

variable = condition ? **value1** : **value2**;



If condition is true,
variable is assigned value1

Execução Condicionada

Apontamentos sobre as condições na linguagem Java: if, else if, else, switch

Page

- Os programas vistos até agora são executados sequencialmente, mas a maioria dos programas não pode ser assim:
 - Há sempre condições especiais que é necessário acautelar
 - O utilizador pode usar opções diferentes de cada vez que executa um programa
- É necessário que existam instruções que permitam mudar o curso normal de um programa numa dada condição
- Em Java uma condição pode ser:
 - Um valor booleano, ou uma expressão cujo resultado é um valor booleano
- Os operadores que retornam valores booleanos são:

Operador	Descrição	Exemplo	Resultado
=	Igual a	num1 == num2	Se num1 é igual a num2, é true, senão é false
!=	Diferente de	num1 != num2	Se num1 é diferente de num2, é true, senão é false
>	Maior que	num1 > num2	Se num1 é maior que num2, é true, senão é false
>=	Maior ou igual a	num1 >= num2	Se num1 é maior ou igual a num2, é true, senão é false
<	Menor que	num1 < num2	Se num1 é menor que num2, é true, senão é false
<=	Menor ou igual a	num1 <= num2	Se num1 é menor ou igual a num2, é true, senão é false

JavaScript ▾

```
int num1 = 10, num2 = 3;
```

```

System.out.printf("O resultado de %d == %d é %b", num1, num2, num1 == num2);    // o resultado de 10 == 3 é
false
System.out.printf("O resultado de %d != %d é %b", num1, num2, num1 != num2);    // o resultado de 10 != 3 é
true
System.out.printf("O resultado de %d > %d é %b", num1, num2, num1 > num2);      // o resultado de 10 > 3 é
true
System.out.printf("O resultado de %d >= %d é %b", num1, num2, num1 >= num2);   // o resultado de 10 >= 3 é
true
System.out.printf("O resultado de %d < %d é %b", num1, num2, num1 < num2);      // o resultado de 10 < 3 é
false
System.out.printf("O resultado de %d <= %d é %b", num1, num2, num1 <= num2);   // o resultado de 10 <= 3 é
false

```

If

- A instrução `if` é uma instrução que permite controlar o fluxo de execução de um programa
 - A sintaxe é:
 - `if (condição) instrução;`
 - Se a condição for verdadeira, a instrução é executada
 - Se a condição for falsa, a instrução não é executada
 - A condição tem de ser sempre um valor do tipo `boolean`

```

JavaScript ▾
int num;

// calcular o valor de num

if (num < 0)
    num=-num;                      // Se num = 123 -> num = 123 ; se num = -14 -> num = 14

System.out.printf("num = %d", num);

```

```

JavaScript ▾
char ch;

// calcular o valor de ch

if (ch >= 'a')
    ch -= 32;                      // Se ch = 'D' -> ch = D ; se ch = 'g' -> ch = G

System.out.printf("ch == %c", ch);

```

- A instrução a seguir ao `if` pode ser um bloco de código

```

JavaScript ▾
if (condição) {
    instrução1;
    instrução2;
}

```

- Isto é válido para qualquer instrução em Java

```

JavaScript ▾
int num;

// calcular o valor de num

if (num < 0) {
    System.out.printf("Introduziu um valor negativo!\nIrá ser convertido para positivo!\n");
    num=-num;                      // Se num = 113 -> num = 113 ; se num = -14 -> Introduziu um valor
negativo!num = 14
}

```

```
System.out.printf("num = %d", num);
```

- O teste pode envolver mais que uma condição
 - Neste caso são necessários operadores booleanos
 - Os operadores booleanos são os seguintes:

Operador	Descrição	Exemplo	Resultado
<code>&&</code>	E (and)	<code>cond1 && cond2</code>	Se <code>cond1</code> e <code>cond2</code> , forem <code>true</code> , o resultado é <code>true</code> , senão é <code>false</code>
<code> </code>	Ou (or)	<code>cond1 </code>	Se <code>cond1</code> ou <code>cond2</code> , forem <code>true</code> , o resultado é <code>true</code> , senão é <code>false</code>
<code>!</code>	Negação (not)	<code>!cond</code>	Se <code>cond</code> , for <code>false</code> , o resultado é <code>true</code> , senão é <code>false</code>

```
JavaScript ▾
char ch;

// calcular o valor de ch

if (ch >= 'a' && ch <= 'z') {
    System.out.printf("Introduziu uma letra minúscula!\nVai ser convertida para maiúscula!");
    ch -= 32; // Se ch = 'D' -> ch = D ; se ch = 'g' -> Introduziu uma letra minúscula! ch
            = G
}
System.out.printf("ch == %c", ch);
```

- Se uma condição for suficiente para decidir o resultado as seguintes não são verificadas

```
JavaScript ▾
char ch;

// calcular o valor de ch

if (ch >= 'a' && ch <= 'z') { // se ch < 'a' não vai verificar a 2ª condição
    System.out.printf("Introduziu uma letra minúscula!\nVai ser convertida para maiúscula!");
    ch -= 32;
}
System.out.printf("ch == %c", ch);
```

- A instrução tem outra sintaxe que é:

```
JavaScript ▾
if (condição)
    instruçãoV; // bloco de código
else
    instruçãoF; // bloco de código
```

- Se a condição for verdadeira, a `instruçãoV` é executada
- Se a condição for falsa, a `instruçãoF` é executada

- Exemplos de uso do `if-else`;

JavaScript ▾

```
int num;

// calcular num

if( num < 0 )
    System.out.printf("número é negativo" );
else
    System.out.printf("número é positivo" );

// Supondo num = 123 -> número é positivo
// Supondo num = -14 -> número é negativo
```

JavaScript ▾

```
int num1, num2;

// calcular num1 e num2

if( num1 > num2 )
    System.out.printf("o primeiro número é maior");
else
    System.out.printf("o segundo número é maior");

// Supondo num1 = 123 e num2 = 5 -> O primeiro número é maior
// Supondo num1 = 1, num2 = 5 -> O segundo número é maior
```

- O segundo programa poderia ser melhor:
 - E se os números fossem iguais?
 - Como se faz para testar 3 condições?
- Podem-se usar `if-else` encadeados

JavaScript ▾

```
int num1, num2;

if( num1 == num2 )
    System.out.printf("os números são iguais");
else if( num1 > num2 )
    System.out.printf("o primeiro número é maior");
else
    System.out.printf("o segundo número é maior");

// Supondo num1 = 123 e num2 = 5 -> O primeiro número é maior
// Supondo num1 = 1, num2 = 5 -> O segundo número é maior
// Supondo num1 = 5 e num2 = 5 -> Os números são iguais
```

⚠ Quando se encadeiam `if-else` o `else` é sempre referente ao último `if`. Para evitar isto podem-se usar as `{ }`

Operador Ternário - `? :`

- O operador `? :` é semelhante a um `if-else` mas deve ser usado apenas em atribuições
- A sintaxe é:

JavaScript ▾

```
condição ? expressãoV : expressãoF;
```

- Se a condição for verdadeira, o resultado é dado por `expressãoV`
- Se a condição for falsa, o resultado é dado por `expressãoF`

JavaScript ▾

```
int num1, num2;
int maior = num1 >= num2 ? num1 : num2;

System.out.printf("O maior número é o %d", maior);
```

Switch Case

- A instrução deve ser usada apenas quando se prevêem várias hipóteses de resposta
- A sintaxe é:

JavaScript ▾

```
switch (expressão) {
    case constante1 : instrução1;
    case constante2 : instrução2;
    case constante3 : instrução3;
    // ...
    default: instrução;           // opcional
}
```

- O controlo é passado para a instrução que corresponde ao valor da expressão, caso haja
- Se não houver correspondência o controlo passa para a instrução `default` (se houver)

JavaScript ▾

```
char estadoCivil;

// ...

switch( estadoCivil ) {
    case 's': System.out.println("solteiro");
    case 'c': System.out.println("casado");
    case 'v': System.out.println("viúvo");
    case 'd': System.out.println("divorciado");
    case 'u': System.out.println("união de facto");
    default: System.out.println("valor errado!");
}

// Supondo estadoCivil = 't' -> valor errado
// Supondo estadoCivil = 'u' -> união de facto valor errado
// Supondo estadoCivil = 'c' -> casado viúvo divorciado união de facto valor errado
```

- O controlo é passado para a instrução que corresponde ao valor da expressão, caso haja, depois a execução segue na linha seguinte
- Quando se quer terminar a execução dentro do `switch`, basta usar o `break`

JavaScript ▾

```
// Exemplo do switch corrigido

char estadoCivil;

//...

switch( estadoCivil ){
    case 's':
        System.out.println("solteiro");
        break;

    case 'c':
        System.out.println("casado");
```

```

        break;

    case 'v':
        System.out.println("viúvo");
        break;

    case 'd':
        System.out.println("divorciado");
        break;

    case 'u':
        System.out.println("união de facto");
        break;

    default:
        System.out.println("valor errado!");
}

// Supondo estadoCivil = 'c' -> casado
// Supondo estadoCivil = 'v' -> viúvo
// Supondo estadoCivil = 'C' -> valor errado!

```

JavaScript ▾

```

// Se quisermos fazer com que o programa aceite, tanto letras minúsculas como maiúsculas...

char estadoCivil;

// ...

switch( estadoCivil ){
    case 's':
    case 'S':
        System.out.printf("solteiro");
        break;

    case 'c':
    case 'C':
        System.out.printf("casado");
        break;

    case 'v':
    case 'V':
        System.out.println("viúvo");
        break;

    case 'd':
    case 'D':

// ...

}

// Supondo estadoCivil = 'c' -> casado
// Supondo estadoCivil = 'C' -> casado
// Supondo estadoCivil = 't' -> valor errado!

```