



Compilado - Aula 06

- Vimos na última aula que dentro do `if` podemos fazer **comparações** para o código ser executado. Por exemplo, uma comparação pode ser `nota >= 7`
- Quando trabalhamos com condicionais e booleanos, uma situação muito comum é querermos levar mais de uma comparação em consideração ao mesmo tempo. Por exemplo, se a regra pra alguém ser aprovado for: **ficar com pelo menos 7 de média e ter menos de 10 faltas**. Teríamos um algoritmo mais ou menos assim:

Algoritmo aprovacao(recebe nota1, nota2, nota3 e faltas) - calcular media usando as notas - **se a media for pelo menos 7 E TAMBÉM as faltas forem menores que 10** - responder aprovado - senao (caso qualquer uma das comparações acima forem falsas) - responder reprovado

- Fazer cada comparação já sabemos: `media >= 7` é a primeira e `faltas < 10` é a segunda. Mas como fazer aquele "e também" ali do meio? É simples, podemos usar o operador do JavaScript: `&&` (dois "e-comerciais" juntos, chamamos esse operador de AND).

```
function aprovacao(nota1, nota2, nota3, faltas) { let media = (nota1 +  
nota2 + nota3) / 3; if(media >= 7 && faltas < 10) { return "Aprovado";  
} else { return "Reprovado"; } }
```

- Podemos usar a técnica de armazenar o resultado das comparações em variáveis e deixar nosso código mais legível:

```
function aprovacao(nota1, nota2, nota3, faltas) { let media = (nota1 +
nota2 + nota3) / 3; let aprovadoPorMedia = (media >= 7); // Vai
armazenar true ou false let aprovadoPorPresenca = (faltas < 10); //
Vai armazenar true ou false if(aprovadoPorMedia &&
aprovadoPorPresenca) { // Um pouco mais fácil de ler a regra :) return
"Aprovado"; } else { return "Reprovado"; } }
```

- Repare que o operador `&&` nos dá verdadeiro somente se ambas as condições forem verdadeiras. Se uma delas ou ambas forem falsas, ele dará falso.
- Existe também um outro operador pra um caso muito comum que é quando você tá interessado(a) que pelo menos uma condição seja verdadeira. Por exemplo, imagina que o aluno pode ser classificado pra uma faculdade em duas situações: caso tire pelo menos 8 no vestibular da faculdade ou pelo menos de 700 no ENEM:

```
function classificado(notaVestibular, notaEnem) { let
classificadoVestibular = (notaVestibular >= 8); let classificadoEnem =
(notaEnem >= 700); if(classificadoVestibular || classificadoEnem) {
return "Classificado"; } else { return "Desclassificado"; } }
```

- Esse operador `||` (duas barras verticais coladas, chamado de OR) retorna verdadeiro quando pelo menos 1 das comparações forem verdadeiras. Ele só dará falso se ambas derem falso, ou seja, se o aluno não passou nem pelo vestibular nem pelo ENEM.
- Por fim, um último operador booleano muito útil é quando precisamos inverter um booleano, ou seja, transformar `true` em `false` ou vice-versa. O operador que faz isso no JS é a exclamação `!`, chamado de NOT. Por exemplo:

```
function aprovado(nota1, nota2, faltas) { let media = (nota1 + nota2)
/ 2; let aprovadoPresenca = (faltas < 10); if(!aprovadoPresenca) { //
Leia como: "se NÃO aprovado por presença". Ou seja, se
aprovadoPresenca for false, vira true e entra no if e vice-versa
return "Reprovado por presença"; } ... }
```