**JAVASCRIPT QUESTIONS**

1. **What is event bubbling and capturing in JavaScript?**

Event bubbling is series of events when: An event is triggered on the inner most element first and then it triggers on the parent elements in nested order.

DOM elements could be nested inside each other. The handler of the parent works even if you click on its child. Event bubbling can make this happen.

Event Capturing is the opposite of bubbling where the event is triggered on the outer most element and then it triggers the inner children in nested order.

**(2) What is a closure in JavaScript?**

JavaScript allows you to declare a function within another function, and the local variables still can be accessible even after returning from the function you called.

In other words, a closure is the local variables for a function which is kept alive after the function has returned.

**(3) What is a callback function in JavaScript?**

A JavaScript callback function is a function that is passed to an another function as an argument. As a result, a callback function is called via an another function.

**(4) What are various methods you’d use to embed JavaScript onto a webpage?**

You can use inline Javascript or embed an external JS file source into your existing webpage to call javascript and execute at run time.

**(5) What is the difference between GET and POST methods in HTML form?**

When you use a GET method, key and values will be appended at the end of the URL as a query string. GET method is not recommended to use when you are passing sensitive data over the internet.

While sending data, the GET method appends the data to the URL. And the length of a URL is limited. Maximum URL length is 2048 characters supported by latest browsers.

POST is used to pass sensitive data as it’s not appended and displayed within the URL. There is no limitation as to how much data can be passed using a POST method.

(6) **What is the difference between “==” and “===”?**

== and === both are equality operators in JavaScript. However, they are both used for different purposes since the functionality is different.

== will check whether the values of two objects are same or not

=== will check the equality and also whether both the objects are of same type or not

**(7) Why do we use isNaN function?**

isNAN will return TRUE if the argument is not a number. You can think of this as isNotANumber then returned value should be true.

**(8) Explain the keyword “this” in JavaScript?**

The keyword “this” in JavaScript points to the current object within your code.

If you have created an Object using “new” keyword, and then if you use “this”; it will point to the Object you have recently created.

“this” keyword is widely used when you want to change values of the current Object with the flow of the user interaction or logical updates to variable within your code.

**(9) What is the difference between SessionState and ViewState in JavaScript?**

**ViewState**

ViewState is specific persist to a page in a session. Which means when another page is loaded, the data from previous is no longer available.

**SessionState**

SessionState is specific to user specific data within the session, pretty self explanatory. SessionState is the data of a user session and is maintained on the server side. This data available until user closes the browser or session time-outs.

SessionState is the data of a user session and is maintained on the server side. This data available until user closes the browser or session time-outs.

SessionState is the data of a user session which is maintained on the server side. This data available until user closes the browser or session time-outs.

This type of data can be accessed across all pages within the web application.

**1. Difference between window.onload and onDocumentReady?**

The onload event does not fire until every last piece of the page is loaded, this includes css and images, which means there’s a huge delay before any code is executed.  
That isnt what we want. We just want to wait until the DOM is loaded and is able to be manipulated. onDocumentReady allows the programmer to do tha

**3. What does “1”+2+4 evaluate to? What about 5 + 4 + “3”?**

Since 1 is a string, everything is a string, so the result is 124. In the second case, its 93.

**4. What is the difference between undefined value and null value?**

**8.  What is Javascript namespacing? How and where is it used?**

Using global variables in Javascript is evil and a bad practice. That being said, namespacing is used to bundle up all your functionality using a unique name. In JavaScript, a namespace is really just an object that you’ve attached all further methods, properties and objects. It promotes modularity and code reuse in the application.

**9.  What datatypes are supported in Javascript?**  
Number, String, Undefined, null, Boolean

10. **What is the difference between innerHTML and append() in JavaScript?**

InnerHTML is not standard, and its a String. The DOM is not, and although innerHTML is faster and less verbose, its better to use the DOM methods like appendChild(), firstChild.nodeValue, etc to alter innerHTML content.

**HTML 5 QUESTIONS**

**(1) What is the use of Canvas Element in HTML5?**

HTML5 Canvas element can be used to draw graphics images on a web page by using javascript.

**(2) Can you give an example of Canvas element how it can be used?**  
  
<canvas id=“DGTCanvas” width=“500″ height=“400″>  
</canvas>  
<script type=“text/javascript”>  
var DGTCanvas=document.getElementById(“DGTCanvas”);  
var DGTText=DGTCanvas.getContext(“2d”);  
DGTText.fillStyle=“#82345c”;  
DGTText.fillRect(0,0,150,75);  
</script>

This book is far better than any other learning material. It has very basic information that includes both HTML5 and CSS3 with sample code and comprehensive examples.

**(3) What is the purpose of HTML5 versus XHTML?**

HTML5 is the next version of HTML 4.01, XHTML 1.0 and DOM Level 2 HTML. It aims to reduce the need for proprietary plug-in-based rich internet application (RIA) technologies such as Adobe Flash, Microsoft Silverlight, Apache Pivot, and Sun JavaFX. Instead of using those plugins, it enables browser to serve elements such as video and audio without any additional requirements on the client machine.

**(4) What is the difference between HTML and HTML5 ?**

HTML5 is nothing more then upgraded version of HTML where in HTML5 supports the innovative features such as Video, Audio/mp3, date select function , placeholder , Canvas, 2D/3D Graphics, Local SQL Database added so that no need to do external plugin like Flash player or other library elemenents.

**(5) WHAT are some other advantages of HTML5?**

**a)**Cleaner markup than earlier versions of HTML  
**b)**Additional semantics of new elements like <header>, <nav>, and <time>  
**c)** New form input types and attributes that will (and in Opera’s case, do) take the hassle out of scripting forms.

**(6) What is the  <!DOCTYPE>? Is it mandatory to use in HTML5?**

The <!DOCTYPE> is an instruction to the web browser about what version of HTML the page is written in. The <!DOCTYPE> tag does not have an end tag. It is not case sensitive.

The <!DOCTYPE> declaration must be the very first thing in HTML5 document, before the <html> tag.  As In HTML 4.01, all <! DOCTYPE > declarations require a reference to a Document Type Definition (DTD), because HTML 4.01 was based on Standard Generalized Markup Language (SGML). WHERE AS HTML5 is not based on SGML, and therefore does not require a reference to a Document Type Definition (DTD).

**(7) What are the New Media Elements in HTML5?**

New Media Elements in HTML5 are :

|  |  |
| --- | --- |
| TAG | DESCRIPTION |
| <audio> | For multimedia content, sounds, music or other audio streams |
| <video> | For video content, such as a movie clip or other video streams |
| <source> | For media resources for media elements, defined inside video or audio elements |
| <embed> | For embedded content, such as a plug-in |
| <track> | For text tracks used in mediaplayers |

**(8) What is the major improvement with HTML5 in reference to Flash?**

Flash is not supported by major mobile devices such as iPad, iPhone and universal android applications. Those mobile devices have lack of support for installing flash plugins. HTML5 is supported by all the devices, apps and browser including Apple and Android products. Compared to Flash, HTML5 is very secured and protected. That eliminates major concerns that we have seen with Flash.

**(10) What is the sessionStorage Object in html5 ? How you can create and access that?**

The HTML5 sessionStorage object stores the data for one session. The data is deleted when the user closes the browser window. We can create and access a sessionStorage, created “name” as session

<script type=“text/javascript”>  
sessionStorage.name=“DGTECH”;  
document.write(sessionStorage.name);  
</script>

**(1) Tell me different types of storage in HTML5?**

HTML5 can store data locally which is great. It is very secured and fast as well which is an exciting feature of HTML5.

There are two different objects that are able to store data.

* **localStorage object**which stores data for a longer period of time even if the browser is closed.
* **sessionStorage object** which stores data for a specific session.

Normally in non-HTML5 environments it is done via **cookies**.

**(2) What is the difference between localStorage and sessionStorage objects?**

There are two key differences between localStorage and sessionStorage objects.

First, localStorage object stores the data without any expiry date; whereas sessionStorage object stores the data for one session only.  
Second, with localStorage object, data will not be deleted when the browser window is closed. With sessionStorage object, the data is deleted when the browser window closes.

**(3) What HTML4 elements are no longer used (deprecated) in HTML5?**  
Following elements :

* frame
* frameset
* noframe
* big
* applet
* basefront
* center

**(4) What are the new <form> attributes available in HTML5?**

There are two new <form> attributes available in HTML5.

**(i) autocomplete**

* It specifies whether a form or an input field should have autocomplete as on or off.
* If autocomplete is set to **on**, the browser should be able to fill values based on the values filled by the user previously.
* Note that autocomplete works for following input types: **text, search, url, tel, email, password, color, range and datepickers.**

**(ii) novalidate**

* novalidate is a boolean attribute.
* When present, it signifies that the form-data should not be validated when submitted.

**CSS Questions**

**1. What are different ways to apply styles to a Web page?**

**2. How do CSS precedence/cascading rules work? How does the !important directive affect the rules?**

CSS style rules “cascade” in the sense that they follow an order of precedence. Global style rules apply first to HTML elements, and local style rules override them. For example, a style defined in a style element in a webpage overrides a style defined in an external style sheet. Similarly, an inline style that is defined in an HTML element in the page overrides any styles that are defined for that same element elsewhere.

The !important rule is a way to make your CSS cascade but also have the rules you feel are most crucial always be applied. A rule that has the !important property will always be applied no matter where that rule appears in the CSS document. So if you wanted to make sure that a property always applied, you would add the !important property to the tag. So, to make the paragraph text always red, in the above example, you would write:

p { color: #ff0000 !important; }

p { color: #000000; }

## 3. What is a class? What is an ID?

## 4. What is the difference between an ID selector and CLASS?

## 5. What is Contextual Selector?

Contextual selector addresses specific occurrence of an element. It is a string of individual selectors separated by white space (search pattern), where only the last element in the pattern is addressed providing it matches the specified contex

## 6. What is Grouping?

When more than one selector shares the same declaration, they may be grouped together via a comma-separated list; this allows you to reduce the size of the CSS (every bit and byte is important) and makes it more readable. The following snippet applies the same background to the first three heading elements.

h1, h2, h3 {background: red;}

## 7. What are Child Selectors?

A child selector is used when you want to match an element that is the child of another specific element. The parent and child selectors are separated by spaces. The following selector locates an unordered list element within a paragraph element and makes a text within that element bold.

p > ul {font-weight: bold;}

## 8. What are Pseudo Classes?

Pseudo classes allow you to identify HTML elements on characteristics (as opposed to their name or attributes). The classes are specified using a colon to separate the element name and pseudo class. A good example is the :link and :visited pseudo classes for the HTML A element. Another good example is first-child, which finds an element’s first child element.

The following CSS makes all visited links red and green, the actual link text becomes yellow when the mouse pointer is positioned over it, and the text of the first element of a paragraph is bold.

a:link {font-color: red;}  
a:visited {font-color: green;}  
a:hover {font-color: yellow;}

p.first-child {font-weight: bold;}

**9. What are some ways you might target IE (or IE6) only, without affecting other browsers?**

Below are the example for browser specific Style Sheet, which targets defined browser.

#### Target ALL VERSIONS of IE

<!--[if IE]>

<link rel="stylesheet" type="text/css" href="all-ie-only.css" />

<![endif]-->

#### Target everything EXCEPT IE

<!--[if !IE]><!-->

<link rel="stylesheet" type="text/css" href="not-ie.css" />

<!--<![endif]-->

#### Target IE 6 ONLY

<!--[if IE 6]>

<link rel="stylesheet" type="text/css" href="ie6.css" />

<![endif]-->

**10. What’s the difference between an inline element and a block element?**

A block-level element is an element that creates large blocks of content like  paragraphs or page divisions. They start new lines of text when you use them, and can contain other blocks as well as inline elements and text or data.  
  
An inline element is an element that define text or data in the document like STRONG makes the enclosed text strongly emphasized and Q says the enclosed text is a quotation. They don’t start new lines when you use them, and they generally only contain other inline tags and text or data. Or they include nothing at all, like the BR tag.

Some properties that inline elements ignore include:

* width and height
* max-width and max-height
* min-width and min-height

The CSS property display lets you change an inline property to block or a block to inline or either of them to not display at all.

display: block;

display:inline;

display:none;

### **WHEN TO CHANGE THE DISPLAY PROPERTY**

* **Horizontal list menus** – Lists are block-level elements, but if you want your menu to display horizontally, you need to convert the list to an inline element, so that newlines aren’t added between each list item.
* **Headers in the text** – Sometimes you might want a header to remain in the text, but have the HTML header values. Changing the h1-h6 values to inline will let the following text flow next to it.
* **Removing the element** – And of course, if you want to remove the element completely from the document flow, you can set the display to none.

**11. Explain how the CSS box model works. Draw a box and then explain what the border is, what the margin is, what the padding is, etc.? If you assign a width: or height: to something, what specifically does that refer to?**

All HTML elements can be considered as boxes. In CSS, the term “box model” is used when talking about design and layout.

The CSS box model is essentially a box that wraps around HTML elements, and it consists of: margins, borders, padding, and the actual content.

The box model allows us to place a border around elements and space elements in relation to other elements.

[](http://dglobaltech.com/wp-content/uploads/2013/05/boxdim.png)

Explanation of the different parts:

* **Margin** – Clears an area around the border. The margin does not have a background color, it is completely transparent
* **Border** – A border that goes around the padding and content. The border is affected by the background color of the box
* **Padding** – Clears an area around the content. The padding is affected by the background color of the box
* **Content** – The content of the box, where text and images appear

Here is the snippet to draw box model to your web page.

<head>  
<style>  
div.test  
{  
width:220px;  
padding:10px;  
border:5px solid gray;  
margin:0px;  
}  
</style>  
</head>

<body>

<div class=”test”>The picture above is 250px wide. The total width of this element is   also 250px.</div>

</body>

**12. Explain vertical margin collapsing.**

Top and bottom margins of blocks are sometimes combined (collapsed) into a single margin whose size is the largest of the margins combined into it, a behavior known as **margin collapsing**.

Margin collapsing occurs in three basic cases:

**Adjacent siblings**

The margins of adjacent siblings are collapsed (except when the later sibling needs to be [cleared](https://developer.mozilla.org/en-US/docs/CSS/clear) past floats).

**Parent and first/last child**

If there is no border, padding, inline content, or clearance to separate the margin-top of a block with the margin-top of its first child block, or no border, padding, inline content, height, min-height, or max-height to separate the margin-bottom of a block with the margin-bottom of its last child, then those margins collapse. The collapsed margin ends up outside the parent.

**Empty blocks**

If there is no border, padding, inline content, height, or min-height to separate a block’s margin-top from its margin-bottom, then its top and bottom margins collapse.

vertical margins collapse between certain boxes.

To understand how vertical margins collapse lets start with a basic html and CSS example.

## Adjacent Elements With Positive Margin

Vertical adjoining margins collapses. If two elements has positive vertically adjoining margin than only the maximum of both will take effect.

### CSS CODE

|  |  |
| --- | --- |
| 1  2  3 | #parent{ background-color:yellow; width:300px; }  #red {background-color:red; height:50px; margin-bottom:30px;}  #blue {background-color:blue; height:50px; margin-top:20px;} |

### BROWSER RESULT

[](http://960development.com.s142956.gridserver.com/wp-content/uploads/2010/08/02_margin.png)

Margins of [floating](https://developer.mozilla.org/en-US/docs/CSS/float) and [absolutely positioned](https://developer.mozilla.org/en-US/docs/CSS/position) elements never collapse.

There are other situations where elements do not have their margins collapsed:

* inline-block elements
* elements with overflow set to anything other than visible (They do not collapse margins with their children.)
* cleared elements (They do not collapse their top margins with their parent block’s bottom margin.)
* the root element

**13. How do you float an element, and what does that mean? What’s Clearing?**

With CSS float, an element can be pushed to the left or right, allowing other elements to wrap around it.

Float is very often used for images, but it is also useful when working with layouts.

Elements are floated horizontally, this means that an element can only be floated left or right, not up or down.

A floated element will move as far to the left or right as it can. Usually this means all the way to the left or right of the containing element.

Below is the example, which shows using css float property you can set image to the right of your web page.

<head>  
<style>  
img  
{  
float:right;  
}  
</style>  
</head>

<body>

<p>

<img src=”logocss.gif” width=”95″ height=”84″ />

This is some text. This is some text. This is some text.This is some text. This is some text. This is some text.This is some text. This is some text. This is some text.This is some text. This is some text. This is some text.This is some text. This is some text. This is some text.This is some text. This is some text. This is some text.This is some text. This is some text. This is some text.This is some text. This is some text. This is some text.This is some text. This is some text. This is some text.This is some text. This is some text. This is some text.</p></body>

**14. What’s the difference between position: relative, position: fixed & position: absolute?**

Here, I’ve listed basic difference between css positioning properties.

**Position-Relative**. This type of positioning is probably the most confusing and misused. What it really means is “relative to itself”. If you set position: relative;on an element but no other positioning attributes (top, left, bottom or right), it will no effect on it’s positioning at all, it will be exactly as it would be if you left it as position: static; But if you DO give it some other positioning attribute, say, top: 10px;, it will shift it’s position 10 pixels DOWN from where it would NORMALLY be.

**Position-Absolute**. This is a very powerful type of positioning that allows you to literally place any page element exactly where you want it. You use the positioning attributes top, left bottom and right to set the location. Remember that these values will be relative to the next parent element. If there is no such parent, it will default all the way back up to the <html> element itself meaning it will be placed relatively to the page itself.

**Position-Fixed**. This type of positioning is fairly rare but certainly has its uses. A fixed position element is positioned relative to the viewport, or the browser window itself. The viewport doesn’t change when the window is scrolled, so a fixed positioned element will stay right where it is when the page is scrolled.

**15. What does z-index do?**

The z-index property specifies the stack order of an element.

An element with greater stack order is always in front of an element with a lower stack order.

Here is the Example, where using z-index property you can display in front of image.

<head>  
<style>  
img  
{  
position:absolute;  
left:0px;  
top:0px;  
z-index:-1;  
}  
</style>  
</head>

<body>  
<h1>This is a heading</h1>  
<img src=”w3css.gif” width=”100″ height=”140″ />  
<p>Because the image has a z-index of -1, it will be placed behind the text.</p>  
</body>

**REACT Questions**

**Q 2: ReactJS is an MVC based framework?**

**False**

|  |  |
| --- | --- |
| **Q 3: Which of the following concepts is/are key to ReactJS?**   Component-oriented design   Event delegation model   Both of the above | |
| **Q 4: ReactJS focuses on which of the following part when considering MVC?**   M (Model)   V (View)   C (Controller) | |
| **Q 5: Which of the following needs to be updated to achieve dynamic UI updates?**   props   state | |
| **Q 6: Which of the following API is a MUST for every ReactJS component?**   getInitialState   render   renderComponent | |
| **Q 7: 'div' defined within render method is an actual DOM div element**   True   False | |
| **Q 8: Which of the following is used to pass the data from parent to child**   state   props | |
| **Q 9: A component in ReactJS could be composed of one or more inner components**   True   false | |
| **Q 10: JSX transformer is a MUST to work with ReactJS**   True   False | |
| **Q 11: One can define default values for properties, props, using which of the following method?**   getDefaultProps   getPropsValue   getInitialState | |
| **Q 12: PropTypes is used for \_\_\_\_\_\_\_\_**   Determine properties types   Validation | |
| **Q 13: Mixins are used to take care of cross-cutting concerns**   True   False | |
| **Q 14: Life cycle methods of a components fall under following categories?**  Mounting, Unmounting   Mounting, Updating   Mounting, Updating, Unmounting | |
| **Q 15: Pre and post lifecycle methods of components are represented using \_\_\_\_\_\_\_**  Before, After   Will, Did | |
| **Q 16: Which of the following mounting methods is/are invoked before the component is inserted into DOM?**   getInitialState   getInitialState, componentWillMount   getInitialState, componentDidMount   componentWillMount | |
| **Q 17: Which of the following methods change the state of the component?**  forceUpdate   setState   Both of the above   None of the above | |
| **Q 18: Input field that does not supply a value is called as \_\_\_\_\_\_\_\_\_**  Controller component   Uncontrolled component | |
| **Q 19: The default value of input field could be set using which of the following attribute?**  value   defaultValue   Both of the above | |
| **Q 20: The value of textarea is set using its children in the same way like HTML**   True   False | |

|  |
| --- |
| **Q 21: ReactJS renders HTML tags if the element is defined in \_\_\_\_\_\_**   Lowercase   Uppercase |
| **Q 22: ReactJS renders Components (classes) if the element is defined in \_\_\_\_\_\_**   Uppercase   Lowercase |
| **Q 23: To use native javascript as an attribute value, the expression should be wrapped within \_\_\_\_\_**   curly-braces{}   double-quotes |
| **Q 24 : Which of the following is used to trigger a UI update?**   props   state |
| **Q 25: A component can mutate its properties as and when required**   True   False |
| **Q 26: An owner component defines its children component within render method**   True   False |
| **Q 27: A parent component could access or read its children components properties**   True   False |
| **Q 28: Data flow from owner to owned components using which of the following?**   state   props |
| **Q 29: Change or update to large number of nodes is optimized using which of the following techniques?**   Batching   Change detection   Both of the above |

<https://www.codementor.io/reactjs/tutorial/5-essential-reactjs-interview-questions>

## TypeScript Interview Questions

### **Q1. What is TypeScript and Why Do We Need It?**

**Ans.** JavaScript is the only client side language universally supported by all browsers. But JavaScript is not the best designed language. It’s not a class-based object-oriented language, doesn’t support class based inheritance, unreliable dynamic typing and lacks in compile time error checking. And TypeScript addresses all these problems. In other words, TypeScript is an attempt to “fix” JavaScript problems.

TypeScript is a free and open source programming language developed and maintained by Microsoft. It is a strict superset of JavaScript, and adds **optional static typing** and **class-based object-oriented programming** to the language. TypeScript is quite easy to learn and use for developers familiar with C#, Java and all strong typed languages. At the end of day “TypeScript is a language that generates plain JavaScript files.”

As stated on [Typescript official website](http://www.typescriptlang.org/), “TypeScript lets you write JavaScript the way you really want to. TypeScript is a typed superset of JavaScript that compiles to plain JavaScript. Any browser. Any host. Any OS. Open Source.” Where “**typed**” means that it considers the types of variables, parameters and functions.

### **Q2. What are the Benefits of TypeScript?**

**Ans.** TypeScript has following benefits:

* It helps in code structuring
* Use class based object oriented programming
* Impose coding guidelines
* Offers type checking
* Compile time error checking
* Intellisense

### **Q3. What are Different Components of TypeScript?**

**Ans.** There are mainly 3 components of TypeScript:

1. **Language** – The most important part for developers is the new language. The language consists of new syntax, keywords and allows you to write TypeScript.
2. **Compiler** – The TypeScript compiler is open source, cross-platform and open specification, and is written in TypeScript. Compiler will compile your TypeScript into JavaScript. And it will also emit error, if any. It can also help in concatenating different files to a single output file and in generating source maps.
3. **Language Service** – TypeScript language service which powers the interactive TypeScript experience in Visual Studio, [VS Code](http://www.talkingdotnet.com/what-is-visual-studio-code-and-difference-between-visual-studio-2015/), Sublime, the TypeScript playground and other editor.

### **Q4. How Can You Get TypeScript and Install It?**

**Ans.** TypeScript can be installed and managed via npm, the Node.js package manager. To install TypeScript, first ensure the npm is installed properly. And then run the following command to install TypeScript globally on your system.

Hide   Copy Code

npm install -g typescript

TypeScript is included in Visual Studio 2013 Update 2 and Visual Studio 2015 by default. TypeScript also provides support for other editors like [Visual Studio Code](http://www.talkingdotnet.com/what-is-visual-studio-code-and-difference-between-visual-studio-2015/), sublime, Emacs and Vim.

### **Q5. How Do You Compile TypeScript Files?**

**Ans.** The extension for any TypeScript file is **“.ts”**. And any JavaScript file is TypeScript file as it is a superset of JavaScript. So change extension of “.js” to “.ts” file and your TypeScript file is ready. To compile any .ts file into .js, use the following command.

Hide   Copy Code

tsc <TypeScript File Name>

For example, to compile “Helloworld.ts”:

Hide   Copy Code

tsc helloworld.ts

And the result would be helloworld.js.

### **Q6. Is It Possible to Combine Multiple .ts Files into a Single .js File?**

**Ans.** Yes, it's possible. While compiling add --outFILE [OutputJSFileName] option.

Hide   Copy Code

tsc --outFile comman.js file1.ts file2.ts file3.ts

This will compile all 3 “.ts” file and output into single “comman.js” file. And what will happen if you don’t provide a output file name.

Hide   Copy Code

tsc --outFile file1.ts file2.ts file3.ts

In this case, file2.ts and file3.ts will be compiled and the output will be placed in file1.ts. So now your file1.tscontains JavaScript code.

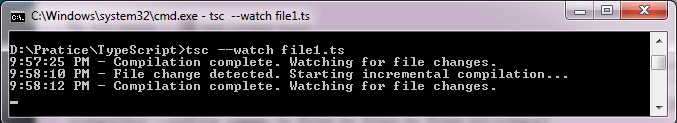
### **Q7. Is It Possible to Compile .ts Automatically with Real Time Changes in .ts File?**

**Ans.** Yes. Using --watch compiler option, this can be achieved.

Hide   Copy Code

tsc --watch file1.ts

This will first compile file1.ts in file.js and watch for the file changes. As soon as there is any change detected, it will compile it again. But you need to ensure that command prompt must not be closed, used with --watchoption.

[](http://www.talkingdotnet.com/wp-content/uploads/2015/12/TypeScript-Watch-option.png)

### **Q8. Does TypeScript Support All Object Oriented Principles?**

**Ans.** The answer is **YES**. There are 4 main principles to Object Oriented Programming: Encapsulation, Inheritance, Abstraction, and Polymorphism. TypeScript can implement all four of them with its smaller and cleaner syntax. Read [Write Object-Oriented JavaScript with TypeScript](http://rachelappel.com/write-object-oriented-javascript-with-typescript).

### **Q9. Which Object Oriented Terms are Supported by TypeScript?**

**Ans.** TypeScript supports the following object oriented terms:

* Modules
* Classes
* Interfaces
* Data Types
* Member functions

### **Q10. Which are the Different Data Types Supported by TypeScript?**

**Ans.** TypeScript supports the following data types:

* Boolean var bValue: boolean = false;
* Number var age: number = 16;
* String var name: string = "jon";
* Array var list:number[] = [1, 2, 3];
* Enum

Hide   Copy Code

enum Color {Red, Green, Blue};

var c: Color = Color.Green;

* Any var unknownType: any = 4;
* Void

Hide   Copy Code

function NoReturnType(): void {

}

### **Q11. How TypeScript is Optionally Statically Typed Language?**

**Ans.** TypeScript is referred as optionally statically typed, which means you can ask the compiler to ignore the type of a variable. Using any data type, we can assign any type of value to the variable. TypeScript will not give any error checking during compilation.

Hide   Copy Code

var unknownType: any = 4;

unknownType = "Okay, I am a string";

unknownType = false; *// A boolean.*

### **Q12. What are Modules in TypeScript?**

**Ans.** Modules are the way to organize code in TypeScript. Modules don’t have any features, but can contain classes and interfaces. It is same like namespace in C#.

### **Q13. What are Classes in TypeScript?**

**Ans.** The concept of classes is very similar to .NET/Java. A Class can have constructor, member variables, properties and methods. TypeScript also allows access modifiers “private” and “public” for member variables and functions.

### **Q14. How Do You Implement Inheritance in TypeScript?**

**Ans.** Using extends keyword, we can implement inheritance.

Hide   Copy Code

class Animal {

public domestic:boolean;

constructor(public name: string) { }

}

class Cat extends Animal {

constructor(name: string, domestic: boolean)

{

super(name);

this.domestic = true;

}

}

class Tiger extends Animal {

constructor(name: string, domestic: boolean)

{

super(name);

this.domestic = false;

}

}

### **Q15. How to Call Base Class Constructor from Child Class in TypeScript?**

**Ans.** Using super(), we can call base class constructor, as seen in the above code.

### **Q16. What is the Default Access Modifier for Members of a Class in TypeScript?**

**Ans.** In TypeScript, each member of class is public by default.

### **Q17. How Can You Allow Classes Defined in a Module to be Accessible Outside of the Module?**

**Ans.** Classed define in a module are available within the module. Outside the module, you can’t access them.

Hide   Copy Code

module Vehicle {

class Car {

constructor (

public make: string,

public model: string) { }

}

var audiCar = new Car("Audi", "Q7");

}

*// This won't work*

var fordCar = Vehicle.Car("Ford", "Figo");

As per above code, fordCar variable will give us compile time error. To make classes accessible outside module, use export keyword for classes.

Hide   Copy Code

module Vehicle {

export class Car {

constructor (

public make: string,

public model: string) { }

}

var audiCar = new Car("Audi", "Q7");

}

*// This works now*

var fordCar = Vehicle.Car("Ford", "Figo");

### **Q18. How Does TypeScript Support Optional Parameters in Function as in JavaScript Every Parameter is Optional for a Function?**

**Ans.** In JavaScript, every parameter is considered optional. If no value is supplied, then it is treated asundefined. So while writing functions in TypeScript, we can make a parameter optional using the **“**?**”** after the parameter name.

Hide   Copy Code

function Demo(arg1: number, arg2? :number) {

}

So, arg1 is always required and arg2 is an optional parameter. Remember, **optional parameters must follow required parameters.**If we want to make arg1 optional, instead of arg2, then we need to change the order andarg1 must be put after arg2.

Hide   Copy Code

function Demo(arg2: number, arg1? :number) {

}

Similar to optional parameters, default parameters are also supported.

Hide   Copy Code

function Demo(arg1: number, arg2 = 4) {

}

### **Q19. Does TypeScript Support Function Overloading as JavaScript Doesn’t Support Function Overloading?**

**Ans.** Yes, TypeScript does support function overloading. But the implementation is odd. When you overload in TypeScript, you only have one implementation with multiple signatures.

Hide   Copy Code

class Customer {

name: string;

Id: number;

add(Id: number);

add(name:string);

add(value: any) {

if (value && typeof value == "number") {

*//Do something*

}

if (value && typeof value == "string") {

*//Do Something*

}

}

}

The first signature has one parameter of type number whereas the second signature has a parameter of type string. The third function contains the actual implementation and has a parameter of type any. The any data type can take any type of data. The implementation then checks for the type of the supplied parameter and executes a different piece of code based on supplier parameter type.

**OR**You can also use union type introduced in TypeScript 1.4. Union types let you represent a value which is one of multiple types.

Hide   Copy Code

add(a:string|number) {

*//do something*

}

Using union type, you can typically remove the need for an overload.

### **Q20. Is It Possible to Debug Any TypeScript File?**

**Ans.** Yes, it is possible. To debug it, you need .js source map file. If you are new to source map, read more [here](http://www.html5rocks.com/en/tutorials/developertools/sourcemaps/). So compile the .ts file with the --sourcemap flag to generate a source map file.

Hide   Copy Code

tsc -sourcemap file1.ts

This will create file1.js and file1.js.map. And last line of file1.js would be reference of source map file.

Hide   Copy Code

//# sourceMappingURL=file1.js.map

### **Q21. What is TypeScript Definition Manager and Why Do You Need It?**

**Ans.** TypeScript Definition Manager (TSD) is a package manager to search and install TypeScript definition files directly from the community driven [DefinitelyTyped](http://definitelytyped.org/" \t "_blank) repository. Let’s see with an example.

Suppose, you want to use some jQuery code in your .ts file.

Hide   Copy Code

$(document).ready(function() { //Your jQuery code });

And now when you try to compile it using tsc, you will get compile time error **Cannot find name “$”.** That’s because TypeScript can’t understand what does “$” means. So somehow we need to inform TypeScript compiler that it belongs to jQuery. That’s where TSD comes into play. You can download jQuery Type Definition file and include it in your .ts file. First, install TSD.

Hide   Copy Code

npm install tsd -g

In your typescript directory, create a new typescript project by running:

Hide   Copy Code

tsd init

Then install the definition file for jquery.

Hide   Copy Code

tsd query jquery --action install

This will download and create a new directory containing your jquery definition file. The definition file ends with “.d.ts”. So now include it by updating your typescript file to point to the jquery definition.

Hide   Copy Code

*/// <reference path="typings/jquery/jquery.d.ts" />*

$(document).ready(function() { *//To Do*

});

Now try to compile again and this time, js will be generated without any error.

So TSD will help you to get type definition file for required framework. If you wish to use angular, then download angular type definition file. You can find the complete list [here](http://definitelytyped.org/tsd/).

### **Q22. What is TypeScript Declare Keyword?**

**Ans.** It’s quite possible that JavaScript libraries/frameworks don’t have TypeScript definition files and yet you want to use them without any errors. Te solution is to use the declare keyword. The declare keyword is used for **ambient declarations** where you want to define a variable that may not have originated from a TypeScript file.

Hide   Copy Code

declare var unKnownLibrary;

TypeScript runtime will assign unKnownLibrary variable any type. You won’t get Intellisense in design time but you will be able to use the library in your code.

### **Q23. How to Generate TypeScript Definition File from Any .ts File?**

**Ans.** You can generate TypeScript definition file from any .ts file via tsc compiler. Generating a TypeScript definition will make your TypeScript file reusable.

Hide   Copy Code

tsc --declaration file1.ts

### **Q24. What is tsconfig.json File?**

**Ans.** The presence of a tsconfig.json file in a directory indicates that the directory is the root of a TypeScript project. The tsconfig.json file specifies the root files and the compiler options required to compile the project. And using this file, we can streamline building TypeScript project. Below is a sample tsconfig.json file.

Hide   Copy Code

{

"compilerOptions": {

"removeComments": true,

"sourceMap": true

},

"files": [

"main.ts",

"othermodule.ts"

]

}

Within files section, define all the .ts files in the project. And when you invoke tsc without any other arguments with the above file in the current directory, it will compile all the files with the given compiler option settings.

### **Q25. What are the Disadvantages of TypeScript?**

**Ans.** Well, TypeScript is great but there are some disadvantages as well.

* TypeScript is just another way to write JavaScript. It doesn’t fix the problems of JavaScript. It just creates an illusion that it does.
* One more tool to learn.
* TypeScript has dependency on type definition files, if you wish to use any existing JavaScript libraries.
* Quality of type definition files is a concern as how can you be sure the definitions are correct?
* Frequent releases of new versions JavaScript library is also a pain area. Because if their type definition files are not updated then you can’t use them instantly.
* In order to run the application in the browser, a compile step is required to transform TypeScript into JavaScript.
* Web developers are using JavaScript from decades and TypeScript doesn’t bring anything new.
* To use any third party library, definition file is you need. And not all the third party library have definition file available.

**1) Explain what is d3.js?**

D3.js is a JavaScript library for creating and manipulating documents based on data.  It uses digital data to drive the formation and control of dynamic and interactive graphical presentation, which runs in web browsers.

**2) When using d3.js is helpful?**

D3.js is extremely helpful in viewing huge data reports of account detail, e-commerce budgeting, population, etc.  For such data’s, data visualization is the best way understand, represent and analyze it.

**3) Explain what is SVG?**

SVG or Scalable Vector Graphics (SVG) is an XML, the markup language for determining two-dimensional vector graphics. SVG is crucial for graphics what XHTML to text.

**4) Explain how D3.js selects method?**

D3.js select method uses CSS3 selectors to choose DOM elements.  D3 looks at the document and choose the first descendant DOM element that consists the tag body.  Once the element is selected, D3.js enables you to implement operators to the element selected.

**5) Explain about d3.js Scales?**

D3.js scales come with

* Quantitative Scales: The quantitative scales have a continuous domain like dates, times, real numbers etc.
* Ordinal Scales: While the ordinal scales are for separate domains like categories, colors, names,
* Linear Scales: It converts one value in the domain interval into a value in the range interval
* Identity Scales: It is good for pixel values
* Power and Logarithmic Scales: It is used for exponentially increasing values like log,pow,sqrt

**6) Mention what are the slider available in d3.js?**

The slider available in d3.js are

* Default slider
* Slider with start value
* Slider with slide event
* Slider with slide event
* Slider with custom axis
* Slider with min, max, and step values
* Vertical Slider

**7) Explain what is Domain in d3.js?**

In d3.js, domain is the start and end of your dataset. It can be any kind of value that can be compared in JavaScript. Domains have to change if your dataset changes.

**8) Explain what is the role of “Path Data Generator” in d3.js?**

In respect to convert our data to the SVG path command, we have to tell the line Path Data Generator, about how to access the y and x co-ordinates from the data.

**9) Mention what does path generators include in it?**

Path generator includes

* svg.line- Make a new line generator
* svg.line.radial- Make a new radial line generator
* svg.area – Make a new area generator
* svg.chord – Make a new chord generator and so on

**10) Explain what d3.js enter method does?**

D3.js enter method returns the **virtual enter selection** from the data operator.  This method is only applicable to Data Operator as such data operator is the only one that returns three virtual selections.

**11) Mention what is the difference between jQuery and d3.js?**

|  |  |
| --- | --- |
| **D3.js** | **JQuery** |
| * D3 creates or manipulates data-driven document that is manipulating or creating visual documents from your data using D3’s data/exit/enter methods * D3 has numerous visualization extensions | * JQuery is a general purpose Ajax/js library which offers general Ajax/js functionalities for creating web apps, but it does not provide data-driven functionality of D3 * jQuery has many general web app extensions |

**12) Explain what is the role of D3.js Axis component?**

D3.js Axis component enables easy addition of a horizontal axis and the vertical axis to any graph. It shows reference lines for D3.js Scales automatically. It also allows you to draw a horizontal axis line, axis ticks and correct spacing to make axis appear appropriate.

**13) Mention the command used to create simple axis in d3.js?**

The command to create simple axis in d3.js is **var xAxis = d3.svg.axis().**

**14) Explain what is SVG group element?**

SVG group element is used to group SVG element together; each SVG group element is a container which consists of child SVG elements.  It is defined by <g> and </g>.

**15) Explain how you can several classes at once?**

To set several classes at once you can use the object literal as

**selection.classed({ ‘foo’:true, ‘bar’: false})**

**16) Explain what is a transition in d3.js?**

Transition in d3.js gradually interpolate attributes and styles over time, transition is used for animation purpose.  It is based on only two key frames, **start,** and **end**.The starting key frame defines the current state of the DOM, while the ending key frame is a set of styles, attributes and other properties you specified.

**17) Mention what is the command to interpolate two objects in d3.js?**

To interpolate two objects in d3.js command **d3.interpolateObject(a,b)**is used. Object interpolation is useful particularly for data space interpolation, where data is interpolated rather than attribute values.

**18) Explain what is the command “d3.ascending (a, b)” is used?**

This command is comparator function that is used for a natural order, and can be used along with the built-in-array sort method to arrange elements in ascending order.

**19) Explain how XML file is called in d3.js?**

By using the command **d3.xml(url[mimeType][,callback])**XML file can be called. This command will create a request for the XML file at the specified *url*. If a call back is declared, the request will be immediately processed with the GET method and the call back will be invoked when the file is loaded, or request fails.

**20) What happens if no call back is specified for XML file in d3.js?**

If no call back is specified, the returned request can be issued using xhr.get and handled using xhr.on.

**21) Mention the command to join the specified array of data in d3.js?**

To join the specified array of data in d3.js you can use the command selection.data([values[,key]]).  The **values** here specifies the data for each group in the selection while a**key** function determines how data is connected to elements.

**22) Mention what does the command d3.csv.parseRows(string[,accessor]) ?**

This command parses the specified string, which is the content of a CSV file, returning an array of arrays representing the parsed rows.

**23) Mention what is the use of “Enter” and “Exit” selection in d3.js?**

By using “Enter” and “Exit” selection in d3.js, you can create new nodes for incoming data and eliminate outgoing nodes that are no longer required.