Estrutura Condicional

**If – else – else if**

* **Bloco só é executado se a afirmação for verdadeira**
* **Caso seja falsa é executado o else**
* **Dividido em estruturas simples, compostas e encadeadas**

function resultado(*nota*) {

    if(*nota* >= 7) {

        console.log("Você está aprovado   " + *nota*)

    }

}

// bloco só irá acontecer se a nota for maior ou igual a 7

resultado(8) // só retorna esse valor

resultado(5)

function resultado2(*valor*) {

    if(*valor*) { // pode passar algo que não é boolean, se o valor for verdadeiro ele retorna

        console.log('VERDADE...' +*valor*)

    }

}

resultado2(!!1)

function teste1(*num*) {

    if(*num* > 7)

    console.log(*num*) // se não houver a abertura de chaves apenas a primeira será considerada (dentro do if)

    console.log('Final') // vai ser mostrada independente do resultado do if, pois não está associada a ele

}

teste1(6)

**é recomendado utilizar sempre chaves, mesmo quando houver apenas uma única sentença**

function teste2(*num*) {

    if(*num* > 7); {

        console.log(*num*) // não utilizar ponto e vírgula, pois ele não considera o que está dentro do bloco

    }

}

teste2(6)

teste2(8) // imprime os dois valores, pois o bloco não está associado ao if por conta do ( ; )

const imprimirResultado = function(*nota*) {

    if(*nota* >= 7) {

        console.log('Aprovado')

    } else {

        console.log('Reprovado')

    }

}

imprimirResultado(8) // retorna aprovado

imprimirResultado(4) // retorna reprovado

imprimirResultado('OBS')

**tomar cuidado com isso, pois como o JS é fracamente tipado. Ele vai comparar a string atribuída a nota e retornar false, ou seja, executa o else.**

const imprimirResultado = function(*nota*) {

    if (*nota*.entre(9, 10)) {

        console.log('Testando')

    } else if (*nota*.entre(7, 8.99)) {

        console.log('teste 2')

    } else {

        console.log('teste 3')

    }

}

**Chamada de estrutura encadeada, um elemento dentro do outro**

**Stwich**

* **Seleção múltipla**
* **Não é relacional, não retorna verdadeiro ou falso**
* **Parecida com a estrutura if-else, mas tem o comportamento de executar todos os comandos**
* **É utilizado o break para que pare de executar**
* **Compara tipo e valor**

const imprimirResultado = function (*nota*) {

    switch (Math.floor(*nota*)) {

        case 10: case 9:

            console.log('Quadro de Honra') // entra nessa linha de código tanto se o case for 10 ou 9

**case para cada número, é a única forma, não existe abreviação. EX: case 1, 2, 3.**

case 6: case 5: case 4:

            console.log('Recuperação')

**quando o valor for 4, 5 ou 6 vai mostrar ‘Recuperação’**

 break

**utilizar abaixo de uma condição, para que caso seja essa o switch não execute as demais. Se não utilizá-lo ele executa todas condições de baixo.**

default:

            console.log('Nota inválida')

**comparado ao else da estrutura if-else, caso não seja nenhum dos acima ele aparece**

**pode ficar em qualquer posição**

**Estrutura de Repetição**

**While**

* **Utilizado normalmente quando não se tem uma quantidade determinada de vezes**
* **Executa o bloco enquanto a condição for verdadeira**

**Sintaxe:**

while(condição) {

    instruções

}

**Do While**

* **Mostra o elemento pelo menos uma vez antes de testar a condição, obrigatoriamente**

**Sintaxe:**

do {

    instruções

} while(condição) // verifica se deve continuar ou não

**For**

* **Repete um bloco uma determinada quantidade de vezes**
* **Pode ser usada pra acessar os índices de uma array**

**Sintaxe:**

for(declaração da variável; condição; incrementador)

for(declaração da variável; array.length ; incrementador)

**For/IN**

* **Acessa os indices da array**
* **Não é a melhor forma de usar**

const notas = [3.2, 5.4, 6.7, 3.6]

for (let i in notas) {

    console.log(i, notas[i]) // se colocar apenas o [i] não acessa o conteúdo do índice

} // acessa os índices da array

for(let atributo in pessoa) {

    console.log(`${atributo} = ${pessoa[atributo]}`)

}

console.log(atributo)

**Break e continue**

* **Não influencia num bloco do tipo if/else, apenas no switch, for e while**
* **Continue só faz sentido agir em estruturas de repetição**
* **Evitar o uso, porque pode trazer mais confusão**

const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

for (x in numbers) {

    if (x == 5) {

        break // não age em cima do bloco if, age em cima do bloco associado

        // interrompe o laço e sai pra próxima sentença de código fora do laço que ele está relacionado

    }

    console.log(`${x} = ${numbers[x]}`)

}

for (y in numbers) {

    if (y == 5) {

        continue // interrompe a execução atual em cima do laço em que está relacionado e vai para próxima, ou seja, quando o índice for 5 ele pula pro próximo índice

    }

    console.log(`${y} = ${numbers[y]}`)

}

externo: for (a in numbers) {

    for (b in numbers) {

        if(a == 2 && b == 3) break externo // se colocar apenas o break ele vai agir apenas dentro do bloco interno. Para dar um break em tudo foi criado um rótulo externo

    }

}

**NÃO UTILIZAR**