

Null Safety

Variables by default are not nullable ex: `int x=2;`, null safety helps us distinguish between nullable variables and non nullable variables, nullable variables can hold either a non null value or a null value, but non nullable variables must always hold non null value.

Null Safety Types:

- `?` it means a variable can be null ex: `int? x;`
- `??` if null operator it checks if a variable is null and assigns an error message if it is, ex:

```
String? name1;  
  
String name2;  
  
if (name1 != null) { name1 = name2; }  
  
else  
  
{name2= 'Error'; } print(name2); }
```

- `??=` null aware assignment operator if you have a single nullable variable and you want to assign it a value ex:

```
result=?? 'Error'
```

- `?.` null aware access and method invocation ,if you have a nullable string variable to access the property and the method of the string type, you need to check if it is not null without using the if statement ex: `String? input;`
`print(input?.length); // null`
`print(input?.toLowerCase()); // null`
- `!` null assertion operator, its written after the variable and it means we are certain

that this variable is not null ex:

```
String? name="Mona"
```

```
print(name?.length);
```

- `?.` null cascade operator , if the object of the class exists access the property
ex: `var student=Student()`

```
?.name="Mona";
```

- `?[]` null aware index operator similar to the null cascade operator but it is used for indexes of lists ,access of certain index of a list if it is not null.

```
ex: List<int>? numbers;
```

```
print(numbers?[0]);
```

- `...?` null aware spread operator, what spread operator usually does is adding multiple values to a collection so, if I have more than one list and I want to make one of these lists has the values of the other list plus its own values, null aware spread operator check if the item exists to spread the list if not it won't spread the list.

```
ex: List<int>? numbers;
```

```
var numbers2=[...?numbers,3,9];
```

```
print(numbers2)
```

