**DART**

**CONTROL FLOW STATEMENTS**

**IF STATEMENTS**

**IF** é uma **keyword** usada quando queremos determinada ação mas apenas se os **valores preencherem certos requisitos.**

Por exemplo:

**int age = 21;**

**if (age >= 18) {**

**print('ADULT');**

Também temos o **else/else if** se quisermos por outras condições caso o **valor não chegue nos requisitos da primeira condição.**

Por exemplo:

**int age1 = 14;**

**if (age1 >= 21) {**

**print('Adult');**

**} else if ((age1 <= 18) & (age1 >= 13)) {**

**print('teenager');**

**}**

**else  {**

**print('Child');**

**}**

Podemos usar o **IF** para comparar valores, utilizando operadores relacionais

Por exemplo

if (age1 == 14) {

  print('It´s equal');

 } else {

  print('It´s not euqal');

 }

Para **comparar** se os valores são iguais utilizamos **== ( não utilizar o = já que ele é usado para atribuir valor)** e se são valores diferentes usamos !=.

OBS : o sinal ! Pode ser visto como not , aplicarmos ele em um algoritmo cujo o resultado é verdadeiro os algoritmo ira nos mostrar falso e vice - versa

IF também pode ser utilizado com a variável bool

Por exemplo:

**bool isAllowed = false;**

**int age3 = 20;**

**if (age3 != 18 && !isAllowed) {**

**print('YESSSS');**

**} else {**

**print('NOOOOO');**

**}**

Nesse caso temos um bom exemplo do operador **!** usado para negativar tanto o valor quanto outros operadores **(!= por exemplo)**

**Operadores**

| **Description** | **Operator** | **Associativity** |
| --- | --- | --- |
| unary postfix | *expr*++    *expr*--    ()    []    ?[]    .    ?.    ! | None |
| unary prefix | -*expr*    !*expr*    ~*expr*    ++*expr*    --*expr*      await *expr* | None |
| multiplicative | \*    /    %  ~/ | Left |
| additive | +    - | Left |
| shift | <<    >>    >>> | Left |
| bitwise AND | & | Left |
| bitwise XOR | ^ | Left |
| bitwise OR | | | Left |
| relational and type test | >=    >    <=    <    as    is    is! | None |
| equality | ==    != | None |
| logical AND | && | Left |
| logical OR | || | Left |
| if-null | ?? | Left |
| conditional | *expr1*    ?    *expr2*    :    *expr3* | Right |
| cascade | ..    ?.. | Left |
| assignment | =    \*=    /=   +=   -=   &=   ^=   *etc.* | Right |
| spread ([See note](https://dart.dev/language/operators#spread-operators)) | ...    ...? | None |

Minhas observações

**isEmpty/isNotEmpty** , podemos usar essa função para decidir se certa variável esta vazia ou não.

.**SartsWith** , pode ser usada para descobrir se o valor da variável começa com ‘x’ letra.

Exemplo: **if (somevalue.startsWith('h')) {**

**print('Corect');**

**} else {**

**print('try again');**

**}**

Outra maneira de escrever esse algoritmo seria **sintaxe abreviada** seria:

**String value = somevalue.startsWith('H') ? 'WOW' : 'naha';**

**print(value);**

Esta é a forma a qual chamamos de **Ternary**, seria escrever o código em apenas uma sentença **(?)** marca o código como pergunta no próximo campo podemos por entre **(’’)** o que queremos dizer caso seja verdadeiro e no segundo caso seja falso a divisão entre os dois tem que ser marcado por **(:)** .

Dependendo do resultado a nossa variável vai receber os valores colocados entre **(’’)**.

**SWITCH STATEMENTS**

Podemos usar **SWITCH** para por casos para verificar uma variável .

Por exemplo

**// Switch statement**

**switch (somevalue) {**

**case 'Hi':**

**print('Hello!');**

**case 'Hi!':**

**break;**

**case 'Hi!!':**

**print('OUUUUUUU');**

**default:**

**print('NOOOOOOOOOO');**

Quando não quisermos determinar uma ação podemos usar **BREAK**.

**WHEN**

Podemos usar **when** para adicionar mais condicionantes

Por Exemplo:

**switch (somevalue) {**

**case 'Hi!!' when age <= 20:**

**print('YEEEP');**

**case 'Hi':**

**print('It is');**

**case 'Hi!!!':**

**break;**

**default:**

**print('Nothing');**

Exercicio

**/\* Exercise :**

**Develop a program to calculate the shipping cost based on**

**the destination zone and the weigth of the package**

**Calculate the shipping cost accorting to the these conditions:**

**If the destination zone is 'XYZ', the shipping cost is $5 per kilogram.**

**If the destination zone is 'ABC', the shipping cost is $7 per kilogram.**

**If the destination zone is 'PQR', the shipping cost is $10 per kilogram.**

**If the destination zone is not 'XYZ', 'ABC', or 'PQR', display an error message. \*/**

**void main() {**

**String destinationZone = 'WQE';**

**double packageW = 8.5;**

**double shippingP = 0;**

**switch (destinationZone) {**

**case 'XYZ':**

**shippingP = (packageW \* 5);**

**case 'ABC':**

**shippingP = (packageW \* 7);**

**case 'PQR':**

**shippingP = (packageW \* 10);**

**default:**

**print('--------------------------------------------');**

**print('The acurrent location is not unavailable');**

**print('--------------------------------------------');**

**return;**

**}**

**print('--------------------------------------------');**

**print('The shipping price for $destinationZone is  $shippingP ');**

**print('--------------------------------------------');**

**}**

Obs : para formar um **print** com variáveis temos de por **$ para identificar a variável**.

Se queremos que o código termine em determinada condição basta usar **return** como não temos nada para retornar o código ira parar.

**Loops**

**For**

Usamos o for para iniciar um **loop** e tem a seguinte estruturar.

**for (init; condition; increment/decrement;)**

Por exemplo:

**for (int i = 0; i < 10; i++) {**

**print('Hello World ${i+1}');**

// i++ -> i = i + 1

 // i-- -> i = i - 1

OBS: Normalmente contagens em programação e na língua do computador começam em zero.

Pela seguinte usabilidade a baixo :

**for (int i = 0; i < 12; i++) {**

**String hi = 'Hello World';**

**print(hi.substring(0, i));**

Neste caso a contagem começa em 0 porque caso fosse por 1 iriamos pular uma letra.

Podemos também por o numero de letras da string como referencia para repetições

Por exemplo:

**String value = 'I´m here';**

**for (int i = 0; i < value.length; i++) {**

**print(value[i]);**

**While**

While é uma forma de loop , usado quando o numero de interações já está determinado.

Por exemplo :

**String value = 'I do it';**

**int i = 0;**

**while (i < value.length) {**

**print(value[i]);**

**i = i + 1;**

**DO While continuar**

| **Critério** | **while** | **for** |
| --- | --- | --- |

| **Uso principal** | É utilizado quando não se sabe previamente o número de iterações; depende de uma condição ser verdadeira. | Usado quando o número de iterações é conhecido ou pode ser facilmente controlado. |
| --- | --- | --- |

|  | while (condição) { | for (inicialização; condição; incremento) { |
| --- | --- | --- |

|  | // Código a ser executado | // Código a ser executado |
| --- | --- | --- |

| **Condição de parada** | Avalia a condição antes de cada iteração; se for falsa, o loop termina. | A condição é avaliada antes de cada iteração; se falsa, o loop para. |
| --- | --- | --- |

| **Controle da iteração** | Não possui estrutura embutida para inicialização ou incremento; você deve gerenciar manualmente. | Permite configurar inicialização, condição e incremento na própria estrutura do loop. |
| --- | --- | --- |

|  | int i = 0; | for (int i = 0; i < 5; i++) { |
| --- | --- | --- |

|  | while (i < 5) { | print(i); |
| --- | --- | --- |

|  | print(i); | } |
| --- | --- | --- |

|  | i++; |  |
| --- | --- | --- |

| **Leitura do código** | Pode ser menos intuitivo, especialmente com condições ou incrementos complexos. | Geralmente mais legível para loops simples e iterativos. |
| --- | --- | --- |

| **Iteração em coleções** | Não é ideal para percorrer coleções como listas; requer manipulação manual de índices ou iteradores. | Pode usar for-in para iterar diretamente sobre coleções. Exemplo: for (var item in lista) { ... }. |
| --- | --- | --- |