**TYPESCRIPT TYPE GUARDS**

* The **Type Guards** in typescript is the idea to put a check for the type of the arguments or the properties to avoid error at runtime.
* The *‘property’ in object* and *instance instanceof Class* are some popular type guards.
* //? Type Guards.
* //\* It is just a term that defines the idea of checking the type before performing a operation.
* //\* It is mainly used in union type to ensure that we always get proper result.
* // In case of Object
* *type* Tree = {
* name: *string*,
* height: *number*,
* getTreeName: () *=>* *string*
* }
* *const* tree: Tree = {
* name: 'Christmas tree',
* height: 10,
* getTreeName() {
* return this.name
* }
* }
* *type* Vehicle = {
* name: *string*,
* speed: *number*,
* getVehicleName: () *=>* *string*
* }
* *const* vehicle: Vehicle = {
* name: 'Hayabusa',
* speed: 200,
* getVehicleName() {
* return this.name;
* }
* }
* *const* printProperties = (*prop*: Tree | Vehicle) *=>* {
* console.log(`The name of the item is ${*prop*.name}`);
* if ('height' in *prop*) console.log(`The height of item is => ${*prop*.height}`); //? The type guards
* if ('speed' in *prop*) console.log(`The speed of item is => ${*prop*.speed}`); //? The type guards
* }
* printProperties(tree);
* printProperties(vehicle)
* //In case of Classes
* *class* Sky {
* getSkyColor() {
* console.log(`The sky is blue.`);
* }
* }
* *const* sky = new Sky();
* *class* Land {
* getLandColor() {
* console.log(`The land is brown.`);
* }
* }
* *const* land = new Land();
* *const* printColorDetails = (*prop*: Sky | Land) *=>* {
* if(*prop* instanceof Sky) *prop*.getSkyColor();        //? The type guard
* if(*prop* instanceof Land) *prop*.getLandColor();      //? The type guard
* }
* printColorDetails(sky);
* printColorDetails(land)