**Create a new TypeScript Program**

* Every Typescript program must have the extension “.ts”
* Go to “src” folder
* Add a new file “main.ts”

let username:string = "John";

document.write("Hello ! " + username);

* Transcompile “main.ts” into “JavaScript File”
  + Open “src” folder Terminal
  + Run the command  
    **> tsc main.ts**
  + It will generate JavaScript file **“main.js”**
* Link the JavaScript file to your **Index.html** page

**<script src=”../src/main.js”> </script>**

**TypeScript Programs without HTML Page**

* Add a new File into “src” – hello.ts

let msg:string = "Hello ! Welcome to TypeScript";

console.log(msg);

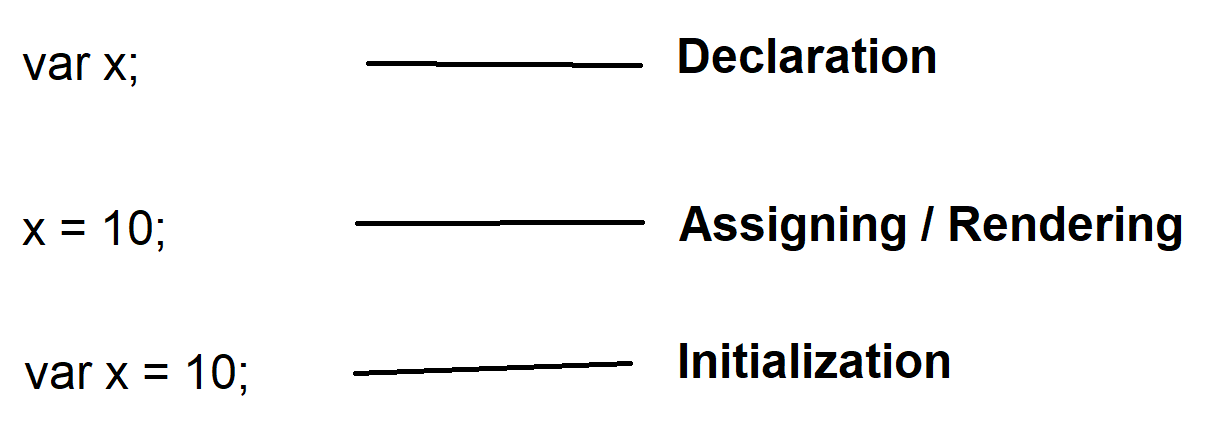
* Transcompile   
  **> tsc hello.ts**
* Run the JavaScript file using node compiler  
  **> node hello.js**

**TypeScript Language Basics**

* Variables
* Data Types
* Operators
* Statements

**Variables**

* Variables are storage locations in memory where you can store a value and use it as part of any expression.
* Variable Configuration comprises 3 phases
  + Declaration
  + Assigning / Rendering
  + Initialization



* In JavaScript declaring variable is not mandatory, if it is not in strict mode.
* In strict mode declaring variable is mandatory.
* In TypeScript declaring variable is mandatory.

**Declare Variables**

* TypeScript variables are same like JavaScript variables, declared by using
  + var
  + let
  + const
* **Declare by using “var”**
  + It defines a function scope for variable.
  + You can declare in any block of function and access from any another block in the function.
  + Var allows
    - Declaring
    - Assigning
    - Initialization
  + Var allows **shadowing**: It is a technique of re-declaring same name identifier within the scope.
  + Var allows **hoisting:** It is a technique of using variable before declaring.
* **Declare by using “let”**
  + It defines a block scope for variable.
  + It is accessible only the block where it is declared or initialized.
  + It allows
    - Declaring
    - Assigning
    - Initialization
  + It will not allow **shadowing.**
  + It will not allow **hoisting.**
* **Declare by using “const”**
  + It defines block-scoped variable.
  + It allows only “**Initialization”.**
  + It will not allow declaring and assigning.
  + It will not allow shadowing.
  + It will not allow hoisting.
* **Global Scope for Variable**
  + If you want to access a variable across various functions then you can define at Global Scope.
  + Module Scope is known as Global Scope.
  + You can declare a global variable with
    - var
    - let
    - const

//Module Scope

let x = 10;

var y = 20;

const z = 30;

function f1(){

console.log(`f1 x=${x} y=${y} z=${z}`);

}

function f2(){

console.log(`f2 x=${x} y=${y} z=${z}`);

}

f1();

f2();

**FAQ: Can we define a global variable inside function?**

1. **Yes with “browser -window object”**

function f1(){

window.x = 10;

}

function f2(){

document.write("x=" + x);

}

f1();

f2();

* **Variable Naming conventions**
  + Variable name can be up to 255 chars long.
  + Variable name must start with alphabet or \_ [underscore]
  + Variable name can be alpha numeric, but can’t start with number or any special character other than \_
  + No Blank spaces, no special chars, only \_ allowed

Ex:

var \_productName = “TV”;

var product\_name = “TV”;

var product123 = “TV”;

* + Always use “Camel Case” for naming “txtName”

**Data Type**