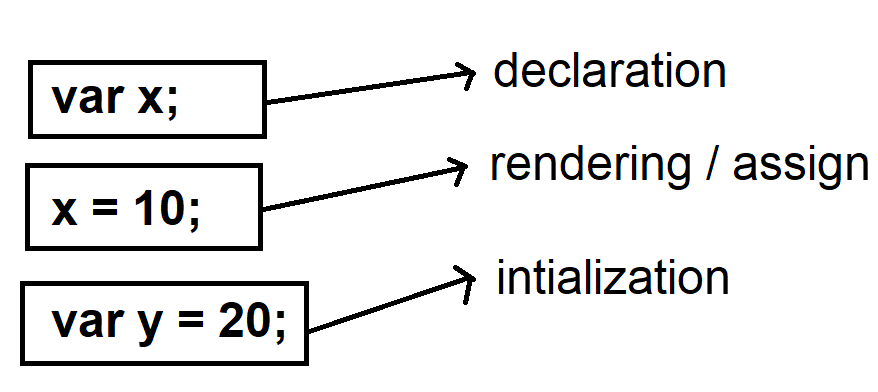
**TypeScript Language Basics**

* Variables
* Data Types
* Operators
* Statements

**Variables**

* Variables are simply storage locations in memory, where you can store a value and use it as a part of any expression.
* JavaScript allows to use variable directly without declaration if it is not in strict mode.
* TypeScript is by default in strict mode of JavaScript.
* Declaring variable in TypeScript is mandatory.
* Variable configuration comprises of 3 stages:
  + Declaration
  + Rendering or Assignment
  + Initialization



* Variables in TypeScript are declared by using following keywords
  + var
  + let
  + const

|  |  |
| --- | --- |
| **var** | * It defines a function scope variable. * You can declare variable in any block inside the function access from any location in the function. * Var supports declaration, rendering and initialization.   Ex:  **Add a new program “variables.ts”**  function f1(){  var x; // declaring  x = 10; // rendering  if(x==10)  {  var y = 20; //initialization  }  console.log(`x=${x}\ny=${y}`);  }  f1();  **Open Terminal / Command Prompt**  **> tsc variables.ts**  **> node variables.js**   * Var allows shadowing. * Shadowing is the process of re-declaring a variable within the scope. * It can have same name identifier re-defined in the scope.   Ex:  function f1(){  var x; // declaring  x = 10; // rendering  if(x==10)  {  var y = 20; //initialization  **var y = 40; // shadowing**  }  console.log(`x=${x}\ny=${y}`);  }  f1();   * Var allows hoisting.   Ex:  function f1(){  x = 10;  console.log(`x=${x}`);  var x; // hoisting  }  f1(); |
| **let** | * It is used to define block scope variable. * It can be accessed only in the block where it is declared. * It allows declaration, rendering, initialization. * It will not allow shadowing. * It will not allow hoisting. |
| **const** | * It is used to define block scope variable. * It will not allow declaration and rendering. * It will allow only initialization. * It will not allow shadowing * It will not allow hoisting. |

**FAQ: What is shadowing?**

**FAQ: What is hoisting?**

**FAQ: What is rendering, assign, declaring and initialization?**

**Global Scope for Variable:**

* A variable declared in module scope is considered as Global Variable.
* You can declare global variable by using var, let or const.
* Module is a collection of components like classes, functions, variables etc.

Ex:

//Module Scope

var x = 10;

let y = 20;

const z = 30;

function f1()

{

// function scope

console.log(`Function-1 X=${x} Y=${y} Z=${z}`);

}

function f2()

{

//function scope

console.log(`Function-2 X=${x} Y=${y} Z=${z}`);

}

f1();

f2();

**Variable Naming Conventions:**

* Variable name must start with an alphabet or \_ under score.
* Variable name can be alpha numeric. It can’t start with number.
* Variable name can’t contain blank space, periods [.] or any special character only \_ allowed.
* Variable name must use camel case. [txtName, btnSubmit]
* Variable name length can be maximum 255 chars.

**TypeScript Data Types**

* JavaScript is implicitly typed language.
* The data type of variable is determined according to the value assigned.
* JavaScript variables are not strongly typed. [Can’t restrict the type].
* TypeScript is strongly typed.
* You can define data type explicitly for variable so that it can handle only the specified type.
* **It supports type inference.**
* If you are not specifying the data type explicitly then typescript can configure the type of **variable according to value initialized** and will restrict only that type of value.   
  Syntax:

**let variableName: datatype = value;**

**What is Data Type?**

* Defines the data structure.
* Data Structure defines the type of value and size of value.
* Data Types are classified into 2 types
  + **Primitive Types**
  + **Non-Primitive Types**

**Primitive Types**

* Primitive types are Immutable types.
* The structure of data can’t be changed according to state and situation.
* They have fixed range for value.
* They use stack memory. [LIFO – Last-In-First-Out]
* TypeScript Primitive Types are
  + number
  + string
  + null
  + undefined
  + boolean

**Non-Primitive Type**

* Non-Primitive Type are Mutable types.
* They structure can change according to state and situation.
* They don’t have any fixed range for values.
* Values range changes according to the memory available.
* They use memory heap. [Allows to access random]
* TypeScript Non-Primitive types are
  + Array
  + Object
  + Regular Expression

**All data types in typescript are derived from a base type called “any”.**

let x : any = anyTypeofValue;

let x : any = 10; x : number; - type inference

x = “A”; // invalid

let x:any;

x = 10;

x = “A”;

**Primitive Data Type**

**Number Type**

* It is defined by using “number”.
* A number value can be
  + Signed Integer  
    let x: number = -10;
  + Unsigned Integer  
    let x: number = 10;
  + Floating Point
  + Double
  + Decimal
  + Exponent
  + Binary - **ES5 [ECMAScript 2015]**
  + Octal
  + Bigint
  + Hexadecimal

**Ex:**

let signed:number = -10;

let unsigned:number = 10;

let floating:number = 45.53;

let double:number = 456.670;

let decimal:number = 5.69797978868434; //29 decimal

let exponent:number = 2e3; // 2 x 10[3] = 2000

let binary:number = 0b1010;

let octal:number = 0o744;

let hexa:number = 0xf00d;

console.log(`Exponent=${exponent}\nBinary=${binary}`);

**String Type**

* String is a literal with group of characters.
* String literal comprises of alpha numeric and special characters.
* String literal can be enclosed in
  + Single Quote ‘ ‘
  + Double Quote “ “
  + Back Tick ` ` [ES5]
* Single and Double Quotes are used to swap between inner and outer string.

**Syntax:**

let home:string = "<a href='home.html'>Home</a>";

let about:string = '<a href="about.html">About</a>';

* Back tick [`] is used to define a string with embedded expression.
* Expression can be embedded into string by using “${}”. It is allowed only for backtick.

**Ex:**

let username:string = "John";

let age:number = 22;

let year:number = 2020;

let msg:string = "Hello !" + " " + username + " " + "you will be" + " " + (age+1) + " " + "in" + " " + (year + 1) + ".";

let newmsg:string = `Hello ! ${username} you will be ${age+1} in ${year+1}.`;

console.log(msg);

console.log(newmsg);

**Ex:**

let title:string = "User Login";

let template:string =

`

<h2>${title}</h2>

<dl>

<dt>User Name</dt>

<dd><input type="text"></dd>

<dt>Password</dt>

<dd><input type="password"></dd>

</dl>

<button>Login</button>

`;

document.write(template);

**Note: A string representation with special characters may escape printing of certain characters. To print the non-printable characters in a string you have to use “\”.**

Ex:

let path:string = "\"C:\\Images\\shoe.jpg\"";

console.log(`Path=${path}`);

\n – new line – console, alert, confirm

<br> new line

**String Formatting and String Manipulations**

* TypeScript uses all JavaScript string functions to manipulate string.
* String Manipulation Methods
  + charAt()
  + charCodeAt()
  + concat()
  + endsWith()
  + startsWith()
  + indexOf()
  + lastIndexOf()
  + match()
  + localeCompare()
  + replace()
  + slice()
  + substring()
  + trim() etc.
* String Formatting Methods
  + bold()
  + italic()
  + sup()
  + sub()
  + fontsize()
  + fontcolor()
  + toUpperCase()
  + toLowerCase() etc.

Ex: charCodeAt()

* **Add a new file “string.ts”**

//Password must start with Uppercase Letter

let password : string = prompt("Enter Password");

if(password.charCodeAt(0)>=65 && password.charCodeAt(0)<=90){

document.write("Password Verified..");

} else {

document.write("Error: Password must start with an Uppercase Letter");

}

* **Trans compile**> tsc string.ts
* **Create a new HTML file “home.html”**

<!DOCTYPE html>

<html>

<head>

<title>Login</title>

<script src="string.js"></script>

</head>

<body>

</body>

</html>

Ex: match()

let pwd:string = "john123A";

let regExp:any = /(?=.\*[A-Z])\w{4,15}/;

if(pwd.match(regExp)){

console.log("Strong Password");

} else {

if(pwd.length<4){

console.log("Poor Password");

} else {

console.log("Weak Password");

}

}

Ex: split(), trim(), toUpperCase(), toLowerCase(), substring()

* Password.ts

**function SubmitClick(){**

**let userPwd:string = document.getElementById("txtPwd").value;**

**if(userPwd.trim()=="admin123") {**

**document.write("Success..");**

**} else {**

**document.write("Invalid Password");**

**}**

**}**

* Home.html

**<!DOCTYPE html>**

**<html>**

**<head>**

**<script src="password.js"></script>**

**</head>**

**<body>**

**Your Password:**

**<input type="password" id="txtPwd">**

**<button onclick="SubmitClick()">Submit</button>**

**</body>**

**</html>**

Boolean Types

Null Type

Undefined Type