838, 839, 840, 841, 842, 843, 844, 845, 846, 847, 848, 849, 850, 851, 852, 853, 854, 855, 856, 857, 858, 859, 860, 861, 862, 863, 864, 865, 866, 867, 868, 869, 870, 871, 872, 873, 874, 875, 876, 877, 878, 879, 880, 881, 882, 883, 884, 885, 886, 887, 888, 889, 890, 891, 892, 893, 894, 895, 896, 897, 898, 899, 900, 901, 902, 903, 904, 905, 906, 907, 908, 909, 910, 911, 912, 913, 914, 915, 916, 917, 918, 919, 920, 921, 922, 923, 924, 925, 926, 927, 928, 929, 930, 931, 932, 933, 934, 935, 936, 937, 938, 939, 940, 941, 942, 943, 944, 945, 946, 947, 948, 949, 950, 951, 952, 953, 954, 955, 956, 957, 958, 959, 960, 961, 962, 963, 964, 965, 966, 967, 968, 969, 970, 971, 972, 973, 974, 975, 976, 977, 978, 979, 980, 981, 982, 983, 984, 985, 986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000
- Boolean → true or false
- Undefined
- Null

let a;

console.log(a);

undefined

Page 15 / 49

# Dynamic Typing

C++/Java  
 int a = 5;  
 a = "Babbar";

let a = 5;  
 a = "Babbar";

Page 16 / 49

# Reference Types

- Objects
- Arrays
- Functions

let person = {  
 firstName: 'John',  
 age: 24;  
 };

Person  
 name  
 age

dot notation  
 person.age

bracket notation  
 person[age]

Why?

Page 17 / 49

Arrays - D's used to contain a list of items

let names = ['love', 'Rahul', 'Sangam']

Access  
↓  
index

names[0] names[1] names[2]

names[3] → ?

names[3] = "Raman"

names[1] = 2

1995  
2005

Page 18 / 49

# Operators

- Arithmetic ✓
- Assignment ✓
- Comparison ✓
- Bitwise ✓
- Logical ✓



# Arithmetic

$+$   
 $-$   
 $*$   
 $/$

$\text{let } a = 2$   
 $\text{let } b = 3$   
 $\text{console.log}(2 \times 3)$   
 $8 \div 2^3 = 8$

$\text{let } a = 1$   
 $\text{let } b = 2$   
 $\text{let } c = a + b$   
 $\text{console.log}(c)$  → 3

$2 \times 4 = 2^4 = 16$

$\text{let } a = 12$   
 $\text{let } b = 5$   
 $\text{console.log}(a \div b)$  → 2

$2 \sqrt{5} \begin{matrix} 2 \\ 4 \\ 1 \end{matrix}$   
 $5 \div 2 = 1$

$12 \div 5 = 2$

$\text{remainder op}$   
 $2 \times 4 \rightarrow 2 \times 4$

## Pre/Post Inc/dec Operator

$++n$  → pre-increment  
 first → increment the value  
 second → use the value

$n++$  → post-increment operator  
 first → use the value  
 second → increment the value

$\text{let } n = 10$   
 $\text{console.log}(++n)$  → 11

$\text{let } a = 6$   
 $\text{console.log}(a++)$  → 6  
 $\text{console.log}(a)$  → 7

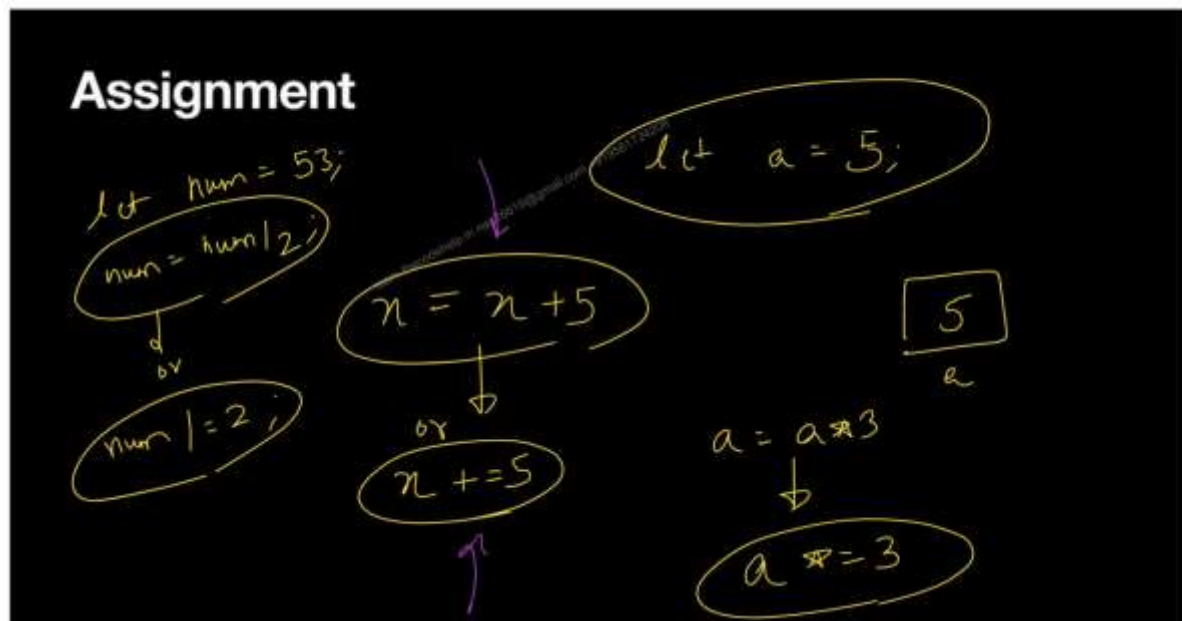
$\text{let } a = 6$   
 $\text{console.log}(a++)$  → 6  
 $\text{console.log}(a)$  → 7

$\text{let } a = 6$   
 $\text{console.log}(a++)$  → 6  
 $\text{console.log}(a)$  → 7

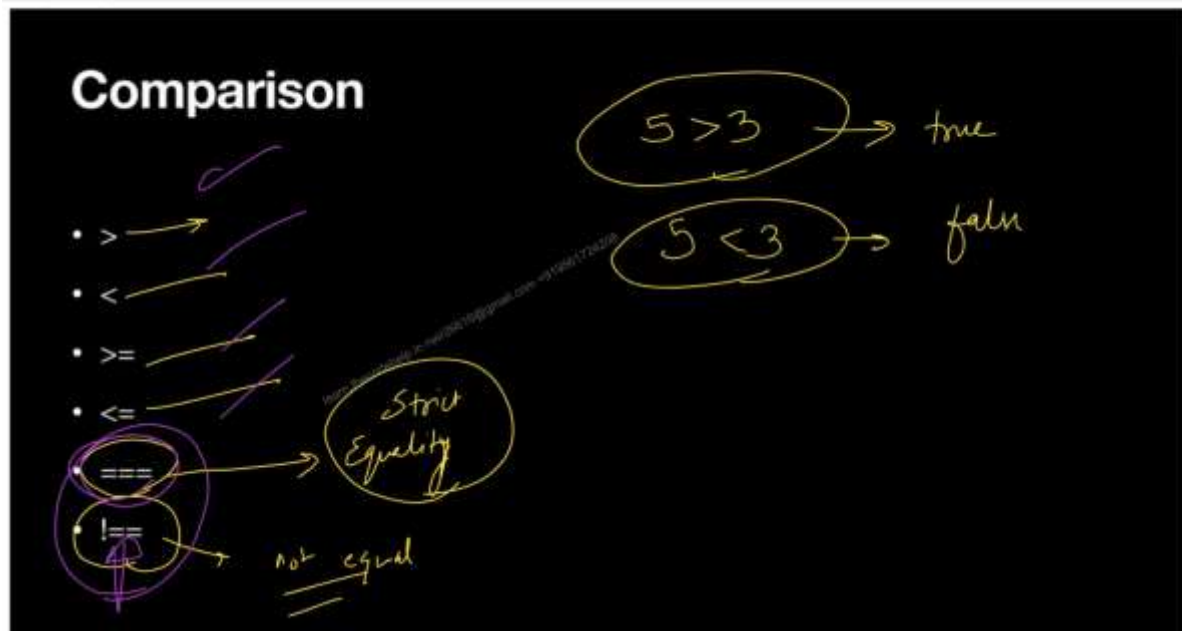




## Assignment

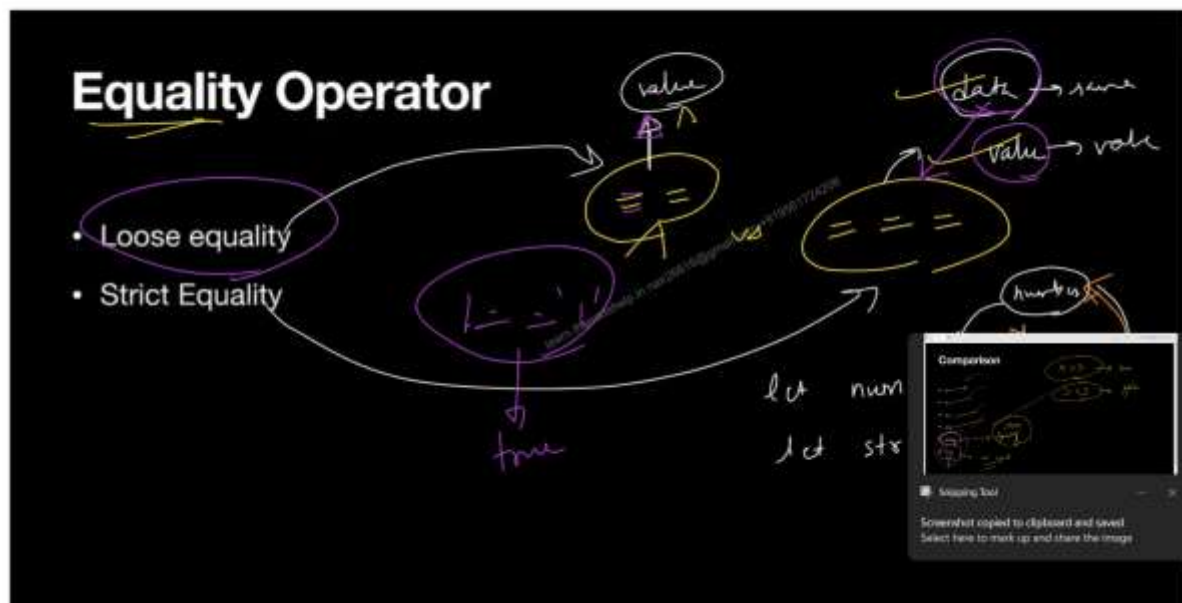


## Comparison

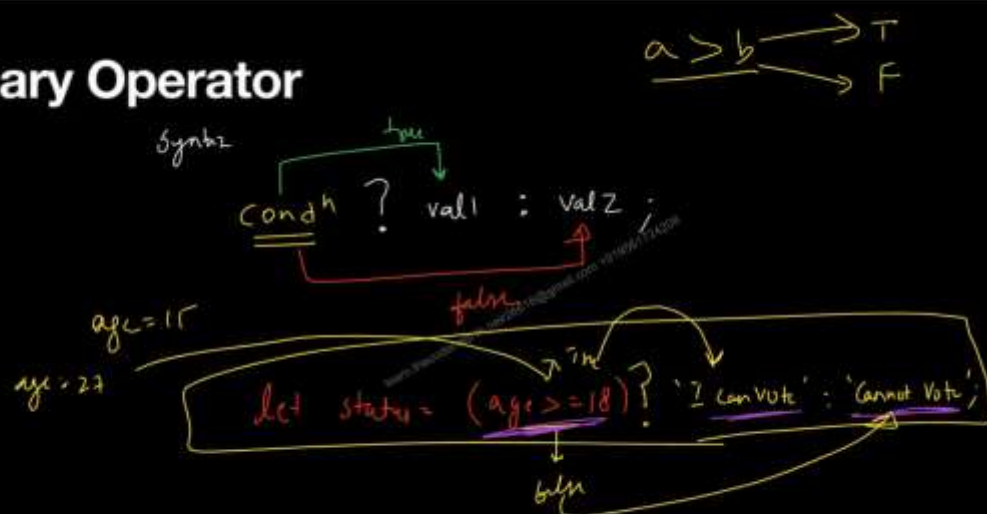


## Equality Operator

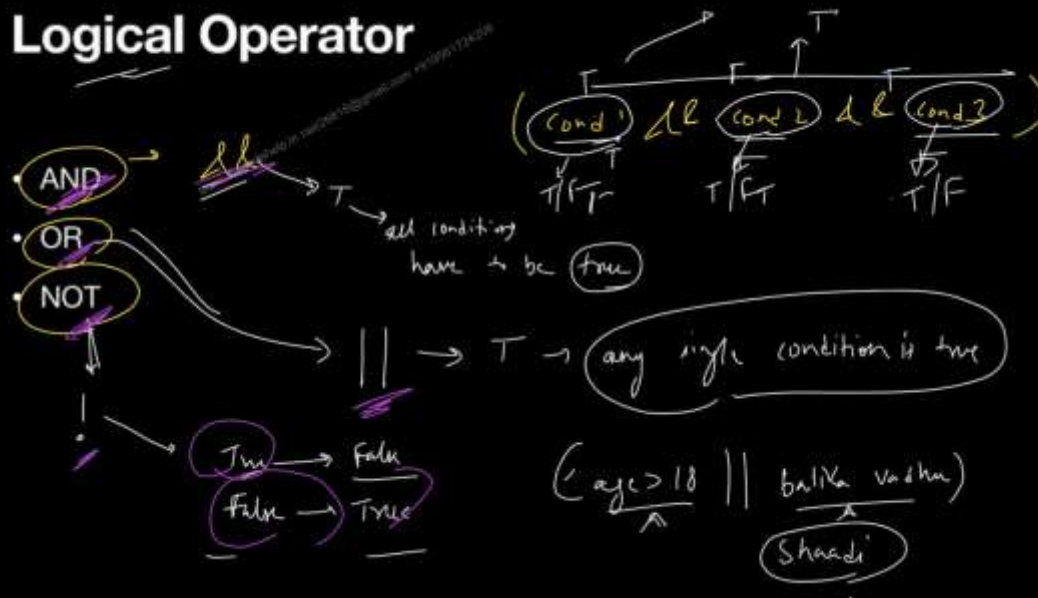
- Loose equality
- Strict Equality



## Ternary Operator

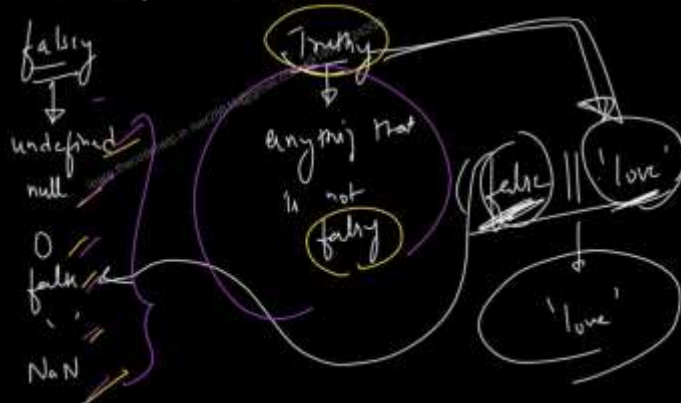


## Logical Operator



## With Non-Booleans

- Concept of False and Truth



$(\underline{\text{True}} \text{ \& \& } \underline{\text{False}})$   
 $\downarrow$   
 $\text{false} \parallel \text{true}$   
 $\downarrow$   
 $\text{true}$

## Bitwise Operator

• Bitwise AND.  $\rightarrow 2$

• Bitwise OR  $\rightarrow 1$

22

A	B	∧
0	0	0
0	1	0
1	0	0
1	1	1

Bits  $\rightarrow 0$   
 $\rightarrow 1$

011

a	b	∨
0	0	0
0	1	1
1	0	1
1	1	1

let  $a=2$   
let  $b=3$

00000010

00000011

2  
01

1  $\rightarrow$  True  
0  $\rightarrow$  false

2 | 3  $\rightarrow$  3

00000011  $\rightarrow$  3

## Operator Precedence

let  $c = a + b * d / c$

Brackets

let  $c = (a + (b * d) / c)$

Page 34 / 59

# Control Statements

- If-else
- Switch

marks  $\geq 50 \rightarrow A$   
 marks  $\geq 50 \rightarrow B$   
 else

www.thecodingladder.in | www.thecodingladder.com | 918947726208

Page 35 / 59

# If-elseif-else

single { if (cond) {  
 }  
 multiple { else if ( ) {  
 }  
 single { else {  
 }

```
// line below is used to print on the console window
console.log('Namaste Dunia version4');

// let a = true;
// console.log(a);

// const num = 12;
// //num = 13;
```

```
// let lastName = 3;
// console.log(lastName);

// lastName = true;
// console.log(lastName);

// console.log(5 !== 3);

// let age = 27;
// let status = (age >= 18) ? 'Yes Vote' : 'No Vote';

// console.log(status);

// let marks = 12;

// if(marks >= 90) {
//     console.log('A');
// }
// else if(marks >= 80) {
//     console.log('B');
// }
// else if(marks >= 70) {
//     console.log('C');
// }
// else if(marks >= 60) {
//     console.log('D');
// }
// else {
//     console.log('E');
// }

// let num = 1;

// switch(num) {
//     case 1: console.log('A');
//     break;
//     case 2: console.log('B');
//     break;
//     case 3: console.log('C');
//     break;
//     default: console.log('D');
// }

// for(let i = 1; i<=5; i=i+1) {
//     console.log("Babbar" + i);
// }

// let x = 1;

// while(x <= 5) {
//     console.log(x);
//     x++;
// }
```



```
let y = 1;

do {
  console.log(y);
  y++;
} while(y < 10);
```