**///Dart 1st warm-up Exercise///**

***Name:-ABHIJEET KUMAR***

void main() {

print("Hello World");

print("This is my first application");

}

**Operators in Dart**

**1) Multiplicative :- \* / % ~/**

**eg:- 1**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1\*num2;

print("${result}");

}

**eg:- 2**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1/num2;

print("${result}");

}

**eg:- 3**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1%num2;

print("${result}");

}

**eg:- 4**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1~/num2;

print("${result}");

}

**2) Additive :- + -**

**eg:- 1**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1+num2;

print("${result}");

}

**eg:- 2**

void main() {

var num1=10;

var num2=20;

var result=0;

result=num1-num2;

print("${result}");

}

**3) Unary postfix :- expr++ expr--**

**eg:- 1**

void main() {

var num1=10;

var result=0;

result=num1++

print("${result}");

}

**eg:- 2**

void main() {

var num2=20;

var result=0;

result=num2--;

print("${result}");

}

**4) Unary prefix :- -expr !expr ~expr ++expr --expr await**

**eg:-1**

void main() {

var num2=20;

var result=0;

result=--num2;

print("${result}");

}

**eg:-2**

void main() {

var num1=20;

var result=0;

result=++num1;

print("${result}");

}

**eg:-3**

void main() {

var num1=10;

var num2=-20;

var result=0;

result= -num2+num1;

print("${result}");

}

**eg:-4**

void main() {

var num1=true;

if(!num1==false){

print("hello");

}

}

**eg:-5**

void main() {

var num1=20;

if(~num1==20){

print("hello");

}

}

**eg:-6**

Future hello() async {

print('Hi,there');

}

void main() async{

await hello();

print('Enjoy Flutter and Dart');

}

**5) Shift :- << >> >>>**

**eg:-1**

void main() {

var num1=001001;

print(num1 >> 3);

}

**eg:-2**

void main() {

var num1=001001;

print(num1 << 3);

}

**6) Bitwise AND :- &**

**eg:-1**

void main() {

var num1=5;

var num2=8;

print(num1 & num2);

}

**7) Bitwise XOR :- ^**

**eg:-1**

void main() {

var num1=5;

var num2=8;

print(num1 ^ num2);

}

**8) Bitwise OR :- |**

**eg:-1**

void main() {

var num1=5;

var num2=8;

print(num1|num2);

}

**9) Relational and type test :- >= > <= < is is! as**

**eg:-1**

void main() {

var num1=12;

var num2=8;

if(num1>=num2){

print("greater than or eqaul");

}

}

**eg:-2**

void main() {

var num1=12;

var num2=8;

if(num1>num2){

print("greater than");

}

}

**eg:-3**

void main() {

var num1=12;

var num2=12;

if(num1<=num2){

print("less than or eqaul");

}

}

**eg:-4**

void main() {

var num1=4;

var num2=8;

if(num1<num2){

print("less than");

}

}

**eg:-5,6**

void main()

{

String a = 'hello';

double b = 8.8;

print(a is String);

print(b is !int);

}

**10) Equality :- == !=**

**eg:-1**

void main() {

var num1=4;

var num2=8;

print(num1==num2);

}

**eg:-2**

void main() {

var num1=4;

var num2=8;

print(num1!=num2);

}

**11) Logical AND :- &&**

**eg:-1**

void main() {

var num1=4;

var num2=8;

if(num2>=num1 && num1%2==0)

{

print("Logical AND");

}

}

**12) Logical OR :- ||**

**eg:-1**

void main() {

var num1=4;

var num2=8;

if(num2>=num1 || num1%2==0)

{

print("Logical AND");

}

}

**13) Conditional :- condition ? expr2 : expr3**

**eg:-1**

void main()

{

var a = 5;

var b = 7;

var c = (a < 10) ? "Statement is Correct : "Statement is Wrong";

print(c);

}

**14) if null :- ??**

**eg:-1**

void main()

{

int n;

var d = n ? ? "n has Null value";

print(d);

n = 10;

d = n ? ? "n has Null value";

print(d);

}

**15) Cascade:- .. ?..**

**eg:-1**

class hello {

var a;

var b;

void set(x, y)

{

this.a = x;

this.b = y;

}

void add()

{

var z = this.a + this.b;

print(z);

}

}

void main()

{

// Creating objects of class GFG

hello h1 = new hello();

GFG h2 = new hello();

// Without using Cascade Notation

h1.set(1, 2);

h1.add();

// Using Cascade Notation

h2..set(3, 4)

..add();

}

**16) Assignment :- = \*= /= += -= &= ^=**

**eg:-1**

void main()

{

var num1=10;

var num2=11;

var result=0;

result+=num1+num2; // result\*=num1\*num2;//result/=num1;

print(result);

}

**More Examples:-**

1.>

void main() {

int a = 1;

print(-a);

print(-(-a));

}

2.>

int x; // The initial value of x is null.

x ??= 6;

print(x); // 6

3.>

x ??= 3;

print(x); // 6

4.>

void main() {

var x = 6;

x++;

x++;

print(x);

x--;

print(x);

}

5.>

var x = 6;

x++;

x++;

6.>

void main() {

var r1 = increase(3);

print(r1);

var r2 = increase2(3);

print(r2);

}

int increase(int x) {

return ++x;

}

int increase2(int x) {

return x++;

}

7.>

void main() {

var a = 1;

a = a + 1;

print(a);

a += 5;

print(a);

a \*= 3;

print(a);

}

8.>

void main() {

var a = 10;

var b = 11;

var c = 12;

var add = a + b + c;

var sb = c - a;

var mult = a \* b;

var div = c / 3;

var rem = c % a;

print(add);

print(sb);

print(mult);

print(div);

print(rem);

}

9.>

void main() {

var c = 5 / 2;

print(c);

var d = 5 ~/ 2;

print(d);

}

10.>

void main() {

var x = 3;

var y = 8;

print(x == y);

print(y > x);

if (y > x) {

print("y is greater than x");

}

}

11.>

void main() {

var a = true && true;

var b = true && false;

var c = false && true;

var d = false && false;

print(a);

print(b);

print(c);

print(d);

}

12.>

void main() {

var a = true || true;

var b = true || false;

var c = false || true;

var d = false || false;

print(a);

print(b);

print(c);

print(d);

}

13.>

void main() {

print(!true);

print(!false);

print(!(4 < 3));

}

14.>

void main() {

print(3 < 4);

print(3 == 4);

print(4 >= 3);

print(4 != 3);

}

15.>

void main() {

print(3 < 4);

print(3 == 4);

print(4 >= 3);

print(4 != 3);

}

16.>

void main() {

var vals = [1, 2, 3];

var vals2 = [...vals, 4, 5, 6];

print(vals);

print(vals2);

}

17.>

class Base {}

class Derived extends Base {}

void main() {

var b = Base();

var d = Derived();

print(d is Base);

print(d is! Base);

print(b is Derived);

print(d is Object);

print(d is! Object);

}

18.>

void main() {

print(3 + 5 \* 6);

print((3 + 5) \* 6);

print(!true || true);

print(!(true || true));

}

19.>

void main() {

var j = 0;

j \*= 3 + 1;

print(j);

}