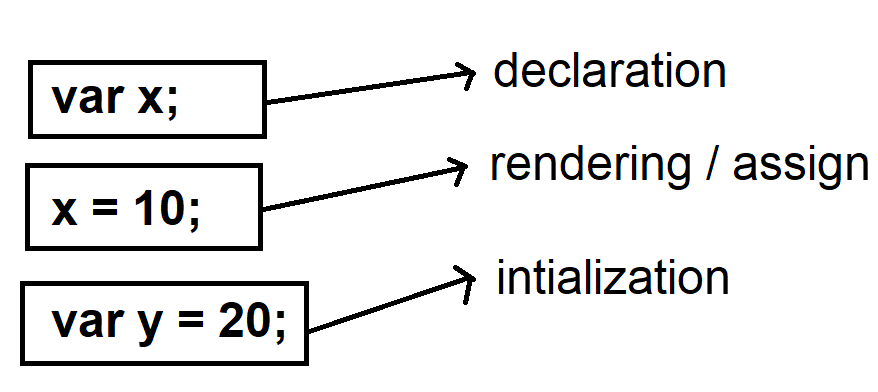
**TypeScript Language Basics**

* 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. |