Types

- **any** - default javaScript behaviour (type inference based on assignment)
  - last resort
  - leave noImplicitAny : true
- **unknown -** better than any type
  - TS will not allow any operation with such type
- **boolean**
  - null, undefined can not be assigned to boolean values in TS
- **number**
  - integers, floats
- **bigInt**
  - es2020 onwards
  - BIgInt(number || string)
    - created bigInt with 'n' character
  - 78n
  - only Integer can be assigned. no floats
  - Math obj does not work on bigInt
- **string**
- **object {}**
- **optional** = ?
- **Date**
- **type declaration** end with ';'
- **type Aliases** = { }
- **Union** - firstType | secondType
  - One or Two or Both
- **Intersection Type** - & (takes all properties)
  - All properties of Both types, common property will be one
  - All unique properties, combined with common properties
- **Index signatures** - object keys and value types
- **Arrays** - type [ ]
  - Good to have only one type of array
- **Tuple** - [string, string] fixed size array along with types
  - We can optional properties in Tuple, (?) but it must be defined at the end
  - Helpful in creating heterogenous list
- **readonly** - can not be changed later on.
  - We can create tuples, arrays
- **null -** no value
  - if we try to access DOM elements which does not exists, we will get 'null' as value
- **undefined -** not defined
  - When variables values are not defined.
- TS provides protections against undefined, null values
- We can use '!' for telling TS, that value will not be 'null'

- **void** - returns nothing, does complete
- **never** - never completes
- **enum**
    - By default numerical values

Type Inference

- TS can infer types on basis of initial assignment
- Best practice to let TS infer values on its own
- Explicitly declare types where its needed