# Contrôle du flow avec les conditions

* On peut ajouter des conditions pour réaliser différentes actions dans Ruby selon le cas à l'aide des mots-clés if, else, elsif :

if trajet\_minutes > 120

puts "Prends un film avec toi"

elsif trajet\_minutes < 5

puts "Pas le temps de t'asseoir..."

else

puts "Tu as le temps de lire au moins quelques chapitres d'un bouquin !"

end

* On peut combiner des conditions à l'aide des signes &&  (pour "ET") et || (pour "OU").
* Les booléens sont des variables utiles pour tester si une condition est vraie ou fausse. Dans l'exemple ci-dessous, code\_indenté est un booléen qui vaut "true" ou "false".

if code\_indenté

puts "Bravo, ton code est bien indenté !"

else

puts "Pense à bien indenter ton code pour améliorer sa lisibilité !"

end

* Pour vérifier une condition d'égalité, il faut utiliser deux signes égal (==). Faites attention à ne pas confondre avec le signe égal tout seul (=) qui sert à affecter une valeur à une variable :

age = 18 #range la valeur 18 dans la variable age et renvoie donc 18

age == 18 #booléen qui renvoie True si age vaut 18, False sinon